Different environments and the possibility on counting people by air quality

Arend-Jan van Hilten (4695038), Willem de Laat (4587960), Pim Jansen (4537165)

November 30, 2021

1 Introduction

Houses nowadays are isolated better than they were a couple of decades ago, which has caused natural draft to disappear that first ventilated the homes [1]. Besides this, COVID-19 has made the importance of monitoring the air in spaces even more necessary. Studies have shown that the role of aerosols played an undeniable role in the spread of virus particles [2], [3]. The level (and quality) of aerosols is influenced by temperature and humidity [4].

Moreover, CO_2 levels impact the mental performance of humans when these levels are high. Research shows that humans can mentally perform better when working in an environment with a CO_2 level of around 600 ppm.[5] When humans spend 2 to 3 hours in a closed space with CO_2 levels of about 3000 ppm, their well-being and concentration level decreases.

Note that air quality is a comprehensive definition. Many chemical substances influence air quality, but for the most part, we do not know how badly they affect our health. This paper focuses on the air quality in everyday life inside the home or buildings. In addition, the focus is on harmful substances produced by humans.

Air measurement systems can help gain insights, but above all help take timely action to improve air quality in the room. The goal is to design a low cost and easy to use IoT sensor node that can be placed in houses to monitor the air quality. Eventually, these sensors could turn on the ventilation or open a window with actuators. The integration of such a system has become feasible due to recent developments in IoT and the rise in smart home automation.

First, some background knowledge is presented in the next section. Thereafter, in Section 3, experiments are presented to gain insight in CO_2 levels in different rooms and influence on the placement of the sensor. In addition, an experiment has been performed to be able to estimate the amount of people present in a room. Section 4 describes the algorithm and formula that has been derived from the experiments to estimate number of people in a (closed) space can be estimated with the measurements of our sensor node. Other results are presented in Section 5, including guidelines on the placement based on our research in Subsection 5.3. The design of the product, as well as the software design and structure, will be presented in Section 6.

2 Background Knowledge

2.1 Air quality metrics

The official European Union air quality metric is the Common Air Quality Index (CAQI). This metric takes NO_2, O_3, PM_{10}, CO and SO_2 in consideration and maps the concentration of each gas to a 1-100 scale. The highest score(i.e. the worst value) of all the types of sensors is the resulting air quality index. This metric is used to compare cities [6]. The same gasses can be measured for indoor air quality. CO is the most important of them because it has a direct health impact on humans and can be emitted from faulty boilers and such. For these cases, one should use a CO detector.

The European Commission recognizes that still a lot of research has to be done on the impact of a lot of substances that can be present inside buildings [7]. The focus for them is mainly on identifying how much more harmful one substance is compared to another. The impact of, for example, wall paints on air quality is very high and different, but also very dependent on the raw materials used. The focus is mainly on identifying products that produce unhealthy gasses or particles compared to daily pollutants. Our focus is not on "incidental" contaminants or pollution from outside. Our focus is on air pollution caused by people in a room and by their activities.

A common (collection of) substance in air quality research is volatile organic compounds (VOC's). VOC is an umbrella term for all (carbon-containing) organic substances that evaporate readily into the atmosphere at room temperature. It is therefore not an individual substance like CO or NO_2. Different VOC's, such as benzene and ethers (e.g. used in scent diffusers), can have other impacts on health [8]. Sensors measuring these VOC's mainly report a total VOC (TVOC) value, thus do not specify which specific substances are present. Therefore these TVOC values should be used as a general indicator and not be used for risk assessment, and a high TVOC value does not necessarily indicate a high impact on human health. As a result, governments nor health institutions don't have guidelines for healthy or unhealthy TVOC values.

One might have noticed that CO_2 is not included in the previously mentioned official air quality metrics. The reason is that the index is designed for outdoor pollutants (such as ozone and CO), which are emitted more into the outdoor air and are proportionally more harmful compared to CO_2. Nevertheless, it is an important indoor harmful substance if present in large concentrations. Due to the current climate change, there has been an increase in the isolation of houses. There is a trend towards using as little energy as possible to heat your home. Resulting in deficient ventilation because ventilation lets cold air into the house [1]. As a result, more and more CO_2 accumulates in homes and, consequently, people are quickly exposed to the negative effects of CO_2 concentrations. Recent studies show that indoor CO_2 levels, even as low as 1000 ppm, have an influence on the cognitive functioning of humans. [9] [10] Rooms that are not properly ventilated can quickly reach values above 1000 ppm. A balance must be found between keeping buildings warm in a sustainable way and fresh healthy air. Fortunately, our solution can help.

2.2 Related research

Previous studies mainly focused on small parts of the larger problem. In [11] a low cost occupancy detection system was presented with the aid of PM2.5 sensors. And different sensors (such as humidity, PIR, CO_2 and noise) were evaluated in [12] in order to compare effectiveness in detecting presence and used the ZigBee protocol to transmit the data. Other studies mainly explore the effectiveness of different ML or AI algorithms for occupancy detection based on CO_2 levels [13], [14].

The present work did not combine multiple aspects such as low cost, ease of use, presence detection based on the gradient of CO_2, placement guidelines and a complete (production-ready) product.

3 Experimental setup

This section describes the two experiments performed to derive CO_2 characteristics of different rooms and the impact of humans on the CO_2 in a room. Two sensor nodes will be used during the experiments. Then a detailed explanation of the sensors, core components and software structure of the nodes can be read in Section 6. In short, both nodes contain the same CO_2, TVOC, CO, temperature and humidity sensors. The sensor nodes are connected to the internet. They will collect data of the sensors and send it to a database every three seconds. A visualizer, Grafana, is connected to the database to analyze the data. A global overview of the setup can be seen in Figure 1. By using two sensor nodes, it is possible to examine whether the placement of the device is of importance.

The experiments in this report were held in a student housing in which six students live. Two different experiments were carried out. For the first experiment, the air quality of several rooms within the house was measured to analyze the variation of air quality within the home. The second experiment was to see how the influence of people present in a room affected the CO_2 levels of the room. During the two experiments, two sensor nodes were used, of which one was placed near the ceiling of the room and the other on the ground.

