AdvTech AirPollution

DESCRIPTION OF SOME SENSORS KITS AND LABORATORY ACTIVITIES

This material was funded by the European Commission within the Erasmus+ project
Applying some advanced technologies in teaching and research, in relation to air pollution
Project Code: 2021-1-RO01-KA220-HED-000030286

Funded by the European Union

University of Craiova

Paisii Hilendarski University of Plovdiv

Adana Alparslan Türkeş Science and Technology University

Matej Bel University, Banská Bystrica

Authors:
Mihaela Tinca Udristioiu
Silvia Puiu
Silvia Galoi
Rumen Popov
Martin Hruska
Radu Motisan
Silviu Constantin Sararu
Iulian Petrisor
Ion Buligiu
Tugce Pekdogan

Editor:
Mihaela Tinca Udristioiu

ISBN 978-619-7663-82-2 (print)
ISBN 978-619-7663-83-9 (web)

TABLE OF CONTENTS

INTRODUCTION

This section was written by Mihaela Tinca Udristioiu, Silvia Puiu and Silvia Galoi from the University of Craiova, Romania

Four universities – University of Craiova (UCv), Paisii Hilendarski University of Plovdiv (PU), Matej Bel University of Banská Bystrica (UMB), and Adana Alparslan Türkeş University of Science and Technology (ATU) are partners in an Erasmus+ project "Application of advanced technologies in teaching and research in relation to air pollution." In this framework, implementation team members collaborated and shared experiences in preparing STEM students. A set of laboratory activities helpful to students and faculty is presented in this book. New technologies, innovation, and entrepreneurship must be incorporated into the education of students to help them, upon graduation, to adapt to a challenging work environment and to access well-paid jobs. Moreover, students must be sensitized to environmental issues, understand how pollution affects their health, and know the elements of legislation in environmental protection.

This lab activities book shares good practices and some ideas of faculty who teach students in engineering majors and science from the four universities – UCv, PU, UMB, and ATU on how to use a sensor kit to make, program, connect, and transmit data to an extensive independent network of sensors. Also, this laboratory book will present how students can visualize the data produced by each sensor, how it can be downloaded from the database, and how it can be analysed and used. Practical skills and competencies for experimental work are essential for future engineers and scientists. At the same time, graduates should have an entrepreneurial perspective and know some existing environmental protection laws. For this reason, two different points of view regarding sustainable business and respecting the law are presented in the following paragraphs.

Recently, interest in developing entrepreneurial initiatives has grown quite a bit. The focus on sustainability has helped many people to turn their business ideas into ones with the potential to solve socio-economic problems. According to the Cambridge Academic Dictionary Content (n.d.), an entrepreneur is "a person who tries to make a profit by starting a company or operating alone in the business world, especially when it involves taking risks." From the previous definition, we note that the main determining factor for an entrepreneur is the desire to make a profit, even with the inherent risks. In addition to this main factor, entrepreneurs are also motivated by other factors such as the desire to be financially independent, the desire to develop and prove that they can create a profitable business, the willingness to cover unsatisfied needs in particular market niches for themselves or other people; the desire to create a family business involving other family members; the desire to have a safety net, especially in times of crisis when finding a job can prove difficult; the desire to be in control and not be subordinate to other people; the desire to create social businesses that could have a positive impact in the community, helping people and managing issues such as climate change and pollution. Stephan et al. (2015, p. 5) conducted a study on motivations to become an entrepreneur in the UK, and their results showed that "autonomy," "family," and "flexibility" are other essential motivation factors that could be considered a consideration in addition to the traditional ones related to financial motivations. Another critical aspect of their study is that businesses motivated by the desire to support the family and have more

autonomy and freedom have "a greater chance of survival." Carter et al. (2003, p. 13) consider that the most important reasons for starting a business are "personal fulfilment, financial success, roles, innovation, recognition, and independence."

Sustainability means "meeting the needs of the present without compromising the ability of future generations to meet their own needs" (United Nations, 1987). In this context, we can see that both entrepreneurship and sustainability are about meeting needs. Thus, a new business concept appears, especially sustainable businesses, which consider their impact on the community and focus on reducing negative consequences and creating positive changes that could solve some of the community's problems. The Economic and Social Council of the United Nations (n.d.) – the Economic and Social Council of the United Nations emphasizes the three pillars of sustainability: the social pillar focused on people and their well-being; the economic pillar focused on economic growth and making a profit; and the environmental pillar focused on protecting the planet and mitigating climate change.

In 2015 the United Nations established 17 Sustainable Development Goals (SDGs) as part of the 2030 Agenda. Member countries have set goals to be achieved by 2030. These goals are: SDG1- "No poverty"; SDG2 – "Zero hunger"; SDG3 – "Good health and well-being"; ODG4 – "Quality education"; SDG5 – "Gender equality"; SDG6 – "Clean water and sewage"; ODG7 – "Affordable and clean energy"; ODG8 – "Decent work and economic growth"; ODD9 – "Industry, innovation and infrastructure"; ODG10 – "Reduced inequalities"; SDG11 – "Sustainable cities and communities"; ODG12 – "Responsible consumption and production"; ODG13 – "Climate action"; ODG14 – "Life under water"; ODD15 – "Life on land"; SDG16 – "Peace, justice and strong institutions"; ODG17 – "Partnerships to achieve objectives". Individuals, businesses, and public authorities can act on the above goals to create a more sustainable world for people and the planet, now and in the future, for generations to come. These SDGs are meant to solve many problems in the world related to eradicating poverty and hunger, protecting the environment, and creating equal rights so that everyone can grow. Looking closely at the 17 SDGs, we notice that many can be linked to protecting the environment, reducing the carbon footprint, and creating equal opportunities and rights. For example, by reducing waste and implementing the principles of circular economy (nothing is wasted, everything is transformed), hunger and poverty (SDG1 and SDG2) can be at least diminished and, hopefully, shortly, eradicated. SDG6, SDG7, SDG12, SDG13, SDG14, and SDG15 are all goals related to climate change and the need to protect the environment (air, land, and water). These goals can be achieved through many local, national, regional, or global business ideas and initiatives.

The European Parliament (2023) defines the circular economy as "a model of production and consumption, which involves sharing, leasing, reusing, repairing, refurbishing and recycling existing materials and products as long as possible. In this way, the life cycle of products is extended". The advantages of creating a circular economy and also promoting it in the community are significant: less waste, increased efficiency, savings for consumers, reduced costs in the long term, new jobs for the specifics of this economy; a solution to the scarcity and the high price of some raw materials; developing new industrial sectors and thus contributing to a raise of the gross domestic product.

Air, water, and land pollution are some of the most critical problems that affect not only future generations but also everyone that lives today because of the many health problems they create (World Health Organization, 1982; Rodrigues and Römkens, 2018; Righi et al., 2005; Little, 2003; Kampa and Castanas, 2008; European Environment Agency, 2022). Breathing polluted air and eating food grown on contaminated lands or in polluted waters are aspects we

cannot escape. We should recognize their magnitude and do everything possible to reduce the negative impact we create as individuals or businesses. And here comes the role of sustainable companies that should be encouraged by consumers who might orient their consumption habits toward responsible companies. Educational campaigns for raising awareness in the community should be developed by the governments and by private initiatives (NGOs, businesses).

According to the European Environment Agency (2023), 238,000 deaths in 2020 could be attributed to air pollution in Europe. 2020 is a year in which all the restrictions imposed during the lockdown (March-May, 2020) diminished pollution worldwide, as shown in Figure 1. The previous year had 364,200 deaths (European Environment Agency, 2021).

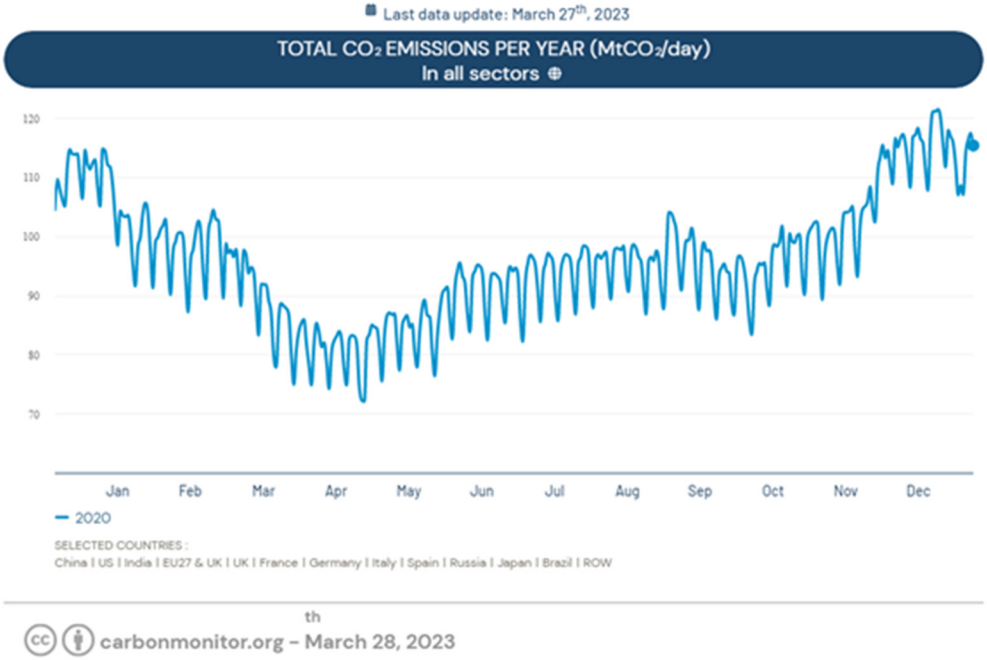


Figure 1. *Evolution of CO_2 emissions in 2020*
Source: Carbon Monitor (https://carbonmonitor.org/)

According to the European Environment Agency (2019), air pollution is essential in health issues such as cardiovascular diseases, lung cancer, anxiety, inflammation, allergies, asthma, and reproductive problems. Considering these issues created by air pollution, the government and private actors created sensor networks to monitor air quality and thus take the appropriate measures. The following section will discuss other private initiatives focused on sustainable businesses that contribute to a cleaner and healthier world, considering the environment, people, and profitability (the three pillars of sustainability), with the present and the future at their core. The advantages of sustainable businesses and their role for the community at large consist of increased competitive advantages, better performance, and solutions to some of the problems created by a crisis like the COVID-19 pandemic (Mattera et al., 2022). The benefit of creating competitive advantages is also vital for SMEs, for which the impact might be significant. Cantele and Zardini (2018) showed a positive relationship between sustainable initiatives and financial results, mediated by competitive advantage. Thus, sustainable businesses are more competitive and have better financial results. Rigby and Tager (2008) consider sustainability a growth strategy for companies.

Goals like cutting emissions, reducing pollution, using sustainable materials, and having sustainable partners bring benefits such as:

- having a better image in the community
- attracting more customers and quality employees
- being more trustworthy for their partners
- tax benefits granted by the government to encourage these behaviours and initiatives
- in the long run, it leads to higher profits.

For sustainable businesses to be successful, consumers should be more oriented to notice these types of businesses and buy products and services from them. Here comes the role of educational campaigns that contribute to raising the awareness level in the community, to having more educated, knowledgeable, and responsible citizens and, thus, buyers and entrepreneurs. Many studies in the professional literature highlight the role of education in promoting sustainability in business and the community in general (Calder and Dautremont-Smith, 2009; Rowe, 2007; Hill and Lee, 2012). Hill and Lee (2012) researched Generation Y consumers whose perceptions of sustainability are influenced by their knowledge of these issues.

For sustainable businesses to be profitable and efficient in the long run, entrepreneurs should be creative (generate new and great ideas) and innovative (focused on novelty and usefulness). Being sustainable and innovative in business ensures higher competitive advantages, higher chances of survival in crisis, loyalty from consumers, and increased market share. Sustainability and innovation in business are extensively researched (Todeschini et al., 2017; Evans et al., 2017; Shakeel et al., 2020; Seebode et al., 2012; Borkovskaya, 2013). As Shakeel et al. (2020) state, "Business Model Innovation is a future to deal with sustainability challenges". Seebode et al. (2012) appreciate that innovation is necessary for coping with "growing pressures and emerging opportunities in the 'sustainability' agenda."

There are many examples of good practices of innovative and sustainable businesses that are focused on solving some of the problems referred to by the 17 SDGs:

- **Bio-bean** (https://www.bio-bean.com/) is a UK business that uses coffee grounds from various partners and gives them value. Thus, they transform it into coffee logs and pellets, natural coffee flavors, and raw materials for multiple industries looking for more environmentally friendly alternatives for their businesses.
- **Fairphone** (https://www.fairphone.com/) is a Dutch company that offers phones made from recycled and sustainable components. They have easy-to-repair designs and put people and the planet first. Fairness is the core value of their business.
- **Studio Roosegaarde** (https://www.studioroosegaarde.net/) is a Dutch research laboratory focused on creating sustainable projects such as the Smog Free Tower, Smog Free Ring, and Smog Free Bicycle (aimed at reducing pollution and cleaning the air); Spark (organic fireworks), Space Waste Lab (cleaning up waste from space and recycling it); Seeing Stars (aimed at raising awareness of urban pollution); Grow (using a specific light to help plants grow and make people enjoy it as an art form).
- **Eonef** (http://eonef.com/) is a French company focused on bringing solar balloons to regions with no electricity and balloons equipped with sensors and cameras to monitor air quality or detect fires (Lampoon Magazine, 2023).
- **Groasis** (https://www.groasis.com/) is a Dutch company focused on transforming arid areas into living land by using efficient technologies, saving water, and considering the specifics of these areas. Some of their products are Groasis Waterboxx (saves water and helps plants grow without irrigation), Groasis Growboxx (helps plants grow in mountainous regions), Growsafe Telescoprotexx (protects plants from being eaten by

animals), Groasis Capillary Drills (excavators special for desert areas) or Groasis Terracedixx (to use rainwater more efficiently).

- **Demetra** (https://www.demetrafood.it/) is an Italian food company "determined to reduce polluting emissions and collect waste separately."
- **The John Lewis Partnership** (https://www.johnlewispartnership.co.uk/) includes two major UK retailers, John Lewis and Waitrose. This partnership announced in 2020 its "commitment to net zero carbon emissions" for the year 2050 by using biomethane fuel for their vans.
- **The German Water Foundation** (https://www.wasserstiftung.de/) has developed CloudFisher, a technology that could turn fog into safe drinking water.
- **Sheep Inc.** (https://eu.sheepinc.com/) offers clothes made from wool from sheep farms in New Zealand. Their "Buy a Sweater, Adopt a Sheep" campaign informs people of where their clothes come from. Also, the NFC tag that all the company's clothes have can be used by consumers to find out more details about the products bought, the wool, the sheep it comes from, the farm of origin, and other information that makes people aware of which it is their carbon footprint (Malachosky, 2022).
- **Rimping**, a supermarket in Thailand, started using banana peels for packaging in 2019 as a more sustainable way to use materials that would otherwise be thrown away and thus reduce plastic (Nace, 2019).

The world has many socioeconomic challenges and, therefore, multiple opportunities for governments, private organizations, NGOs, and citizens to act more responsibly and sustainably to solve some of these problems. Likewise, the 17 SDGs in the 2030 Agenda are all possibilities that can be translated into public and/or private initiatives to reduce climate change and provide equal rights for everyone. Sustainable businesses offer benefits to their communities and significant benefits for companies.

The business world is changing, but educational campaigns are also needed to raise awareness of climate change, global warming, poverty, hunger, and inequality and create more informed consumers and entrepreneurs. We are all responsible for the quality of the environment in which we live, and we have an obligation to future generations to leave them clean air, water, and land.

We are in a situation where there is no radical solution to stop air pollution altogether, but there are solutions to reduce emissions of certain air pollutants.

According to Directive (EU) no. 2016/2284 of December 14, 2016, of the European Parliament and the Council regarding the reduction of national emissions of certain atmospheric pollutants, Romania must significantly reduce emissions of NOx (nitrogen oxides), VOCnm (non-methane volatile organic compounds), SO_2 (carbon dioxide sulphur), NH_3 (ammonia) and PM2.5 (fine particles in suspension) by 60%, 45%, 88%, 25%, and 58% respectively by 2030 (compared to 2005).

We might ask how these emissions can exist in the air and their origin. How to be careful? Do we have the free will not to breathe these emissions into the body?

Through a short presentation, we can answer the above questions:

- NO_x (nitrogen oxides) is primarily the result of power plants, vehicles, and industrial and domestic combustion processes. Road transport is the primary cause;
- VOC (volatile organic compounds) – are the result of paints, varnishes, waxes, oil-dissolving solvents, detergents, fuels, disinfectants, cosmetics, and glues; these compounds can also be produced from smoking and fuel burning;

- SO_2 (sulphur dioxide) – is the result of heating systems of the population that do not use methane gas, thermoelectric power plants, industrial processes (steel, refinery, production of sulfuric acid), pulp and paper industry;
- NH_3 (ammonia) – is mainly the result of poultry and pig farms;
- PM2.5 (fine particles in suspension) – results from burning petrol, oil, diesel, or wood.

So, we know how these emissions exist in the air and where they come from, but can we have the free will not to inhale these emissions?

According to the National Air Pollution Control Program, the Ministry of the Environment, Waters and Forests has a significant role in the field of ambient air quality assessment, being the competent authority that coordinates the development and implementation, together with the central public authorities, of the National Air Pollution Control Program Atmospheric Pollution. The Ministry of the Environment, Water and Forests, in collaboration with the significant public authorities with responsibilities in the fields of economy, energy, health, agriculture, rural development, sanitary-veterinary and food safety, regional development, and public administration and transport, establish measures to reduce annual the national anthropogenic emissions SO_2, NO_X, VOC, NH_3, and PM2.5, so as not to exceed the commitments assumed at the national level to reduce emissions. The same National Air Pollution Control Program certifies that emissions have already decreased significantly, mainly SO_2 emissions, which were by approx. 88% lower in 2020 compared to 2005, followed by reductions in NOx emissions (by approx. 42%), VOC (by approx. 30%), NH_3 (by approx. 19%) and PM2.5 (by 7%). Thus, the progress made due to the policies and measures in force, evaluated considering the historical emissions reported for 2020 and the reference year 2005, indicates a downward trend for all pollutants and compliance with commitments to reduce emissions for VOC, SO_2, and NH_3. In 2020, reduction commitments are exceeded for NO_x (by approximately 3%) and PM2.5 (about 21%).

It somewhat "sounds" good in numbers, but the feeling is still not a comfortable; in other words, you do not have that electric impulse of your heart to want to go outside the house and breathe in a big breath of fresh air, because the air we breathe is harmful, in spite of all the commitments assumed by Romania. "Assumed," why do we feel that this word does not protect us 100%? On the contrary, it creates a state of unease, but at the same time, we cannot help but think that human nature over the years has evolved but not also imply that the instinct of survival is above all else.

As humanity has evolved, it has gradually learned the harmful effects of, starting from a catastrophic experience in London between December 5 and 9, 1952, there was a severe air pollution phenomenon called "The Great Smog of 1952" when home heating was based predominantly on coal, the combustion gases together with the fog formed a layer of smog, a phenomenon due to which more than 4,000 people died (from cardiovascular and respiratory diseases). As a result of this catastrophic experience, in 1956, the Clean Air Act was adopted, which aimed to reduce air pollution in large British cities.

Another example is an accident at a pesticide factory near Seveso in northern Italy. It released a cloud of dioxin into the atmosphere, a hazardous substance known for its carcinogenic effects, even in tiny doses. Unfortunately, the affected area, covering an area of 18 km^2, was a residential area; thus, 37.000 people encountered the contaminated air. The disaster was so big that the Council of Europe later issued a unique directive, the so-called "Seveso Directive," which brought strict regulations for producing and storing about 80 substances considered very dangerous.

We cannot help but think of another catastrophic disaster known to the whole population, namely the Chernobyl disaster, when one of the reactors of the Chernobyl power plant in Ukraine exploded, releasing an enormous amount of radiation into the atmosphere, more significant than the bombs in Hiroshima and Nagasaki.

Finally, in Romania, an example of a catastrophic disaster was that of Copsa Mică when the city was buried by pollution. There were two factories in Copsa Mică; one produced black carbon, used in tires, and the other produced zinc, lead, copper, cadmium, and other non-ferrous metals. Together, they spew 30,000 tons of particulates and soot each year.

Humanity indeed needs power plants, road transport, detergents, disinfectants, steel, refineries, and even poultry and pig farms, etc., but how can we manage to need all these essential things in our existence without affecting the environment without polluting the air and without affecting us, humans?

From 16 March 2023 – The Green Deal Industrial Plan (often referred to as the European Industrial Plan) is an initiative of the European Union (EU) and supports the transition to climate neutrality by increasing the competitiveness of Europe's net zero emissions industry. In March, three of the main proposals of this plan were presented: The European Critical Raw Materials Act, the Net Zero Emissions Industry Act, and the Electricity Market Organization Reform. They will support the plan by creating a more straightforward and predictable regulatory environment for clean technologies so that they start or continue to develop in the EU. The benefits of the European Green Deal provide support for fresh air, clean water, healthy soil, and biodiversity; renovated, energy-efficient buildings; healthy food at affordable prices; more public transport options; cleaner energy and clean, cutting-edge technological innovation; more resistant products, which can be repaired, recycled, and reused; jobs adapted to the demands of the future and the training of skills necessary for the transition resilient and globally competitive industry.

The nature of humans over time is to evolve, to want to make their existence more effortless, and to do great things that not only help us but help and protect the environment in which we live. We cannot help but wonder if there is a legal framework regarding air quality if the public knows information or data collected regarding the surrounding air quality or if Romania respects agreements, conventions, or treaties to which it is a party.

As a legal framework regarding air quality, European regulations are fully transposed into national legislation through the adoption of Law no. 104/2011 on ambient air quality, with subsequent amendments and additions (provided by GD no. 806/2016 for amending annexes no. 4, 5, 6 and 7 to Law no. 104/2011 on ambient air quality. According to the provisions of Art. 2 of Law no. 104/2011 on the quality of the surrounding air, "this law provides measures at the national level regarding:

- defining and establishing objectives for the quality of the surrounding air intended to avoid and prevent the occurrence of harmful events and reduce their effects on human health and the environment as a whole;
- the assessment of the quality of the surrounding air throughout the country based on standard methods and criteria established at the European level;
- obtaining information on the quality of the surrounding air to support the process of combating air pollution and the discomfort caused by it, as well as to monitor long-term trends and improvements resulting from the measures taken at national and European level;
- guaranteeing that information on the quality of the surrounding air is made available to the public;

- maintaining the quality of the surrounding air where it is appropriate and/or improving it in other cases;
- promoting increased cooperation with the other member states of the European Union to reduce air pollution;
- fulfilling the obligations assumed by the agreements, conventions, and international treaties to which Romania is a part".

Also, under the provisions of art. 55 paragraph 1 of Law no. 104/2011 "if any of the alert thresholds, limit values or target values are exceeded, plus the corresponding tolerance margin or the long-term objective long, due to a cross-border transport of atmospheric pollutants or their precursors, the central public authority for environmental protection cooperates directly with the counterpart authority from the neighbouring member states of the European Union or through the central public authority for the implementation of the foreign policy of the Romanian state with the counterpart authority from neighbouring states that are not members of the European Union and where appropriate, establishes joint actions, such as the development of joint or correlated air quality plans to eliminate these exceedances by applying appropriate measures, which do not involve disproportionate costs, not least, para. 4 of Art. 55 of Law no. 104/2011 specifies that "in case the information threshold and/or the alert threshold is exceeded in an area or agglomeration near the border, the central public authority for environmental protection informs as soon as possible it is possible the competent authorities from the neighbouring member states of the European Union concerned."

Starting from the desire to have something tangible, to see in numbers, to see a diagram, to see concrete and correct data, University of Craiova, Department of Sciences in partnership with Adana Türkeş Alparslan University of Science and Technology, University from Plovdiv "Paisii Hilendarski" and Matej Bel University from Banská Bystrica within the Erasmus+ project "Applying Advanced Technologies in Air Pollution Teaching and Research" offers students the opportunity to learn how to build an air monitoring sensor as well as what these sensors entail and how they can draw attention to air quality.

Air quality monitoring sensors help us, both individuals (since we can purchase such a sensor for our home as well) and legal entities (such as public institutions); in this way, we can have a representation of air quality and why not, if one day the air quality is not clean, it would be ideal to avoid leaving the house that day, or to notify the competent authorities. The creation and the existence of these sensors in our daily lives is imperative given the high degree of air pollution we face.

This lab book presents teaching-related teachers' ideas about the role of technology in the teaching-learning process. It is also an exchange of good practices and ideas of some people involved in training engineering and science students from four universities – UCv, PU, UMB, and ATU.

The book aims to help students become familiar with some technologies focused on the environment (air pollution). Lab work shows how to make a sensor starting from some electronic components, how to program it, connect it to a network, and how the sensor transmits data to a sizeable sensor-independent network. Also, the lab book presents how to download the data produced by each sensor, how the data can be organized and visualized, and how students can analyse the obtained data set. Knowledge, especially experimental skills, is essential for future engineers and science graduates.

In chapter 1, a brief description of a kit of sensors that can measure air temperature is made. The role and place of each component are discussed in this chapter. The second chapter focuses on the practical realization of some activities based on solar irradiance sensors. It shows how to

connect the electronic components to the main board containing an Arduino-type microcontroller. Some educational possibilities in this respect are evaluated. The third chapter proposes a few activities based on a sound level sensor. The fourth chapter is about how GM counter is made and how we can protect ourselves from radiation effects. The fifth chapter presents a meteorological station, and the sixth is about a sensor for air quality monitoring.

Chapter 7 addresses the challenges of 3D printing a shield to protect the sensor from the elements. Chapters 8, 9, and 10 will discuss the role of an independent sensor network within educational initiatives for the benefit of citizens, highlighting the advantages of sensors and the limitations of measurements in training future engineers and science graduates, comparing laboratory measurement methods using sensors with the measurements made by some official air quality monitoring stations. These chapters describe the procedure for downloading and processing the collected data with a study case from Adana.

This laboratory activities book shares faculty's ideas about how students can learn by exploring and doing. This book contains a short description of some kits of sensors. Each laboratory activity has a theoretical part in which the concepts necessary to understand each laboratory work are explained, the aim, the objectives, a brief description of the equipment that is used, the set-up, the steps taken, the way of work, warnings related to a series of hazards that may occur when using devices, data collection, data analysis, real-life connection, challenges, and conclusions.

References

1. Bio-bean. Available online at https://www.bio-bean.com/ (Accessed on 23 March 2023)
2. Borkovskaya, V. G. (2013). The Concept of Innovation for Sustainable Development in the Construction Business and Education. Applied Mechanics and Materials, 475–476, 1703–1706. https://doi.org/10.4028/www.scientific.net/amm.475-476.1703
3. Calder, W., & Dautremont-Smith, J. (2009). Higher education: More and more laboratories for inventing a sustainable future. Agenda for a sustainable America, 93–107. Available online at http://ulsf.org/wp-content/uploads/2015/06/ESDHigherEdAmericaCalder20091.pdf (Accessed on 31 March 2023)
4. Cambridge Academic Dictionary Content (n.d.). Entrepreneur Definition. Available online at https://dictionary.cambridge.org/dictionary/english/entrepreneur (Accessed on 25 March 2023)
5. Cantele, S., & Zardini, A. (2018). Is sustainability a competitive advantage for small businesses? An empirical analysis of possible mediators in the sustainability–financial performance relationship. Journal of cleaner production, 182, 166–176.
6. Carbon Monitor. CO2 emissions in 2020. Available online at https://carbonmonitor.org/ (Accessed on 29 March 2023)
7. Carter, N. M., Gartner, W. B., Shaver, K. G., & Gatewood, E. J. (2003). The career reasons of nascent entrepreneurs. Journal of Business Venturing, 18(1), 13–39. https://doi.org/10.1016/S0883-9026(02)00078-2
8. Demetra. Available online at https://www.demetrafood.it/ (Accessed on 23 March 2023)
9. Eonef. Available online at http://eonef.com/ (Accessed on 23 March 2023)
10. Evans, S., Vladimirova, D., Holgado, M., Van Fossen, K., Yang, M., Silva, E. A., & Barlow, C. Y. (2017). Business model innovation for sustainability: Towards a unified perspective for creation of sustainable business models. Business strategy and the environment, 26(5), 597–608.

11. European Environment Agency (2019). Healthy environment, healthy lives. Available online at https://www.eea.europa.eu/publications/healthy-environment-healthy-lives (Accessed on 29 March 2023)
12. European Environment Agency (2021). Air quality in Europe 2021. Health impacts of air pollution in Europe, 2021. Available online at https://www.eea.europa.eu/publications/air-quality-in-europe-2021/health-impacts-of-air-pollution (Accessed on 29 March 2023)
13. European Environment Agency (2022). Air pollution: how it affects our health. Available online at https://www.eea.europa.eu/themes/air/health-impacts-of-air-pollution (Accessed on 29 March 2023)
14. European Environment Agency (2023). Air quality in Europe 2022. Available online at https://www.eea.europa.eu//publications/air-quality-in-europe-2022 (Accessed on 29 March 2023)
15. European Parliament (2023). Circular economy: definition, importance, and benefits. Available online at https://www.europarl.europa.eu/news/en/headlines/economy/20151201STO05603/circular-economy-definition-importance-and-benefits (Accessed on 28 March 2023)
16. Fairphone. Available online at https://www.fairphone.com/ (Accessed on 31 March 2023)
17. German Water Foundation. Available online at https://www.wasserstiftung.de/ (Accessed on 31 March 2023)
18. Groasis. Available online at https://www.groasis.com/ (Accessed on 31 March 2023)
19. Hill, J., & Lee, H. H. (2012). Young Generation Y consumers' perceptions of sustainability in the apparel industry. Journal of Fashion Marketing and Management: An International Journal, 16(4), 477–491.
20. John Lewis Partnership. Available online at https://www.johnlewispartnership.co.uk/ (Accessed on 31 March 2023)
21. Kampa, M., & Castanas, E. (2008). Human health effects of air pollution. Environmental pollution, 151(2), 362–367. https://doi.org/10.1016/j.envpol.2007.06.012
22. Lampoon Magazine (2023). EONEF's tethered balloons: air quality monitoring, wildlife conservation and forest fire detection. Available online at https://www.lampoonmagazine.com/article/2023/02/12/eonefs-tethered-balloons-air-quality-monitoring-wildlife-conservation-and-forest-fire-detection/ (Accessed on 31 March 2023)
23. Little, M. P. (2003). Risks associated with ionizing radiation: Environmental pollution and health. British medical bulletin, 68(1), 259–275. https://doi.org/10.1093/bmb/ldg031
24. Malachosky, E. (2022). Buy a Sweater, Adopt a Sheep. Available online at https://www.gearpatrol.com/style/a38831352/merino-wool-sheep-inc/ (Accessed on 31 March 2023)
25. Mattera, M., Alba Ruiz-Morales, C., Gava, L., & Soto, F. (2022). Sustainable business models to create sustainable competitive advantages: strategic approach to overcoming COVID-19 crisis and improve financial performance. Competitiveness Review: An International Business Journal, 32(3), 455–474.
26. Nace, T. (2019). Thailand Supermarket Ditches Plastic Packaging For Banana Leaves. Available online at https://www.forbes.com/sites/trevornace/2019/03/25/thailand-supermarket-uses-banana-leaves-instead-of-plastic-packaging/ (Accessed on 31 March 2023)

27. Rigby, D., & Tager, S. (2008). Learning the advantages of sustainable growth. Strategy & Leadership, 36(4), 24–28.
28. Righi, S., Lucialli, P., & Bruzzi, L. (2005). Health and environmental impacts of a fertilizer plant–Part I: Assessment of radioactive pollution. Journal of environmental radioactivity, 82(2), 167–182. https://doi.org/10.1016/j.jenvrad.2004.11.007
29. Rodrigues, S. M., & Römkens, P. (2018). Human health risks and soil pollution. In Soil Pollution (pp. 217–250). Academic Press. https://doi.org/10.1016/B978-0-12-849873-6.00009-1
30. Rowe, D. (2007). Education for a sustainable future. Science, 317(5836), 323–324.
31. Seebode, D., Jeanrenaud, S., & Bessant, J. (2012). Managing innovation for sustainability. R&d Management, 42(3), 195–206. https://doi.org/10.1111/j.1467-9310.2012.00678.x
32. Shakeel, J., Mardani, A., Chofreh, A. G., Goni, F. A., & Klemeš, J. J. (2020). Anatomy of sustainable business model innovation. Journal of cleaner production, 261, 121201. https://doi.org/10.1016/j.jclepro.2020.121201
33. Sheep Inc. Available online at https://eu.sheepinc.com/ (Accessed on 25 March 2023)
34. Stephan, U., Hart, M., Mickiewicz, T., & Drews, C. C. (2015). Understanding motivations for entrepreneurship. Available online at https://publications.aston.ac.uk/id/eprint/25296/1/Understanding_motivations_for_entrepreneurship.pdf (Accessed on 25 March 2023)
35. Studio Roosegaarde. Available online at https://www.studioroosegaarde.net/ (Accessed on 31 March 2023)
36. Todeschini, B. V., Cortimiglia, M. N., Callegaro-de-Menezes, D., & Ghezzi, A. (2017). Innovative and sustainable business models in the fashion industry: Entrepreneurial drivers, opportunities, and challenges. Business Horizons, 60(6), 759–770. https://doi.org/10.1016/j.bushor.2017.07.003
37. United Nations (1987). Report of the World Commission on Environment and Development: Our Common Future. Available online at http://www.un-documents.net/our-common-future.pdf (Accessed on 27 March 2023)
38. United Nations (n.d.). Sustainable Development. Available online at https://www.un.org/ecosoc/en/sustainable-development (Accessed on 27 March 2023)
39. World Health Organization (1982). Rapid assessment of sources of air, water, and land pollution. Geneva, Switzerland.
40. https://ro.wikipedia.org/wiki/Marele_Smog_din_1952 (Accessed on 7 April 2023)
41. https://www.manager.ro/articole/afla-67/analizele-managerro-cele-mai-mari-catastrofe-ecologice-produse-de-mana-omului-norul-de-dioxina-din-italia-(iv)-8516.html (Accessed on 7 April 2023)
42. https://www.digi24.ro/stiri/sci-tech/natura-si-mediu/cele-mai-mari-catastrofe-ecologice-din-romania-si-din-lume-993038 (Accessed on 7 April 2023)
43. https://adevarul.ro/stiri-locale/hunedoara/secretele-oraselor-toxice-din-romania-comunista-2027764.html (Accessed on 7 April 2023)
44. https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal_ro (Accessed on 7 April 2023)
45. https://lege5.ro/App/Document/gi2tqmjzgq/legea-nr-104-2011-privind-calitatea-aerului-inconjurator (Accessed on 7 April 2023)

CHAPTER 1.
HIGH-PRECISION MEASUREMENT OF THE AIR TEMPERATURE USING A RTD PT100 SENSOR

This chapter was written by Rumen Popov
from the Paisii Hilendarski University of Plovdiv, Bulgaria

1.1. Theory

1.1.1. Temperature measurement using RTDs

The principle of operation of a Resistance Temperature Detector (RTD) is based on the property of electrically conductive materials to change their electrical resistance [1] when the temperature changes. Pure metals have found the most comprehensive application for making RTD. The formula determines their specific resistance ρ:

$$\rho = \frac{1}{n\_e\mu\_}, \quad (1)$$

where

$n\_$, m^{-3} is the number of free electrons per unit volume;

e, C is the charge of the electron;

$\mu\_$, m^2/Vs – the mobility of electrons, numerically equal to the speed they would have in an electric field of unit intensity.

