Development of an intelligent multi-sensor system for real-time monitoring and analysis of soil pollution on construction sites

By: Allard van der Hooft

Date: 13-07-2024

Supervisor: Rob Bemthuis

Critical Observer: Berend-Jan van der Zwaag

Faculty of Electrical Engineering, Mathematics, and Computer Science (EEMCS), University of Twente

Abstract

The thesis "Development of an intelligent multi-sensor system for real-time monitoring and analysis of soil pollution on construction sites" presents a comprehensive study on creating a sophisticated sensor system for monitoring and analyzing soil pollution in real-time at construction sites. The research addresses the need for sustainable practices in the construction industry by measuring various soil pollution indicators, such as CO_2, conductivity, temperature, vibration, pH, and humidity levels. The study aims to provide construction site managers and researchers with valuable insights to improve sustainability on construction sites.

Through background research and state-of-the-art analysis, it was found that the limitations of existing soil measurement tools emphasize the necessity for an integrated multi-sensor system capable of measuring multiple indicators simultaneously. Multiple methods are used to arrive at the final design, namely stakeholder analysis, system requirements, realization, and user evaluation together these methods ensure the development of a user-friendly and effective sensor system.

The results of the user and system testing reveal that the users like the state of the current system, especially the simple design of the user interface. However, more information should be placed on this interface to make it easier to interpret the displayed data. Users also stated that more sensors could be implemented to increase the amount of data received. The results of the system testing showcase that the sensors are most likely accurate, but this can only be fully concluded if lab testing is done. Testing also showcases that the vibration sensor does not operate properly.

To conclude, creating an intelligent multi-sensor system for real-time monitoring and analysis of soil pollution on construction sites is feasible, but the real impact of the system can only be found if testing on actual construction sites is conducted.

Acknowledgement

Firstly, I want to thank my supervisor, Rob Bemthuis, and critical observer, Berend-Jan van der Zwaag, for always being ready to help me whenever I had questions during the weekly meeting and for quickly responding to my random team messages at 11 p.m. I also want to thank them for their positive feedback during the creation of this thesis, especially when I did not know how to approach a specific problem further. Secondly, I would like to thank all the user testers who participated in testing the final sensor system and provided me with important feedback on the sensor system. Special thanks go to Bouwe van der Tuuk from the company Dura Vermeer for allowing me to explore a live construction site and for answering questions I had on construction practices in the early stages of this research project.

Lastly, I want to thank my dad for his help when I encountered issues with the creation of the sensor system and, his guidance on the text when I did not know how to approach a section further, and my mom for reviewing the final text and giving notes on parts that could be further improved or that were not yet super clear.

List of Figures

Table of Contents

Glossary

Soil pollution = contamination from the soil, such as CO_2.

Real-time = the immediate measuring of a specific indicator using a sensor, where information is delivered or updated as soon as it becomes available without significant delay.

Pre-construction phase = the period of construction before the actual building begins

IO = Input Output

ROM = Read Only Memory

RAM = Random access memory

IO = input/output

PCB = printed circuit board

SDA = The Serial Data

SCL = Serial Clock

embodied carbon emissions = emissions caused during the extraction, production, transport, and manufacturing of products used in the construction industry

IRGA = Infrared gas analyzer

CO_2 = carbon dioxide

N_2O = Nitrous oxide

SO_2 = Sulfur dioxide

CO = carbon monoxide

TDR = Time domain reflectometry

SET = Sensor Electronic Technology

BFT = Byzantine fault tolerance

VWC = Volumetric water

IP65 = The IP is a code that indicates how well a device is protected against water and dust. IP65 means that the product has a high level of dust protection, and is able to withstand low-pressure water jets from all directions.

IP68 = water resistant in fresh water to a maximum depth of 1.5 metres for up to 30 minutes

Ppm = parts per million

SRAM = Static random-access memory

baud rate = speed of the serial communication functionality. Given as the number of changes in those signals per second

Chapter 1 Introduction

This introduction aims to provide the background for the graduation project, outline the challenge, and present the main research question along with its sub-questions.

1.1 Situation and relevance

The construction industry is significantly impacted by the ongoing global emission crisis. The construction sector is responsible for approximately 40% of global CO_2 emissions, with operational and embodied carbon emissions posing serious challenges to reduce. The need for sustainable practices in the construction industry is becoming more urgent due to growing pressures from public and regulatory bodies [44][43]. According to the UN Environment Program, improvements are necessary across various construction phases, including manufacturing, transportation, construction, operations/management, and end-of-life deconstruction [43].

Several regulatory frameworks, such as the Paris Climate Agreement, Clean Air Agreement, and the Nitrogen Approach [42][45][47], aim to reduce and prevent emissions by at least 49 percent by 2030. The goals in the construction industry are to achieve a 60 percent reduction in nitrogen emissions and a 0.4 Mton CO_2 reduction [1]. Research shows that mining and manufacturing of materials account for approximately 90 percent of the total CO_2 emissions. However, this area of sustainability improvement on construction sites has been researched extensively, and good theoretical guidelines are already in place to lower these emissions [3].

The pre-construction phase includes planning the expected CO_2 emissions and making the building phase more CO_2 efficient. Having better and more data sources helps to make this transition more straightforward and manageable. Early investigation shows that CO_2 measurement in the air and above the ground is relatively well-researched. Measuring data in the soil is less popular. Searching with Google Scholar shows that CO_2 soil measurements (+CO_2 +soil +sensor +construction give 31000 results) have less than half the number of results compared with air measurements (+CO_2 +air +sensor +construction give 77000 results). When looking into more detail to the first 1000 results, a rough estimate is that only around a quarter of those soil results really are about soil measurements. In the case of air measurements, this number is estimated to be more than half. When looking at the “CO_2 sensor” results, only a fraction of the 500 results are about measuring CO_2 in the soil.

Due to big inefficiencies and a lack of data sources, reducing emissions on construction sites can be challenging, because of disruptions such as weather changes, traffic congestion, equipment breakdowns, and diseases. Even minor disruptions can result in delays, cost overruns, project cancellations, and increased emissions. The complexity of the construction sector and the diversity of data, coupled with unreliable data sources, make effective emission reduction management nearly impossible [2].

1.2 Challenge

This project aims to design a system with sensors that measure quantitative soil data, such as CO_2, conductivity, temperature, and vibration. Together with climate data, these values help to gain insight into when and what causes soil pollution, and they can be compared with similar data measured in the air. The results give construction site managers and researchers better insight and can help make construction sites more sustainable during construction and make the pre-construction phase more efficient for construction managers.

1.3 Research Question

From the situation and challenge, the main research question is obtained:
How can an intelligent system with multiple interconnected sensors be developed to simultaneously measure and compare soil pollution attributes on construction sites, ensuring accurate and effective monitoring of environmental pollution indicators?

To help answer the main research question, these sub-questions are added:

- What are the most significant types of pollutants found on construction sites?
- Which soil pollution indicators can be reliably measured using ground-based sensors?
- What methods can be implemented to ensure each sensor accurately measures the targeted data?
- How can sensor data be effectively integrated and analyzed to provide meaningful insights into soil pollution levels?

1.4 Report Outline

The succeeding parts of this thesis are as follows. First, in Chapter 2, the background research will be discussed and analyzed. This chapter will include a literature review and a detailed state-of-the-art analysis. In chapter 3, the design methods used to arrive at the final iteration will be explained. This research uses a specific methodology to arrive at the final iteration, namely the Creative Technology Design Process, which uses an improvement feedback loop to keep improving the product with the help of user feedback. In the ideation chapter, the stakeholder analysis discussed in chapter 3 will be performed to find the relevant stakeholders involved in the research. Next, the system requirements will be analyzed, and the system evaluation method will be used during the realization. Chapter 4 uses the methods discussed in chapter 3 to perform the ideation of the research. In chapter 5, the sensors and tools used to create the final prototype will be explained next to functional architecture and time-sequence diagrams used to show the prototype's upper-level workings. Chapter 6 dives into the realization of the prototype, discussing how the final prototype is created. Chapter 7 discusses the measurement results of the sensor system, showing that the sensors can indeed measure the environmental pollution indicators. Chapter 8 discusses the user evaluations of the sensor system. Chapter 9 is the discussion of the performed challenge including future research. Lastly, chapter 10 dives into the thesis's conclusion, answering this report's research questions.

Chapter 2 Background Research/state of the art

This chapter includes background research into what should be measured and whether it is feasible in the scope of the research. First, the types of construction site pollutants will be explained, and the focus of this research will be. Second, the different types of soil pollution indicators will be established and explained. Lastly, an overview of the state of the art will be given.

2.1 Types of Construction Site Pollutants

Before the different types of soil pollution indicators can be established, the different types of soil pollutants on the construction site must be established. Research found that there are five different types of construction pollutants [41]:

- Noise Pollution: Noise pollution includes unwanted noise from the construction site caused by heavy equipment. This can be harmful to humans and animals [41].
- Dust Pollution: Dust pollution includes dust. Dust is powder from dirt that is already present (soil, sand) in or near the construction site. It can also come from the building process itself, machines, or activities like sawing, grinding, and more [41].
- Water Pollution: Water pollution includes contaminating groundwater or a body of water, like a lake or river, with a chemical or non-chemical substance [41].
- Land Pollution: Land pollution includes the contamination of land and soil. This can be due to the decomposition of waste materials. It can result in liquid and solid pollution and can affect both the soil and the groundwater. land that affects the soil and groundwater adversely [41].
- Air Pollution: Air Pollution is the contamination of the air surrounding the construction site. It can be both outdoor and indoor. Air pollution can come from equipment, machines, transport, and building materials, as well as from the soil. It includes gasses like CO_2, SO_2, and CO [41].

This research will focus on monitoring soil pollution's contribution to air pollution on construction sites. To find meaningful information on soil pollution, it is essential to understand what can be measured in the soil and what is practically feasible within the scope and time of this research. Initial investigation shows that the following parameters can potentially be measured in the soil:

- pH [7]
- Temperature [12]
- CO_2 [4]
- Conductivity [7]
- Humidity [7]
- Vibration [11]
- N_2O [15]

In the following paragraphs, all these parameters will be examined in more detail to determine whether measuring them is feasible, and the significant types of soil pollutants found on construction sites will be explored.

2.2 Measuring CO_2

Over the last thirty years, global carbon dioxide emissions have doubled, and they are projected to triple by 2040. If the current rate of fossil fuel consumption continues, this projection might become a reality. This rise in CO_2 emissions results in a rise in the average global temperature, which can lead to significant environmental consequences [16]. Reducing the total amount of CO_2 emissions will, therefore, be essential to reducing the rise in the average global temperature. Understanding where and how this CO_2 is emitted is essential to reducing the total CO_2 emissions by improving these processes.

Measuring CO_2 levels from construction sites in the soil can be challenging due to the significant contribution of natural CO_2 emissions from soil to the global carbon release in the atmosphere, with a global average CO_2 flux value estimated at around 6 grams per square meter daily [4]. With this in mind, it could be challenging to differentiate between the natural degassing of the earth and the extra CO_2 that is being pumped back into the soil due to the natural spread of the CO_2 emitted from vehicles and other construction equipment being used on construction sites. A possible solution for this could be to measure soil CO_2 levels alongside air CO_2 measurements and measurements at the exhaust of the construction equipment. Measurements were taken before the construction period began at the construction site itself, and indicators linked to the surrounding area of the construction site. To determine how much extra carbon dioxide is emitted into the soil becomes feasible.

Camarda [4] states that measuring ground CO_2 levels can be grouped into two main classes: remote sensing methods and in situ measurement methods. Remote sensing methods produce higher fluctuations and increase the average values, while situ measurement methods mask the influence of exogenous parameters. For this research, situ measurements are more suited to minimizing outside factors when measuring CO_2 levels, and measuring them at the construction site itself is a priority. When taking measurements using situ analyzers, dynamic and static measurements must be considered. The difference between these two measurement techniques is that the air in the dynamic chamber method is circulated from the chamber to an IRGA and then back to the chamber, while in the static method, the air is trapped above the soil and then measured [46]. Due to the movement of the air in the dynamic method, measurements are taken much more frequently in the range of minutes to hours, while static measurements are in the range of hours to days. Jensen [5] shows that dynamic CO_2 measurement offers more accurate estimates of actual surface level CO_2 flux than the static method. The longer time intervals between each reading can introduce biases in static measurement. Another problem with the static method is that disruptions, such as changes in temperature and moisture inside the chamber, lead to inaccurate readings [46]. Both of these factors lead to less accurate assessments of relative differences between flux rates. By continuously monitoring CO_2 levels, dynamic measurement offers a more accurate picture of flux rates. For the purpose of this study, the static method seems the obvious candidate since we want to measure ground CO_2 over more extended periods ranging in weeks/months. However, the static method could also be helpful because it provides more accurate results.

To measure the CO_2, a CO_2 sensor in a closed container inserted in the soil will be used (static chamber method). To ensure accurate CO_2 measurements, it is crucial to seal the container correctly. Air leaks would mean measuring a mix between the outside air CO_2 level and the soil CO_2 level. To guarantee the gas-tightness of the system, the casing should be made from an airtight material with a lid that fits tightly on the casing and an airtight seal [6].

2.3 Measuring conductivity

In the scope of sustainability, conductivity helps to identify soil salinity, pollution, and soil health. It is necessary to know whether soil health is related to increased or decreased conductivity and increased emissions such as CO_2.

TDR has great potential for measuring conductivity in the soil. It works by sending an electromagnetic pulse through soil guided by a transmission line or probe embedded in the soil. If the pulse encounters a boundary between soil with different dielectric properties (such as dry soil and moist soil), part of the pulse is reflected back to the instrument [7]. The time delay between the point where the pulse is shot and when it returns to the instrument is used to determine the velocity of propagation through the soil [8].

Measurement of the soil's conductivity before and after construction is important to determine differences in soil quality. Higher electrical conductivity (EC) values can indicate soil salinity or pollution, which may harm plant growth and soil health. Moderate EC values do not always imply poor soil quality [9]. A TDR can accurately measure conductivity and other soil pollutants in real-time. This is useful in the research context since accurately determining whether there is a significant rise, decrease, or no change is important.

Good calibration is vital when using the TDR. Differences in soil material, such as clays and organic soils, can drastically affect the results you receive from the sensor. Furthermore, the length and spacing of the cables between multiple TDR probes play a critical role in the precision and reliability of the measurements. Longer cables may increase signal degradation and noise, while inconsistent spacing between probes can result in inconsistent measurements, further complicating the interpretation of soil moisture levels [7].

2.4 Measuring vibration

The objective is to measure vibration to see if there is a correlation between increased vibrations due to big motor vehicles and construction equipment on construction sites and an increase in emissions such as CO_2. This information can also be used to identify which vehicles release the most CO_2, which could be used to make these vehicles more efficient and decrease emissions to adhere to EU regulations.

Measuring vibrations in the ground is relatively easy. An accelerometer can measure acceleration, including vibrations in the ground. They can be attached directly to the soil or bored into the ground to measure vibration at different depths [10]. Gupta [11] shows that outside vibration factors, such as road traffic, heavily influence the vibration levels of the testing location. The dominant frequency range induced by road traffic was found to be between 10 and 30 Hz. Vibrations from the subway near the test location contributed to an even wider range of frequencies ranging from 20 to 100 Hz.

This external influence must be considered to ensure that the vibrations measured are from the construction site. To address this, a pre-construction site test could be conducted to determine the background vibration frequencies. Proper sensor placement and appropriate distances are crucial for reducing external frequencies or disturbances while optimizing and maximizing the capture of on-site frequencies.

2.5 Measuring temperature

Measuring soil temperature serves the same purpose as measuring vibrations in the soil. The aim is to see if there is a relationship between temperature and emissions such as CO_2. Higher temperatures also accelerate microbial activity and decomposition rates, thereby increasing CO_2 [17].

Soil surface temperature is difficult to measure due to its dynamic interaction with the atmospheric and subsurface temperature interactions. Deploying a sensor capable of maintaining thermal equilibrium with the soil surface while minimizing its influence on temperature can be challenging [12]. To address this, a possible solution could be to place the thermometer deeper in the ground, but the problem is that it might not accurately measure the top layers of the soil. Thermoelectric contact sensors and remote infrared techniques are the primary and historical ways to measure the top soil temperature. The biggest limitation of thermoelectric sensors is that they are prone to errors due to radiative, advective, and conductive heating or cooling, all of which can induce measurement errors [12].

Two new methods of measuring temperature could be used to mitigate this: SET sensors and BFT probes. Both of these solve many issues that thermoelectric sensors pose, such as errors due to radiation and lead-wire heat conduction. However, both sensing methods tend to underestimate surface temperature during the day and overestimate the temperature during the night [12]. The above-mentioned issues make it difficult to measure soil temperature accurately using traditional methods.

2.6 Measuring pH

Measuring pH could be useful for assessing environmental quality and soil health. An increase in pH could result in increased environmental emissions, such as CO_2, which must be compared to other indicators discussed above.

Different methods exist to measure soil pH, from pH test strips to electronic sensors. Test strips or other manual methods are not viable options as they do not result in automatic pH measurements. However, Electronic pH sensors could be a good solution for this project; they are generally quite accurate and enable real-time measurements. However, some pH sensors need a wet environment to make a pH measurement possible. However, the soil is not always that wet, which could result in the sensor not working correctly [13].

2.7 Measuring humidity

The humidity of the soil could have a huge impact on the release of carbon dioxide from the ground. Higher soil humidity increases CO_2 emissions by providing favorable conditions for microbial growth and enzymatic activity [17].

Measuring humidity in the ground is relatively easy. It uses the same principle as conductivity: an electromagnetic pulse is shot through the soil and arrives at the other probe. The time between start and arrival corresponds to the humidity [14]. Since most humidity sensors are conductivity sensors, these can be measured using the same sensor. The only issue that might occur is calibration since conductivity and humidity are not the same parameters and are only measured similarly.

2.8 Measuring N_2O

Measuring N_2O in the ground is relatively easy. It works the same as measuring the CO_2 levels in the ground using the static chamber method. However, the difference with CO_2 is that NO_2 is not abundant in the ground [15], meaning there is a limited need to measure it for the scope of this research.

2.9 State of the art

In this section, the state-of-the-art research will be presented. This research is conducted to determine which applications exist in the field of soil measuring and to determine the strengths and weaknesses of these applications.

	CO_2	Conductivity	Vibration	temperature	pH	Moisture	N2O
Arable Mark 3 [18]	-	-	-	✓	-	✓	-
Sentek EnviroSCAN [19]	-	-	-	✓	-	✓	-
MeterGroup TEROS 12 [20]	-	✓	-	✓	-	✓	-
RST Instruments Ltd [21]	-	-	✓	-	-	-	-
LI-COR Biosciences [22]	✓	-	-	-	-	-	✓
Spectrum Technologies [23]	-	-	-	-	✓	-	-

Table 2.1: research table with state-of-the-art application and their indicators.

In table 2.1, the state-of-the-art applications are listed along with the indicators they measure. A checkmark is added if the application contains one of the indicators. This allows for a clear comparison of which applications measure specific indicators. Due to various companies making state-of-the-art applications in this field. Some of the systems in table 2.1 only cover one indicator. Some cover multiple indicators, the most common overlap being both temperature and moisture in a single sensor system. Interestingly, no sensor system was found that covers more than 3 of the indicators in one system. Since most of the systems that were found were pretty similar or another implementation of a single sensor, it was chosen only to cover these six.

2.9.1 Metergroup (TEROS 12)

The metergroup TEROS 12, seen in figure 1, offers the most diverse set of measurements. It can measure conductivity, temperature, and moisture. These parameters are important for understanding soil composition and moisture levels, which impact plant growth. Metergroup claims that the sensor has an unverifiable ±1.0% VWC accuracy, which is good enough for precise measurements. Although this product is meant for agriculture, it could also be used on construction sites [20].

Figure 1: Metergroup (TEROS 12) multi-sensor [20].

2.9.2 Sentek EnviroSCAN

The Sentek EnvirSCAN, shown in Figure 2, can measure salinity, temperature, and moisture at depths up to 30 meters. Although it has fewer measurement features, it lacks the conductivity that TEROS 12 provides. But it makes up for that with its ability to place up to 16 sensors at 10-cm intervals [19]. The ability to place multiple sensors in close proximity could be usefull on construction sites.

Figure 2: Sentek EnviroSCAN multi-sensor [19].

2.9.3 Spectrum Technologies

Spectrum Technologies pH probe, shown in Figure 5, is an easy-to-use soil pH measurement tool. It has the added functionality of Bluetooth, so it can easily connect to your phone. It also gives real-time notifications whenever the pH value is out of range [23]. Both of these functionalities can be included in the final sensor system.

Figure 5: Spectrum Technologies sensor [23].

2.9.4 Arable Mark 3

The Arable Mark 3, shown in figure 3, is a system that combines weather, plant temperature, light level reflection, and soil/irrigation data with advanced modeling and machine learning for comprehensive, real-time crop insights. It can measure temperature and moisture but does not measure other indicators like CO_2, conductivity, or N_2O [18].

Figure 3: Arable Mark 3 multi-sensor [18].

2.9.5 RST Instruments Ltd

The RST Instruments Ltd, shown in figure 4, is a sensor designed to measure soil and structure vibrations for various construction, mining, and seismic monitoring applications. The one challenge with this sensor in relation to the scope is that it only measures soil vibration [21].

Figure 4: RST Instruments sensor [21].

2.9.6 LI-COR Biosciences

LI-COR Biosciences soil gas flux sensors, shown in figure 6, can measure CO_2 or N_2O. They can also measure whether CO_2 or N_2O is being put in the soil, whether it stays there, and for how long. The box is designed to reduce disruptions of natural soil gas transport across the soil surface to ensure high-precision flux measurements of CO_2 and N_2O.