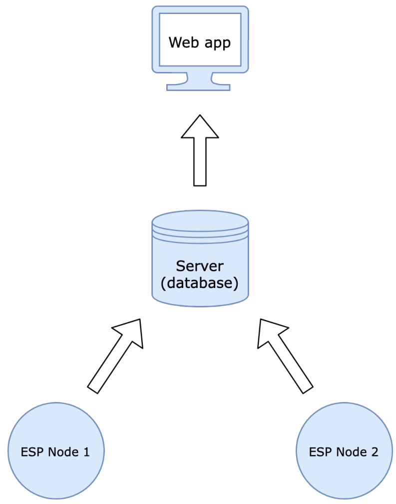


Figure 1: Global setup network topology

3.1 Variation of air quality between rooms

This experiment is done in four different locations in the student house. The sensors were at two different heights in the same room for every experiment. The four rooms in which the air was evaluated were the kitchen, the laundry room, the downstairs hallway and a bedroom (c.q. workspace). In each of the rooms, measurements were taken for at least 24 hours.

The first goal was to determine whether the CO_2 concentrations differ per room. The second goal of this experiment was to determine the difference in height placement in a single room and the best arrangement of a sensor node.

The laundry room is on the first floor and the other three rooms are situated on the ground floor. During the experiment, everyday activities were done, such as eating, working, sleeping, and doing the laundry. Graphical visualization of the rooms can be seen in Figure 2. In each of the figures, the red dotted circle represents a ventilation shaft from which air is drawn out of the room.

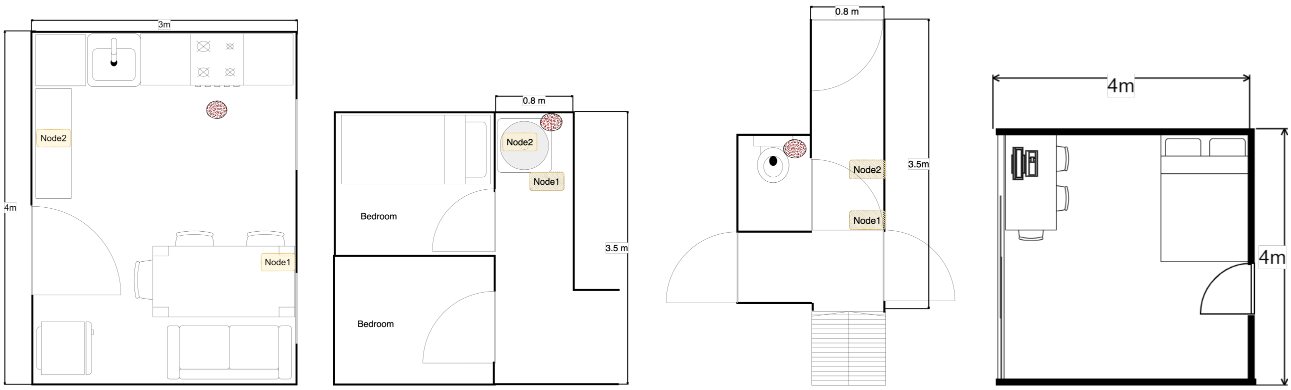
As can be seen from the figures the node placement is shown. During every measurement period, node two was placed at the highest possible location in the room. In Table 1 it can be seen that node two was placed around 2 meters above the ground for every measured room. Node 1 was placed on the floor.

Table 1: Height placement node 2

Location	Height node 2 [cm]
Kitchen	190
Hallway	205
Bedroom	185
Laundry room	175

3.2 Influence of people on CO_2 levels

This experiment aims to derive the impact of people on the CO_2 level in a room. This experiment was conducted in the kitchen as well as in the bedroom, the volume of these rooms are $28.7m^3$ and $67.3m^3$ respectively. By letting up to three people enter, the expec-



(a) Kitchen (b) Laundry room (c) Hallway (d) Bedroom

Figure 2: Visualization different rooms

tation is that the CO_2 emission will vary and therefore the slopes of the increase in the CO_2 level.

The experiment was conducted in the following manner:

1. The room is filled with fresh air. This is done by opening the door and windows.

2. The doors and windows are closed. A 15 minuted wait time is taken so that the air can stabilize.

3. One person enters the room.

4. After 12 min the second person enters.

5. After another 12 min the third person enters.

During the experiment, the ventilation was shut off and the doors and windows remained closed. Therefore the environment is constant with barely any airflow.

4 Estimating people

From the results, it can be seen when people entered the rooms, in Figure 3 and Figure 4 the gradient has been estimated. These values can be seen in Table 2. From this it can be seen that the CO_2 levels can in fact be linked to the number of people in a room. Moreover, the gradient is also correlated to the size of the room. As the kitchen is smaller than the bedroom, the gradient is higher.

The algorithm in Listing 1 is developed to estimate the number of people in the room. This algorithm calculates a linear gradient of the current CO_2 values over the last 5-10 minutes. After this, the gradients of the kitchen are taken and a linear gradient is taken of this to get the relation between CO_2 gradient and the number of people in the kitchen. Then the current CO_2 gradient is plugged into the generated formula to get an estimate. For each room, a different gradient array has to be made to account for the different sizes, rooms' layout, and airflows.

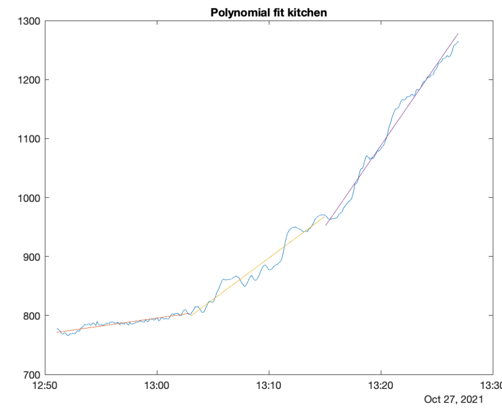


Figure 3: Gradient estimation kitchen

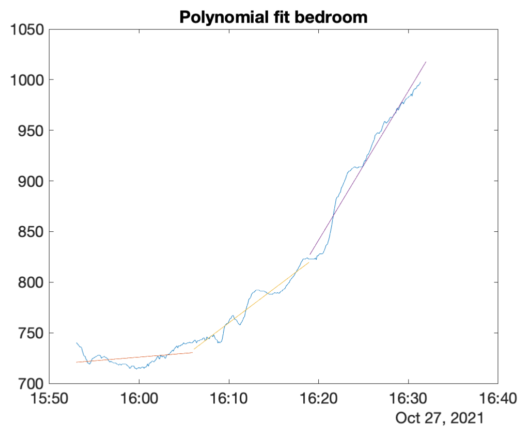


Figure 4: Gradient estimation bedroom

Table 2: Gradient estimation

Room	**Number of people**	**Gradient**
Kitchen	1	2.7ppm/min
	2	14.1 ppm/min
	3	27.5 ppm/min
Bedroom	1	0.7 ppm/min
	2	6.7 ppm/min
	3	14.7 ppm/min