The specific resistance ρ of metals is relatively small – from 10^{-10} to 10^{-5} Ω m. This is due to the high concentration of electrons, which does not depend on temperature.

At all temperatures, ρ depends on the fluctuations of the crystal lattice and is determined by the mobility of electrons and the presence of impurities. Therefore, the specific resistance of pure metals can be represented in the form

$$\rho = \rho_0 + \rho(T), \quad (2)$$

where ρ_0 does not depend on temperature.

1.1.2. Precious metal RTDs

These RTDs are widely used in practice due to the high stability of their parameters and the high reproducibility of measurement results. The main applications are platinum for temperatures from 10 to 1200 K and titanium for low temperatures.

Platinum up to 1800 K does not enter chemical reactions and retains its properties. Due to the influence of a few processes – diffusion, structure change, etc., the application of platinum RTDs in some environments is limited to lower temperatures (1200 °C). Platinum has a relatively high resistivity. The dependence of the resistance R_Θ on the temperature Θ in a reasonably wide temperature range (for temperatures above 0 °C) is described accurately enough by Callender's formula

$$R_\Theta = R_0(1 + A\Theta + B\Theta^2), \quad (3)$$

where R_0 is the resistance at 0°C, *A* and *B* are constants that are usually determined from the thermistor resistances measured at three reference points on the international temperature scale.

According to BDS 10726-73, the constants *A* and *B* are determined by measuring the resistance at the triple point of water θ_{tp} = 0.01 °C, the boiling point of water θ_k = 100 °C, and the solidification point of zinc θ_{Zn}= 419.58 °C.

At temperatures from 0 to -200 °C, the dependence of the resistance R_θ on the temperature θ is described by the formula

$$R_\theta = R_0[1 + A\Theta + B\Theta^2 + C(\Theta - 100)\Theta^3] \quad (4)$$

Here, the constants *A* and *B* are as in equation (3), and the constant *C* is determined at the boiling point of oxygen θ_{O2} = -182.97 °C. According to BDS 10713-73, the constants can have values *A* = (3.90785÷3.9685).10^{-3} °C $^{-1}$, *B* = – (5.875÷5.85).10^{-7} °C^{-2}, C = 4.2.10^{-12} °C^{-4}.

For the precision thermometers' conversions, the considered constants' values must be determined individually. From equation (4) it is obtained accordingly

$$A = \frac{(R_к - R_0)\theta_{Zn}^2 - (R_{Zn} - R_0)\theta_к^2}{R_0\theta_к\theta_{Zn}(\theta_{Zn} - \theta_к)}, °C^{-1} \quad (5)$$

$$B = \frac{(R_к - R_0)\theta_{Zn} - (R_{Zn} - R_0)\theta_к}{R_0\theta_к\theta_{Zn}(\theta_{Zn} - \theta_к)}, °C^{-2} \quad (6)$$

$$C = \frac{R_{O_2} - R_0 - AR_0\theta_{O_2} - BR_0\theta_{O_2}^2}{R_0\theta_{O_2}^3(\theta_{O_2} - 100)}, °C^{-4} \quad (7)$$

where R_k, R_{Zn}, and R_{O2} are the resistances at the boiling temperature of water θ_k, of zinc solidification θ_{Zn}, and of oxygen boiling θ_{O2}, respectively. The temperature θ_k may differ from 100°C depending on atmospheric pressure, altitude, and latitude. Then, the resistance R_{100} at a temperature of 100°C is obtained from

$$R_{100} = R_k + (R_k - R_0)\frac{100 - \theta_k}{\theta_k} - 5,83.10^{-5}R_0(100 - \theta_k). \quad (8)$$

The resistance R_0 at 0°C is determined either by measuring the resistance of the RTD at the melting temperature of the ice or in a particular device reproducing the triple point of water θ_T = 0.01°C. In the latter case (if increased accuracy is required), it is necessary to reduce the resulting resistance R_{tr} according to the expression

$$R_0 = R_{tr}(1 - 3,93.10^{-5}). \quad (9)$$

In the practically important case, when R_θ is known, then the current temperature θ can be determined by Eq. (10) using the sum of two components θ_1 and θ_Δ Eq. (11) and Eq. (12):

$$\theta = \theta_1 + \theta_\Delta \quad (10)$$

$$\theta_\Delta = 0,045\frac{\theta}{100}\left(\frac{\theta}{419,58} - 1\right)\left(\frac{\theta}{630,74} - 1\right), \quad (11)$$

$$\theta_1 = \frac{2\left(\frac{R_\theta}{R_0} - 1\right)}{A + \sqrt{A^2 + \left(\frac{R_\theta}{R_0} - 1\right)\cdot 4B}} \pm \theta_\Delta, \quad (12)$$

where *A* = 3.9083×10^{-3} °C^{-1} and *B* = –5.775×10^{-7} °C^{-2}.

Fig. 1.1 graphically shows the value of this correction depending on the temperature calculated by (12).

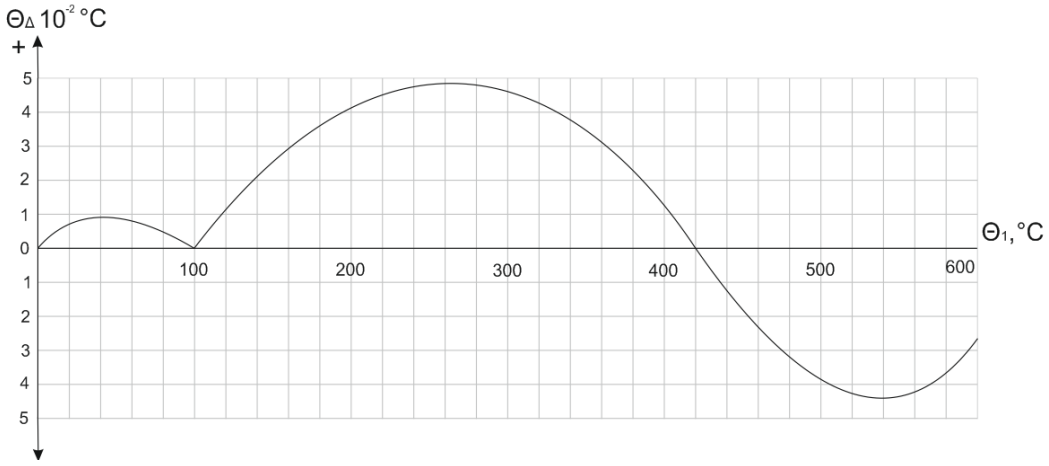


Figure 1.1. *Dependence of the correction Θ_Δ on the measured temperature*

1.1.3. PRTD signal conditioner

Microchip Technology also proposes a low-cost solution (Fig. 1.2) of the PRTD signal conditioner using low power, single supply, rail-to-rail operational amplifier MCP609 [2]. A current generator, realized by A1, A2, and a precision voltage source, excites the sensor. An op-amp (A3) cancels the wire resistance RW error.

The design shown in Fig. 1.2 has been used to develop the Expansion card's RTD signal conditioner section. The main reasons for this were a low price, simple structure, and flexible modular design. Each one of DAQ-board's four analog input channels is multiplexed by 8, using two separate multiplexers, one for the current source switching to the RTD excitation and another one for the output voltage switching to the DAQ-board's analog input channel.

The structure of the eight-channel multiplexed signal conditioner described above is entirely suitable for the LabJack UE9 DAQ board.

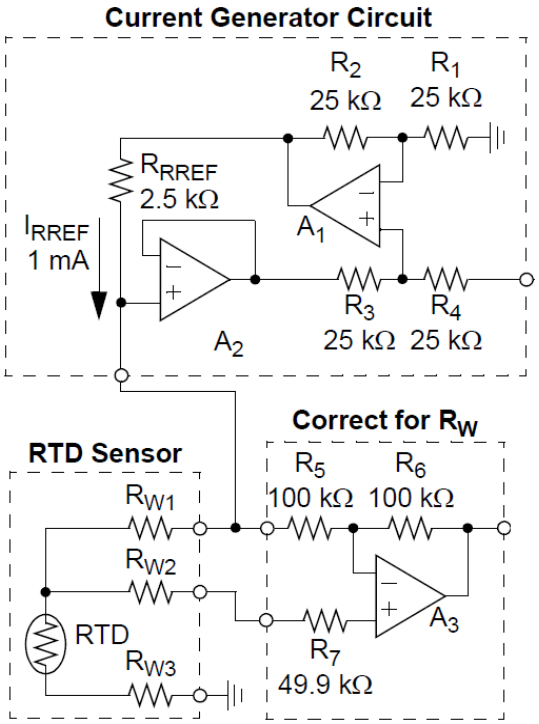


Figure 1.2. *RTD signal conditioner using op-amp, source [2]*

1.1.4. Data Acquisition (DAQ) Board

A Data Acquisition (DAQ) Board, often called a DAQ card or DAQ module, is a device that is used to acquire analog and digital data from a variety of sensors, measuring instruments and devices and convert it into digital form for analysis and processing by a computer.

The large number of measuring points requires the use of multichannel instrument. The "Expansion card" has been developed for this purpose. It allowed (Fig. 1.3) to expand the number of the analog channels provided by our DAQ-board LabJack UE9 [3] as listed below:

- 32 temperature Pt100 3-wire RTD channels;
- 16 differential low voltage input channels (for DC currents, solar radiation instruments, thermocouples, etc.);
- Up to 128 DS1820 temperature sensors or humidity sensors (1-Wire interface);
- 10 DC voltage input channels (for ACV, ACA meter outputs);
- 8 counter inputs (for the flow rate and the energy meters).

Additionally, 12 relay and 8 TTL-level outputs are available to control loads.

The 3-wire RTD frontend was connected to the four existing DAQ analog voltage channels by 8-fold multiplexing to obtain 32 RTD channels. Fig. 1.4 presents a schematic diagram of one of the four electronic sections connected to the DAQ board's analog input channel AIN6. The Expansion card consists of three more identical electronic sections connected to the channels AIN7, AIN8, and AIN9 of the Lab Jack UE9 board.

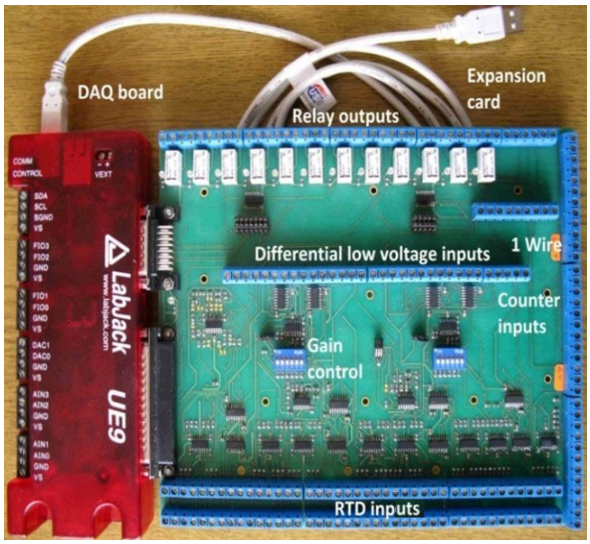


***Figure 1.3**. Expansion card and DAQ board*

The component count has been minimized by using only one current source for the eight RTD Pt100 sensors excitation (total of four in the designed board). Also, only one 2.5 V source provides reference voltage for all 32 RTD signal conditioners mounted on the board.

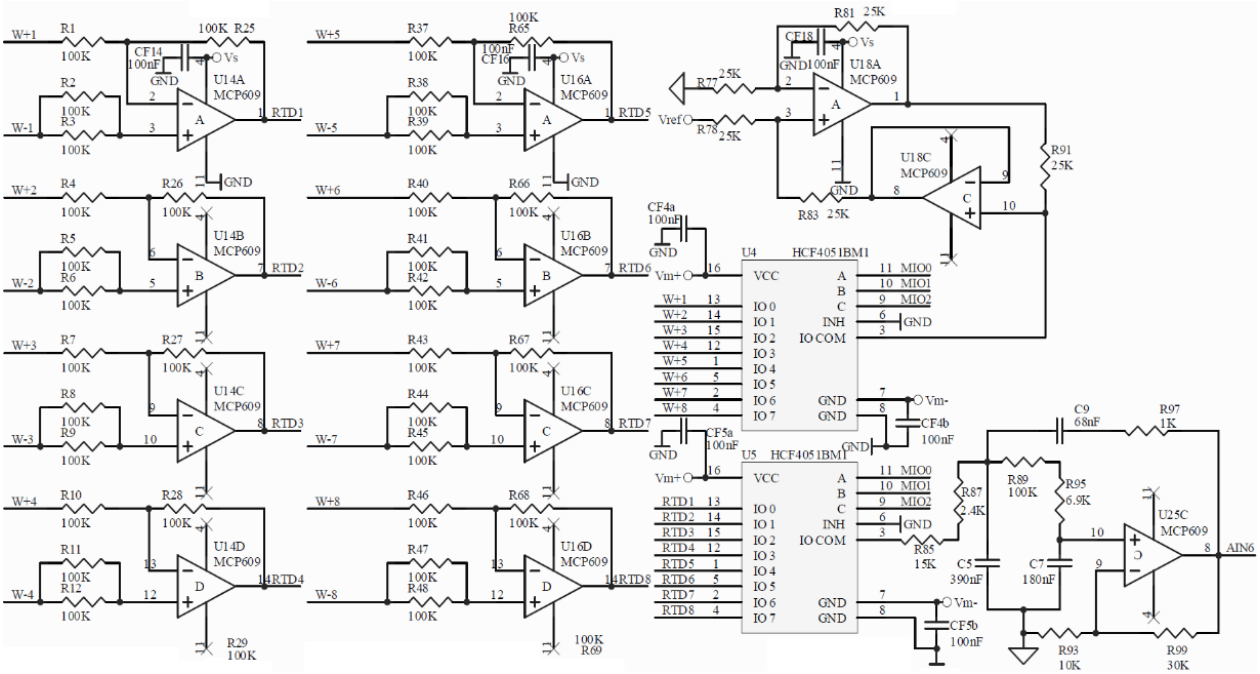


Figure 1.4. *The schematic diagram for one of the 8-channel signal conditioner electronic sections*

Two low-cost eight-channel multiplexers form the switching network. When opened, the current generated by the excitation current source does not depend on the resistance of the HCF4051 switch U4. Both multiplexers are powered by the bipolar ±6.3V supply (Vm- and Vm+), available from LabJack UE9 to ensure all the switchable voltage range. The input channel is addressed using three signals (MIO0, MIO1, and MIO2) generated automatically by DAQ-board and software available in the address field of LabJack's UE9 driver for National Instruments LabVIEW.

The output voltage signal of the multiplexer U5 (IO COM) on pin No 3 is filtered with a 2nd order, low pass Sallen-Key filter. This filter is set to gain of 7.47 V/V. and to 8 Hz cutoff frequency. It is suitable only if low sampling rates are used. If the sampling rate exceeds 0.33 Hz, the capacitors C5, C7, and C9 must be removed to ensure fast switching capability. Software filtering is additionally performed in this case.

1.1.5. The RTD Resistance Calculation

The value of the RTD Resistance R_θ depends on the current reading of the LabJack UE9 DAQ board. It is calculated using equation (13).

$$R_\theta = (U_{IN\_UE9}*213.79)/(3.4), \quad (13)$$

where U_{IN_UE9} is the current voltage reading of the LabJack UE9 DAQ board. The value of the R_θ is put into equation (11) to calculate the temperature.

1.2. Laboratory setup

1.2.1. Hardware configuration

The laboratory setup is presented in Fig. 1.5 and is composed of the following components

- DAQ-board LabJack UE9;
- Expansion card;
- 3-wire RTD Pt 100 temperature sensor, connected to RTD channel 1 of the expansion card;

- PC;
- USB cable.

Figure 1.5. *The laboratory test setup*

1.2.2. The software application

The data logging software application LJLogUD (Fig. 1.6) collects measurement data and stores the results. It also allows pre-processing of the measurement data using simple custom equations.

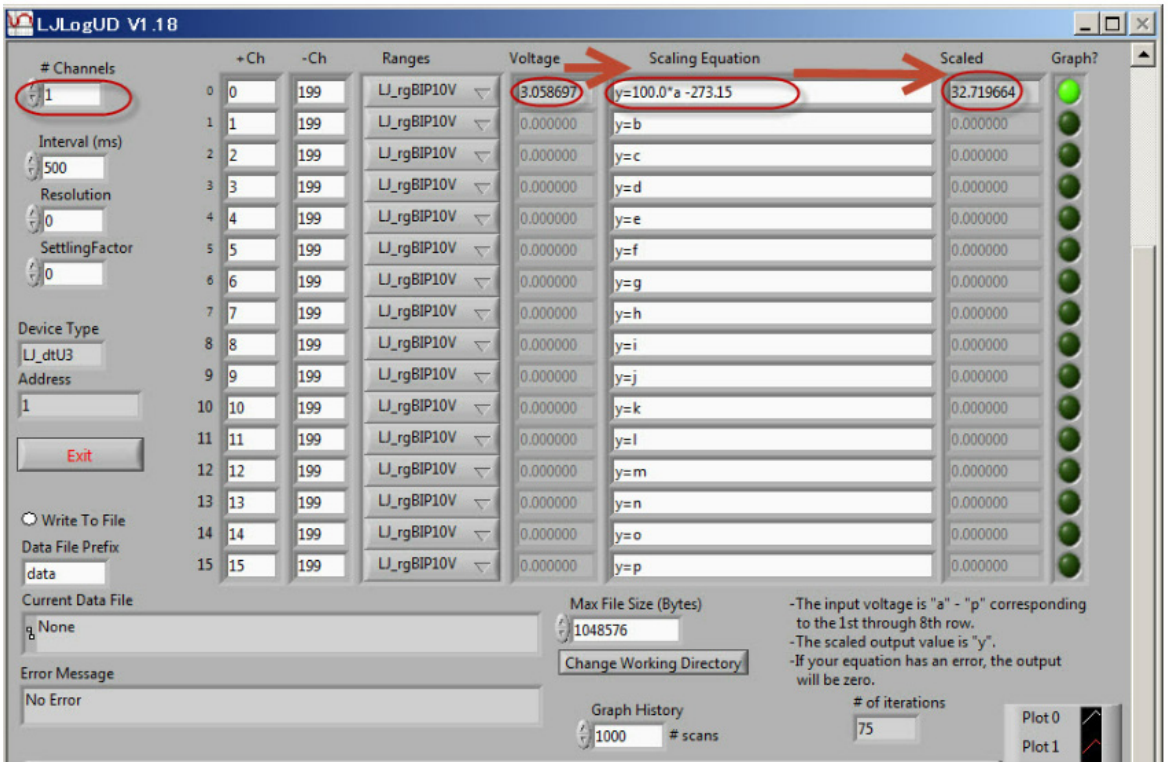


Figure 1.6. *Software application LJLogUD*

1.3. Tasks

1. *Connect all components of the laboratory test setup (Fig. 1.5);*
2. *Start the software application and configure measurement parameters, setting the custom equation for R_θ measurement;*
3. *Start measurement and store results (about 20 samples);*
4. *Import data (.csv format) to an Excel worksheet and use equations (10), (11), (12), and (13) to calculate the air temperature;*
5. *Create a graph of the temperature change during the time;*
6. *Write a report.*

References:

[1] Куртев И., к.т.н. инж. Д. Самоловлийски, Измерване на температура, Държавно издателство „Техника“, София, 1982

[2] Bonnie C. Baker. Precision Temperature-Sensing with RTD Circuits, Application Note AN687, Microchip Technology Inc., 2008.

[3] LabJack Corporation, UE9 Multifunction DAQ board datapage https://labjack.com/products/ue9

CHAPTER 2. MEASUREMENT OF THE SOLAR IRRADIATION COMPONENTS

This chapter was written by Rumen Popov
from the Paisii Hilendarski University of Plovdiv, Bulgaria

2.1. Theory

2.1.1. Direct, Global and Diffuse Solar Radiation

The direct solar beam arriving directly at the earth's surface is called **direct solar radiation**.

Global solar radiation is the total amount of solar radiation falling on a horizontal surface (i.e., the direct solar beam plus diffuse solar radiation on a horizontal surface).

Direct solar radiation is observed from sunrise to sunset, while global solar radiation is observed in the twilight before sunrise and after sunset, despite its diminished intensity at these times.

Units:

The solar irradiance is expressed in watts per square meter (W/m^2) and the total amount in joules per square meter (J/m^2). Conversion between the currently used unit (SI) and the former unit (calories) can be performed using the following formulae:

Solar irradiance: $1\ kW/m^2 = 1.433\ cal/cm^2/min$

Total amount of solar radiation: $1\ MJ/m = 23.89\ cal/cm^2$

The value of direct solar irradiance is about 120 W/m^2 at around sunrise and sunset and about 800 W/m^2 at around noon on a clear day in summer.

Diffuse solar radiation is sunlight that has been scattered by atmospheric constituents and/or by the surface. The three main components of the solar radiation reaching the earth's surface are interrelated through the equation given in Fig. 2.1.

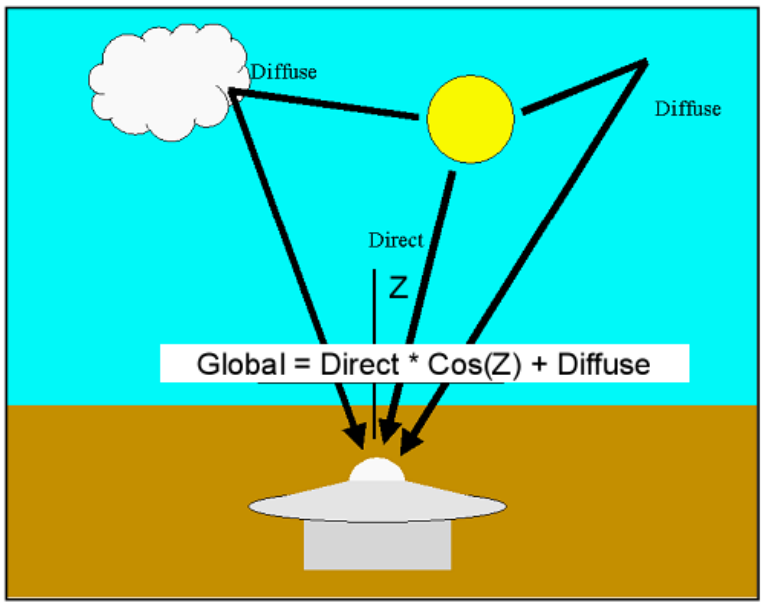


Figure 2.1. *Relation between Direct, Global, and Diffuse components of the solar radiation*

We can calculate the value of each of these by knowing the other two and the zenith angle of the direct component Z.

$$I_G = I_D * \cos(Z) + I_{DIF}, \quad (1)$$

where:

- I_G is the global solar irradiance;
- I_D is the direct solar irradiance;
- I_{DIF} is the diffuse solar irradiance;
- Z is the zenith angle.

The zenith angle Z depends on the sun's position relative to the measurement point.

2.1.2. Measurement Instruments

The radiometers used for ordinary observation are pyrheliometers and pyranometers, measuring direct and global solar radiation, respectively. A radiometer absorbs solar radiation at its sensor, transforms it into heat, and measures the resulting amount of heat to ascertain the level of solar radiation.

Methods of measuring heat include taking out heat flux as a temperature change (using a water flow pyrheliometer, a silver-disk pyrheliometer, or a bimetallic pyranograph) or as a thermoelectromotive force (using a thermoelectric pyrheliometer or a thermoelectric pyranometer). In current operation, types using a thermopile are generally used.

In this work, two radiometers are used CMP6 Pyranometer and CHP1Pyrheliometer, both manufactured by Kipp & Zonen.

2.1.2.A. CMP6 Pyranometer

The CMP6 pyranometer [1] is intended for routine global solar radiation measurement research on a plane/level surface. Fully compliant with ISO 9060:2018 spectrally flat Class B specifications, the CMP6 features a sixty-four-thermocouple junction (series connected) sensing element. The sensing element is coated with a highly stable carbon-based non-organic coating, which delivers excellent spectral absorption and long-term stability characteristics (see Fig. 2.2).

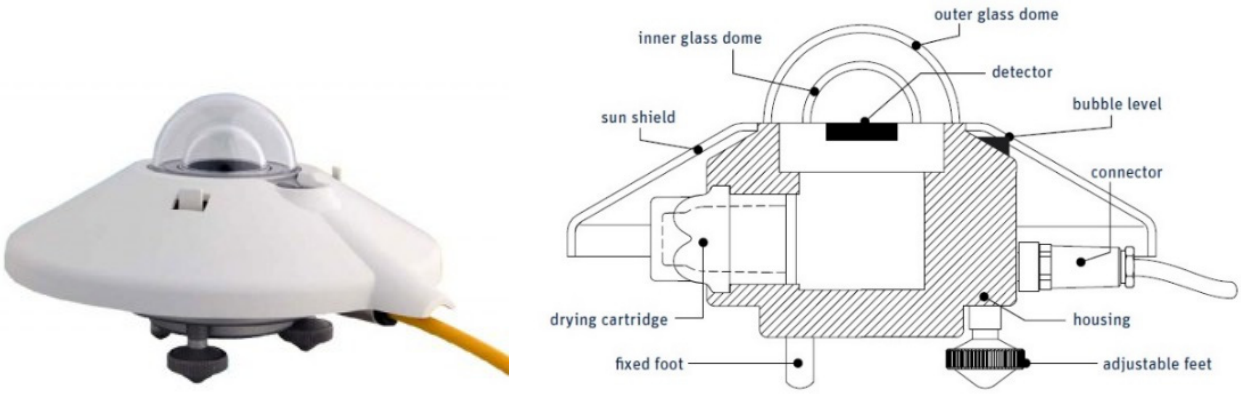


Figure 2.2. *CMP6 Pyranometer construction*

The technical specifications of the CMP6 pyranometer are listed in Table 1.

Table 1. *CMP6 Pyranometer Specifications*

Spectral range (50% points)	285 to 2800 nm
Sensitivity	5 to 20 µV/W/m²
Response time	12 s
Zero offset A	< ±8 W/m²
Zero offset B	< ±2 W/m²
Directional response (up to 80° with 1000 W/m² beam)	< ±20 W/m²
Temperature response (-10 to +40°C)	< ±2 %
Operating and storage temperature range	-40 °C to +80 °C
Maximum solar irradiance	2000 W/m²
Field of view	180 °

2.1.2.B. CHP 1 Pyrheliometer

A pyrheliometer is designed to measure the direct beam solar irradiance with a field of view limited to 5°, also known as Direct Normal Irradiance, DNI. This is achieved by the shape of the collimation tube, precision apertures, and the detector design. The front aperture has a quartz window to protect the instrument and act as a filter that passes solar radiation between 200 and 4000nm in wavelength.

The CHP 1 Pyrheliometer [2] (see Fig. 2.3) is an all-weather pyrheliometer available for continuous direct solar radiation measurements. The construction of the CHP 1 Pyrheliometer is shown in Fig. 2.4. The CHP 1 exceeds the specifications for high-end solar radiation networks, such as the Baseline Surface Radiation Network (BSRN) of the World Climate Research Programme (WCRP). These networks require accurate and reliable long-term measurements for climate change investigations.

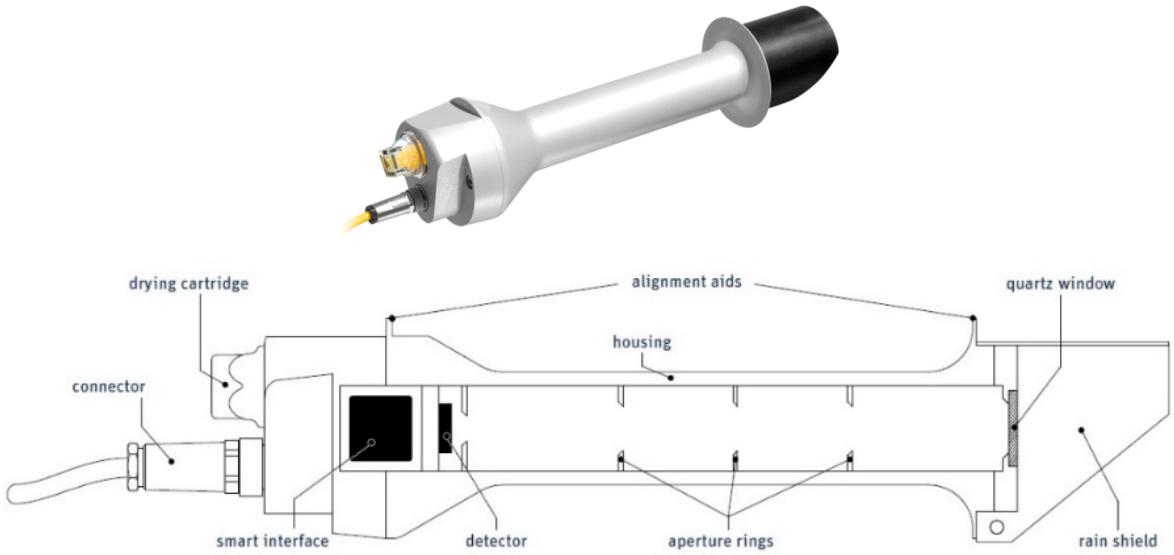


Figure 2.3. *CHP 1 Pyrheliometer construction*

The technical specifications of the CHP 1 Pyrheliometer are listed in Table 2.

Table 2. *CHP 1 Pyrheliometer Specifications*

Spectral Range	200 to 4000 nm
Sensitivity	7 to 14 µV/W/m^2
Response Time	< 5 s
Zero Offset B	< 1 W/m^2
Temperature Dependence of Sensitivity	< 0.5% (-20° to +50°C)
Field of View (FOV)	5° ±0.2°
Operating Temperature Range	-40° to +80°C
Non-Linearity	< 0.2%
Maximum Irradiance	4000 W/m^2
International Standards	First Class ISO
Body Diameter	3.8 cm (1.5 in.)

2.1.3. Direct, Global, and Diffuse Solar Radiation Measurement

2.1.3.A. Data Acquisition (DAQ) Board

The large number of measuring points requires the use of multichannel instrument. The "Expansion card" has been developed for this purpose. It allowed (Fig. 2.4) to expand the number of the analogue channels provided by our DAQ-board LabJack UE9 [3] as listed below:

- 32 temperature Pt100 3-wire RTD channels;
- 16 differential low voltage input channels (for DC currents, solar radiation instruments, thermocouples, etc.);
- Up to 128 DS1820 temperature sensors or humidity sensors (1-Wire interface);
- 10 DC voltage input channels (for ACV, ACA meter outputs);
- 8 counter inputs (for the flow rate and the energy meters).

Additionally, 12 relay and 8 TTL-level outputs are available to control loads.

The 3-wire RTD front end was connected to the four existing DAQ analog voltage channels by 8-fold multiplexing to obtain 32 RTD channels.

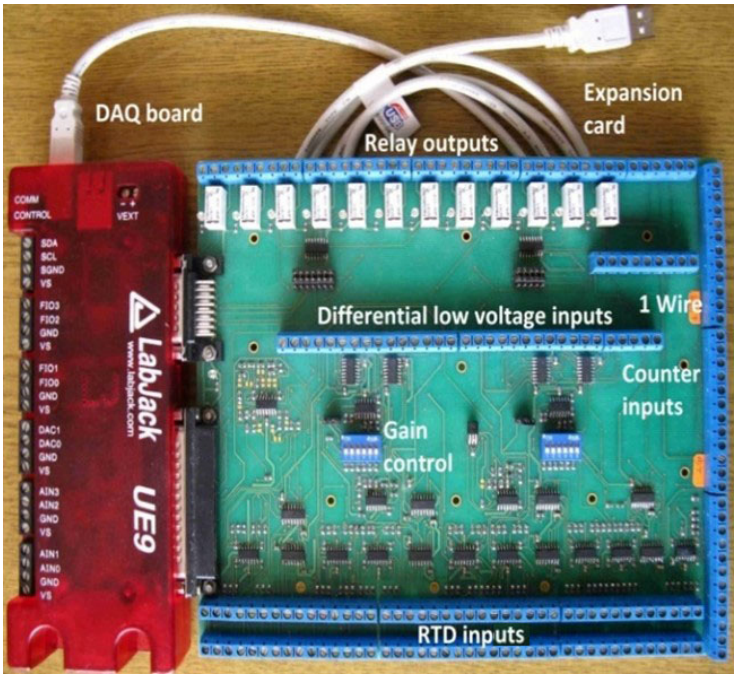


Figure 2.4*. Expansion card and DAQ board*

Fig. 2.4 presents a schematic diagram of the input connection from radiometers to the DAQ-board's analog input channel AIN10 of the LabJack UE9.

A classic instrumentation amplifier circuit with three operational amplifiers is used to gain a low-level signal from the thermopile sensor (see Fig. 2.5). The two input operational amplifiers [4] have a gain factor for differential and common-mode signals of +1. The output op-amp is a differential amplifier that converts the input signal from differential to single-ended; it suppresses common-mode signals at its input. The resistor (RG) can easily adjust the circuit's gain. The reference voltage V_{REF} in unipolar supply circuits is usually half of the supply voltage (VDD/2)

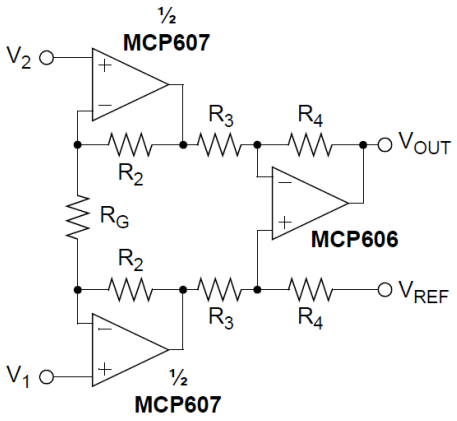


***Figure 2.5**. Instrumentation amplifier with three operational amplifiers*

2.1.3.B. Direct Solar Radiation Measurement

Suppose the output signal of the CHP 1 Pyrheliometer is connected to AIN10 of the LabJack UE9 through the instrumentation amplifier presented in Fig. 2.5. In that case, several steps should be performed to calculate the value of the incoming direct solar irradiation:

– The calculation of the maximum voltage at the output of the pyrheliometer U_{PIRH_MAX} is done by formula (2):

$$U_{PIRH\_MAX} = S\ I_{D\_MAX}, \tag{2}$$

where S is the sensitivity of the CHP 1 device – 10 μVm²/W; I_{D_MAX} is the maximum measured direct solar radiation – 1000 W/m².

Then, the calculated value is

$$U_{PIRH\_MAX} = 10.10^{-6}.1000 = 10\ \text{mV}. \tag{3}$$

- The maximum gain *k* is found by dividing the maximum value of the input range of UE9 U_{UE9_MAX}, which is chosen by setting 625 mV to the maximum voltage at the output of the pyrheliometer U_{PIRH_MAX}.

$$k_{max} = U_{UE9\_MAX} / U_{PIRH\_MAX}, \tag{4}$$

$k_{max} = 630.10^{-3}/10.10^{-3} = 63$ V/V. We choose gain $k = 50$. Thus, the maximum measurement range from the device (500 mV) will be within the limits of the maximum for the input of UE9 (625 mV).