Figure 6: LI-COR Biosciences sensors [22].

2.9.7 Conclusion State-of-the-art sensors

According to the state-of-the-art analysis, many soil measurement tools exist. While most are designed for agricultural purposes, many can also be used for other applications, like construction sites. Only six state-of-the-art solutions were analyzed, based on all the measurement tools/systems that exist.

Although many companies exist in the soil measurements industry, most measure only one or two indicators at a time, with some exceptions that measure three of the indicators identified in the background research.

One significant limitation of the soil measuring applications in the scope of this research is the lack of measured indicators. Measuring three of the seven indicators at maximum is insufficient to cover all the indicators essential to drawing conclusions on the construction site. Besides that, it is also unknown how precisely they measure all three indicators simultaneously. It could not be nearly precise enough to draw accurate conclusions for the construction site managers.

2.10 Conclusion

This section presents the conclusion of the background research. This literature review has delved into the aspects of measuring environmental indicators in the ground relevant to improving sustainability on construction sites. This research is helpful for construction site companies to meet strict emission reduction targets set by regulatory frameworks.

The review identified several parameters for assessing environmental impact. These include CO_2, conductivity, vibration, temperature, humidity, pH, and N_2O levels. Each of these parameters has its challenges and methods of measuring the specific indicator in the soil. The research highlighted the necessity of considering outside factors such as road traffic vibrations, the already present CO_2 in the ground, and the earth's internal heat. To mitigate these factors, a baseline measurement could be conducted to properly compare it with the measurements during the construction period so that the proper conclusions can be drawn during the construction phase.

Furthermore, the review emphasizes the importance of real-time and autonomous measuring to ensure no further input is needed and to improve the pre-construction phase with extra data. Integrating advanced technologies like AI for data analysis and decision-making could improve the construction process and enhance efficiency on the construction site.

To conclude, while significant steps have been taken to understand and mitigate environmental impact, the existing research reveals a notable gap in the availability of soil measurement tools capable of simultaneously assessing all essential indicators.

Chapter 3 Design method

This chapter introduces the design method used in this research project. First, the design process will be explained, followed by the stakeholder analysis, requirements, and evaluation. The design method will be used to make the ideation process more straightforward.

3.1 Creative Technology Design Process

For this project, the creative technology design process will be used [24]. Figure 3.1 shows this process. It starts with the design question and has four main phases: ideation, Specification, Realization, and Evaluation.

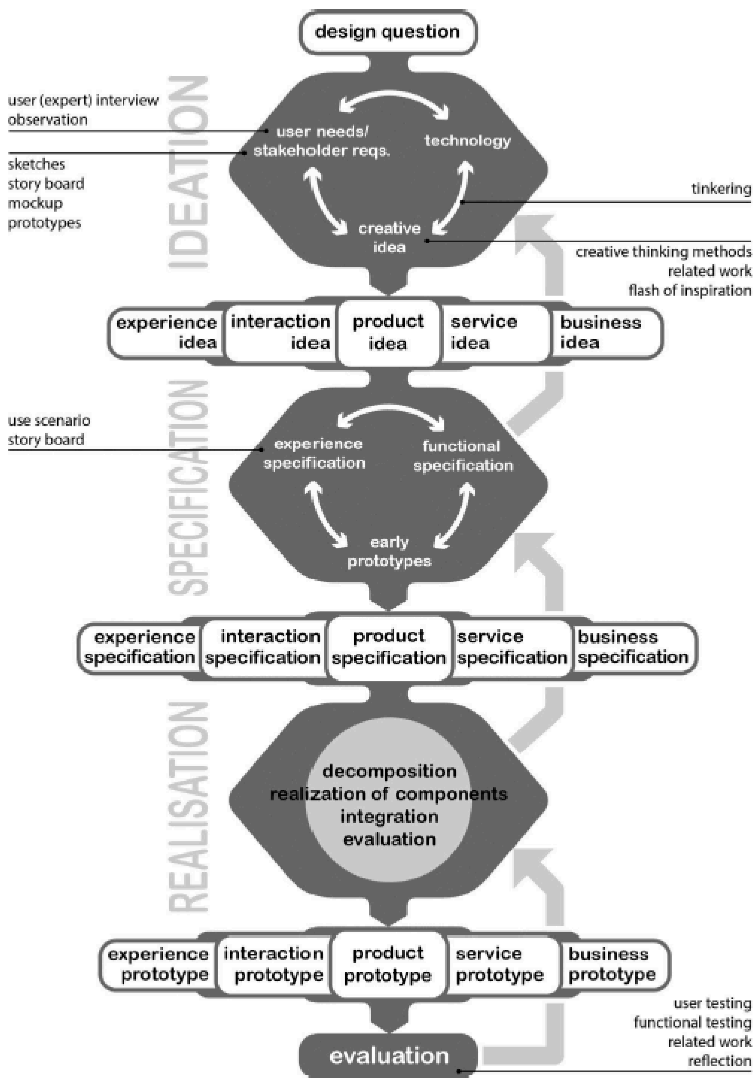

Figure 3.1: Creative Technology Design Process [24]

3.1.1 Ideation phase

During the Ideation phase, the design question is investigated, resulting in the project idea in the form of requirements. This is done by examining the design question from different perspectives. It involves checking current technology and talking to the project sponsor and other stakeholders to identify the project requirements.

3.1.2 Specification phase

In the specification phase, the requirements will be used to create several prototypes to test the various parts of the project. This helps to learn what is possible with current technology and how to meet other project boundaries like budget and time. These prototypes will help to finalize the final product specification.

3.1.3 Realization phase

In the realization phase, the final product specification from the specification phase is used to create the product's first iteration, the first prototype. This phase involves creating and integrating the different components that make up the whole product, and the different design considerations are important during this phase.

3.1.4 Evaluation phase

When the product's first version is created, it needs to be tested. That is done in the evaluation stage. The first step will be internal functional tests to check the functional requirements. Depending on the project timeline, improvements can be made to the system. After the functional test, further testing with stakeholders and nonfunctional requirements will be conducted.

3.2 Stakeholder Analysis

Before all the stakeholders can be identified, an understanding of the definition of stakeholder needs to be established. In the public and nonprofit sectors, typical definitions of stakeholders include:

- ‘All parties who will be affected by or will affect [the organization’s] strategy
- Any person, group, or organization that can place a claim on the organization’s attention, resources, or output, or is affected by that output
- People or small groups with the power to respond to, negotiate with, and change the strategic future of the organization
- Those individuals or groups who depend on the organization to fulfill their own goals and on whom, in turn, the organization depends [25]’

To find and correctly identify all the relevant stakeholders, a table will be created like the one in table 3.1. Column one provides the name of the stakeholder. In column two, the power/influence of the stakeholder will be added. In column three, the stakeholder's interest will be added. In column 4, the probability of participation will be shown, meaning how likely these stakeholders are to contribute to the research. In column 5, the role of the stakeholder will be shown, telling what their role in the research could be.

Stakeholder	Influence/power	Interest	Probability of Participation	Role
X	X	X	X	X

Table 3.1: Stakeholder identification table.

As shown in Figure 3.2, the power vs. interest grid of Eden and Ackermann [26] will be used to correctly assess all the stakeholders identified in Table 3.1.

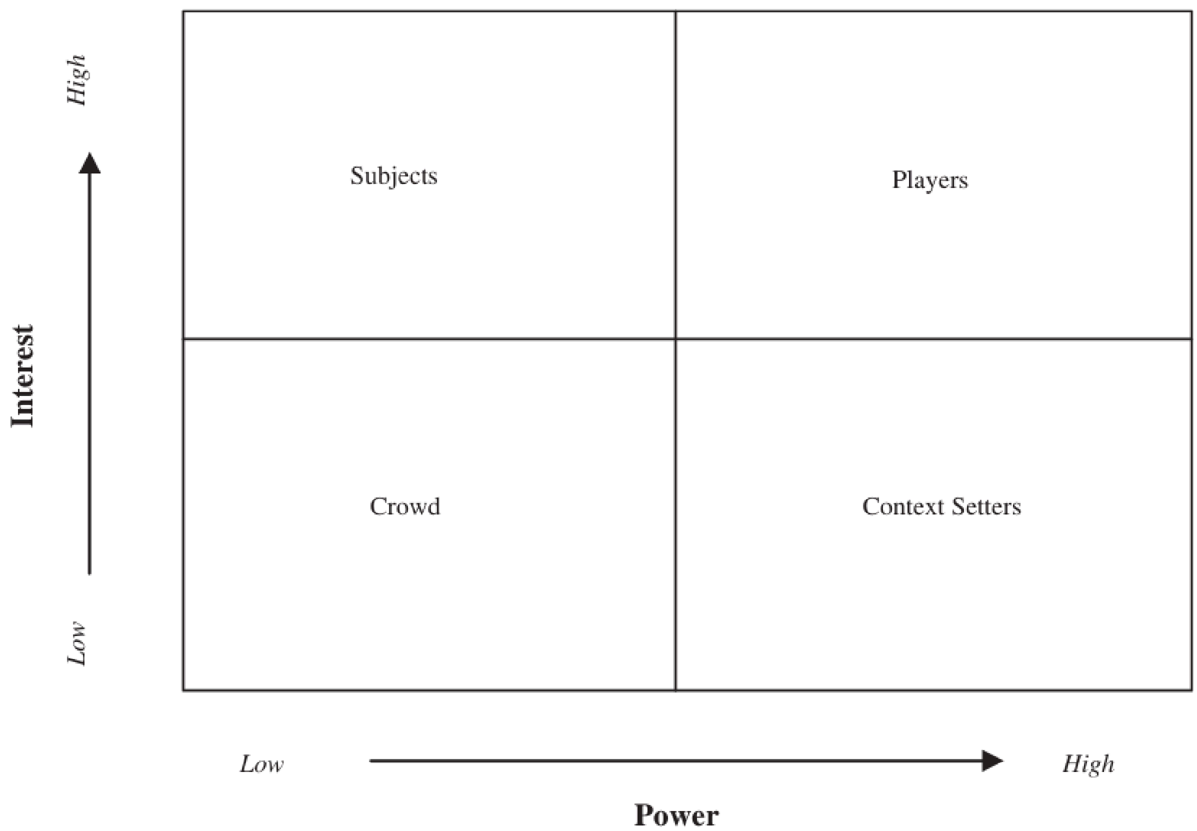

Figure 3.2: Power versus interest grid [26].

The matrix consists of a Y-axis representing the stakeholder's interest in the research. The X-axis represents the power of the stakeholder. This results in four stakeholder categories: ‘players who have both an interest and significant power; subjects who have an interest but little power; context setters who have power but little direct interest; and the crowd, which consists of stakeholders with little interest or power [25][26]’. Different actions need to be taken based on the placement of a specific stakeholder. A context setter, for example, needs to be given more interest in the research, while a subject needs to be given more power in the research to keep them interested. To complete this step of the stakeholder analysis, the stakeholders will individually be placed in the grid based on their power and interest to find out how to incorporate them in the research and their importance on decision-making later on during user evaluation.

3.3 Requirements

Identifying and categorizing the product requirements is crucial for the project's success. The requirements are gathered during the Ideation phase and can be categorized into functional requirements and non-functional requirements. Nonfunctional requirements describe the system's operational capabilities and constraints. These include requirements defining how well the product should operate, including speed, security, reliability, cost, etc.

When the requirements are clear, they need to be prioritized. For this process, the MoSCoW method will be used [27]. The result will be a list of requirements ranked by four types of priorities: must have, Should have, Could have, and Won't have [27]. This will follow a method shown in table 3.2.

Requirement	Must	Should	Could	Won't
Example requirement	X			

Table 3.2: Example requirement table following the MoSCoW method [27].

The categorized list will be used as input for the specification phase and, later in the project, the evaluation phase.

3.4 Evaluation

Evaluation is an important part of the project. The goal is to check if the project prototype meets the requirements and works correctly. This evaluation will be done in 4 steps:

1. Functional requirements tests
2. Non-functional requirements test
3. Stakeholder test
4. Field test

During the functional tests, the prototype is tested to see if it meets the functional requirements. This is the first test, which will be done internally before other stakeholders evaluate the prototype. The product must meet at least the 'must-have' requirements. The result of this phase of the evaluation is a list showing the functional requirements that are met or not met. This includes the ‘should have’ and ‘nice to have’ requirements. The project owner will do the functional test.

The next step is to test the non-functional requirements. As with the functional requirements, the prototype should meet the ‘must have’ requirements. However, the ‘should have’ and ‘nice to have’ requirements will also be tested. The result of this evaluation phase is a list showing the non-functional requirements that are met or not met. The project owner will do the non-functional requirements test.

When the internal evaluation steps are completed, the next step is to involve the stakeholders. The project sponsor or somebody appointed by the project sponsor is expected to do this. When the testing of the sensor system has finished, the stakeholder will be interviewed to gather their feedback.

The last step of the evaluation process is the field test. During this step, the prototype will be tested in the field. This could be on a real building site or in a location that closely resembles such a site. However, due to time constraints, this step could be skipped, in which case it will be put in the future research section.

Chapter 4 Ideation

This chapter will apply the knowledge and methods gained from chapters 2 and 3. First, the stakeholder analysis will be performed. Second, the Impact analysis will be performed. Third, the requirements will be established, and lastly, the concept phase will be shown.

4.1 Stakeholder analysis

The stakeholder analysis ranks the stakeholders on their power, interest, participation probability, and role in the research. This is important to the research as it will be highly useful later on in the research. Some of the individual stakeholders might view the research in a different manner. This table could be used to determine which stakeholders' comments are better to follow. In Table 4.1, the stakeholder analysis is performed according to the guidelines established in chapter 3.2.

Stakeholder	Influence/power	Interest	Probability of Participation	Role
Allard van der Hooft	Very high as I am the researcher	-	Very high	Main researcher
Rob Bermthuis	High	-	Very high	Critical observer
Berend-Jan van der Zwaag	High	-	Very high	Supervisor
Fran Karlović	Fairly high	-	High	Co-researcher (on a different RQ but same field-construction site sustainability)
Target group	Relatively high as their input can be very useful	Getting a product that is easy to use	Fairly high	User-based testing and interest testing
Hegeman-Nijverdal	Medium	Data or product to improve sustainability	Medium	Company
BouWatch	Medium	Data or product to improve sustainability	Medium	Company
University of Twente	Very low	-	Medium	University

Table 4.1: Stakeholder identification table

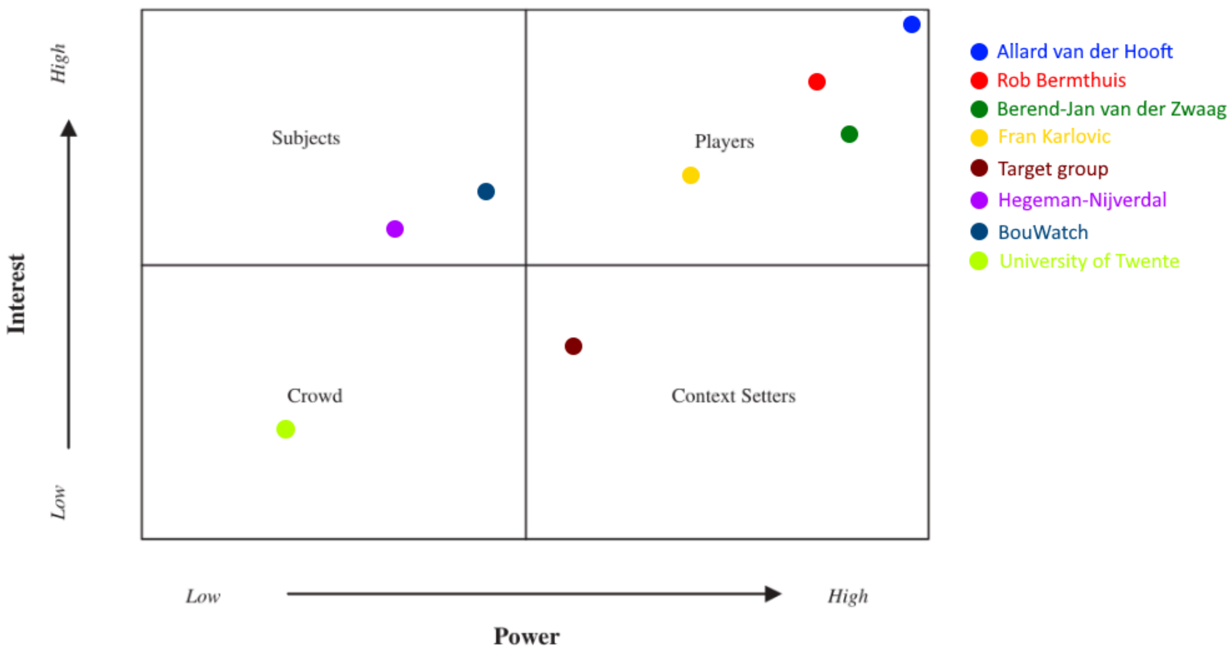

Figure 4.1: Power versus interest grid filled in [26].

In figure 4.1, the different stakeholders are placed based on their power and interest in the research.

4.2 System requirements

To correctly determine all the requirements for the whole system and the individual sensors, they will be split and individually analyzed using the MoSCoW method [27]. Table 4.2 shows the High-level System requirements. These requirements contain the basic building blocks of the overall system, including the different sensors. However, it is important to note that some of these requirements are meant for a final system used by construction managers on construction sites and are, therefore, not necessarily applicable to the final sensor system created in this paper.

Requirement	**Must**	**Should**	**Could**	**Won't**
Measuring soil pH	**X**			
Measuring soil Temperature	**X**			

Measuring soil CO_2	**X**			
Measuring soil conductivity	**X**			
Measuring soil humidity	**X**			
Measuring soil vibration	**X**			
Computer	**X**			
Computer Cloud platform		**X**		
Measuring N_2O			**X**	

Table 4.2: system-wide requirements following the MoSCoW method [27].

4.2.1 Generic System Requirements

The generic system requirements contain the requirements for the system as a whole. These requirements can be seen in table 4.3.

Requirement	**Must**	**Should**	**Could**	**Won't**
Must be used without mains connection	**X**			
Must work without mains connection for at least one week (> 1 week)	**X**			
It must be easy to continue using the sensor system for prolonged times (months) without a mains connection.	**X**			
Watertightness above the ground of at least IP65 [xx]	**X**			
Watertightness below the	**X**			

ground of at least IP68 [xx]				
Easy to install	**X**			
Sturdiness: Needs to be construction site proof	**X**			
Temperature range: -20 to 50 Celsius	**X**			
One button to turn on the system		**X**		
Total cost no more than 500 euros		**X**		
Tempered proof		**X**		
The ground surface of the system is smaller than 40 cm^2		**X**		
Height of the system smaller than 60 cm			**X**	
The weight of the system is less than 10kg.			**X**	
Passive cooled system			**X**	

Table 4.3: Generic system requirements following the MoSCoW method [27].

4.2.2 Measuring pH

The requirements for measuring pH can be seen in table 4.4.

Requirement	Must	Should	Could	Won't
pH measurement every hour	X			
Easy to interface to local computer	X			
Measuring Range of 0 - 14pH		X		
Accuracy of ± 0.1pH (25 ℃)		X		
Response Time of ≤ 90 sec		X		

Table 4.4: pH sensor requirements following the MoSCoW method [27].

4.2.3 Measuring Temperature

The requirements for measuring temperature can be seen in table 4.5.

Requirement	Must	Should	Could	Won't
Easy to interface to local computer	X			
Temperature measurement every 10 min	X			
accuracy of ±0.5 °C or better		X		

Table 4.5: temperature sensor requirements following the MoSCoW method [27].

4.2.4 Measuring CO_2

The requirements for measuring CO_2 can be seen in table 4.6.

Requirement	**Must**	**Should**	**Could**	**Won't**
Measurement Range of 0-5000 ppm, or better	**X**			
Good sensitivity and selectivity to CO_2	**X**			
Easy to interface to local computer	**X**			
CO_2 measurement every 10 min	**X**			
Response Time <90s		**X**		
Average Power <1W		**X**		
Low humidity and temperature dependency		**X**		

Table 4.6: CO_2 sensor requirements following the MoSCoW method [27].

4.2.5 Measuring Conductivity

The requirements for measuring conductivity can be seen in table 4.7.

Requirement	**Must**	**Should**	**Could**	**Won't**
Good sensitivity	**X**			
Easy to interface to local computer	**X**			
Conductivity measurement every hour	**X**			

Measurement range of 200 (very dry) to 2000 (very wet)		X		

Table 4.7: conductivity sensor requirements following the MoSCoW method [27].

4.2.6 Measuring Humidity

The requirements for measuring CO_2 can be seen in table 4.8.

Requirement	Must	Should	Could	Won't
Easy to interface to local computer	X			
Humidity measurement every hour	X			
Usage Humidity of 0 to 100% relative humidity		X		
Accuracy of ±3% relative humidity, or better		X		

Table 4.8: CO_2 sensor requirements following the MoSCoW method [27].

4.2.7 Measuring Vibration

The requirements for measuring vibration can be seen in table 4.9.

Requirement	Must	Should	Could	Won't
Non-directional measurements	X			
Respond to vibration	X			
Easy to interface to local computer	X			
Continuous measurement	X			

Adjustable high-sensitivity measurements		**X**		

Table 4.9: vibration sensor requirements following the MoSCoW method [27].

4.2.8 Computer

The requirements for the computer platform can be seen in table 4.10.

Requirement	**Must**	**Should**	**Could**	**Won’t**
Enough RAM to store sensor data	**X**			
Enough ROM to store the local program and sensor data	**X**			
Low power solution of at least < 500 mW		**X**		
Wifi connectivity		**X**		
Digital IO			**X**	
Analog IO			**X**	
3/4/5G module			**X**	
External antenna			**X**	

Table 4.10: vibration sensor requirements following the MoSCoW method [27].

