INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Air quality monitoring with bicycles using low-cost sensors

Francisco Sousa Nunes

Master in Computer Engineering

Supervisor:
PhD Alexandre Manuel de Castro Passos de Almeida, Assistant Professor
ISCTE - Instituto Universitário de Lisboa

Co-Supervisor:
PhD Pedro Figueiredo Santana, Associate Professor
ISCTE - Instituto Universitário de Lisboa

September, 2024

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

Air quality monitoring with bicycles using low-cost sensors

Francisco Sousa Nunes

Master in Computer Engineering

Supervisor:
PhD Alexandre Manuel de Castro Passos de Almeida, Assistant Professor
ISCTE - Instituto Universitário de Lisboa

Co-Supervisor:
PhD Pedro Figueiredo Santana, Associate Professor
ISCTE - Instituto Universitário de Lisboa

September, 2024

Acknowledgments

I would like to express my heartfelt gratitude to the individuals and organizations whose support has been invaluable during this research and the completion of my thesis.

First and foremost, I am deeply thankful to my thesis supervisor, Professor Alexandre Manuel de Castro Passos de Almeida, and co-supervisor, Professor Pedro Figueiredo Santana, whose expertise, guidance, and patience have been essential to this work. Their insightful advice and constructive feedback have been instrumental in the development of this thesis.

I would also like to extend my sincere thanks to all the participants involved in the usability tests. Their time and feedback were crucial to refining this project. A special thanks to CoolGym Colares for generously allowing me to set up a testing area at their facility, providing a valuable environment for gathering real-world data.

Lastly, I am profoundly grateful to my family for their unwavering support and encouragement throughout this journey. Their constant love, understanding, and belief in me helped me navigate the inevitable challenges along the way.

Resumo

A poluição atmosférica continua a ser uma preocupação crítica que afecta tanto a saúde pública como o ambiente. Esta tese apresenta o desenvolvimento de um sistema autónomo de monitorização da qualidade do ar, de baixo custo, montado em uma bicicleta. O sistema utiliza fontes de energia verde, como painéis solares e dínamos de bicicleta, para alimentar sensores que recolhem dados em tempo real sobre partículas (PM), óxidos de nitrogênio (NOx) e compostos orgânicos voláteis (VOCs). Os dados são transmitidos para uma aplicação móvel via Bluetooth Low Energy (BLE), permitindo aos utilizadores visualizar medições de qualidade do ar sobrepostas num mapa, fornecendo indicacões acionáveis sobre as condições ambientais durante a condução.

A arquitetura do sistema integra um sensor Sensirion SEN55, um microcontrolador ESPDuino-32 e um módulo de energia solar Waveshare, todos alojados numa caixa personalizada. A aplicação Android, denominada "AirFinder", serve como interface do utilizador, exibindo dados do sensor e permitindo registo e análise de caminho. Testes de usabilidade com um grupo diversificado de utilizadores confirmaram a facilidade de utilização do sistema e o seu potencial para aumentar a consciencialização sobre a poluição atmosférica.

Este projeto demonstra a viabilidade de criar uma ferramenta de monitorização da qualidade do ar portátil e energeticamente eficiente, com potencial para uso generalizado em áreas urbanas, aplicações de pesquisa e cenários de emergência onde a infraestrutura de monitorização tradicional não está disponível.

Palavras-chave: Monitorização da qualidade do ar, sensores de baixo custo, energia verde, dados em tempo real, aplicativos móveis, bicicleta

Abstract

Air pollution remains a critical concern affecting both public health and the environment. This dissertation presents the development of a low-cost, autonomous air quality monitoring system mounted on a bicycle. The system utilizes green energy sources, such as solar panels and bicycle dynamos, to power sensors that collect real-time data on particulate matter (PM), nitrogen oxides (NOx), and volatile organic compounds (VOCs). The data is transmitted to a mobile application via Bluetooth Low Energy (BLE), allowing users to visualize air quality measurements overlaid on a map, providing actionable insights into environmental conditions while riding. The system's architecture integrates a Sensirion SEN55 sensor, an ESPDuino-32 microcontroller, and a Waveshare solar power module, all housed in a custom-designed enclosure. The Android app, named "AirFinder," serves as the user interface, displaying sensor data and enabling path recording and analysis. Usability tests with a diverse group of users confirmed the system's ease of use and its potential to raise awareness about air pollution. This project demonstrates the feasibility of creating a portable, energy-efficient air quality monitoring tool, with the potential for widespread use in urban areas, research applications, and emergency scenarios where traditional monitoring infrastructure is unavailable.

Keywords: Air Quality Monitoring, Low-Cost Sensors, Green Energy, Real-Time Data, Mobile Application, Bicycle

Contents

List of Tables

List of Abbreviations and Acronyms

PM – Particulate Matter

NOx – Nitrogen Oxides

VOC – Volatile Organic Compounds

BLE – Bluetooth Low Energy

LCD – Liquid Crystal Display

GPS – Global Positioning System

ESP32 – A microcontroller model used in the system

PWM – Pulse Width Modulation

MPPT – Maximum Power Point Tracking

PCB – Printed Circuit Board

SD – Secure Digital (referring to the card used for data storage)

UI – User Interface

API – Application Programming Interface

R/H – Relative Humidity

I2C – Inter-Integrated Circuit (a communication protocol)

Chapter 1: Introduction

1.1. Motivation

Throughout history, mankind has evolved to meet its ever-growing needs, and in doing so, has profoundly altered the natural environment. This evolution, driven by technological advancements and a quest for progress, has been a double-edged sword—bringing about remarkable achievements and significant challenges. One of the most pressing issues arising from this dynamic is the state of air quality. From the dawn of the Industrial Revolution in the late 18th century, humanity embarked on an era of mass industrialization. Factories sprang up, and with them, the consumption of fossil fuels skyrocketed. This rapid industrialization, while fueling economic growth and societal advancements, also initiated the large-scale release of pollutants into the atmosphere. The burning of coal, oil, and gas for energy and transportation has left the quality of air in many regions in a deplorable state. Today, the situation remains dire. According to work [1], Deaths attributable to ambient PM2.5 increased from 3.5 million in 1990 to 4.2 million in 2015. Exposure to ozone caused an additional 254 000 deaths and a loss of 4.1 million DALYs (Disability-adjusted life years) from chronic obstructive pulmonary disease in 2015. This pervasive issue of air pollution has far-reaching consequences. Air pollution is a major contributor to a host of serious health conditions, work [2] indicates that exposure to air pollutants is associated and may be causal to hypertension, myocardial infarction (MI), atherosclerosis and strokes (non-fatal and fatal). Chronic exposure to polluted air can exacerbate respiratory conditions such as asthma and bronchitis and has been linked to cognitive decline and adverse birth outcomes. Alarmingly, it is not just outdoor air pollution that poses a threat. Indoor air pollution is a significant, yet often overlooked, problem. Around 2.4 billion people globally are exposed to hazardous levels of indoor air pollution. This exposure is often due to the use of traditional biomass fuels (like wood, dung, and crop residues) for cooking and heating in poorly ventilated homes. When combined with outdoor air pollution, the cumulative effect is staggering, contributing to an estimated 7 million premature deaths each year. These deaths are preventable and highlight the urgent need for effective air quality management. The origins of air pollution are predominantly anthropogenic stemming from human activities such as industrial processes, vehicle emissions, and agricultural practices. Natural events like wildfires and volcanic eruptions also contribute to air pollution, but on a less regular basis. Industrial and metropolitan areas bear the brunt of this pollution due to dense populations and concentrated industrial activities. Cities in developing countries often suffer from the worst air quality, where rapid urbanization and lack of stringent environmental regulations exacerbate the problem. Given the severe implications for public health and the environment, monitoring and improving air quality have become imperative. Advances in technology have enabled more sophisticated methods for tracking air pollutants, providing critical data that can inform policies and interventions. Efforts to reduce air pollution include transitioning to renewable energy sources, implementing stricter emissions standards, and enhancing public transportation systems to reduce reliance on fossil fuels. Urban planning initiatives, such as creating green spaces and improving building ventilation, can also play a crucial role in mitigating air pollution. Addressing air quality degradation requires a comprehensive and multifaceted approach. Governments, industries, and individuals all have roles to play in reducing emissions and enhancing air quality. Global cooperation is essential, as air pollution transcends national borders and demands coordinated action. As we move forward, the integration of sustainable practices into our daily lives and industrial processes will be crucial in reversing the damage and ensuring a healthier future for generations to come. The quality of the air we breathe is not just a local issue but a global concern that reflects the broader health of our planet.

1.2. Research Questions

In the interest of contributing to public health policies, a plan was devised to monitor air quality while on a bicycle. For this purpose, certain questions arise that must be tackled in order to have a proper application of this concept:

- Considering power supply, what components can be used, can it be done using green-energies, and are they enough to achieve power autonomy?
- Considering mobility, what can be done to get reliable/accurate in real-time geographical data and values of air quality?
- Considering visualization, how can air quality be mapped and shown in a user-friendly, informative way?
- Considering financial costs, can this device be replicated at a low-cost for the average citizen?

1.3. Goals

The goal of this work is the development of an affordable fully autonomous air quality measurement system installed on a bicycle and powered only by green energies. The system should be autonomous not only on energy but also in information analysis with a user interface runnable in low power mobile devices like phones or tablets. This system would be of particular importance in areas where Internet and/or electricity is not available, like remote regions, regions in a military conflict and after natural disasters like volcano eruptions or tsunamis, for research and public safety purposes.

The system should also be easily upgraded with new sensors and more powerful components. For this it must support common devices bus like i2c, 2-wire, 3-wire, CAN bus or serial port. All the components used in this development should be commonly used components with low-cost but with a satisfactory level of quality.

Chapter 2: State of the Art

2.1. Background Concepts

2.1.1. Air Pollutants

The atmosphere is composed of a multitude of chemicals and particles that can be beneficial or detrimental to humans, indirectly by affecting their surroundings or directly by inhaling them. These compositions can be formed naturally as the world is ever changing, making so chemical reactions are naturally triggered. On the other hand, humanity has forced these reactions through means of production, transformation and consumption of goods, which has become a main point of contention in talks and studies on air pollution. These activities produce and release harmful air molecules into the atmosphere at a rate much higher than those naturally produced and higher than the rates of air recycling [3]. As the number of these harmful molecules are diverse and varying in percentage on the composition of air, organizations like WHO (World Health Organization) have established 5 major air pollutants as they are the most common and harmful to human beings. These air pollutants are Particulate Matter (PM), Ozone (Tropospheric, O_3), Nitrogen Oxides (NO_X) of which Nitrogen Dioxide (NO_2) is the most harmful, Sulfur Oxides (SO_X) of which Sulfur Dioxide (SO_2) is the most harmful, and Carbon Monoxide (CO). In this work, 3 of these pollutants were analyzed, namely PM, O_3 (through index of VOCs in the air) and NO_X. Work [4]in Malaysia gives an idea of an API, Air Pollution Index, separating the value range of the API into ranges which are then categorized, and color labeled for ease of analysis. This work is good to take as a basis however I don’t consider a generalized index of air pollution to be of much help since one air pollutant could be very bad and another very good, thus balancing out the API value. With this in mind, work [5] in Badajoz, Spain was chosen because the region where these tests were made closely resembles Lisbon’s weather conditions and both regions are geographically close being both apart of the Iberian Peninsula; and in this work the previous API value is now separated into various indexes for each air pollutant considered, in this case $PM_{2.5}$, PM_{10}, O_3 and NO_2, which also closely resembles the air pollutants considered in this work. Lastly, work [6] in Stuttgart, Germany despite not having closely resembling weather and geographical conditions, offers good insight to the index and values of air pollutants in a urbanized, central European area at different days of the week and times of day, showing the impact of human interaction on the air pollutant levels. Cross-evaluating all these works has given an estimate of the value range for each index level, and so the index used as indicated in Table 2.1.

Table 2.1 - Chosen Air Pollution Index

Index\Air pollutant (µg/m³)	$PM_{2.5}$	PM_{10}	O_3	NO_2
Good	0-10	0-15	0-40	0-30
Satisfactory	11-20	16-30	41-80	31-75
Moderate	21-30	31-50	81-120	76-110
Inadvisable	31-50	51-75	121-180	111-170
Poor	51-75	76-100	181-240	171-250
Very Poor	>75	>100	>240	>250

2.1.2. Particulate Matter

Particulate matter, also known as particle pollution, is a chemical mixture of various elements, be them solid particles or liquid droplets, in the air. This matter can differ in size, and so they have been categorized in groups:

- Coarse particles below 10 micrometers are labeled as PM_{10} and PM_4 if below 4 micrometers.
- Fine particles below 2.5 micrometers are labeled as $PM_{2.5}$.
- Ultrafine or extremely fine particles below 1 micrometer are labeled as PM_1 and known as the most dangerous.

As PM is a complex mixture of pollutants, it's virtually impossible to form in writing any chemical reaction that would result in the creation of PM. Regardless, we know that certain events lead to the creation of PM. Such events can range from Fossil fuel combustion, like burning coal, oil and/or gas for transportation, electricity generation and heating; Industrial activities such as manufacturing, mining and construction produce large amounts of PM especially when involving heavy machinery, raw materials and combustion of fuels; Agricultural activities like soil tilling using machinery, burning crop residue and livestock management/operations.

The effects of PM are unfortunately only harmful. To humans, PM can penetrate deep into the lungs and enter the bloodstream, causing various respiratory and cardiovascular diseases, lung cancer an even lead to a premature death. The finer the particles, the bigger is the chance of provoking damage in other organs by traversing in the bloodstream. Given the composure of PM, it is only natural that it would also be harmful to the environment, as it can create dense fogs of said matter, harming crop longevity and produce, and acidifying lakes and water streams.

Weather conditions can affect PM readings as air molecule behavior changes in accordance with wind, temperature and humidity.

2.1.3. Nitrogen Oxides

Nitrogen Oxides (NO_X) are a group of highly reactive gases composed of Nitrogen (N) and oxygen molecules (O). Of this group, two oxides stand out as the most common, Nitric oxide (NO) and Nitrogen Dioxide (NO_2).

- Nitric Oxide (NO) is a colorless and odorless gas formed mainly through combustion processes, where a nitrogen and oxygen atom react at high temperatures. It's a highly reactive molecule that plays an important role in the atmosphere, as it is a centerpiece in most chemical reactions that form Ozone and other air pollutants.
- Nitrogen Dioxide (NO_2) on the other hand is reddish-brown gas with a pungent smell, resultant of one nitrogen and two oxygen atoms reacting at high temperatures. It´s heavier than air so it tends to stay close to the ground. Due to this particularity and being a strong oxidizer, it readily reacts to other substances, such as water, forming nitric acid (a main component of acid rain), and in the formation of ground-level ozone.

Nitrogen oxides level can be influenced by weather conditions. They play a major role in the formation, dispersion and concentration of nitrogen oxide gases.

- Sunlight and temperature work in tandem in the formation of nitrogen oxide gases, as sunny days provide both higher temperatures and a higher concentration of solar radiation, both necessary for photochemical reactions responsible for the formation of NO and NO_2 consecutively.
- Wind can affect dispersion or concentration depending on its properties. If the wind is strong and unidirectional, it picks and disperses NO_X over distant and larger areas, reducing concentration. Conversely, a calm wind may lead to the buildup of air pollutants in a localized area, leading to mass concentration of NO_X;
- Precipitation can be tricky as it acts as on the dispersion of air pollutants by “cleaning” them out the air, however in the case of NO_2, the formation of nitric acid may lead to acid rains which can be devastating.

2.1.4. Ozone

There are two types of Ozone in Earth’s atmosphere, stratospheric and tropospheric.

- Stratospheric ozone is a layer of ozone gas present in the stratosphere, a region of the Earth's atmosphere located approximately 10 to 50 kilometers (about 31.07 mi) above the Earth's surface. It forms a thin layer that plays a crucial role in protecting life on Earth from harmful ultraviolet (UV) radiation coming from the Sun. The primary component of the atmospheric ozone is a molecule called ozone (O_3). The formation and destruction of atmospheric ozone occurs through a process called the ozone-oxygen cycle. Briefly speaking about it, O_2 gets dissociated by solar UV into singular oxygen atoms; These atoms then react with oxygen molecules, forming ozone.

- Tropospheric Ozone, also known as ground-level or “bad” ozone, is present on the lowest layer of the Earth’s atmosphere, hence the denomination. This type of Ozone is not beneficial and considered an air pollutant, since it’s primarily formed through complex chemical reactions involving nitrogen oxides (NO_x), Volatile organic compounds (VOCs), and sunlight. The biggest sources of tropospheric Ozone derive from photochemical reactions, which as the name suggests involves sunlight reacting with chemicals of the groups mentioned above; and Combustion processes, mainly fossil burning, that release nitrogen oxides (NO_x) that contribute to the formation of tropospheric ozone.

Tropospheric ozone levels can be influenced by various weather conditions, playing a significant role in the formation, dispersion and concentration of tropospheric ozone.

- Temperature is a major variable in the formation of tropospheric ozone, higher temperatures enhance the photochemical reactions that produce ozone, thus tropospheric ozone levels tend to be higher during hot summer days.
- Sunlight and solar radiation play a conjoined role with temperature, as increased sunlight and increased daylight hours provide more energy for photochemical reactions to occur, and solar radiation, especially ultraviolet radiation (UV), as explained above, is a crucial factor in chemical reactions involved in the formation of ozone molecules.
- Wind can be either helpful or destructive, as strong winds carry and disperse air pollutants like ozone, reducing concentration in a specific area. On the other hand, stagnant or low wind may lead to the accumulation of air pollutants in a localized area.
- Atmospheric Stability is a byproduct of wind as it refers to the vertical movement of air in masses. As referenced above, it may impact positively or negatively, a calm and stratified air doesn’t allow for dispersion of air pollutants. Conversely unstable atmospheric conditions, like convective mixing provoked by temperature inversions or thunderstorms, allow for a great dispersion of air pollutants, leading to lower concentration of ozone.
- Precipitation is an effective agent in reducing ozone, as rain and other types of precipitation carry and “wash-out” air pollutants from the atmosphere. Hence events like heavy rain and rainfalls can temporarily reduce tropospheric ozone levels by “cleaning” the air.

- Geography and Topography can influence formation and dispersion of tropospheric ozone. Mountainous or high sea-level areas may provoke temperature inversion, inhibiting vertical mixing of the atmosphere, acting as a lid or a cap on lower regions. This results in the entrapment and accumulation of air pollutants, like valleys and low-lying regions.

2.1.5 Air quality measurement

To capture and monitor previously mentioned air pollutants, a device is needed consisting of sensors, a microcontroller to manage communications between components and alter their functioning accordingly, a screen to display current real-life values and a power supply system capable of providing a stable amount of energy during long periods of time if needed.

For air monitoring sensors, there are multiple ways to capture and measure air pollutants. The most common methods sensors use can be extracted from work [7] and go as follows:

- **Chemical Reaction-Based Sensors**, can be segmented into 2 subtypes: **Chemiresistive sensors**, that rely on changes in electrical resistance due to chemical reactions when specific gases or substances come into contact with a sensing material, concentration of the target gas being proportional to resistance change; and **Electrochemical sensors** utilizing as the name suggests, electrochemical reactions to measure the concentration of gases, consisting of an electrolyte and electrodes, commonly used to detect gases like CO,CO_2 and NO_2.
- **Optical sensors** are also categorized into 3 subtypes: **Infrared (IR) sensors**, measure the absorption or emission of infrared light by gases; **Ultraviolet (UV) sensors**, detect UV light absorption or emission by gases or particles in the air, often used for ozone monitoring; **Laser-Based sensors** use lasers and light scattering or absorption principles to detect particles or gases with high precision, utilized in particle counters and gas analyzers.
- **Mass Spectrometers** ionize gas molecules and then separate and measure them based on their mass-to-charge ratios. Mass spectrometry is highly sensitive and can identify multiple gases simultaneously.
- **Gravimetric sensors** measure changes in mass or weight due to the absorption of gases or particles onto a sensor surface. The change in mass is used to calculate gas concentrations.
- **Piezoelectric sensors** generate an electrical charge in response to mechanical stress. They can be used to detect airborne particles when the particles impact a piezoelectric surface.
- **Capacitive sensors** measure changes in capacitance, which occur when the spacing between capacitor plates changes due to presence of particles or changes in humidity.
- **Acoustic sensors** use sound waves to measure properties of the air, such as pressure or density.

- **Radio Frequency Identification**, utilizing RFID technology can be used to track and monitor items or assets in the air.
- **Photodetectors** convert light intensity into electrical current, making them suitable for detecting light levels or changes in ambient light.

2.2. Related Work

As air quality measurement is an all-encompassing endeavor, previous work has been conducted on the research and perfectionism of measurement systems capable of surveying the area for its air pollution.

Starting from the mobility, the first option would be aerial surveying, so in works [8] [9] [10] unmanned aerial drones and UAVs are utilized to map out air pollution in a 3D environment, putting into question the effects of altitude on air pollutant concentration. This type of vehicle can operate in areas where ground vehicles are not allowed or do not have physical access. Unfortunately, they require specific pilot skills, have low autonomy, and have their flight restricted at low altitudes where most air pollutants reside. Fossil fuel motor vehicles have good autonomy and can provide electrical power. A DC-DC converter can be used to adapt different vehicle voltages (12V in a car or 24V in a truck or bus). The presence of an exhaust pipe can contaminate the environment and consequently influence the measurements; to minimize this, special care must be taken in the sensor placement in the vehicle. Electric vehicles are a solution to this problem with the requirement of availability of charging stations. Bicycle usage is a very popular green option. On a bicycle, solar panels and dynamos can be attached to provide electric power. Charging batteries from two sources at the same time is possible using a hybrid controller which is complex and expensive. A more simple but effective solution proposal uses a simple relay to switch from the two sources and a simple battery management system.