- The calculation of the direct radiation I_D from the voltage at the input of UE9 U_{IN_UE9} is done by the equation (5):

$$I_D = U_{IN\_UE9} /S.k, \text{ so } I_D = U_{IN\_UE9} /10.10^{-6}.50 \tag{5}$$

2.1.3.C. Global Solar Radiation Measurement

Suppose the output signal of the CMP6 pyranometer is connected to AIN11 of the LabJack UE9 trough instrumentation amplifier presented in Fig. 2.5. In that case, the same steps described above should be performed to calculate the value of the incoming global solar irradiation. The resulting equation, in this case, becomes as:

$$I_G = U_{IN\_UE9} / S.k, \text{ so } I_G = U_{IN\_UE9} / 12{,}5.10^{-6}.50, \quad (6)$$

where S (for the CMP6 pyranometer) is known to be12.5 μVm²/W; *k* is the same as in the case above (*k* = 50).

2.1.3.D. Diffuse Solar Radiation Calculation

The value of the diffuse solar irradiation I_{DIF} is obtained using equation (1) and the results of the last two measurements: I_D and I_G. The value of the zenith angle *Z* is relatively difficult to calculate manually, so an Internet tool "SunPosition" [5] should be used. Current clock time, time zone, and the measurement point location are needed as input data. The altitude angle is obtained as an output. Then, the zenith angle should be calculated as

$$Z = 90 \text{ deg.} - \text{altitude angle value.} \quad (7)$$

2.2. Laboratory setup

2.2.1. Hardware configuration

The laboratory setup is presented in Fig. 2.6 and is composed of the following components

- DAQ-board LabJack UE9;
- Expansion card;
- CMP6 pyranometer, connected to the differential low voltage channel 10 of the expansion card;
- CHP1 pyrheliometer, connected to the differential low voltage channel 11 of the expansion card;
- PC;
- USB cable.

Figure 2.6. *The laboratory test setup*

2.2.2. The software application

The data logging software application LJLogUD (Fig. 2.7) collects measurement data and stores the results. It also allows pre-processing of the measurement data using simple custom equations.

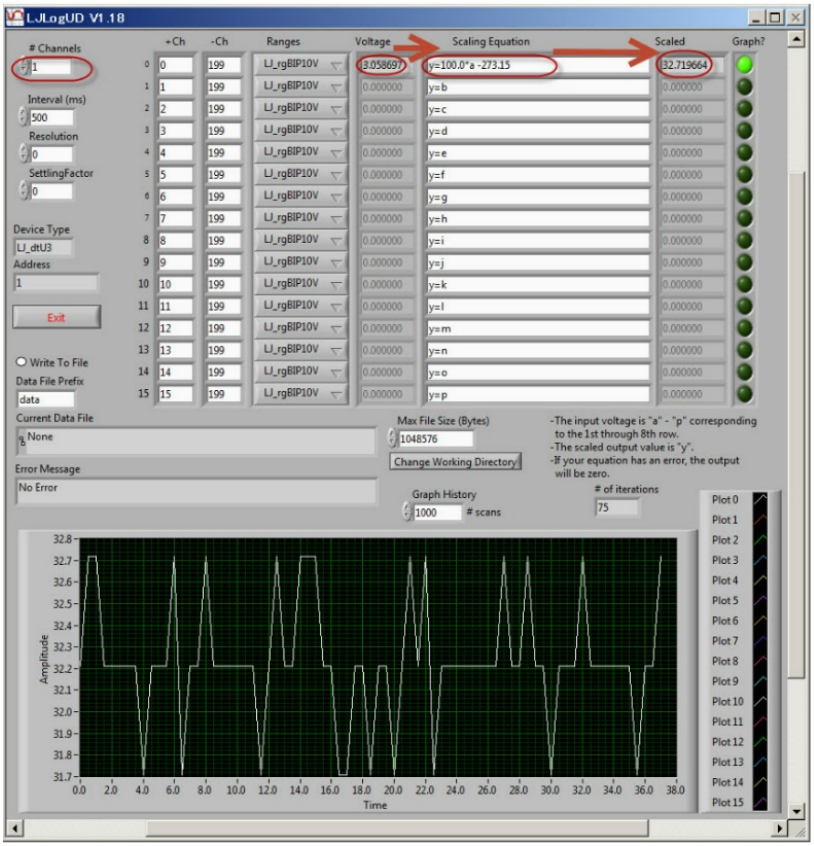


Figure 2.7. *Software application LJLogUD*

2.3. Tasks

1. *Connect all components of the laboratory test setup (Fig. 2.5);*
2. *Start the software application and configure measurement parameters, setting the custom equation for I_D and I_G measurement;*
3. *Start measurement and store results (about 20 samples). During this step, the CHP 1 Pyrheliometer should be directed precisely to the sun, using a proper stand;*
4. *Import data (.csv format) to Excel worksheet and use equation (1) to calculate the diffuse irradiance value I_{DIF};*
5. *Create graphs of the change of solar irradiation components during the time;*
6. *Prepare a report.*

References

[1] https://www.kippzonen.com/Product/12/CMP6-Pyranometer
[2] https://www.kippzonen.com/Product/18/CHP1-Pyrheliometer
[3] LabJack Corporation, UE9 Multifunction DAQ board datapage https://labjack.com/products/ue9
[4] Microchip App. Note, MCP606/7/8/9. http://ww1.microchip.com/downloads/en/devicedoc/11177f.pdf
[5] Internet tool "SunPosition" http://www.susdesign.com/sunposition/index.php

CHAPTER 3. A SOUND LEVEL METER USING AN ARDUINO

This chapter was written by Martin Hruska
from the Matej Bell University of Banska Bystrica, Slovakia

3.1. Theory

3.1.1. What does sound intensity level mean?

The sound intensity level, often called noise, is one of the fundamental environmental parameters. However, its measurement is very complicated; therefore, it is dedicated to specialized companies with a certificate for performing such activity. The state's relevant regulations and standards determine the sound measurement methodology and permissible values. Today, enough quality noise meters are already available on the market, with which we can carry out our measurements.

The following text will show how we can construct a relatively simple sound level meter using the Arduino Uno microcontroller (Fig. 3.1).

We will present our experience with the development and construction of a sound meter and the pitfalls that must be dealt with during the construction of a sound level meter. Such a device does not reach the quality of professional instruments. We will explain the reasons below. Nevertheless, we can use it for indicative measurement, which enables us to check up or confirm some laws regarding sound and its propagation in the environment.

Figure 3.1. *Sound meter with Arduino Uno microcontroller*

3.1.2. How to calculate sound intensity level?

In the following text, we will present selected factors related to the peculiarities of noise measurement, which we must keep in mind when designing a sound level meter or designing our program.

The text does not aim to describe all aspects and physical laws related to noise propagation in the environment. If the reader needs to learn more about noise measurement, it is necessary to study the relevant standards and methodologies that are currently in force in the given country.

Loudness or sound intensity level L [dB] can be calculated using the formula:

$$L = 10 \log \frac{I}{I_0} = 20 log \frac{p}{p_{ref}} , \quad (1)$$

where I is sound intensity [Wm^{-2}], I_0 is the lowest registerable intensity of a pure tone with a frequency of 1 kHz ($I_0 = 10^{-12}$ Wm^{-2}), $p_{ref} = 2 \cdot 10^{-5}$ Pa is a threshold value of acoustic pressure (the lowest value of pressure that can be detected by hearing – threshold of audibility), and p is the instantaneous sound pressure value. Sound intensity is directly proportional to the square of the sound pressure. During measurements, the instantaneous value of the sound pressure p must be replaced by the effective value of the sound pressure (p_{rms}) over some time since the sound pressure changes very quickly over time.

That is why noise meters often have several measurement modes. For example, sound level meter RFT 00024 measures in modes S (Slow) when it calculates the level of noise intensity in 1 s, mode F (Fast) in 125 ms, and mode I (Impulse) in 35 ms.

The physical reasons why formula (1) contains the logarithmic dependence of the share of acoustic pressures multiplied by a factor of 20, as well as the reasons that lead to the need to work with the effective value of the acoustic pressure function, are well known, but exceed the scope of this text, so we will not discuss them here to dissect. Let us only say that the effective value of the acoustic pressure is directly proportional to the energy of the investigated acoustic signal.

Let us remember that for the effective value of the function y_{ef}, respectively y_{rms}, whose instantaneous value $y(t)$ changes at time t, for the (appropriate) time interval $\Delta t = t_2 - t_1$ holds:

$$y_{rms} = \sqrt{\frac{1}{\Delta t} \int_{t_1}^{t_2} y^2(t).dt} . \quad (2)$$

The microcontroller measures a continuous function (usually voltage U) by recording its instantaneous values at certain time intervals with a selected sampling frequency (Fig. 3.2), which is called signal quantization (Gray & Neuhoff, 1998). Therefore, we can replace the integral with the sum:

$$U_{rms} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} U^2(i)} , \quad (3)$$

where n is the number of instantaneous voltage values from the microphone sensor that the Arduino Uno records in the selected time interval.

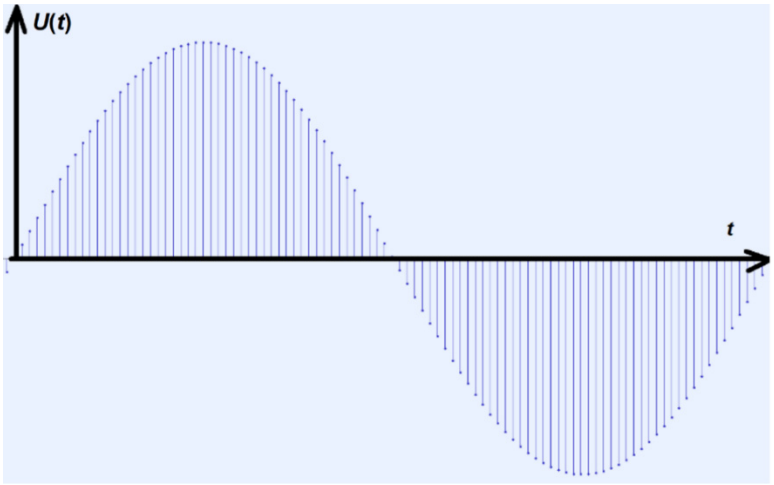


***Figure 3.2**. The course of the harmonic signal generated in the Audacity program*

The resulting subjective perception of sound, which corresponds to the level of its intensity, depends on many other factors, while research in this area is still bringing new knowledge.

We will at least mention the dependence of the sound intensity level on its frequency.

The audibility of human hearing ranges from 20 Hz to 20 kHz. However, the sensitivity to sound sensations depends on the sound frequency, and we are most sensitive to sounds with frequencies from 2 kHz to 5 kHz. At the same time, the frequency dependence of hearing is more pronounced at low sound pressure levels and less pronounced at high sound pressure levels. Therefore, weight filters A, B, C, D, Lin, and Z are currently used, which consider the sensitivity of human hearing depending on the magnitude of the acoustic intensity level (Degro, 2009), (Frequency-Weightings for Sound Level Measurements, 2023), (Sound Level Frequency Weightings – A, B, C, D, Lin, Z, 2023).

For calculations, the area of sound frequencies is often divided into octave bands with intervals of 1:2 middle frequencies, while, e.g., filter A assigns the following corrections to individual bands (tab. 1).

Nowadays, some websites calculate the effect of frequency filters on individual frequencies, e.g. (Calculation of Frequency Weightings, 1996).

Table 1. *Corrections of frequency weighting filter A*

Band mid-frequency f_m [Hz]	63	125	250	500	1000	2000	4000	8000
Correction K_{Ai} [dB]	-26,2	-16,1	-8,6	-3,2	0	1,2	1,0	-1,1

The resulting L_A sound intensity level must then be calculated according to formula (4):

$$L_A = 10 \cdot log \sum_{i=1}^{n} 10^{\frac{L_i + K_{Ai}}{10}} . \quad (4)$$

For the above reasons, a professional sound level meter must also contain a so-called weight filter or several weight filters that adjust the meter sensitivity depending on the frequency interval and the sound intensity value. Decomposition of the signal into individual frequency components or division into intervals is part of frequency analysis. However, from the point of view of creating a program for a simple Arduino sound level meter, it is a more complicated matter, so we will not deal with it further.

Tab. 2 shows indicative values of the sound intensity level depending on the source of the appliance in the interior and on another equivalent sound source (Výber spotrebiča podľa hladiny hluku, 2019).

Table 2. Selected sound sources and their typical sound intensity values

***L* [dB]**	**Appliance**	**Equivalent activity**
39 – 50	refrigerator, dishwasher	silent street, normal sound background
51	washing machine during washing	quiet conversation, light rain
55	TV, hand blender	loud conversation
60	digestor	
60 – 80	vacuum cleaner	social event, applause in the hall, strong reproduced music
70 – 75	washing machine during spinning, electric kettle	
90	–	a train passing by
110	–	discotheque
120 – 130	–	pain threshold

3.1.3. Propagation of sound in the environment

The simplest example of a sound source is a point source in free space. Such a source of sound can be, for example, an airplane flying high above the ground.

In this case, the acoustic power *P* of the source is uniformly distributed over the surface of an imaginary sphere with the corresponding radius *R* so that we can write for the intensity (5):

$$I_R = \frac{P}{4\pi R^2}\,. \tag{5}$$

It follows from relation (5) that doubling the distance from the source to 2*R* will cause the intensity to drop to a quarter of the original value:

$$I_{2R} = \frac{P}{4\pi(2R)^2} = \frac{I_R}{4} \tag{6}$$

If we want to calculate how the sound intensity level, or noise, will change, we must proceed using formula (1) as follows:

$$\Delta L = L_{2R} - L_R = 10\left\{\log\frac{I_{2R}}{I_0} - \log\frac{I_R}{I_0}\right\}$$

$$\Delta L = 10\log\frac{\frac{I_{2R}}{I_0}}{\frac{I_R}{I_0}} = 10\log\frac{I_{2R}}{I_R} = 10\log 0{,}25 \cong \underline{-6\ \mathrm{dB}} \tag{7}$$

It means that doubling the distance from a point source causes a decrease in sound intensity by approximately 6 dB.

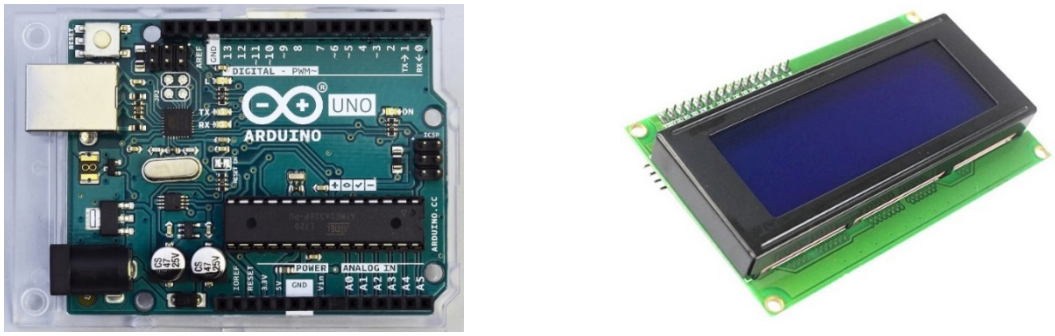
3.2. Laboratory setup

3.2.1. A brief description of the individual components of the developed Arduino sound level meter

- ***Arduino Uno, liquid-crystal display (LCD) 20 x 4 with blue backlight***

For the construction of a simple sound level meter, we will need an Arduino UNO microcontroller and a liquid-crystal display (Fig. 3.3a, b).

These components are described in more detail in the text describing the construction of the meteorological station in Chapter 5, so we will not go into them in depth.



a) b)

Figure 3.3. *Arduino Uno microcontroller (a) together with LCD (b)*

- ***Microphone module with MAX4466 amplifier***

Electret microphone with low-noise operational amplifier Maxim MAX4466 is manufactured by Adafruit. According to the manufacturer's website (Adafruit, 2023), the sensor provides high-quality sound with a small amount of unwanted noise.

On the same page, we can also find CAD files with the wiring and PCB of the sensor, the basic catalogue data of the electret microphone used, and the MAX4466 amplifier. The operating frequency of the microphone is 20 Hz to 20 kHz; the sound characteristic of the microphone is omnidirectional, meaning it captures sounds from all directions (Fig. 3.4).

We can adjust the sensor's gain with the potentiometer on the back in the interval from 25x to 125x.

For the project needs, we recommend setting the sensor gain at the very beginning and not changing it anymore, as the final calibration of the calculated sound intensity will need to be adjusted to the gain set!

The sensor does not require installing a library for its operation. Its dimensions are approximately 14 mm x 22 mm, and it also has two mounting holes for fixing with screws.

***Figure 3.4**. Electret microphone with operational amplifier MAX4466*

- ***AMS1117 voltage stabilizer (3.3 V), microphone module power supply***

During the construction of the Arduino sound level meter, it became clear that it is necessary to use a sufficiently high-quality stabilizer to stabilize the supply voltage of the microphone module. If we powered the microphone module directly from the 3.3 V DC output of the Arduino, the measurements were highly inaccurate.

In the second step, we used the 3.3 V stabilizer AMS1117 (Fig. 3.5), which, even though it was intended directly for modules sensitive to the power supply to Arduino, did not provide a sufficiently smoothed voltage.

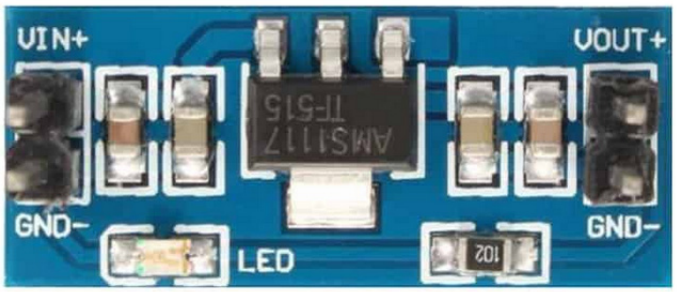


***Figure 3.5**. AMS1117 3.3V voltage stabilizer, powered by 5V DC output of Arduino*

Finally, as a stabilizer or reference voltage source, we have used the LP2950ACZ-3.3/NOPB circuit, whose output voltage accuracy reaches a value of 0.5 % (it represents a deviation from the reference voltage of ± 16.5 mV).

Fig. 3.6 shows its connection in the sound meter with the capacitors at the input and output.

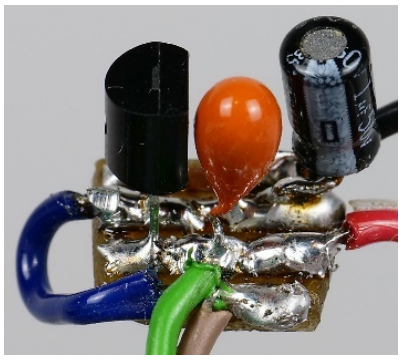


***Figure 3.6**. Voltage stabilizer LP2950ACZ-3.3/NOPB with 1 µF and 2.2 µF capacitors, located on the PCB and powered by the 5 V DC output of the Arduino*

Insufficient voltage stabilization of the microphone module manifested itself in several ways:

- Noise in the power supply of the microphone module.

By default, various modules in Arduino applications are powered by Arduino itself with a voltage of 3.3 V DC or 5 V DC. In the case of a sound level meter, this power supply is not stable enough, as it contains various low- and high-frequency interferences, the source of which is primarily the Arduino itself (mostly various communication signals). This problem is subsequently reflected in the impossibility of measuring low and medium sound intensity values because the power supply is a source of so much interference in the amplifiers that the sound intensity values do not fall below 60 dB even in complete silence. By its very nature, the electret microphone capsule contains an operational amplifier that amplifies the signal from the microphone, and it is powered directly through the circuit board of the microphone module through the so-called phantom power supply. An additional amplification is provided by the MAX4466 operational amplifier, located on the microphone sensor's PCB. Suppose we power such a system of amplifiers from a voltage source with much interference. In that case, this electronic interference is also amplified (despite using various filters to smooth out the power supply). At the output of the sound module, we will hear a loud noise, which will, of course, adversely affect the calculation of the sound pressure value and, consequently, the sound intensity. Therefore, it is no coincidence that many noise meters are powered by internal accumulators or batteries, which eliminate this type of interference.

- Instability of the working point of the microphone module depending on the power supply of the Arduino sound meter, temperature, etc.

The microphone module on its output provides a voltage signal for the Arduino, and its values can vary from 0 V to 3.20 V (depending on the gain setting of the microphone sensor at the beginning). This means that with zero sound excitation when there is practically absolute silence around the microphone, the value at the output reaches 1.60 V exactly and almost does not change over time. Lower voltage values represent negative sound pressure values; higher values than 1.60 V correspond to positive sound pressure values. Since, in the case of an electret microphone, the voltage at the output is directly proportional to the acoustic pressure, we can easily convert the voltage values to the acoustic pressure. For the sound level meter to measure correctly, the average voltage value of 1.60 V must remain constant as it enters the sound intensity calculation. If this point moves during the measurement, the sound level meter will show incorrect and significantly higher sound intensity values. At the same time, thanks to the very sensitive logarithm – even with a slight deviation from the mean voltage value of 0.1 mV, the sound pressure values move by units to tens of dB. This is

one of the reasons why even some commercially available noise meters can measure from 30 dB to 50 dB.

- Thermal stability of the power supply and the Arduino power supply.

Our experience has shown that the mean voltage value of 1.60 V shifts slightly depending on the device's temperature. It is probably mainly due to the temperature instability of the microphone sensor with the MAX4466 amplifier, the operating point of which changes slightly with temperature.

- Stability of the working point depending on the type of power supply.

Another, even more significant influence on the shift of the working point is the power source of the sound level meter – different DC adapters, as well as USB ports of computers or USB adapters, provide slightly different voltage values, which is also reflected in the shift of the average voltage value of the microphone module. We tried to eliminate this influence by connecting two potentiometers behind the voltage stabilizer – for coarser and finer tuning of the average value of the voltage after connecting the power supply (Fig. 3.7). It is possible to correct the fluctuation of the value of the working point during the measurement. The final wiring diagram of the Arduino sound meter and its block diagram are presented in the following subsection.

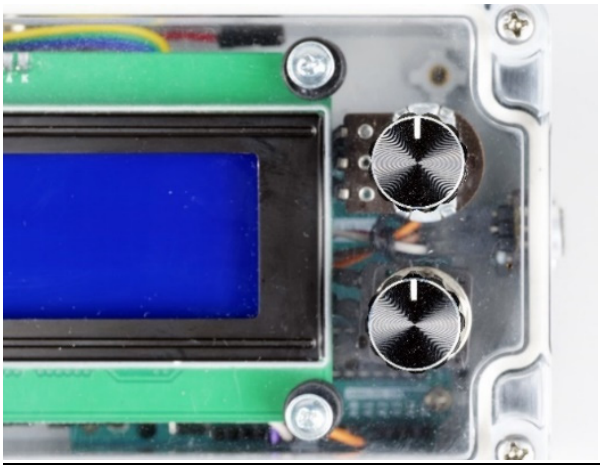


***Figure 3.7**. A view of the two potentiometer rotary controls.*
The upper potentiometer is used for coarser calibration of the working point of the Arduino sound level meter, and the lower potentiometer is used for its fine-tuning. ***Attention – it is necessary to use linear potentiometers (designation LIN or B).*** *Logarithmic potentiometers (LOG or A) are used for volume control.*

3.2.2. Description of the connection and construction of the Arduino sound meter

Before we describe the electrotechnical connection of the sound level meter, we will first focus on its block or functional diagram (Fig. 3.8). The simplified wiring is based on the block diagram of a standard, commercially available sound level meter.

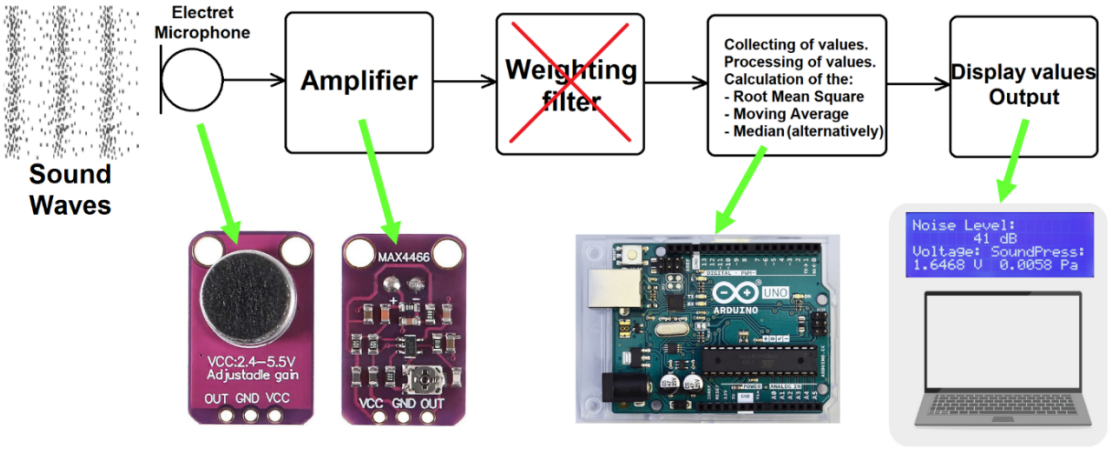


Figure 3.8. *Block diagram of the Arduino sound meter*

The microphone module and the operational amplifier capture the sound impulse, convert it to an electrical signal, and amplify it. The electrical signal subsequently enters the Arduino Uno microcontroller, where the spectral analysis of the signal and its adjustment using a weighted filter are omitted. From a series of voltage values, the effective value of the voltage is calculated, converted to acoustic pressure, and subsequently entered the calculation of the sound intensity value. To reduce random noise, either a floating average is calculated from the measured sound intensity values or the median is determined (depending on the program version and our needs). The results are either shown on the display or recorded on the computer. The display also shows the effective value of the measured voltage and the effective value of the sound pressure.

The circuit diagram of the sound level meter is shown in Fig. 3.9.

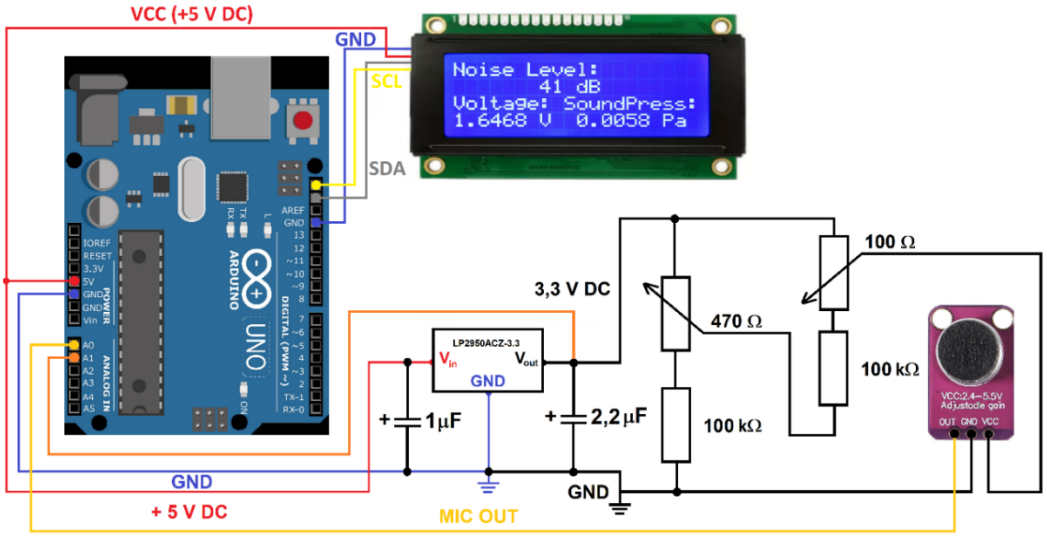


Figure 3.9. *Wiring diagram of the Arduino sound meter*

The LCD is connected to the Arduino by four wires. Two are related to the I²C interface (SCL and SDA), and the others are used to power the display (GND and 5 V DC). The 5 V voltage is also supplied to the stabilizer LP2950ACZ-3.3, which is included before the potentiometers and the microphone module. For better stability and noise elimination, the stabilizer is equipped with 1.0 µF and 2.2 µF electrolytic capacitors at the input and output according to the catalogue connection (Datasheet LP295x-N Series, 2017). A 480 Ω potentiometer, connected in series with

a 100 kΩ resistor, is used for rough calibration of the working point, and a 100 Ω potentiometer in series with a 100 kΩ resistor allows fine-tuning of the working point. The output from the microphone sensor is connected to pin A0 of the Arduino. In addition, the output voltage from the stabilizer is also connected to the pin A1. This voltage had to be recorded for diagnostic purposes during the program construction, mainly because of the quality control of the stabilizer used, and we kept the wire connected just in case, even in the final version of the wiring.

We will make the connection using wires, while it is best to use wires with DuPont type M-F (Male-Female) terminals to connect the microphone sensor to the Arduino, as seen in Fig. 3.10. In addition to these, we will need to make some connections using soldering iron and tin.

***Figure 3.10**. Wires with M-F terminals for connection to Arduino*

So that we can imagine it better, the actual connection of the Arduino, microphone sensor, potentiometers, and LCD is in Fig. 3.11.

In the upper part of the picture, you can see the back part of the transparent cover of the box, to which the LCD and both potentiometers are attached. The Arduino Uno is fixed in the lower part of the assembly box. The microphone sensor is on the left wall (from our point of view), and the power and USB connectors are in the lower part of the picture. The 3.3 V DC stabilizer and capacitors on a small circuit board are attached to the wires, not directly to the box.

The microphone sensor connection to the side wall of the mounting box with the glued anti-slip pads is shown in Fig. 3.12.

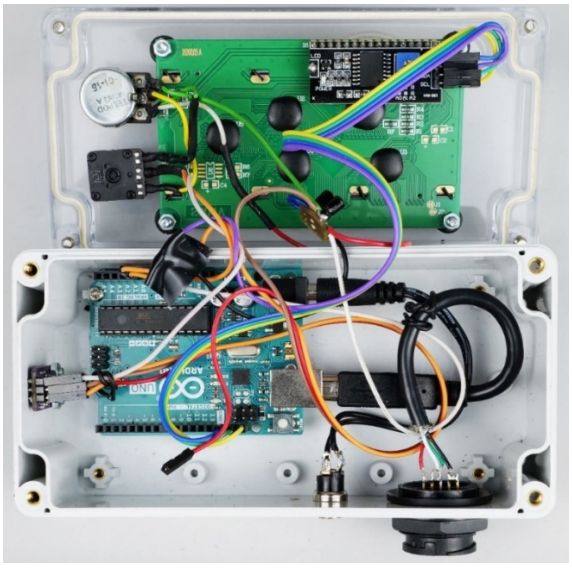


***Figure 3.11**. A look inside the Arduino sound level meter*

***Figure 3.12**. Mounting the microphone sensor*

We describe in the following text the sound level meter program and a brief guide to working with it.

3.2.3. Arduino sound meter programming and program description

We have created two programs for the Arduino sound meter, which differ slightly in the final processing of the measured sound intensity. The first version of the program calculates a moving average from five measured sound intensity values. This procedure is commonly used in data processing to eliminate noise in the measured values. In that case, the disadvantage is that the sound level meter cannot react to rapid changes in sound. If we need to capture faster changes, it is better to use the second version of the program, which calculates the median.

In the following text, we will describe the moving average version of the program in more detail and then point out the differences in the median version.

In the first part of the program (Fig. 3.13), it is necessary to load the libraries for I²C communication and the display, and to be sure, the math library since we will need the decimal logarithm function. It is also necessary to define the needed constants.

NB: The program will ignore the text following the double slash symbol "//."

RMS_Decibelometer_FIN_Cal_Ser_Mov_Av_2.ino

```
1   // Simple Noise Meter, Martin Hruska, 2023, moving average
2   // Launch the necessary libraries
3   #include <Wire.h> // Library for i2C communication
4   #include <math.h> // Mathematical library (probably not necessary)
5   #include <LiquidCrystal_I2C.h> // Library for the LCD display
6   LiquidCrystal_I2C lcd(0x27, 20, 4); // Define display properties
7
8   const float referenceVoltage = 5.0; // Reference voltage, to calculate the current value from the microphone
9   const float referenceSoundPressure = 0.00002; // Minimum pressure detectable by the microphone (Pa) - probably
10  const float maximumSoundPressure = 20.0; // The highest possible sound pressure (Pa) - pain threshold
11  const float mindecibel = 25.0; // The minimum value in decibels that makes sense to measure
12  const float maxVoltage = 3.26; // Calibration voltage = 3.26 V / 2 = 1.63000 V!!!
13
```

***Figure 3.13**. The first part of the code with initialization of libraries and input constants*

In the next part of the program, we define an index for calculating the effective value of the voltage from the microphone sensor. The program calculates the effective value from 3.500 values of the measured voltage (Fig. 3.14). Also, a field for calculating the moving average is defined in the introductory part of the program.

```
// Index for calculation of root mean square voltage:
float i=0;
float samples = 3500; // Number of samples!!!
float sum2 = 0;
float adcValue = 0;
float Voltage;
float result2 = 0;
unsigned long cas2;

// Array size for storage of values for calculating the moving average from 5 RMS Voltage values
const int arraySize = 5;
// Array for storing decibelValues
int decibelValues[arraySize];
// Index for storing new values
int index = 0;
// Variable for storing the sum of values
int sum = 0;


```

Figure 3.14. *Introduction of index and field to calculate the effective value and moving average*

In the initialization part of the program (void setup()), we initialize serial communication, the LCD, and the command to take the text that will not change during the program's run (Fig. 3.15).

```
void setup() {
  Serial.begin(9600); // Initialization of serial communication
  Wire.begin(); // Initialization of I2C communication
  lcd.init(); // Initialization of LCD display
  lcd.backlight(); // Switching on the LCD backlight
  lcd.setCursor(0, 0); // The text that will not be changed during the measurement will be displayed
  lcd.print("Noise Level:");
  lcd.setCursor(0, 2);
  lcd.print("Voltage:");
  lcd.setCursor(9, 2);
  lcd.print("SoundPress:");
  Serial.print("dB ");
  Serial.println();
}

```

Figure 3.15. *The initialization part of the program*

In the basic run of the *void loop()* program in Fig. 3.16, Arduino first reads a digital value in 10-bit resolution from the microphone sensor. Subsequently, it is converted to voltage, squared, and added to the other amplified voltage values. It repeats this process about every 8 μs until it has counted 3.500 voltage values. The sum is divided by the number of samples and squared according to relation (3). The following is the calculation of sound pressure, sound intensity value, and calibrated sound intensity value according to the UNI-T UT352 sound level meter, which is briefly described below in the next section.

```
void loop()
{
  adcValue = analogRead(A0); // Reading a value from the microphone sensor
  if (abs((micros()-cas2))>=100) // For code stability reasons
  {
    cas2=micros();
    float Voltage = adcValue * (referenceVoltage / 1023); // Convert digital value at input A0 to voltage, 10 bits input
    sum2 = sum2 + Voltage * Voltage; // Calculating the square of the voltage, adding to the total sum
    i=i+1;
  }
    if (i == samples)
    {
      result2 = sum2 / samples;
      result2 = sqrt(result2);
      sum2=0;
      i=0;

    float RMSVoltage = result2; // RMS Voltage output
    int sensorValue = analogRead(A1); // Reading the voltage value at the output of the stabilizer, active but not needed now
    float micsupplyvoltage = sensorValue * (referenceVoltage / 1023); // Calculation of the voltage at the output of the stabilizer, active but not needed now
    float RMSpressure = maximumSoundPressure * abs(RMSVoltage - maxVoltage/2)  / (maxVoltage/2); // Calculation of RMSpressure (Pa)
    float decibelRMS = abs(20 * log10(RMSpressure / referenceSoundPressure)) ; // Calculation of noise level (dB), uncalibrated
    float decibelRMSCal = decibelRMS - 0.0098*decibelRMS*decibelRMS + 1.5724*decibelRMS - 40 ; // Calibrated value according to UNI-T UT352

```

***Figure 3.16**. Calculation of the sound intensity value*

In the final program part (Fig. 3.17), a moving average is calculated from the five sound intensity values measured in this way. In doing so, it is necessary to ensure that after its calculation, the last number is always discarded from the data field, and a new one is loaded. After its calculation, only the list of measured sound intensity, effective voltage, and effective sound pressure values are displayed on the LCD. Using the serial connection via the USB port, recording the sound intensity value in decibels is possible on the computer. Values are displayed every second.

```
    // Input of the decibelRMSCal value to the moving average:
  int decibel = decibelRMSCal;
  // Removing the oldest value from the sum
  sum -= decibelValues[index];
  // Adding a new value to the sum
  sum += decibel;
  // Saving a new value to the array
  decibelValues[index] = decibel;
  // Index incrementation (respectively increase the number of index by 1)
  index = (index + 1) % arraySize;
  // Calculation of the moving average
  float average = (float)sum / arraySize;

  delay(100); // For better stability of the code
  lcd.setCursor(0, 1); // Setting the display cursor to the initial position
  lcd.print("               "); // Print clear characters

  if (average <= mindecibel) // Conditional statement - if the value of average is less than 25 dB, do not display it and print only "25 dB".
    { lcd.setCursor(7, 1);
      lcd.print(mindecibel, 1);
      Serial.print(mindecibel, 1);
      Serial.println();
    }
  else if (average > mindecibel) // Conditional statement - if the value of average is higher than 25 dB then display actual value of average.
  { lcd.setCursor(7, 1);
      lcd.print(average, 1);
      Serial.print(average, 1);
      Serial.println();
    }
  lcd.print(" dB");
  lcd.setCursor(0, 3);
  lcd.print("                    ");
  lcd.setCursor(0, 3);
  lcd.print(RMSVoltage, 5); // Display RMSVoltage value - important for setting the operating point!!!
  lcd.print(" V");
  lcd.setCursor(11, 3);
  lcd.print(RMSpressure, 4); // Display RMSPressure value
  lcd.print(" Pa");
  delay(143); // Delay - values are recorded every second
  }
}
```

***Figure 3.17**. Calculation of the moving average and output of values to the LCD and the computer*

As mentioned above, if we need to record faster changes in sound intensity, it is more appropriate to calculate the median from the measured intensity values.