4.2.9 Computer Cloud platform

The requirements for the computer platform are shown in table 4.11.

Requirement	**Must**	**Should**	**Could**	**Won’t**
Connection over the internet	**X**			
Store all sensor data	**X**			

Store data for at least 90 days		X		
Show the number of sensor measurements.		X		

Table 4.11: vibration sensor requirements following the MoSCoW method [27].

4.2.10 Communication with the Cloud Platform

The requirements for communication with the cloud platform are shown in table 4.12.

Requirement	Must	Should	Could	Won't
Wireless communication	X			
Connect with the computer cloud platform	X			
The communication module must be easy to interface with.		X		
Communication speed > 1kb/s		X		

Table 4.12: Communication module requirements following the MoSCoW method [27].

4.3 Concept phase

In the background research in Chapter 2, several soil pollution indicators were identified that will be used in the sensor system: vibration, conductivity/humidity, temperature, CO_2, and pH. As mentioned in Chapter 2.8, NO_2 is not abundant in the ground, so the N_2O sensor will not be included in this project. Besides these indicators, a small computer that manages the sensors and a transmitter that sends the sensor data to the computer cloud platform will be included.

Measuring the pollution indicator CO_2 in the soil is difficult. CO_2 is a gas that floats up in the air due to its density. To correctly measure only the soil CO_2, a PVC pipe will be put in the ground with a gap at the bottom where the soil CO_2 can flow up [28]. Once the CO_2 is in the PVC pipe, it is water- and air-tight, after which the CO_2 flows up to the CO_2 meter, where the sensor measures the CO_2 levels and directs this data to the small computer and computer cloud. The other sensors are more straightforward and require a less difficult setup. These sensors will be placed alongside the PVC pipe respectively. A rough sketch of the concept can be viewed in Figure 4.1.

The computer module, the transmitter, and the CO_2 sensor will be placed on top of the PVC pipe in a watertight and construction site-proof box.

The final system should store the sensor data in an existing cloud solution, like the Arduino cloud. This service stores sensor data and has features like a control center. It also has machine learning applications and the ability to create graphs that visualize the sensor's data over a long period of time [29]. Sensor data can be exported from the cloud to other places, like Excel.

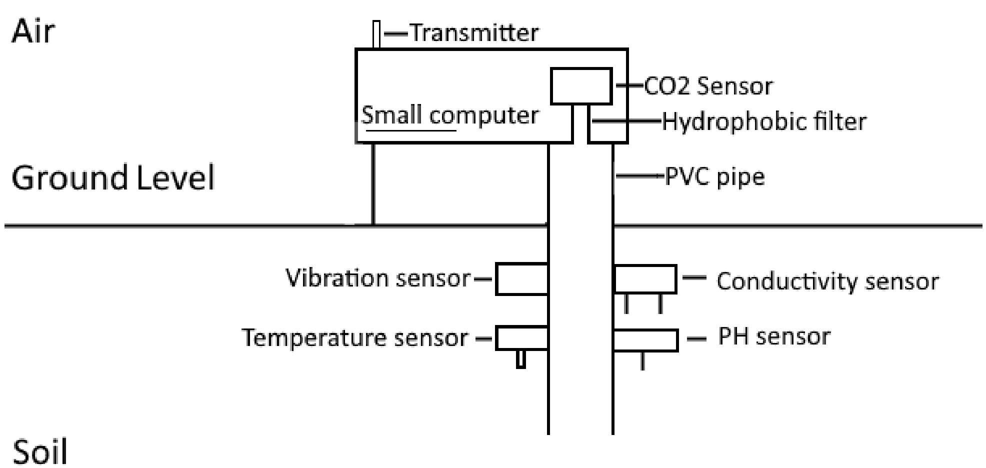

Figure 4.1: concept picture

Chapter 5 Functional Specification

In the specification phase, the final concept from the ideation phase is further elaborated upon. This is done in several steps. First, the sensors used to create the final product are elaborated on and compared with the requirements from the concept phase. Second, the integration and functionality of these sensors within the overall system are established.

5.1 High-Level System Requirements

This section provides detailed information about the sensors selected for the final product, including their specifications, how they meet the project requirements, and their integration within the system. In addition to the sensors, other system requirements, like an internet connection or a cloud platform, will be elaborated on.

5.1.1 Measuring pH

According to the requirements outlined in chapter 4.2.2, the final prototype must measure pH in some way. To do this, a pH sensor will be used, which must be able to measure pH consistently at hourly intervals and easily interface with the required program, such as Arduino. Furthermore, it should adhere to specific range, accuracy, and response time criteria.
The selected Rs485 sensor seen in Figure 5.1 with a built-in pH sensor is able to meet these criteria:

- Measurement Range: The Rs485 sensor demonstrates a pH measurement capability within the range of 3-9 pH units. This is less than the 0-10 pH units required, but it should practically be enough for a prototype.
- Resolution and Accuracy: With a resolution of ±0.1 pH units, the sensor ensures precise pH measurements, which is crucial for accurately monitoring soil conditions. This meets the system requirements set in chapter 4.2.
- Real-time Monitoring: The sensor facilitates real-time pH monitoring.
- Ease of Interface: The Rs485 sensor is easy to integrate with the Arduino computer program, ensuring ease of use when integrating it into the total system.

Figure 5.1: picture of the combination sensor used in the prototype.

5.1.2 Measuring temperature

In line with the specifications outlined in Chapter 4.2.3, the final prototype must measure temperature. To accomplish this, a temperature sensor will be used, which is expected to provide consistent temperature measurements at intervals of 10 minutes and an easy interface using a designated programming program like Arduino. Additionally, it should meet defined criteria regarding measurement range, accuracy, and response time. The chosen temperature sensor in the combination sensor Rs485 seen in Figure 5.1 satisfies these requirements as follows:

- Measurement Range: The temperature sensor's range of -40 to 80 degrees Celsius is more than enough to cover the required temperature range of -20 to 50 degrees Celsius.
- Resolution and Accuracy: The temperature sensor is highly accurate, typically ±0.5 degrees Celsius or better, ensuring precise temperature measurements. This meets the required standard of chapter 4.2, which is also ±0.5 degrees.
- Real-time Monitoring: The sensor facilitates real-time temperature monitoring, allowing the detection of temperature variations within the soil environment in real-time. This means it will easily comply with the 10-minute interval.
- Ease of Interface: The sensor's compatibility with standard interface protocols makes integration with the Arduino program easy. This simplifies the process of incorporating temperature data into the overall system, ensuring ease of use and efficiency in operation.

5.1.3 Measuring conductivity

In line with the requirements in Chapter 4.2.5, the final prototype must measure conductivity. To do this, a conductivity sensor will be used, which is expected to consistently measure conductivity at hourly intervals and integrate seamlessly with the designated programming platform, such as Arduino. Additionally, it should adhere to specific criteria concerning measurement range, accuracy, and response time. The conductivity sensor integrated within the Rs485 combination sensor seen in Figure 5.1 fulfills these requirements as follows:

- Measurement Range: The conductivity sensor's measurement range of 0-20000μS/cm adequately covers the system's requirements for conductivity measurement, which was set at 200 to 2000.
- Resolution and Accuracy: The sensor provides precise conductivity measurements with an accuracy of ±3% FS for conductivity levels between 0-10000μS/cm and ±5% FS for levels between 10000-20000μS/cm.
  Its resolution of 1μS/cm ensures that even small changes in soil conductivity are accurately captured.
- Real-time Monitoring: The sensor enables real-time monitoring of conductivity levels.
- Ease of Interface: The sensor's compatibility with standard interface protocols makes integration with the Arduino programming platform straightforward.

5.1.4 Measuring humidity

In accordance with the requirements in Chapter 4.2.6, the final prototype must measure humidity to accomplish this humidity sensor will be used, which is expected to provide consistent humidity measurements at hourly intervals and interface smoothly with a designated program, such as Arduino. Additionally, it should meet specific criteria for measurement like range, accuracy, and response time. The humidity sensor integrated within the Rs485 combination sensor seen in Figure 5.1 aligns with these requirements as follows:

- Measurement Range: The humidity sensor covers a range of 0-100%, meeting the system's requirements for humidity measurement.
- Resolution and Accuracy: The sensor ensures precise humidity measurements with an accuracy of ±2% within the range of 0-50% and ±3% within the range of 50-100%. This is better than the requirements set in chapter 4.2.
- Real-time Monitoring: The sensor facilitates real-time monitoring of humidity levels.

- Ease of Interface: The sensor's compatibility with standard interface protocols makes integration with the Arduino programming platform seamless.

5.1.5 Measuring CO_2

As per the requirements outlined in Chapter 4.2.4, the final prototype must measure CO_2. To accomplish this, a CO_2 sensor will be used, which is expected to provide consistent CO_2 measurements at 10-minute intervals and be easily integrated with a designated program, such as Arduino. Additionally, it should meet specific criteria concerning measurement range, accuracy, response time, and power consumption. The CO_2 sensor selected for the prototype integration is the SCD41 CO_2 sensor seen in Figure 5.2. It aligns with the requirements as follows:

- Measurement Range: The CO_2 sensor covers a measurement range of 400-2000 ppm. Although this is lower than the requirements set in chapter 4.2, it should still provide a general indication of CO_2 release in the soil.
- Accuracy and Sensitivity: The sensor provides precise CO_2 measurements with an accuracy of ± (40 ppm + 5%), which is more than enough to draw a good conclusion about the results.
- Real-time Monitoring: The sensor facilitates real-time monitoring of CO_2 levels, allowing for timely detection of fluctuations in soil CO_2 concentration.
- Ease of Interface: The sensor's compatibility with standard interface protocols makes integration with the Arduino programming platform straightforward.

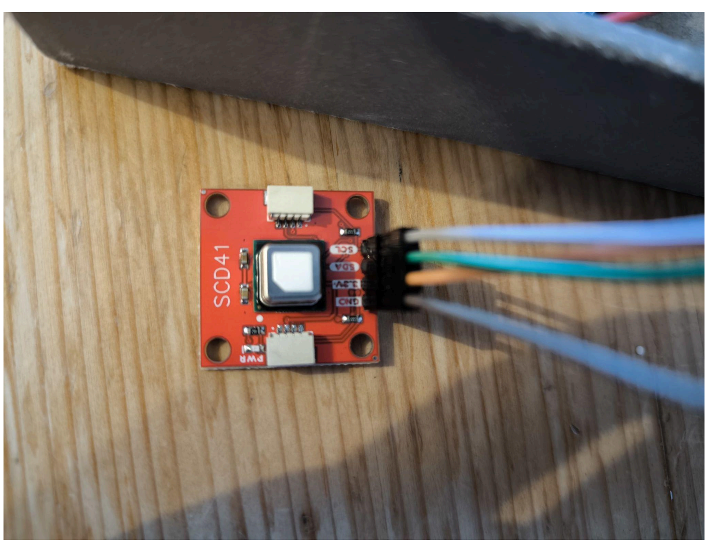

Figure 5.2: CO_2 sensor used in the prototype.

5.1.6 Measuring vibration

In accordance with the requirements outlined in Chapter 4.2.7, the final prototype must measure vibration. To do this, a vibration sensor is used, which is expected to provide reliable detection of vibrations, interface seamlessly with the designated program (such as Arduino), and operate continuously. Additionally, it should be non-directional and offer adjustable high-sensitivity measurements. The selected vibration sensor is the SW-420. The sensor meets these requirements as follows:

- Vibration Detection: The sensor is able to effectively detect vibrations in its vicinity, enabling the system to monitor ground vibrations accurately.
- Non-directional Sensing: The sensor provides non-directional measurements, ensuring that vibrations from all directions are detected equally. This ensures comprehensive monitoring of ground vibrations, regardless of their source or direction.
- Continuous Operation: The sensor operates continuously, allowing for uninterrupted monitoring of ground vibrations over extended periods. This continuous operation ensures that no significant vibration events are missed, effectively enhancing the system's ability to detect anomalies.

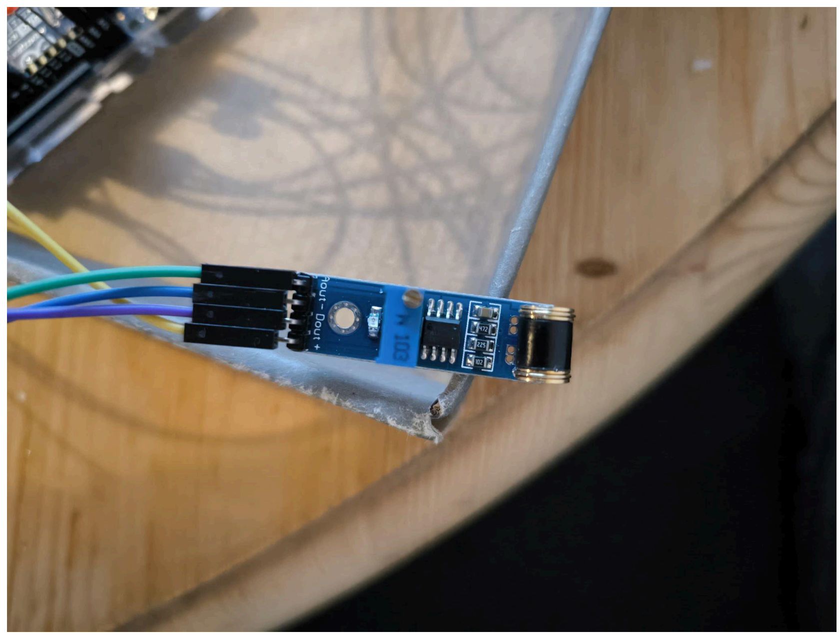

Figure 5.3: The vibration sensor used in the prototype.

- Adjustable High-Sensitivity Measurements: The sensor offers adjustable high-sensitivity measurements, enabling users to customize sensitivity levels based on specific monitoring requirements. This flexibility ensures optimal performance in various environmental conditions and applications.
- Ease of Interface: The sensor's compatibility with standard interface protocols makes integration with the Arduino programming platform seamless. This simplifies the process of incorporating vibration data into the overall system, ensuring ease of use and efficient data integration.

5.1.7 Computer

One of the requirements for building the system is the need for a computer that can connect, integrate, and send the data collected from the sensors as per the requirements of 4.2.9. It needs enough RAM to store sensor data and enough ROM to store the local program and sensor data. Besides this, it could be helpful if it has a low power consumption, wifi connectivity, digital and analog IO, or other methods to send and receive data. The selected computer for the prototype is an Arduino Uno without a built-in wifi connection. The Arduino meets the requirements as follows:

- Enough RAM and ROM to store sensor and program data: An Arduino Uno has 2,048 bytes of RAM. This is not a lot, but it is plenty to store a lot of sensor and program data before it is sent to the cloud platform. Besides this, it has 512 Bytes of EEPROM, which is a memory whose values are kept when the board is turned off. Both of these memories should store enough data to let the prototype run correctly.
- The Arduino mainly runs when connected to a computer cable but also can be connected to a 9v battery. If this is not enough, an outside power supply must be installed.
- Since a normal Arduino Uno does not support Wi-Fi connections, the Esp8266 ESP-01 will be used. If connected to the Arduino, this board can send data over Wi-Fi to the cloud platform.
- An Arduino Uno board has 14 digital IOs and six analog IOs, which should be enough to connect all the sensors and Wi-Fi modules.

5.1.8 Computer Cloud Platform

In line with the requirements outlined in Chapter 4.2.9, the computer cloud platform is expected to provide reliable data storage, easy access over the internet, and compatibility with the Arduino programming environment. Additionally, it should store sensor data for at least 90 days and display all the sensor data in easy-to-read graphs. The chosen computer cloud platform is the Arduino cloud program maker package, which meets the requirements as follows:

- Internet Connectivity: The Arduino cloud platform allows an easy-to-use Wi-Fi connection after inserting the API key through Arduino code. For this work, you do need some special Wi-Fi boards, but there are plenty that work, so that should not result in problems.

- Data Storage: The platform can store all sensor data and retain it for a maximum of 90 days. This extended storage capacity ensures historical data is available for trend analysis and long-term monitoring.
- Display of Sensor Measurements: The platform provides features to display the number of sensor measurements, facilitating easy tracking and analysis of data collected over time. This visualization aids in identifying patterns and anomalies in the monitored parameters and allows for easy comparisons between all the sensor measurements.
- Ease of Interface: Integration with the Arduino program is straightforward, thanks to the platform's support for standard data formats and protocols. This compatibility simplifies uploading sensor data to the cloud and retrieving it for analysis.
- Security and Reliability: The platform ensures secure data transmission and storage, protecting sensitive sensor data from unauthorized access. Arduino claims that security must be updated at least once a month.

5.1.9: Communication with the cloud platform

For communication with the cloud platform, there are multiple options possible, including:

- 4/5G module: This uses the same protocols as a mobile phone [39].
- LoRa module. Lora stands for long-range and is often used to connect low-power sensor modules. A LoRa network with sufficient coverage is needed, like the KPN LoRa network that covers the whole Netherlands [38].
- Mesh network: With a mesh network, different sensor modules can be connected with each other. Possible short-range options include Bluetooth and ZigBee. One node in the mesh network needs a connection to the Internet [40].
- Wifi: This is an easy solution if a wifi network is available on the construction site [38].

Wi-Fi communication is chosen for the first prototype. To maximize connection range, the WiFi module must connect to a 2.4 GHz wifi network [37]. This kind of connection ensures a larger coverage area while also being better at penetrating solid objects; this ensures that it can be more easily used at big construction sites. The module must also connect to the computer cloud system to ensure that the sensor data can be sent from the Arduino to the cloud over WiFi. Besides this, the WiFi module should also be able to connect to a 5 GHz WiFi network and be easy to interface with. The chosen WiFi module is the Esp8266 ESP-01, which meets the requirements as follows:

- 2.4 GHz connection: The Esp8266 ESP-01 has full 2.4 GHz connection capabilities.
- Connection with computer cloud system: As of right now, it is uncertain whether this board can connect to the computer cloud system. This will have to be further researched during the realization. If it is not possible to connect the esp8266 ESP-01 to the computer cloud system, then an alternative, like the Arduino R4 WiFi board, which has integrated WiFi, will have to be found.
- Easy to interface: The esp8266 ESP-01 seems easy to interface since many tutorials are available for the WiFi module, but this will also have to be researched in the realizations.
- 5.0 GHz connection: the esp8266 ESP-01 does not seem to have a 5.0 GHz connection, but this should not matter as it still has 2.4 GHz connection capabilities.

5.2 Functional Architecture and time-sequence diagram

The following diagram will be utilized to explain how the system will operate on a broad level. First, a functional architecture diagram will be created. A functional architecture diagram is a visual representation of all the different components of a system and how they interact with each other to perform a specific function. In the case of this research, that means the data flow from the sensor to the computer cloud system. The second diagram that will be utilized to explain the system is a time-sequence diagram. This diagram shows all the objects and actors that interact in the system and shows the timeline of the interactions in the system.
The functional architecture diagram in Figure 5.2 shows the connection between all system parts.
The system contains three major parts:

1. The external computer: This computer is used for software development and debugging. It is connected directly to the Arduino and can be removed if the code is flashed on the Arduino. In that case, the computer does not power the system anymore, and a separate power supply is needed, as seen in Figure 5.1.
2. The sensor system, which includes the Arduino microcontroller and sensors connected to the Arduino via cables
3. The computer cloud system: The Arduino will collect data from the sensors and use the communication module to send this data to the cloud system. The cloud system stores the sensor data in a database and can show it in easily readable graphs.

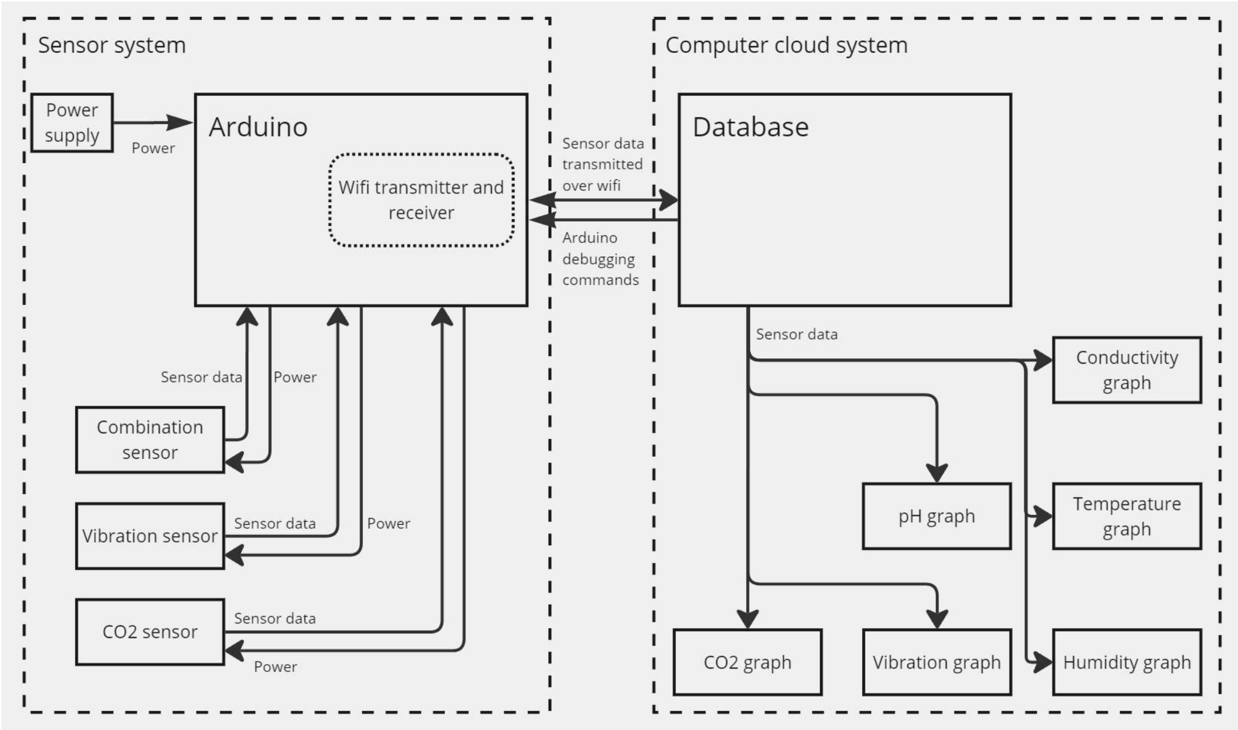

Figure 5.1: Functional architecture diagram of the full system with a power supply.