Listing 1 People estimating

```
const gradientsKitchen = [2.7, 14.1, 27.5];
// or bedroom:
const bedroomGradients = [0.7, 6.7, 14.7];


function run(.) {
    const times =...// times of the last 100 co2 measurements
    const values =...// values of these 100 last co2 measurements
    const gradientCurrently = findGradient(times, values);
    if (gradientCurrently.a < 0) { // negative amount?
        return;
    }

    // calculate the gradient of the gradient per people(second derivative)
    const gradientX = [1..N];
    const gradientPerPerson = findGradient(gradientsKitchen, gradientX);
    // second 'derivative' of the values wrt num of people
    const peopleNum = gradientPerPerson.a * gradient + gradientPerPerson.b;
    const exp = Math.floor(peopleNum);
    if (exp > 1 AND exp !== deviceMem.peopleNum) {
        sendNotitification(exp,...);
    }
    deviceMem.peopleNum = exp;
}
```

This algorithm should work in a closed space without much refreshment of the air and the sensor should be placed somewhere in the room. If there is incoming fresh air and outgoing air, the gradient would be lower and it could be even a static value for each number of people. The algorithm was developed on the basis that each person would exhale a standard amount. Therefore there is a linear correlation between the number of people and the amount of CO_2 they exhale. This linear amount is the gradient that the sensor measures and uses for the calculations. The downside is that the sensor has a maximum range it can sense. Therefore, it cannot sense and calculate after some time due to the CO_2 levels being too high. The upside of placing the sensor in a static room compared to putting it in an exhaust is that it is much easier and cheaper to do. Another thing is that an exhaust sensor cannot sense the actual CO_2 level inside the room, only of the exhaust and therefore cannot warn the occupants of dangerous levels.

5 Placement Airbience

The placement of Airbience is a crucial part of its correct functionality. First, a comparison of the air quality will be made between the rooms to determine if there are rooms that have a substantial benefit of the usage of Airbience. These rooms are the kitchen, laundry room, hallway and bedroom (c.q. workspace). Moreover, the ideal height on which Airbience should be placed will be examined.

5.1 Differences in air quality between rooms

For the development and testing of Airbience, samples for a period of longer than 24 hours were taken from four different rooms within the same housing. These experiments found that the bedroom (c.q. workspace) had emissions significantly higher than the other room, even more so than the kitchen. This is visualized in Figure 5[1] in which the confidence intervals of the different rooms are shown. The humidity has not been taken into account in these boxplots since these values did not give a significant difference between the rooms.

During the experiments, it was found that the CO_2 level only comes above 2000 ppm in the kitchen and bedroom. This is no surprise as these are more confined spaces in which humans spend more time, as they do in a laundry room or a hallway. Another remarkable finding was that the temperature in the bedroom was about 1 degree lower than other rooms in the housing were.

These experiments found that Airbience would be the most useful in the kitchen or the bedroom (c.q. workspace), with a strong preference for the bedroom (c.q. workspace). Since this is a room where a lot of time is spent, having a sound air monitoring system would be extra beneficial.

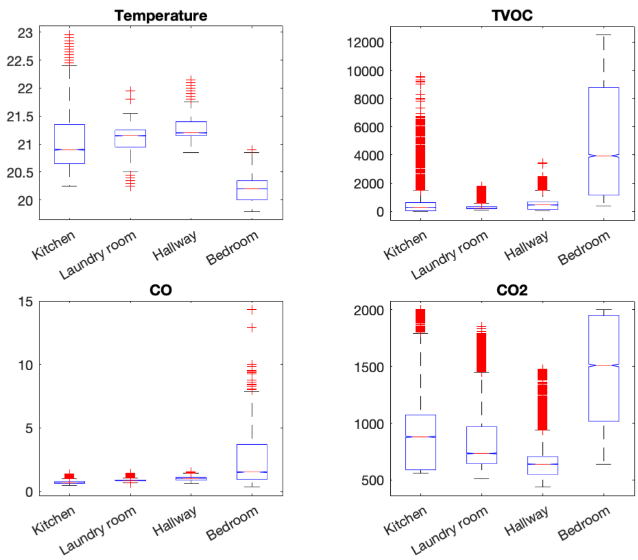


Figure 5: Boxplots of all long-term sensor readings per location

5.2 Height placement Airbience

The height is an essential factor when using sensors that measure the air. Warmer air(from breathing out) will rise and take CO_2 particles with it. Therefore there can be a difference between a lower placed sensor and a higher placed sensor.

[1]The results and measurements that are shown in this section are not taken at the same time.

Table 3: Comparison between rooms.

Room	Temp (°C)	Hum (%)	TVOC (ppb)	CO (ppm)	CO_2 (ppm)
Kitchen	21.03	61.61	530	0.70	909
Laundry	21.07	65.41	299	0.88	818
Hallway	21.26	67.00	491	0.99	649
Bedroom (c.q. work space)	20.21	62.52	4721	2.17	1479

The 24-hour sensor readings are plotted in Figure 6 and show that the higher placed sensors give a higher CO_2 level than the lower placed sensors. This figure also indicates that the laundry room has on average a lower CO_2 level overall than the kitchen, probably because the kitchen was actively used for eating and cooking, where a laundry room would only be used a few minutes at a time, resulting in lower CO_2 levels.

Another interesting thing that shows up when comparing the data from the sensors is that the sensors higher up react earlier than the sensors on the floor and more intensely. This can be seen in Figure 7. The hotter air probably causes this from breathing rising and taking the CO_2 with it, resulting in an earlier and more significant change in the concentration higher up. The time shift is around 2-3 minutes.

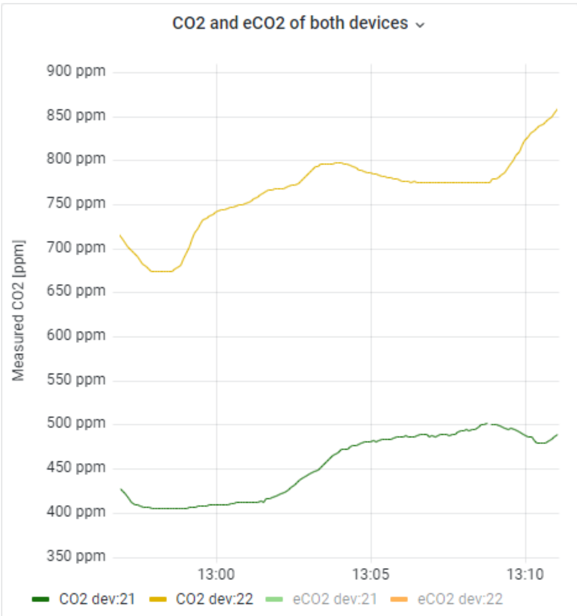


Figure 7: Device 22(higher) starts to rise earlier than device 21(floor)

The same phenomenon happens when the people leave the room and the concentration lowers because of ventilation. Colder air comes in through the window or crack under the door and takes less CO_2 with it and the colder air stays lower. This way, the concentration descends faster on the floor and slower higher up. This can be clearly seen in Figure 8 where the green line falls more quickly than the descent of the yellow line. The earlier start of the drop of the higher sensors can be caused by the movement of the people when leaving and the direct stop of the high concentration before the lower sensor has its equilibrium (higher concentration from above vs. constant ventilation) changed.