Compared with the moving average program, the difference is that the parts of the code that define the initial conditions and variables for calculating the moving average are replaced by the code for the median. As seen in Fig. 3.18, the median is chosen as the middle value after sorting in the case of an odd values number. If the number of values is even, the program calculates the average of the two middle values.

```
23  // Number of values in array for Median Calculation
24  const int arraySize = 5;
25
26  float decibel[arraySize];
27
28  float calculateMedian(float* array, int number) {
29    // Arrangement of values in the array ascending
30    for (int i = 0; i < number - 1; i++) {
31      for (int j = 0; j < number - i - 1; j++) {
32        if (array[j] > array[j + 1]) {
33          float temp = array[j];
34          array[j] = array[j + 1];
35          array[j + 1] = temp;
36        }
37      }
38    }
39
40    // If the number of values is even, calculate the average of the two middle values
41    if (number % 2 == 0) {
42      return (array[number / 2 - 1] + array[number / 2]) / 2.0;
43    }
44    // If the number of values is odd, calculate the value on the mean index
45    else {
46      return array[number / 2];
47    }
48  }
```

***Figure 3.18**. Program input code for calculating the median of five values*

The second change in the code is found in the *void loop()* function right after the calculation of the calibrated sound intensity value, where according to the conditions defined above in Fig. 3.18, the reading of the sound intensity value (decibelRMSCal) and the own calculation of the median of the five values occur (Fig. 3.19). The result is the *median* variable. The following procedure for listing values is identical in both programs.

```
90  // Median calculation - reading values of decibelRMSCal into the array
91   for (int i = 0; i < arraySize; i++) {
92     decibel[i] = decibelRMSCal;
93     delay(50); // For better stability of the code
94   }
95
96   // Median calculation
97   float median = calculateMedian(decibel, arraySize);
```

***Figure 3.19**. Calculation of the median from five values*

For the sake of interest, let us mention that when creating the code for the Arduino sound meter, we partially used the ChatGPT language model developed by OpenAI and based on the GPT-3.5 architecture. Artificial intelligence alone could not generate a more complex code but significantly helped us create some of its (for us) more challenging fragments and observe the syntax of the Wiring language designed for Arduino microcontrollers.

3.2.4. Arduino sound meter calibration

Since we had the UNI-T UT352 sound level meter (Fig. 3.20a) at our disposal, we decided to use it to calibrate the sound level meter we created so that it displays the measured sound intensity values as accurately as possible. Our sound level meter does not contain weight filters, so we used the so-called white noise, which includes the entire spectrum of sound frequencies. This type of sound is also advantageous for measurements in a closed room because we eliminate problems with forming standing waves. The sound in the room reflects from different surfaces and interferes with each other. If there are knots and oscillations in the space between the sound source and the sound level meter, they will invalidate the sound intensity measurement near its source.

We generated white noise as a lossless *.wav file in the freely available audio program for sound processing Audacity. We recorded it in a mini-MP3 player (Fig. 3.20b). The Mini-MP3 player or mini-MP3 speaker in the shape of a cube with an edge length of approximately 3.3 cm has its power source, USB, and a 3.5 mm stereo jack connector.

a) b)

***Figure 3.20**. Tools used for calibration – a) sound level meter UNI-T UT352 and b) mini-MP3 player*

We performed the calibration by moving the mini-MP3 player with white noise towards the sound level meters from 5 cm to 1 m while recording sound intensity data in decibels on both devices for 11 distances. For each distance, we calculated the difference between the device values and plotted these differences on a graph. On the x-axis are the data from the UNI-T UT352 device; on the y-axis are the differences between our and the commercial device. We interpolated the obtained dependence with a polynomial dependence of the second degree, as seen in Fig. 3.21.

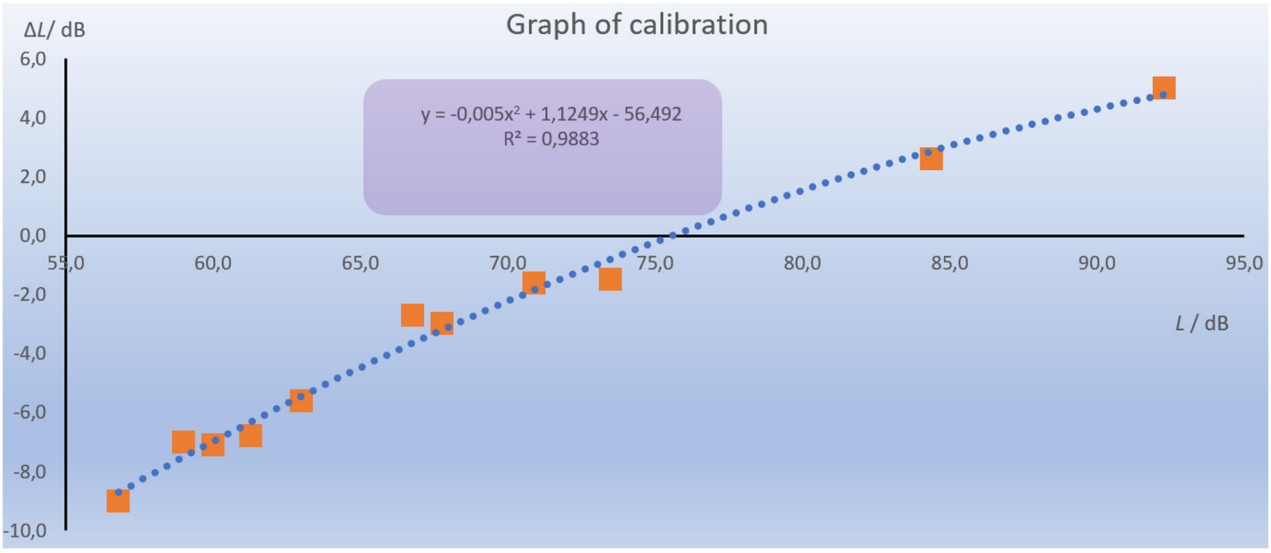


***Figure 3.21**. Graph of sound intensity deviations from UNI-T UT352 device with interpolated calibration dependence*

After re-measurement, it was necessary to manually adjust the mentioned trend equation, and only then was its adjusted version incorporated into both program versions. Due to the absence of frequency weighting filters in the Arduino sound meter, this calibration will only work in the case of white random noise as a noise source. When measuring other sound stimuli, the measurement results will vary. Another critical factor affecting the measurement's accuracy is the microphone's quality and the electronics used. Commercial noise meters use high-quality measuring microphones with precisely defined parameters and more complex and significantly higher-quality electronics.

3.2.5. Data collection from a sound level meter via a computer

As we have already mentioned in the text during the construction of the meteorological station, for the data collection itself using a computer, we can advantageously use one of the freely available programs that can record data sent by serial transmission from the Arduino via the USB connector to a text file. The *CoolTerm* program proved the best for us; its latest version is freely available online (*CoolTerm*, 2022).

Since we dealt with the issue of data collection in more detail during the creation of the meteorological station, while we also prepared a detailed manual on how to record data and get it into the MS Excel spreadsheet, we will not deal with it in the following text.

3.2.6. How to work with an Arduino sound meter

When working with a sound level meter, it is necessary to proceed as follows:

- We can power the device with an adapter from 6 V to 15 V DC or via a USB connector, using a USB adapter or a computer, in case we want to record the measured data.
- After switching on, it is necessary to set the working point. We achieve this with the help of two potentiometers in the right part of the device. The setting should be done in a quiet room to eliminate surrounding noises as much as possible. Using coarser and finer tuning, we try to set the voltage level on the display to (1.63000 ± 0.00005) V. When the working point is reached, the Arduino sound meter program is set not to show a value lower than 25 dB. Since the electronic noise is very significant at such close values to the working point, the device would still show incorrect values.

- Let the sound level meter heat up after switching on; otherwise, the working point will shift significantly. After turning it on, we can use the Arduino sound level meter for short measurements. If we want to record the noise of the surrounding environment for tens of minutes or more, it is better to wait about five minutes for the electronics to warm up to the operating temperature.
- Suppose we record data using a computer, e.g., through the CoolTerm program. In that case, it is best to set the working point of the sound level meter only after turning on the serial communication, that is, after starting the measurement and recording of values. Otherwise, if we set the working point before beginning the serial transfer, it will move after it starts.
- For long-term measurement, the working point of the sound level meter needs to be fine-tuned from time to time.

3.3. Tasks

3.3.1. Noise measurement in the room

Noise in a room, whether at home or in the workplace, can be tiring and stressful. For example, in an activity in which communication is an essential part of the work performed and in which there are high demands on accuracy, speed, or attention, the sound intensity level should not exceed the value of 50 dB. How noisy is it at work or home? Let us measure it.

Aim of the measurement: The experiment aims to use a simple sound meter to measure the sound intensity level in the work or home environment for several tens of minutes and assess the noise level in the room.

3.3.1.A. Measurement procedure

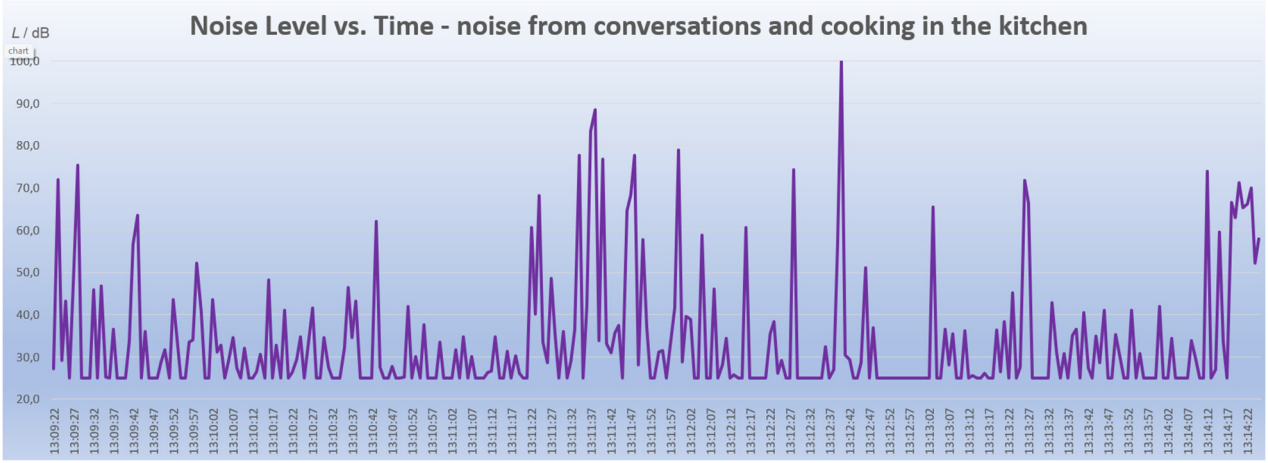
- The experimental setup consists of an Arduino sound meter with a computer. We upload a program with the measurement of the median to the Arduino sound meter (it is more suitable for rapid changes in noise).
- We connect the sound level meter to the computer in the room where we want to record the data.
- We start the *CoolTerm* program and record sound intensity level data into a text file.
- Now, we proceed to the calibration itself – to calibrate the sound level meter, we need a moment of silence in the room.
- After calibration, note the beginning of the measurement, as well as the data in the text file that we will not need and that we recorded before the calibration (the sound level meter must be calibrated only after starting data storage. Otherwise, it's working point will move and the sound level meter will show incorrect values).
- We record data, e.g., half an hour, while continuously checking the setting of the working point when the room is quiet. If necessary, we will fine-tune it.
- We will end the measurement after about 30 minutes or up to an hour (we have recorded around 1800 to 3600 sound intensity level values).

3.3.1.B. Data analysis

Transfer the data from the text file to Excel and build a table and chart – you can see a sample chart in the image below.

If you have recorded the beginning of the measurement, plot the time on the x-axis in hh:mm:ss format.

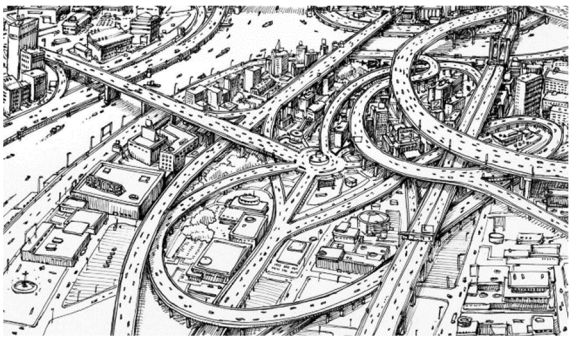
Observe the time course of the noise in the room and try to answer the following questions (either using a graph or directly from a table with measured values).



Date: _____

- The maximum measured sound intensity level in the room was ____dB.
- It was measured at time from ___ : ___ : ___ to ___ : ___ : ___ .
- The minimal measured sound intensity level in the room displayed by the noise meter was ____dB.
- The difference between the maximum and minimum sound intensity levels was ___dB.
- The longest time interval when the room was quiet ($L \leq 25$ dB) occurred at the time from ___ : ___ : ___ to ___ : ___ : ___ .
- The highest increase in sound intensity level occurred at the time ___ : ___ : ___ .

3.3.2. Noise measurement in the place of residence or school



Introduction: Noise in cities, on the streets, is most often caused by traffic. Therefore, the main roads are built as bypasses to protect populated areas. Traffic noise depends mainly on the type of means of transport, as well as on the number of cars moving along the road and on their speed.

What is the noise level of the streets, roads, and intersections around your home or school? Are there places where there is too much noise?

Aim of the measurement: This experiment determines which streets around your home or school are the noisiest.

3.3.2.A. Measurement procedure

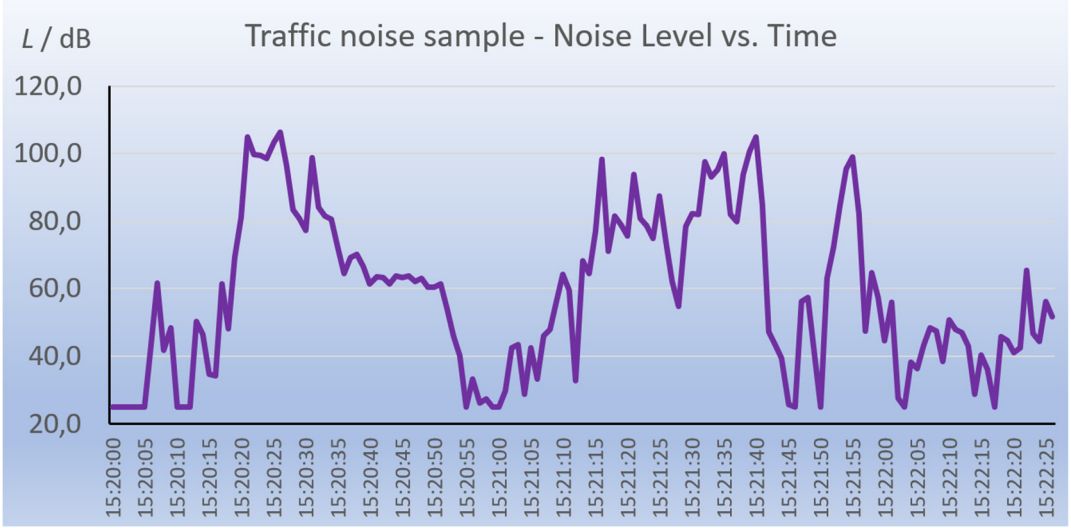
- The experimental setup consists of an Arduino sound meter with a laptop. We upload a program measuring the moving average to the Arduino sound meter.

- We connect the sound level meter to the laptop outside, where we want to record the data. If we cannot go out for various reasons (e.g., due to the weather), we will use a recording of traffic noise downloaded from the Internet, which we will play using active speakers. Various records are available, e.g., on the page: https://www.videvo.net/royalty-free-sound-effects/traffic/
- We start the *CoolTerm* program and record sound intensity level data into a text file.
- Now, we proceed to the calibration itself – to calibrate the sound level meter, we need a moment of silence in the room.
- After calibration, we note the beginning of the measurement, as well as the data in the text file that we will not need and that we recorded before calibration (the sound level meter must be calibrated only after starting data storage; otherwise, the working point will move and the sound level meter will show incorrect values).
- We record data, e.g., half an hour during a walk through the city, continuously checking the setting of the work point when the surroundings are quiet. If necessary, we will fine-tune it.
- We will end the measurement after about 30 minutes or up to an hour (we have recorded around 1800 to 3600 sound intensity level values).

3.3.2.B. Data analysis

Transfer the data from the text file to Excel and build a table and chart – you can see a sample chart in the image below. It is a recording of traffic noise, approximately 2.5 minutes long, freely available on the Internet. If you have recorded the beginning of the measurement, plot the time on the x-axis in hh:mm:ss format.

Observe the time course of noise in the exterior and try to answer the following questions (either using a graph or directly from a table with measured values).



Date: _____

- The maximum measured sound intensity level in the street was___dB.
- It was measured at time from ___ : ___ : ___ to ___ : ___ : ___ .
- The minimal measured sound intensity level displayed by the noise meter on the exterior was ____dB.
- The difference between the maximum and minimum sound intensity levels was ___dB.
- The biggest increase in outdoor sound intensity level occurred at the time to___ : ___ : ___ .

3.3.3. How does sound intensity level change with distance?

Introduction. Noise is often an unwanted sound phenomenon that adversely affects human health and psychology. Regular exposure to loud noise may have adverse effects on hearing and cause damage to the ears, so legislative standards and guidelines set maximum permitted noise levels in various environments to protect public health. Let us try how we can measure noise and its decrease with distance from the source.

Aim of the measurement: This experiment aims to measure the decrease in sound intensity with distance using a simple sound meter. Process the obtained data, for example, in the MS Excel environment, and confirm the theoretical assumption of a 6 dB decrease in noise when doubling the distance from the sound source is fulfilled.

3.3.3.A. Measurement procedure

- The experimental setup consists of an Arduino sound meter, a mini-MP3 player, stands, a tape measure, or a longer ruler for measuring distance (Fig. 3.22).
- Arrange the experiment according to the picture. Place the sound level meter and the MP3 player at 5 cm from each other. The sound level meter and the MP3 player must be at least 50 cm above the table. Otherwise, the tabletop will distort the measurements because it will reflect the sound. The measurement works best if the sound level meter and the sound source are on separate tables with a gap between them.
- You do not need to connect the sound level meter to a computer. It is enough to power it, e.g., through a 9 V DC adapter.
- Turn the sound level meter on by connecting it to the adapter and make sure the room is quiet. Using the potentiometers, carefully adjust the working point of the sound level meter so that it reads 25 dB (the moving average version of the program is best suited for measurement).
- Turn on an MP3 player with a white noise sound file installed and adjust the volume so the sound level meter reads around 90 to 100 dB.
- Record the noise value for 5 cm.
- Gradually move the MP3 player five centimetres up to 1 m and always write down the noise level in decibels. If the values do not decrease, or, on the contrary, decrease too much, turn off the MP3 player and fine-tune the working point value, or repeat the measurement.
- Record distance values and sound intensity levels in a table, e.g., in Excel.
- Try to repeat the measure under different conditions, e.g., leave both the sound source and the sound level meter on the table and increase the distance between them, replace the sound source with another one (e.g., more speakers), or with an appliance such as a hair dryer, vacuum cleaner, etc. Try using a different audio signal for the measurement (e.g., generate a sine signal with a frequency of 1 kHz in Audacity, etc. and compare the results. At the same time, try to answer the questions in the Data analysis section.

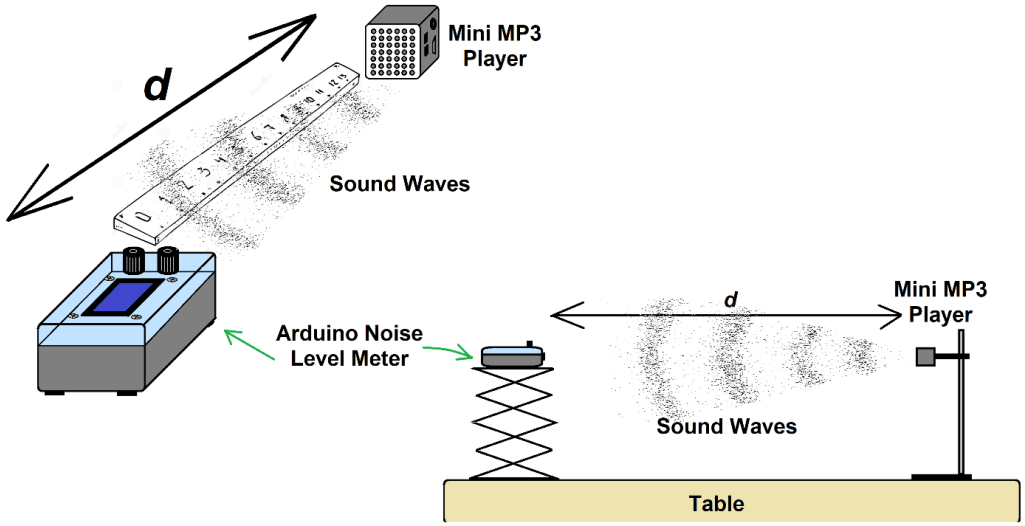


Figure 3.22. *The arrangement of the experiment*

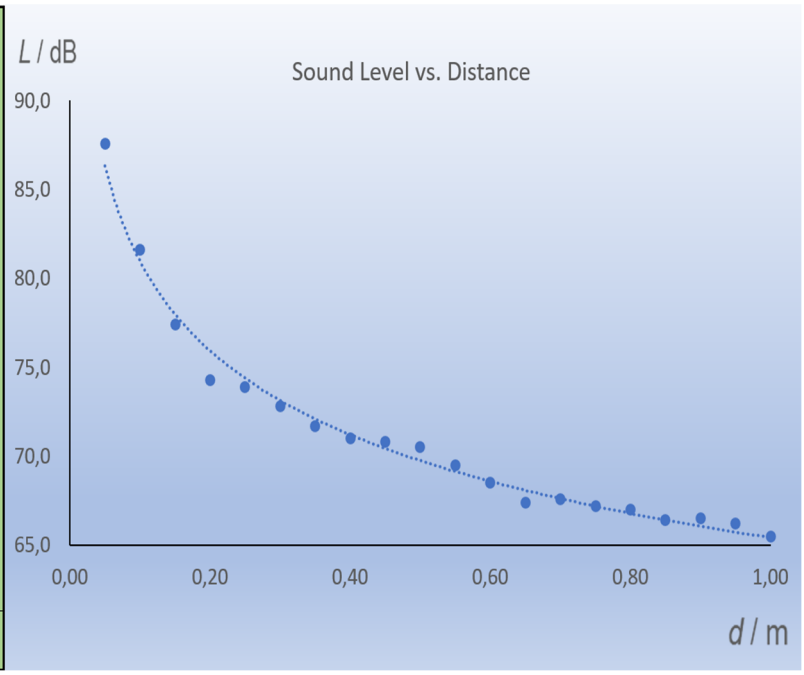
3.3.3.B. Data analysis

Construct a table and graph from the recorded values, as shown in the figure below. Also, calculate the difference between the noise intensity levels for twice the distances, as seen in the figure with the table and graph. For example, a value of 6.0 dB corresponds to the difference in noise at distances of 5 cm and 10 cm; a value of 7.3 dB represents the difference between distances of 10 cm and 20 cm, etc.

Record the resulting values of the level difference in the table as well.

As you can see, the ideal rule for the propagation of noise from a point source is fulfilled approximately only in our case – the average drop at twice the distance is 4.63 dB.

n	*d* / m	*L* / dB	ΔL / dB	
1	0,05	87,6	$L_1 - L_2 =$	6,0
2	0,10	81,6		
3	0,15	77,4	$L_2 - L_4 =$	7,3
4	0,20	74,3		
5	0,25	73,9	$L_3 - L_6 =$	4,6
6	0,30	72,8		
7	0,35	71,7	$L_4 - L_8 =$	3,3
8	0,40	71,0		
9	0,45	70,8	$L_5 - L_{10} =$	3,4
10	0,50	70,5		
11	0,55	69,5	$L_6 - L_{12} =$	4,3
12	0,60	68,5		
13	0,65	67,4	$L_7 - L_{14} =$	4,1
14	0,70	67,6		
15	0,75	67,2	$L_8 - L_{16} =$	4,0
16	0,80	67,0		
17	0,85	66,4	$L_9 - L_{18} =$	4,3
18	0,90	66,5		
19	0,95	66,2	$L_{10} - L_{20} =$	5,0
20	1,00	65,5		
			Average:	4,63 dB



Answer the following questions based on (your own) measured data.

Date: _____

- The maximum measured sound intensity level was ___dB.
- The minimum measured sound intensity level was ___dB.
- The difference between the maximum and minimum sound intensity levels was ___dB.
- We recorded the highest drop in the sound intensity level at the longest / smallest distance between the sound source and the sound level meter (cross out if applicable).
- The average difference in sound intensity levels when the distance between the source and the sound level meter doubled was ___dB.
- This difference value *is exactly / is approximately / is not* by the theoretical assumptions (cross out if not applicable).

References

- Degro, J. (2009). Školské experimenty s hlukomerom. Košice, UPJŠ. [online] [cit. 2023-06-27] <https://physedu.science.upjs.sk/degro/pokus/expzvuk/ExpZvukHl.pdf>.
- Vyhláška č. 549/2007 Z. z. (2007). Vyhláška Ministerstva zdravotníctva Slovenskej republiky, ktorou sa ustanovujú podrobnosti o prípustných hodnotách hluku, infrazvuku a vibrácií a o požiadavkách na objektivizáciu hluku, infrazvuku a vibrácií v životnom prostredí. [online] [cit. 2023-06-27] <https://www.zakonypreludi.sk/zz/2007-549>.
- Gray, R. M.; Neuhoff, D. L. (1998). Quantization. IEEE Transactions on Information Theory. Institute of Electrical and Electronics Engineers (IEEE). 44 (6): 2325–2383.
- Adafruit (2023). [online] [cit. 2023-06-27] <https://www.adafruit.com/product/1063>.
- Datasheet LP295x-N Series. (2017). [online] [cit. 2023-06-28] <https://www.ti.com/lit/ds/symlink/lp2950-n.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1687892430482&ref_url=https%253A%252F%252Feu.mouser. com%252F>.
- Calculation of Frequency Weightings (1996). [online] [cit. 2023-06-28] <https://web.archive.org/web/20061210125050/http://www.measure.demon.co.uk/Acoustics_Software/a_weight.html>.
- Frequency-Weightings for Sound Level Measurements (2023). [online] [cit. 2023-06-29] <https://www.nti-audio.com/en/support/know-how/frequency-weightings-for-sound-level-measurements>.
- Sound Level Frequency Weightings – A, B, C, D, Lin, Z. (2023). [online] [cit. 2023-06-29] <https://www.nti-audio.com/en/support/know-how/frequency-weightings-for-sound-level-measurements>.
- Výber spotrebiča podľa hladiny hluku (2019). [online] [cit. 2023-06-29] <https://www.topbyvanie.sk/magazin/vyber-spotrebica-podla-hladiny-hluku>.

CHAPTER 4. HOW TO MAKE A GEIGER-MÜLLER COUNTER USING A MICROCONTROLLER?

This chapter was written by Martin Hruska
from the Matej Bell University of Banska Bystrica, Slovakia

4.1. Theory

If we need to measure ionizing radiation, we can construct a simple Geiger-Müller (GM) detector (Fig. 4.1) with a liquid crystal display (LCD) based on an Arduino Uno microcontroller.

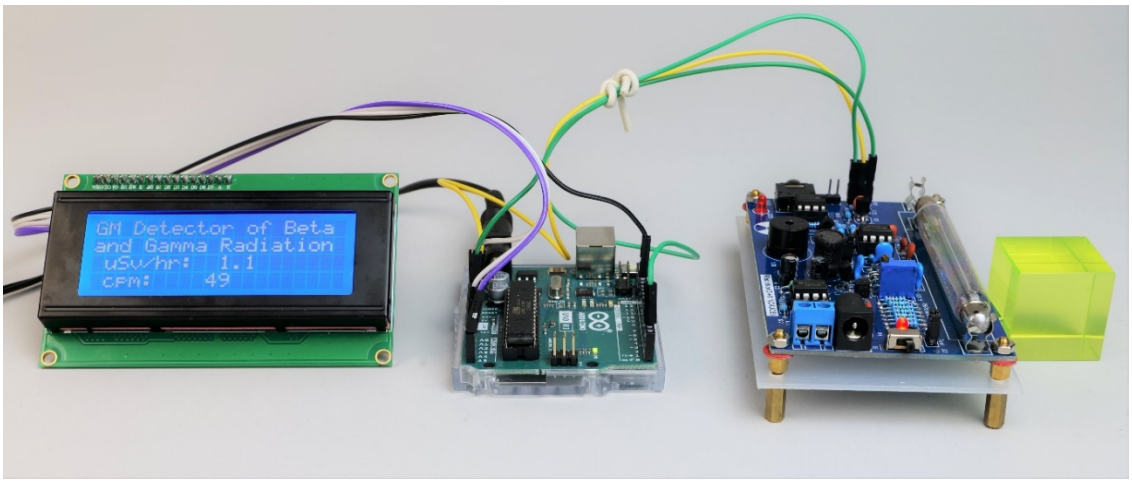


Figure 4.1. *Simple GM detector with Arduino Uno microcontroller; from left to right: LCD, Arduino Uno, GM detector, uranium glass cube*

The data on the measured quantities, briefly described below, can be easily displayed on the display, or recorded in a text file using the serial connection of the Arduino GM detector to the computer with a USB cable.

We designed the GM detector to be easily constructed using several affordable modules (Arduino, display, GM detector set with tube), and it is easy to operate, especially during demonstration experiments with ionizing beta and gamma radiation.

4.1.1. Ionizing radiation and its measurement

Ionizing radiation is high-energy radiation with enough energy to remove an electron from an atom or molecule. Such radiation can cause chemical changes in cells and damage DNA, leading to cancer. Alpha radiation (formed by helium nuclei) and beta radiation (formed by electrons or positrons) can ionize directly. Gamma rays (high-energy photons originating in an atom's nucleus) ionize indirectly. Our Arduino detector can detect beta and gamma radiation.

The same doses of different types of radiation can have different biological effects, but it is generally true that you must protect yourself from ionizing radiation. Most often, we protect ourselves from its effects by maintaining a sufficient distance from the source of radiation, shielding ourselves with various materials and time – we try to shorten our stay in space with radiation to the shortest possible time.

The issue of the biological effects of ionizing radiation is very complex. Therefore, we do not even aim to define them in detail in the following text and only list the most important from our point of view. For example, their respective definitions can be found in the literary sources mentioned below and in the currently valid legislation.

The primary quantity expressing the biological effect of individual types of radiation using a radiation weighting factor is the *equivalent dose*, expressed in units of Sievert (Sv).

Another quantity that expresses the biological effect of radiation on a person is the *dose equivalent*, introduced mainly for the needs of personal dosimetry. It is also expressed in units of Sievert (Sv) and describes the biological effectiveness of different types of radiation in soft tissue at a defined depth using a *quality factor*.

The dose equivalent is, e.g., in the Slovak Hydrometeorological Institute (SHMÚ) monitoring system, measured in the form of dose equivalent input in units of nSv/h, i. e. nano Sievert per hour (SHMÚ – Rádioaktivita, 2023). The GM detector described below also displays this quantity – the dose equivalent input.

Another quantity related to the biological effects of radiation is the *effective dose*. The effective dose is calculated as the sum of the weighted equivalent doses in all organs and tissues, multiplied by the relevant tissue weighting factor – a number expressing the different biological effects of radiation (Nikodemová and Cabáneková, 2009).

Tab. 1 shows an overview of annual effective doses of radiation from various exposure locations caused by natural sources of ionizing radiation.

Table 1: *Overview of the annual effective doses of radiation from various exposure sites (Office of Nuclear Supervision of the Slovak Republic, 2022)*

Source of radiation		Annual effective dose [mSv]	
		Average	Typical range
Cosmic radiation	Directly ionizing and photon component	0,28	
	Neutron component	0,1	
	Cosmogenic radionuclides	0,01	
	Together	0,39	0,3 – 1,0
External terrestrial radiation	When staying outdoors	0,07	
	When staying inside buildings	0,41	
	Together	0,48	0,3 – 1,0
Inhalation	Uranium and thorium decay series	0,006	
	Radon ^{222}Rn	1,15	
	Radon ^{220}Rn	0,1	
	Together	1,26	0,2 – 10
Ingestion (intake by eating)	Potassium ^{40}K	0,17	6

It must be said that there is no boundary between a "safe" and a "dangerous" dose – the risk of long-term effects of ionizing radiation is based on probability.

The GM detector, used to detect ionizing radiation, is named after the physicists Hans Geiger and Walther Müller, who developed it in the 20th century. GM detectors are based on the principle of gas ionization. They contain a gas filling that gets ionized when ionizing radiation passes through the detector. Ionization of the gas leads to the formation of electrical impulses detected and recorded by the detector. Detectors record the number of pulses in a fixed time, e.g., per second (cps – Counts Per Second) or per minute (cpm – Counts Per Minute), as well as our detector. The principle of the GM detector operation is generally known

and described on many websites in the form of texts, images, animations, and videos e.g. (Abbas, 2022).