To measure air quality, it is required to collect the quantity of air pollutants present in the air. For this purpose, various sensors have been developed and used in works. Work [11] has used two gas sensors (namely MQ-7 and MQ-135) to collect CO and NO2 respectively, these being two important air pollutants to monitor. Other substances are also captured (Ammonia, sulfide, benzene steam, LPG, alcohol, propane, hydrogen and methane) which are all considered as VOCs. There is a wide variety of sensors to measure the same air pollutants; work [8] uses a laser sensor to measure PM_{10} and an electrochemical sensor to measure CO, NO and NO_2, work [5] uses two 3-eletrode electrochemical sensors to measure NO_2 and O_3, and an optical particulate sensor to measure PM_{10}

and $PM_{2.5}$. The type of sensor will depend heavily on the conditions set for testing as effects of wind pressure, for example, can drastically affect the performance of certain sensor types.

To utilize all these air quality measurement sensors, the usage of a dedicated development board is in most cases obligatory since they are small, versatile and enough for the workload necessary. There are two types of CPU that can be installed on a development board, microprocessors and microcontrollers, that differ on the role they fill for the integrated system, explained in depth in work [12]. Work [13] uses a microprocessor development board, this being in the form of a Raspberry Pi 3B+, to gather atmospheric data and geo spatial data through I2C connection and $PM_{2.5}$ and PM_{10} through SPI connection. This model has a BCM2837B0 CPU which can send the raw data via Wi-Fi internet links or 3G/4G to a database on the Internet. On the other hand, work [4] utilizes an Arduino Uno microcontroller development board where all the air pollutant and geo spatial sensors and are connected onto and an LCD Screen for visual tracking of incoming values. The data is recorded directly onto an SD Card since the development board does not have an internal storage for data, and once data collecting stops, all the gathered information is uploaded to a data Centre via Bluetooth interface. There is also the case of hybrid usage of a microprocessor and a microcontroller, work [9] uses an Arduino MEGA 2560 microcontroller to receive raw data from multiple air pollutant sensors and process it, and then via USB to Serial connection, the processed data is sent to a Raspberry Pi 3 microprocessor where the data is transmitted to a mobile application in real time via NB-IoT for a reduced power consumption and increased system capacity and bandwidth efficiency. The result is always achievable with any of these 3 methods, it just depends on the number of I/O ports needed, what communication protocol serves best, the workload/processing amount and most importantly the power consumption.

Every device needs a power supply, since the objective is having a mobile air quality measurement device, the normal method of connecting it to a fixed power socket won't serve this purpose. Work [13] connects the air quality measurement device to a 12V or 24V vehicle car battery with a DC-DC converter and utilizes a super capacitor as a backup while it has stored power from the car battery. This can work but sapping the car battery is not ideal since it's an integral part of the car's functioning and so if its voltage goes below the car's minimum requirement, it won't be able to start on ignition. The super capacitor is a considerable alternative since it's fast charging and in case there is a high fluctuation in power delivered as can be observed in work [14], the super capacitor can provide a smoother power delivery and in case the power is cut off from the system, the capacitor will provide power in a down curve until it has nothing left. Work [5] uses a lithium-ion battery with a capacity of 2750 mAh, providing an autonomy of 8 hours, and a jack connector to enable charging from a 9V, 660 mA power supply. This battery is installed alongside the other components inside of a box, making it independent from the transport used. So, in this work both internal and external charging is possible, taking energy from the 2750 mAh battery and when it runs out, gives the user an option to charge the system to an external power supply. Although it is a great idea, the external charging is very limited as it requires a very specific power supply, a jack connection which isn't very common nowadays. The system should have a long longevity and be self-sufficient while being used.

Green energy can be used on these devices, thus making them self-sufficient. There are several works utilizing green energy producing components in a bicycle. Work [15] uses a photovoltaic monocrystalline solar panel with a maximum power capacity of 100-Watt Peak, connected to a Battery Charging Controller that alternates between Pulse Width Modulation (PWM) and Maximum Power Point Tracking (MPPT) algorithms working at a voltage of 12 V - 48 V, to regulate charging on a 37.5V/12 Ah battery. The objective was to power a brushless DC Motor, turning the bicycle into an electric motor assisted bicycle. By using sensors to measure solar irradiation, voltage and current, they were able to discover which algorithm was better. In the end MPPT had a significant difference to the PWM output power, so it is best for battery charging optimization. Work [16] displays a hybrid model of charging making use of solar energy through a 12V 16W solar panel and a bicycle dynamo with a 12 V voltage rating. By adding a bicycle dynamo, the system can make use of the energy expended by the user on cycling (biomechanical energy), as it gets turned into electrical energy by the dynamo depending on the rotational speed of the wheel. As the motor is powered, it also helps to rotate the wheel, therefore helping the dynamo to produce energy.

After gathering data, it needs to be processed. To maintain a smooth functioning of the device, the microprocessor unit needs to be focused on data gathering and internal communication with all the sensors attached to it, so raw data is usually communicated to an external source more capable of processing data. Work [17] has all the sensors send gathered raw data in the form of JSON and CSV files to a local laptop via serial USB connection and the laptop sends the measurements taken to a server in the cloud in real time or near real time using cellular data and Wi-Fi respectfully. The goal with using real-time data streaming is to have a read on the progress of air quality measurements and advise the operator accordingly if necessary. Sometimes a Wi-Fi or cellular data connection may not be achievable due to bad weather conditions or bad network coverage in the area. In those cases, a local device to device connection may be most appropriate. Work [18] utilizes Bluetooth to transfer data between the microcontroller (via the microprocessor's UART interface) and a personal smartphone. Unfortunately, this work does not specify which Bluetooth technology was used, Legacy or Low Energy, however in closing statements it is referenced that a parallel work is using BLE (Bluetooth Low Energy) and for the future it should be integrated onto this work. This is an important aspect to take into consideration since both are designed differently and are best suited for different cases. Work [5] stores all data into a microSD and sends it by utilizing a GSM/GPRS module that communicates with the microcontroller via a UART connection, allowing a wireless transfer of data to a cloud storage. In case that is not a viable option, a USB input was also included for manual data transfer and device programming. Having a wired connection is important for emergency situations where the device is failing, or no type of wireless connectivity is available.

Chapter 3: The Proposed System

3.1. System Architecture

As previously mentioned, the goal is to measure air quality in real-time and make it mobile by using a bicycle, making this a green air quality measurement station to promote awareness on air pollution and public health. So, to achieve this, a sensor device and a phone app were devised. The proposed system can be seen in Figure 3.1.

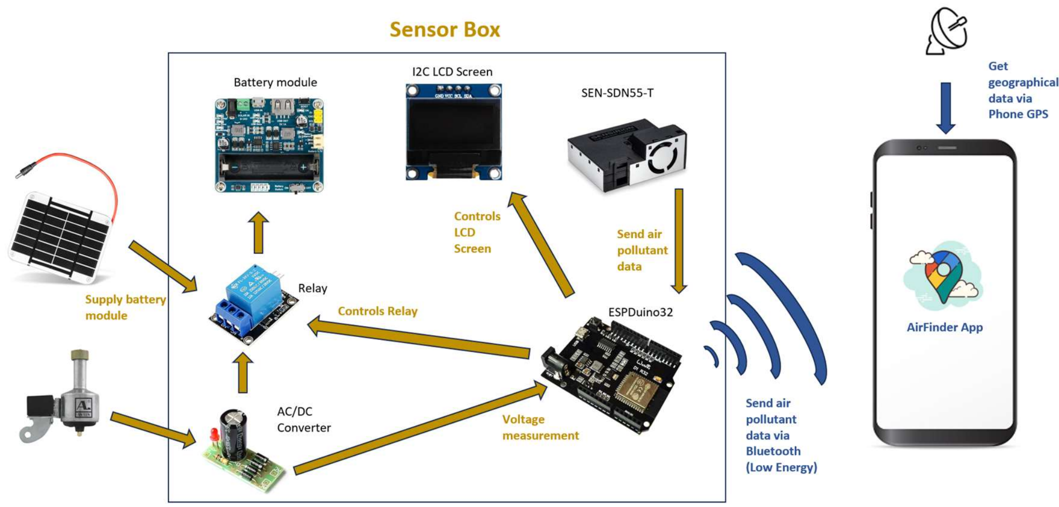


Figure 3.1 - System Arquitecture

As can be seen in this image, the project is split in two parts, hardware focused on data gathering, and software focused on data transformation and display. Let's explore both parts:

- Sensor Box – In the Sensor Box, all the components for air quality measurement are held. An external dynamo and solar panel supply the battery module with energy which subsequently powers the ESPDuino32 microprocessor. The microprocessor handles all types of communication be them internal or external, it must power both the Air quality sensor (SEN-SDN55-T) and the I2C LCD Screen and receive air pollutant data from the SEN-SDN55-T. With this data, the microprocessor sends it to the LCD screen, showing key/value paired data of each air pollutant; and sends it to the user's phone via Bluetooth Low Energy.

- *AirFinder* – The ***AirFinder*** App was developed so that it can work with the data transmitted from the Sensor Box and treat it for visual interpretation. Alongside air pollutant data, the app also requires geographical data taken from the phone's GPS so it can be paired with air pollutant data and be displayed accurately on a world map. The app allows the user to check air pollutant data, record paths and check them later for a graphical analysis of air pollution value changes throughout all the path and an average of every value. It also allows the user to set custom thresholds on every air pollutant so that data evaluation is measured according to the user's preferences.

When addressing mobile communications 4 main methods are available: Wi-Fi, Classic Bluetooth, Bluetooth Low Energy or a serial connection through USB cable. A serial connection would be fine but having a USB cable connected through the bicycle would be a hazard to its normal functioning and there is a high risk of the cable disconnecting. A Wi-Fi connection would be excessive for the task required since all data communications have a size below 1 Mbps, and its very power hungry, especially on the phone. This leaves both Bluetooth protocols left for review, and so as it can be seen in [19], where both protocols are compared, Bluetooth Low Energy comes out on top on important factors such as Total time to send data, being almost 20 times faster (6ms vs 100ms) and Power consumption, requiring at most 50% less power than Classic Bluetooth, and peak current consumption, also requiring at most 50% less current (15mA vs 30mA). Seeing these benefits to battery life and faster communication of data, Bluetooth Low Energy was chosen over the others.

3.2. Materials

3.2.1. Sensirion SEN55-SDN-T

Since this project is based around real time mobile readings, it was opted to use a particle counter laser-based sensor since it can detect finer particles like PM2.5 and ultrafine particles as seen in Figure 3.2, while not being affected by different weather conditions. Particle counter laser-based sensors do immediate readings instead of timed interval readings and are more discrete and protected from environmental hazards and accidents due to their build requirements.

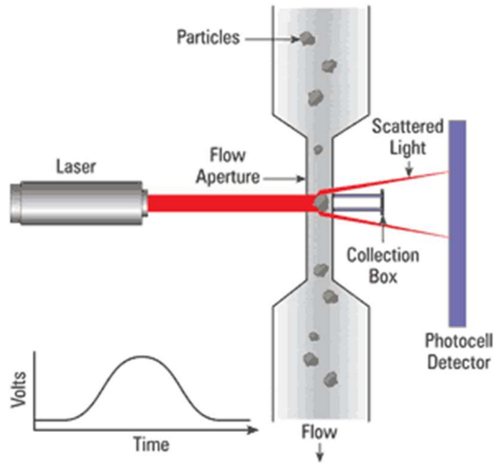


Figure 3.2 - Particle counter functioning

The Sensirion SEN55-SDN-T is an air quality measurement sensor utilizing a laser controller and photo diode for accurate measurement of critical air quality parameters, these being particulate matter, VOC (Resulting in the formation of tropospheric or ground-level Ozone), NO_x, humidity and temperature.

Figure 3.3 - Sensirion SEN 55-SDN-T

It operates on 5 V of supply voltage, requires a 6 pin JST Female cable for supply and communication, and it sends data via an I2C connection. The pinout can be seen in Figure 3.4 and described in Table 3.1.

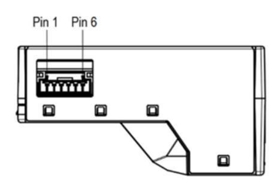


Figure 3.4 - SEN55-SDN-T Pinout order

Table 3.1 - SEN55-SDN-T Pinout description

Pin	Name	Description
1	VDD	Supply Voltage
2	GND	Ground
3	SDA	Serial data input / output
4	SCL	Serial Clock input
5	SEL	Interface Select
6	NC	Do not connect

3.2.2. SSD1306 128x64 Dot Matrix LCD

The SSD1306 LCD is widely known throughout the IoT community as it offers a small yet convenient information screen to display all kinds of information. This LCD comes with either 4 or 7 pins, but the functioning remains the same. The version used in this project is the 4 pins as seen in Figure 3.5. In this case it was used to display the status of the Bluetooth connection with the user's phone (ON/OFF) and each air quality parameter with its respective value at the time of showcase, rotating between each other in order.

Figure 3.5 - SSD1306 LCD

3.2.3. Waveshare Solar Power Manager Module Rev 2.0

The Waveshare Solar Power Manager Module Rev 2.0 is a module capable of receiving energy through a solar panel of 6~24V and other means via micro-USB at 5V and storing it in a ICR14500 3.7V Battery (as seen in Figure 3.6), which can then be used by other components at 5V 1A either through USB or pins. It has a boot button and a battery switch. The module also includes some LEDs that show:

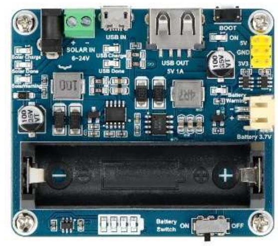


Figure 3.6 - Waveshare Solar Power Manger Module Rev 2.0

- battery level indicator (4 LEDS for 4 levels) and warning
- 2 for micro-USB charging and charged
- 3 for Solar power charging, charged and warning
- Boot LED.

3.2.4. Solar Panel

A simple solar panel capable of providing 6V 2W of energy. A solar panel, also known as a photovoltaic (PV) panel, is a device that converts sunlight into electricity using a technology called photovoltaics. Solar panels can be used to power homes, businesses, and various electrical devices. Solar panels consist of multiple individual photovoltaic cells made from semiconductor materials, typically silicon. These cells are the heart of the panel and are responsible for the conversion of sunlight into electricity. When sunlight hits the solar panel, the photons (particles of light) from the sun interact with the semiconductor material in the photovoltaic cells. This interaction causes electrons in the semiconductor material to become energized and move around. The movement of energized electrons creates an electric current within the solar panel. This electric current is in direct current (DC) form, which is the type of electricity generated by the panels.

Figure 3.7 - 2W Solar Panel

3.2.5. Dynamo

Figure 3.8 - Bicycle Dynamo

A bicycle dynamo (Figure 3.8) is typically attached to the bicycle's wheel close to the rim. As the wheel turns, it also turns the dynamo's internal components. Inside the dynamo, there is a coil of wire and a magnet. As the wheel turns and the magnet rotates within the coil, it creates a changing magnetic field around the coil. According to Faraday's law of electromagnetic induction, this changing magnetic field induces an electrical current to flow through the coil. The electrical current generated in the coil is in the form of alternating current (AC). This AC electricity can be used to power various bicycle accessories, such as lights. However, many modern dynamo systems also include rectifiers that convert the AC electricity into direct current (DC) to make it compatible with most electronic devices. Bicycle dynamos are a convenient and eco-friendly way to generate electricity for lighting and other accessories while cycling. It's important to note that the amount of electricity generated by a dynamo depends on the speed at which the bicycle is ridden, so the power output may vary.

3.2.6. ESPDuino-32

ESPDUINO-32 development board base (Figure 3.9) on ESP-32 Wi-Fi module (Figure 3.10), it can lead to all ESP32 module pins. With Wi-Fi, Bluetooth 4.2, Ethernet, real-time map and other functions, ESPduino-32 is compatible with all versions of Arduino expansion boards. The MCU (Microcontroller unit) utilized is the ESP-WROOM-32, known for its powerful capabilities and built-in 2.4 GHz Wi-Fi (802.11 b/g/n) and Bluetooth 4.2/BLE (Bluetooth Low Energy) connectivity. It provides access to the ESP32's GPIO pins, which can be used for digital input/output, analog-to-digital conversion, PWM (Pulse-Width Modulation), and various other purposes. These pins make it versatile for interfacing with other components and sensors. The ESP-32 operates at an average 70-80 mW during active mode, taking in 3.3V of stable power supply, and allows for a wide range of input voltages, albeit some may require voltage regulation depending on the power source. Although the ESP-32 has a relatively low power consumption, the ESPDuino-32 development board consumes an average of about 700mW of power, about 10x the processor consumption.

Figure 3.9 - ESPDUINO-32 Development Board

Figure 3.10 - ESP-32 WiFi module

3.3 - System requirements and tuning

3.3.1 Power Supply

For the power supply it was opted for green energies since the project itself is focused on the environment and public health. For this reason, two green energy sources were chosen (Figure 3.11), solar energy was picked since the project is focused on outdoors activities and a bicycle was used, so it was also opted to use a dynamo to harness kinetic cyclical energy, storing all of it on a Li-ion battery module.

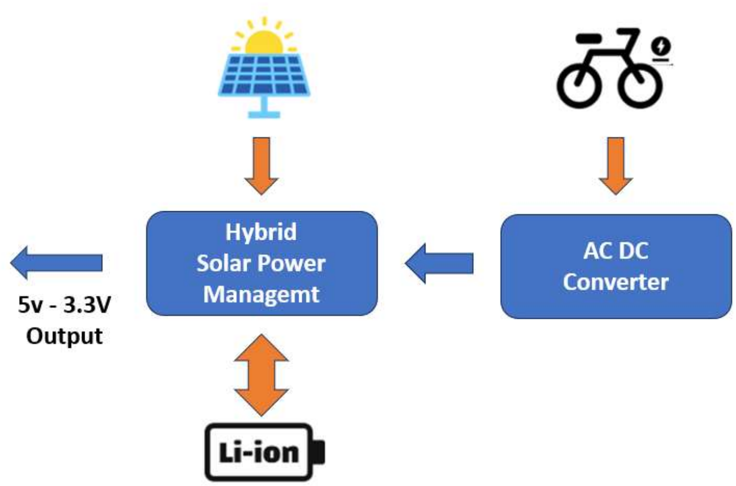


Figure 3.11 - Power generation system

3.3.1.1 Solar Panel

When installing a solar panel in a bicycle three aspects must be considered:

- Impact of wind pressure on the panel and bicycle stability.
- Optimal tilt of solar panels
- Delivery and storage of power

Effect of Wind Pressure

Exposing a solar panel to the wind creates pressure on the panel surface, on its support and on the bicycle. Dynamic air pressure (p_d) is correlated with the density of air (ρ) and wind speed (v). The maximum air pressure in the solar panel is created when the panel surface is placed perpendicular to the wind and can be calculated using the equation (*3.1*):

$$Force = p_d A = \ 0.5\, \rho\, v^2 A \tag{3.1}$$

Note: in practice wind force on an object creates more complex forces due to drag and other effects.

The problem with this force is that it depends on the square of the speed. Doubling the speed causes the quadruple of the force. As an example, a solar panel with 40cmx40cm placed in a lateral of a bicycle, and subject to a sudden lateral wind speed of 60km/h, creates in the panel a force of:

$$Force = 0.5\, \rho\, v^2 A = \ 0.5 * \ 1.2 * 60^2 * 0.16 = 346\, N = 35.28\, Kgf$$

Solar Panel tilt

Solar panel output depends on how efficiently it collects solar radiation. Best results are obtained when the panel is placed with its surface perpendicular to the sun rays. If not placed in the best position, there is a power loss.

Delivery and storage of power

Solar panel power output depends on the output voltage as seen in Figure 3.12, and is maximum at a specific voltage (peak power voltage). This voltage is specified by the solar panel manufactured and has typical values [6V, 9V, 12V, 18V, 24V]. To maximize solar panel energy production, an algorithm named MPPT (Maximum Power Point Tracking) was created. This algorithm is used by battery chargers to regulate the current extracted from the solar panel to keep the solar panel at the peak power voltage. If the solar panel voltage is lower than the peak power voltage the battery charger reduces the current extracted from the panel, and if its higher, increases the current consumption from the panel, regulating in this away the panel output voltage.

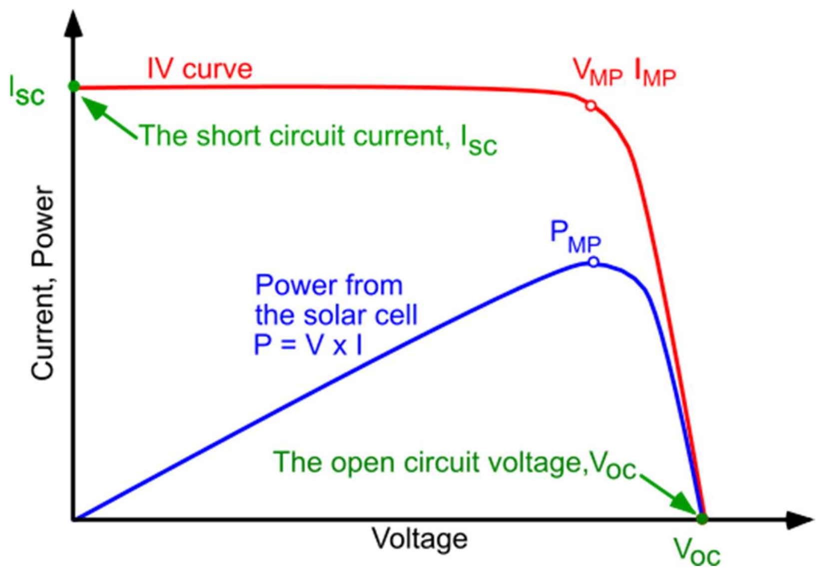


Figure 3.12 - Solar panel power generation graph

I_{SC} – Short circuit current: it is the maximum electrical current that the module can supply.

V_{oc} – Open circuit voltage: it is the maximum voltage that the module can supply.

I_{MP} – Maximum power current: it is the current that the module supplies when operating at its maximum power point.

V_{MP} – Maximum power voltage: it is the voltage that the module presents at its terminals when operating at its maximum power point.

P_{MP} – Maximum power: this is the peak power of the photovoltaic module.