The starting concentration and the peak concentration are different between the different placements. The higher sensor reads a standard higher CO_2 level than the lower sensor. This is probably also caused by the same phenomenon of rising air.

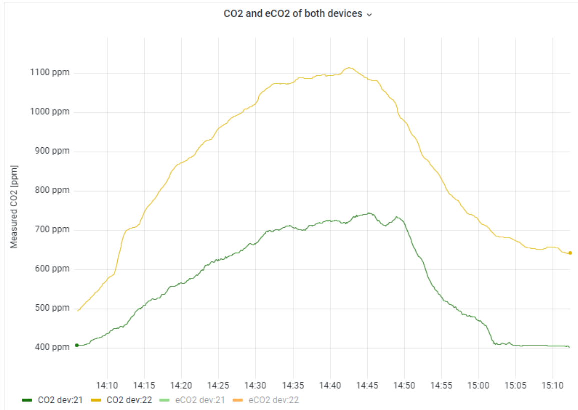


Figure 8: Device 22(higher) has a steeper rise, where device 21(lower) has a steeper fall

5.3 conclusion on the placement of Airbience.

Besides the height placement, other environmental properties could strongly influence the correct operation of Airbience. The following advice is based on the experiments that have been conducted. The device should not be placed near objects that create (extensive) heat or airflow, for example, above a kettle or furnace, below a fridge or next to a window. Placement close to an indoor door should be avoided too, since it can create extra noise on measurements. The noise results from brief, powerful air movements across the sensors, which impacts the reading. Airbience should also not be placed on the ground. When a room gets ventilated (by opening an outside door or window) during winter months, cold (fresh) air shall mainly flow across the floor. Because the fresh cold air is mainly on the floor, Airbience indicates that the temperature and co_2 level have dropped. However, this is only the case locally (on the floor). The air at eye level does not change that quickly and drastically. So to get a better representative picture of the air quality in a room, it is recommended not to place the node on the ground.

In summary, from these findings, it can be deduced that Airbience should not be placed near objects that directly influence the air characteristics. Thus placement in the proximity of windows, stoves, kettles, fridges and doors should be avoided. In addition, one should prefer placement between hip height and ceiling over placing it on the floor. Of course, it is still essential to place Airbience in places where people are present rather than in an unused remote corner. Thus the best spot to place Airbience is where people are often present while keeping as much distance as possible from the aforementioned objects.

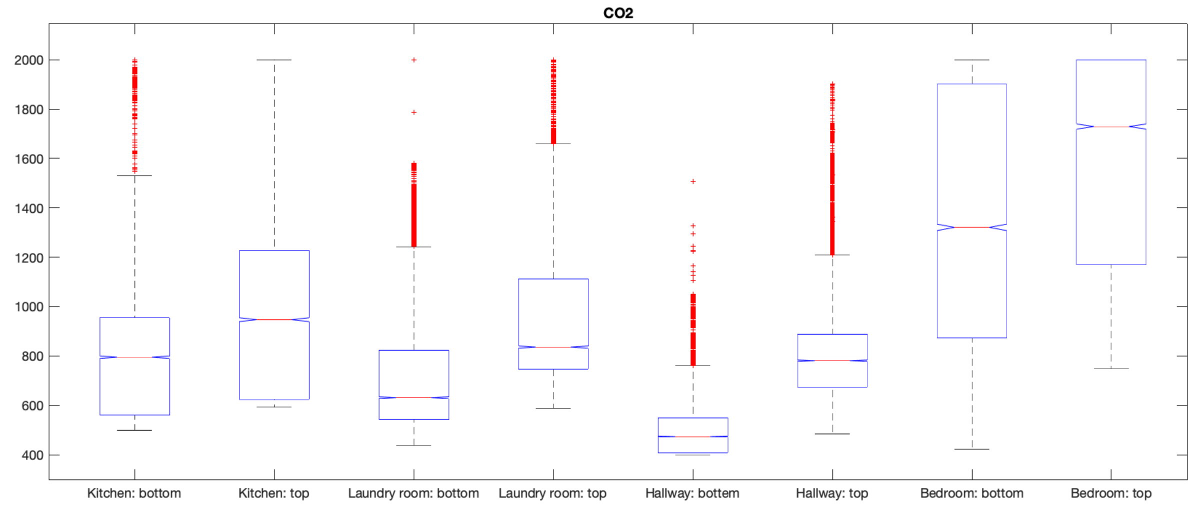


Figure 6: Boxplots of all long-term sensor readings per location and placement

6 Design

This section discusses the design of Airbience, first the selected sensors are addressed in Subsection 6.1. Thereafter the design decisions and the resulting PCB schematic is covered in Subsection 6.2. Lastly, the final product is shown in Subsection 6.4.

6.1 Sensors

The sensors that are used are shown in Table 4. An MH-Z19C sensor has been chosen to measure the CO_2 level. A drawback of this sensor is that it can only measure the CO_2 levels between 400 ppm and 2000 ppm. The lower limit is fine since the CO_2 level outside usually does not drop below 380ppm, the upper limit is less ideal. Although the concentration of CO_2 in a room can easily increase beyond 2000ppm it does already indicate a too large amount of CO_2 is present. Thus the sensor is capable of indicating good and bad amounts of concentration in the air.

Besides the MH-Z19C, a CCS811 is selected to measure the total volatile organic compounds (TVOC) in the air, giving more insight into the air quality as explained in Section 2. The CCS811 also calculates an eCO_2 value, which estimates the CO_2 level, since it can not measure the particles directly. However, these readings weren't as stable and reliable during testing as the MH-Z19C, therefore the MH-Z19C has been retained as a CO_2 sensor. Temperature and humidity values can be sent to the CCS811 in order to increase the accuracy of the sensor. These environment values are attained from the AM2320 sensor, which is able to measure both.