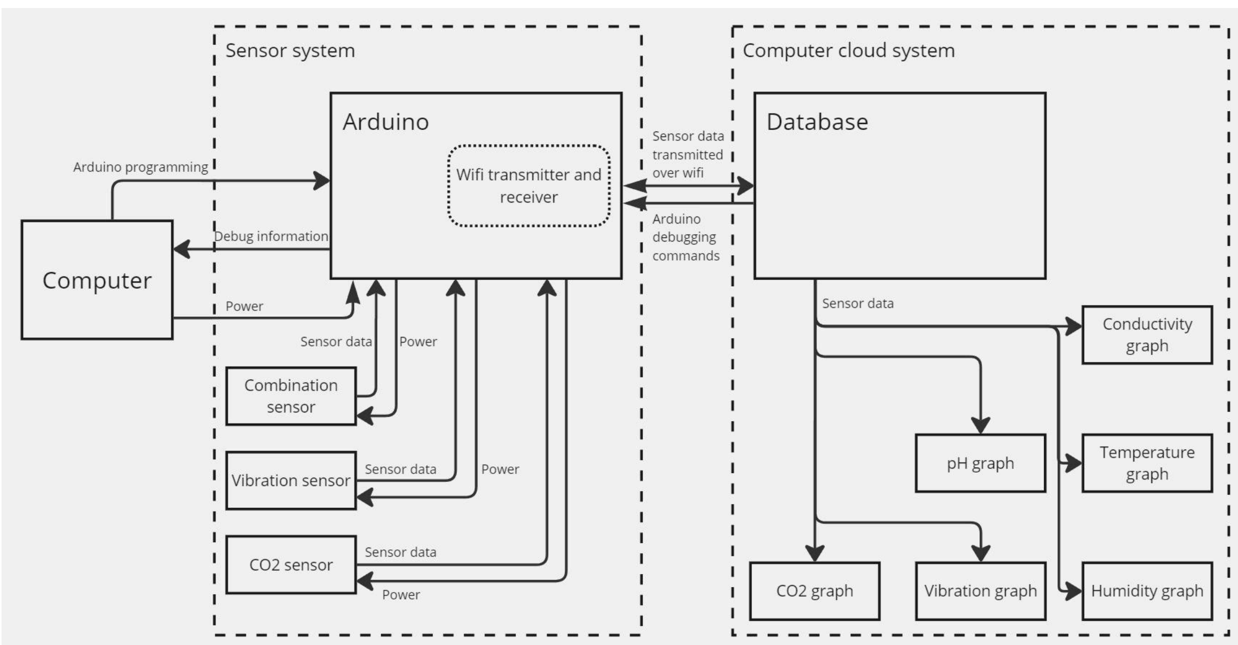

Figure 5.2: Functional architecture diagram of the system with a computer for development and debugging.

The time-sequence diagram is shown in figure 5.3. It is the representation of the system's time flow. The system starts with the user placing the sensor system at the construction site, after which the sensor system will start automatically measuring data and sending this to the computer cloud. When the user opens the table interface of the computer cloud, he can see the graphs with the measured sensor data. These graphs will keep updating as long as the user is on this page. The system will stop when the user removes the sensor system from the construction site.

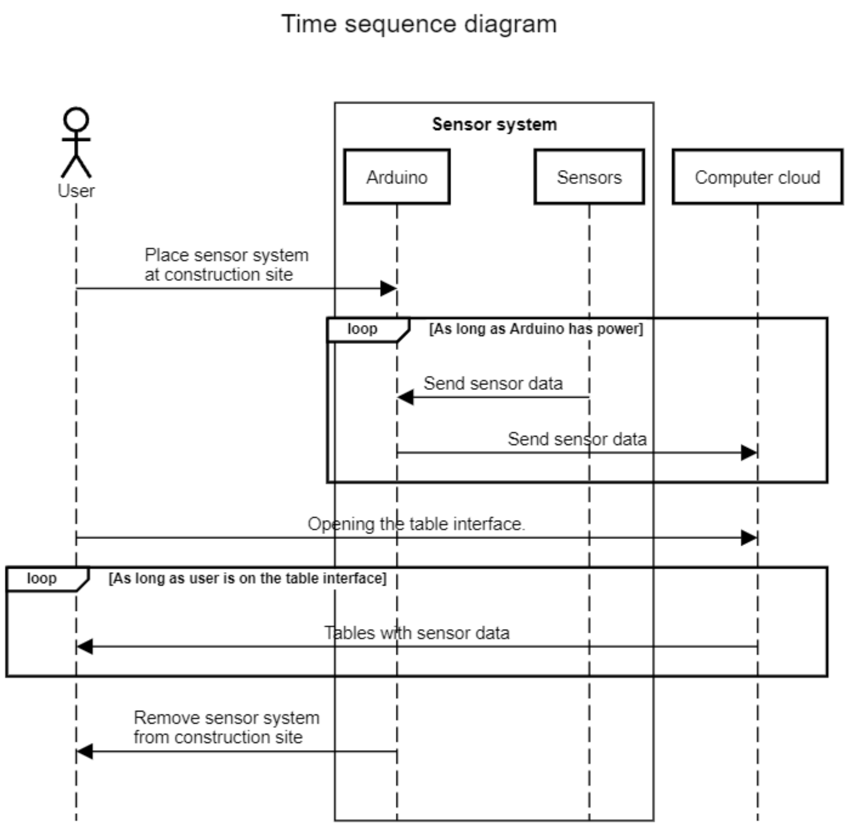

Figure 5.3: time-sequence diagram of the whole system.

Chapter 6 Realization

In this chapter, the creation of the whole sensor system will be explained. The first steps to creating the final sensor system are to connect all the different sensors individually and make the sensors output correct data. The second step is to combine all the sensors into one Arduino and code. The last step is to connect it to the computer cloud system and show the correct graphs with the sensor data on the cloud.

6.1 Arduino board

The computer used for the sensor system is the Arduino Uno, which uses the Renesas RA4M1 Microcontroller. The Arduino Uno has fourteen digital and six analog pins. Most pins are programmable and can be used for different functions. It contains 256 kB flash memory to store the program and data and 32 kB SRAM for data [36].
For communication, it has the following three widely used serial protocols built in:

- Serial Protocol. This is a generic asynchronous serial protocol. Can be used for larger distances.
- I2C Protocol. I2C is a serial protocol mainly used for communication with lower-speed peripherals close to the microcontroller.
- SPI Protocol. The standard for synchronous serial communication is mainly used in embedded systems to connect close-by integrated circuits.

The fourteen digital pins (D0 to D13) can be programmed as input or output and are also used for serial communication with one or more of the built-in serial protocols.

There are six analog pins (A0 to A5). The main purpose of the analog pins is to read analog sensor data. For this purpose, a 10-bit analog-to-digital Converter (ADC) is connected to each pin internally. The ADC converts the analog signal level to a digital number.

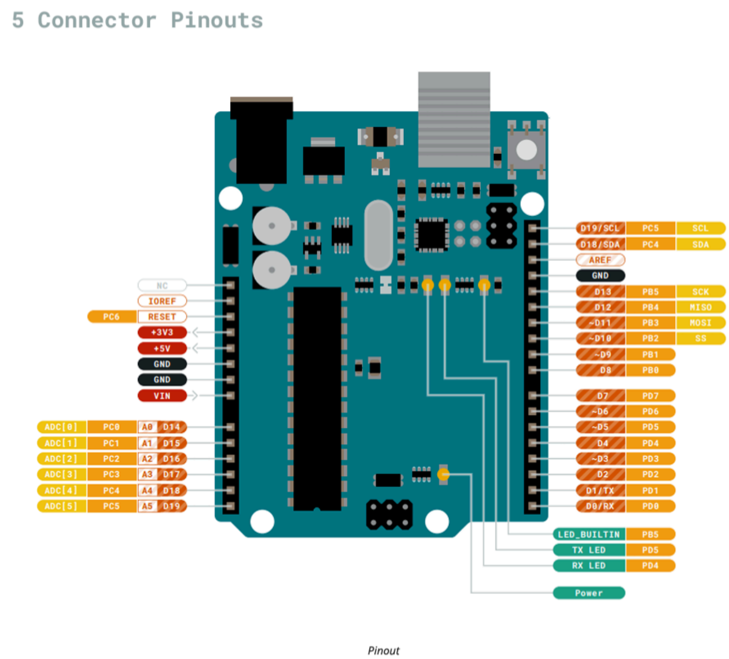

Figure 6.2: Pinout datasheet of Arduino UNO [30].

The Arduino Uno works with a voltage of 5V and can be powered with a voltage between 7V to 12V. The Arduino IDE (Integrated Development Environment) can be downloaded from the official Arduino website to develop and debug Arduino software. This development environment makes it easy to include needed libraries for the different sensors and has several debugging options.

After compilation, the software can be uploaded to the Arduino Uno's flash memory. Debugging information can be sent back to the computer via a serial connection between the computer and Arduino. The computer can be disconnected when the code works fine, and the Arduino can run independently.

6.2 CO_2 sensor

The CO_2 sensor is a critical component of our sensor system, designed to monitor and report the concentration of carbon dioxide in the soil. This chapter details the steps to integrate the CO_2 sensor into the overall system. From initial setup to the final integration with the Arduino Uno and the cloud-based monitoring system.

To connect the CO_2 sensor to the Arduino Uno, we use the I2C interface. The ground (GND) of the CO_2 sensor is connected to the ground (GND) of the Arduino, and the 3.3V pin of the CO_2 sensor is connected to the 3.3V pin on the Arduino. The CO_2 sensor needs two special I2C input/output pins in order to work: the SDA (serial data) pin and the SCL (serial clock) pin.

The Arduino Uno has one of each of these pins, SDA above the AREF input and SCL one above the SDA input, as seen in Figure 6.2 [30]. The full individual connection diagram of the CO_2 sensor can be seen in Figure 6.1.

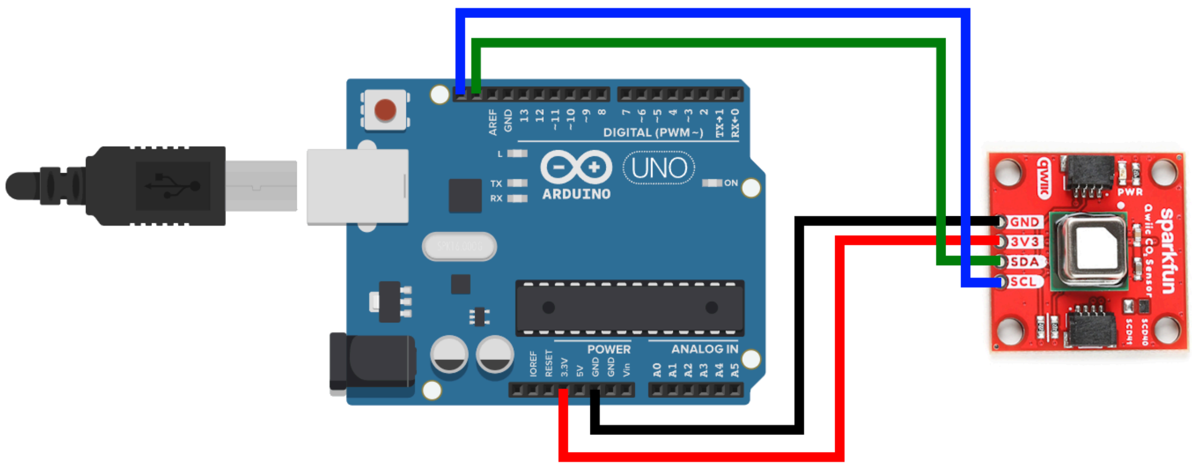

Figure 6.1: CO_2 sensor separate circuit diagram.

The individual CO_2 sensor code used to run and test the individual CO_2 sensor can be viewed in Appendix A. The code uses a library called “SparkFun_SCD4x_Arduino_Library”. This library is used to obtain the sensor data and turn this data into a simple function that can be called in the code and communicate with the sensor and the Arduino. In order to receive sensor data in the serial monitor, the baud rate needs to be 115200. Increasing this ensures that the data rate in bits per second is higher, which is needed to receive data from the sensor.

6.3 Combination sensor: temperature, conductivity, pH, and humidity

The combination sensor can measure temperature, conductivity, pH, and humidity. This sensor uses the RS485 protocol. The difference between the RS485 protocol and the serial protocol from the Arduino board are the signal levels. RS485 uses a balanced signal level. This means that the signal switches between negative and positive voltages.

A converter is needed to connect the Arduino with a device that uses the RS485 protocol. This converter converts the Arduino's 0 and 5 Volt levels to the required balanced RS485 Voltage. The MAX485 converter is used for the combination sensor.

To connect the Rs485 sensor to the Arduino Uno, I used the MAX485 converter and a breadboard since many connections needed to be made. The ground and 5v volt output cables are connected to the 5v and ground of the breadboard, which is connected to the 5v and ground of the Arduino. The blue cable of the sensor is connected to the B input of the converter, and the yellow cable of the sensor to the A input. The VCC input is connected to the 5V line of the breadboard and the ground of the converter to the ground of the breadboard, respectively. Finally, the R0 of the converter is connected to the eight digital pins of the Arduino. Similarly, the RE is connected to the sixth digital input, the DE is connected to the seventh digital input, and finally, the DI output is connected to the ninth input of the Arduino, as shown in Figure 6.3.

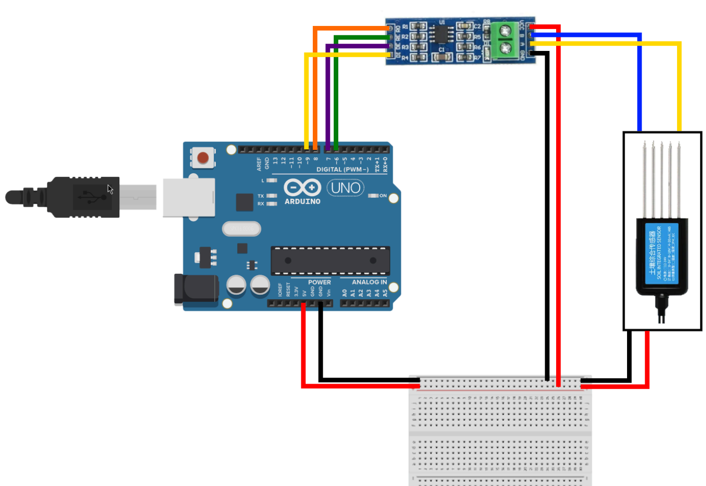

Figure 6.3: combination sensor separate circuit diagram.

The individual combination sensor code used to run and test the individual combination sensor can be viewed in Appendix B. Like the CO_2 sensor, the combination sensor uses a library called “ModbusMaster” to run. This library handles the communication between the Arduino and RS485 sensor and converter. In order to send sensor data to the serial monitor, the baud rate is also set to 115200. Finally, the code seen in Appendix B is mainly used to test if the sensor gives measurements and not to measure the intended values. The correct sensor code will be used in the combined system.

6.4 Vibration sensor

The vibration sensor is in the system to measure vibrations in the ground. This data can be used to see if there are connections between vibrations measured in the ground that are likely from machinery on the construction site and other indicators such as CO_2 or conductivity.

The first sensor tested for this research was the SW-420 vibration sensor. To connect this to the Arduino Uno, the VCC should be connected to the 5V port of the Arduino. The sensor's ground is connected to the ground of the Arduino, and the sensor's digital port is connected to the digital 2-pin of the Arduino, as seen in Figure 6.4.

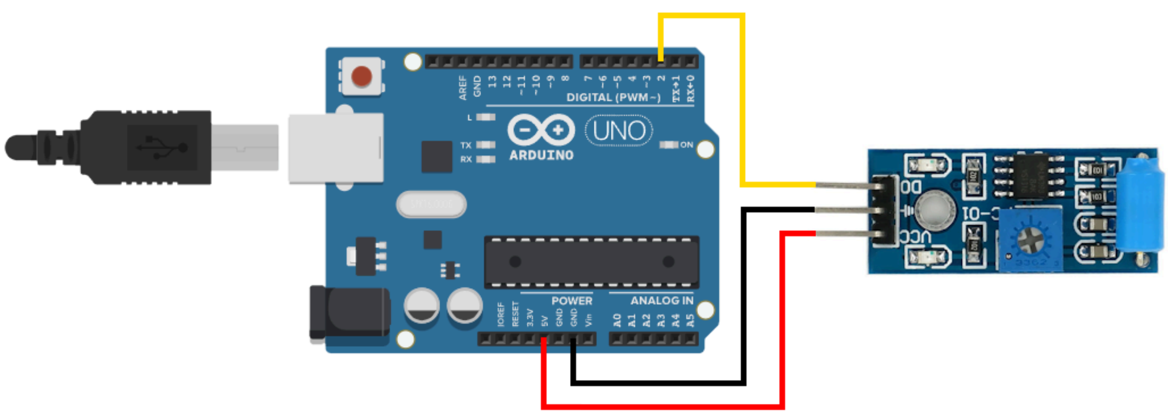

Figure 6.4: SW-420 vibration sensor separate circuit diagram.

The code used to test the SW-420 vibration sensor can be viewed in Appendix E. The code seen in Appendix E uses an additional LED that is only used for testing and does not influence the sensor's working. The big problem with this sensor is that it only has digital output. The sensor outputs a 0 if it detects no vibrations, and when it does detect vibrations larger than a set level, it outputs a 1.

However, for the purpose of this research, it is essential to know the strength of the vibrations. That makes it easier to see if there is a relationship between the vibrations and other measured data.

A vibration sensor with analog output can be used to determine the real strength of the vibrations. For this research, the 801S vibration sensor will be used. To connect the 801S vibration sensor to the Arduino Uno, the VCC of the vibration sensor is connected to the 5v of the Arduino. Similarly, the sensor's ground is connected to the ground of the Arduino, and the sensor's analog output is connected to the A0 analog input of the Arduino. The digital output is connected to the D2 input pin of the Arduino, as seen in Figure 6.5.

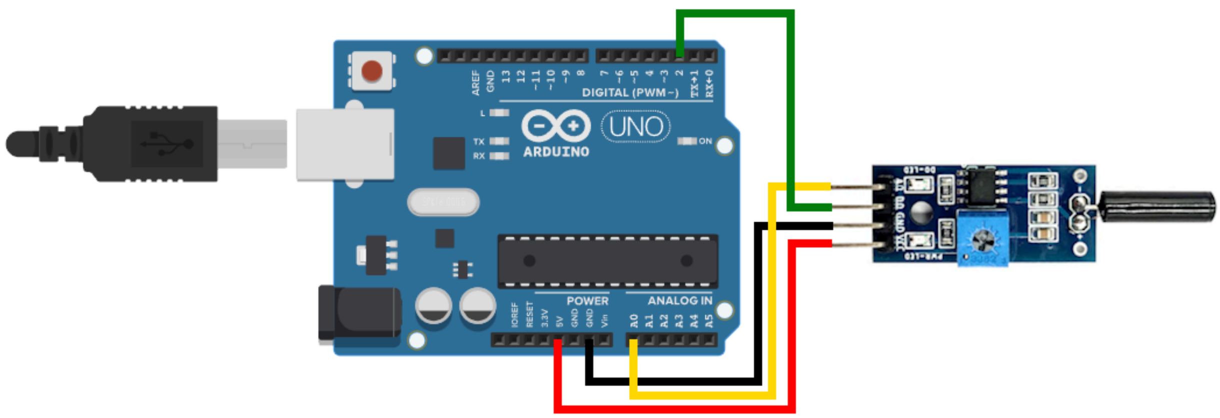

Figure 6.5: 801S vibration sensor separate circuit diagram.

The individual 801S vibration sensor code, which is used during testing and implemented in the final code, can be viewed in Appendix C. The code used to run the vibration sensor is very simple. The sensor is initialized and then read/printed on the serial port.

Despite this, the sensor did not output correct analog or digital values. This is likely due to a faulty sensor since the code of the previous sensor also did not work on the new sensor. However, a new sensor could not be purchased and tested due to time constraints.

6.5 Wifi Connection

Wi-Fi is an essential part of the sensor system, and it is responsible for sending the sensor data received on the Arduino to the computer cloud system. To do this, the Esp8266 will be used. This is a PCB that can receive and send items over the Internet, which will be used to send the sensor data from the Arduino to the computer cloud system.

Connecting the esp8266 to the Arduino Uno is quite simple: The 3.3V and EN ports of the esp8266 have to be connected to the 3.3V input of the Arduino, the ground of the esp8266 to the ground of the Arduino, RX (Receive) to the third digital input of the Arduino, and TX (Transmit) to the second digital input of the Arduino, as can be seen in figure 6.6.

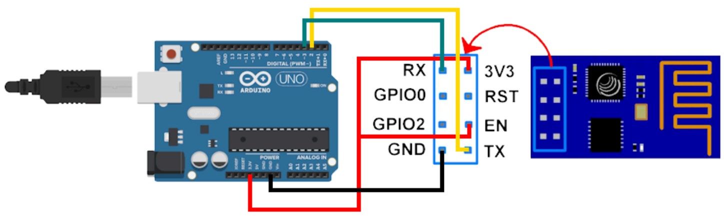

Figure 6.6: Esp8266 separate circuit diagram.