MPP – It is the maximum power point of the photovoltaic module (maximum power point). It is found at the knee of the IV curve and at the peak of the PV curve.

3.3.3 Bicycle Dynamo

A dynamo is a device that is attached to the bicycle and contains an axle that rotates with the bicycle wheel. This axel has a permanent magnet that rotates inside a wire coil (Figure 3.13). According with Faraday law of induction, the rotation of the magnet creates a changing magnetic field that generates an electromagnetic force in the surrounding wire coil.

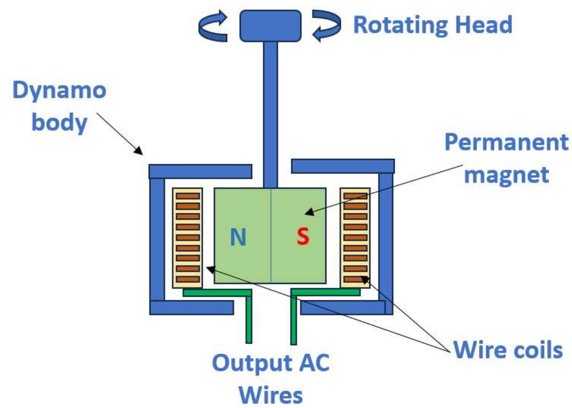


Figure 3.13 - Bicycle Dynamo internal functioning

The voltage generator Vg has an output voltage that is linear dependent on the dynamo rotation (frequency). The internal resistance R_{int} represents resistance of the dynamo wire internal coil and the inductor L_{int} the inductance of the dynamo internal coil (Figure 3.14). At low rotations the resistor is the main limiter of the dynamo performance and at high frequencies is the Inductor (its impedance depends directly on frequency).

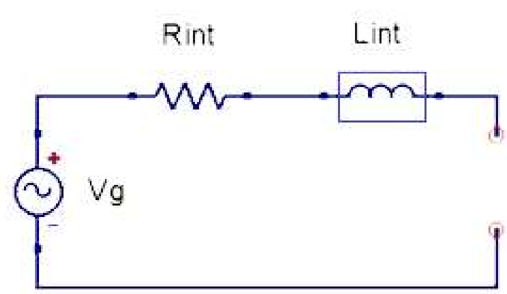


Figure 3.14 - Dynamo circuit representation

When the dynamo is connected to the bicycle, a direct relation can be established between the frequency of the dynamo voltage output, its rotation frequency and the bicycle speed (Figure 3.15). With this relation the Voltage generator Vg output can be expressed through equation (*3.2*).

Relation that is especially important to analyze dynamo performance in the context of a bicycle practical utilization.

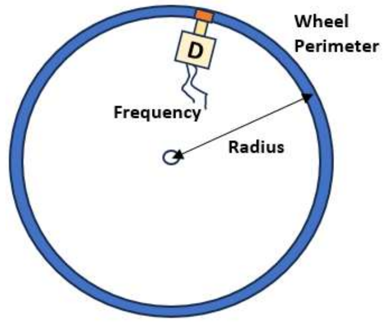


Figure 3.15 - Relation between dynamo performance and bicycle speed

$$\boldsymbol{Speed} = \frac{2 * \pi * wheel_{Radius} * \left(\frac{\boldsymbol{Dynamo}_{\boldsymbol{Frequency}}}{\boldsymbol{K}}\right) * 3600}{1000} \left(\frac{Km}{h}\right) \tag{3.2}$$

Note: K is a constant that represents the mechanical relation between the wheel rotation with the dynamo frequency.

To calculate the electrical model of the dynamo, first the dynamo output voltage is to be measured as a function of its rotation (Frequency) with no load, i.e., with no output connected at its terminals and using linear interpolation on the results. R_{int} and L_{int} are also calculated using the same process but adding a resistive load to the dynamo terminals. There are two types of commercially available dynamos for bicycles:

- The traditional low-cost dynamo that is installed in the bicycle frame and is connected by friction to the bicycle wheel. These dynamos come in two sizes: a small and more common 6V/3W and bigger and less common 12V/5.5W.
- More sophisticated and expensive dynamos that are installed internally in the front or back wheel axis.

3.3.3.1 Dynamo Dimensions

The 6v dynamos (Figure 3.16) are the most common type of dynamo in the market. They were designed to direct drive the bicycle lights in the old days that used to work at 6v. The 12v dynamo

(Figure 3.17) generates more voltage and more output power at the cost of being 1.8x bigger volume. Since the 6v dynamo is very small the volume difference between them is significant.

Figure 3.16 - 6V Dynamo

Figure 3.17 - 12V Dynamo

3.3.4 Switching from input power sources

Solar panel to dynamo switch or vice versa requires a manual operation: The attachment/detachment of the dynamo from the wheel. The dynamo extracts energy from the wheel and by consequence requires more physical effort from the bicycle user. For this reason, it must only be attached to the wheel when needed. This operation is performed by manually pressing a small lever in the dynamo as seen in Figure 3.18.

Figure 3.18 - Mounted dynamo lever mechanism

The commercially available battery charges cannot receive power from two different sources at the same time, so an input switch must be placed at the charger input. This switch is automatically activated when the CPU detects power generated by the dynamo (Figure 3.19). By default, the system connects the charger to the Solar Panel.

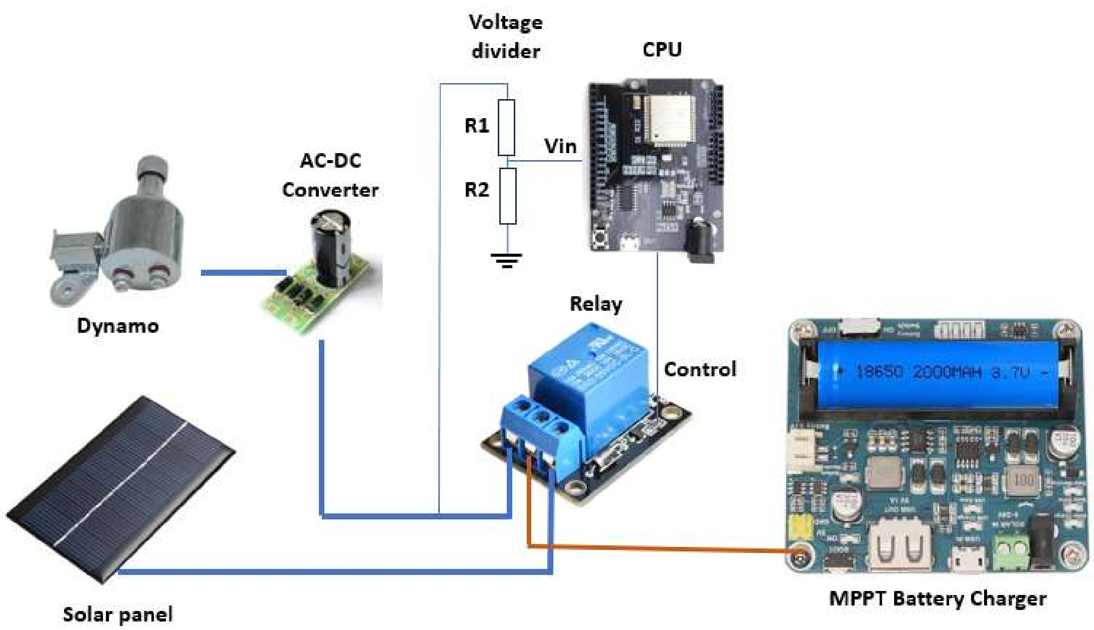


Figure 3.19 - Power source switch system

The ADC (analog to digital converter) of the system CPU is used to measure the dynamo voltage output after ac-dc conversion. A resistor voltage divider is used to scale the measured voltage to a value lower than maximum allowed voltage of the ADC (3.3 V). The CPU can also protect the system against dynamo overvoltage, caused for example by a bicycle high speed. In this case the CPU can switch the battery charger connection to the solar panel. After a fixed interval of normal voltage conditions at the dynamo AC-DC output, the CPU switches back the charger connection to the dynamo.

3.3.4.1 Dynamo AC-DC Converter

The dynamo produces AC voltage that needs to be converted to a steady DC Voltage. This is done by a Bridge Rectifier with a Capacitor to convert and smooth the output (Figure 3.20).

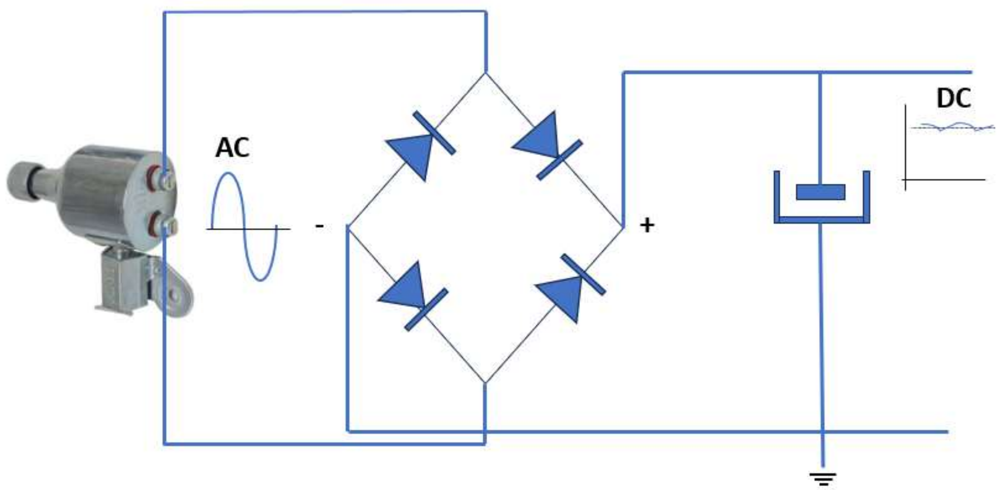


Figure 3.20 - Dynamo AC-DC converter

3.3.5 Battery management system

3.3.5.1 MPPT

For this project a low-cost 1 Cell Lithium 6v battery Solar charger with MPPT support was chosen. The charger (technically a battery management system) was used with both the solar panel and the bicycle dynamo. The battery used is a Lithium Cell with 3.7V with 850mA capacity. Depending on the battery charging levels, the battery management system alternates between 3 charging modes (as seen in Figure 3.21):

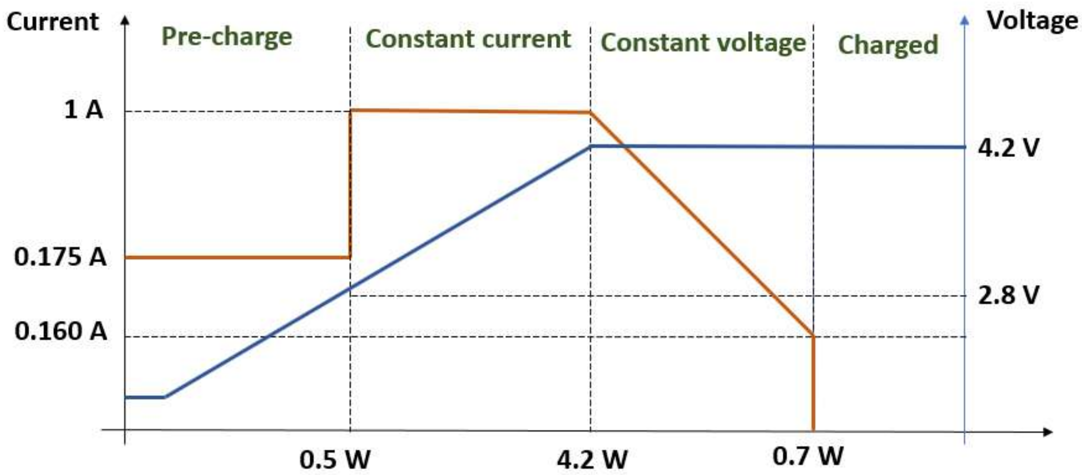


Figure 3.21 - MPPT battery management system

- Pre-charge mode: when the Cell voltage is below 2.8V the Cell is charged with a maximum constant current of 175mA.
- Constant current mode: when the Cell voltage is above 2.8Vand below 4.2V the Cell is charged with a maximum constant current of 1000mA.
- Constant voltage mode: when the Cell voltage is above 4.2V the Cell is charged with a maximum constant voltage of 4.2V.

Cell is considered charged when the charge current is below 160mA. The battery management system at its peak performance charges the battery with 1A @ 4.2V (4.2 Watts). MPPT support (Maximum power point tracking) means that this charger can regulate the current extracted from the input power source to keep the input at its optimum power level. MPPT was developed for solar panels but works with dynamos and other power sources.

3.4. Enclosure Design

To house and protect all the electronic components of the system, a box was designed in 3D (Figure 3.22) and printed using PLA 3D printer filament.

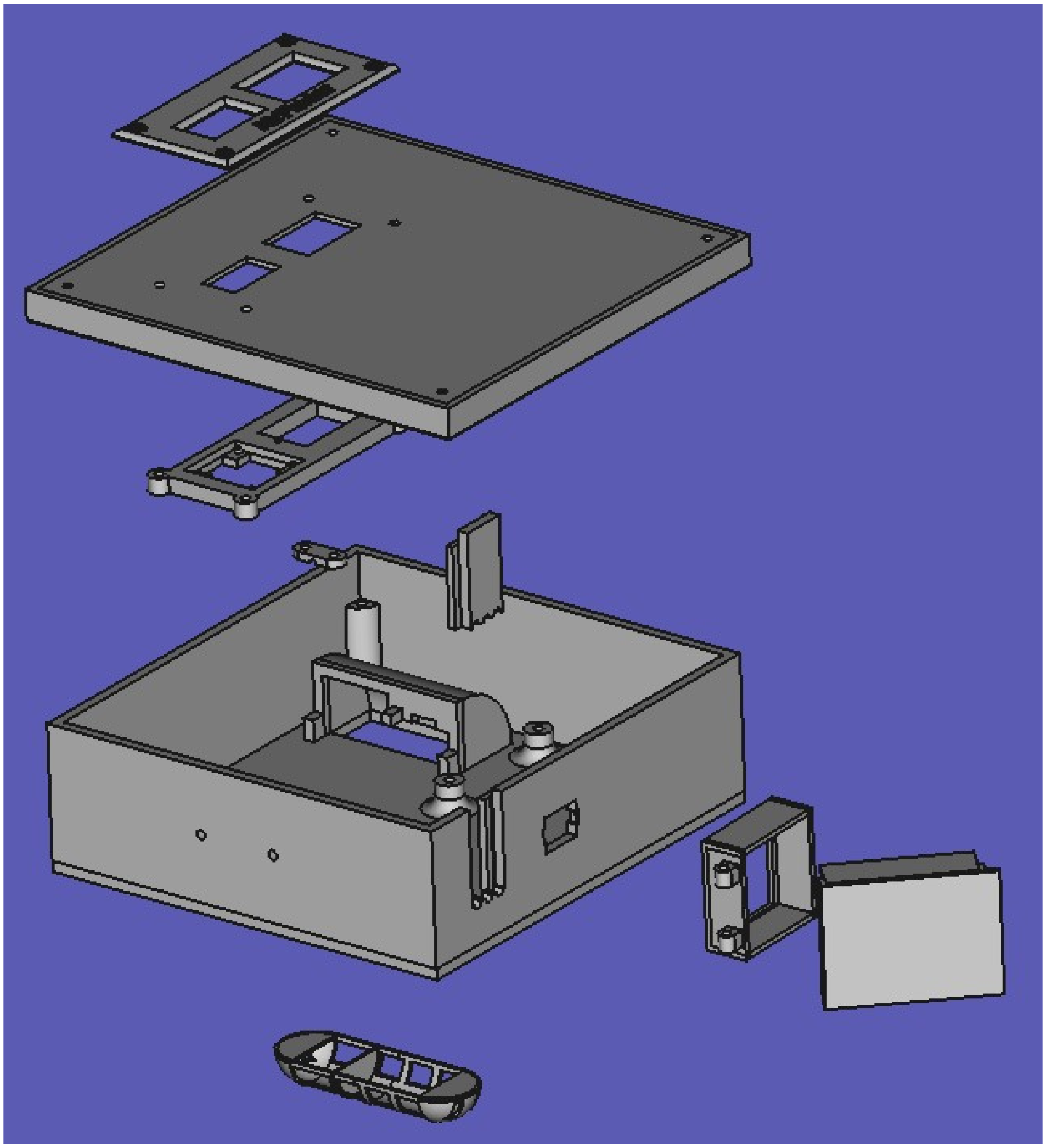


Figure 3.22 - Enclosure architecture in 3D modeler

The box was designed to protect the electric circuits from dust ingress and vertical water spray which represents an ingress protection of IP54 (IEC 60529), but due to the SEN55 an IP53 is more adequate (Figure 4.1). Having an open vent for air intake and extraction causes some issues as the intake fan may pull extracted air, causing wrong reading; and according to the sensor manufacturer in [20] "The sensor should be isolated from the airflow of the final device (e.g., air purifier) if the velocity of this flow is greater than 1 m/s which may be complicated given its mounted on a bicycle generally moving at more than 1 m/s. To avoid these issues, an additional acessory was designed to be mounted onto the open vent (Figure 3.23).

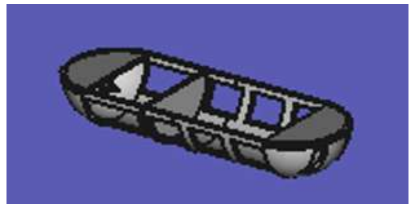


Figure 3.23 - Windshield for the Air Sensor

A wall was also inserted between the input and output of the sensor to prohibit air mixing between input and output, and a dome like wind protector was placed around the vent. Using FreeCAD [21] to design the piece, OpenFOAM [22] to run the simulation of air going through the piece and ParaView [23] to visualize the behaviour of incoming air, the results are shown in Figure 3.24 in a horizontal view, and in Figure 3.25 in a vertical view.

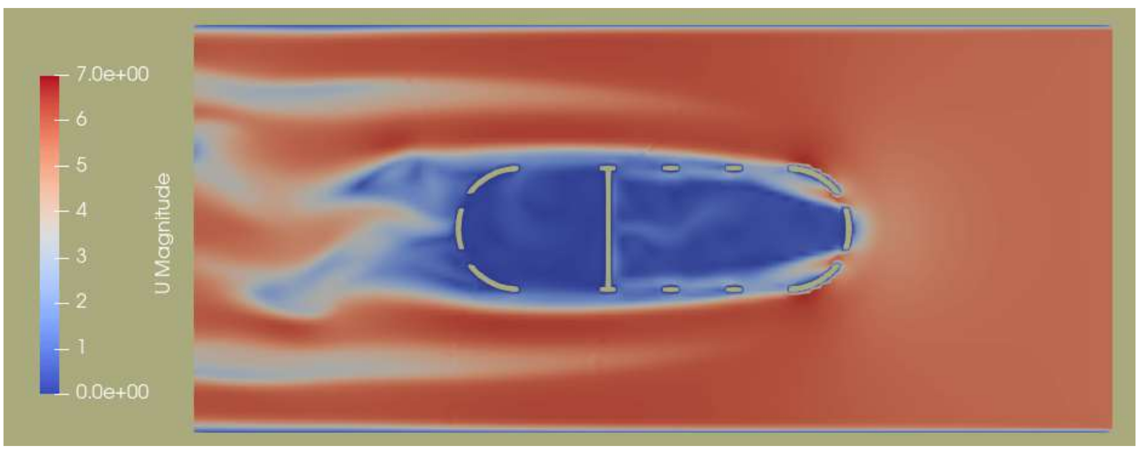


Figure 3.24 - Horizontal view of the behavior of incoming air

Seeing the air coming from the right, it gets barred at the entrance by the dome with little crefaces, limiting the air intake. The already limited air hits the side walls and the wall dividing both fans, therefore losing all its momentum, reducing the velocity of the incoming air to 1m/s or less. Due to the wall between fans, the output air can safely be expelled from the sensor on with exits from the sides and back of the dome. Seeing a vertical view, it confirms that the air goes into the sensor at a speed lower than 1m/s.

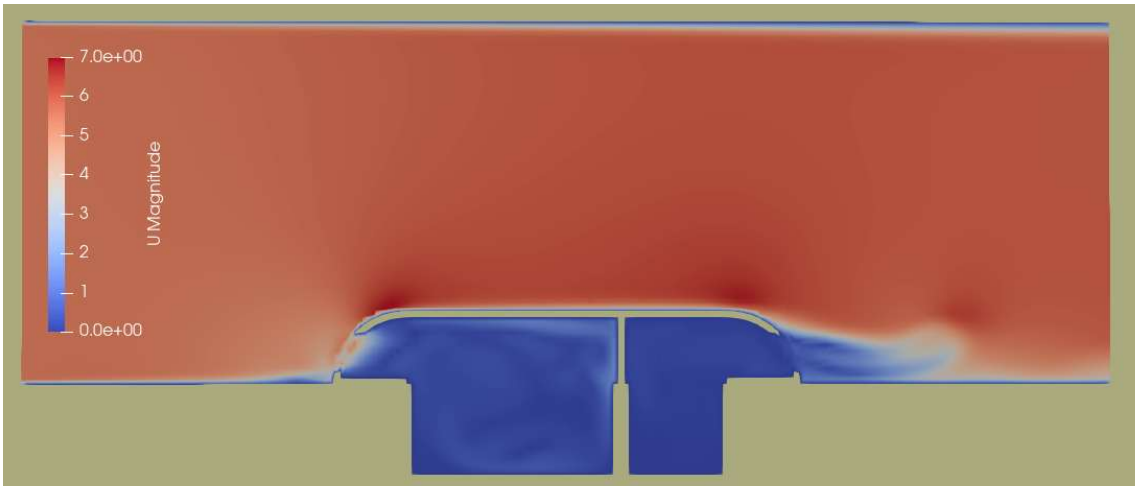


Figure 3.25 - Vertical view of the behavior of incoming air

3.5. Software Interface

This project involves the development of two crucial software components: an Arduino application and an Android native application, each tailored to fulfil specific functions within the system.