Lastly, an MQ-7 sensor is used to measure CO levels. Although this sensor is not expected to be very accurate it does give insights into the degradation of the air quality due to CO.

An overview of the sensors and their accuracy can be seen in Table 4.

Table 4: Sensors equipped on Airbience.

Sensor	Type	Range	Accuracy
MH-Z19C	CO_2 level	400 - 2000 ppm	± 50ppm + 5% reading
CCS811	TVOC &	0 - 1187 ppb	Unknown
	eCO_2 level	400 - 8192 ppm	Unknown
AM2320	Temperature &	-40 - 80	± 0.5°C
	humidity	0 - 99.9	± 3% (on 25°C)
MQ7	CO level	20 - 2000 ppm	Unknown

6.1.1 Considered Sensors

The MQ-9 sensor was considered as well for measuring the CO levels. After performing some simple measurements, it was found that MQ-9 was very unstable, thus the MQ-7 sensor was chosen to measure the CO levels. The GP2Y10 was also tested but turned out not to be sensitive enough, it didn't react on a smoking match and candles. Figure 9 shows some measurement of the sensor while it was exposed to multiple smoke sources. It can be seen that the sensor didn't respond well (or not at all). Finally, the SGP30 was tested for a while in combination with the CCS811, these sensors reacted reasonably similarly and ultimately the CCS811 was chosen because there were already two available.

6.1.2 Calibration and validation

The sensors were placed in the same spot simultaneously for some time to make sure that the used sensors reported the same values. This experiment was repeated multiple times at different heights to ensure that the whole range of possible values was equal and not just at a single point. Both of the devices reported values in the same range and both devices are considered equal because the significance of the difference is very small.

Calibration of the sensors is a bit more difficult than verifying equality. An expensive CO_2 sensor would have been required to calibrate the MH-Z19 and CCS811 sensors. The first one however has an autocalibrate function that was used to calibrate the sensor. It uses some memory on the sensor to record the lowest value and set this as 400 ppm. After having the

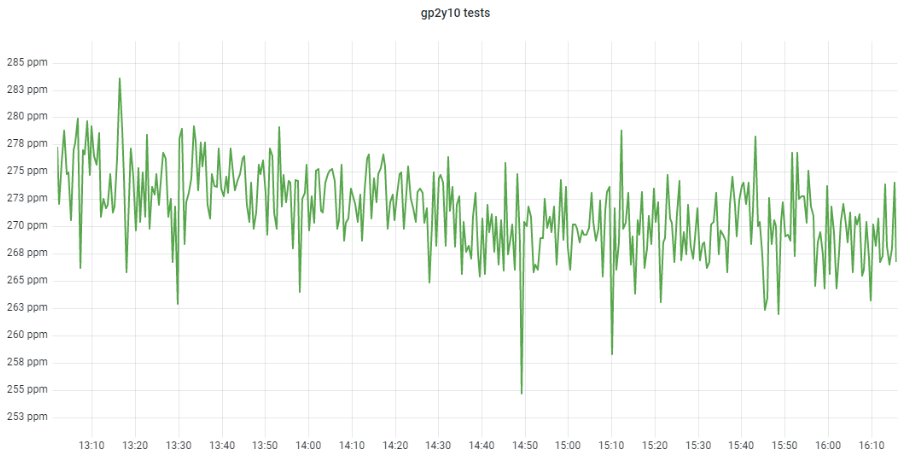


Figure 9: Test result of GP2Y10 sensor.

sensors on for a few days, the auto-calibrate function was disabled to make sure that the sensors did not recalibrate themselves and end up ruining the data. The other sensors were not calibrated because they were either already calibrated (temperature and humidity) or it was impossible to calibrate (TVOC) them.

6.2 PCB design

The PCB was designed in KiCad EDA, an open-source and free schematic and PCB layout editor [15]. The PCB was designed with replaceability and ease of producing in mind. All the parts are through-hole(THT) and easily available parts (sensors, ESP32, LEDs and resistors). It was possible to create a fully surface-mounted PCB, but that would be harder for the producer (user) to solder (at home) and is less serviceable. The chosen modules are all available on multiple web-shops and some are even swappable for other sensors(MQ-7 → MQ-136). Any moderate solderer should be able to desolder a broken module and swap it for a new one, resulting in less waste.

The PCB is designed in such a way that the sensors are all on the right side, where an opening can be made in the box or cables can connect to have the sensors aligned and have a better air flow across all the sensors. The LED's(6 in total) are on the other side of the ESP32. This way the LED's and resistors do not interfere with the sensors and the user should be able to see the LEDs more clearly. The LED's colours can be chosen to have a specific meaning and can all be dimmed to show the intensity of the warning, strength or level. A render of the PCB can be seen in Figure 10. A plot of the PCB that is manufactured can be seen in Figure 11. The big green area is the fill of one copper plane connected to the GND net. Using a ground plane minimizes EMC interference and less EMC radiation coming from the device.

The schematic of the PCB is shown in Figure 12. The generic headers were used for the CCS811, MH-Z19 and the MQ-7 because there were no drawings of them readily available.

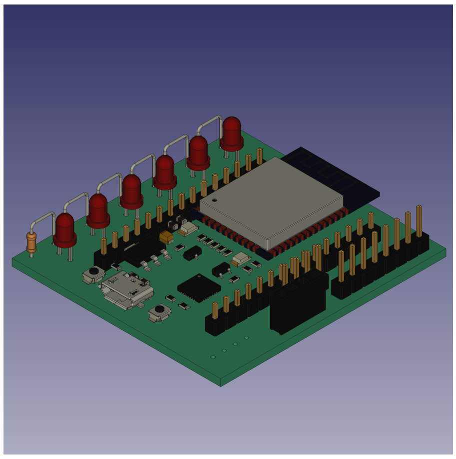


Figure 10: Render of the PCB in FreeCAD. PCB of the ESP32 module is not visible.