Special commands are used to connect the Esp8266 to the internet. These software commands communicate with and control the Wi-Fi modem. The softwareSerial.h library allows serial communication on other digital pins of an Arduino board. The code used to test the Wi-Fi connection can be seen in Appendix F. During the initial testing of the connection with the cloud environment, it was found that the connection was not stable.

Because of this problem, using the Esp8266 board was not feasible. The solution was to switch to an Arduino Uno R4 Wi-Fi, as shown in Figure 6.7. This board is from the same family, connected correctly (and stable) with the cloud environment. The Uno R4 Wi-Fi has the same analog and digital connections, using the same pins on the connectors. The only difference with the Arduino Uno is that it has a Wi-Fi chip and antenna built in.

The Arduino Uno R4 wifi board will be used to integrate the full sensor and cloud systems.

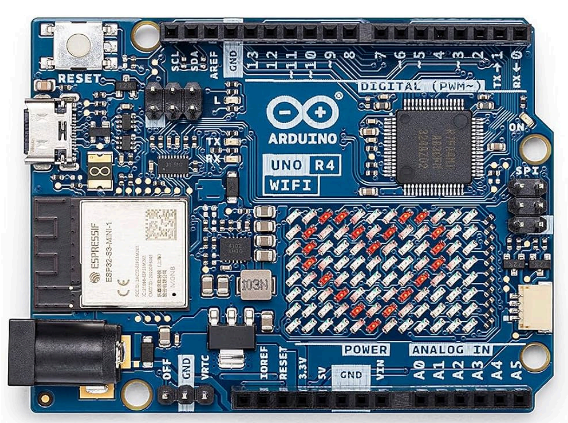

Figure 6.7: Arduino R4 wifi board

6.6 Cloud connection

To store the sensor data sent from the Arduino, the Arduino cloud platform is used. In order to get the sensor data to the Arduino cloud a WiFi connection is used to connect the Arduino sensor platform with the Cloud Platform.

For the storage of the sensor data, the first step is to define the various variables that will be linked with the sensors. Each variable that is added has to be defined, meaning that, for example, the incoming CO_2 data has to be defined as a float inside the program:

- Integer: for values that are a whole number (not a fraction)
- Floating Point: for values that are a fraction
- Boolean: for values that only need 2 states, like Yes or No

For each sensor value that is sent to the cloud, a cloud variable is created:

- CO_2 uses a floating point variable to store the CO_2 sensor data.
- Conductivity uses a floating point variable to store the conductivity sensor data.
- Humidity uses a floating point variable to store the humidity sensor data.
- pH uses a floating point variable to store the pH sensor data.
- Temperature uses a floating point variable to store the temperature sensor data.
- Vibration uses a floating point variable to store the vibration sensor data.

The stored sensor variables can be seen in figure 6.8.

Cloud Variables

Name ↓	Last Value
Co2 `float co2;`	628
conductivity `float conductivity;`	19.1
humidity `float humidity;`	34.5
LED `bool lED;`	true
pH `float pH;`	4.9
temperature `float temperature;`	20.1
vibration `float vibration;`	358

Figure 6.8: stored sensor data in the Arduino cloud environment.

When the various cloud variables have been defined, the next step is to create a dashboard to display them. Each dashboard consists of one or more widgets that can be connected to one of the cloud variables, as seen in Figure 6.9.

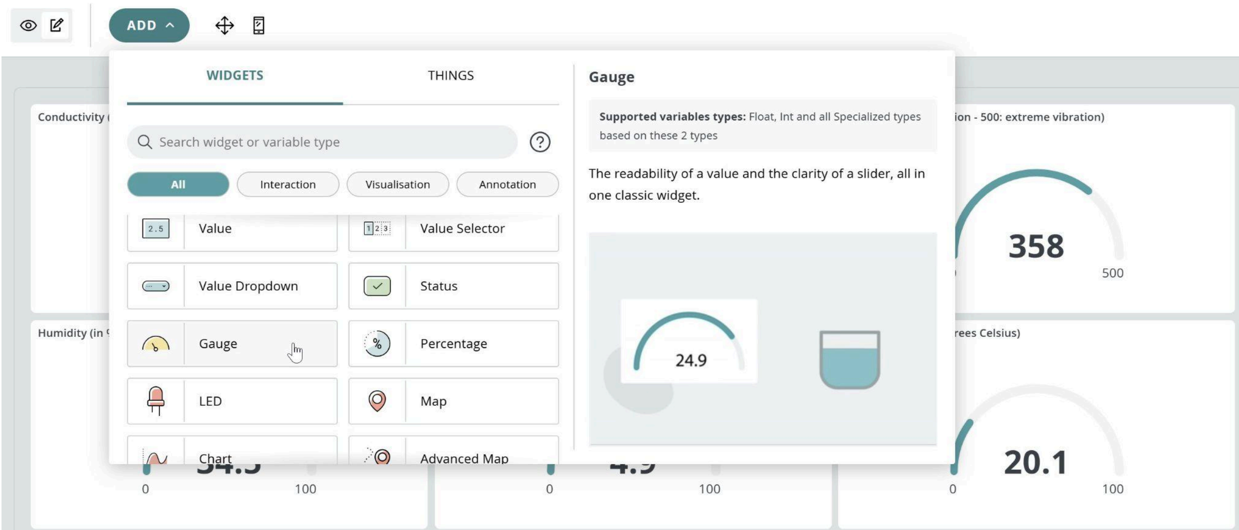

Figure 6.9: creation of widget in Arduino cloud dashboard.

For the dashboard, the following widget will be used:

- Gauge: This will show the last send value in both a gauge and in the middle as a number
- Chart: The chart widget shows the variable during a certain period of time. This period can be set by the user.

For the dashboard used in the system prototype, each sensor will be represented by both a gauge and a chart. This makes it easy to show the current value in the gauge widget and the history in the chart widget, as seen in Figures 6.10 and 6.11.

After adding the widget on the dashboard, it can be connected to the cloud variable, which should result in the sensor data received in the Arduino cloud being sent to the appropriate gauges and charts.

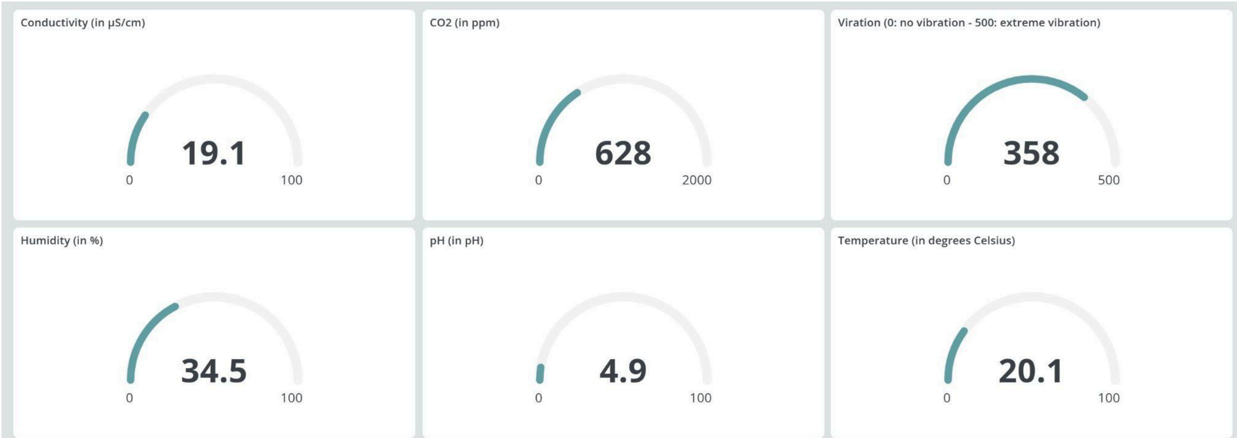

Figure 6.10: gauge widgets for all the sensors in the cloud dashboard.

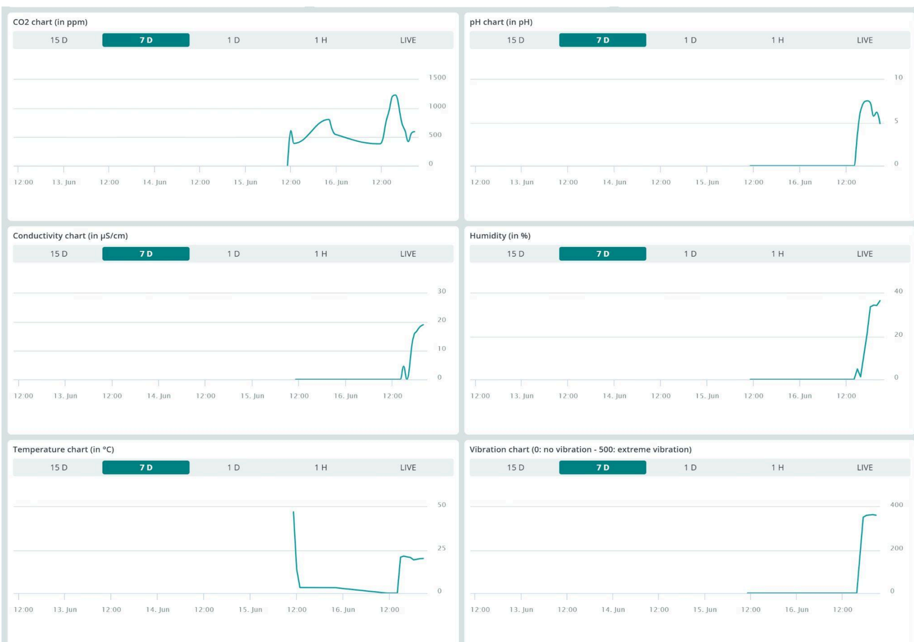

Figure 6.11: chart widgets for all the sensors in the cloud dashboard.

After creating the dashboard, the next step is to send the values from the sensors to the cloud platform using the wifi connection.

For this purpose, the Arduino Cloud Platform has a library called “thingProperties.h” is used to send values to the cloud. It creates a global variable for each cloud variable and manages the connection with the cloud (starting the connection, ending the connection, sending and receiving values), making the software's implementation straightforward. The code used in the Arduino cloud will be discussed in the sub-section of system integration.

6.7 System Integration

To complete the final system, all the components discussed in the chapters will be combined into one full system. The integration process involves three steps: Connecting all the individual components into one Arduino. This step includes checking if the Arduino has enough special pins to adhere to this step. The second step is integrating all the code into a single file and connecting this with the cloud platform. The final step is making a front end that shows graphs containing the sensor data.

6.7.1 Connecting all the sensors

The connection of all the sensors is quite simple in this research since the Arduino has enough special pins to accommodate all the sensors. In principle, the connection process follows the individual connection steps detailed in the preceding chapters, but instead of connecting each sensor independently, it is all integrated into a single Arduino unit. This unified setup is illustrated in Figure 6.5.

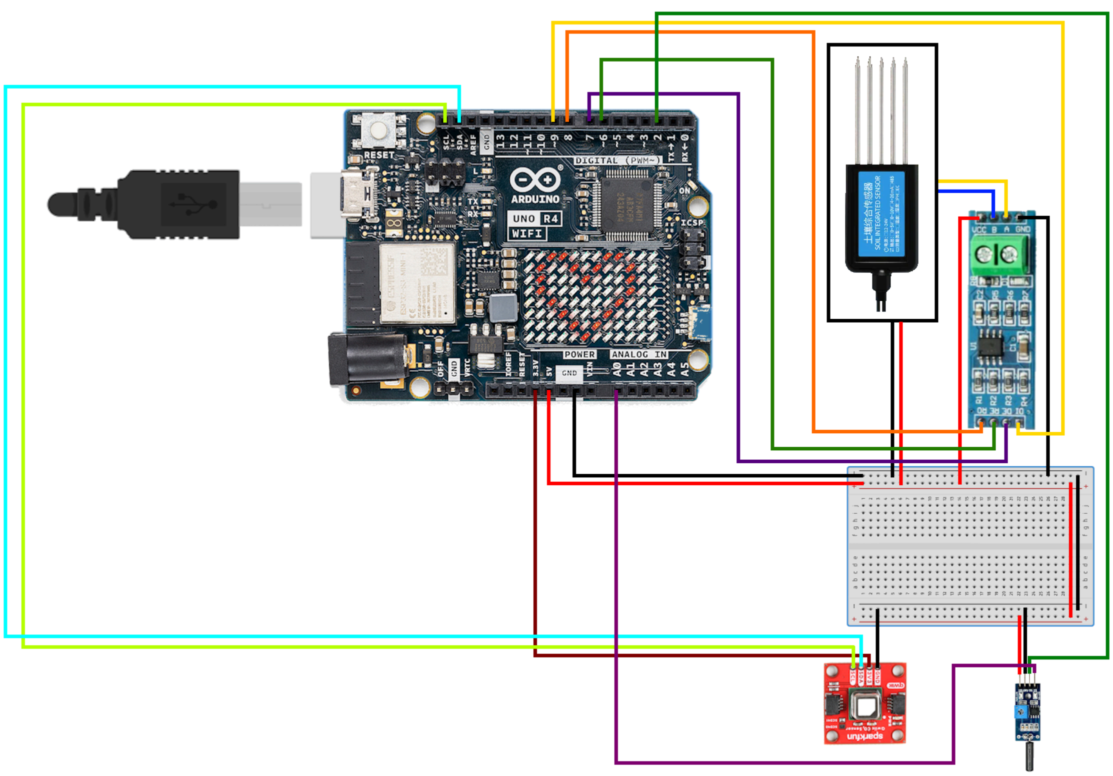

Figure 6.5: Full system circuit diagram.

6.7.2 Coding all the separate sensor codes

Code integration means the practice of combining all the codes into a new document and still making it return the proper values. At first, this code integration seemed pretty straightforward since all the sensor codes work individually. However, a few issues arose during the combination of the codes. The first issue is that the code had to be transferred to the Arduino cloud environment, which, in essence, is the same environment, but it did give some integration issues, especially due to the removal of the serial port usage to print the sensor data. The final code used in the sensor system can be seen in Appendix D. Important to note is the use of the serial port for the rs485 convertor. This sensor uses the serial port to communicate between the sensor and the converter. The combined code uses the individual codes used during the testing of the individual sensors, but instead of printing the values on the serial port, the variables that are linked to the dashboard are called. When these variables see a change in value, they are updated and can be viewed in the dashboard. One issue that could arise due to this is that a value is not updated on the dashboard for a long period of time, and since it stays the same, this could give the illusion that the sensors are not measuring.

Chapter 7 Measurement results

In this chapter, the sensors themselves and their expected correct working will be tested and validated. This data will be used to obtain a correctly grounded conclusion. The results of all the sensors will be shown, each with its own measuring methods, in a line graph obtained from the Arduino cloud dashboard.

Calibration is of utmost importance for correct measurement results. Only when the sensors are correctly calibrated can the data be reliably used. This applies not only to comparisons with similar sensor systems on the same construction site but also with other systems. Due to time constraints, this calibration step could not be done.

7.1 Temperature, conductivity and humidity

The temperature, conductivity, and humidity sensors were tested in four steps, as shown in Figures 7.1, 7.3, and 7.4.

The four measurement steps are as follows:

1. Turning on the system and doing a small test to see if the combination sensor reacts to small changes and a hand touch test at 20:40.
2. The combination sensor is placed in dry soil, as seen in Figure 7.2. It can be seen that the soil temperature is slightly less than the room temperature.
3. Water is added to the soil to make it wet. The water temperature was around 27 degrees Celsius. This heated the soil slowly.
4. Removing the combination sensor from the soil. The sensor went back to room temperature.

Numbers one to four represent the timespans seen in Figures 7.1, 7.3, and 7.4.

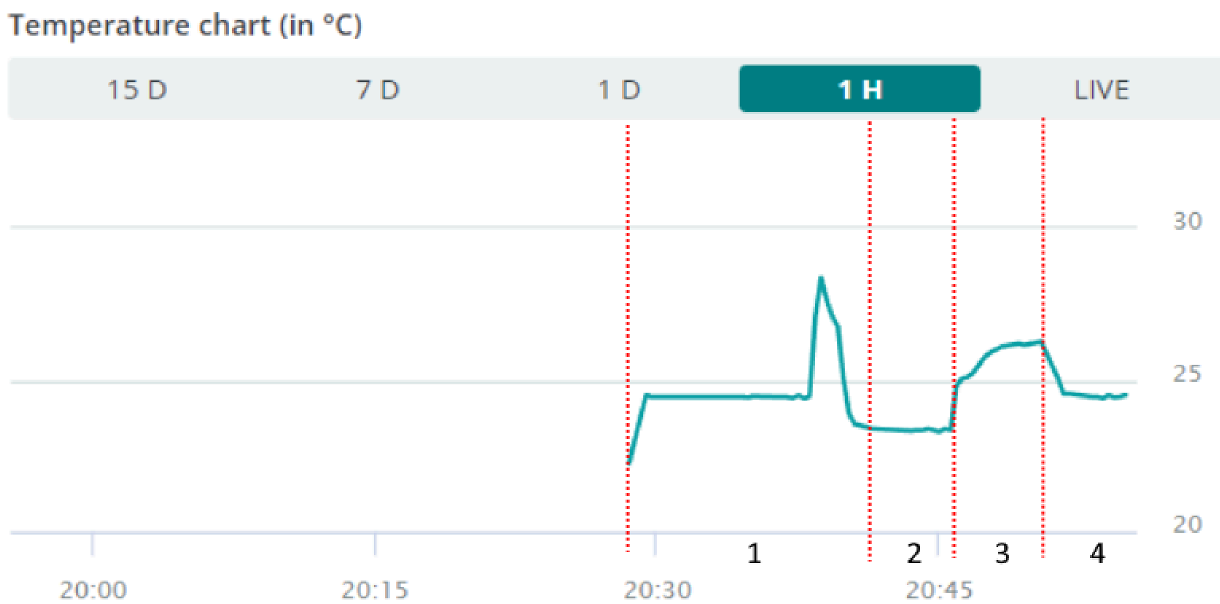

Figure 7.1: temperature sensor results.

Figure 7.2: Picture of testing setup.

Step one of the temperature sensor, as seen in Figure 7.1, starts off with the temperature going up from 22 degrees Celsius to 24 degrees Celsius. This small peak is likely due to the sensor trying to calibrate the correct temperature in the room. At around 20:39, there is a huge peak to 28 degrees and back down to 24 degrees. This happened when I touched the sensor's probes to see if it correctly reacts to temperature change, and since my hand is approximately 36-37 degrees, it should start to rise quickly, which it does [34]. At approximately 20:40, the sensor is put in soil, which is a bit below room temperature. At approximately 20:47 seconds, room temperature water is poured into the soil, but it eventually spikes up to around 27 degrees Celsius, which can not logically be explained since room temperature water should be around 25 degrees Celsius [35]. At 20:50, the sensor is removed from the soil and returned to room temperature, after which it settles again at 24 degrees Celsius.

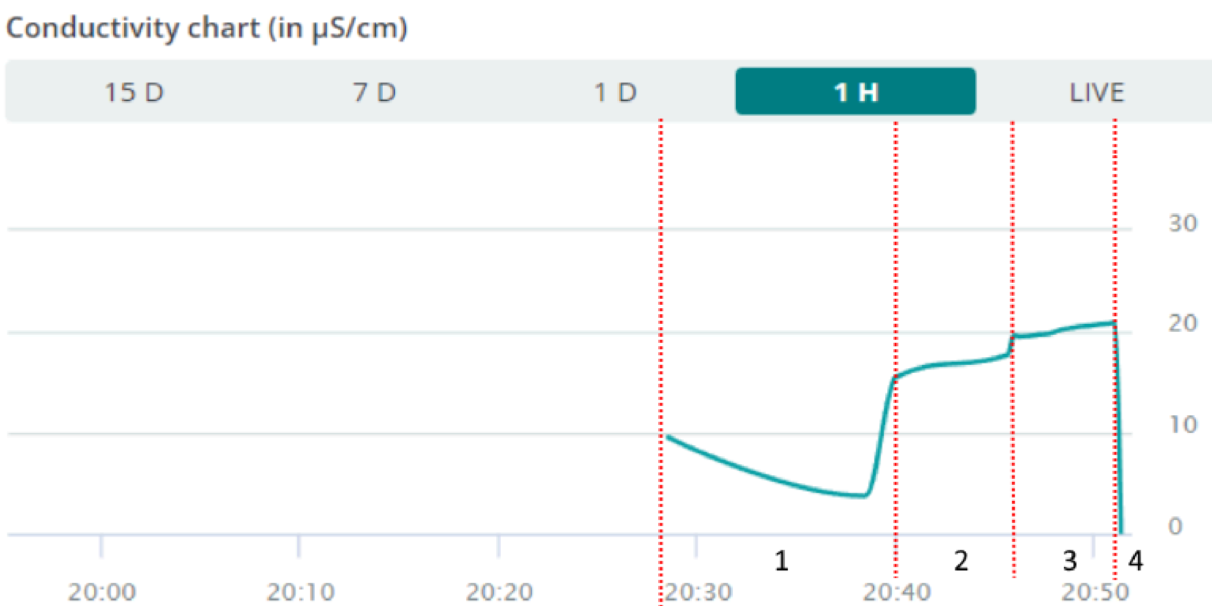

Figure 7.3: conductivity sensor results.