3.5.1. Software architecture and high-level decisions

Arduino Application

The Arduino application was developed using the Arduino IDE. Arduino was selected for its proven robustness and extensive compatibility with a wide variety of sensors, which is essential for the project's diverse sensor requirements. The platform's rich ecosystem of open-source libraries simplifies the integration and control of various sensors. Within this project, the Arduino application acts as the central hub for sensor communication, managing the traffic of information between multiple sensors and ensuring seamless data acquisition. Once data is collected from these sensors, the application consolidates the information and broadcasts it using Bluetooth Low Energy (BLE). BLE was chosen for its efficiency in energy consumption, which is critical for prolonged operation on battery-powered systems.

Android Native Application

The Android native application, developed using Android Studio, serves as the user interface and processing unit. It was chosen primarily for its suitability on mobile devices, making it ideal for real-time use while cycling. Although an iOS application was evaluated, it was ultimately excluded due to several factors: the higher cost of Apple hardware, the expense of an Apple Developer license, and a steeper learning curve associated with iOS development. The Android app offers a sophisticated user interface designed for ease of use and rapid interaction. One of the app's key features is its capability to provide near real-time analysis of air quality data. This functionality allows cyclists to receive immediate feedback on air quality conditions along their route, empowering them to make quick decisions and alter their course to avoid areas with poor air quality. The app also integrates with Google Maps and GPS tracking, enhancing the user experience by providing real-time geolocation data and navigation support. This integration ensures that users have a comprehensive and interactive interface, capable of displaying air quality variations in conjunction with their exact location on the map. By leveraging GPS, the app can deliver precise updates and recommendations based on the user's current route and surroundings, ensuring an authentic and responsive experience. Overall, this dual-component system effectively combines hardware and software capabilities to offer a robust solution for real-time air quality monitoring and navigation for cyclists. The Arduino application handles the intricate details of sensor data collection and communication, while the Android app provides a user-friendly interface for real-time analysis and decision-making, significantly enhancing the cyclist's ability to navigate and respond to varying air quality conditions.

3.5.2. Software detailed view and functioning

3.5.2.1. Arduino Application

In an Arduino application, two primary functions structure the program: setup() and loop().

- setup() Function: This function runs only once, immediately after the Arduino is powered on or reset. It is used for initializing hardware devices, setting pin modes, and configuring initial settings. All initializations and setups required for the hardware are placed here.
- loop() Function: As its name suggests, loop() executes repeatedly in a continuous cycle after setup() has completed. This function contains the core logic of the program, managing ongoing tasks and operations. Code within loop() should avoid using blocking functions that cause waiting, as this can interrupt the continuous execution of the program.

For this application, the setup() function is responsible for initializing all components and establishing necessary communications:

- BLE Server Initialization: A Bluetooth Low Energy (BLE) server is set up with a single service. This service is configured to broadcast a JSON string containing all air pollutant measurements.
- Sensor Initialization: All sensors required for air quality monitoring are initialized and configured for operation.

- Power Source Management: The relay responsible for selecting the system's power input is activated, switching the power source to the solar panel.

The loop() function is responsible for the continuous execution of tasks such as sending sensor data, updating displays, and managing power sources:

- Sensor Data Read and Transmission: It periodically sends sensor data via Bluetooth Low Energy (BLE) notifications at defined intervals, ensuring that up-to-date air quality information is broadcast.
- Display Updates: The latest sensor measurements are used to update the LCD display, providing real-time feedback to the user.
- Power Source Management: The function continuously monitors the status of the dynamo and controls the relay to switch between power sources based on whether the dynamo is active or inactive.

The Code is divided into three files, working with each other (Figure 3.26), all having distinct responsibilities:

1. Comms_PC.ino

This is the main file responsible for internal communication with sensor box components and external communication via Bluetooth Low Energy (BLE). It includes the two standard Arduino functions:

setup(): Initializes all components and establishes communication protocols. This includes:

- BLE Setup: A BLE server is created with one service that has a single characteristic to transmit a JSON string containing air pollutant measurements.
- USB Serial Initialization: Sets up serial communication for debugging and data exchange.
- Power Management: Activates the relay to select the solar panel as the default power source for the power management module.

loop(): Continuously performs several tasks:

- BLE Data Transmission: Sends sensor data via BLE notifications at regular intervals.
- LCD Updates: Updates the LCD display with the latest measurements.
- Power Source Switching: Monitors the dynamo status and switches the relay between power sources, accordingly, using the dynamo when available and defaulting to the solar panel otherwise.

2. SEN55_SDN_T.ino

This file handles the functionality related to the SEN55_SDN_T air quality measurement sensor:

setup() Function: Initializes the SEN55_SDN_T sensor using specific functions from its library to prepare it for data collection.

loop() Function: Continuously reads values of various air pollutants, including:

- Mass Concentrations: PM1.0, PM2.5, PM4.0, PM10.0.
- Environmental Factors: Ambient humidity and temperature.
- Indices: VOC index and NOx index.

Loop() also broadcasts these values using Bluetooth Ble.

3. LCD.ino

This file manages the SSD1306 LCD screen for displaying air pollutant data:

setup() Function: Attempts to connect to the LCD at I2C address 0x3C, preparing it for data display.

loop() Function: Controls the visual display on the screen:

- Connectivity Indicator: Displays a permanent indicator showing the connection status with the user's phone, using text ("ON/OFF") and a Wi-Fi symbol.

- Air Pollutant Display: Rotates through different air pollutants, showing each pollutant's latest value in a dedicated space on the screen.

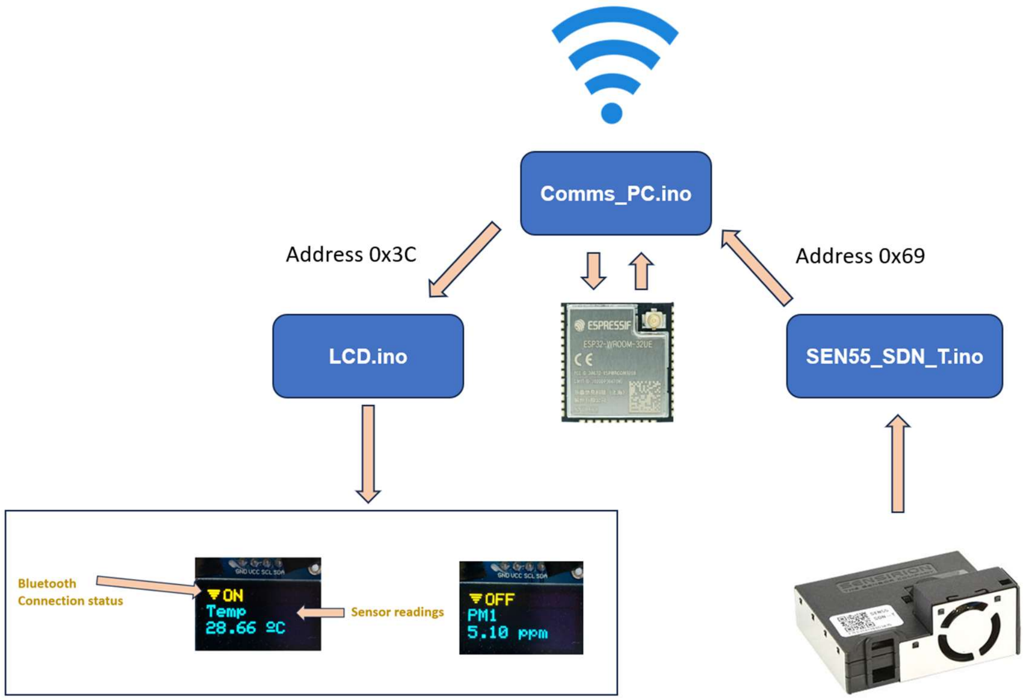


Figure 3.26 - Arduino architecture and interactions

3.5.2.2. Android Application

The Android App, named *AirFinder*, is responsible for the user UI and hosts the system database where all historic measurements are stored. Sensor data is transmitted from the sensor box to the android phone in real time through Bluetooth BLE (Figure 3.27). Air quality data is stored in the user's android phone in a SQLite3 database. Database structure is very simple (Figure 3.28). Measured values with the associated geo location are organized in "Paths", representing the bicycle's journey where the measurements were collected.

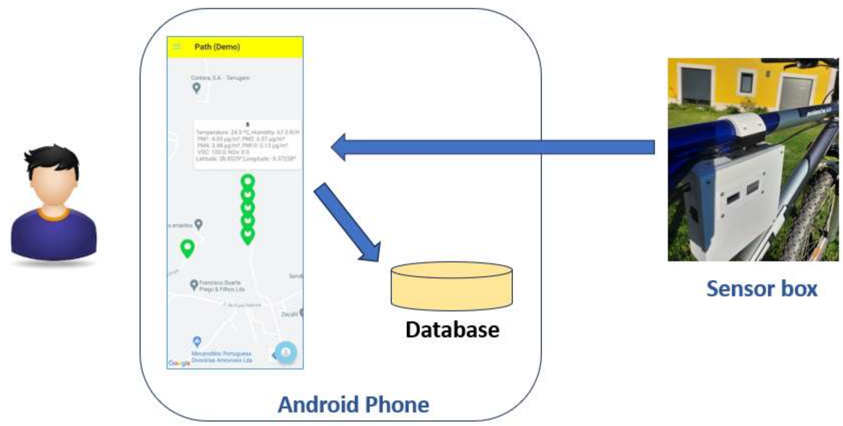


Figure 3.27 - AirFinder Android application

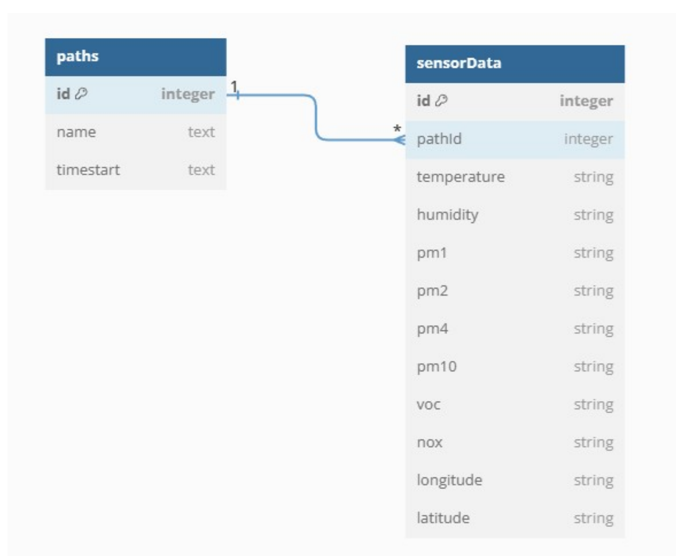


Figure 3.28 - Object classes in database

The Android App is a native application coded in Java. Navigation is based in a Navigation drawer accessible from the top left corner of the App. Figure 3.29 shows App screens flow.

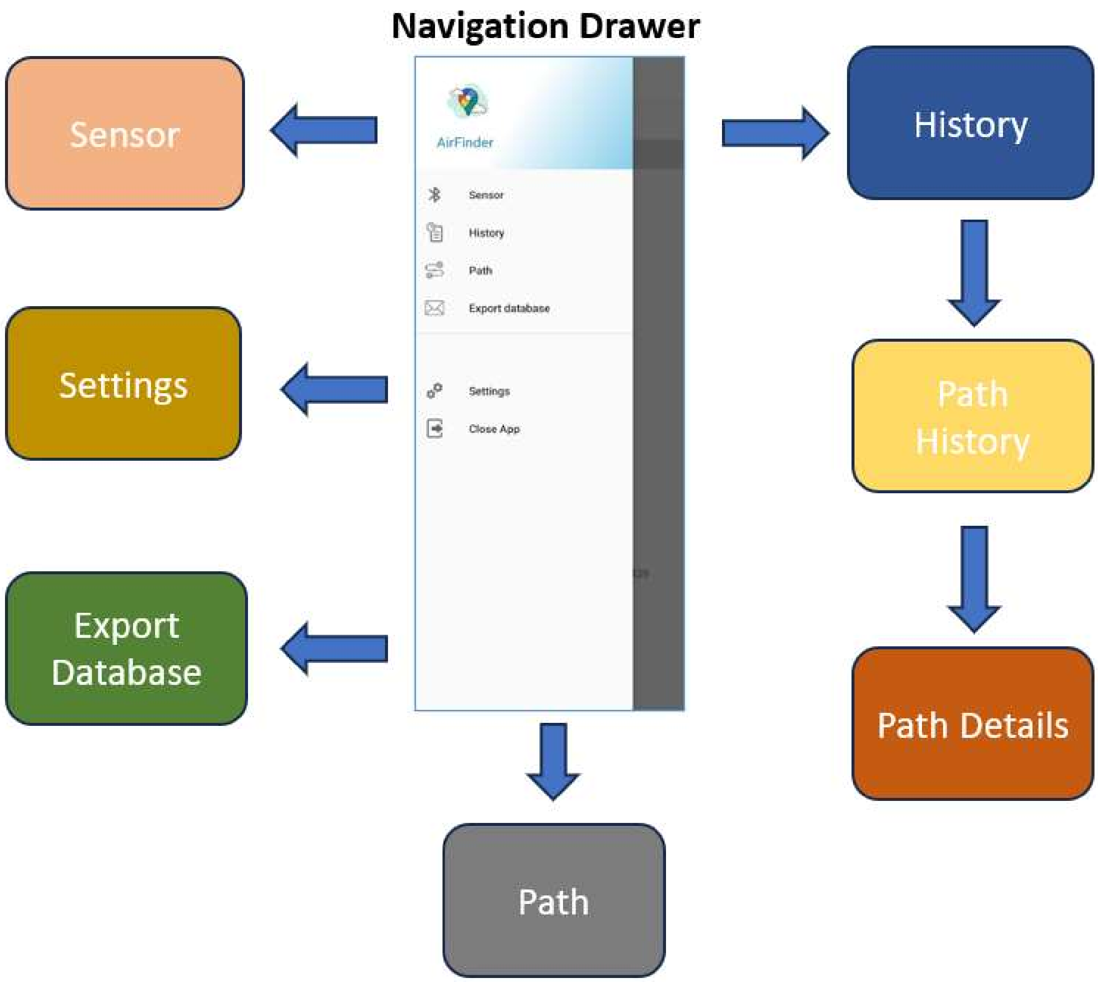


Figure 3.29 - AirFinder activity interaction

SplashScreenActivity Launched at startup. A Splash Screen image is show and the app initialization code is run:

- App permissions are verified, for example GPS permissions, and if missing, a request is made to have user permissions.
- SQL Database is created if nonexistent or its structure is upgraded if needed (for example in a new App version upgrade).

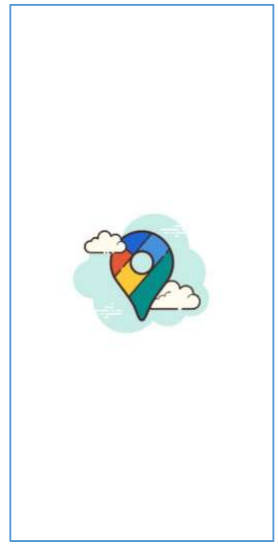


Figure 3.30 - SplashScreenActivity

SensorActivity Once the code behind the splash screen runs, the user is redirected to this Activity. Here the user can manage the sensor box connection and visualize the latest measurements received from the sensor box. Its content is updated every time it receives a Bluetooth broadcast from the sensor box with new data. Sensor box connection is managed through a “connect to sensor” button, when pressed, automatically searches via Bluetooth for a device with the name “ESP32”, and when found connects to it.

Figure 3.31 - SensorActivity

NavigationActivity This activity is responsible for the Navigation drawer accessible from a sidebar icon located in the top left of the app. Opening said sidebar allows for a multitude of options. Depending on what option the user has clicked, the app will launch to the according activity. Choosing the “Close App”, the app is closed, saving all changes internally as cache for possible later usage.

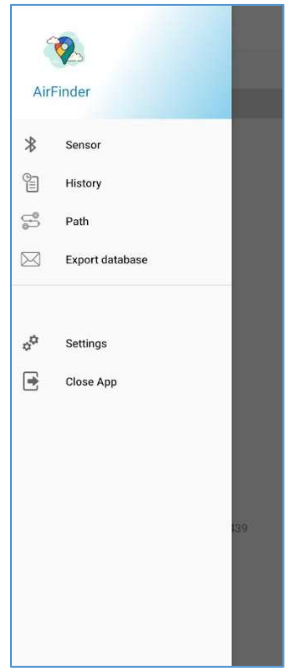


Figure 3.32 - NavigationActivity

MapScreenActivity This activity is launched from the Navigation Drawer by choosing the “Path” option. After launch, the world map appears centered and zoomed to the current user location. Here the user has the option to start recording new Paths, tracing the quality of the air throughout a bicycle route via markers placed every time there is new incoming data from the sensor box. Each measurement is represented on the map using markers, pinpointing air quality along the path. Furthermore, the user can click a marker and check every variable’s value, having the possibility to change direction if the air quality is highly degraded on the current path. If the user is done is the path, he can stop recording by clicking a button, storing all the data of the path to the app’s internal database.

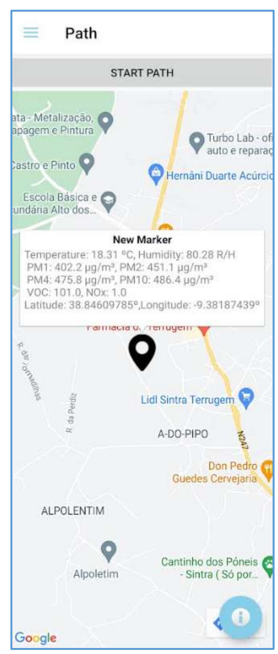


Figure 3.33 - MapScreenActivity

ExportActivity This activity is launched from the Navigation Drawer by choosing the "Export" option. Here the user can export any stored Path data from the database to an external MS Excel file.

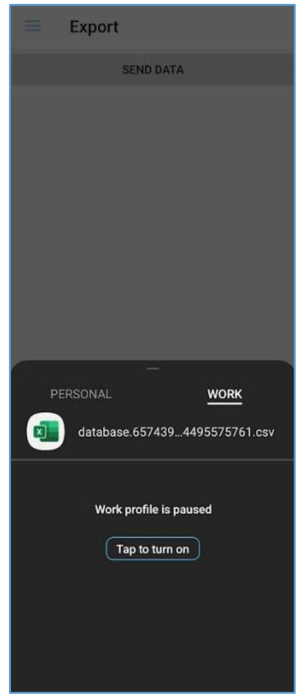


Figure 3.34 - ExportActivity

SettingScreenActivity This activity is launched from the Navigation Drawer by choosing the "Settings" option. As the activity name implies, here can set some application options:

- Enable the Demo mode (which will populate HistoricScreenActivity with a generated database content of data for functionality testing).
- Choose the Bluetooth Sensor box name. This name is used by the SensorActivity to connect to the sensor box.
- Alter the thresholds for every measured variable. These thresholds are used by the app to classify the air as healthy or not.

Figure 3.35 - SettingScreenActivity

HistoricScreenActivity This activity is launched from the Navigation Drawer by choosing the “History” option. Previously recorded paths are shown.

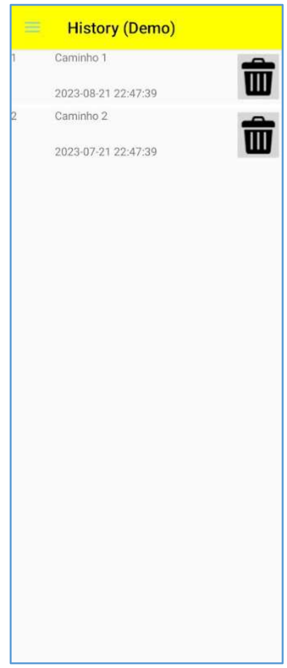


Figure 3.36 - HistoricScreenActivity

HistoricMapActivity This activity shows the world map with a previously recorded path with all its markers. Each marker’s variables are evaluated and depending on their value compared to their threshold, the marker’s color changes between green, yellow and red, signaling the quality of the air on that area in an easier and more visually appealing way. This map also holds a bottom right button that once clicked, takes the user to a more analytical view of the path.

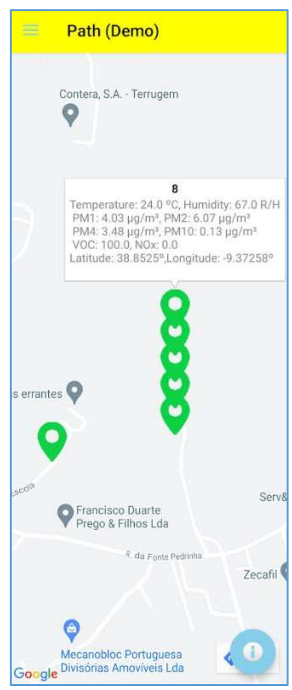


Figure 3.37 - HistoricMapActivity

HistoricStatsActivity This activity holds a graphical analysis of the air quality on the path. First a line graph is shown of a variable's value fluctuation throughout the path, and it can be changed by clicking on the name of another variable; and below there are averages of every variable represented in half gauge graphs, giving an overall evaluation of the quality of each variable, considering the threshold defined earlier in the settings.