Recalculating the number of pulses per dose equivalent, or to the dose equivalent input, is not easy since there is no simple conversion factor – a number by which it would be possible to realize it. Therefore, it is best to calibrate the given GM detector according to another reliable detector or to rely on the manufacturer's data of the specific tube for the GM detector.

More about the issue of ionizing radiation and its measurement can be found in many sources that are also available on the Internet (Úrad jadrového dozoru SR, 2022), (Holá, 2009), (WHO, 2022), (Ryan, 2012), (Canadian Centre for Occupational Health and Safety, 2023), (United States Nuclear Regulatory Commission, 2020), etc.

4.1.2. A brief description of the individual components of the GM detector

We use the Arduino GM detector at the Physics Department of the Faculty of Natural Sciences, Matej Bel University in Banská Bystrica, as a demonstration device placed in display cases in the faculty corridor. Therefore, the GM detector is not installed in a mounting box but consists of separate elements.

The LCD, the Arduino Uno microcontroller, and the GM detector with a tube are connected by wires. A small piece of uranium glass (Fig. 4.2) serves as a radioactive radiation source. UV LEDs illuminate it. (Uranium oxide fluoresces green under UV-A radiation with a maximum wavelength of 360 nm).

Using the GM detector for measurement in the laboratory in such a configuration is dangerous, as there can be a high voltage of around 500 V at the ends of the detector tube!!! To use it in the laboratory, we recommend covering the GM detector with transparent plexiglass or placing it in a mounting box so that the box contains a cut-out for the active part of the GM detector tube; there should be no excess material between the radioactive emitter and the detector tube.

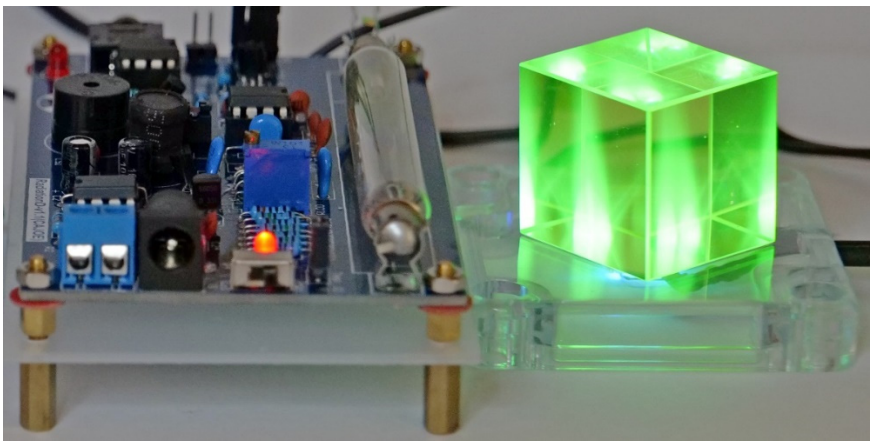
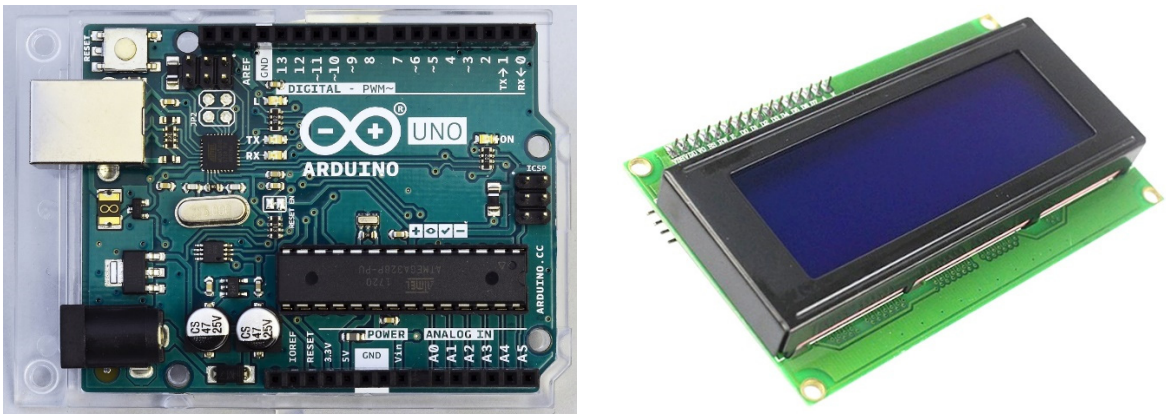


***Figure 4.2**. View of the GM detector with the tube and the cube of uranium glass backlit by LED UV-A diodes*

- Arduino Uno, LCD 20 x 4 with blue backlight

As mentioned, the simple Arduino GM detector construction requires an Arduino Uno microcontroller and a liquid-crystal display (Fig. 4.3a, b). These components are discussed in more detail in Chapter 5, which describes the design of the meteorological station, so they will not be discussed in detail.



a) b)

***Figure 4.3**. Arduino Uno microcontroller (a) together with LCD (b)*

- GM detector

***Figure 4.4**. RadiationD-v1.1(CAJOE) GM detector with J305β tube*

The GM detector designated as *RadiationD-v1.1(CAJOE)* (Fig. 4.4), with a coaxial cylindrical detection tube, can detect beta and gamma radiation with specific energy, depending on the GM tube used. The GM J305β tube contains a tin oxide cathode. The tube is filled with an inert gas with a halogen admixture for pulse quenching.

The total size of the background reaches a value of about 25 cpm, and its lifetime is more than 1.10^9 pulses. The limit value of the operating voltage is 550 V, and the working voltage range is 380 V to 450 V. Since it is a cheap "Noname" GM tube of Chinese production, we will prefer to read about the technical characteristics on various business pages. A comparison of the parameters of several GM tubes is available online, e.g. (IoT-devices, 2023).

A similar situation exists in the case of the GM detector. We can find the basic parameters mainly on the websites of Chinese sellers (Banggood.com, 2023). It is possible to purchase several different versions of the GM detector, which (like our device) can communicate with various microcontrollers or even with minicomputers, such as Raspberry Pi. For example, more detailed information about the specific connection and functionality of the related GM detector Libelium with the same GM tube J305β (Laquai, 2014) is helpful.

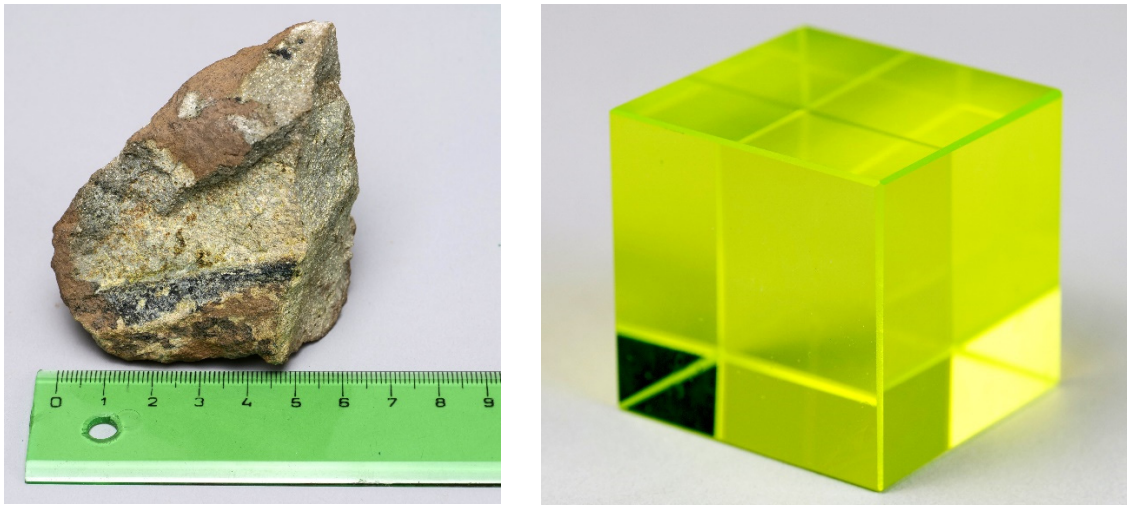
The GM detector we use communicates with the Arduino microcontroller via pin VIN (and, of course, also GND), which must be connected to pin D2 of the Arduino.

This pin is set to input in the program and is used to record the pulses of the GM detector.

As we will explain below when commenting on the program for the Arduino GM detector, the Arduino reacts to a voltage drop on pin D2 or its falling edge – a drop in voltage causes the given variable to increase by one. The GND and 5 V DC pins power the GM detector. Of course, other external sources, e.g., an accumulator or batteries, can also power the detector.

- Uraninite mineral (pitchblende) and uranium glass

We decided to use natural radioactivity sources for easy radioactivity measurements. These are primarily minerals that contain uranium dioxide, such as uraninite (the older name pitchblende). The mineral we used (Fig. 4.5a) is a sandstone with uranium mineralization (uraninite), forming a black oblique band at the bottom of the mineral. The age of the rock is estimated to be Permian, i.e., about 290 million to 250 million years ago. The rock was found in Slovakia. We also use uranium glass (Fig. 4.5b) for demonstration purposes, containing uranium dioxide. According to generally applicable legislation, uranium glass may not contain more than 1 % (by weight) of uranium (Státní úřad pro jadernou bezpečnost, 2004).



a) b)

***Figure 4.5**. Uraninite mineral (a) sample of uranium glass (b)*

It is interesting that nowadays, depleted uranium is used for the production of uranium glass, in which the ratio of isotopes ^{238}U : ^{235}U = 40 : 1 because the isotope ^{235}U is used in nuclear power plants and also in the military industry. In uranium glass produced before the Second World War, the ratio of these isotopes is 25 : 1 (Tenzler, 2016).

Due to the small content of uranium, the glass emits radioactive alpha, beta, and gamma radiation, which arise spontaneously during the conversion of radionuclides, the so-called uranium or uranium-radium decay series (United States Environmental Protection Agency, 2023). Its first member is the uranium isotope ^{238}U, which makes up 99.27 % of all-natural uranium. The mineral uraninite, used for measurement, contains more natural uranium than uranium glass and is a source of stronger ionizing radiation.

Interestingly, gamma radiation in the uranium decay series is emitted mainly by the decay products of ^{226}Ra. Therefore, the presence of uranium in rocks is determined indirectly by measuring gamma radiation (Matolín, 1970).

4.2. Laboratory setup

4.2.1. Description of the connection and construction of the Arduino GM detector

The connection diagram of the Arduino Uno microcontroller with LCD and GM detector is in Fig. 4.6. As already mentioned in the description of the meteorological station, as well as in the description of the sound level meter, the LCD uses the I²C serial interface to communicate with the Arduino Uno microcontroller.

Therefore, data transmission is ensured using only two wires. The wire marked as SCL (Serial Clock) is used for clocking the communication, and the other wire marked as SDA (Serial Data) is used for data transmission. In addition, the display is powered using the GND and 5 V DC wires.

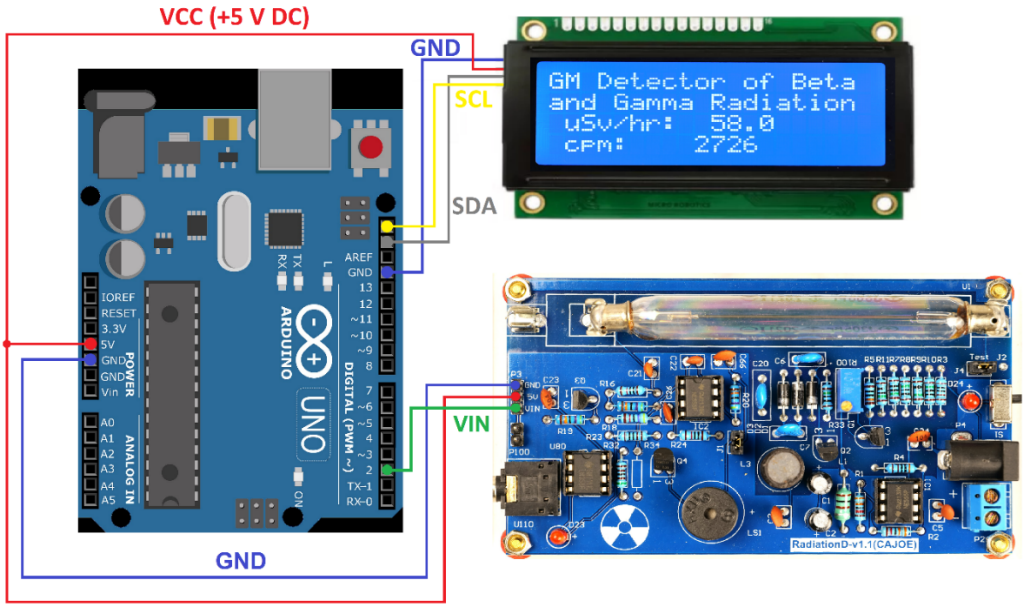


***Figure 4.6**. Connecting the Arduino GM detector*

The GM detector requires three wires to connect to the Arduino – two of them (GND and 5V DC) for power supply, and the third (VIN), which connects to the digital pin of the Arduino (D2), is used to count the pulses of the detector.

The connection will be made using wires; the wires with DuPont type M (*Male*) and F (*Female*) terminals are the best for connection to the Arduino. An example of such wires is in the chapter related to the meteorological station and a simple Arduino sound meter.

4.2.2. Programming the Arduino GM detector and description of the program

The Arduino program of the GM detector periodically measures the number of pulses (cpm) for 10 seconds and calculates the dose equivalent input in units of μSv/h. These values are then displayed on the LCD and written out via serial transmission to the computer.

In the first part of the program (Fig. 4.7), the libraries *Time.h* and *LiquidCrystal_I2C.h* are loaded, which are necessary for working with time and controlling the LCD.

geiger.ino

```
// GM Detector of Beta and Gamma radiation, Martin Hruška, 2023
// Launch the necessary libraries
#include <Time.h>
#include <TimeLib.h>
#include "LiquidCrystal_I2C.h"
LiquidCrystal_I2C lcd(0x27,20,4);
// Connect the GND pin on Arduino to the GND pin on the Geiger counter.
// Connect the 5V pin on Arduino to the 5V pin on the Geiger counter.
// Connect the VIN pin on the Geiger counter to the D2 pin on Arduino.

unsigned long counts;              // variable for GM Tube events
unsigned long previousMillis;      // variable for measuring time
#define LOG_PERIOD 10000           // count rate is 10000 ms or 10 s
#define usv_multiplier 0.05        // Due to Vernier Digital Radiation Monitor TH30044
#define cpm_multiplier 2.3456      // For the J305β tube

void impulse() {
  counts++;
}
```

***Figure 4.7**. The introductory part of the Arduino GM detector program*

Note: The double slash symbol "//" in the program means that the text after will be ignored. Therefore, we can write explanatory comments on individual parts of the program here.

Variables related to the number of pulses (variable *counts*) and time (variable *previousMillis*) measurements are also defined. The measurement interval is set to 10 s, and conversion factors for the calculation of cpm and dose equivalent power in μSv/h. The second-mentioned value was adjusted according to the measurement results gained using the GM Vernier Detector with type designation TH30044 (Fig. 4.8). Finally, the *impulse()* function is declared at the beginning of the program, within which impulses will be counted through the *counts* variable.

***Figure 4.8.** Measurement of uraninite activity using GM detector Vernier TH30044*

In the initialization part of the *void setup()* program, the connected hardware is initialized – serial communication for possible connection to the computer via USB and LCD (Fig. 4.9). The GM detector does not need to be initialized. Since it generates voltage pulses, it is enough to set the connected digital Arduino's pin no. 2 as an input and create an external interrupt to the *impulse()* function.

This function is triggered on each falling edge of the voltage signal on input pin no. 2. At the same time, it increases the variable *counts* by one (the function *counts*++ is a shorthand notation for increasing the value of the variable counts by 1). This expression is equivalent to writing *counts* = *counts* + 1 or *counts* + = 1, where the operation is called incrementing. In the *setup()* function, the *counts* variable is also initialized to 0.

Such a setting of the *impulse()* function allows us to track the total number of impulses detected during a specific period (in our case, 10 s).

In the next step, via serial communication and the display, we will write basic information about the detector and symbols that do not need updating during the measurement (cpm and µSv/hr).

```
21  void setup()
22  {
23      lcd.init();
24      lcd.backlight();
25          Serial.begin(9600);
26      pinMode(2, INPUT);
27      attachInterrupt(digitalPinToInterrupt(2), impulse, FALLING); // define external interrupts
28      counts = 0;
29      Serial.println("GM Detector of Beta and Gamma Radiation");
30      Serial.println(" cpm:  µSv/hr:");
31      lcd.setCursor(0,0);
32      lcd.print("GM Detector of Beta");
33      lcd.setCursor(0,1);
34      lcd.print("and Gamma Radiation");
35      lcd.setCursor(1,2);
36      lcd.print("uSv/hr: ");
37      lcd.setCursor(1,3);
38      lcd.print("cpm:    ");
39  }
```

***Figure 4.9**. Hardware initialization*

The main program loop (*loop()*) performs measurements and calculations (Fig. 4.10).

The *usv* and *cpm* variables, which are of type *String*, allow us to record the dose equivalent values in micro Sieverts per hour (µSv/hr) and pulses per minute (cpm) in text form. The batch equivalent power values are calculated from the cpm value and stored in these variables. They are then written out via the serial monitor and displayed on the LCD.

In the central part of the program, the *currentMillis* variable is initialized with the *millis()* function, which returns the number of milliseconds since the program started.

The "if" condition is used to compare the difference between *currentMillis* and *previousMillis* with the *LOG_PERIOD* value set to 10.000 ms. This condition is met if the specified time has passed. Inside the condition, it is first checked whether at least one pulse value from the GM detector has been detected (counts != 0).

If the condition is met, the program gets the current time using the *now()* function and stores it in the variable *t*. Next, the time is converted to seconds (minute(t) * 60) + second(t) and stored in the *time* variable.

Then, the dose equivalent input in units of µSv/hr (counts * usv_multiplier) is calculated and stored in the *usv* variable. This value is printed on the serial monitor and the LCD on the first line. In the same way, the number of pulses in cpm units is calculated (counts * cpm_multiplier) and stored in the *cpm* variable. This value is also output to the serial monitor and the LCD. At the end of the loop, the *counts* variable is reset to 0.

```
41  void loop()
42  {
43      String usv, cpm;
44      unsigned long currentMillis = millis();
45      if (currentMillis - previousMillis > LOG_PERIOD)
46      {
47          previousMillis = currentMillis;
48
49          if(counts != 0)
50          {
51            // Log the time
52            time_t t = now();
53            String time = String((minute(t) * 60) + (second(t)));
54
55            // Print in CPM
56            cpm = String(counts * cpm_multiplier,0);
57            Serial.print("  ");
58            Serial.print(cpm);
59            lcd.setCursor(9,3);
60            lcd.print("          ");
61            lcd.setCursor(9,3);
62            lcd.print(cpm);
63
64            // Print in µSv/hr
65            usv = String(counts * usv_multiplier, 1);
66            Serial.print("    ");
67            Serial.print(usv);
68            lcd.setCursor(10,2);
69            lcd.print("        ");
70            lcd.setCursor(10,2);
71            lcd.print(usv);
72
73
74          }
75
76          Serial.println();
77          counts = 0;
78      }
79  }
```

***Figure 4.10**. The main loop of the Arduino GM detector program*

4.2.3. Data collection via computer

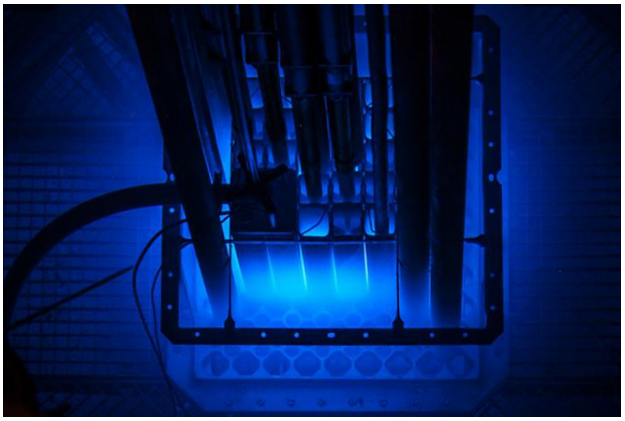
As we have already mentioned, for the actual collection of data using a computer, we can advantageously use one of the freely available programs that can record data sent by serial transmission from the Arduino via the USB connector to a text file. The latest version of the *CoolTerm* program worked best for us.

The installation of this program, as well as the collection of data into a text file and their transfer to the MS Excel spreadsheet are described in more detail in Chapter 5 focused on the construction and use of a simple meteorological station, so we will not deal with this issue anymore.

4.3. Tasks

4.3.1 How to use a GM in measurement and protect yourself from radiation (distance protection)

In the following section, we present a proposal for activities aimed at protection from ionizing radiation – distance and shielding.

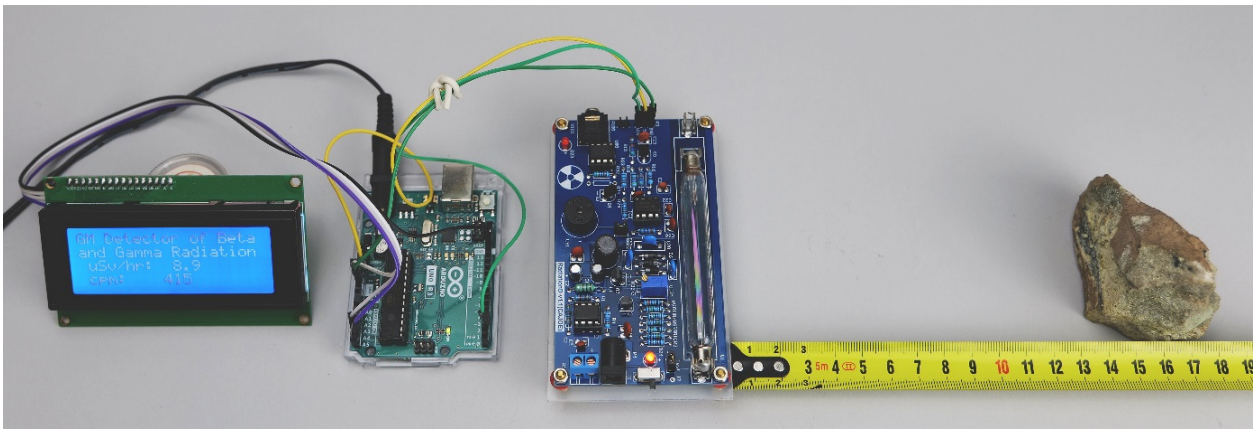


Introduction: As you know, the main types of ionizing radiation include alpha, beta, and gamma radiation. Uranium, which occurs in nature mainly in the similar form of uranium dioxide, is a source of all three types of radiation, and its isotope ^{235}U is also used in nuclear reactors and for military purposes, as it can split into lighter products. Ionizing radiation is also used in medicine, for example, in various imaging methods, such as X-ray, computed tomography, positron emission tomography, etc. However, we will show you how to easily protect ourselves from it. We will use a small piece of uranium ore – uraninite as a radiation source and an Arduino GM detector, which can capture beta and gamma radiation, as a detector.

Aim of the measurement: The experiment aims to measure the dependence of the number of pulses per minute (cpm) and the dose equivalent input on the distance between the GM tube and the emitter using a simple Arduino GM detector.

4.3.1.A. Measurement procedure

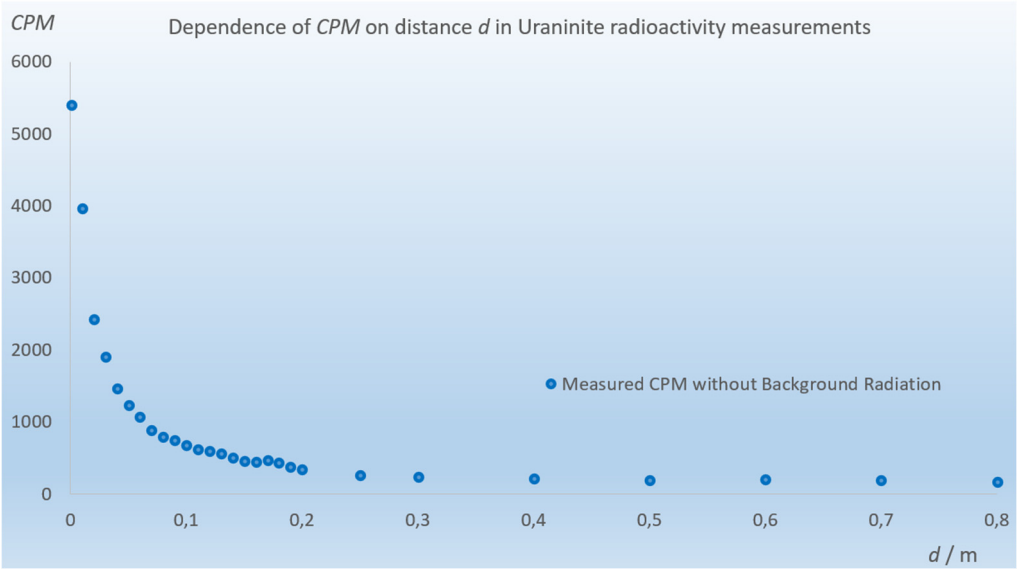
- The experimental setup consists of an Arduino GM detector, a measuring tape or ruler, and uranium ore, as shown in the image below.
- Before starting the actual measurement, first determine the background value – let the GM detector measure without the emitter and calculate the background radiation from the ten measured values of the number of pulses per minute.
- Place the uranium mineral as close to the GM detector tube as possible and record the pulse count (cpm) value 10 times. You can write them directly in Excel or record them in a text file through the *CoolTerm* program. Also, record the dose equivalent input values.



- Move the uranium mineral one centimetres at a time and record 10 pulse count values for each distance. Proceed in this way up to 20 cm.
- After overcoming 20 cm, you can move the emitter by five or ten centimetres up to 1 m.
- Do not forget to record 10 pulses per minute and dose equivalent input values per each distance.
- Calculate the average number of pulses per minute for each distance. Do not forget to subtract the background value.
- Construct a graph of the dependence of the average number of impulses on the distance.

4.3.1.B. Data analysis

The following figure shows the dependence of the number of pulses on the distance for the uranium ore used in our measurement. As you can see, the number of pulses per minute decreases with distance. However, this is a complicated dependence influenced by several factors (radiator geometry, GM tube geometry, different behaviour of beta and gamma radiation when passing through the air, etc.), so we do not express it with a trend line or an equation.



Try to answer the following questions based on (your own) measured data.

Date: _____

- The number of pulses per minute tends to increase / decrease with distance (cross out if not applicable).
- The maximum measured value was _____ cpm at a distance of _____cm.
- The minimum measured value was _____ cpm at a distance of _____cm.
- The usual effective dose of radiation is 13 mSv per year.
  The radiation dose close to the used emitter was _____ μSv/h.
- The radiation dose at 20 cm from the used emitter was _____ μSv/h.
- The radiation dose at 1 m from the used emitter was _____ μSv/h.

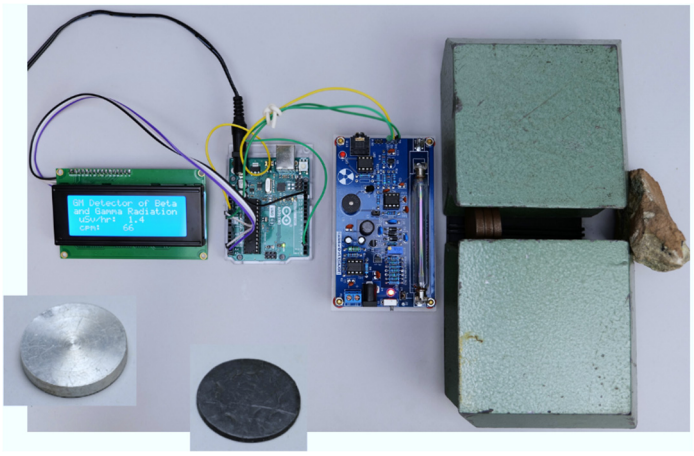
4.3.2. How to protect yourself from radiation (shielding protection)

Introduction: Sometimes, it is necessary to protect ourselves from radiation. One of the possibilities is protection by shielding. As we know, uranium emits all three types of radiation, with gamma radiation having the most extended range and penetration. The best materials to protect against gamma radiation are materials with a high proton number and high density (lead, concrete, steel, tungsten, etc.). Let us try how radiation shielding works. What will prove better protection – lead or aluminium – in the case of uranium mineral?

Aim of the measurement: This experiment uses a simple Arduino GM detector to measure the dependence of the number of pulses per minute (cpm) on the thickness of the metal material – aluminium and lead discs.

4.3.2.A. Measurement procedure

- Arrange the experimental setup like the one shown in the picture. Two lead blocks are placed between the Arduino GM detector and the uranium mineral. Thus, a narrow tunnel was created in the cross-section of which metal discs can be placed. It will ensure no other radiation from the uranium mineral will penetrate the detector. Aluminium and lead discs with diameters of 3.5 cm are also visible in the picture. The thickness of the aluminium disc used was 0.5 cm, and the lead disc was 1 mm.

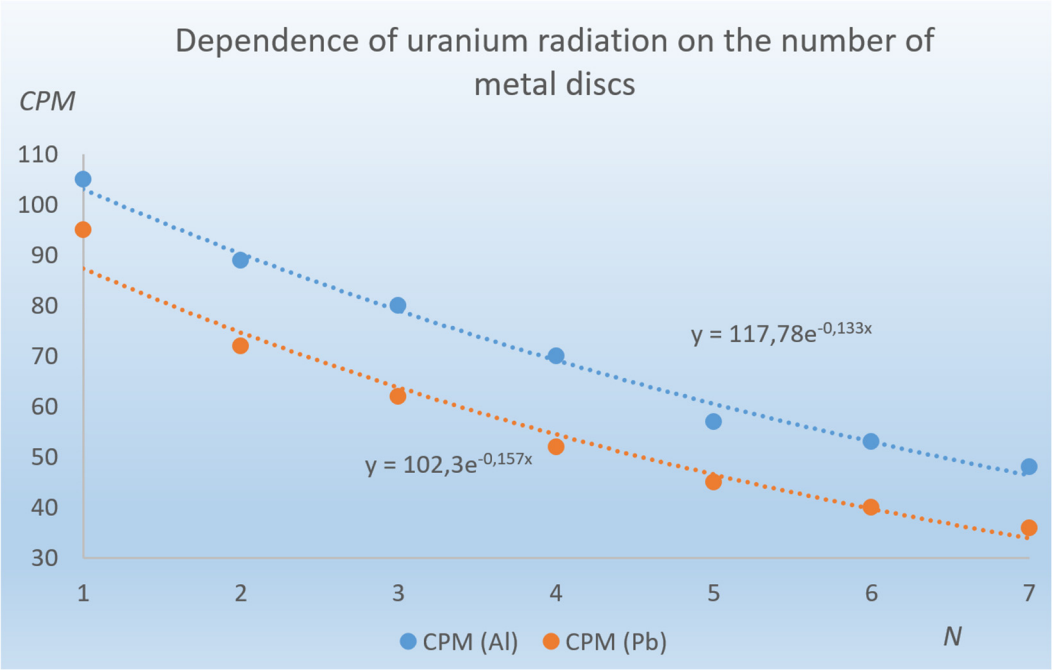


- Gradually insert metal discs into the tunnel between the lead bricks. For each number of discs N, determine ten pulses per minute (cpm) values. Then, calculate the average cpm value for each number of discs. Mark all values, e. g. to Excel.
- Construct graphs of the dependence of the number of cpm pulses on the number of discs.

4.3.2.B. Data analysis

The figure below shows the graph of the dependence of the number of pulses on the number of metal discs. If we plot the trend line through both dependencies, we find that the best fit is exponential, which is also consistent with the theory. (If we wished, we could also calculate other physical parameters for both materials from the measured data, for example,

their linear attenuation coefficient, which, however, exceeds the complexity of the purpose of these materials).



Based on your measured data, try to answer the following questions.

Date: _____

- In both cases, the number of pulses per minute tends to increase / decrease with increasing material thickness (number of discs) (strike out if not applicable).
- In the case of aluminium, the maximum value of _____ cpm was measured with the number of discs *N* = _____.
- In the case of lead, the maximum value of _____ cpm was measured with the number of discs *N* = _____.
- In the case of aluminium, the minimum value of _____ cpm was measured with the number of discs *N* = _____.
- In the case of lead, the minimum value of _____ cpm was measured with the number of discs *N* = _____.
- The total thickness of the aluminium discs used was _____ cm.
- The total thickness of the used lead discs was _____ cm.
- If we consider the thickness of the aluminium and lead discs, aluminium/lead is a much better material to protect against ionizing radiation coming from uranium minerals (cross out if applicable)

References

- SHMÚ – Rádioaktivita. 2023. [online] [cit. 2023-07-11] <https://www.shmu.sk/sk/? page=20>.
- Nikodemová, D., Cabáneková, H. (2009). Radiačná ochrana. SZU Bratislava. [online] [cit. 2023-07-11] <https://www.nuclear.sk/wp-content/uploads/2021/06/ Nikodemova-radiacna-ochrana.pdf>.

- Zákon č. 87/2018 Z. z. (2018). [online] [cit. 2023-07-12] <https://www.slov-lex.sk/static/pdf/2018/87/ZZ_2018_87_20230415.pdf>.
- Zákon č. 119/2023 Z. z. (2023). [online] [cit. 2023-07-12] <https://www.slov-lex.sk/pravne-predpisy/SK/ZZ/2023/119/20230415>.
- Úrad jadrového dozoru SR. 2022. Základné údaje o rádioaktivite. [online] [cit. 2023-07-11] <https://www.ujd.gov.sk/jadrovy-program/zakladne-udaje-o-radioaktivite/>.
- Abbas, A. (2022). Geiger Muller Counter-Construction and Working of Geiger Muller Counter. [online] [cit. 2023-07-11] <https://eduinput.com/geiger-muller-counter/>.
- Holá, O. (2009). Ionizujúce žiarenie a jeho vlastnosti. [online] [cit. 2023-07-11] <https://www.nuclear.sk/wp-content/uploads/2021/06/Hola-Ionizujuce-ziarenie.pdf>.
- WHO. (2022). Ionizing radiation, health effects and protective measures. [online] [cit. 2023-07-11] <https://www.who.int/news-room/fact-sheets/detail/ionizing-radiation-health-effects-and-protective-measures>.
- Ryan, J., L. (2012). Ionizing Radiation: The Good, the Bad, and the Ugly. In: J Invest Dermatol. 2012 Mar; 132(3 0 2): 985–993. [online] [cit. 2023-07-11] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3779131/>.
- Canadian Centre for Occupational Health and Safety. (2023). [online] [cit. 2023-07-11] <https://www.ccohs.ca/oshanswers/phys_agents/ionizing.pdf>.
- United States Nuclear Regulatory Commision. (2020). [online] [cit. 2023-07-11] <https://www.nrc.gov/about-nrc/radiation/health-effects/measuring-radiation.html>.
- IoT-devices. (2023). Geiger-Muller tubes: Comparison of SBM20, J305 and LND712. [online] [cit. 2023-07-11] <https://iot-devices.com.ua/en/comparison-of-geiger-muller-tubes-sbm20-j305-and-lnd712/>.
- Banggood.com. (2023). Geekcreit Assembled Geiger Counter Module. [online] [cit. 2023-07-11] <https://www.banggood.com/Geekcreit-Assembled-Geiger-Counter-Module-Miller-Tube-GM-Tube-Nuclear-Radiation-Geekcreit-for-Arduino-products-that-work-with-official-Arduino-boards-p-1136883.html?imageAb=2&rmmds=search&cur_warehouse=CN&akmClientCountry=SK>.
- Laquai, B. (2014). Geigerzähler-Shield für den Arduino von Libelium. [online] [cit. 2023-07-11] <http://www.opengeiger.de/LibeliumDoku.pdf>.
- Státní úřad pro jadernou bezpečnost. (2004). Problematika uranem barveného skla. [online] [cit. 2023-07-11] <https://www.sujb.cz/radiacni-ochrana/oznameni-a-informace/problematika-uranem-barveneho-skla>.
- United States Environmental Protection Agency. (2023). Radioactive Decay. [online] [cit. 2023-07-12] <https://www.epa.gov/radiation/radioactive-decay>.
- Matolín, M. (1970). Radioaktivita hornin českého masívu. Praha : Academia, 1970, p. 99.
- Tenzler, D. (2016). Radioaktivita, která se stala uměním – uranové sklo. [online] [cit. 2023-07-11] <https://danatenzler.blog.idnes.cz/blog.aspx?c=545595>.