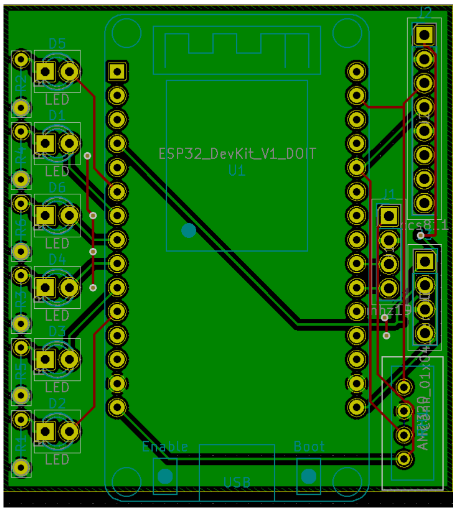


Figure 11: PCB layout

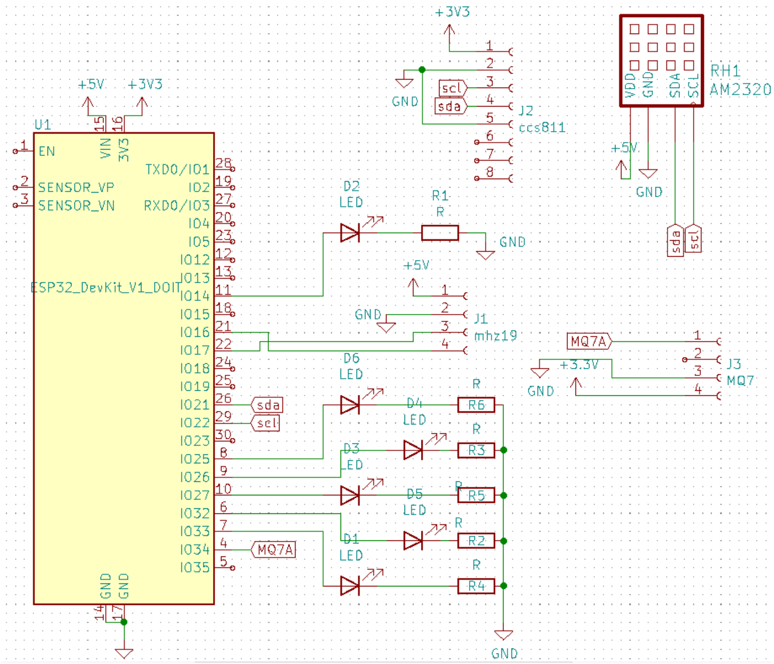


Figure 12: PCB schematic

6.3 Software

The software is an important part of the working of the device. The software is designed to be as reusable and modular as possible and is not designed specifically for this product. There are three parts to the software: the IoT device, the server and the webapp.

6.3.1 Embedded IoT software The software for the ESP32 chip that is on the device is written in C++ with the Arduino framework. Platformio was chosen to be used instead of the Arduino IDE, because this system is more flexible, allows for multiple files and is easier to work with. To allow for multiple sensors and make it easy to add sensors, the sensor framework was designed [16]. This framework can be seen in Figure 13. For each sensor a header file and source code file should be created, where the initialization code, for setting pin modes or connecting a sensor, must be called in an init function. The code for reading the sensors must be added in another function that also directly adds the values to a JSON object. The init and data functions then are grouped in a sensor struct. After creating this code, the only thing to get the sensor readings is adding the struct to the global sensors array variable. The code loops over each sensor when needed(e.g., initializing, getting data) and checks that the needed function exists before calling it. This way not every function has to be implemented before adding it. The pseudocode of the device is shown in Listing 2. For reading back the response the same approach is taken with the array of structs. There is no project/product-specific code besides the specific sensors and the URL of the server.

When the device is unable to connect to a known Wi-Fi access point(AP), it will create an AP. The user can then connect to this AP and enter their Wi-Fi details to let the device connect to it. This way, an end-user does not have to write code or enter their details beforehand but just enter it when needed.

Listing 2 Device pseudocode

```
#include sensors
const sensor sensors[] = {.};
void setup() {
    // connect to internet
    // loop over sensors and readers
    // and initialize them
}
void loop() { // function in while(1)
    if(wifi == connected) {
        // get all data from sensors
        // send the data to the server
        // parse response
        // give readers the response
    }
    // delay some time before
    // next reading
}
```

6.3.2 Server The server has to handle all the data packages from the IoT devices, store it, analyze it, return a response and handle requests from the user interface. The server code is written in JavaScript (NodeJS) with the Express package for handling the HTTP requests. The data is stored in a PostgreSQL database. The server is also designed with modularity and reusability in mind. No code except the analyzers is project-specific. A rough overview of the server can be seen in Figure 14.

When a device sends data, the password is checked first to make sure it is coming from the correct device. After that, the previous data of that device and the user connected to it is retrieved from the database. This data can be used by the analyzers. The analyzers are JavaScript files that are put in the *analyzers* folder. The analyzers are started sequentially with the newest data, the older data and the user data. The analyzers can add information to the current data, which will be sent back to the device, and use the older data for things like calculating trend lines. Also, an analyzer can send a notification to a user.

The server code can be run on any NodeJS capable device, like a Raspberry Pi, a normal PC or an actual server. Using a server(or virtual private server(VPS)) removes the requirement that the devices and the server are in the same network and creates the opportunity to add an SSL certificate. The database can be put anywhere, as long as it is approachable from the server.

The implemented notifications are:

- Brainpower reduction. This notification is sent when the reduction is more than 40%. It is a linear mapping taken from the graphs of [9].
- People counter. Using the algorithm of Section 4 an estimate is done of the number of people in the room.
- Positive notifications. When the CO_2 level is low enough for an hour, it will send a notification.
- Temperature alarm. A notification is sent when the temperature is too high.

The first 3 notifications can be seen in Figure 15. These notifications are just a small subset of all the possible notifications that can be generated with the data received from the IoT devices.

6.3.3 Webapp The UI is written in Vue.js with Vuetify. Vue.js has some features that make it easier to create a website, like automatic re-rendering on variable change(2-way binding) and easy reusability of components. The web app can be served from the same server as static files. To use the web app, one has to log in first and connect their device to their account by entering the device's ID and (part of) password. After this, notifications are automatically sent to the users(computer, phone, ...) and the user can see the past data, compare it, and see the current sensor readings. A screenshot of the Webapp can be seen in Figure 16.

6.3.4 Conclusion Overall, the software is designed with modularity and changeability in mind while using the latest modern tools. The device code(C++ ESP32 code) can be easily changed to a manufacturer's or user's needs. The server code is also easily adjusted to add more analyzers or features. The web app is created with a modern framework that most front-end developers should be able to work with. Therefore, a company can easily create a new IoT device and web app with this full-stack software solution. A hobbyist can also create their own system in a few hours. The code and schematics with a tutorial to install and edit the whole stack can be found at Github.com/ArendJan/easyIoT.