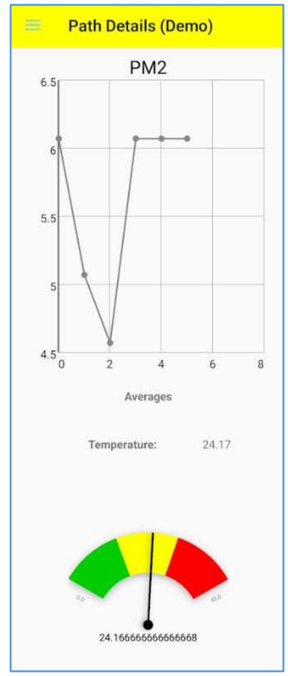


Figure 3.38 - HistoricStatsActivity

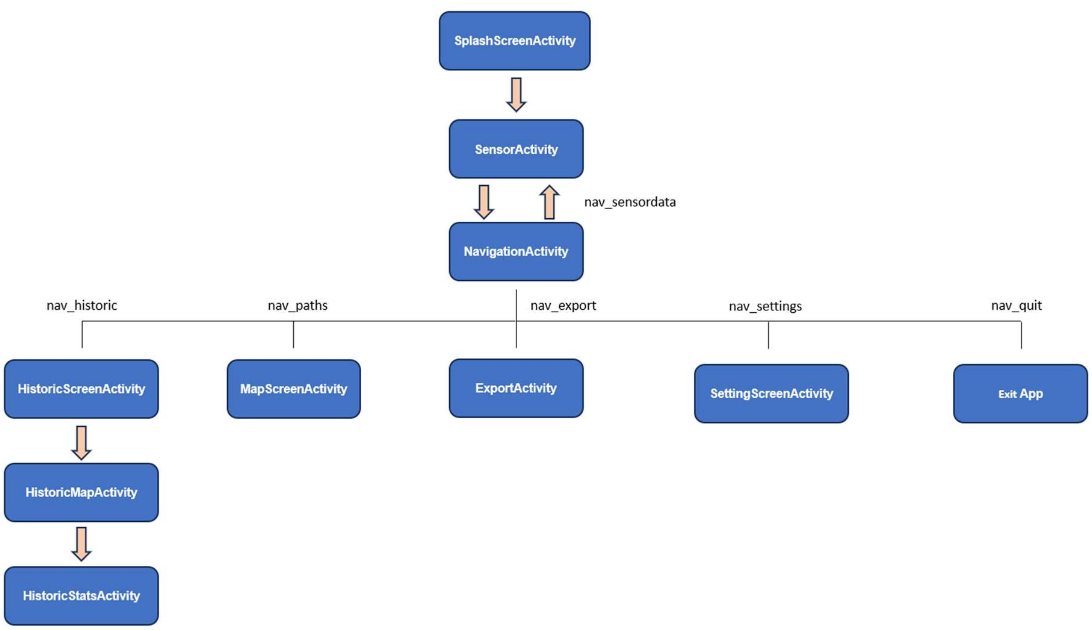


Figure 3.39 - AirFinder class interaction fluxogram

Support classes

Object models objects have been created for Paths, **SensorData** and **ExportData** to make it easier to manipulate the database.

SensorDatabaseHelper holds all the methods used in manipulation of Path and SensorData database, namely creation, deletion, insertion of rows, and get data functions.

PathAdapter is used to allow paths stored in **HistoricScreenActivity** to be clickable despite not having natively clickable components, sending user to **HistoricMapActivity** with the id of the path clicked.

SensorService handles all types of communication, be it internal broadcast communication of **MapScreenActivity**, connectivity to sensorBox, and initial treatment of data coming from the sensor box.

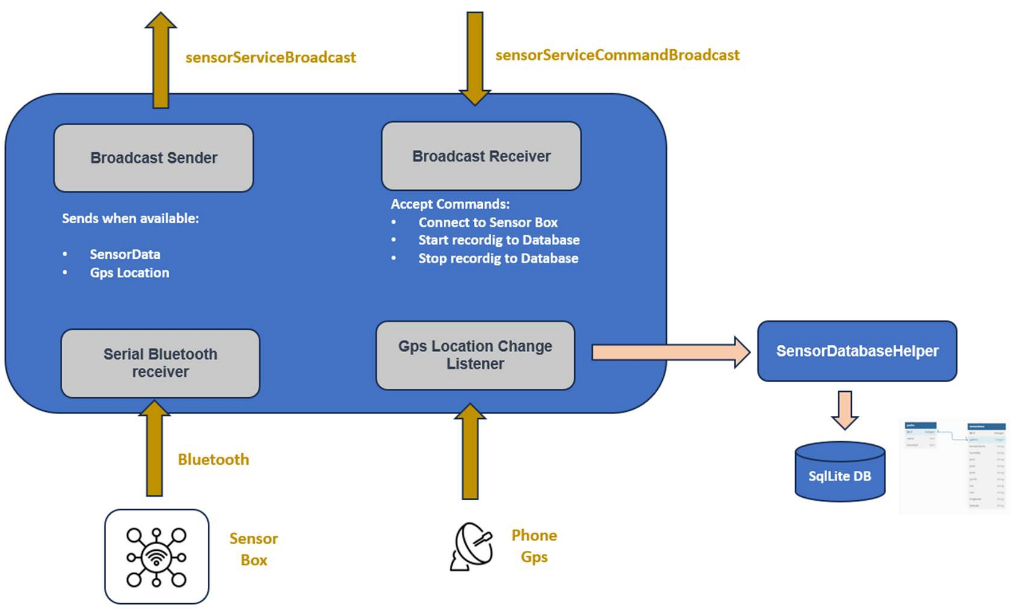


Figure 3.40 - AirFinder internal communications

AppSettings holds all threshold initial app defined and saved user defined values from **SettingScreenActivity** and activities that require said thresholds. According to official sources, the app defined values are as following: PM10 – 45 µg/m^3, PM4 - 45 µg/m^3, PM2.5 - 15 µg/m^3, PM1 - 10 µg/m^3, NO_X - 250 NO_X Index, VOC - 250 VOC Index, Temperature – 30 C and Humidity – 66.6 % R/H

AirFinderApplication counts the current path ID to be used in path recording and database manipulation.

Detail_SensorData_Activity is used to treat data and load it into info Windows of **MapScreenActivity** and **HistoricMapActivity** markers.

3.6 Usability tests

During software development usability tests were developed to assess how easily users can navigate the app, complete tasks, and achieve their goals. The primary objective was identifying usability issues, understanding user satisfaction, and gathering insights for improving the app's design and functionality.

3.6.1 Test Users

Given the objective of the app to promote public health, the ideal target audience is quite large. It was defined that our target audience would be teenagers and adults who are more likely to possess the physical capabilities required for operating a bicycle equipped with a dynamo. Considering the characteristics of this audience, we designed the system to be user-friendly for this specific demographic. The app is also designed exclusively for Android operating phones, further narrowing our focus to Android users within the target age range. Therefore, our usability tests will concentrate on teenagers and adults who own an Android mobile device, ensuring the system is effectively tailored to their needs and capabilities.

Rationale for Target Audience

Physical Capability

Teenagers and Adults: This group is generally better equipped to handle the higher physical effort required to operate the dynamo on the bicycle. Including this demographic ensures that the usability test reflects realistic user experiences and feedback.

Platform Specificity

Android Users: Since the app is developed for Android devices, it is essential to test with users who are familiar with the Android operating system. This ensures that the feedback is relevant to the app's intended user base and considers the nuances of the Android platform.

Sample Size

For the initial phase of usability testing, we have chosen a sample size of 5 participants. This decision is based on the understanding that a small, focused group can effectively identify major usability issues and areas for improvement. According to Jakob Nielsen’s principles, testing with five users typically uncovers about 85% of usability problems, making it a cost-effective and efficient approach for early-stage testing. After applying changes to the app driven by the first batch of user feedback, a second batch of tests target 22 participants aged 15 to 58 to evaluate the application. Throughout these sessions, participants provided their feedback verbally in response to the questions.

Summary

Target Audience: Teenagers and adults using Android devices.

Physical Requirements: Focus on users capable of handling higher physical exertion.

Sample Size: First batch of 5 participants to identify key usability issues and improvements, second batch of 22 participants to evaluate the app

By targeting this specific audience, we aim to gather actionable insights that will help refine the app and ensure it meets the needs and expectations of its primary users.

3.6.2 Usability Test Script

To conduct the usability tests, a detailed script was developed to guide the moderator through each session. These tests were conducted in Lisbon, in the Campo Grande Garden, involving participants aged 19 to 24 who were passing by the bike lane. Throughout the sessions, participants provided their feedback verbally in response to the questions.

Introduction

- Welcome the participant.
- Explain the purpose of the session: "We are conducting a usability test for the AirFinder app, which is designed to help users monitor air quality while riding a bicycle."
- Ensure the participant understands there are no right or wrong answers, and their honest feedback is valuable.
- Obtain consent to use the data recorded in the session.

Scenario 1: Planning and Recording a Path

Moderator:

- "In this first scenario, you are planning to travel a path of 1 kilometer on your bicycle and decide to record air quality data using the AirFinder app to monitor the environmental conditions during your ride."
- "Please perform the tasks as you would naturally. I won’t be able to help, but I’ll observe how you interact with the app."

Tasks

1. Ensure successful Bluetooth connection with the SensorBox:
    - Moderator: "Please connect your phone to the SensorBox via Bluetooth."

- Subtasks:
  - Turn on Bluetooth on your phone.
  - Locate the SensorBox in the device list.
  - Establish a connection.
- Expected Outcome: Successful pairing with a confirmation message on the app.
- Post-Task Likert Scale Question: "How easy was it to connect your phone to the SensorBox via Bluetooth?"
  - Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

2. Commence recording a path on the AirFinder app:
   - Moderator: "Now, please start recording a new path in the AirFinder app."
   - Subtasks:
     - Open the app.
     - Navigate to the path recording feature.
     - Start a new recording session.
   - Expected Outcome: Recording begins with real-time air quality data display.
   - Post-Task Likert Scale Question: "How easy was it to start recording a new path?"

     Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

3. After traveling the planned path, end path recording:
   - Moderator: "Please stop the recording session, save the recorded path, and name the session."
   - Subtasks:
     - Stop the recording session.
     - Save the recorded path.
     - Name the session.
   - Expected Outcome: Path recording is successfully saved with all data intact.
   - Post-Task Likert Scale Question: "How easy was it to stop, save, and name the recorded path?"

     Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

Scenario 2: Reviewing and Analyzing Recorded Paths

Moderator:

- "In this next scenario, you've used the AirFinder app to record several paths. Now, you decide to review the recording of your most recent path and analyze the collected air quality data"

Tasks:

1. Choose your most recent recorded path:
   - Moderator: "Please access and select your most recent recorded path."
   - Subtasks:
     - ➢ Access the recorded paths list.
     - ➢ Identify the most recent one.
     - ➢ Select it.
   - Expected Outcome: The app displays the details of the selected path.
   - Post-Task Likert Scale Question: "How easy was it to access and select your most recent recorded path?"
     - ➢ Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

2. Check the analytics page of your path:
   - Moderator: "Now, please navigate to the analytics section from the path details page."
   - Subtasks:
     - ➢ Navigate to the analytics section.
   - Expected Outcome: The analytics page shows comprehensive data visualizations and summaries.
   - Post-Task Likert Scale Question: "How easy was it to navigate to the analytics section?"
     - ➢ Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

3. Visualize the air pollutant metrics along the path:
   - Moderator: "Please explore the data visualizations to see variations in air quality along the path."
   - Subtasks:
     - ➢ Explore the data visualizations.
   - Expected Outcome: Clear and interactive graphs or maps showing pollutant levels.
   - Post-Task Likert Scale Question: "How clear and helpful were the data visualizations for understanding air quality variations?"
     - ➢ Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

4. Export the path data and send it via email for more in-depth independent analysis:
    - Moderator: "Please export the path data and send it via email."
    - Subtasks:
        - ➢ Use the export function.
        - ➢ Select email as the sharing method.
        - ➢ Send the data file.
    - Expected Outcome: Data is successfully exported and received via email.
    - Post-Task Likert Scale Question: "How easy was it to export the path data and send it via email?"
        - ➢ Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

Scenario 3: Customizing Air Quality Metrics

Moderator:

- "In this final scenario, after analyzing the air quality data and comparing it with other sources, you realize that the default metrics may not be the most appropriate for your conditions, so you decide to customize them."
- "Please perform the tasks as you would naturally. I won't be able to help, but I'll observe how you interact with the app."

Tasks:

1. Change the air pollutant threshold metrics:
    - Moderator: "Please adjust the air pollutant threshold metrics in the app settings."
    - Subtasks:
        - ➢ Access the settings or preferences menu.
        - ➢ Find the section for threshold metrics.
        - ➢ Adjust the values.
    - Expected Outcome: New threshold values are saved and applied.
    - Post-Task Likert Scale Question: "How easy was it to adjust the air pollutant threshold metrics?"
        - ➢ Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

2. Check the analytics page of your path:
    - Moderator: "Please revisit the analytics section for any previously recorded path."

- Subtasks:
  - Revisit the analytics section.
- Expected Outcome: Analytics page reflects the updated threshold metrics in its visualizations and summaries.
- Post-Task Likert Scale Question: "How accurately did the analytics page reflect the updated threshold metrics?"
  - Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

3. Visualize the air pollutant metrics along the path with the updated threshold limits:
   - Moderator: "Please review the graphs or maps to see how the data is displayed with the new thresholds."
   - Subtasks:
     - Review the graphs or maps.
   - Expected Outcome: Visualizations clearly indicate the impact of the updated metrics.
   - Post-Task Likert Scale Question: "How clear were the visualizations after updating the threshold metrics?"
     - Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)

Conclusion

Moderator:

- Thank the participants for their time and feedback.
- Ask if they have any additional comments or suggestions regarding the app.
- End the session.

3.6.3 Interview Analysis

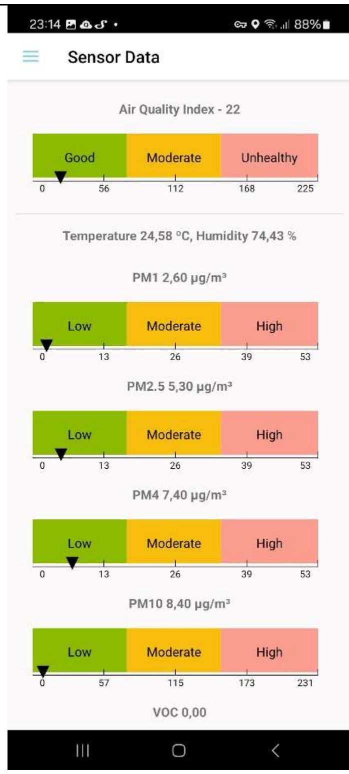
First round with 5 participants

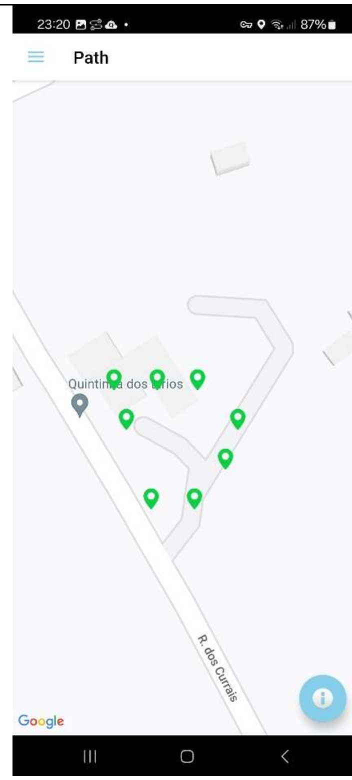
Following the completion of the usability tests with the five participants in the first round, an informal discussion was held to gather feedback on potential improvements for the app. Table 3.2 summarizes the findings and the corresponding resolutions implemented for each identified issue. After these enhancements, the app's user interface underwent significant positive changes, as illustrated in Figure 3.41.

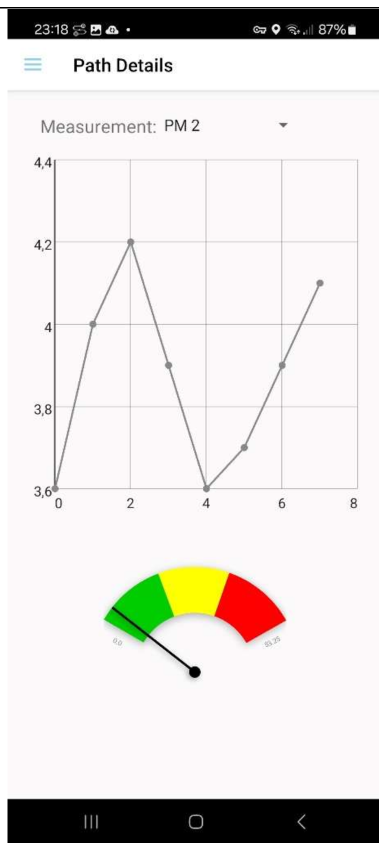
Table 3.2 - Issues identified in first group usability test and the implemented resolutions

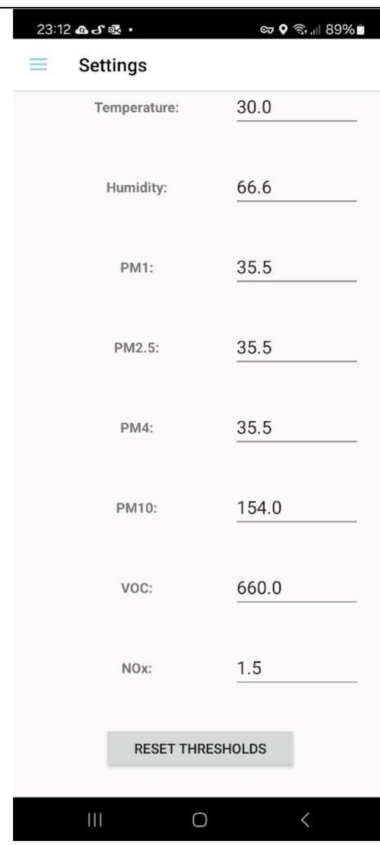
ID	Issue description	Severity	Implemented resolution

1	The settings screen lacks a feature to reset thresholds to predefined values	Major	In Settings, an additional button has been included to reset thresholds.
2	The measurement markers on Google Maps are excessively large.	Minor	The size of Google map markers has been reduced
3	When starting a new path, the previous markers are not cleared.	Critical	Markers are now automatically cleared when starting a new path
4	The path details are unclear.	Major	A combo box was added to allow users to select each measurement. Upon selection, a detailed technical data graph and a qualitative gauge are displayed
5	Users should receive notifications when air quality parameters exceed configured values	Minor	An Android Notification is sent every minute when Sensor parameters exceed the configured values. A new setting has been added to enable or disable this feature.
7	The app's initial screen displays technical sensor data that may be unreadable for typical users.	Major	All sensor readings are now presented with qualitative gauges alongside their numerical values. The sensor screen has been redesigned to include a standard Air Quality Index (AQI) specifically for PM2.5.
6	The exported data does not include the timestamp for each measurement; it only includes the timestamp of the start of the path.	Minor	Database schema changed to contain measurement timestamps and Recording and Export procedure changed accordingly.
7	During a Path recording, the Android screen enters sleep mode.	Critical	Sleep mode disabled during app usage

<table>
<tr>
<td>


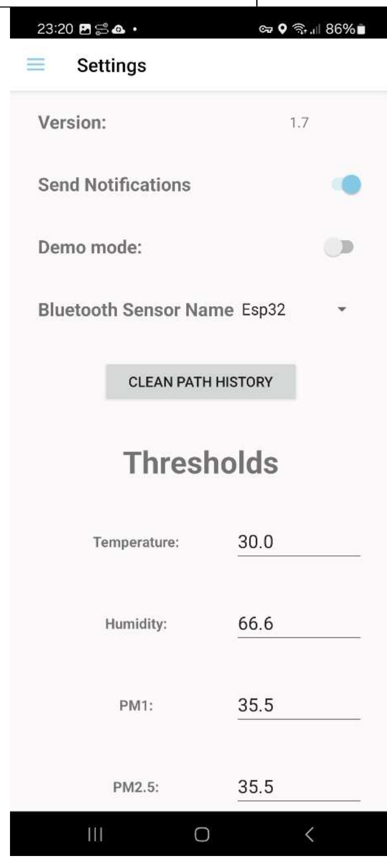
</td>
<td>


</td>
</tr>
<tr>
<td>SensorActivity – Rework, now displaying qualitative information</td>
<td>HistoricMapActivity/MapScreenActivity – Decreased marker size</td>
</tr>
<tr>
<td>


</td>
<td>


</td>
</tr>
</table>

HistoricStatsActivity – Data visualization cleanup	**SettingScreenActivity** – “Reset Thresholds” Button



SettingScreenActivity – “Send Notifications” Toggle

Figure 3.41 - Implemented changes after first group usability test

Second round with 22 participants

The second round was performed after correcting the issues identified in round 1, at a gym in Colares to participants aged between 16 and 59 years old. Table 3.3 shows the statistical values of the Post-Task Likert Scale Question. Statistical values show very high satisfaction ratings with the App performance. A future point of improvement is the Bluetooth connection with the Sensor box.

Table 3.3 - Likert Scale Question statistical analysis

Post-Task Likert Scale Question	Mean	Median	Mode	Stdev
How easy was it to connect your phone to the SensorBox via Bluetooth?	3,9	4	4	0,81
How easy was it to start recording a new path?	4,5	5	5	0,74
How easy was it to stop, save, and name the recorded path?	4,1	4	4	0,77
How easy was it to access and select your most recent recorded path?"	4,4	4,5	5	0,73
How easy was it to navigate to the analytics section?"	4,1	4	5	0,83
How clear and helpful were the data visualizations for understanding air quality variations?	4,2	4	4	0,69
How easy was it to export the path data and send it via email?	4,3	5	5	0,84
How easy was it to adjust the air pollutant threshold metrics?	4,1	4	4	0,81
How accurately did the analytics page reflect the updated threshold metrics?	4,1	4	4	0,68
How clear were the visualizations after updating the threshold metrics?	4,5	4	4	0,51

There were also some interesting suggestions related with future developments and additional features:

Real-Time Weather Conditions

- **User Need:** Cyclists often encounter changes in weather conditions that can affect their ride.
- **Upgrade Feature:** Integrate a real-time weather monitoring system that provides users with information on temperature, humidity, wind speed, and precipitation. This feature can help users plan their routes and prepare for adverse weather conditions, enhancing their overall cycling experience.

Pollen Count Monitoring

- **User Need:** Allergy sufferers are often concerned about pollen levels while cycling.
- **Upgrade Feature:** Add a pollen count feature that displays current pollen levels in the area. This addition would help users with allergies to avoid cycling during high pollen days, promoting better health and comfort.

Air Quality Comparison

- **User Need:** Users may want to compare air quality data between different routes or locations.
- **Upgrade Feature:** Implement a feature that allows users to compare air quality readings from various routes in real-time. This could help bicyclists choose paths with the best air quality, encouraging healthier riding decisions.