CHAPTER 5. A METEOROLOGICAL STATION BASED ON AN ARDUINO

This chapter was written by Martin Hruska
from the Matej Bell University of Banska Bystrica, Slovakia

5.1. Theory

Suppose we must record primary meteorological data (air pressure, temperature, humidity, etc.). We can build a simple meteorological station (Fig. 5.1) with a liquid-crystal display (LCD) using an Arduino Uno microcontroller.

Figure 5.1. *The simple meteorological station with the Arduino Uno microcontroller*

With the help of the display, we can easily display the meteorological data and record them concurrently in a text file using the serial connection of the meteorological station to the computer with a USB cable.

We designed a meteorological station that could be easily constructible using a few affordable modules (Arduino, display, and three sensors). However, it has shown high reliability in long-term measurements and has been easy to operate and install.

Based on our experience, a simple LCD and three sensors combination has proved the most effective. The BH170 sensor measures illumination (lighting intensity) in lux. The BMP180 sensor enables measuring barometric pressure, and the HTU21D sensor records the relative air humidity and temperature.

5.1.1. A brief description of each component of the meteorological station

- **Arduino Uno**

For constructing a simple meteorological station, we chose the Arduino UNO board (Fig. 5.2). This microcontroller is suitable for the first experience of creating your projects. Most projects have been created for it, described most frequently on publicly available forums. The Arduino UNO microcontroller works with the ATmega328P microprocessor. It is an 8-bit 16 MHz single-core processor with 32 KB of program memory and a 1024-byte EEPROM. The board has

14 digital input/output pins, six analogue pins, a USB connector, a power connector, and a restart button. The microcontroller can be powered via USB, a DC power adapter, or an external battery with 6 V to 15 V voltage (Voda et al., 2018).

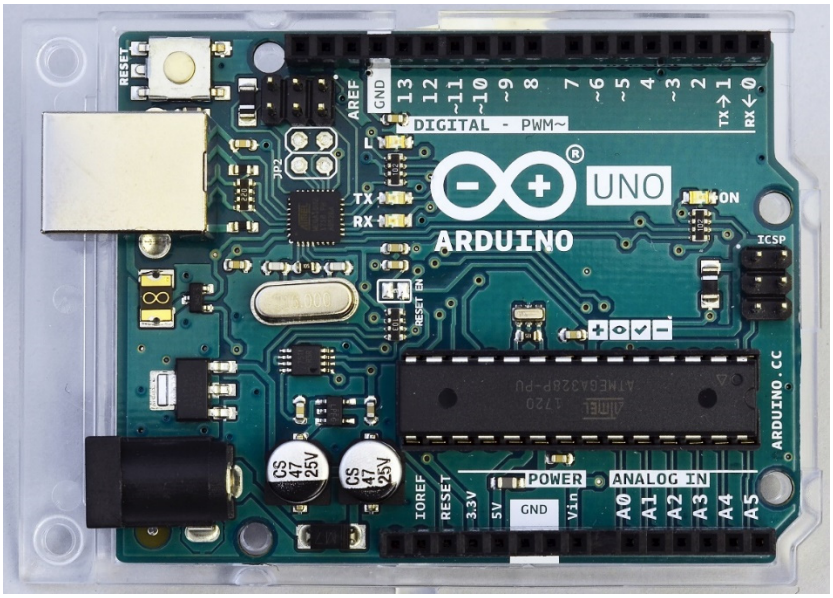
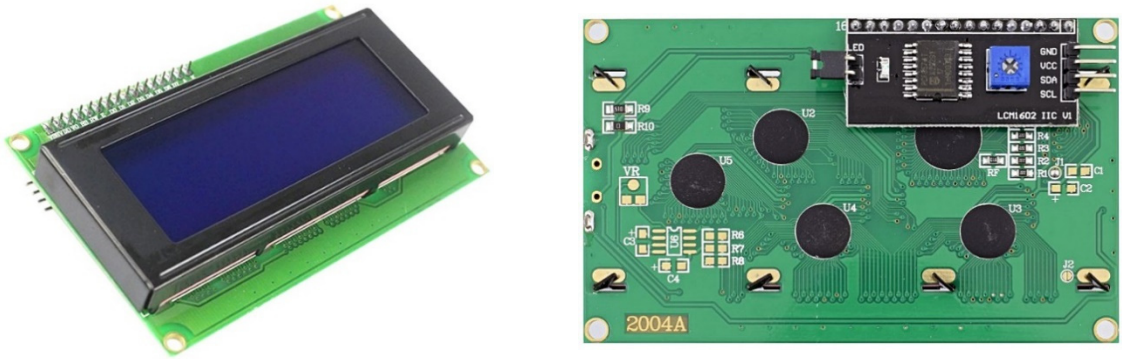


Figure 5.2. *Arduino Uno microcontroller*

- **Liquid-crystal display 20 x 4**

It is a standard liquid-crystal display with a blue backlight, with the number of characters in a line of 20 and with four lines (Fig. 5.3a, b). The display also includes an I²C bus that enables its connection to the Arduino using four wires: SLC, SDA, GND, and 3.3 V or 5 V DC. The potentiometer on the I²C interface module on the back of the display can be used to adjust its contrast depending on the power supply. At the same time, it is possible to switch the display backlight. In our case, we connected the contacts so that the display shines continuously.



a) b)

Figure 5.3a, b. *LCD 20 x 4 with a blue backlight and I²C bus*

For the display to function correctly, the *LiquidCrystal_I2C.h* library, which is freely available on the Internet, must be installed in the Arduino IDE environment on the computer. The easiest way to install the library is to download it to your computer in the ZIP archive form. Subsequently, in the Arduino IDE environment on the *Sketch* tab, select the option *Include Library*/*Add ZIP.Library*…, select the location of the downloaded ZIP file with the library and confirm.

Arduino IDE will inform us about the successful installation of the library with a text notification.

- **Light intensity sensor BH1750**

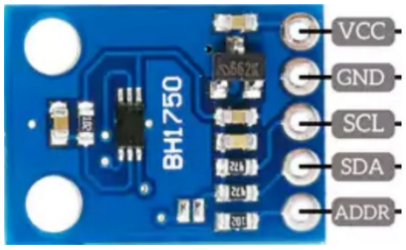


***Figure 5.4**. BH1750 sensor, intended for measuring the intensity of illumination in lux*

The light intensity sensor (Fig. 5.4) communicates using the I²C interface while converting the measured light intensity to a digital output in the numerical value form.

The measurement result is expressed in lux in the 0 lx to 65535 lx interval. So, the sensor contains a 16-bit AD converter, which provides up to 2^{16} illumination intensity values. The temperature operating range of the sensor is from -40 °C to +85 °C; the current consumption reaches a negligible 0.12 mA. The sensor is adapted to the spectral properties of the human eye. As seen in Fig. 5.7 below, we used a voltage of 3.3 V provided directly by the Arduino microcontroller to power this and the other two measurement modules. The sensor is not suitable for applications where we need to measure rapid temperature changes because the minimum time between measured values reaches, according to the manufacturer, 120 ms to 180 ms.

The sensor can be used in three modes, depending on how fast and accurately we want to measure the light intensity. With two slower measurement modes, we can achieve a resolution of 1 lx (H-Resolution Mode) or up to 0.5 lx (H-Resolution Mode2), while the measurement takes 120 ms to 180 ms. If we set the fast measurement mode with a low resolution of 4 lx (L-Resolution Mode), the measurement will take 16 ms to 24 ms. The high-resolution mode is more appropriate when measuring, for example, lower lighting values. We receive lower noise values as well. The dimensions of the sensor are 18.6 mm x 14.5 mm. The sensor has two mounting holes and uses the BH1750.h library, accessible freely on the Internet (Siepert, 2022, Datasheet BH1750, 2011).

- **BMP180 barometric pressure sensor**

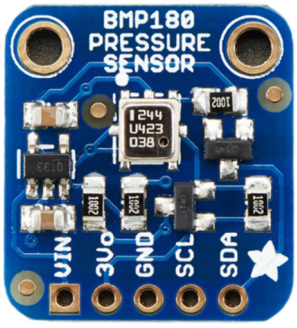


***Figure 5.5.** BMP180 sensor designed for measuring barometric pressure*

The sensor for measuring barometric pressure BMP180 (Fig. 5.5) from Bosch can also measure temperature (however, we will not use this function in our circuit due to insufficient measurement accuracy). Typical pressure measurement accuracy is ± 1hPa, while the sensor measures pressure in 300 hPa to 1100 hPa range. This sensor also communicates using the I²C interface, so four wires are enough to connect to the Arduino. The dimensions of the sensor

are 21 mm x 18 mm. The sensor has two mounting holes for attachment (Datasheet BMP180, 2013).

The BMP180 sensor uses the *Adafruit_BMP180.h* library, which is freely available on the Internet. This library is compatible with both BMP085 and BMP180 sensors.

- **Relative humidity and temperature sensor HTU21D**

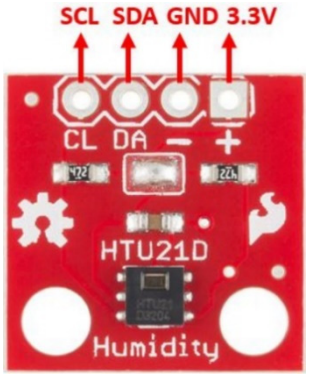


Figure 5.6. *HTU21D sensor designed for measuring relative humidity and air temperature*

The HTU21D sensor (Fig. 5.6) enables to measure the temperature in the -40 °C to +125 °C range and the relative air humidity from 0 % to 100 %. Regarding the accuracy of the measurement, in the case of temperature, the accuracy is usually ± 0.3 °C, while the most accurate measurement is achieved in the interval of 5 °C to 60 °C. In the case of humidity, the accuracy is ± 1 % and the most accurate sensor measures in the range of 10 % to 90 %. The supply voltage for this module recommended by the manufacturer is in the 3.3 V to 5 V range. The electric current drawn is low, reaching a maximum of 140 nA at rest and a maximum of 0.5 mA during measurement. To all the mentioned advantages, we can also add the size of the module, which is 12 x 10 mm, and the mounting hole for attaching the sensor (Datasheet HTU21D, 2013).

The HTU21D humidity and temperature sensor uses the *SparkFunHTU21D.h* library, which is freely available on the Internet.

5.1.2. Description of the connection and construction of the meteorological station

The LCD and all three mentioned sensors use the I²C serial interface for communication with the Arduino Uno microcontroller. This interface enables the connection of a device (a sensor or a display) to the Arduino using only two wires. The wire marked SCL (Serial Clock) serves for clocking the communication. The other wire (SDA – Serial Data) serves for data transfer. In addition, the GND and 3.3 V or 5 V power wires must also be connected to the devices. One device that controls the communication must be of the Master type (it is the Arduino in our case). Other devices are of the Slave type. Each device is identified by its unique address when communicating using I²C (Voda, Z. et al., 2018). The connection diagram of the Arduino Uno microcontroller with the LCD and sensors in the meteorological station is shown in Fig. 5.7.

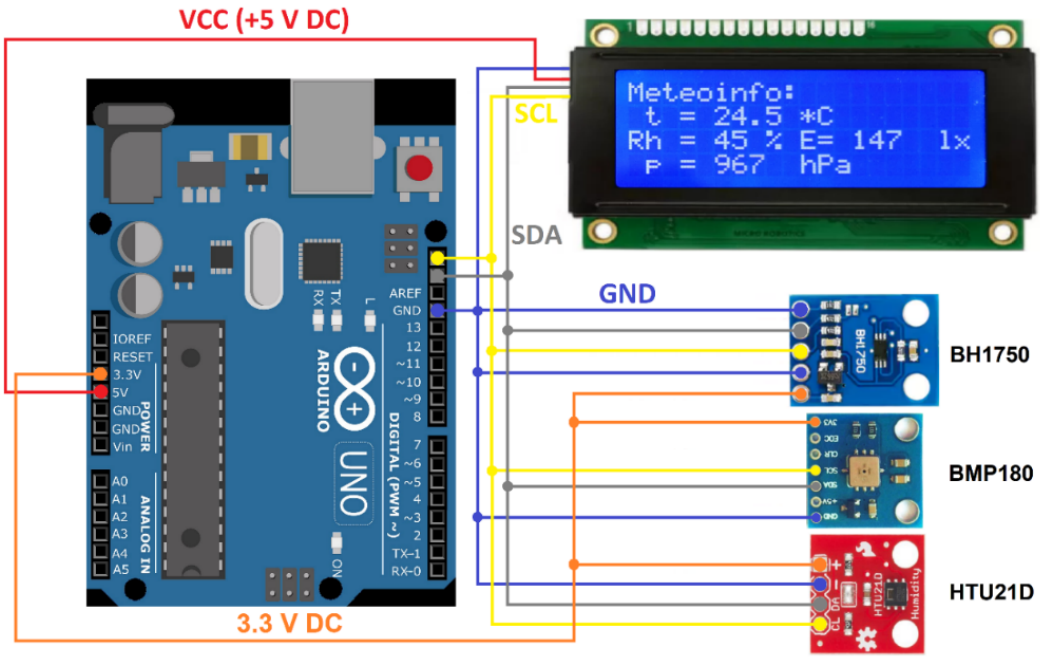


***Figure 5.7**. Connection of the simple meteorological station*

We will make connections using wires with DuPont type M (Male) terminals that are the best for connection to the Arduino, as seen in Fig. 5.8.

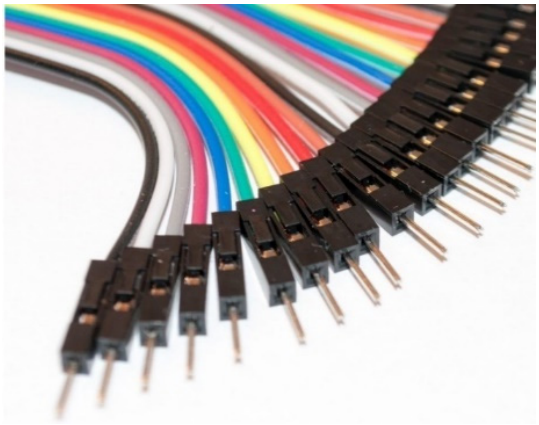


***Figure 5.8**. Wires with terminals to connect to Arduino*

Due to the use of the I²C bus and the same 3.3 V DC power supply, the sensors can be installed on a suitable printed circuit board, connected according to the diagram in Fig. 5.7, and placed in an appropriate small assembly box with a transparent cover (Fig. 5.9a, b), which can be placed in an outdoor environment, for example behind a window. It will allow us to connect the sensor box and the larger meteorological station box to the display and the Arduino using a four-wire cable with SCL, SDA, GND, and 3.3V DC wires.

The meteorological station itself (Fig. 5.10a) was also installed in a mounting box with a transparent cover. The rectangular hole in the housing that houses the display was cut using industrial water jet cutting equipment, but it is also possible to use a powerful laser.

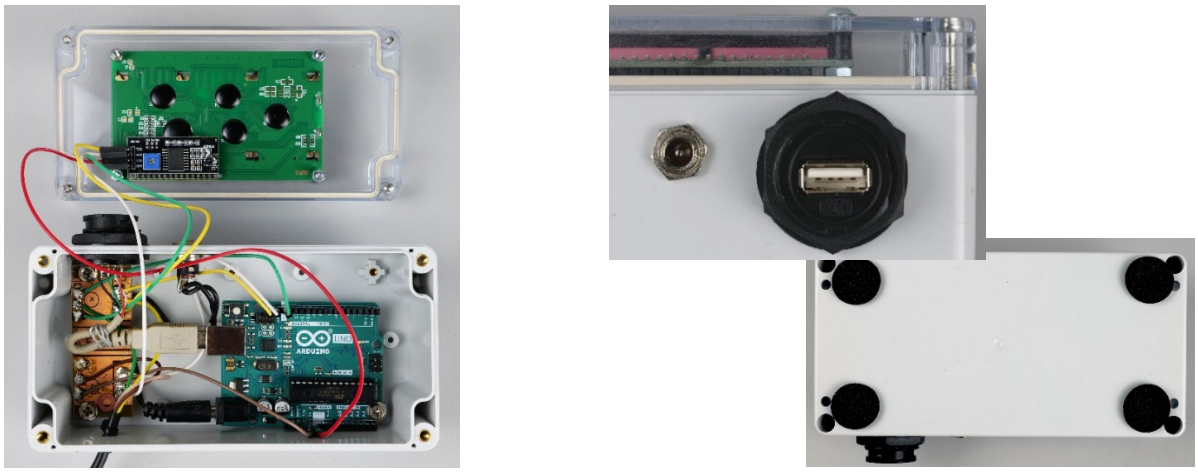
If we do not have any of these tools, it is possible to drill a hole in the corner of the future opening, cut out the opening gradually (e.g., with a chip saw), and clean it with a file. But it is more laborious, and there is more room for error.

a) b)

Figure 5.9a, b. *Mounting the sensors on the printed circuit board and fixing the printed circuit board with the sensors in the assembly box with a transparent top cover (a), view of the drilled holes from the side of the box (b)*

When placing the box with sensors, we must comply with several conditions:

- Make sure it is placed in the shade and not in direct sunlight. It will ensure that the HTU21D sensor will measure the actual temperature of the outside air and the sensor box will not overheat.
- It is necessary to drill suitable holes in the side walls of the box with sensors (in our case, we have made a total of 8 holes with a diameter of 8 mm in the walls of the box, i.e., two holes on each side). In this way, we achieve that the air pressure in the box will be the same as in its surroundings, and the air in the box changes well. Thanks to this, the HTU21D sensor can also measure the relative humidity of the surrounding air. Of course, holes cannot be drilled in the upper transparent cover of the sensor box. Otherwise, it could rain or snow into the box.
- The Arduino itself and the display should be placed in a larger mounting box and left in the room connected to a power source or (in the case of continuous data collection) to the USB port of a computer or laptop.
- It is convenient to place the power connector and USB connector on the back side of the assembly box (Fig. 5.10b). This allows us to use a suitable DC adapter to power the Arduino, as well as to program the meteorological station via USB from the computer without the need to open the assembly box.
- Both mounting boxes (with sensors, as well as with the display and Arduino) should be equipped with adhesive anti-slip mats (Fig. 5.10b).



a) b)

Figure 5.10a, b. *View inside the meteorological station (a), power and USB connector of the station, view of the anti-slip pads on the bottom (b)*

5.2. Laboratory setup

5.2.1. Programming the meteorological station and description of the program

In the first part of the program (Fig. 5.11), it is necessary to load the libraries that contain a code for controlling and communicating with the connected hardware. We also need to specify which devices are connected to the Arduino for the correct operation of the initialized libraries.

Note: The double slash symbol "//" in the program means that the program will ignore the subsequent text. Therefore, we can write explanatory comments on individual parts of the program here.

Meteoinfo_LCD_Serial_fin.ino

```
1   // Simple weather station, Martin Hruska 2023
2   // Launch the necessary libraries
3   #include <Wire.h> // Library for I2C communication
4   #include "SparkFunHTU21D.h"
5   #include <Adafruit_BMP085.h>
6   #include <BH1750.h>
7   #include <LiquidCrystal_I2C.h> // Library for LCD display
8
9   // Create an instance of the object
10  HTU21D myHumidity; // Relative Humidity (%), Temperature (*C)
11  Adafruit_BMP085 bmp; // Air Pressure (hPa)
12  BH1750 lightMeter; // Light Intensity (lx)
13  LiquidCrystal_I2C lcd(0x27, 20, 4); // Define display properties
14
```

***Figure 5.11**. The introductory part of the meteorological station program*

We can initialize the connected hardware in the *void setup()* section (Fig. 5.12). First, we initialize the serial and I²C communication, the display, and the connected sensors. We write a greeting and an instruction to save data to Excel via serial communication. Finally, we inscribe the expression "Meteoinfo:" on the display:

```
15  void setup()
16  {
17    Serial.begin(9600); // Initialization of serial communication
18    Wire.begin(); // Initialization of I2C communication
19    lcd.init(); // Initialization of LCD display
20    lcd.backlight(); // Switching on the LCD backlight
21    myHumidity.begin(); //Initialization of sensors
22    !bmp.begin();
23    if (lightMeter.begin(BH1750::CONTINUOUS_LOW_RES_MODE)) {
24      Serial.println(F("Simple weather station, KF FPV UMB BB 2023. Copy this data to Excel:")); // Prints the initial text
25    } else {
26      Serial.println(F("Error initialising BH1750"));
27    }
28    lcd.setCursor (0,0); // Setting the display cursor to the initial position
29    lcd.print("Meteoinfo:"); // The text that will not be changed during the measurement will be displayed
```

***Figure 5.12**. Hardware initialization*

Subsequently, we write symbols and markings on the display and the computer during serial communication, which will not change (Fig. 5.13).

```
31    lcd.setCursor (1,1);
32    lcd.print("t = ");
33    lcd.setCursor (9,1);
34    lcd.print(" *C");
35
36    lcd.setCursor (0,2);
37    lcd.print("Rh = ");
38    lcd.setCursor (7,2);
39    lcd.print(" %");
40
41    lcd.setCursor (10,2);
42    lcd.print("E= ");
43    lcd.setCursor (17,2);
44    lcd.print(" lx");
45
46    lcd.setCursor (1,3);
47    lcd.print("p = ");
48    lcd.setCursor (9,3);
49    lcd.print(" hPa");
50
51    Serial.print("t/*C  ");
52    Serial.print("   Rh/%  ");
53    Serial.print("     p/hPa  ");
54    Serial.print("      E/lx ");
55    Serial.println();
56    }
```

Figure 5.13. *Printing of fixed text on the display and in the computer during serial transmission*

The next section of the program – *void loop()* – contains a part of the code that is constantly repeated (Fig. 5.14). In it, we will introduce the necessary variables and request values from the sensors, which we will then display on the LCD. We still must convert the pressure value to hectopascals:

```
58  void loop()
59  {
60    // Creating the necessary variables
61    float temp = myHumidity.readTemperature();
62    float humd = myHumidity.readHumidity();
63    float lux = lightMeter.readLightLevel();
64
65    // Temperature value display
66    lcd.setCursor (4,1);
67    lcd.print("      ");
68    lcd.setCursor (5,1);
69    lcd.print(temp, 1);
70
71    // Relative humidity value display
72    lcd.setCursor (4,2);
73    lcd.print("    ");
74    lcd.setCursor (5,2);
75    lcd.print(humd, 0);
76
77    // Illuminance value display
78    lcd.setCursor (12,2);
79    lcd.print("      ");
80    lcd.setCursor (13,2);
81    lcd.print(lux, 0);
82
83    // Air pressure value display
84    lcd.setCursor (4,3);
85    lcd.print("      ");
86    lcd.setCursor (5,3);
87    lcd.print(bmp.readPressure()/100); // Convert pressure value to hPa
88
```

Figure 5.14*. Introduction of variables and writing of measured values on the display*

In the last part of the program (Fig. 5.15), we write the measured values to the computer and stop the program for a while using the *delay()* command so the values are recorded 1 x per minute during the long-term measurement.

It means that the values of pressure (hPa), temperature (°C), relative humidity (%), and light intensity (lx) are refreshed at the display and recorded to the computer approximately every minute:

```
89  // Output of values via serial interface
90  Serial.print(temp, 1);
91  Serial.print("     ");
92  Serial.print(humd, 1);
93  Serial.print("       ");
94  Serial.print(bmp.readPressure()/100); // Convert pressure value to hPa
95  Serial.print("          ");
96  Serial.print(lux);
97  Serial.println();
98
99  delay(59800); // Delay - values are recorded every minute
```

***Figure 5.15**. Writing the measured values to the computer and interrupting the program for the required time*

If we need to record values, e.g., every second, you only need to correctly set the value of the *delay()* command in the last line of the program. For example, *delay(1000)* means the program will pause for 1000 ms. But we must not forget the program itself takes some time, so it is necessary to determine the delay value experimentally.

5.2.2. Data collection via computer

For the actual data collection using a computer, we can advantageously use one of the freely available programs that can record data sent by serial transmission from the Arduino via the USB connector to a text file. The latest version of the *CoolTerm* program, available on the Internet (CoolTerm, 2022), proved to be the best for us.

You do not need to install the program; just run it after downloading. After starting the program, select the option to set communication parameters automatically. Subsequently, initialize serial transmission using the button 

on the top bar of the program (Fig. 5.16).

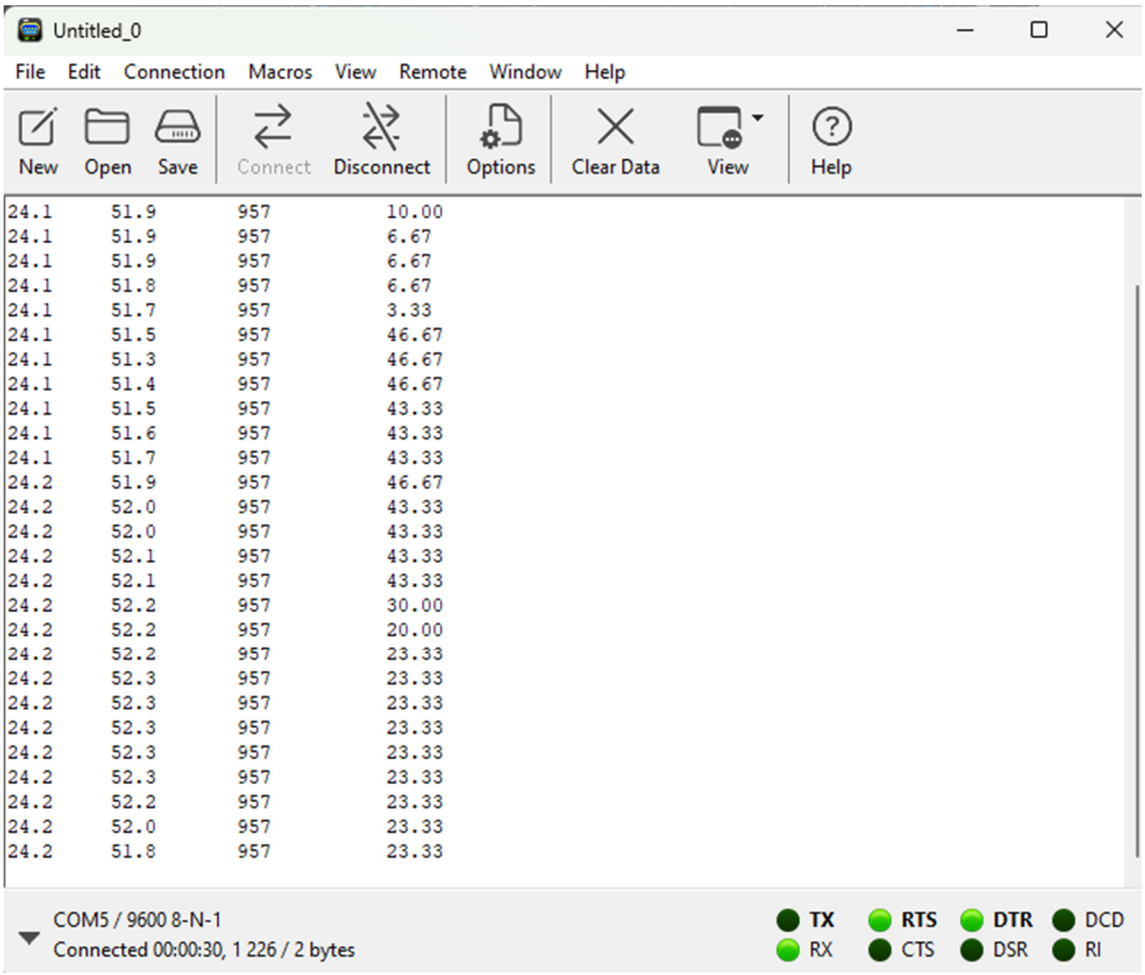


***Figure 5.16**. A sample of the CoolTerm program environment and a view of the recorded values*

If we want to save the data in a text file, we must select the command *Connection/Send Text/Binary File...* on the top bar of the program and confirm the saving in a text file with a specific name in the given folder (Fig. 5.17).

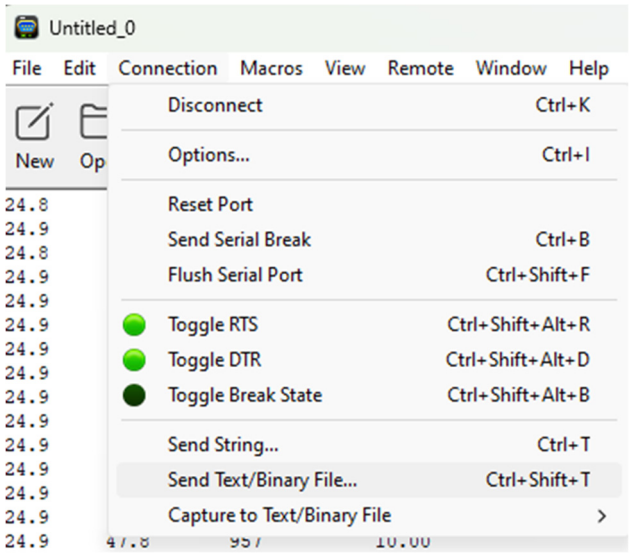


***Figure 5.17**. Settings for saving data to a text file*

When transferring data to Excel, it is sometimes necessary to consider the symbol that separates whole and decimal numbers. It may be a decimal point in some countries, a dot in others. To avoid unnecessary problems when transferring data to Excel, it is easiest to replace the separator symbol for whole and decimal numbers directly in the text file using the command *Ctrl+H* (Fig. 5.18).

After replacing all the separator characters, we can use the command *Ctrl+C* to mark all the data in the text file and use the command *Ctrl+V* to insert them into Excel, edit the cells with data into the format *Number* with the given number of decimal places (usually one place is enough) and construct the desired graphs.

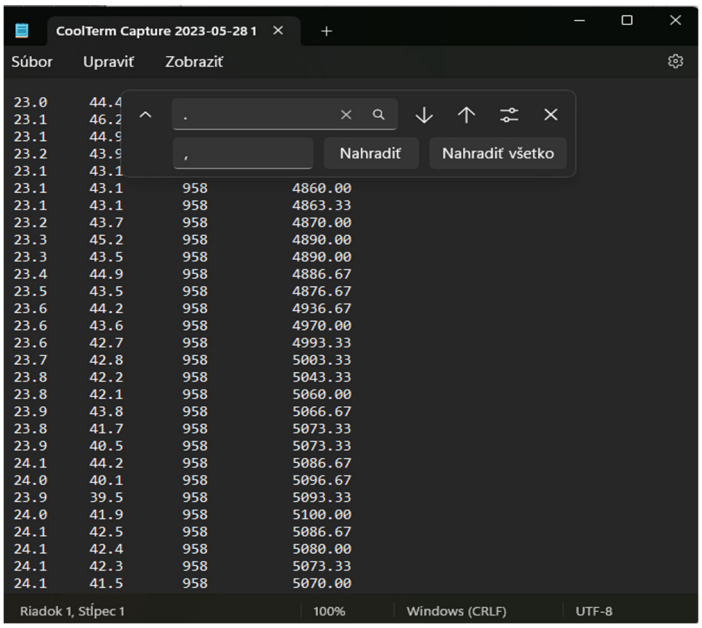


***Figure 5.18**. Example of replacing the decimal separator symbol in the Slovak localization of the text editor with measured data*

5.3. Tasks

5.3.1. Use in measurement. Meteorological observation

In the following section, we present a proposal for an activity called Meteorological Observation.

The activity is processed in the form of a student worksheet.

Introduction: Correct predictions of the weather, or the state of the atmosphere above a given place on the earth's surface in the future, are essential not only for the general public but especially for such areas of human activity as air and maritime transport, agriculture, various industries, etc. If we want to predict the weather, we first need to measure the meteorological elements, i.e., the quantitative characteristics of the atmosphere. They include, for example, air temperature and humidity, duration of sunshine, wind direction and speed, amount of precipitation, etc.

How does the temperature change during the day and night? How do we know when it is day and when it is night? Come and try how some variables characterizing the weather can be recorded.

Aim of the measurement: The experiment aims to use a simple meteorological station to measure selected weather characteristics over several days. Process the obtained data, for example, in the MS Excel environment, and determine how the weather changed during the measured period.

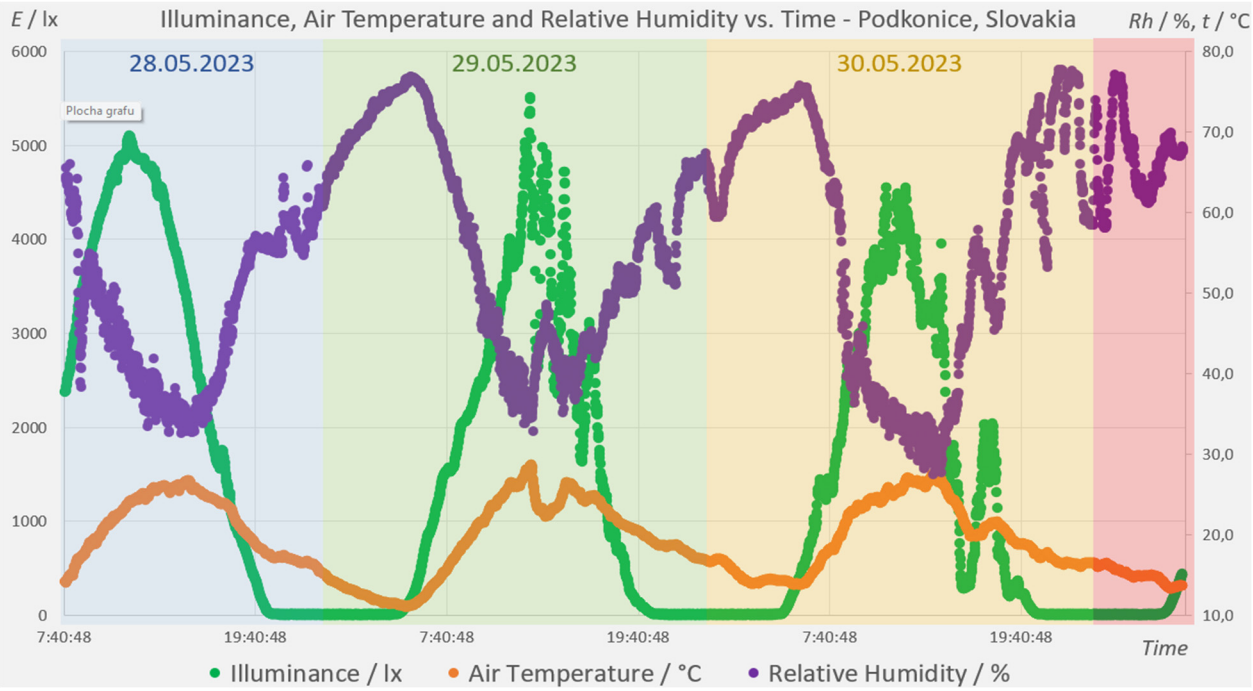
5.3.1.A. Measurement procedure

- The experimental setup consists of a meteorological station with sensors and a computer to record measured data long-term.
- Place the box with the meteorological station sensors, e.g., on the windowsill so it is not in direct sunlight.
- Connect the meteorological station to the computer using a USB cable and start the CoolTerm data recording software.
- Adjust the program to start saving data to a text file. Do not forget to record the beginning of the measurement – the meteorological station measures data every minute, so you can readily determine, based on the measurement beginning, when the four values (air temperature, relative air humidity, atmospheric pressure, and light intensity) were recorded.
- Let the measurement take place for several days, or at least 24 hours (do not forget to set the computer so that it does not turn off or switch to power saving mode).