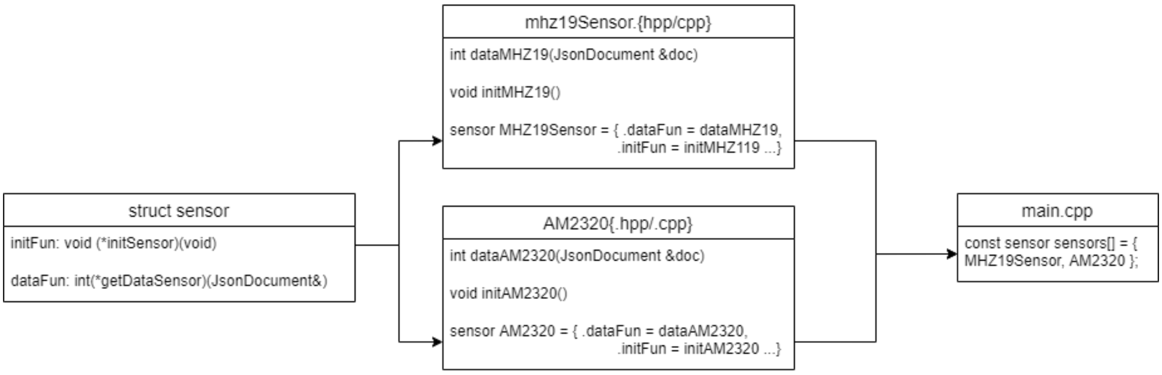


Figure 13: The sensor framework designed for modularity and changeability

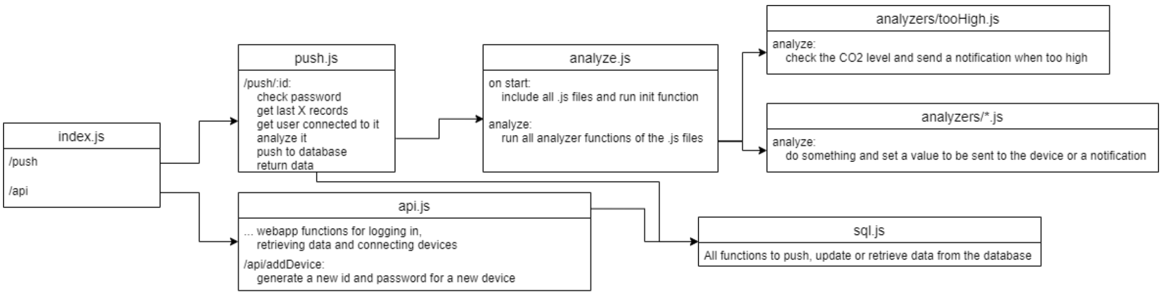


Figure 14: Overview of the server architecture

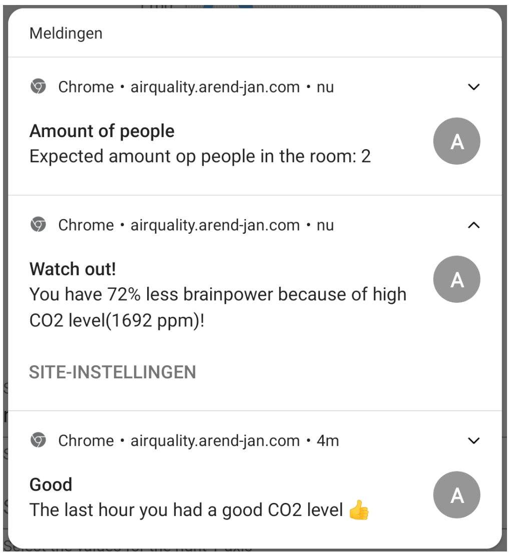


Figure 15: Screenshot of notifications

6.4 The finished product

And then there is the final product, a manufactured PCB with soldered components in a self-designed and produced housing. Let's first discuss the case. The case is designed using Fusion 360 from Autodesk [17]. The housing consists of a tray and a sliding top. The housing is free of screws, making it easy to open and close without tools. Nonetheless, the lid always remains in place due to the slight friction between the two parts. There are many elliptical holes in the casing, which allow air to move in and out quickly. Larger oval holes were chosen because they are faster to 3D print (and in different orientations) than, for example, fine-grained mesh structures. In addition, a cutout was made to see the LEDs and to connect the micro-USB cable to the ESP32.

Finally, after all the development steps, the finished product is shown in Figure 17 and Figure 18. The casing fits perfectly around the hardware, the micro-USB is easily accessible and last but not least, the LEDs are visible.

7 Future works

Due to the MHZ19C CO_2 sensor's constraints, it was unfortunately impossible to measure the CO_2 level in a room above 2000 ppm. In addition, this sensor is a rather low-cost CO_2 sensor which means that it might not be able to estimate the true values as accurately as other sensors might do. In future experiments, it might be interesting to see if more expensive CO_2 sensors could be equipped to have a broader measuring range and check whether the MH-Z19C has significant deviations.

If more experiments were to be done in multiple different rooms and with more people, the CO_2 estimating algorithm could be improved to work in any room and more accurately. This would require a formula to only use a room size to get an algorithm that does not need specific details about the room. The room size should be known since it has a significant impact on the CO_2 gradients.

Finally, personalization could be improved. If users themselves are able to activate individual LEDs according to their wishes (sensor values, time, etc.), they can make even better use of our device.

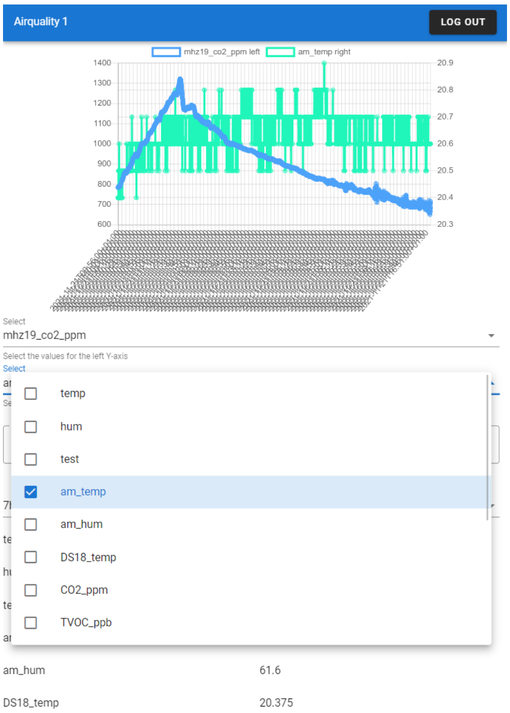


Figure 16: Webapp screenshot

8 Conclusion

Placing the sensor nodes in different rooms and at various locations enabled us to understand variations in CO_2 levels caused by multiple sources. Regarding the placement of sensors, several set-ups have shown different impacts due to the environment, such as a stove, fridge, and vibrations from the washing machine. Moreover, the height of a node also had an impact. Despite the local environmental influences and offset by height difference, both nodes showed roughly the same trends. Therefore, there is not just one right place to place a sensor, but rather a few guidelines should be followed to obtain the best possible measurements as mentioned in Subsection 5.3.