Dynamic Routing Based on Air Quality

- **User Need:** Cyclists may wish to avoid routes with poor air quality.
- **Upgrade Feature:** Create a dynamic routing feature that suggests alternative cycling paths based on current air quality levels. This can help users plan healthier rides while avoiding polluted areas.

Community Feedback and Reporting

- **User Need:** Cyclists often share local knowledge about air quality and road conditions.
- **Upgrade Feature:** Add a community feedback feature where users can report on air quality and environmental factors along their routes. This could create a more engaged user community and provide real-time updates based on user observations.

Fitness and Performance Metrics

- **User Need:** Cyclists are often interested in tracking their fitness performance alongside environmental data.
- **Upgrade Feature:** Incorporate features that allow users to track fitness metrics such as speed, distance, and calories burned, alongside air quality data. This comprehensive approach enables users to monitor both their health and the environmental factors affecting their ride.

By integrating these features, the AirFinder app could significantly enhance its utility for cyclists, promoting not only safer rides but also a deeper understanding of how environmental factors impact their health and well-being.

3.6.4 Conclusion

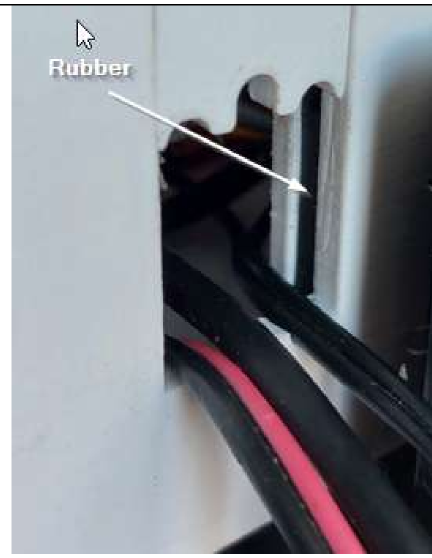
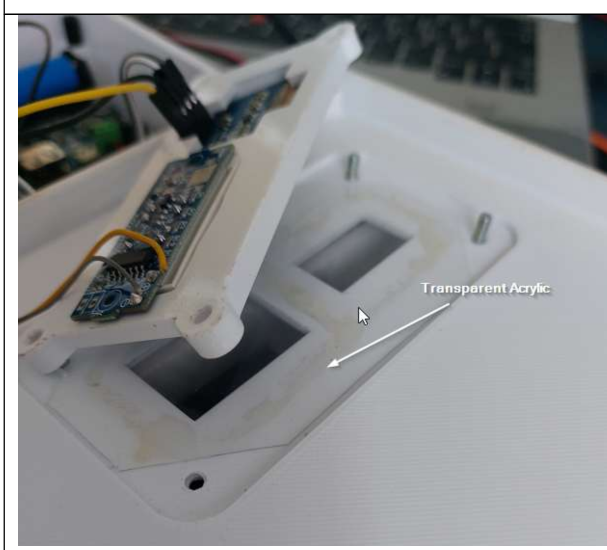
The usability test interviews conducted with the first group revealed that although the AirFinder app is generally easy to use and navigate, there are notable opportunities to enhance the user experience, particularly in the areas of data interpretation and feature accessibility. After addressing these concerns, the app was tested with a significantly larger group, and the results indicated that the app had become more intuitive and valuable for users. This improvement has enhanced its effectiveness in promoting public health through better air quality monitoring.

Chapter 4: Testing, Results and Discussion

4.1. Assembly

4.1.1. Box assembly

Figure 4.1 illustrates ingress protection measures, which involve carving rubber in the plastic enclosure. After much tinkering with the modules and placing them in a smallest area as possible the sensor box is displayed as follows in Figure 4.2.

	
O-rings in top cover	Ruber in cable door
	
Gluded tranparent acrylic in displays	Silicone in cables


Protective door for power switch and emergency charge

Figure 4.1 - Ingress protection measures

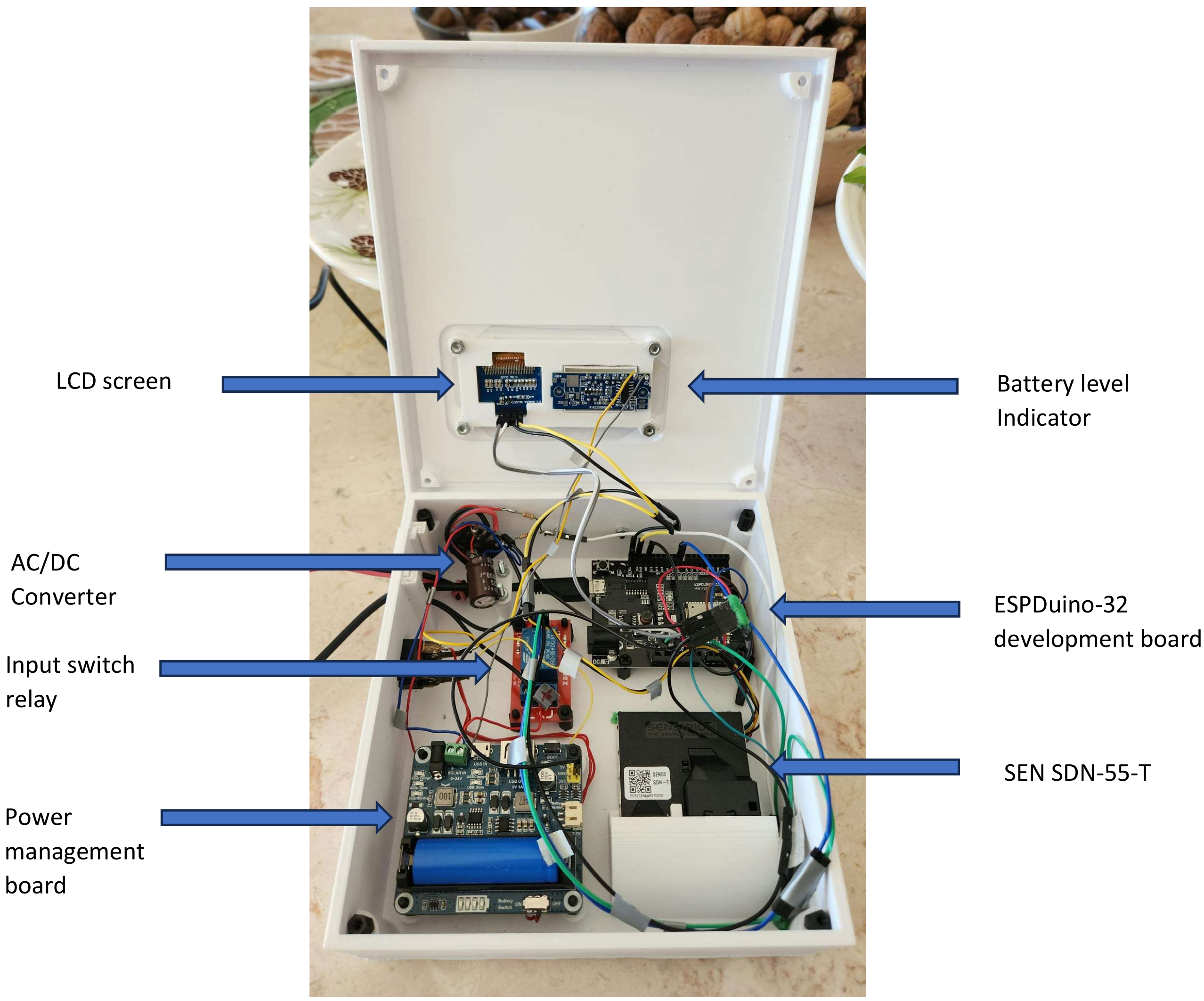


Figure 4.2 - Fully assembled system in enclosure

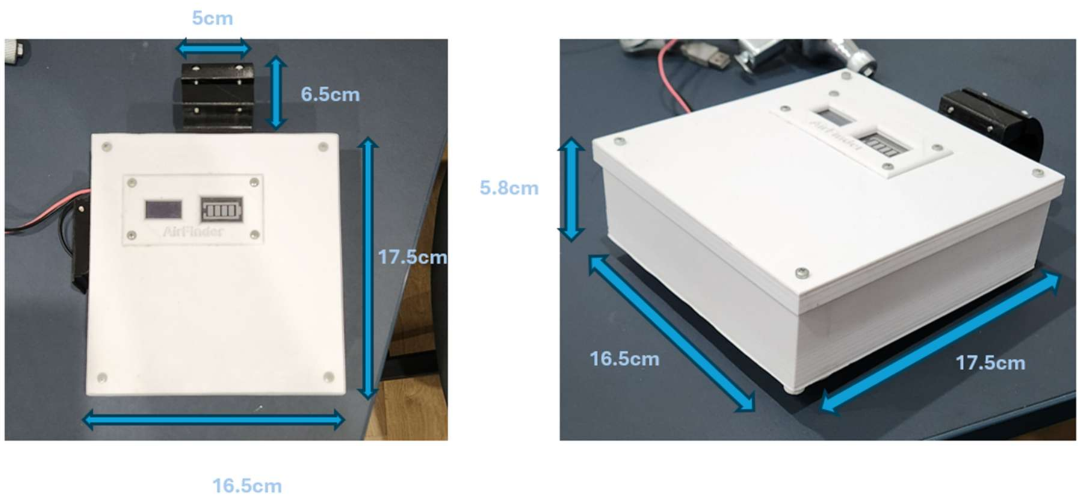


Figure 4.3 - Sensor Box Dimensions

The sensor and LCD screen communicate with the microprocessor unit through I2C, the power management module connects to the microprocessor to supply power to previously mentioned modules, the relay switches the power source (solar panel or bicycle dynamo) depending on the evaluation made from the microprocessor and supplies energy to the power management module. A battery indicator is connected to the power management module to inform the user of the battery level of the sensor box, and a switch is also connected to it, making it able to turn off and on the whole system. The box itself is fully 3D printed using PLA 3D printer filament and was designed with all the modules in mind, having special placements for each module and an air vent for the air quality sensor (Figure 4.4).

The box lid comes with two openings for the LCD and battery module and is fitted with a shield for the modules, so water doesn't seep through. Other than that, the lid was also made so it covers the outlines of the box not letting anything enter the box. The box and lid are screwed having the head in the lid and the body of the screws in the corners of the box. There is also a gate on the side of the box where the solar panel, dynamo and debugging USB cable go in and out from, having the possibility to take them out anytime and still have some sort of external protection.

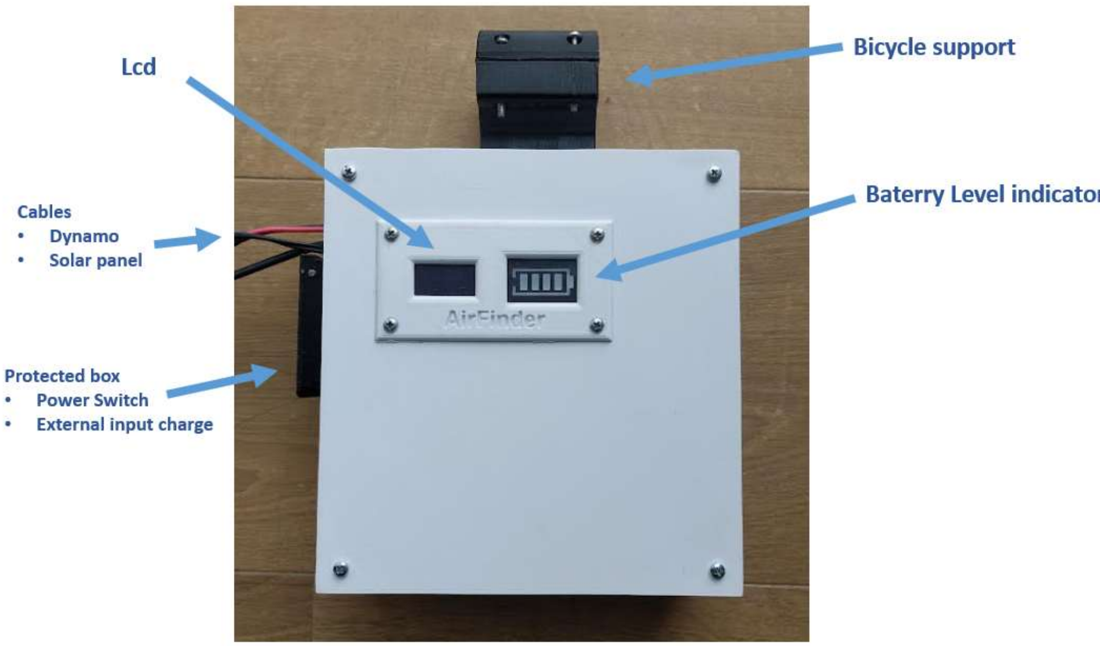


Figure 4.4 - Enclosure design with lid

On the back of the sensor box there is an air vent that allows the air quality sensor to both suck in and spill out the air molecules it needs to evaluate (Figure 4.5), and in Figure 4.6 the same air vent can be seen with the windshield mounted.

Figure 4.5 - Enclosure bottom design without wind protector

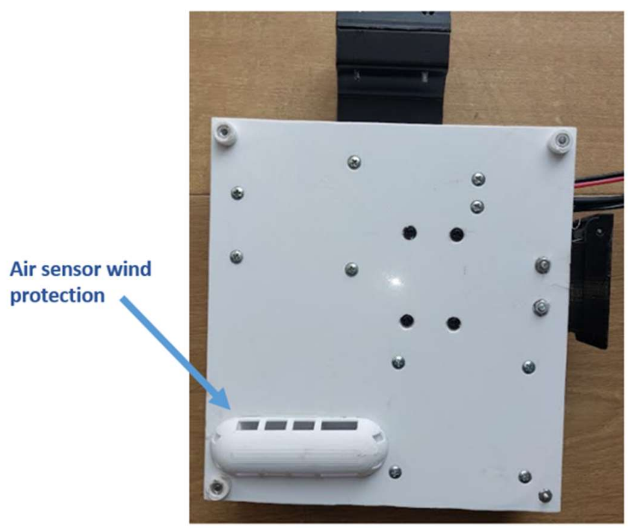
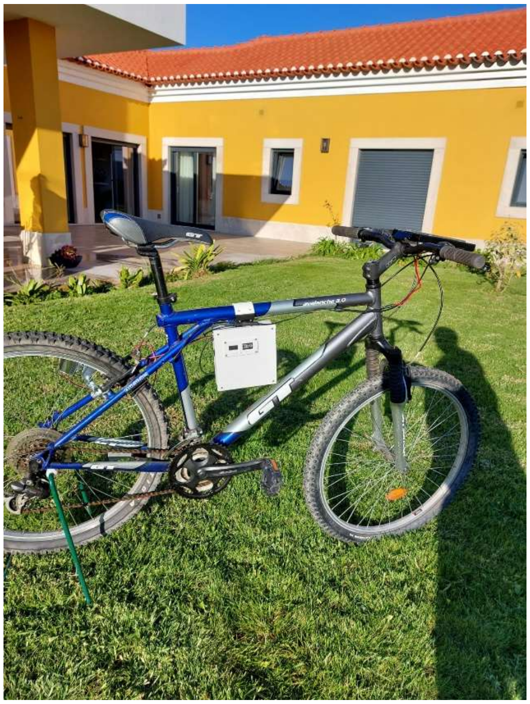
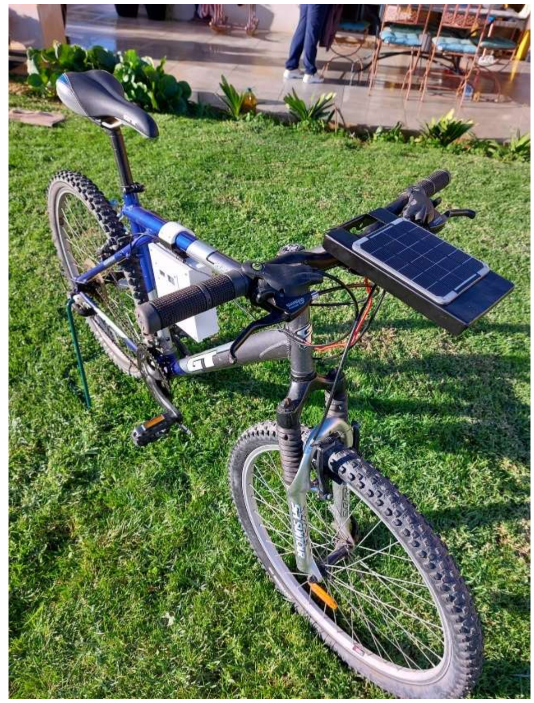


Figure 4.6 - Enclosure bottom with wind protector

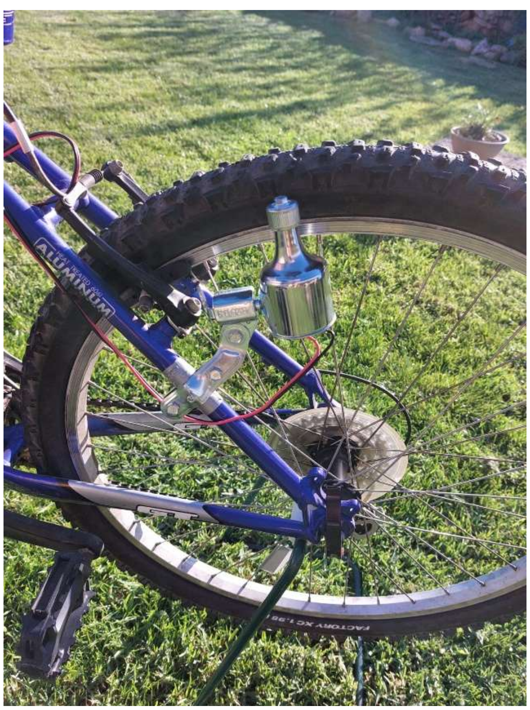
Here is the sensor box integration on the bicycle for testing (Figure 4.7):



Full profile



Solar panel mounted on bicycle handle



Dynamo mounted on back wheel

SensorBox mounted on bicycle frame

SensorBox turned on at night measuring air pollutant values

Figure 4.7 - SensorBox integration with the bicycle

4.1.2 Power consumption

The complete system (the ESPDuino-32 CPU with the air sensor, LCD and a Relay), was powered by a USB cable and the power consumption was measured using a USB Tester (Figure 4.8).

Total power consumption = 150m A * 5.06 V = **759 mW**

Figure 4.8 - System power consumption

Total energy consumption = 0.759 Watt-Hour

This is about 100% of the energy stored in an 800mA Li 3.7V Battery Cell.

$$Energy = Capacity\ (A\ Hour) * \ Voltage = \text{800 mAh * 3.7V = 2.97 Watt Hour}$$

This means that the system can be active for about 1 hour without charge. By activating the solar panel and the dynamo this time can be extended indefinitely.

Considering that the system includes a solar panel and a bicycle dynamo to charge the battery it's reasonable to design the system with a charging controller with this battery capacity. This high-power consumption is due mostly to the ESPDuino-32 development board. In a production scenario this development board would change to a dedicated board with lower power consumption.

3.1.3 Solar Panel

A small size monocrystalline solar panel with a rating of 3W at 6V was considered for this project and its performance evaluated. The total solar surface area is 0.0126 m^2 (Figure 4.9) with a typical efficiency around 18% for this type of panels.

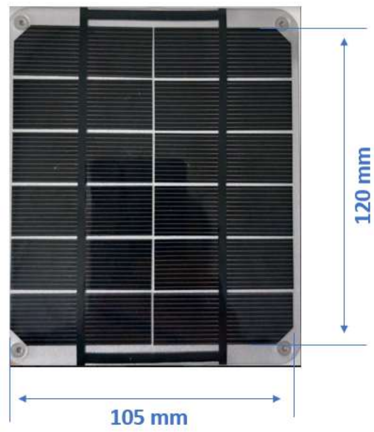


Figure 4.9 - Solar panel dimensions

4.1.3.1 Performance versus solar orientation

Solar orientation is represented by 3 rotational vectorial lines, as can be seen in (Figure 4.10). The solar panel was connected to a resistive load and measurements were taken with different pitch and yaw angles. Results show that pitch angles between 30º and 60º causes a power reduction of less than 10%. Outside this range power loss quickly drops to about 30% power loss (Figure 4.11). The yaw angle was also tested (Figure 4.12), and results show that 0º is the optimum angle and only after angles above 40º, it starts to show a significant loss in power.

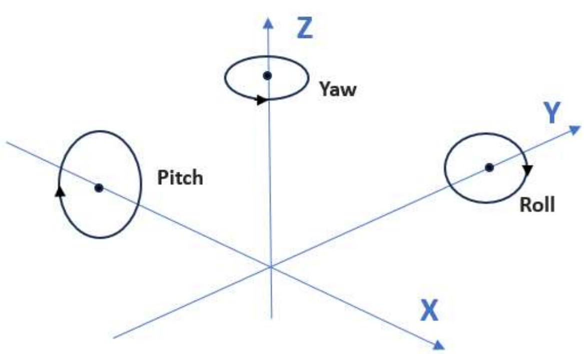


Figure 4.10 - Vectorial representation of Yaw, Pitch and Roll

- Rotation around the front-to-back axis is called **roll**.
- Rotation around the side-to-side axis is called **pitch**.
- Rotation around the vertical axis is called **yaw**.