5.3.1.B. Data analysis

After 24 hours or a few days, stop the measurement and copy the recorded data from the text file to Excel. Construct a graph from the measured values or several graphs, depending on whether you want to display the time dependence of the measured variables in the form of separate graphs or not.

An example of what such a graph might look like is below. The graph shows the values of illuminance, relative air humidity, and air temperature measured over several days. Individual days are highlighted in colour to make the chart easier to read. We can approximately determine the values directly from the graph, but we can obtain more complete information from the table of measured data.



Try to answer the following questions based on (your own) measured data.

Date: _____

- The maximum measured air temperature was ___°C at the time: _____.
- The minimal measured air temperature was ___°C at the time _____.
- The daily temperature amplitude during the observation had a value of _____°C.
- Maximum relative air humidity reached ___% at the time: _____.
- Minimal relative air humidity was ___% at the time _____.
- The night (E = 0 lx) began at _____ .
- The dawn (E > 0 lx) has come _____.
- The night lasted _____.
- The air pressure had a rising/falling tendency; it was stable (strike out if not applicable).

References

- CoolTerm 2.0.1.1150 (2022). [online] [cit. 2023-06-26] <https://coolterm.en.lo4d.com/windows>.
- Datasheet BH1750 (2011). [online] [cit. 2023-06-26] <https://www.mouser.com/datasheet/2/348/bh1750fvi-e-186247.pdf>.
- Datasheet BMP180 (2013). [online] [cit. 2023-06-26] <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>.
- Datasheet HTU21D (2013). [online] [cit. 2023-06-26] <https://cdn-shop.adafruit.com/datasheets/1899_HTU21D.pdf>.
- Siepert, B. (2022). Adafruit BH1750 Ambient Light Sensor. Available at: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-bh1750-ambient-light-sensor.pdf>.
- Voda, Z. et al. (2018). Průvodce světem Arduina. [online] [cit. 2023-06-26] <https://arduino.adamit.eu/books/Zbysek_Voda__2_vydanie_2018__Pr%C5%AFvodce-sv%C4%9Btem-Arduina-CZ.pdf>

CHAPTER 6. A KIT OF PM SENSOR DESCRIPTION AND HOW IS MADE A PM SENSOR

This chapter was written by Radu Motisan from Magnasci SRL (6.1, 6.1.1.), Romania

6.1 Theory

PM Smoggie sensor is a low-cost automatic air quality monitor with rainproof housing and a simple mounting system to make installation easy. It has a high-quality laser scattering sensor for PM1, PM2.5, and PM10 particles and an additional temperature sensor and humidity. It connects to the internet via Wi-Fi and can be powered with a standard 5V micro-USB cable.

The measurements are automatically transmitted to the uRADMonitor data server, from where they can be accessed through the existing API or accessible and decentralized directly through the local network. This monitor is laboratory-tested for data accuracy.

This sensor is a very low-cost device, where all its construction components have been optimized from the point of view of costs, except for one: the quality of the resulting product. The design is open source, with complete hardware and software details publicly available on Github. Its software can be modified using Arduino. By default, all measurements are sent to the uRADMonitor servers and are accessible via the API or viewed online (www.uradmonitor.com).

6.1.1. Kit content and the role of each component

PM Smoggie sensor holds a high-precision laser scattering sensor and a MEMS[1] temperature and humidity sensor. A built-in fan ensures an active airflow on the detection elements. The device connects to your wireless internet router via Wi-Fi to send the readings online (Fig. 6.1). [1]

The Smoggie sensor measures three meteorological parameters: air temperature (0.5 °C resolution and ±1 °C accuracy), barometric pressure (±0.25% accuracy), and relative humidity (1% resolution and ±2% accuracy) using Microelectromechanical systems (Sensirion SHT21). It also uses an integrated laser scattering detector (Plantower PMS5003) to measure PM1, PM2.5, and PM10 concentrations in the air.

A pulse of coherent infrared light shines through a cavity with a PIN photodiode located sideways to detect PM concentrations. The fan forces the air into the chamber. As a particle reaches the laser beam, it scatters the laser light, and the photodiode detects the scattered light. The amplitude of the recorded scattered signal is proportional to the particle size. This thing helps to correlate the number of events to the mass concentration.[2]

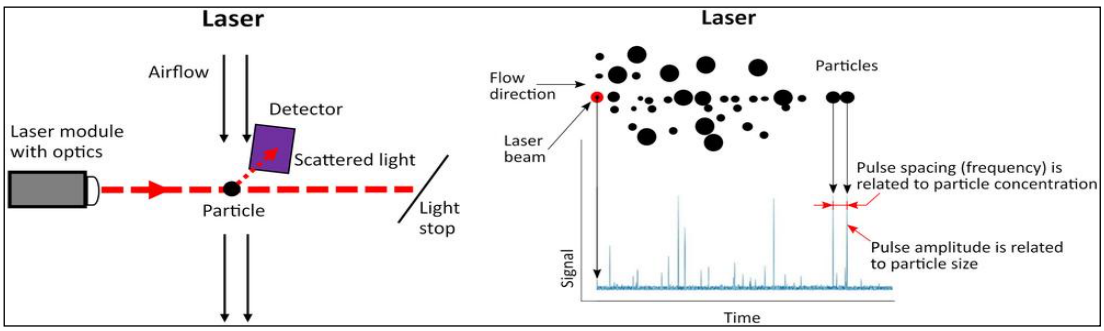


***Figure 6.1**. Functional scheme of the optical particulate matter sensor [2]*

[1] Micro-Electro-Mechanical Systems (MEMS) - the integration of mechanical elements, sensors, actuators, and electronics on a common silicon substrate through microfabrication technology

Table 1. *The sensors give parameters*

sensors	Parameter	Minimum	Maximum value	Error
Sensirion SHT21	Temperature	-40 ° C	+125 ° C	± 0.3 ° C
	Moisture	0% RH	100% RH	± 2%
Plantower PMS5003	PM1.0 PM2.5 PM10	0 µg / m³	1000 µg / m³	± 5%

In the KIT version, the PM Smoggie sensor comes disassembled. The aim is to be used as teaching material by students in laboratory activities. Such a lab activity presents the students with modern digital technologies for operating an automatic sensor. At the same time, assembling the SMOGGIE KIT sensor familiarizes students with specific assembly tools and techniques like soldering stations, tinning, 3D printing, programming, and testing.

All sensors are interconnected in a network (uRADMonitor) focused on continuous environmental surveillance. It generates fully transparent open data used to monitor the air quality. The data is accessible in real-time via an API interface directly from the cloud. Environmental data recorded by sensors covers an extensive area and is provided in real-time at a one-minute resolution.

The main components of the kit

Each KIT comes in a cardboard box containing the following:

1. motherboard with glued electronics and double-sided tape
2. particle sensor type Plan tower PMS5003
3. particle sensor wire connector
4. 220V 5V adapter for powering the device
5. micro-USB cable for programming and power supply
6. documentation on paper
7. plastic cover

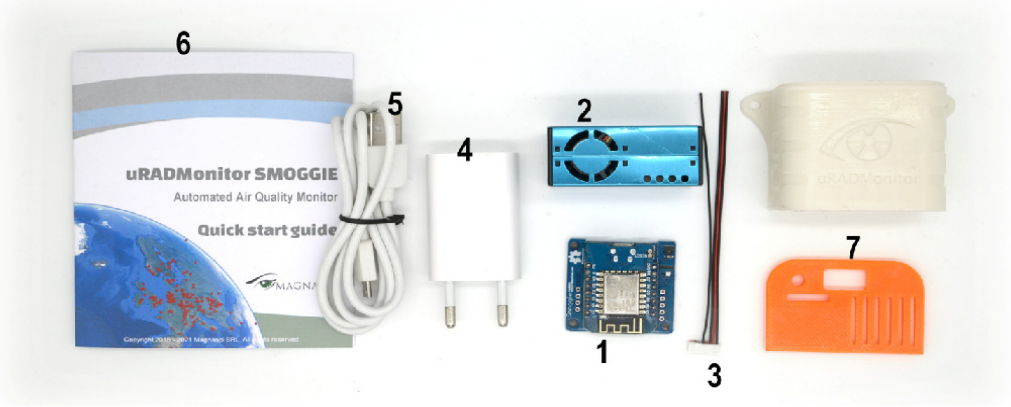


Figure 6.2. *The main components of the PM Smoggie kit*

Necessary time

The average assembly time is about 30 minutes, including tinning the wires, fixing the sensor to the base plate, and inserting it into the plastic case. Printing the case takes about one hour and 30 minutes but varies depending on the FDM or SLA technology used. Programming the device takes a few minutes using Arduino for Windows, Linux, or Mac OS.

Data upload

Data is collected automatically, with a temporal sampling of one minute. One can configure the interval in firmware or via USB terminal commands to cope with the deployment purpose (e.g., mobile units need faster sampling, while remote units operating on lower power or limited bandwidth (or both) need a reduced sampling rate). The sensors connect to the internet via several connectivity means: depending on device type, these include cable links via Ethernet or radio links including GSM, WIFI, LORAWAN, HELIUM, or Bluetooth Low Energy. One can access such recordings remotely via the uRADMonitor API or decentralized on users' local networks.

Data access

The SMOGGIE device only needs to be connected to the power supply and the WIFI network, and the data will be immediately available:

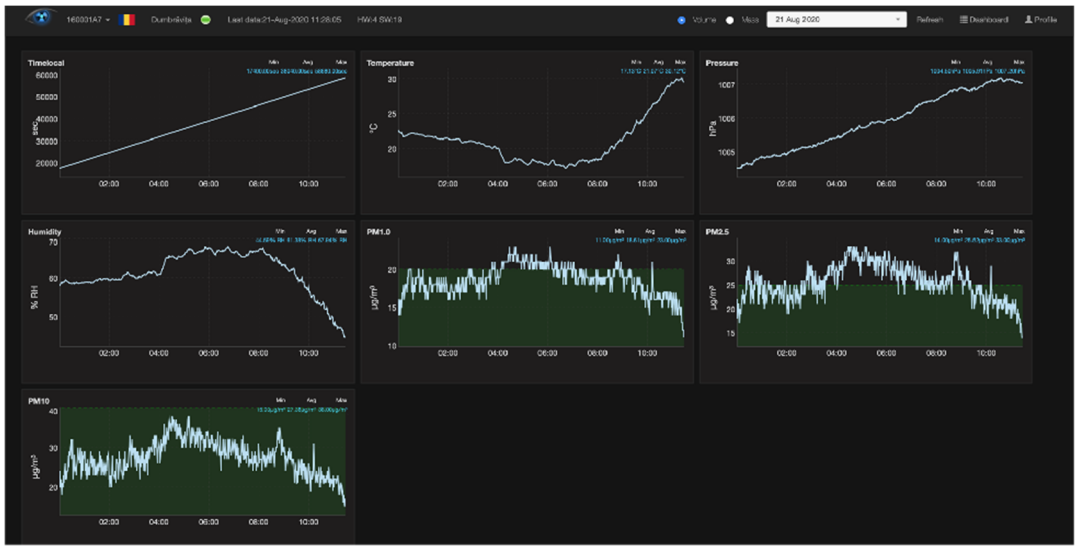


***Figure 6.3**. Data measurements in real-time*

Data sets can be accessed directly for display in third-party software applications, for integration with other systems (Home Assistant, Alexa, etc.), or display on information panels.

The data can be accessed in two ways:

- ***Local access***

This applies where the unit is part of a LAN. The unit presents an internal web page accessible through port 80, which can be opened in a standard Internet browser. Open the unit's LAN IP to access the content on a computer or phone. The web page served is as follows:

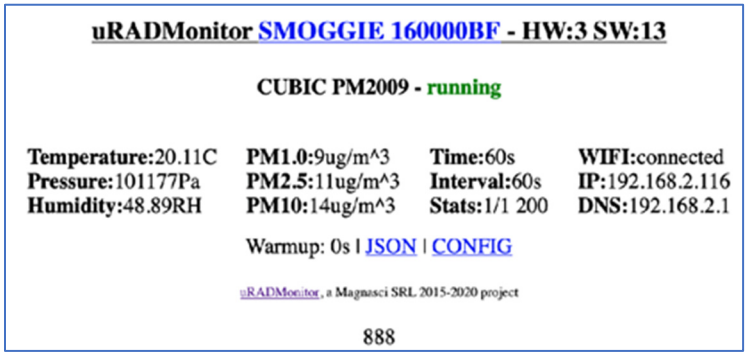


***Figure 6.4**. The web page*

The internal page of a SMOGGIE device showing measurements and other technical information

The JSON link at the bottom of the web page is used to access data sources in JSON format (a simple data format that is used to transfer and store data and is independent of programming language). Since the sensor is connected directly to the network, it is not possible to use offset processing on the server to process the acquired data, so only the raw data is available.

For this reason, this access mode is not the preferred way, and additional compensation must be applied (e.g., temperature compensation to correct internal heating, other corrections, etc.). This functionality is provided instead for debugging and decentralized operation in critical situations such as server failure or failure. For more details on direct data access, go to https://www.uradmonitor.com/direct-data-access/

- ***Data access through the Server REST API interface***

This is the preferred method of data access. The API does not prompt the client to know anything about its structure. Instead, the server must provide the client's information to interact with the service. An HTML form is an example of this: The server specifies the location of the resource and the required fields. The browser does not know in advance where to send the information and does not know in advance what information needs to be sent. The server fully provides both forms of information.

The API is called for both directions of data transfer, data transmission = upload and download = download. When we talk about data access, we mean download. The sensors use the API to upload measurements to the server for further processing and storage in the database. The API is then used to access data from the front end, mobile app, or third-party systems that need a dataset. For more information about API, please consult the following links: https://www.uradmonitor.com/api, https://www.uradmonitor.com/dashboard/.

References

1. Udristioiu, M. T., Velea, L., & Motisan, R. (2023). First results given by the independent air pollution monitoring network from Craiova City, Romania. In AIP Conference Proceedings (Vol. 2843, No. 1). AIP Publishing.
2. Velea, L., Udriștioiu, M. T., Puiu, S., Motișan, R., & Amarie, D. (2023). A Community-Based Sensor Network for Monitoring the Air Quality in Urban Romania. Atmosphere, 14(5), 840. MDPI AG. Retrieved from http://dx.doi.org/10.3390/atmos14050840
3. https://www.uradmonitor.com/products/
4. https://www.uradmonitor.com/wp-content/uploads/2022/08/datasheet_smoggie_v5-stev_compressed.pdf
5. https://www.uradmonitor.com/direct-data-access/
6. https://www.uradmonitor.com/api
7. https://www.uradmonitor.com/dashboard/

These sections (6.1.2., 6.2, 6.2.A., 6.2.B., 6.3.) were written by Silviu Constantin Sararu, from the University of Craiova, Romania

6.1.2. How to make a PM Smoggie sensor using a kit?

A SMOGGIE-PM sensor is composed of four main parts:

- a box that is made by 3D printing;
- ESP8266 WeMos D1 Mini – microcontroller;
- Smoggie-PM PCB;
- Plantower PMS5003 – a sensor that measures particulate matter (PM) (1µm (PM1), 2.5µm (PM2.5) si 10µm (PM10)).

PM is used for a mixture of solid particles and liquid droplets in the air. The PMS5003 particulate sensor is a laser scattering sensor that gives digital outputs of the concentration of PM based on Mie Theory.

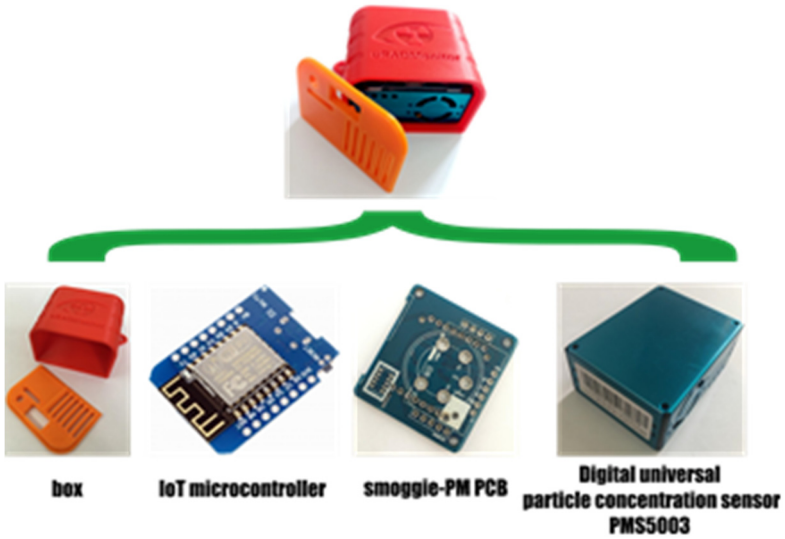


***Figure 6.5**. The main parts of the PM sensor*

About microcontroller

A microcontroller is an integrated circuit device that controls other portions of an electronic system, usually via a microprocessor unit, memory, and some peripherals. The most common way to refer to this category of integrated circuits is microcontroller with the abbreviation MCU (= microcontroller unit) or occasionally µC (µ =micro). Microcontroller is a well-chosen name because it emphasizes the defining characteristics of this product category.

microcontroller = micro + controller,

The prefix micro implies smallness while the term controller implies an enhanced ability to perform control functions.

The microcontrollers are optimized for embedded applications requiring processing functionality and agile, responsive interaction with digital, analogue, or electromechanical components. The microcontrollers have played a fundamental/dominant role in the technological

revolution 4.0 that has shaped modern life. Microcontrollers are small, versatile, inexpensive devices that can be successfully implemented and programmed by experienced electrical engineers and hobbyists, students, and professionals from other disciplines.

What is worth mentioning is the fact that microcontroller ≠ computer because:

- a microcontroller performs a specific task one at a time,
- while the computer performs millions of instructions at a time.

About ESP8266 WeMos D1 Mini

The WeMos D1 Mini is based on the ESP8266 microcontroller, designed and produced by Espressif Systems in Shanghai, PRC. Based on the built-in bootloader, it is easy to flash the board with your program code. The notable features of ESP8266 WeMos D1 Mini that are worth mentioning here are

- easy to use for IoT projects with micro-USB connection and build-in WiFi (IEEE 802.11 b/g/n);
- low energy consumption in the deep sleep power mode (0.17mA) and therefore very well suited for battery-powered projects;
- fast processing power up to 160 MHz compared to 16 MHz for the ATmega328p (on Arduino).

You can find more information about the ESP8266 WeMos D1 Mini on the website producer: https://www.wemos.cc/en/latest/index.html. A feature description of the ESP8266 WeMos D1 Mini, along with a diagram showing the arrangement of pins on an integrated circuit and their functions, can be found at URL https://diyi0t.com/esp8266-wemos-d1-mini-tutorial/.

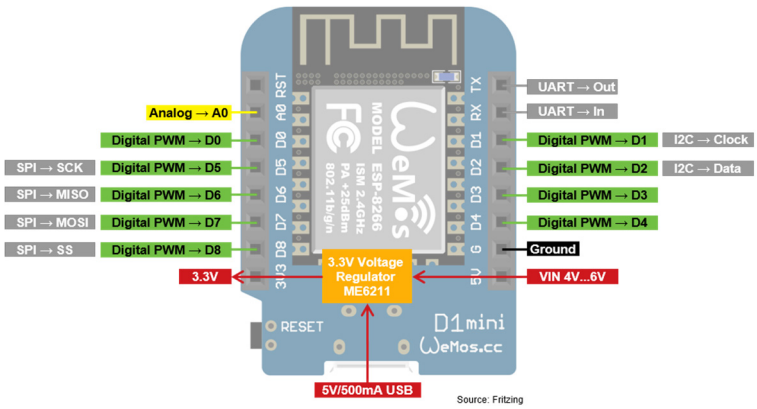


***Figure 6.6**. ESP8266 WeMos D1 Mini*

6.2. Laboratory setup

6.2.A. How to Power the ESP8266 WeMos D1 Mini?

Micro USB. USB cable is the most popular and easiest way to power the microcontroller. The standard USB connection delivers 5V and allows you to draw 500mA.

5V Pin. You can use the VIN pin if it uses an external power supply like a battery or laboratory power supply. The voltage must be between 4.3V and 6V. Therefore, you can power the WeMos D1 Mini with a LiPo battery with a JST connector in combination with a battery shield for the WeMos D1 Mini

https://www.wemos.cc/en/latest/d1_mini_shield/battery.html.

3V Pin (not recommended). The board has a built-in voltage regulator (ME6211) that provides a stable 3.3V for the WeMos D1 Mini and the corresponding pins. The voltage regulator is connected to the 5V pin and the USB port. Since the 3.3V pin is directly connected to the ESP8266 after the voltage regulator, the board can also be powered via a stable 3.3V voltage on the 3.3V pin.

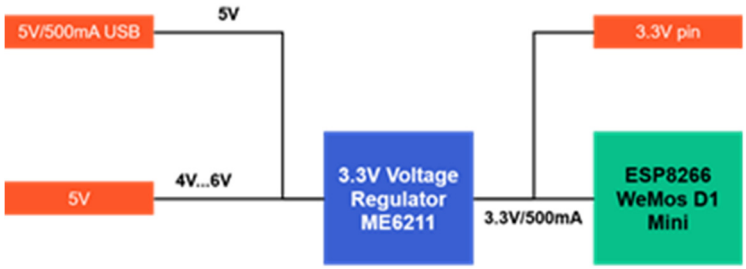


***Figure 6.7**. Voltage regulator*

6.2.B. How to flash the program code on the ESP8266 WeMos D1 Mini

You can use the Arduino IDE to flash your program code to the ESP8266 WeMos D1 Mini.

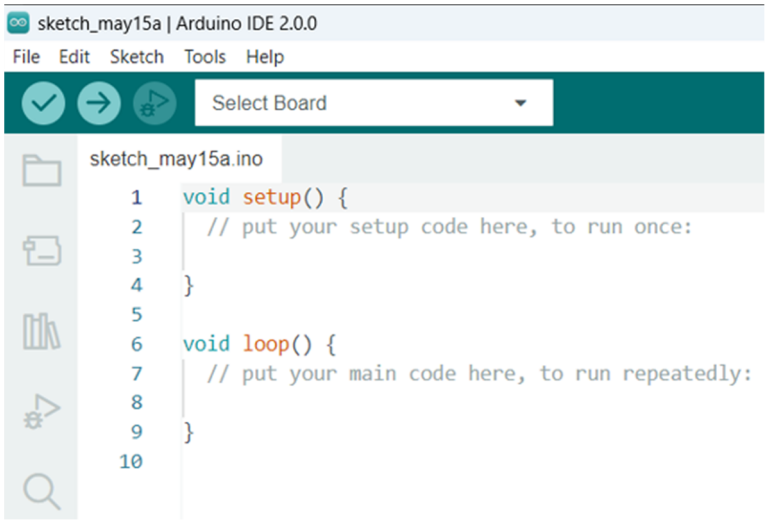


***Figure 6.8**. Printscreen Ardino IDE*

To do this, the following steps are taken:

- In the Arduino IDE menu, click on File → Preferences and insert the following URL in the field Additional Boards Manager URLs:
  http://arduino.esp8266.com/stable/package_esp8266com_index.json

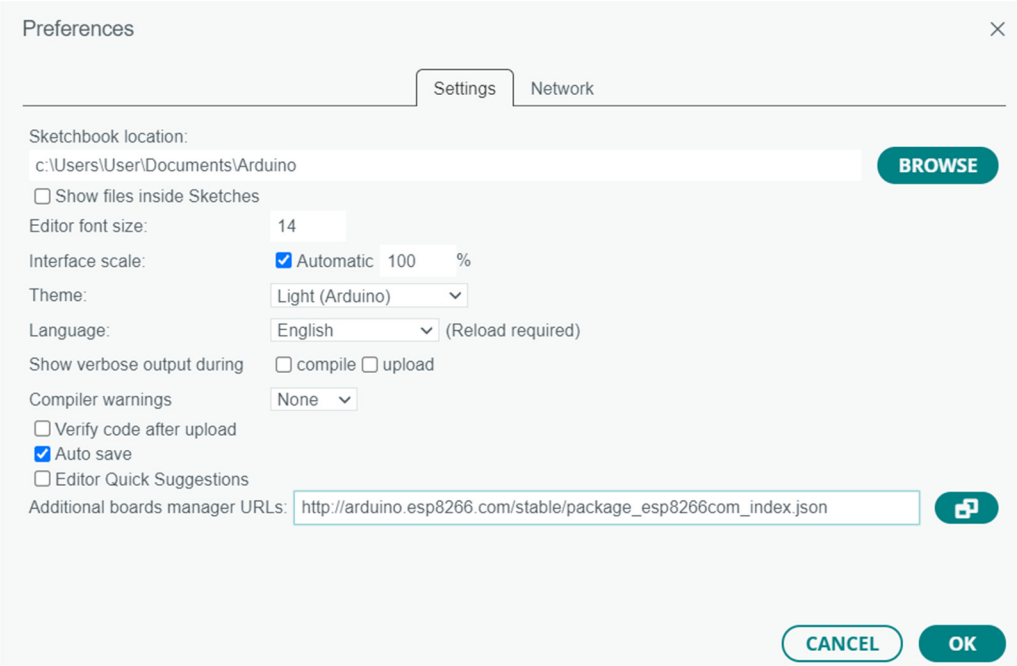


***Figure 6.9**. Printscreen File → Preferences and insert the URL in the field Additional Boards Manager URLs*

- Install the necessary packages on the left side of the Arduino IDE, click Board Manager, search esp8266 by ESP8266 Community, and install the latest version of the board.

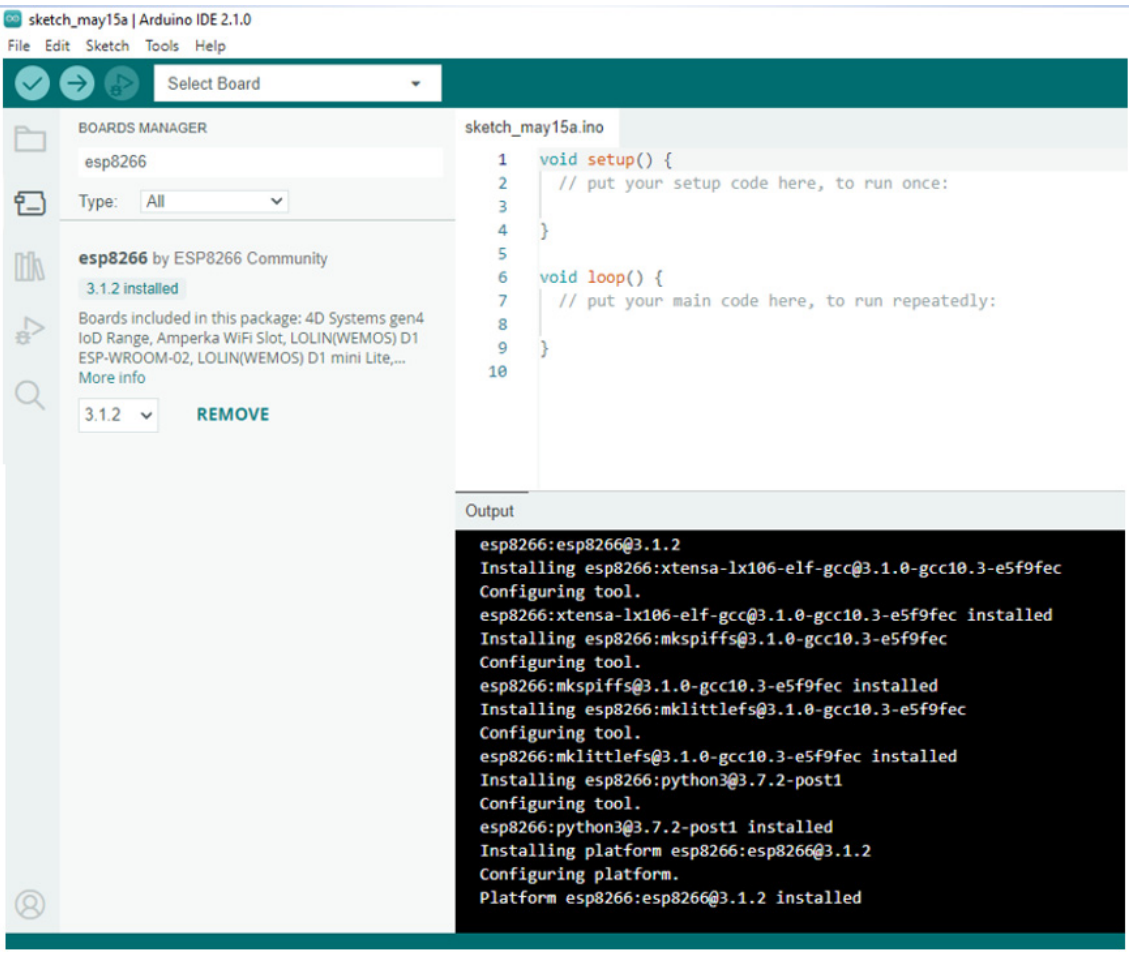


***Figure 6.10.** Printscreen esp8266 by ESP8266 Community*

6.3. Tasks

We also need a soldering station, a wire cutter, a picoblade cable, and a double adhesive tape to build the sensor. Usually, the WeMos board is usually soldered on the Smoggie-PM PCB's top side. The wires of the Picoblade cable are shortened to a length of a few cm. The tips are stripped to prepare them for tinning.

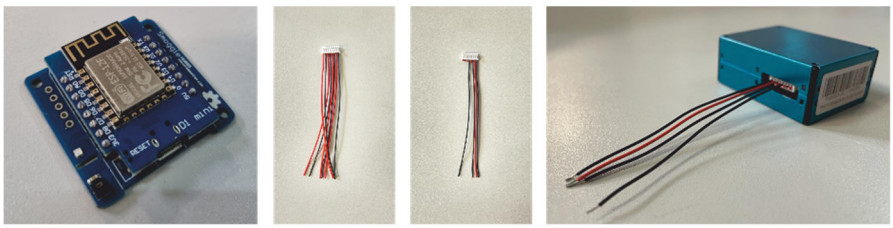


Figure 6.11. *The WeMos board on the Smoggie-PM PCB*

The stripped ends of the connector wires are inserted into the holes of the base plate, as in the picture below. Then, using the soldering station, they are tinned to fix them. Solders must be flat, not exceeding in height the thickness of the double-sided tape.

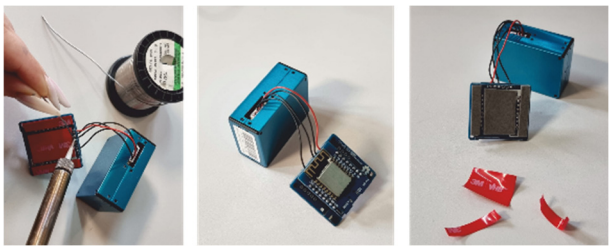


Figure 6.12. *Soldering wires*

With the help of the double-adhesive tape, stick the assembly consisting of the WeMos board and smoggie-PM PBC to the case of Plantower PMS5003 sensor, proceeding as follows: align the base of the assembly consisting of the WeMos board and Smoggie-PM PBC cantered on the surface of the sensor, taking care that the edge with the power plug is right on the edge of the sensor, as in the picture below. The back wires lay parallel without crossing over the sensor area. Insert the assembly into the case and staple the cover.

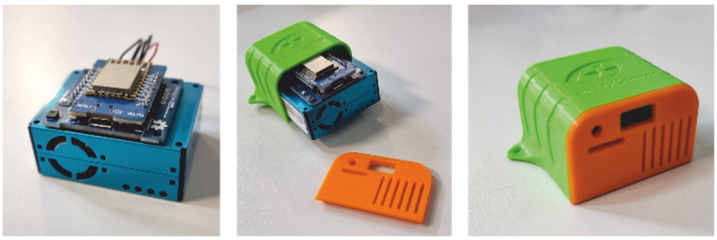


Figure 6.13. *The sensor inside of its cover*

The electronic part is ready; you can flash the board with the program code and install the sensor.

References

1. https://www.uradmonitor.com/products/
2. https://www.wemos.cc/en/latest/index.html
3. https://www.epa.gov/pm-pollution/particulate-matter-pm-basics
4. https://diyi0t.com/esp8266-wemos-d1-mini-tutorial/
5. https://www.arduino.cc/
6. G. Mie, Beitr¨age zur optik tru¨ber medien, speziell kolloidaler metall¨osungen [Contributions to the optics of turbid media, particularly of colloidal metal solutions]. Ann. Phys., 377, 25(3), 1908
7. https://ioct.tech/edu/sites/default/files/2019-04/PMS5003-Educational%20Version%202.pdf

Chapter 7. 3D PRINTING. BASIC CONCEPTS ABOUT HOW TO PREPARE YOUR MODEL FOR 3D PRINTING

This chapter was written by Iulian Petrisor from the University of Craiova, Romania

7.1. Theory

Technological progress has allowed us to print 3D models easily, quickly, and, importantly, economically. Historically, there have been many technological breakthroughs such that in the last 10–15 years, we have seen a massive, global spread of the possibilities to print 3D models. New software, bookstores, dedicated websites appeared, and many free, even scientific journals (journals) related to these new technologies. The first books dedicated to the field started to appear after the 2010s (such as [1–3] or some of the more recent ones [4–11].

Many types of materials are used in 3D printing, from plastics, various resins, metals or metal powders, ceramic materials, composite materials (such as in construction), and more recently, food.

Relatively common types of materials used in standard 3D printers:

- ***Plastic materials:***
  - PLA (Polylactic acid);
  - ABS (Acrylonitrile Butadiene Styrene);
  - Nylon (Aliphatic polyamide);
  - PET (Polyethylene terephthalate).
- ***Epoxy resins;***
- ***Metallic materials:***
  - Metal powders (mixed with other materials that are later removed);
  - Steel, nickel, bronze, copper, etc. (which can be submitted directly).

We also emphasize that depending on the material or technology in the printing process, it is essential to consider some effects on the health of people in the premises where the 3D printer works. Some printers have a closed, thermally controlled enclosure, as well as the possibility of evacuating some gases released as a result of the technological process (for example, in the case of using ABS, gases are released that must be evacuated), or if the 3D printed product will be used in the food industry because some materials are not suitable for it. Thus, we are considering a printer dedicated explicitly to the food industry (meeting additional standards) to print, for example, chocolate products or PET printers to print models that can store food, etc. So, it is essential also to know where or how the final (3D printed) product will be used.