Step one of the conductivity sensor, as seen in Figure 7.3, is the testing phase of the sensor. It starts off with a conductivity of 10 μS/cm, after which it slowly goes down. Since the sensor is now in the air, it should be close to 0 [33], so the sensor giving values at 10 μS/cm can not be explained, but the sensor data slowly going down to 0 makes sense. At 20:39, I touched the sensor, after which the conductivity rose very quickly, which makes sense since the hand has more conductivity than the air. At 20:40, the sensor is put into the soil, after which it keeps around 15 μS/cm but keeps rising slightly, which does not make sense. At 20:45, the water is added, after which the conductivity rises slightly and settles at 20 μS/cm. The slight rise after adding water makes sense since water conducts more electricity than dry soil. After that, the sensor is removed, after which it instantly drops to 0, which is the sensor's expected behavior.

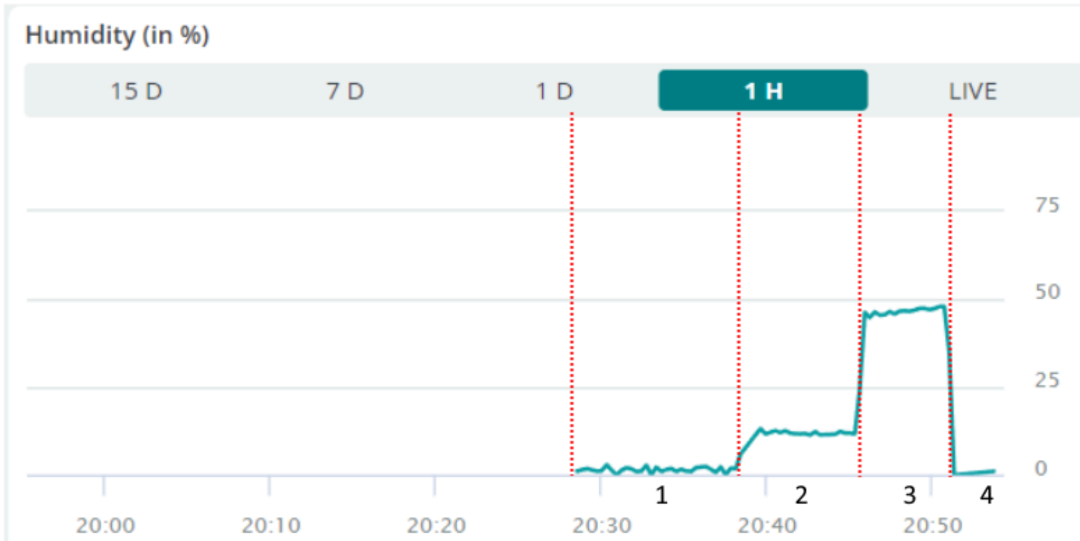

Figure 7.4: humidity sensor results.

Step one of the humidity sensor, as seen in Figure 7.3, is the testing phase of the sensor. The graph shows that it hovers around 0, which is what is the expected humidity in the air. After that, the combination sensor is placed in the soil, after which the humidity spikes up. Even though it is dry soil, it is still a bit humid. In step 3, the water is added, after which the humidity spikes up massively, which is expected. Finally, the sensor is removed from the soil, after which it goes back down to 0, which is what you would expect.

7.2 pH

Testing the pH was done in two steps. First, testing the pH in water. This should result in a pH of around 6.5 to 8.5. The second step involves testing the sensor in vinegar, which should result in a pH between 2 to 3 [31][32]. To do this, a bowl is filled with both water and vinegar, after which the sensor is placed in the bowl, after which a 5-minute timer is set. When the 5 minutes are over, the sensor is removed from the liquid. Figure 7.5 shows the pH chart with water, and figure

7.6 shows the pH chart with vinegar.

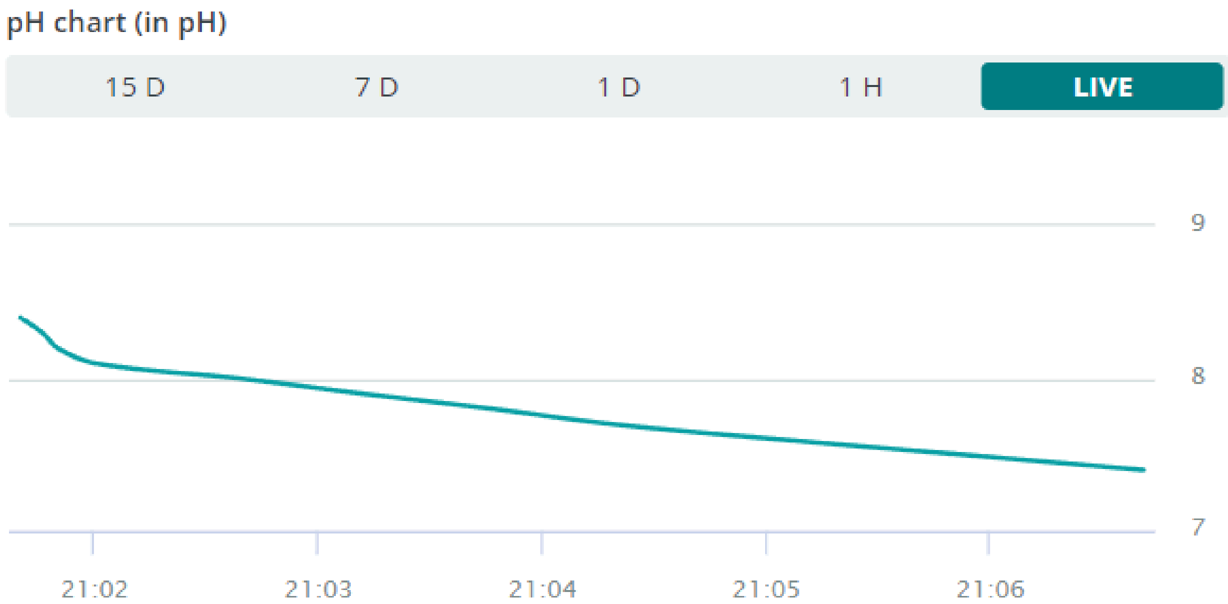

Figure 7.5: pH sensor test with water.

Figure 7.5 shows that the pH hovers between 7.5 and 8.5 but is slightly dropping over time, which is unexpected since you would expect it to remain steady at a single pH because chemically, nothing changes in the water.

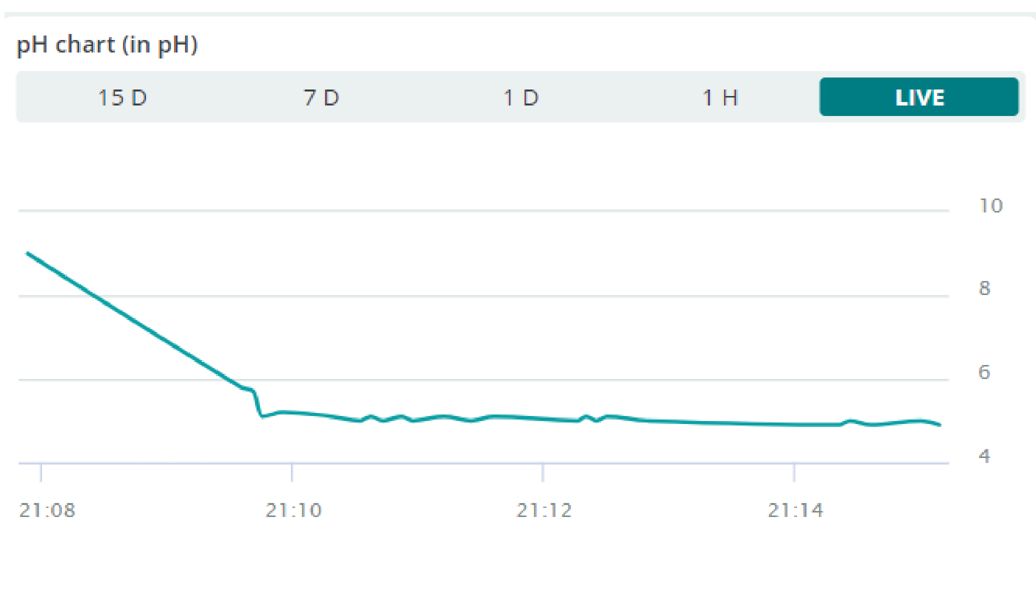

Figure 7.6: pH sensor test with vinegar.

Figure 7.6 shows the pH in vinegar. Between 21:08 and 21:10, the pH is slowly dropping. This is because it was transferred from the water bucket to the vinegar bucket, which made the pH drop to a steady 5 pH.

7.3 CO_2

The testing of the CO_2 sensor is done in two steps. First, a test in the open air to see what the normal ppm levels are in the air. Secondly, a soil test is done with an enclosed space (realized using a glass jar) above the soil where the CO_2 sensor lies, as shown in Figure 7.7. Figure 7.8 shows the results of the CO_2 sensor in the open air. The CO_2 ppm seems to hover around 700 ppm with an average deviation of approximately 100 ppm. This deviation could be due to the inaccuracy of the sensor but is most likely due to my breathing into the sensor, which makes the CO_2 reading spike up for a few minutes and then settle again. The big drop at around 20:50 can not be logically explained but is likely due to a faulty reading sent to the Arduino cloud. Figure 7.9 shows the CO_2 chart in the enclosed area. Here, the average CO_2 ppm lies around 1050 ppm, note that the drop in the end happens when I remove the jar from the soil. After this is done, the CO_2 drops almost instantly to around 800 ppm, which is the level of the air CO_2. The two big spikes at around 12:50 can also not be explained logically but are likely due to a faulty reading.

Figure 7.7: CO_2 *sensor test enclosed in a glass jar on top of soil.*

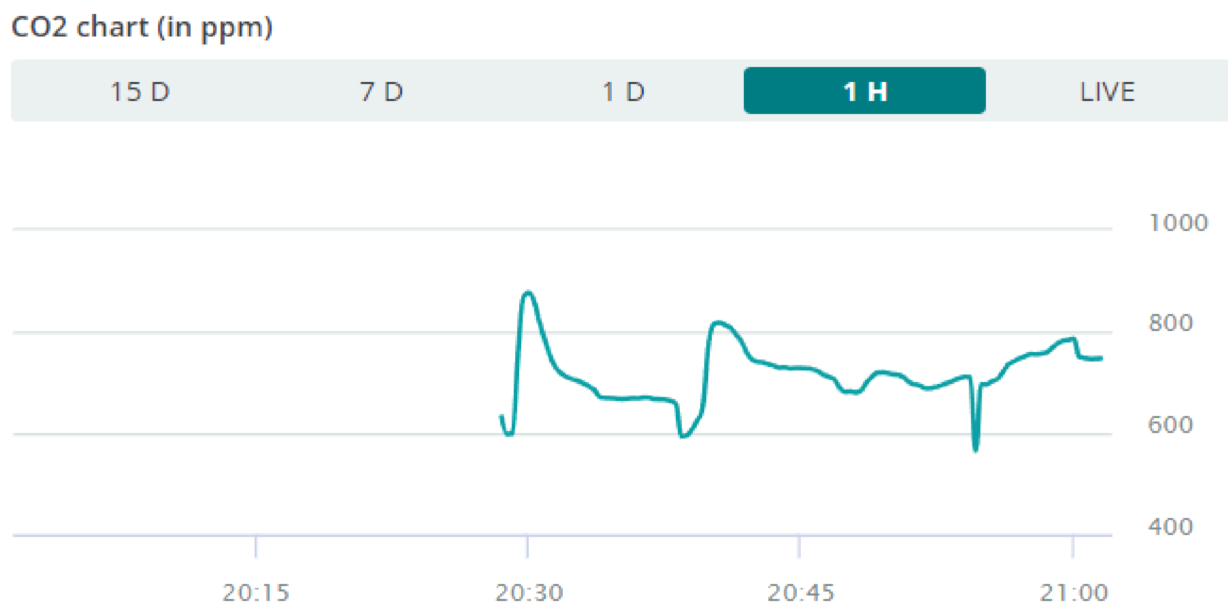

Figure 7.8: CO_2 *sensor test in the open air.*

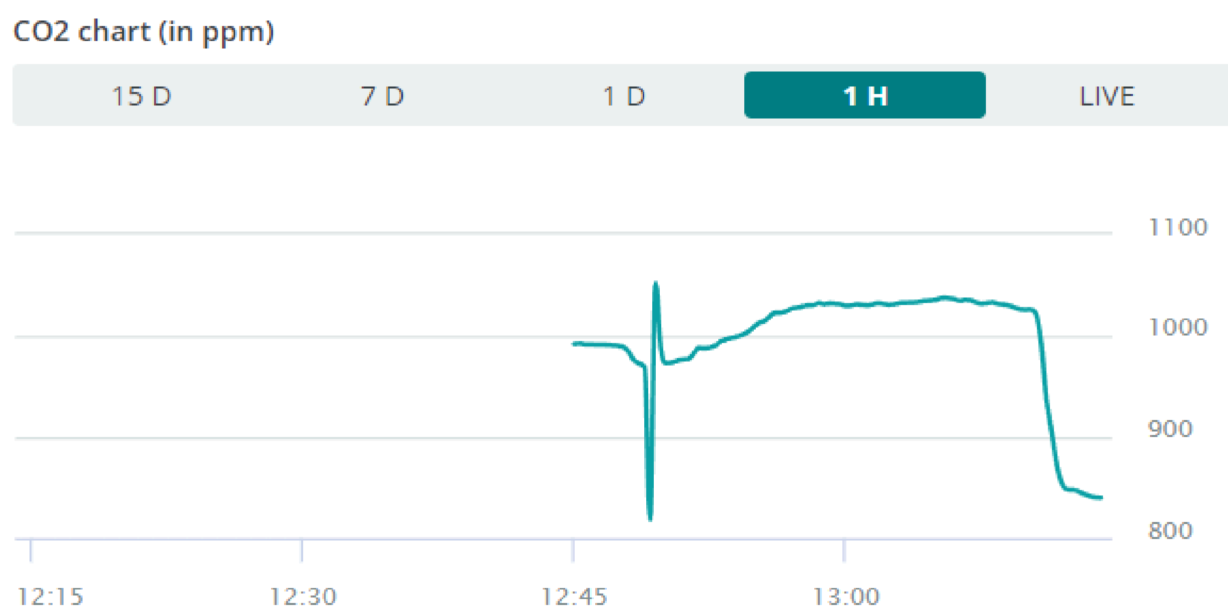

Figure 7.9: CO_2 *sensor test in the enclosed area.*

7.4 Vibration

To test the vibration sensor, all that was done was shaking the sensor itself and knocking on the table around the sensor. Figure 7.10 shows that the vibrations seem to hover around 350 with an average deviation of approximately 10. The vibration seems to hover around this number, no matter if it is shaken or left still. This means the sensor is likely faulty since the code and connection of the sensor are correct.

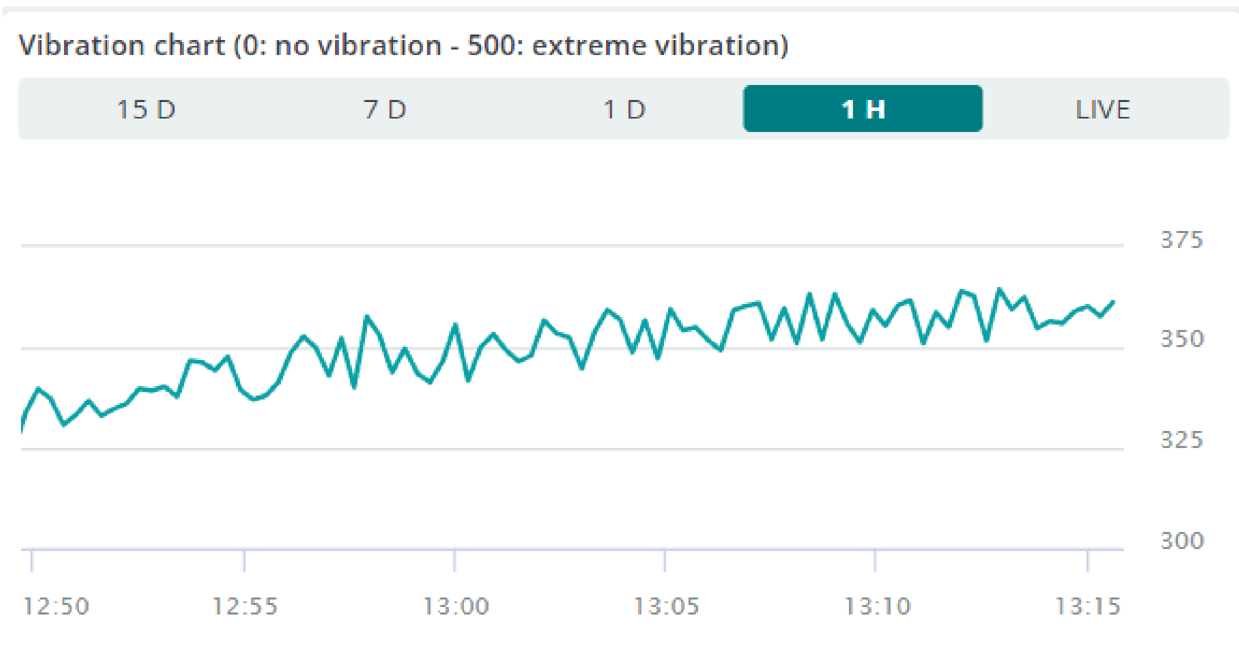

Figure 7.10: vibration sensor test results.

Chapter 8 User Evaluation

The user evaluation is an important step in the creative design process. It is used to evaluate the user's opinion on the current state of the design in order to find out possible points of improvement and parts that can remain unchanged. The evaluation of this research was done in the following way:

1. Tell the tester background information on the project so they know the context of the product that is designed.
2. Show them how the system operates on the hardware level, but also show them the user dashboard, where the data is displayed in graphs.
3. Ask the user several questions about the system (see Appendix G).

These questions are meant to gather feedback on the system's quality, runtime when used on a construction site, and sensor system coverage.

The following questions were asked to the interviewees:

- How would you prefer to get the sensor feedback?
- What area should ideally be covered by one system?
- What would be an acceptable running time before the batteries have to be replaced (recharged)
- Do you miss certain measurements? If so, what would they be?
- Do you think the table interface is user-friendly?
- Would such a system be helpful at a construction site in improving sustainability?
- What is your overall opinion of the system?

Appendix G contains the full anonymized answers from the user testers. To get the full picture of the user evaluation, all the individual questions will be summarized together to form the general consensus of each question. If massive differences between the answers exist then this will be explained in the summary.

How would you prefer to get the sensor feedback?

The user tester's evaluation shows that the users generally appreciate the current visual dashboard with its graphs and live sensor data. They like that it allows for data export for use in other applications. Specific features that stand out include the historical graphs and real-time feedback. Suggestions for improvement include adding more relevant statistics, creating a phone app, displaying sensor values numerically and on a scale with desired and undesired ranges, and providing daily reports in both table and graph formats.

What area should ideally be covered by one system?

User testers emphasize the importance of having multiple sensors on the construction site to ensure accurate and valid data collection. Using a single sensor is probably insufficient, especially for larger construction sites. Users say that the sensor system should at least cover a 20-square-meter area. The number of sensors used should be proportional to the site's size. It is recommended that there be at least three systems, potentially ten, for very large areas. However, this amount depends on the system's overall cost.

What would be an acceptable running time before the batteries have to be replaced (recharged)?

The user testers state that the system should have a minimal running time of at least a week, with a month being the ideal goal. This duration allows for practical battery replacement schedules, such as at the beginning or end of the week, and this provides ample operational time without needing frequent maintenance or interference. Solar cells or other power-generating methods are suggested to extend the system's running time indefinitely.

Do you miss certain measurements? If so, what would they be?

The user testers stated that the following measurements could be included: Incorporating GPS/location tracking for precise data collection measuring particulate matter (PM) and chemical substances in the soil. Additionally, estimating soil composition (such as the proportions of sand and clay) and detecting chemical contamination in the ground are recommended. They also propose adding a tamper-detection sensor to prevent interference with the system. Finally, expanding them to include noise and radiation levels could provide even more comprehensive data.

Do you think the table interface is user-friendly?

User testers appreciate the interface for its user-friendly design and aesthetic appeal. However, they suggest several improvements, like adding more explanations to clarify the meaning of the different widgets/charts and data displayed. They question the need for the initial gauges and emphasize the importance of historical graphs for their use case. They request additional information to help less experienced users understand the data better. Finally, they want more context for the data to improve readability.

Would such a system be helpful at a construction site in improving sustainability?

User testers have varied perspectives on the system's impact on improving sustainability on construction sites. They say that it could offer insights into the sustainability of construction sites, although some are unsure about its immediate benefits towards this goal. Overall, the system is seen as vital for monitoring construction site sustainability and validating governmental regulations. Important to note is that no one thinks that it is useless towards the goal.

What is your overall opinion of the system?

Users appreciate the system's current state, saying that its simple format seems useful. They suggest making it smaller to make it easier to use on the construction site, and the system needs to be able to withstand heavy equipment used on construction sites and the sometimes difficult environmental conditions. Users mention that this system could benefit from integrating into a broader environmental monitoring system, including air measurements. They view it as a promising start but stress the need for ongoing development, robustness, and integration with trends and historical data for comprehensive analysis.

Chapter 9 Discussion & Future Work

This chapter discusses the design and implementation of the sensor system, including hardware, software, and integration details. Key challenges and solutions are highlighted, along with user tester evaluations. Future work is addressed, covering potential improvements and research directions.

9.1 Discussion

This project only focuses on a small part of the sustainability of construction sites. Namely, how to monitor some of the factors contributing to soil pollution's contribution to air pollution on construction sites. Of course, there are many other factors that contribute to pollution and the overall sustainability of construction sites.

Overall, the system functions as expected. Initially, the selected Wi-Fi board could not connect with the cloud system, so a different computer board with built-in Wi-Fi was used. The initially selected vibration sensor was not suitable, so a different one was chosen that was expected to give better results.