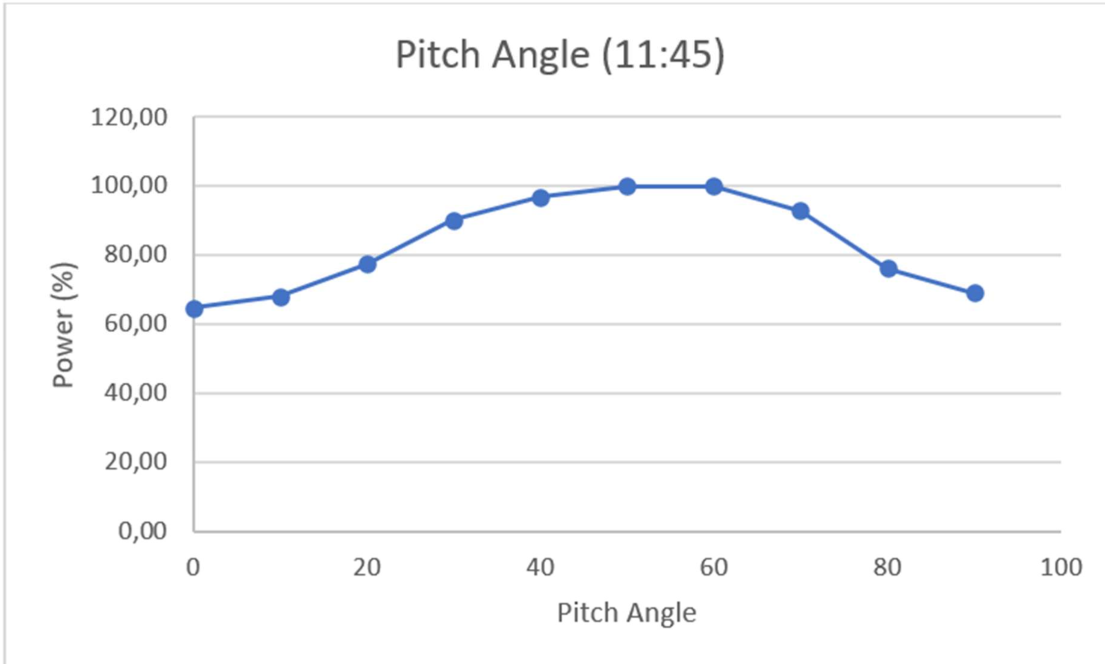


Figure 4.11 - Optimal pitch angle graph

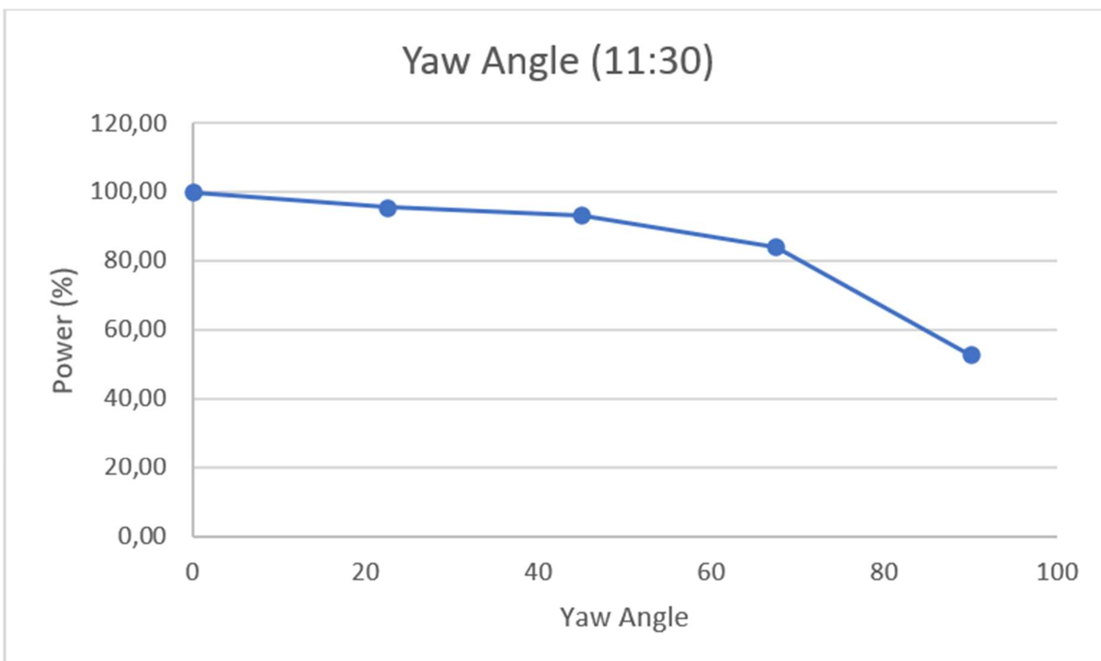


Figure 4.12 - Optimal Yaw angle graph

Considering that the bicycle moves most of the time in the pitch angle and considering the wind pressure, a metal support was fabricated with a 1.8mm metal sheet to strongly support the solar panel. The support was mounted in the bicycle with a pitch angle of 15º and a yaw angle of 0º, position limited by the bicycle brake cables, representing a 30% decrease from optimum power when the bicycle is in a stationary horizontal position (Figure 4.13). While parking this angle can be compensated for by parking in a tilt area.

Figure 4.13 - Mounted Solar panel with special adapter to bicycle handle

4.1.3.2 Electrical performance of a small solar panel

Measurements were taken on a very sunny day at 11am in late September 2023. A group of resistors was used to create different load scenarios. Output voltage and current was measured, and the output power calculated with the formula:

$$Power = \text{Voltage} * \text{Current} \tag{4.1}$$

Output power measurements

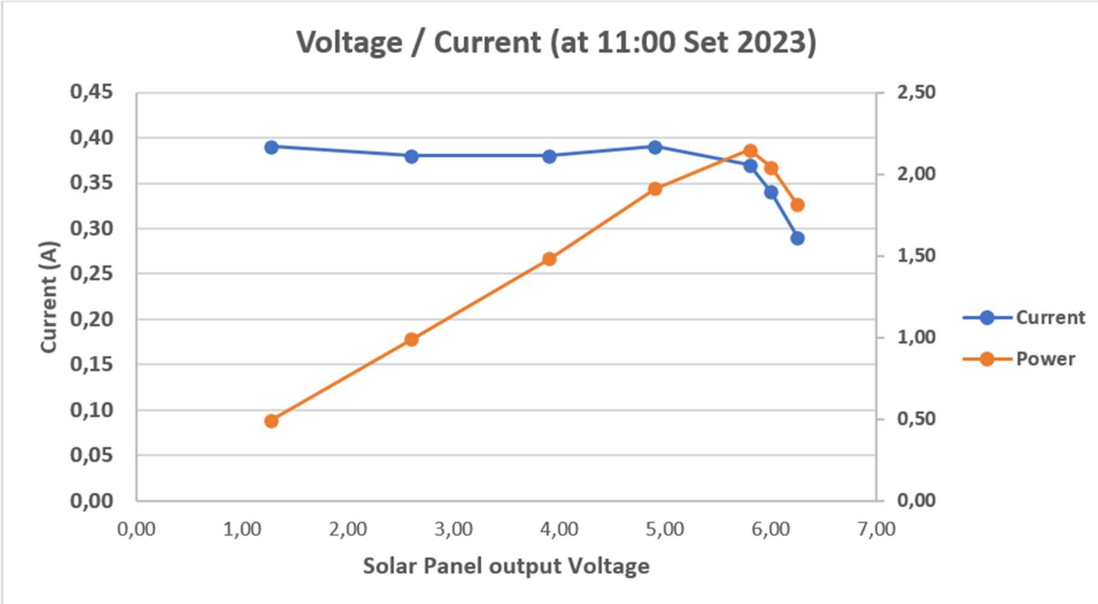


Figure 4.14 - Optimal solar panel voltage / current graph

The results in Figure 4.14 show a maximum output power of 2.04 Watts obtained around 6 Volts; the optimum voltage announced by the manufacturer. The output power of 2.04 Watts is consistent with the expected solar radiation at this time of the year (Late September) and the panel efficiency.

$$Output\ Power = Solar\ radiation\ * Panel\ Eficiency * Panel\ Surface\ Area \qquad (4.2)$$

$$2.04\ w = Solar\ radiation\ * 18\% *\ 0.0126\ m^2$$

$$Solar\ radiation = 899\ w/m^2$$

Note: These values are consistent with the time of the year and location where the measurements were taken.

Considering all these tests the solar panel output measurements are consistent with the manufacturer advertised specifications.

4.1.4 Bicycle Dynamo

4.1.4.1 Dynamo rotation and bicycle speed

To calculate the relation between the dynamo frequency and the bicycle speed 3 instruments were used: a high-speed camera, a multimeter to measure the dynamo output frequency and a tape measuring tool. The formula, as stated before, is:

$$\boldsymbol{Bycicle\ Speed} = \frac{2 * \pi * wheel_{Radius} * \left(\frac{\boldsymbol{Dynamo_{Frequency}}}{\boldsymbol{K}}\right) * 3600}{1000} \left(\frac{Km}{h}\right) \tag{4.3}$$

K is the constant that was calculated from the measurements:

	6V dynamo	12V dynamo
K value	104	91

4.1.4.2 – Measuring Vg

For this measurement a digital oscilloscope was connected to the dynamo output with the circuit open, i.e., with no load connected. Measurements were taken for different wheel rotating frequencies (Figure 4.15). The 12V dynamo is significantly better than the 6V dynamo, for this reason it was chosen for this project.

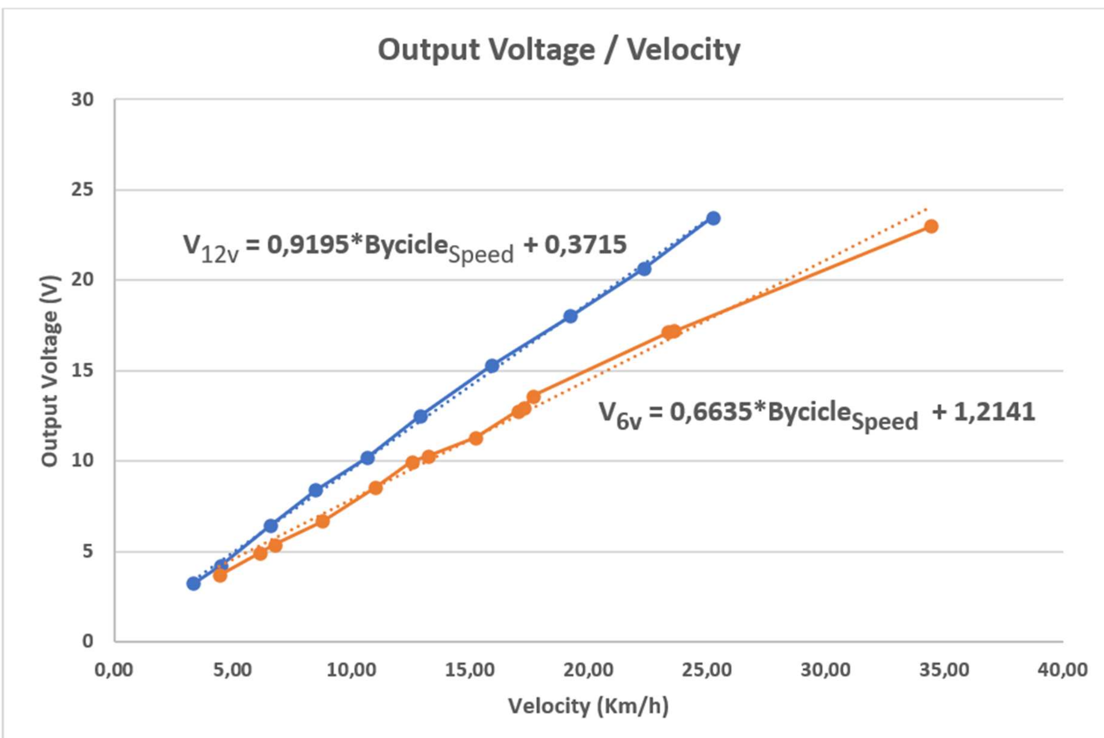


Figure 4.15 - Output voltage to bicycle speed relation

Output voltage has a linear variation with the dynamo frequency which is itself proportional to the bicycle speed. Using linear numeric interpolation, it was possible to obtain the relation between frequency and output voltage.

V_{12v} = 0,9195*Bycicle_{Speed} + 0,3715

The battery management system at its peak performance charges the battery with 1A @ 4.2V (4.2 Watts).

MPPT support (Maximum power point tracking) means that this charger can regulate the current extracted from the input power source to keep the input at its optimum power level. MPPT was developed for solar panels but works with dynamos and other power sources.

The battery management system by design only works when the input voltage is above 5.5V and since the dynamo has an output voltage that is linearly dependent on the bicycle speed, the dynamo output is only used by this system when the bicycle is ridding above 4.5 km/h (accordingly with Figure 4.15).

Using an electric circuit simulator available at [24] and the previous obtained electric model for the dynamo, a simulation was run to estimate the necessary bicycle speed to supply a 4.2W Power (1A @ 4.2V) to the Battery management system (Figure 4.16). This is the maximum power needs to charge the battery with 1A.

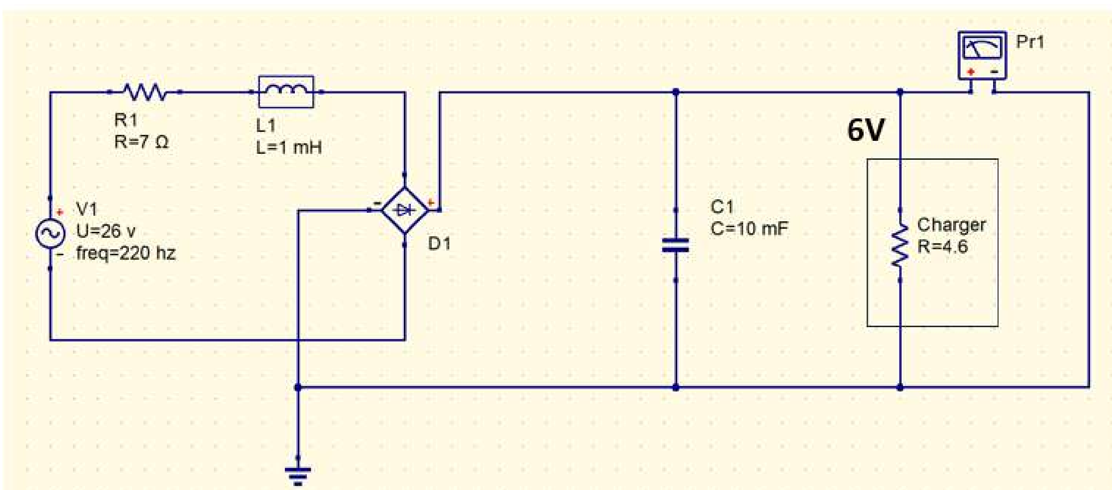


Figure 4.16 - Circuit simulation model of the bicycle dynamo

In the schematic the charger is represented by a resistance of 4.6 Ω

$$R = \frac{V^2}{\text{Power}} \tag{4.4}$$

$$R = \frac{6^2}{4.2} = 4.6\ \Omega$$

The simulation showed that to obtain 6V at the charger input (The MPPT target voltage) a 26 V amplitude (18V Rms) is needed at the dynamo. Using the previously obtained formula for the dynamo:

$$\boldsymbol{Dynamo\ Output_{RMS\,Voltage} = 0{,}9195 * \ Bycicle_{speed} + 0.3715\ (Volts)}$$

It's possible to estimate the necessary bicycle speed:

$$\boldsymbol{Bycicle_{speed} = \frac{Dynamo\ Output_{RMS\,Voltage} - 0.3715}{0{,}9195}} = \textbf{19.17 Km/h}$$

4.1.4.3 Dynamo integrated tests

To measure the performance of the dynamo integrated with the sensor box, a logging device was used to register the power delivered to the sensor box in a real scenario (Figure 4.17).

This device gathers the voltage generated and the current delivered to the power management module, and with those measurements, calculates the power being delivered. This device also includes in the collected data of the geo coordinates and bicycle speed read from a GPS module. All data is stored on an SD card.

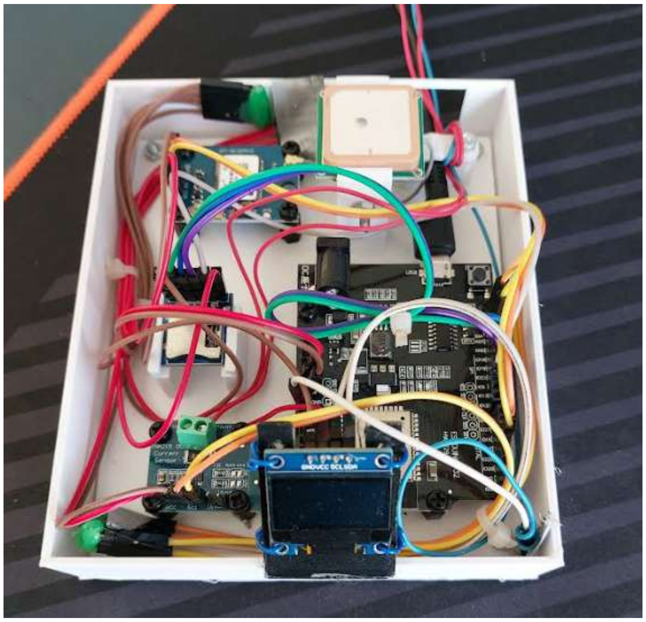


Figure 4.17 - Power delivery logging device

This device is composed of 5 modules: an ESP32 development board, an I2C LCD Screen, a GPS module and a current and voltage sensor module. The device connects to the sensor box as shown in Figure 4.18:

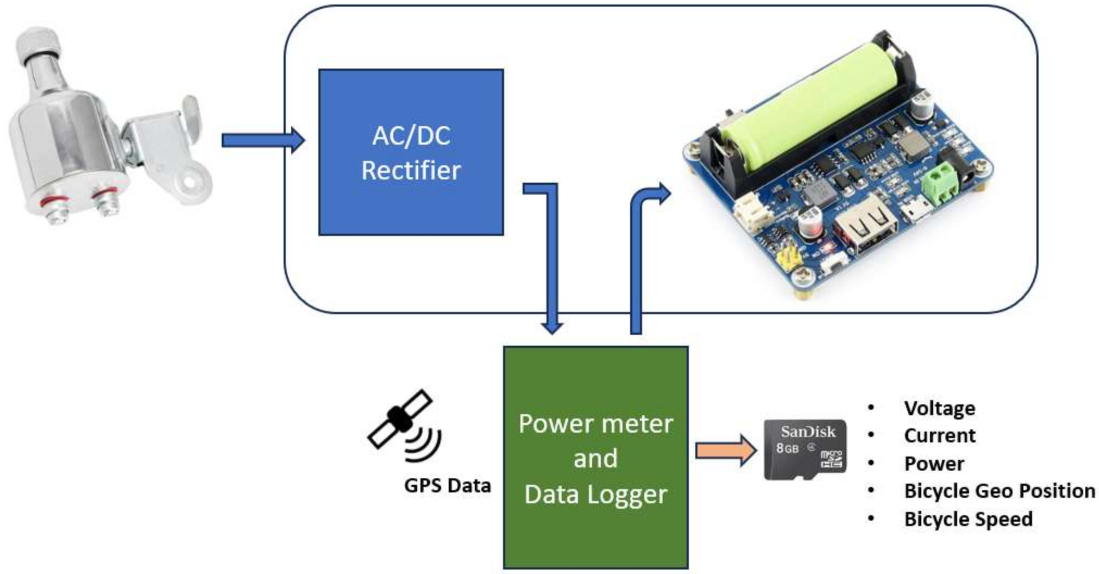


Figure 4.18 - Logging device integration into the system

4.2. Testing

4.2.1 – Pilot test

A countryside route of 1.1km, fluctuating a lot in elevation, was selected for this field test (Figure 4.19). This course is perfect to test a vast range of speeds, and the respective power generated by the dynamo.

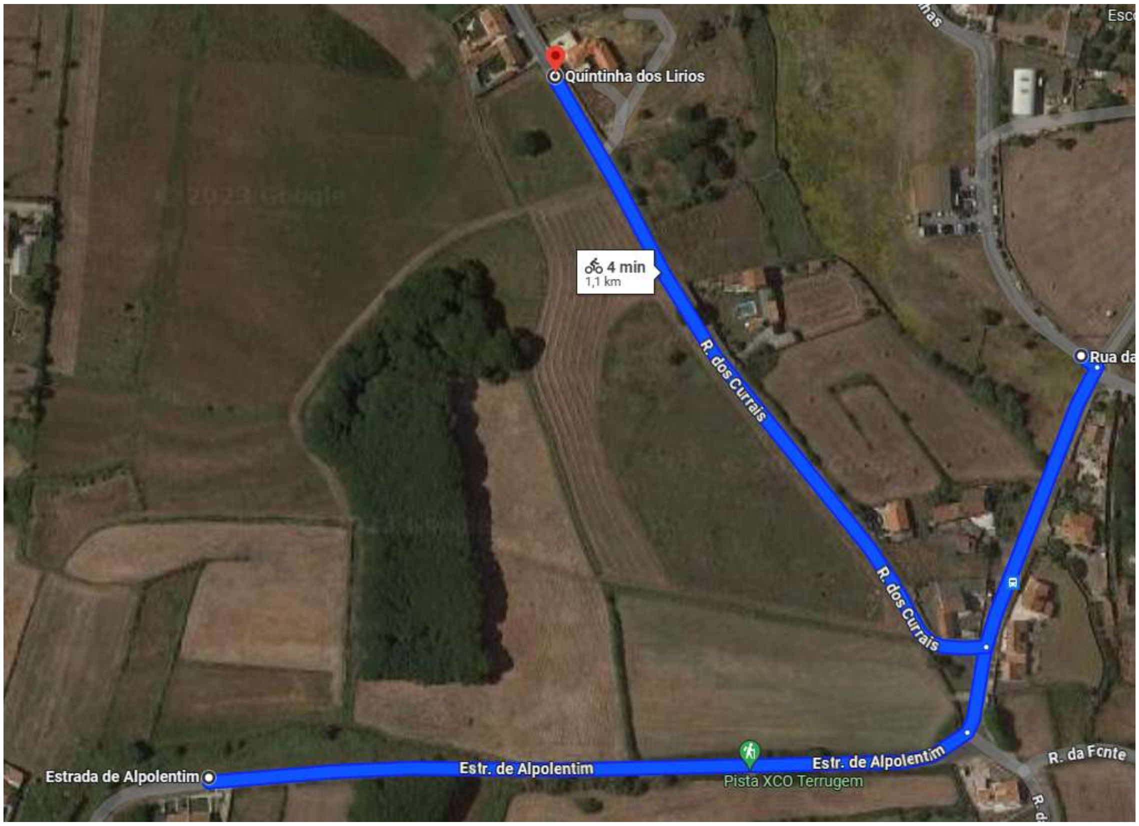


Figure 4.19 - Field test route

After completing the route above the following data was extracted from the SD card and processed.