From the results, we can conclude that it is possible to detect if someone is present in a room by measuring the amount of CO_2 in that room. Moreover, our algorithm was able to calculate the number of people present in a closed room.

Last but not least, A consumer-ready product has been designed. A custom PCB has been designed and manufactured to place the sensors in an optimal manner. In addition, a custom 3D printed case is created to store Airbience safely. The final product also comes with a lot of embedded software that is designed to be modular. Moreover, a thoroughly developed front-end and back-end software that enables users to read the sensor data anywhere in the world has been created. Our product can help millions of people in the future to understand their indoor air quality, allowing them to let fresh air in on time and not inhale too much bad air. This will improve people's health and make the use of energy more sustainable during the winter months.

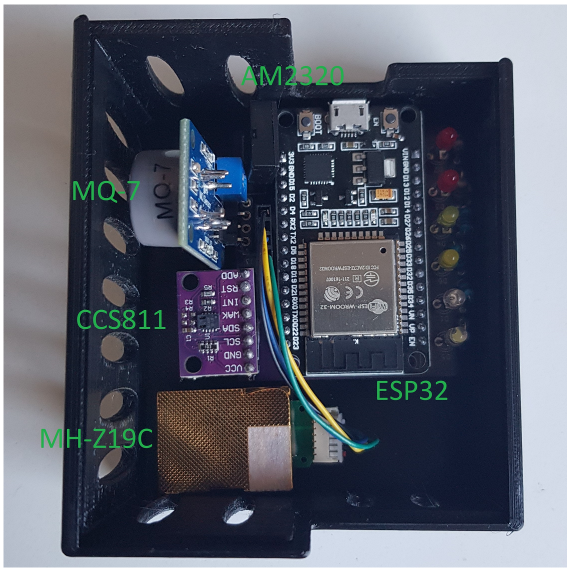


Figure 17: View of the inside of Airbience.

Figure 18: Closed casing of Airbience.

References

[1] Mark Gillott et al. *Improving the airtightness in an existing UK dwelling: the challenges, the measures and their effectiveness.* Sept. 2015. URL: `https://hdl.handle.net/2134/19511`.

[2] Mahesh Jayaweera et al. "Transmission of COVID-19 virus by droplets and aerosols: A critical review on the unresolved dichotomy". In: *Environmental research* 188 (2020), p. 109819.

[3] Renyi Zhang et al. "Identifying airborne transmission as the dominant route for the spread of COVID-19". In: *Proceedings of the National Academy of Sciences* 117.26 (2020), pp. 14857–14863. ISSN: 0027-8424. DOI: `10.1073/pnas.2009637117`. eprint: `https://www.pnas.org/content/117/26/14857.full.pdf`. URL: `https://www.pnas.org/content/117/26/14857`.

[4] Paul Dabisch et al. "The influence of temperature, humidity, and simulated sunlight on the infectivity of SARS-CoV-2 in aerosols". In: *Aerosol Science and Technology* 55.2 (2021), pp. 142–153. DOI: `10.1080/02786826.2020.1829536`. eprint:

https://doi.org/10.1080/02786826.2020.1829536. URL: https://doi.org/10.1080/02786826.2020.1829536.

[5] László Kajtár and Levente Herczeg. "Influence of carbon-dioxide concentration on human well-being and intensity of mental work". In: *Időjárás* 116 (May 2012), pp. 145–169.

[6] Sef van den Elshout et al. May 2012. URL: http://www.airqualitynow.eu/sl/download/CITEAIR-Comparing_Urban_Air_Quality_across_Borders.pdf.

[7] European Commission Scientific Committee on Health and Environmental Risks. *Indoor Air Quality*. 2008. URL: https://ec.europa.eu/health/scientific_committees/opinions_layman/en/indoor-air-pollution/index.htm#9 (visited on 11/16/2021).

[8] D Kotzias et al. "The INDEX project: critical appraisal of the setting and implementation of indoor exposure limits in the EU". In: *Final report, EUR* 21590 (2005).

[9] Joseph G. Allen et al. "Associations of Cognitive Function Scores with Carbon Dioxide, Ventilation, and Volatile Organic Compound Exposures in Office Workers: A Controlled Exposure Study of Green and Conventional Office Environments". In: *Environmental Health Perspectives* 124.6 (2016), pp. 805–812. DOI: 10.1289/ehp.1510037. eprint: https://ehp.niehs.nih.gov/doi/pdf/10.1289/ehp.1510037. URL: https://ehp.niehs.nih.gov/doi/abs/10.1289/ehp.1510037.

[10] Tyler A Jacobson et al. "Direct human health risks of increased atmospheric carbon dioxide". In: *Nature Sustainability* 2.8 (2019), pp. 691–701. DOI: 10.1038/s41893-019-0323-1.

[11] Yunwan Jeon et al. "IoT-based occupancy detection system in indoor residential environments". In: *Building and Environment* 132 (2018), pp. 181–204. ISSN: 0360-1323. DOI: https://doi.org/10.1016/j.buildenv.2018.01.043. URL: https://www.sciencedirect.com/science/article/pii/S0360132318300611.

[12] Theis Heidmann Pedersen, Kasper Ubbe Nielsen, and Steffen Petersen. "Method for room occupancy detection based on trajectory of indoor climate sensor data". In: *Building and Environment* 115 (2017), pp. 147–156. ISSN: 0360-1323. DOI: https://doi.org/10.1016/j.buildenv.2017.01.023. URL: https://www.sciencedirect.com/science/article/pii/S0360132317300367.

[13] Ming Jin et al. "Occupancy Detection via Environmental Sensing". In: *IEEE Transactions on Automation Science and Engineering* 15.2 (2018), pp. 443–455. DOI: 10.1109/TASE.2016.2619720.

[14] Khee Lam et al. "Occupancy detection through an extensive environmental sensor network in an open-plan office building". In: *IBPSA 2009 - International Building Performance Simulation Association 2009* (Jan. 2009).

[15] KiCad. *KiCad EDA*. Version 5.1.12. Nov. 18, 2021. URL: https://www.kicad.org/.

[16] pioLabs. *PlatformIO*. Version 5.2.2. Nov. 18, 2021. URL: https://platformio.org/.

[17] AUTODESK. *Fusion 360*. Version undefined. Nov. 18, 2021. URL: https://www.autodesk.nl/products/fusion-360/overview.