The initial results of the prototype sensor system show that most of the sensors measure the data correctly. Only the vibration sensor had an issue and kept on giving wrong results. It needs to be replaced. In addition, it needs to be seen if the data from a correct working vibration sensor is usable or if another type of sensor needs to be used to measure the vibration.

During testing, some of the sensors occasionally showed strange behavior that led to inexplicable results, like a sudden spike. If less obvious results are seen, it is difficult to trust the data, as it can also have been a technical problem. The reasons for the strange data could be due to many reasons. The sensor could be broken, resulting in giving strange data and word peaks and lows, although this is likely only applicable for the vibration sensor. The most likely reason for the strange behavior of the sensor is that the sensors measured a wrong value at that moment but still sent this data point to the Arduino cloud. This can be fixed by implementing an algorithm that discards a value that is likely wrong.

Looking at the results in context with the research question: What methods can be implemented to ensure each sensor accurately measures the targeted data? Most of the sensors measure the expected values. After ensuring that the sensor data is correct and does not contain the earlier mentioned strange behavior, further improvements can be made by using higher-quality sensors and finding a way to calibrate the sensor automatically in the software to make the result even more accurate. A good example is the pH sensor, where the used sensor is less capable in very wet conditions, which is not uncommon in the Netherlands.

User evaluation was done with a low number of more technical users but without real experience in the construction industry.

The user group thinks that the system in its current state seems useful for improving sustainability on construction sites. However, there are many things that could be improved. But, this research's overall impact is probably marginal compared to the big climate goals.

Users also state that some improvements can be made by adding more measured indicators and making the sensor dashboard more intuitive to use with features that add more context to the data. Although they state that the interface is user-friendly.

Lastly, the potential for integrating the sensor system on construction sites in its current state is minimal. The pros of the sensor system in its current state are that it measures pollution indicators in the ground. A good-looking interface to view the measured data has been created. The measured data can be sent over the internet to this interface. The data can be compared on this interface based on time, and this could give construction managers or researchers insights into what happens in the ground during construction and possible solutions to increase sustainability. The cons of the system in its current state is that it is not construction site proof, meaning that it will likely be destroyed due to weather such as rain or construction equipment driving around on the site due to a lacking safety box that protects it from heavy construction equipment and vehicles. The system in its current state does not have a battery, meaning it has to be run connected to a computer and it does not have code installed that minimizes power consumption to increase the battery lifespan of the system. The wifi transmitter range has not been tested, but it is likely not far, which would be a problem on a large construction site where the data has to be transmitted over long distances. Finally, comparing the data in the graph interface can be difficult since nothing is put in place to help compare the data except visually representing the data in time graphs.

9.2 Future work

The existing research reveals a notable gap in the availability of soil measurement tools capable of simultaneously assessing all essential indicators. This should be further investigated. Based on the results from the first bench prototype, several steps can be taken to improve the system. This chapter describes the first two steps that need to be finished with the build prototype before real field testing with one or more systems can start.

9.2.1 Step 1: Improving the bench prototype

From the realization and initial testing, there are a few parts of the system that need further improvement:

- First, the vibration sensor module does not correctly measure vibrations. This is due to problems with the vibration sensor on the module itself. After replacing the sensor module with a correctly working one, it needs to be evaluated. If the sensor is working fine, but the results are not accurate enough, it could be replaced with an accelerometer with some additional software.
- As discussed in Chapter 2.2, the CO_2 sensor should be mounted in a PVC pipe to capture soil CO_2 accurately.
- Some sensors occasionally show strange and incorrect results. This needs to be fixed.
- After all the sensors are implemented and working correctly, they must be calibrated.

9.2.2 Step 2: Building the prototype for field testing

When the bench prototype works well, the next step is to build the first field-testing prototype. This means the system needs a suitable power source, like a battery. Additionally, the electronics must be mounted in a suitable enclosure meeting the requirements. This enclosure must be construction site-proof, meaning it can withstand rough usage, heavy equipment and tough environmental conditions.

9.2.2 Additional Future Work

The literature review found that more fundamental research needs to be done to improve the current pollution indicators and find better methods of measuring these specific indicators. Moreover, more research can be done to determine whether N2O might still be worth measuring in the soil.

Besides this, testing on actual construction sites has to be considered when further researching this subject, as this can determine whether the sensor system is worth implementing on construction sites.

Chapter 10 Conclusion

This chapter will conclude whether the project's goals have been reached and whether the research questions stated in chapter 1.3 can be answered.

In order to help improve sustainability on construction sites, this research aims to find out how to develop an intelligent multi-sensor system for real-time monitoring and analysis of soil pollution's contribution to air pollution on construction sites. This system aims to help construction managers and researchers determine if there are significant emissions in the soil and if there is a relationship between these emissions and other ground indicators such as conductivity or humidity. To help answer this question, the following main research question was formulated:

How can an intelligent system with multiple interconnected sensors be developed to simultaneously measure and compare soil pollution attributes on construction sites, ensuring accurate and effective monitoring of environmental pollution indicators?

To help answer the main research question, these sub-questions are added:

- *What are the most significant types of pollutants found on construction sites?*
- *Which soil pollution indicators can be reliably measured using ground-based sensors?*
- *What methods can be implemented to ensure each sensor accurately measures the targeted data?*
- *How can sensor data be effectively integrated and analyzed to provide meaningful insights into soil pollution levels?*

The system is designed to measure pollution indicators in the soil to help monitor the overall sustainability of construction sites and to make it easier for construction managers to plan out the construction practices beforehand with the help of new data.

To answer the main research question, first the most significant types of soil pollutants found on construction sites has to be established. It is established that there are five types of soil pollutants found on construction sites. Noise pollution caused by heavy equipment, dust pollution released during construction practices; water pollution, which is the contamination of water in the area of a construction site; land pollution, which is the contamination of land and soil on construction sites; and air pollution which is the contamination of the air surrounding the construction site. This research focuses on monitoring soil pollution's contribution to air pollution on construction sites.

In order to link soil pollution further to improving sustainability practices on construction sites, soil pollution indicators are used to identify what happens in the soil during construction. The following pollution indicators were found to be measurable in the soil:

- CO_2 can be reliably measured in real-time in the soil using an airtight container in which a soil CO_2 sensor is installed. Only when the CO_2 rises from the soil will the soil CO_2 be measured.
- Conductivity helps to identify soil salinity, pollution, and health. It can be reliably measured in real-time using TDR. This works by sending a pulse through the soil guided by a transmission line and measuring the time it takes to go from the sending probe to the receiving probe.
- Soil vibration is an indicator that can be linked to construction practices above ground; higher vibration can be linked to heavy equipment usage above the soil. Vibrations can be reliably measured in real-time using an accelerometer, which measures vibrations using acceleration.
- Temperature can be reliably measured in real-time using thermoelectric contact sensors; however, this sensor has some big limitations since it is prone to errors due to radiative, advective, and conductive heating or cooling, all of which can induce measurement errors.
- pH can be measured in real-time using electronic pH sensors; however, some of these sensors need a wet environment to work properly. Humidity can be reliably measured in real-time and works the same as conductivity using a TDR.
- N_2O can be reliably measured in the soil in real-time using the same method as CO_2; however, N_2O is not abundant in the ground. All these indicators except N_2O are measured in the system and should be compared to see what and how pollution is spread on construction sites.

Calibration is paramount to ensure that all the sensors accurately measure the desired indicator. Good sensor calibration ensures that the measured data is correct and reliable.

Using a sensor cloud to display the measured sensor data in graphs is an effective way to read and analyze data over a specific period of time. Graphs that have the same timespan can make analysis and comparison between different indicators more straightforward. This results in proper analysis of the sensor data and, therefore, meaningful insights for the construction managers to improve sustainability on construction sites. Moreover, the research emphasizes the need for future research on construction sites with an improved construction-proof sensor system to ensure that the sensor system has a meaningful place on the site.

Creating an intelligent multi-sensor system for real-time monitoring and analysis of soil pollution on construction sites is feasible. It can help track changes in the soil and relate this to other pollution indicators from other systems. Combining all the sensors into one dashboard and showcasing it with history graphs allows easy comparison between the various indicators. According to the user evaluation, this should help construction managers and researchers with monitoring to improve sustainability.

Appendix A: CO_2 sensor Arduino code

```
/*
  Reading CO2, humidity and temperature from the SCD4x
  By: Paul Clark
  Based on earlier code by: Nathan Seidle
  SparkFun Electronics
  Date: June 3rd, 2021
  License: MIT. See license file for more information but you can
  basically do whatever you want with this code.

  Feel like supporting open source hardware?
  Buy a board from SparkFun! https://www.sparkfun.com/products/18365

  This example prints the current CO2 level, relative humidity, and
temperature in C.
  Edited to only show CO2.

  Hardware Connections:
  Attach RedBoard to computer using a USB cable.
  Connect SCD40/41 to RedBoard using Qwiic cable.
  Open Serial Monitor at 115200 baud.
*/

//make serial monitor 115200 baud

#include <Wire.h>

#include "SparkFun_SCD4x_Arduino_Library.h"
SCD4x mySensor;

void setup() {
  Serial.begin(115200);
  Serial.println(F("SCD4x Example"));
  Wire.begin();
```

```
  //mySensor.enableDebugging(); // Uncomment this line to get helpful
debug messages on Serial
  //.begin will start periodic measurements for us (see the later examples
for details on how to override this)
  if (mySensor.begin() == false) {
    Serial.println(F("Sensor not detected. Please check wiring.
Freezing..."));
    while (1)
      ;
  }
  //The SCD4x has data ready every five seconds
}

void loop() {
  if (mySensor.readMeasurement())  // readMeasurement will return true
when fresh data is available
  {
    Serial.println();
    Serial.print(F("CO2(ppm):"));
    Serial.print(mySensor.getCO2());
  } else
    Serial.print(F("."));
  delay(500);
}
```

Appendix B: combination sensor Arduino code

```
#include <SoftwareSerial.h>
/* ModbusMaster For communication with RS485+MODBUS sensors */
#include <ModbusMaster.h>

#define RE 6
#define DE 7
```

```
SoftwareSerial Serial1(8, 9);

/* Setup the Modbus transmission basics */

void PreTransmission() {
  // Set transmit mode
  digitalWrite(RE, HIGH);
  digitalWrite(DE, HIGH);
}

void PostTransmission() {
  // Set receive mode
  digitalWrite(RE, LOW);
  digitalWrite(DE, LOW);
}

// instantiate ModbusMaster object
ModbusMaster node;

/* END - Setup the Modbus transmission basics */
void setup() {
  pinMode(RE, OUTPUT);
  pinMode(DE, OUTPUT);
  digitalWrite(RE, HIGH);  // Set transmit mode
  Serial.begin(115200);
  Serial1.begin(4800);
  node.begin(1, Serial1);
  node.preTransmission(PreTransmission);
  node.postTransmission(PostTransmission);
}

void print_rs485_modbus_npk(float temperature, float sm, int ec, float ph,
int nitrogen, int phosphorus, int potassium, int salinity, int tds) {

  //temp 34.0, sm 80.2, ph 8.4, N:2, P:51, K: 44107, salinity98, tds:
1539.0
```

```
  Serial.print(temperature, 1);
  Serial.print(',');
  Serial.print(sm, 1);
  Serial.print(',');
  Serial.print(ph, 1);
  Serial.print(',');
  Serial.print(nitrogen);
  Serial.print(',');
  Serial.print(phosphorus);
  Serial.print(',');
  Serial.print(potassium);
  Serial.print(',');
  Serial.print(salinity);
  Serial.print(',');
  Serial.print(tds);

  Serial.print("\n");
}

void full_print_rs485_modbus_npkphcth(float temperature, float sm, int ec,
float ph, int nitrogen, int phosphorus, int potassium, int salinity, int
tds) {

  Serial.print("Temperature: ");
  Serial.print(temperature, 1);
  Serial.print(" C°");
  Serial.print(", SM: ");
  Serial.print(sm, 1);
  Serial.print(" %");
  Serial.print(", EC: ");
  Serial.print(ec, 1);
  Serial.print(" uS/cm");
  Serial.print(", pH: ");
  Serial.print(ph, 1);
  Serial.println();
```

```
  Serial.print("N: ");
  Serial.print(nitrogen);
  Serial.print(" mg/kg");
  Serial.print(", P: ");
  Serial.print(phosphorus);
  Serial.print(" mg/kg");
  Serial.print(", K: ");
  Serial.print(potassium);
  Serial.print(" mg/kg");
  Serial.println();

  Serial.print("Salinity: ");
  Serial.print(salinity);
  Serial.print(" mg/L");
  Serial.print(", TDS: ");
  Serial.print(tds);
  Serial.print(" mg/L");
  Serial.println();

  Serial.print("\n");
}

void rs485_modbus_npkphcth() {

  uint8_t resultMain;
  float sm = -999.0;
  float temperature = -999.0;
  int ec = -999;
  float ph = -999;
  int nitrogen = -999;
  int phosphorus = -999;
  int potassium = -999;
  int salinity = -999;
  int tds = -999;
```

```
// 0x0000 = Moisture (%) (0-100%) (<4s)
delay(500);
resultMain = node.readInputRegisters(0x0000, 1);
if (resultMain == node.ku8MBSuccess) {
  sm = float(node.getResponseBuffer(0x00)) / 10;
}

// 0x0001 = Temperature (C) (-40-80C°) (<15s)
delay(500);
resultMain = node.readInputRegisters(0x0001, 1);
if (resultMain == node.ku8MBSuccess) {
  temperature = float(node.getResponseBuffer(0x00)) / 10;
}

// 0x0002 = ec (uS/cm) (0-20000 uS/cm) (<1s)
delay(500);
resultMain = node.readInputRegisters(0x0002, 1);
if (resultMain == node.ku8MBSuccess) {
  ec = float(node.getResponseBuffer(0x00)) / 10;
}

// 0x0003 = ph (3-9PH) (<1s)
delay(500);
resultMain = node.readInputRegisters(0x0003, 1);
if (resultMain == node.ku8MBSuccess) {
  ph = float(node.getResponseBuffer(0x00)) / 10;
}



// 0x0004 = nitrogen (mg/kg) (1-2999 mg/kg) (<1s)
delay(500);
resultMain = node.readInputRegisters(0x0004, 1);
if (resultMain == node.ku8MBSuccess) {
  nitrogen = node.getResponseBuffer(0x00);
}
```

```
  // 0x0005 = phosphorus (mg/kg) (1-2999 mg/kg) (<1s)
  delay(500);
  resultMain = node.readInputRegisters(0x0005, 1);
  if (resultMain == node.ku8MBSuccess) {
    phosphorus = node.getResponseBuffer(0x00);
  }

  // 0x0006 = potassium (mg/kg) (1-2999 mg/kg) (<1s)
  delay(500);
  resultMain = node.readInputRegisters(0x0006, 1);
  if (resultMain == node.ku8MBSuccess) {
    potassium = node.getResponseBuffer(0x00);
  }



  // 0x0007 = salinity (ppm)
  delay(500);
  resultMain = node.readInputRegisters(0x0007, 1);
  if (resultMain == node.ku8MBSuccess) {
    salinity = node.getResponseBuffer(0x00);
  }

  // 0x0008 = tds (ppm)
  delay(500);
  resultMain = node.readInputRegisters(0x0008, 1);
  if (resultMain == node.ku8MBSuccess) {
    tds = node.getResponseBuffer(0x00);
  }

  full_print_rs485_modbus_npkphcth(temperature, sm, ec, ph, nitrogen,
phosphorus, potassium, salinity, tds);
}

void loop() {
  Serial1.flush();
  rs485_modbus_npkphcth();
```

```
}
```

Appendix C: 801S vibration sensor code

```
#define analogpin 1
#define digitalpin 2

void setup() {
  Serial.begin(9600); // Increased baud rate for debugging
  pinMode(analogpin, INPUT); // set the OUT signal pin as an input
  pinMode(digitalpin, INPUT); // set the OUT signal pin as an input
}

void loop() {
  vibration = analogRead(analogpin);  // read the voltage level on the A0
  digital = digitalRead(digitalpin);  // read the voltage level on the D2
  Serial.println((String)"Light level: Analog " + vibration + " Digital "
+ digital ); // send the result to the serial monitor
  delay(2000); // pause for a moment before repeating
}
```

Appendix D: Integrated code of sensor system

```
/*
  Reading CO2, humidity and temperature from the SCD4x
  By: Paul Clark
  Based on earlier code by: Nathan Seidle
  SparkFun Electronics
  Date: June 3rd, 2021
  License: MIT. See license file for more information but you can
  basically do whatever you want with this code.

  Feel like supporting open source hardware?
  Buy a board from SparkFun! https://www.sparkfun.com/products/18365

  This example prints the current CO2 level, relative humidity, and
temperature in C.
  Edited to only show CO2.

  Hardware Connections:
  Attach RedBoard to computer using a USB cable.
  Connect SCD40/41 to RedBoard using Qwiic cable.
  Open Serial Monitor at 115200 baud.
*/
/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/fb71ec3a-3955-4b15-9f3d-29956d10222
f

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when
changes are made to the Thing

float co2;
float conductivity;
float humidity;
```

```
float pH;
float temperature;
float vibration;
bool lED;
CloudTime measureInterval;

Variables which are marked as READ/WRITE in the Cloud Thing will also have
functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of this
sketch.
*/

#include "thingProperties.h"
#include <Wire.h>
#include <SparkFun_SCD4x_Arduino_Library.h>
#include <SoftwareSerial.h>
#include <ModbusMaster.h>

#define RE 6
#define DE 7
#define LED1 13
#define analogpin 1
#define digitalpin 2

float vib;
float temp = 0;
float ec = 0;
float ph = 0;
float CO2 = 0;
float sm = 0;
int digital;

// instantiate sensors
ModbusMaster node;
SCD4x mySensor;
```

```
SoftwareSerial rs485Serial(8, 9); // Renamed from Serial1 to rs485Serial

void setup() {
    Serial.begin(9600); // Increased baud rate for debugging
    rs485Serial.begin(4800);
    Serial.println("Setup started");
    pinMode(RE, OUTPUT);
    pinMode(DE, OUTPUT);
    digitalWrite(RE, HIGH);
    digitalWrite(DE, HIGH);

    node.begin(1, rs485Serial);
    node.preTransmission(PreTransmission);
    node.postTransmission(PostTransmission);

    initProperties();
    ArduinoCloud.begin(ArduinoIoTPreferredConnection);

    setDebugMessageLevel(2);
    ArduinoCloud.printDebugInfo();

    Wire.begin();
    if (mySensor.begin() == false) {
        Serial.println(F("Sensor not detected. Please check wiring.
Freezing..."));
        while (1);
    }
    pinMode(analogpin, INPUT); // set the OUT signal pin as an input
    pinMode(digitalpin, INPUT); // set the OUT signal pin as an input
    // Serial.println("Setup completed");
}

void loop() {
  Serial.println("In main");
  CO2Sensor();
  combinationSensor();
```

```
  vibrationSensor();
}

void onCo2Change() {
}

void onConductivityChange() {
}

void onPHChange() {
}

void onVibrationChange() {
}

void onTemperatureChange() {
}

void onHumidityChange() {
}

void combinationSensor(){
  Serial.println("started combinationSensor");
  uint8_t resultMain;
  resultMain = node.readInputRegisters(0x0000, 1);
  if (resultMain == node.ku8MBSuccess) {
    sm = float(node.getResponseBuffer(0x00)) / 10;
    Serial.print("humidity reading is: ");
    Serial.println(sm);
    humidity = sm;
    ArduinoCloud.update();
    delay(500);
  }
  // 0x0001 = Temperature (C) (-40-80C°) (<15s)
  resultMain = node.readInputRegisters(0x0001, 1);
  if (resultMain == node.ku8MBSuccess) {
```

```
    temp = float(node.getResponseBuffer(0x00)) / 10;
    Serial.print("Temperature reading is: ");
    Serial.println(temp);
    temperature = temp;
    ArduinoCloud.update();
    delay(500);
  }

  // 0x0002 = ec (uS/cm) (0-20000 uS/cm) (<1s)
  resultMain = node.readInputRegisters(0x0002, 1);
  if (resultMain == node.ku8MBSuccess) {
    ec = float(node.getResponseBuffer(0x00)) / 10;
    Serial.print("conductivity reading is: ");
    Serial.println(ec);
    conductivity = ec;
    ArduinoCloud.update();
    delay(500);
  }

  // 0x0003 = ph (3-9PH) (<1s)
  resultMain = node.readInputRegisters(0x0003, 1);
  if (resultMain == node.ku8MBSuccess) {
    ph = float(node.getResponseBuffer(0x00)) / 10;
    Serial.print("pH reading is: ");
    Serial.println(pH);
    pH = ph;
    ArduinoCloud.update();
    delay(500);
  }
}

void CO2Sensor() {
  Serial.println("arrived in CO2");
  if (mySensor.readMeasurement()) { // readMeasurement will return true
when fresh data is available
    co2 = mySensor.getCO2();
```

```
    if (co2 != 0) {
      ArduinoCloud.update();
      Serial.print("Sending CO2: ");
      Serial.println(co2);
    }
  } else
    // Serial.print(F("#"));
  delay(500);
}

void vibrationSensor() {
  vibration = analogRead(analogpin);  // read the voltage level on the A0
  digital = digitalRead(digitalpin);  // read the voltage level on the D2
  Serial.println((String)"Light level: Analog " + vibration + " Digital "
+ digital ); // send the result to the serial monitor
  delay(2000); // pause for a moment before repeating
}

void onLEDChange() {
  if (lED == 1) {
    digitalWrite(LED1, HIGH);
  } else {
    digitalWrite(LED1, LOW);
  }
}

void PreTransmission() {
  // Set transmit mode
  digitalWrite(RE, HIGH);
  digitalWrite(DE, HIGH);
}

void PostTransmission() {
  // Set receive mode
  digitalWrite(RE, LOW);
  digitalWrite(DE, LOW);
```

```
}
```

Appendix E: SW-420 vibration sensor code

```
const int vsPin =5;
const int LED = 4;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(vsPin, INPUT);
  pinMode(LED, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  bool value = digitalRead(vsPin);
  if (value == 1) {
    digitalWrite(LED,HIGH);
  }else if(value == 0){
    digitalWrite(LED,LOW);
  }
}
```

Appendix F: Wifi connection code

```
#include <SoftwareSerial.h>
#include <stdlib.h>

SoftwareSerial ESP8266(2, 3);  // RX, TX
unsigned char check_connection = 0;
unsigned char times_check = 0;
```

```
void setup() {
  Serial.begin(115200);
  ESP8266.begin(115200);

  ESP8266.print("***VER:");
  delay(2000);
  ESP8266.println("AT+RST");
  delay(1000);
  ESP8266.println("AT+GMR");
  delay(1000);
  ESP8266.println("AT+CWMODE=3");
  delay(1000);
  ESP8266.println("AT+CWLAP");
  delay(1000);
}


void loop() {
  Serial.println("Connecting to Wifi");
  while (check_connection == 0) {
    Serial.print(".");
    ESP8266.print("AT+CWJAP=\".....\",\"......\"\r\n");
    ESP8266.setTimeout(5000);
    if (ESP8266.find("WIFI CONNECTED\r\n") == 1) {
      Serial.println("WIFI CONNECTED");
      break;
    }
    times_check++;
    if (times_check > 3) {
      times_check = 0;
      Serial.println("Trying to Reconnect..");
    }
  }
  while (1);
}
```

Appendix G: User evaluation questions

Name: Rob

Question: How would you prefer to get the sensor feedback?
Answer: Table and graph, report a daily report.