Figure 4.20 shows the voltage and power delivered to the Power management module during the duration of the course.

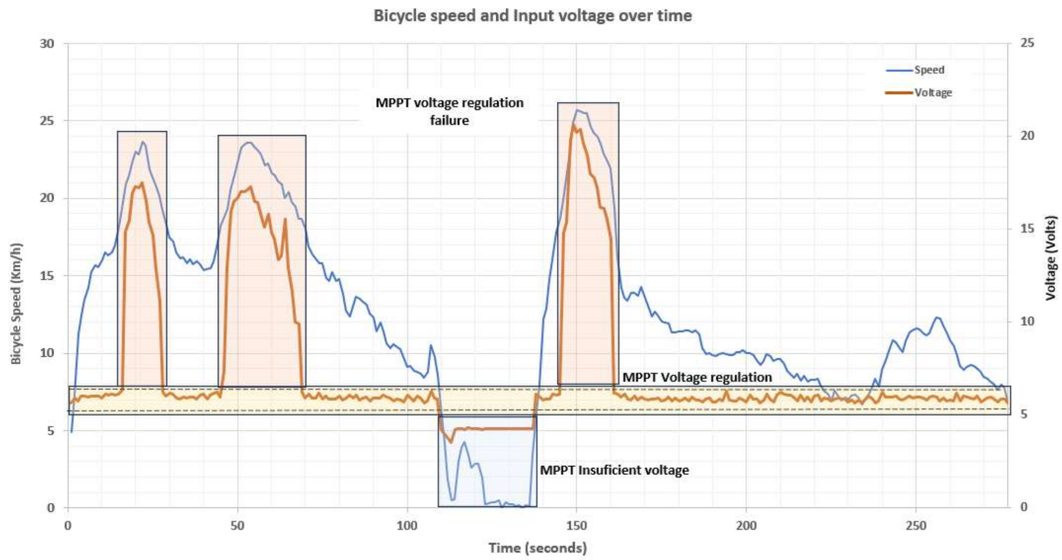


Figure 4.20 - Voltage regulation with power management module

First and foremost, as referred to earlier the Power management module is an MPPT regulator designed to extract the maximum power from the solar panel, in this case configured for 5.7v. The voltage graph shows that this device works as expected when the bicycle speed is between 5 km/h and 18km/h. Below this range no energy is collected (the observed 5 volts level is due to the capacitor still holding voltage but the charging module is not working) and above it the battery management system is over charged and no longer regulating the input voltage, although it keeps collecting energy.

Figure 4.21 shows the power delivered and bicycle speed over time. Below 5 km/h no energy is delivered to the MPPT as expected (there was a small peak of energy when the bicycle stopped due to the capacitor still delivering energy). Above this, a high performance is observed. For a normal bicycle cruise speed of about 12km/h a 3-Watt delivery is observed, which is very good.

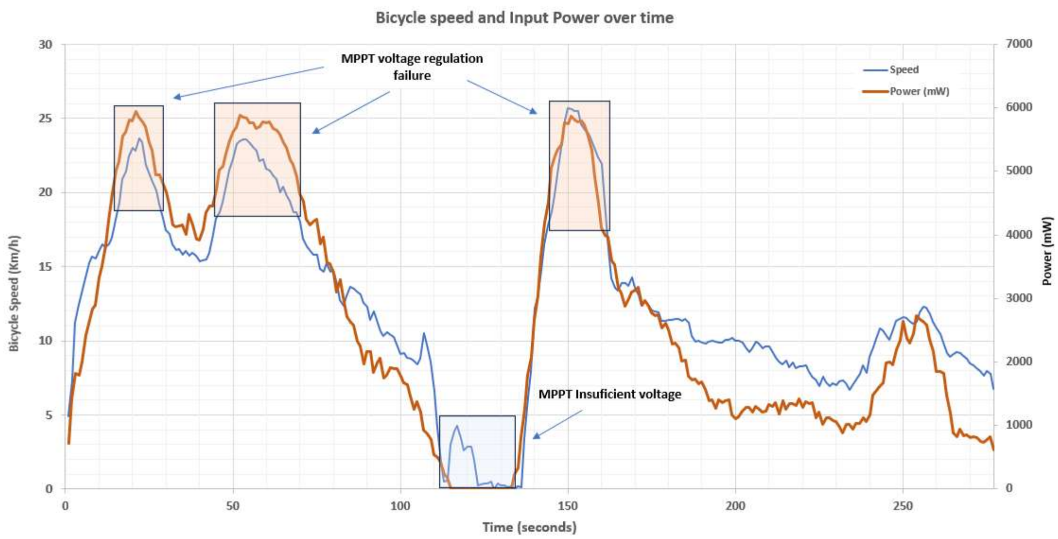


Figure 4.21 - Power regulation with power management module

Figure 4.22 plots the relation between power delivered and bicycle speed.

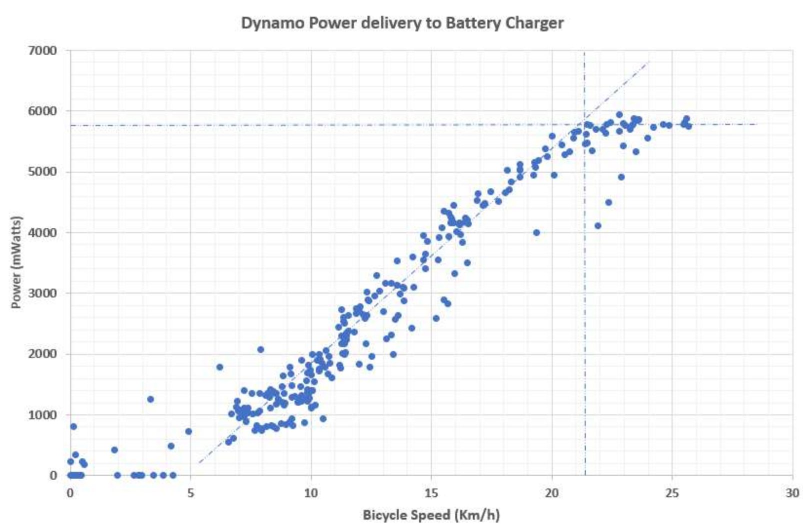


Figure 4.22 - Optimal power delivery relation to bicycle speed

Until 4-5 km/h the energy delivered is practically 0, from 5 to 18 km/h (normal bicycle speeds), the power delivered increases linear with the bicycle speed, and from 20 km/h onwards the power delivery starts to stagnate (stagnating at 22 km/h).

Lastly, Figure 4.23 shows current delivered samples plotted over voltage. This shows the MPPT Algorithm working to keep the voltage regulate around.

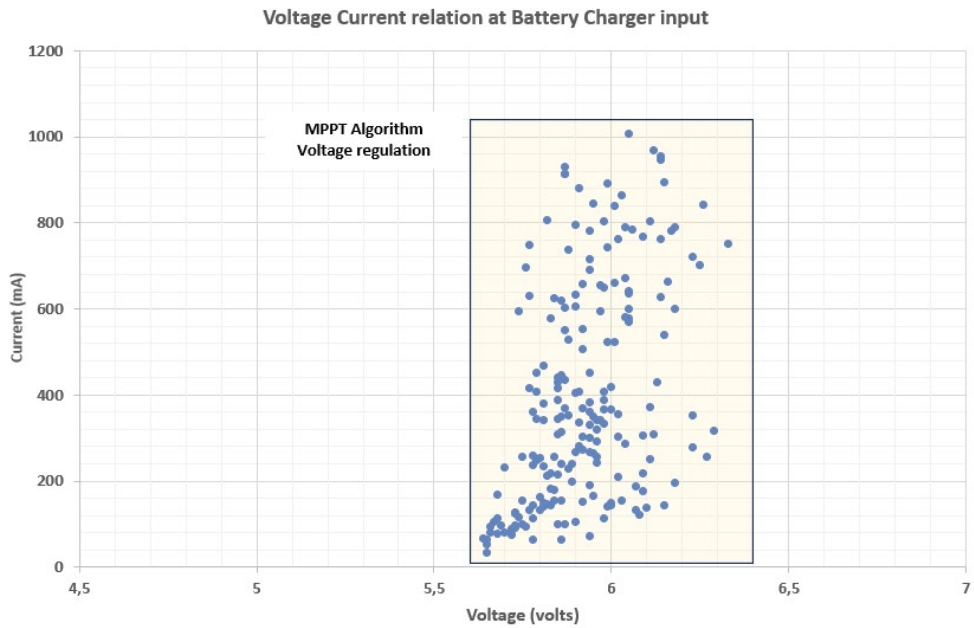


Figure 4.23 - MPPT voltage regulation at different current levels

4.2.2. Sensor measurement comparative tests

The objective of this work is not to evaluate the quality of the sensors used; however, whenever possible, a simple comparison between the measured values and those provided by official measurement stations can be performed.

The complete system was used to measure the air quality in two paths at different locations:

- R Filipe Da Mata, Lisbon, Portugal, 13th July 2024.
- CC Arroios, R Filipe Da Mata, Lisbon, Portugal, 13th July 2024.

These locations were selected because they have an air quality measurement station installed nearby:

- At R Filipe Da Mata (Figure 4.24), a monitoring station provided by European Environment Agency and managed by Agencia Portuguesa do Ambiente.
- At CC Arroios (Figure 4.25), a monitoring station installed in a public school.

Figure 4.24 - Path taken in R Filipe da Mata

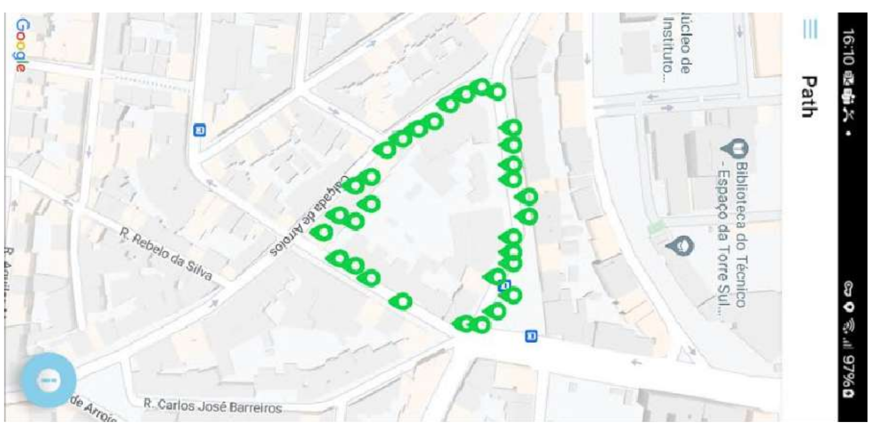


Figure 4.25 - Path taken in Arroios

Table 4.1 and Table 4.2 show the comparative values measure using the AirFinder and the local Measurement Stations. The values of PM10 show a significant difference from those measure from the local stations. The discrepancy can be explained by the fact that in the SEN55-SDN-T sensor, the PM10 value is not measured, but calculated based on PM1 and PM2.5 values using a typical particle distribution profile. The PM2.5 values are very close, and the difference can be explained by the fact the measurements were not taken side by side.

Table 4.1 - Comparative values at Praça de Espanha

	Praça de Espanha	AirFinder	Delta
PM2.5	13,9	14,05	1,07%
PM10	10,1	14,48	30,25%

Table 4.2 - Comparative values at Arroios

	Escola Básica Leão Arroios	AirFinder	Delta
PM2.5	8,3	9,3	10,75%
PM10	5,8	9,5	38,95%

Chapter 5: Conclusion and future work

5.1. Conclusion

This thesis aimed to design a low-cost, autonomous air quality monitoring system integrated into a bicycle, powered by green energy, and capable of real-time data collection and visualization through a mobile application. Each of the initial research questions has been addressed through the findings of this work:

Power Autonomy: Through the integration of green energy sources, such as a solar panel and bicycle dynamo, the system demonstrated partial power autonomy. Results showed that while power generation from these sources was effective in maintaining system operation, enhancing autonomy could be achieved by increasing the solar panel area. The combination of solar and kinetic energy sources reflects the system's commitment to sustainable, eco-friendly solutions.

Real-time Data Accuracy and Mobility: The system's architecture, utilizing the ESP32 microcontroller with Bluetooth Low Energy (BLE), enabled accurate, real-time collection of air quality data. This data was effectively visualized on the user's mobile device through the AirFinder application. User tests confirmed that the data provided was both reliable and valuable for cyclists, empowering them to make informed decisions about their routes based on current air quality conditions.

Data Visualization and Usability: The AirFinder app successfully visualized air quality data through a clear and interactive map interface, allowing users to view pollutant levels in real-time. Usability testing confirmed that users found the app accessible and informative, with the potential to raise awareness about environmental conditions. The app's capability to map air quality along a cyclist's path provides actionable insights into urban air pollution.

Affordability and Replicability: The system met its affordability goal by using low-cost components, such as the Sensirion SEN55 sensor and ESP-32 microcontroller. These choices make the system economically viable for broad deployment, accessible to individuals and communities, and promising for future expansion.

Overall, this project demonstrated the feasibility of creating an effective, sustainable, and accessible air quality monitoring solution. While there were limitations, such as dependence on environmental conditions for power autonomy, the system's potential for urban and personal use is promising, paving the way for future enhancements and broader applications.

5.2. Future work

In future iterations of this project, there are several areas for enhancement, particularly focusing on IT and system functionality aspects. These enhancements would improve data processing, user accessibility, and system capabilities, making the device even more robust and practical for real-world applications.

Database Enhancements:

Developing a centralized database would allow remote data storage, enabling users to save and access their data across devices and potentially share pollution data with others. This database could be hosted on a cloud-based service, supporting broader analysis and fostering a community-driven approach to monitoring urban air quality. The incorpored timestamped entries for every data point collected would enable more granular tracking and historical analysis of air quality levels, rather than just session-based timestamps.

Mobile Application Expansion:

Expanding the app to include support for iOS devices would enhance accessibility, enabling a wider audience to benefit from the system's functionalities. Given the importance of public health data, cross-platform compatibility would be a significant advancement. Integrating real-time weather data through APIs would allow users to consider weather conditions' impact on air quality along their routes. This additional layer of information would help users make even more informed decisions. Implementing a heat-map based on air quality index and a dynamic routing feature that suggests cleaner paths in real-time based on current pollution levels would further enhance the app's usability. This feature could guide cyclists toward safer routes, reducing their exposure to pollutants.

Advanced Features for Data Analysis:

Developing comparative analytics capabilities within the app would allow users to analyze air quality across multiple routes or over time. This could help cyclists compare different paths or assess the effectiveness of their chosen routes for minimizing pollution exposure. Integrating pollen count monitoring would add value to users with specific health needs, particularly those with respiratory conditions or allergies. This feature could make the app even more beneficial, providing a comprehensive picture of environmental quality.

These proposed advancements would further strengthen the system's utility, enhancing its effectiveness as a tool for real-time air quality monitoring and public health awareness. Future development will focus on leveraging IT solutions to increase accessibility, improve data management, and expand the app's functionality, ultimately making air quality information more actionable and beneficial for the public.

System Performance:

Reducing power consumption and increasing efficiency are crucial goals. The project would significantly benefit from implementing a dual-channel MPPT controller, enabling simultaneous charging from both the solar panel and the dynamo. Introducing an active rectifier for the dynamo would further reduce power loss at the rectifier diodes. Utilizing a custom-made ESP32 board designed for low power consumption could also lead to considerable energy savings. Additionally, to optimize power usage, the LCDs providing user information (such as measurements and battery charge status) could be activated only when needed, for example, by pressing a switch. The sensor box could benefit from the implementation of a cooling fan, as it would reduce the overall internal temperature of the box and all the integrated components, and its speed could be controlled to preserve power consumption.

References

[1] A. J. Cohen, M. Brauer, R. Burnett, H. R. Anderson, J. Frostad, K. Estep, K. Balakrishnan, B. Brunekreef, L. Dandona, R. Dandona, V. Feigin, G. Freedman, B. Hubbell, A. Jobling, H. Kan, L. Knibbs, Y. Liu, R. Martin, L. Morawska, C. A. Pope, H. Shin, K. Straif, G. Shaddick, M. Thomas, R. Van Dingenen, A. Van Donkelaar, T. Vos, C. J. L. Murray and M. H. Forouzanfar, "Estimates and 25-year trends of the global burden of disease attributable to ambient air pollution: an analysis of data from the Global Burden of Diseases Study 2015," *The Lancet,* vol. 389, p. 1907–1918, May 2017.

[2] J. De Bont, S. Jaganathan, M. Dahlquist, Å. Persson, M. Stafoggia and P. Ljungman, "Ambient air pollution and cardiovascular diseases: An umbrella review of systematic reviews and meta-analyses," *Journal of Internal Medicine,* vol. 291, p. 779–800, June 2022.

[3] R. Munsif, M. Zubair, A. Aziz and M. Nadeem Zafar, "Industrial Air Emission Pollution: Potential Sources and Sustainable Mitigation," in *Environmental Emissions*, R. Viskup, Ed., IntechOpen, 2021.

[4] T. S. Gunawan, Y. M. Saiful Munir, M. Kartiwi and H. Mansor, "Design and Implementation of Portable Outdoor Air Quality Measurement System using Arduino," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 8, p. 280, February 2018.

[5] J. Gómez-Suárez, P. Arroyo, R. Alfonso, J. I. Suárez, E. Pinilla-Gil and J. Lozano, "A Novel Bike-Mounted Sensing Device with Cloud Connectivity for Dynamic Air-Quality Monitoring by Urban Cyclists," *Sensors,* vol. 22, p. 1272, February 2022.

[6] A. Samad and U. Vogt, "Investigation of urban air quality by performing mobile measurements using a bicycle (MOBAIR)," *Urban Climate,* vol. 33, p. 100650, September 2020.

[7] European Commission. Joint Research Centre., Review of sensors for air quality monitoring., LU: Publications Office, 2019.

[8] X. N. Bui, C. Lee, Q. L. Nguyen, A. Adeel, X. C. Cao, V. N. Nguyen, V. C. Le, H. Nguyen, Q. T. Le, T. H. Duong and V. D. Nguyen, "Use of Unmanned Aerial Vehicles for 3D topographic Mapping and Monitoring the Air Quality of Open-pit Mines," *Inżynieria Mineralna,* vol. 2, January 2022.

[9] S. Duangsuwan and P. Jamjareekulgarn, "Development of Drone Real-time Air Pollution Monitoring for Mobile Smart Sensing in Areas with Poor Accessibility," *Sensors and Materials,* vol. 32, p. 511, February 2020.

[10] T. Villa, F. Gonzalez, B. Miljievic, Z. Ristovski and L. Morawska, "An Overview of Small Unmanned Aerial Vehicles for Air Quality Measurements: Present Applications and Future Prospectives," *Sensors,* vol. 16, p. 1072, July 2016.

[11] A. Hilary Kelechi, M. H. Alsharif, C. Agbaetuo, O. Ubadike, A. Aligbe, P. Uthansakul, R. Kannadasan and A. A. Aly, "Design of a Low-Cost Air Quality Monitoring System Using Arduino and ThingSpeak," *Computers, Materials & Continua,* vol. 70, p. 151–169, 2022.

[12] Y. Güven, E. Coşgun, S. Kocaoğlu, H. GeziCi and E. Yilmazlar, "Understanding the Concept of Microcontroller Based Systems To Choose The Best Hardware For Applications".

[13] P. Santana, A. Almeida, P. Mariano, C. Correia, V. Martins and S. M. Almeida, "Air quality mapping and visualisation: An affordable solution based on a vehicle-mounted sensor network," *Journal of Cleaner Production,* vol. 315, p. 128194, September 2021.

[14] M. S. Halper, "Supercapacitors: A Brief Overview," 2006.

[15] A. Asrori, F. Fatur, E. Faizal and M. Karis, "The Design and Performance Investigation of Solar E-Bike using Flexible Solar Panel by Different Battery Charging Controller," vol. 10, p. 14431–14442, September 2020.

[16] S. Yashas and E. W. I. of Technology, "Solar and Dynamo Bike," *International Journal of Engineering Research and,* vol. V9, p. IJERTV9IS090133, September 2020.

[17] B. Elen, J. Peters, M. Poppel, N. Bleux, J. Theunis, M. Reggente and A. Standaert, "The Aeroflex: A Bicycle for Mobile Air Quality Measurements," *Sensors,* vol. 13, p. 221–240, December 2012.

[18] J. A. Afonso, F. J. Rodrigues, D. Pedrosa and J. L. Afonso, "Automatic Control of Cycling Effort Using Electric Bicycles and Mobile Devices," 2015.

[19] V. Samosuyev, "BLUETOOTH LOW ENERGY COMPARED TO ZIGBEE AND BLUETOOTH CLASSIC," 2010.

[20] Sensirion, "Mechanical Design and Assembly Guidelines SEN5x," [Online]. Available: https://sensirion.com/media/documents/546FBC5B/61E9586E/Sensirion_Mechanical_Design_and_Assembly_Guidelines_SEN5x.pdf. [Accessed 17 5 2024].

[21] "FreeCAD: Your own 3D parametric modeler," FreeCAD, [Online]. Available: https://www.freecadweb.org. [Accessed 27 6 2024].

[22] "OpenFOAM - The Open Source CFD Toolbox," OpenFOAM, [Online]. Available: https://www.openfoam.com. [Accessed 28 6 2024].

[23] "ParaView - Open Source Scientific Visualization," ParaView, [Online]. Available: https://www.paraview.org. [Accessed 28 7 2024].

[24] "Online circuit simulator & schematic editor," CircuitLab, [Online]. Available: https://www.circuitlab.com. [Accessed 26 4 2024].