Question: What area should ideally be covered by one system?
Answer: At least 3, quite limited, and two, you can say a bit more, but then you have a lot of noise and combine it with other data. One cannot provide accurate data.

Question: What would be an acceptable running time before the batteries have to be replaced (recharged)
Answer: At least a week, preferably a month.

Question: Do you miss certain measurements? If so, what would they be?
Answer: GPS/location. PM and perhaps chemical substances

Question: Do you think the dashboard is user-friendly?
Answer: Access to the data is fine, but it lacks some context.

Question: Would such a system be helpful at a construction site in improving sustainability?
Answer: Not directly, but it does contribute to providing insight into the construction site. Indirect potential.

Question: What is your overall opinion of the system?
Answer: Interestingly, they work with heavy equipment. How resistant is the very heavy case to protect it. That's a challenge.
Quite compact. The size of a telephone.
Must be seen in the broader context. Together with trends and historical data.
Adds something like an extra data point.

Name: Pieter Jan

Question: How would you prefer to get the sensor feedback?
Answer: It is definitely good to get real-time feedback in that way.

Question: What area should ideally be covered by one system?
Answer: If you have multiple, you can ensure the data is valid. That would be wise. Check out the effect on locations. Place it on the places where they rest. One is not enough.

Question: What would be an acceptable running time before the batteries have to be replaced (recharged)
Answer: It is a wireless low-power device, so it needs a battery. Maybe add solar power or other types of sensors to generate power. A day is too short. At least one weekend. So, at least three days, but it would be very impractical. Test out how long it lasts. A week is a nice start, but a month is even better.

Question: Do you miss certain measurements? If so, what would they be?
Answer: I don't really know, but maybe moisture could have an impact on the air quality.

Question: Do you think the dashboard is user-friendly?
Answer: Probably yes, but I don't know if it is responsive. Maybe make it so you can scroll into the data. Drag left and right, but it is good that it shows the actual value. People can log in with their accounts and add nodes. Save them to their account. Have a list somewhere with those comments.

Question: Would such a system be helpful at a construction site in improving sustainability?
Answer: Can't say. It's hard to know that. That would come up for the people researching after me.

Question: What is your overall opinion of the system?
Answer: It's a nice start. It is very much still in the early face. Founder of the project. It's good work.

Name: Lucas

Question: How would you prefer to get the sensor feedback?
Answer: I would like the sensor to display the values as numbers and on a scale, with the undesired and desired values displayed on a spectrum.

Question: What area should ideally be covered by one system?
Answer: About 1 square km, maybe add more sensors if they are expected to be inaccurate.

Question: What would be an acceptable running time before the batteries have to be replaced (recharged)
Answer: It should last multiple sessions, like four or something. These devices should not require that much.

Question: Do you miss certain measurements? If so, what would they be?
Answer: Maybe the estimated composition of the soil is this much sand, this much clay, etc.

Question: Do you think the dashboard is user-friendly?
Answer: More info on what the data actually means, but that's for noobs like me.

Question: Would such a system be helpful at a construction site in improving sustainability?
Answer: Yes, I think it will. If you know what the soil is made up of, you can anticipate better the construction, which types of materials, and how much foundation you should use.

Question: What is your overall opinion of the system?
Answer: It looks useful. I don't know how much of a difference the soil type will make on construction, but I think this will be very useful for an expert.

Name: Roelof

Question: How would you prefer to get the sensor feedback?
Answer: I like the current graphs, particularly the history. It is really good that the data can be exported.

Question: What area should ideally be covered by one system?
Answer: A single system should be sufficient for small construction sites. One system should be able to cover 30 square meters, but a larger area would be better.

Question: What would be an acceptable running time before the batteries have to be replaced (recharged)
Answer: One month would be a good duration. Solar cells may be used to extend the period indefinitely.

Question: Do you miss certain measurements? If so, what would they be?
Answer: Maybe also measure chemical contamination in the ground.

Question: Do you think the dashboard is user-friendly?
Answer: Yes. It looks nice. I like the history graphs most.

Question: Would such a system be helpful at a construction site in improving sustainability?
Answer: I think it will help if it is really integrated into the company's care for the environment. If you only measure and create reports, I do not know if systems like this help. Maybe some government rules are needed to ensure that the soil is monitored, and if it is not okay, something needs to be done about it.

Question: What is your overall opinion of the system?
Answer: I like the system and the idea behind it. It can really help make construction sites more environmentally friendly. But it should be part of a bigger system with all kinds of other measurements (like in the air).

Name: Yord

Question: How would you prefer to get the sensor feedback?
Answer: The current dashboard is handy. Maybe a phone app later would be good.

Question: What area should ideally be covered by one system?
Answer: I do not know how big construction sites generally are, but I know some can be really big. Three systems should be able to cover larger sites. Multiple systems also help to spot differences between different locations on the same site.

Question: What would be an acceptable running time before the batteries have to be replaced (recharged)
Answer: I think a month would be an ideal goal.

Question: Do you miss certain measurements? If so, what would they be?
Answer: Measure chemical substances

Question: Do you think the dashboard is user-friendly?
Answer: Yes, but I do not know if the gauges at the start of the page are needed. I think the history will be what counts for your use case.

Question: Would such a system be helpful at a construction site in improving sustainability?
Answer: Yes. I think it will clearly show when something is not right. Maybe an alarm would be useful when something out of the ordinary happens. The sooner you act on something, the cheaper it is to fix it.

Question: What is your overall opinion of the system?
Answer: It looks good. You have done a lot of work in a short amount of time. I know that things can get quite rough on construction sites, so making the system fully proof and able to withstand environmental challenges, like a truck hitting it, can be difficult.

Name: Fionna

Question: How would you prefer to get the sensor feedback?
Answer: I like the dashboard. Exporting the data so it can be used in other applications is good.

Question: What area should ideally be covered by one system?
Answer: I think it will depend a little bit on the system price. If the price comes near the value of a phone, you can have more systems on a site, maybe ten systems for a larger construction site. More systems help discover differences between them.

Question: What would be an acceptable running time before the batteries have to be replaced (recharged)
Answer: If think the minimum running time should be a week. That means somebody can replace the batteries at the end or beginning of the week,

Question: Do you miss certain measurements? If so, what would they be?
Answer: Have a sensor to see if somebody tries to tamper with the system.

Question: Do you think the dashboard is user-friendly?
Answer: Yes, it looks nice. But it might be helpful to add some more explanation of what the different widgets mean.

Question: Would such a system be helpful at a construction site in improving sustainability?
Answer: I do not know. I have the feeling that most companies are only trying to look green, but in reality, they are only trying to make a profit and do not really care about the environment that much.

Question: What is your overall opinion of the system?
Answer: The idea about your system is really nice. It would be great if it could be made much smaller. Like the size of the multisensor you use. In that case, you could bury the system in the ground.

Name: Lucian

Question: How would you prefer to get the sensor feedback?
Answer: I like it as a visual dashboard with graphs and statistics. However, certain statistics relating to the data could be added.

Question: What area should ideally be covered by one system?
Answer: Multiple sensors should be placed at equal distances from each other on a construction site. I would expect them to cover a radius of about 20 meters. Based on this, the total number of systems could be calculated.

Question: What would be an acceptable running time before the batteries have to be replaced (recharged)
Answer: I would expect the running time of such a system to be around a month, which would give ample time for it to run without needing maintenance.

Question: Do you miss certain measurements? If so, what would they be?
Answer: The measurements taken are already quite useful; however, additional measurements such as noise or radiation could be taken.

Question: Do you think the dashboard is user-friendly?
Answer: The interface is displayed user-friendly, which is very easy to understand.

Question: Would such a system be helpful at a construction site in improving sustainability?
Answer: Such a system would be instrumental in monitoring and managing a construction site's sustainability and would help validate government quotas.

Question: What is your overall opinion of the system?
Answer: I think the system is done well. It provides a good number of measurements, and it displays them in a very simple manner, which would allow a construction site manager to make quick decisions based on the data.

References

[1] M. van E. Z. en Klimaat, “National Climate Agreement - The Netherlands - Publicatie - Klimaatakkoord,” www.klimaatakkoord.nl, Jun. 28, 2019. https://www.klimaatakkoord.nl/documenten/publicaties/2019/06/28/national-climate-agreement-the-netherlands

[2] G. Kikwasi, “Causes and Effects of Delays and Disruptions in Construction Projects in Tanzania,” Australasian Journal of Construction Economics and Building - Conference Series, vol. 1, no. 2, p. 52, Feb. 2013, doi: https://doi.org/10.5130/ajceb-cs.v1i2.3166.

[3] B. Sizirici, Y. Fseha, C.-S. Cho, I. Yildiz, and Y.-J. Byon, “A Review of Carbon Footprint Reduction in Construction Industry, from Design to Operation,” Materials, vol. 14, no. 20, p. 6094, Oct. 2021, doi: https://doi.org/10.3390/ma14206094.

[4] M. Camarda, S. De Gregorio, G. Capasso, R. M. R. Di Martino, S. Gurrieri, and V. Prano, “The monitoring of natural soil CO2 emissions: Issues and perspectives,” Earth-Science Reviews, vol. 198, p. 102928, Nov. 2019, doi: https://doi.org/10.1016/j.earscirev.2019.102928.

[5] L. S. Jensen, T. Mueller, K. R. Tate, D. J. Ross, J. Magid, and N. E. Nielsen, “Soil surface CO2 flux as an index of soil respiration in situ: A comparison of two chamber methods,” Soil Biology and Biochemistry, vol. 28, no. 10–11, pp. 1297–1306, Oct. 1996, doi: https://doi.org/10.1016/s0038-0717(96)00136-8.

[6] R. Wang et al., “Measurement of N2, N2O, NO, and CO2Emissions from Soil with the Gas-Flow-Soil-Core Technique,” Environmental Science & Technology, vol. 45, no. 14, pp. 6066–6072, Jun. 2011, doi: https://doi.org/10.1021/es1036578.

[7] J. M. Wraith, D. A. Robinson, S. B. Jones, and D. S. Long, “Spatially characterizing apparent electrical conductivity and water content of surface soils with time domain reflectometry,” Computers and Electronics in Agriculture, vol. 46, no. 1–3, pp. 239–261, Mar. 2005, doi: https://doi.org/10.1016/j.compag.2004.11.009.

[8] D. Corwin, Past, present, and future trends of soil electrical conductivity measurement using geophysical methods. researchgate.net: Handbook of Agricultural Geophysics, 2008. Available: https://www.researchgate.net/publication/43271767_Past_present_and_future_trends_of_soil_electrical_conductivity_measurement_using_geophysical_methods

[9] D. L. Corwin and S. M. Lesch, “Apparent soil electrical conductivity measurements in agriculture,” Computers and Electronics in Agriculture, vol. 46, no. 1–3, pp. 11–43, Mar. 2005, doi: https://doi.org/10.1016/j.compag.2004.10.005.

[10] M. Varanis, A. Silva, A. Mereles, and R. Pederiva, “MEMS accelerometers for mechanical vibrations analysis: a comprehensive review with applications,” Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 40, no. 11, Oct. 2018, doi: https://doi.org/10.1007/s40430-018-1445-5.

[11] S. Gupta, W. F. Liu, G. Degrande, G. Lombaert, and W. N. Liu, “Prediction of vibrations induced by underground railway traffic in Beijing,” Journal of Sound and Vibration, vol. 310, no. 3, pp. 608–630, Feb. 2008, doi: https://doi.org/10.1016/j.jsv.2007.07.016.

[12] J. M. Ham and R. S. Senock, “On the Measurement of Soil Surface Temperature,” Soil Science Society of America Journal, vol. 56, no. 2, pp. 370–377, Mar. 1992, doi: https://doi.org/10.2136/sssaj1992.03615995005600020006x.

[13] G. W. Thomas, “Soil pH and Soil Acidity,” SSSA Book Series, pp. 475–490, Sep. 2018, doi: https://doi.org/10.2136/sssabookser5.3.c16.

[14] P. Bhadani and V. Vashisht, “Soil Moisture, Temperature and Humidity Measurement Using Arduino,” IEEE Xplore, 2019. https://ieeexplore.ieee.org/abstract/document/8776973/ (accessed May 05, 2021).

[15] H. Clayton, J. R. M. Arah, and K. A. Smith, “Measurement of nitrous oxide emissions from fertilized grassland using closed chambers,” Journal of Geophysical Research, vol. 99, no. D8, p. 16599, 1994, doi: https://doi.org/10.1029/94jd00218.

[16] S. Zhang, Q. Fan, R. Xia, and T. J. Meyer, “CO2 Reduction: From Homogeneous to Heterogeneous Electrocatalysis,” Accounts of Chemical Research, vol. 53, no. 1, pp. 255–264, Jan. 2020, doi: https://doi.org/10.1021/acs.accounts.9b00496.

[17] M. F. Dilekoglu and E. Sakin, “EFFECT OF TEMPERATURE AND HUMIDITY IN SOIL CARBON DIOXIDE EMISSION. | JAPS: Journal of Animal & Plant Sciences | EBSCOhost,” openurl.ebsco.com, Oct. 01, 2017.
https://openurl.ebsco.com/EPDB%3Agcd%3A15%3A21917343/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A126243243&crl=c

[18] “Arable Mark 3 and Arable Vision,” Arable. https://www.arable.com/mark3/

[19] “Sentek Technologies - Home,” Sentek Technologies, Apr. 12, 2023. https://sentektechnologies.com/

[20] “METER Group,” metergroup.com, Nov. 16, 2022. https://metergroup.com/

[21]“Try Our Easy to Use Geotechnical Vibration Sensors | RST Instruments,” RST Instruments Ltd. https://rstinstruments.com/product-category/instruments/vibration-overpressure/ (accessed Apr. 20, 2024).

[22]“Soil Gas Flux Systems | LI-COR Environmental,” www.licor.com. https://www.licor.com/env/products/soil-flux/ (accessed Apr. 20, 2024).

[23]“FieldScout Wireless ISFET pH Probes | Spectrum Technologies,” www.specmeters.com. https://www.specmeters.com/store/fieldscout-wireless-isfet-ph-probes/?F_Sort=2 (accessed Apr. 20, 2024).

[24] A. Mader and W. Eggink, A DESIGN PROCESS FOR CREATIVE TECHNOLOGY, vol. 16. University of Twente; Enschede (the Netherlands), 2014. Available: https://www.researchgate.net/publication/265755092_A_DESIGN_PROCESS_FOR_CREATIVE_TECHNOLOGY

[25] J. M. Bryson, “What to do when Stakeholders matter,” Public Management Review, vol. 6, no. 1, pp. 21–53, 2004, doi: https://doi.org/10.1080/14719030410001675722.

[26] C. Eden and F. Ackermann, Making Strategy. London: SAGE Publications, 2013. https://books.google.nl/books?hl=en&lr=&id=UnYWfXigU4UC&oi=fnd&pg=PP1&dq=Making+Strategy:+The+Journey+of+Strategic+Management&ots=FFw98yw2Yx&sig=pMPuErWg2XcpTjEp879tMhLR7bI&redir_esc=y#v=onepage&q&f=false

[27] T. Kravchenko, T. Bogdanova, and T. Shevgunov, “Ranking Requirements Using MoSCoW Methodology in Practice,” Cybernetics Perspectives in Systems, vol. 503, pp. 188–199, 2022, doi: https://doi.org/10.1007/978-3-031-09073-8_18.

[28] “Compost & Soil Gas Testing: 3 Inexpensive Solutions,” CO2 Meter, Mar. 10, 2014. https://www.co2meter.com/blogs/news/12783181-compost-soil-gas-testing-3-inexpensive-solutions (accessed Apr. 26, 2024).

[29] “Arduino Cloud - Plans,” cloud.arduino.cc. https://cloud.arduino.cc/plans/

[30] ARDUINO, “Arduino Uno Rev3,” Arduino Official Store, 2021. https://store.arduino.cc/products/arduino-uno-rev3

[31] World Health Organization, “Revised background document for development of WHO Guidelines for Drinking-water Quality,” 2007. Available:

https://cdn.who.int/media/docs/default-source/wash-documents/wash-chemicals/ph.pdf?sfvrsn=16b10656_4#:~:text=The%20pH%20of%20most%20drinking

[32] Healthline, “Is Vinegar an Acid or Base? And Does It Matter?,” Healthline, Mar. 07, 2023. https://www.healthline.com/nutrition/vinegar-acid-or-base#:~:text=Vinegar%20is%20very%20acidic%2C%20with

[33] E. Seran, M. Godefroy, E. Pili, N. Michielsen, and S. Bondiguel, “What we can learn from measurements of air electric conductivity in 222 Rn-rich atmosphere,” Earth and Space Science, vol. 4, no. 2, pp. 91–106, Feb. 2017, doi: https://doi.org/10.1002/2016ea000241.

[34] Medline Plus, “Body temperature norms,” medlineplus.gov, 2021. https://medlineplus.gov/ency/article/001982.htm#:~:text=The%20average%20normal%20body%20temperature

[35] “What is the Best Temperature for Drinking Water? | BRITA,” www.brita.co.uk. https://www.brita.co.uk/news-stories/dispenser/what-is-the-best-temperature-for-drinking-water#:~:text=Room%20temperature%20water%20is%20around

[36] Arduino.cc, 2024. https://docs.arduino.cc/hardware/uno-r4-wifi/#tech-specs (accessed Jul. 02, 2024).

[37] Centurylink, “Should I use 2.4GHz or 5GHz WiFi? | CenturyLink,” www.centurylink.com. https://www.centurylink.com/home/help/internet/wireless/which-frequency-should-you-use.html

[38] T. M, “Wireless Communication Protocols: Bluetooth vs. Wi-Fi vs. LoRa,” Mifra Electronics, Nov. 22, 2023. https://mifraelectronics.com/wireless-communication-protocols-bluetooth-vs-wi-fi-vs-lora/ (accessed Jul. 02, 2024).

[39] “Your Next Smart Phone: 5G vs 4G: What is the difference?,” www.tcl.com. https://www.tcl.com/global/en/blog/your-next-smart-phone-5g-vs-4g-what-is-the-difference#:~:text=4G%2C%20the%20fourth%20generation%20of (accessed Jul. 02, 2024).

[40] “What Is Mesh Wifi? Our 2024 Exclusive Guide,” NETGEAR Blog. https://www.netgear.com/hub/technology/what-is-mesh-wifi/#:~:text=more%20per%20satellite- (accessed Jul. 02, 2024).

[41] “Redirecting,” Google.com, 2024. https://www.google.com/url?q=https://www.scientificacademic.com/index.php/tsj/article/download/15/17&sa=D&source=docs&ust=1719927728617654&usg=AOvVaw3pIBKtes0HA75S3HS9uHXZ (accessed Jul. 02, 2024).

[42] M. van A. Zaken, “The nitrogen strategy and the transformation of the rural areas - Nature and biodiversity - Government.nl,” www.government.nl, Jan. 05, 2023. https://www.government.nl/topics/nature-and-biodiversity/the-nitrogen-strategy-and-the-transformation-of-the-rural-areas

[43] United Nations Environment Programme, “Building Materials And The Climate: Constructing A New Future,” UNEP - UN Environment Programme, Sep. 12, 2023. https://www.unep.org/resources/report/building-materials-and-climate-constructing-new-future

[44] “CO2 emissions from buildings and construction hit new high, leaving sector off track to decarbonize by 2050: UN,” UN Environment Programme, Nov. 09, 2022. https://www.unep.org/news-and-stories/press-release/co2-emissions-buildings-and-construction-hit-new-high-leaving-sector

[45] “Press corner,” European Commission - European Commission. https://ec.europa.eu/commission/presscorner/detail/en/ip_24_886

[46] A. Singh, “Comparison of closed static and dynamic chamber methods for the measurement of soil CO2 efflux,” Available: https://www.researchgate.net/profile/Arvind_Singh56/post/How_to_use_EGM-3_to_measure_soil_CO2_efflux/attachment/59d63d3079197b807799a0b1/AS%3A418769198436353%401476853765922/download/2.pdf

[47] “PARIS AGREEMENT,” Available: https://unfccc.int/sites/default/files/english_paris_agreement.pdf