In the most common 3D printers, the technology is based on melting and then fusing plastic materials; basically, using energy melts the material in a particular head (which dictates the resolution of the final product) and through an application of plastic extrusion will obtain, usually through successive depositions, layer by layer, the designed 3D model.

7.2. Laboratory setup

Possible software used for 3D printing:

- *Blender*
- *Tinkercad*
- *FreeCAD*
- *DesignSpark Mechanical*
- *Autodesk Fusion 360*
- *SolidWorks*
- *Mathematica, etc.*

Each 3D printer contains, as a rule, software with which it is possible, on the one hand, to develop and design 3D models and, of course, on the other hand, to print models made with other software that has been exported in a format appropriate, i.e., to be "understood" by the printer. Usually (beginner students and anyone who wants to learn), they start with free software, often online. It provides first-time users with a range of examples and small tutorials to understand how to design and print a simple 3D model.

One can start with *Blender or Tinkercad* (which we will get to later). The purpose of this material ***is not*** to teach 3D design. This is an intuitive, systematic process and is highly dependent on the complexity of the 3D model. There are tutorials and helpful tips in every software. The structure of the menu and the help are fundamental in achieving the desired model. Open-source libraries are also handy. The basic concepts for common choices in 3D printing (design, model, material type, or technology) are systematically described in Chapters 2-4 of ref. [8].

Design – the primary stage of 3D printing. It represents the transposition of an idea or concept into a physical object. Sometimes unachievable directly! By default, we are referring to 3D printers with the most used filament.

At this stage, we will consider:

- what sizes will have the final product; we correlate with the maximum size of the models we can produce with the 3D printer we have;
- the type/size of the print nozzle, i.e., the resolution we will have (also dependent on the printer owned);
- Avoid some very sharp edges or too fine (thin) surfaces.

Tinkercad is a straightforward and intuitive software (*especially for beginners*). It does not require local installation but creating an account associated with an email address, preferably connected to the education system, mainly oriented towards education. More mathematical software can be used if the user's needs are for the realization of very precise, elaborate, or complex projects or models. For example, we mention the need to produce surface models with rigorous mathematical equations, such as having a parabolic surface (from the 3D model) whose equation is known. Thus, we will turn to software that has, in the design area, the possibility of entering equations (*Mathematica, DesignSpark Mechanical*, and many others).

In Figure 7.1, we have a straightforward picture of the working plan in ***Tinkercad***. This mini wind turbine will later be attached to a mini electric motor (which can also be used as an electric generator).

We further highlight the following elements, which are relatively common to all 3D design software (for 3D printing):

- A distinct color highlights the workspace, and, very notably, it can be marked, thus understanding the scale and size of the 3D model. Here, it is marked in mm. It can be

seen very quickly that we are working on an object with a maximum size of 90 mm (9 cm). It is essential, implicitly, to have very clearly in mind the size of the model;

- We can rotate the object (model) in space according to the three axes. But, let us keep in mind, all the time when designing, that 3D printing will be done layer-by-layer. So, we have the printing plan (of the future printer) in mind, and usually, even what type of printer (plastic, metal, or resin) we will have available, the maximum size that printer can produce. If we must scale our model (to enlarge or reduce it, preferably to work at a 1:1 scale);
- Depending on its complexity, some parts of the model must be supported with excess material, which will be removed later. (It also depends on the printing technology! In this spirit, we will also do the design);
- The color or colors of the designed object are also important. 3D printed with 1, 2, or more print heads can be used, with different colors, possibly with different materials.

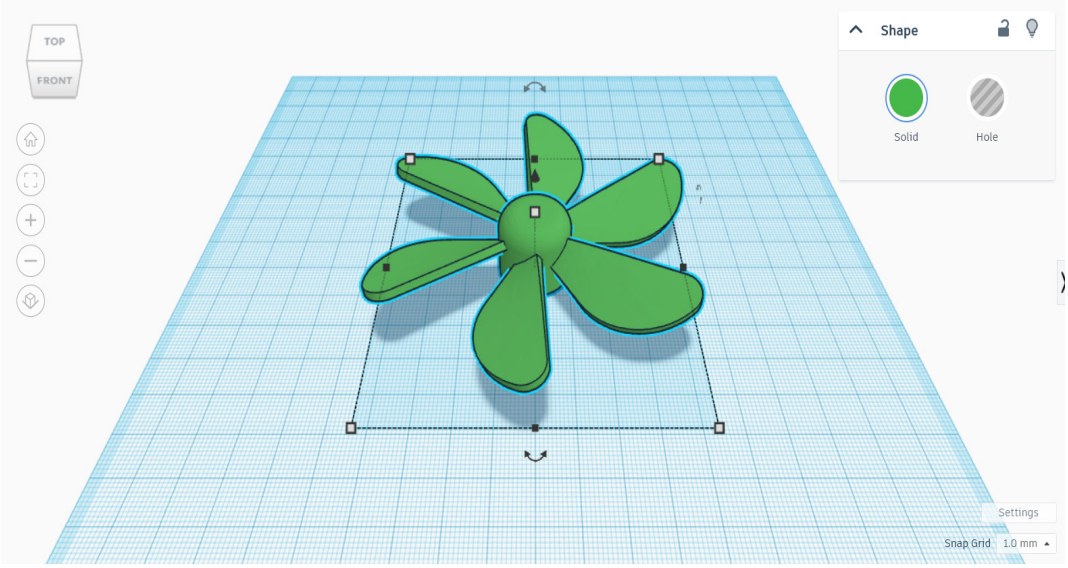


***Figure 7.1**. The workspace in Tinkercad with a mini-turbine as a 3D object*

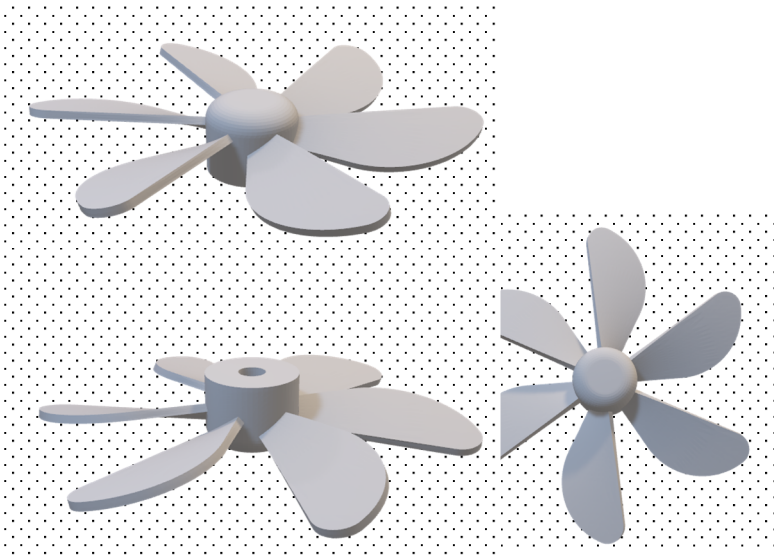


***Figure 7.2.** The mini-turbine in Figure 7.1, with different projections, is to imagine the complexity of the object. The model was composed of existing parts in the collection of objects made available in Tinkercad.*

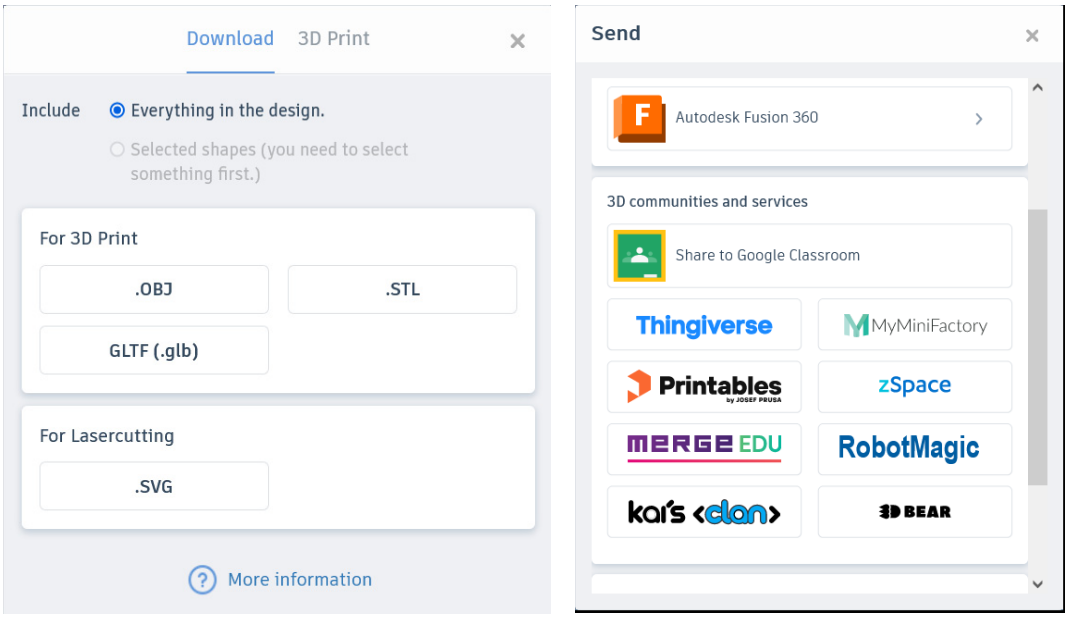


Figure 7.3. *After the 3D design, different operations can be performed from Tinkercad: export to a standard format (file type) for 3D printers, such as .OBJ or .STL (left figure) or export as an image (in a standard, usual format, for example .GIF) or sharing with different applications or platforms, we highlight only Autodesk Fusion 360 and Google Classroom (figure on the right)*

Later, the CAD program file will be taken over by the 3D printer. The designed model must be structured on two-dimensional layers (slices). That is why we need a program like Slicer 3D – which allows converting the 3D model into a language (blocks of data) that the 3D printer used in the process can read, understand, and execute. The user can use open-source 3D slicing programs (such as 3DSlicer or Ultimaker Cura).

We used a Raise3D printer with the 3D Slicer – IdeaMaker Software Slicer. We can prepare (for printing with Raise3D) the 3D models with IdeaMaker and later send the product that will be 3D printed to the printer through the print management platform.

Finally, for printing, we will send a g-code. This is unique (usually) for each model. In our case, it was used the simulation in Slicer.

Figure 7.4*. Covers and protective boxes for environmental sensors – you can see the scale (using the pen next to the two orange covers). They were made from PLA.*

7.3. Tasks

We focused on all the following images (Figures 5-14) from Slicer for 3D printing environmental sensor boxes. Several steps are required to ensure that 3D printing will be done right. From the start, we must consider the piece's orientation. Additionally, we cannot 3D print surfaces that are not supported by additional support (i.e., in the air, such as, for example – a ceiling or something similar). A print screen indicates each step of this task in the following:

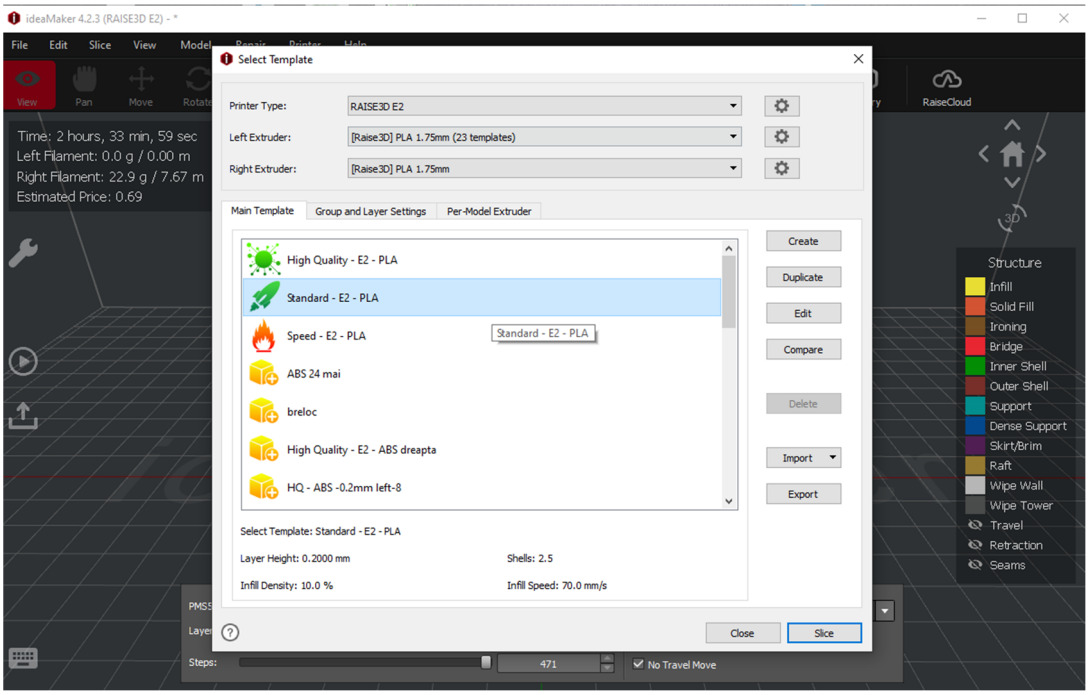


Figure 7.5. *Printing a box for environmental sensors. We use PLA as filament material. We started from a Standard variant (provided by the printer software), which we adapted to our model.*

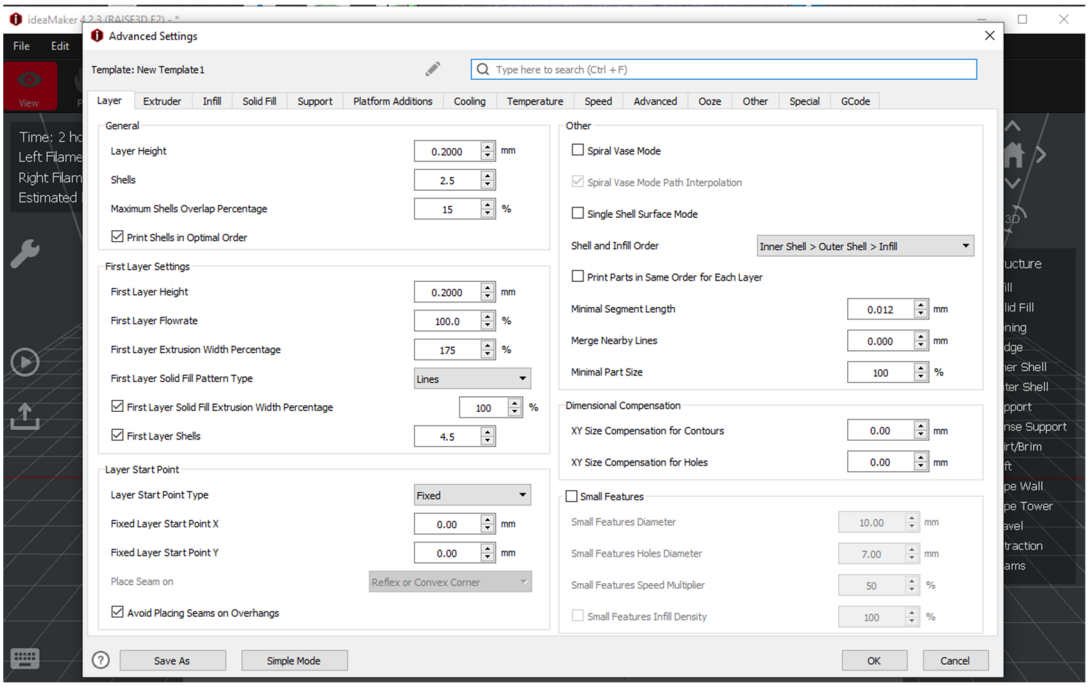


Figure 7.6. *Idem Fig. 7.5. I initiated commands to provide the printer with information on how to build the first layers (layers) and, later, the body of the model*

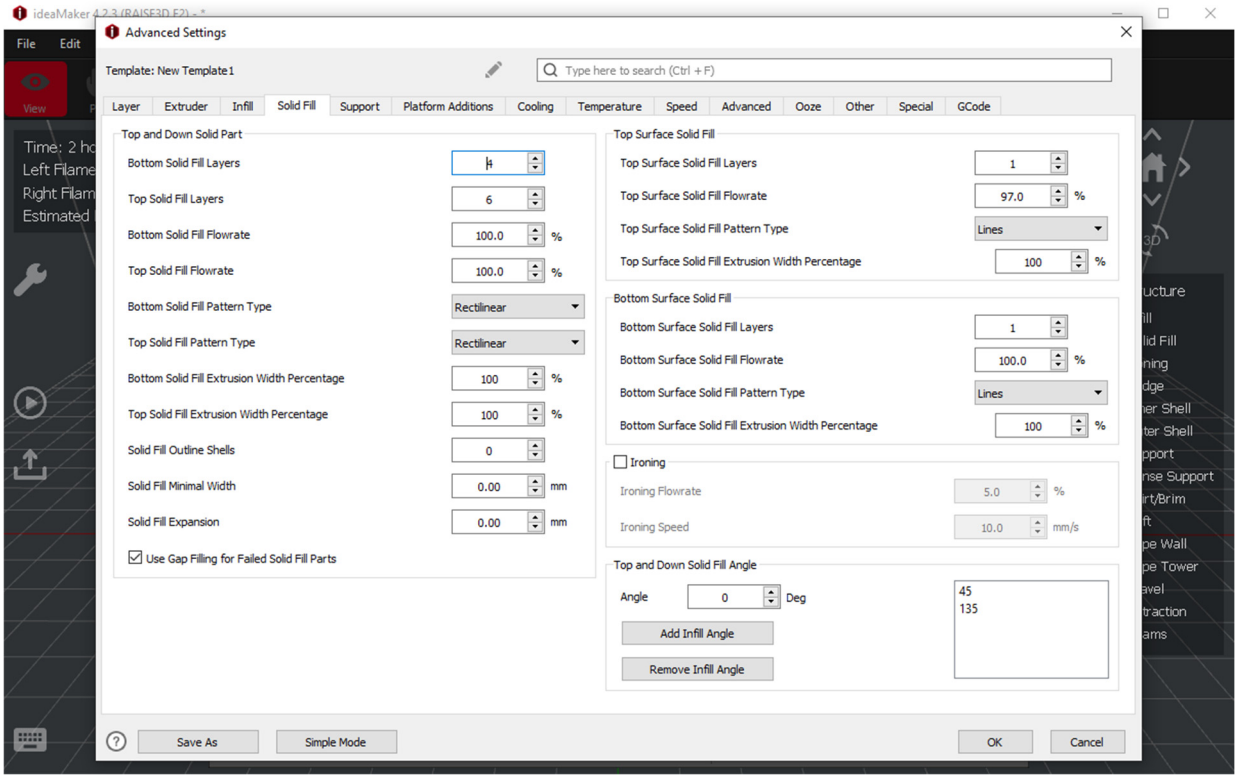


Figure 7.7. *Idem Fig. 7.5-6. We have determined what type of solid layer it will use in 3D printing.*

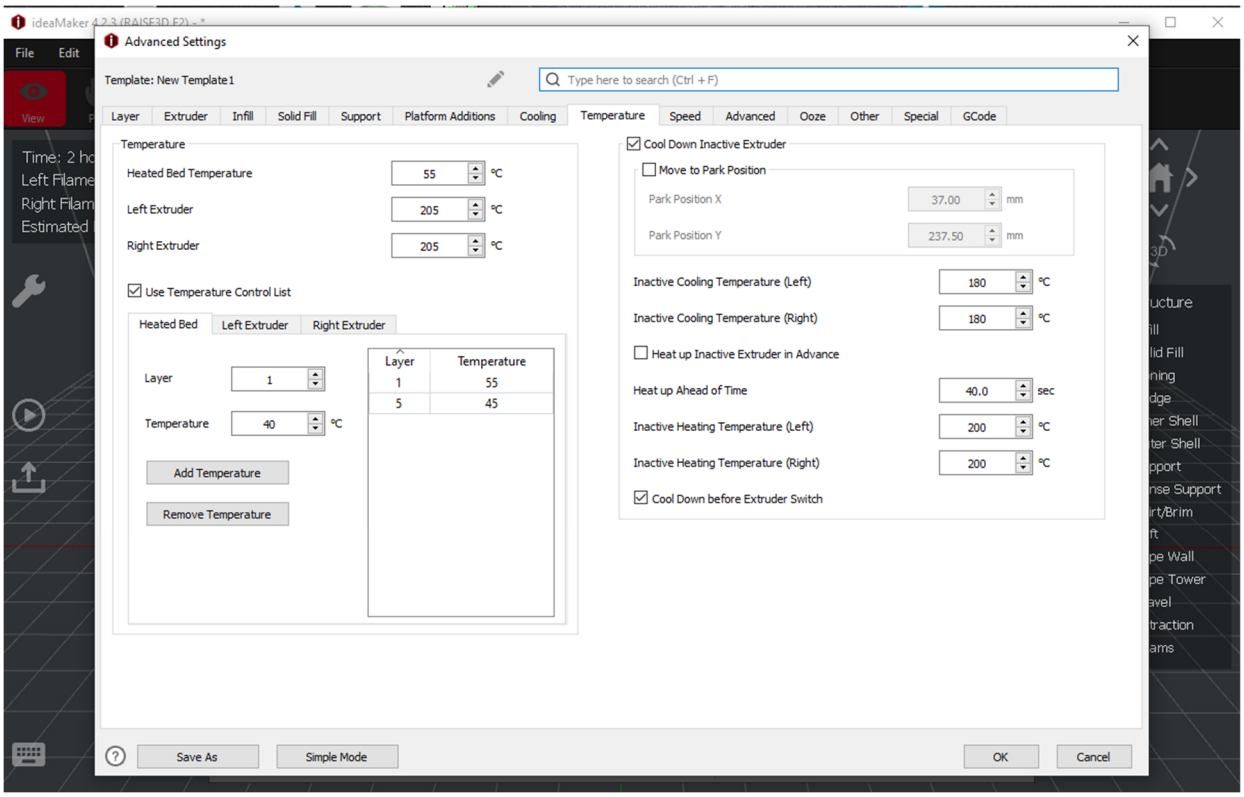


Figure 7.8. *Idem Fig. 7.5-7. We have established the temperatures used in 3D printing, the temperature of the support (heated bed), of the extruder used (we used the left extruder; the printer used has two print heads)*

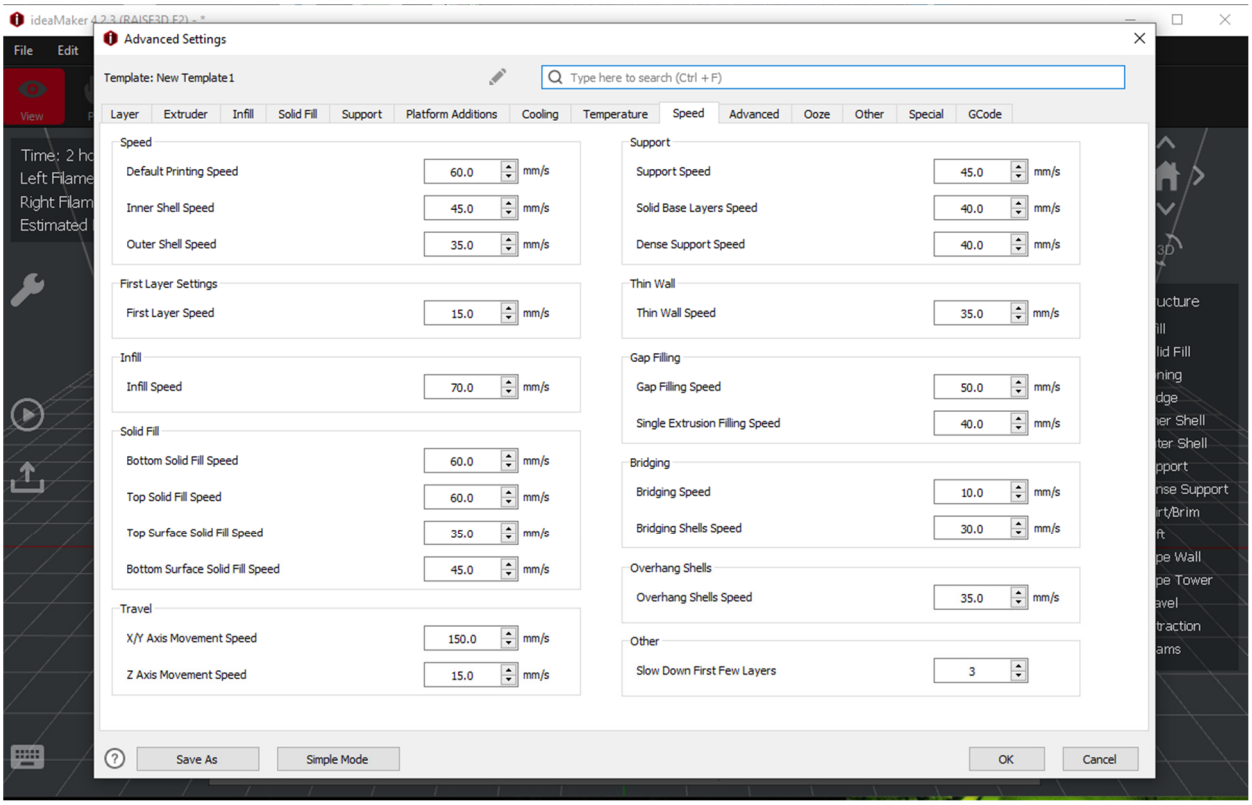


Figure 7.9. *Idem Fig. 7.5-8. We checked and set the print speed. It is preferable, especially for tall or asymmetric parts, to use the standard speed, not a very high speed, because it is possible, due to inertia, that the part will come off during printing – which automatically leads to a scrap.*

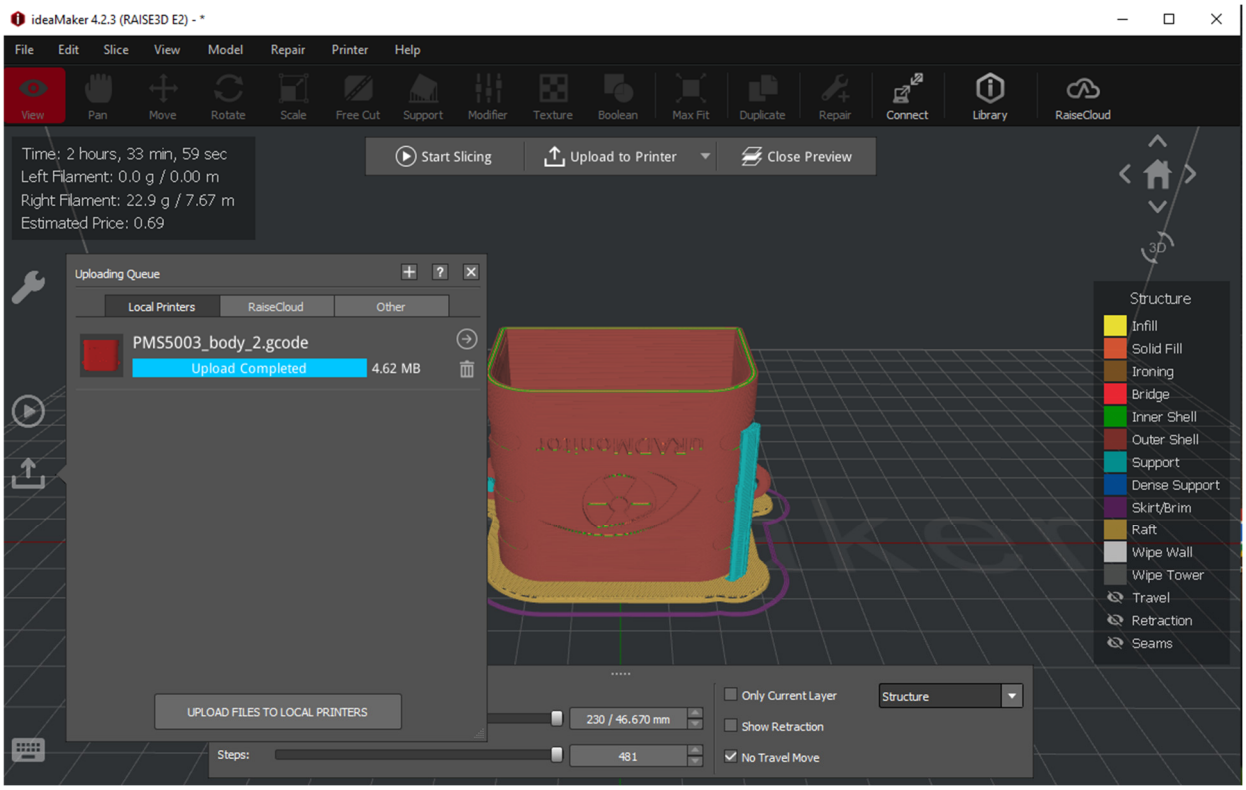


Figure 7.10. *Idem Fig. 7.5-9. After fixing all the parameters, we can produce the g-code. It is then wirelessly uploaded to the printer.*

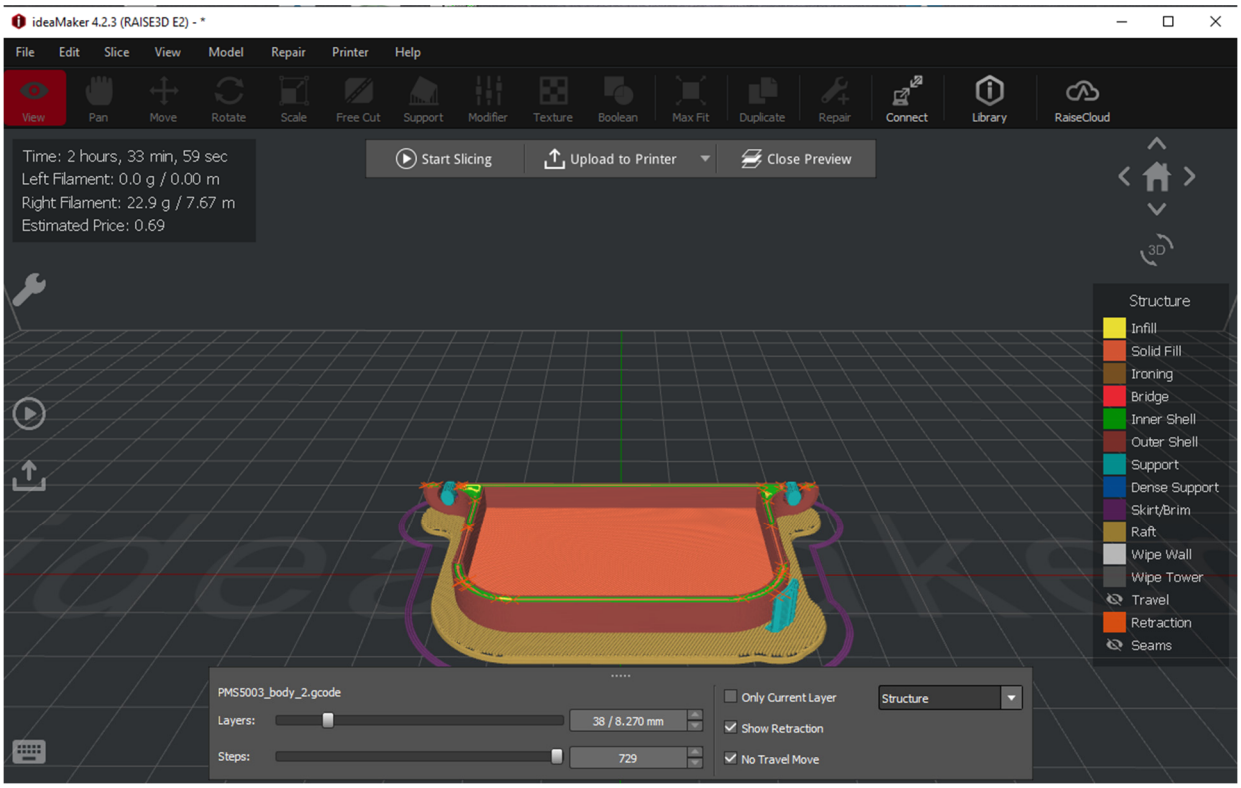


Figure 7.11. *A preview to understand how 3D printing our model will work*

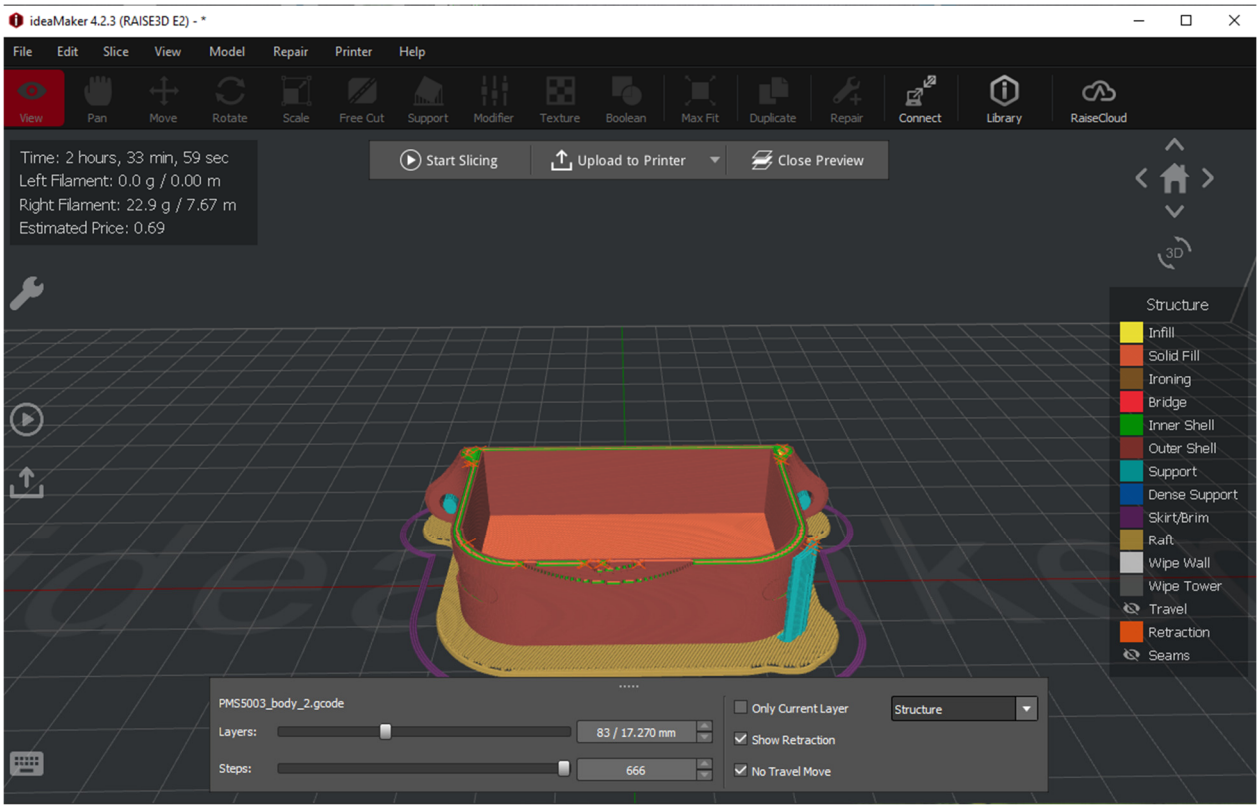


Figure 7.12. *idem Fig. 7.11. Different ways/colours can be chosen to understand how the base support (what will be thrown away), the 3D part (base – the desired model), and the additional support (what will be thrown away, removed – after the total realization of the 3D print) will be built.*

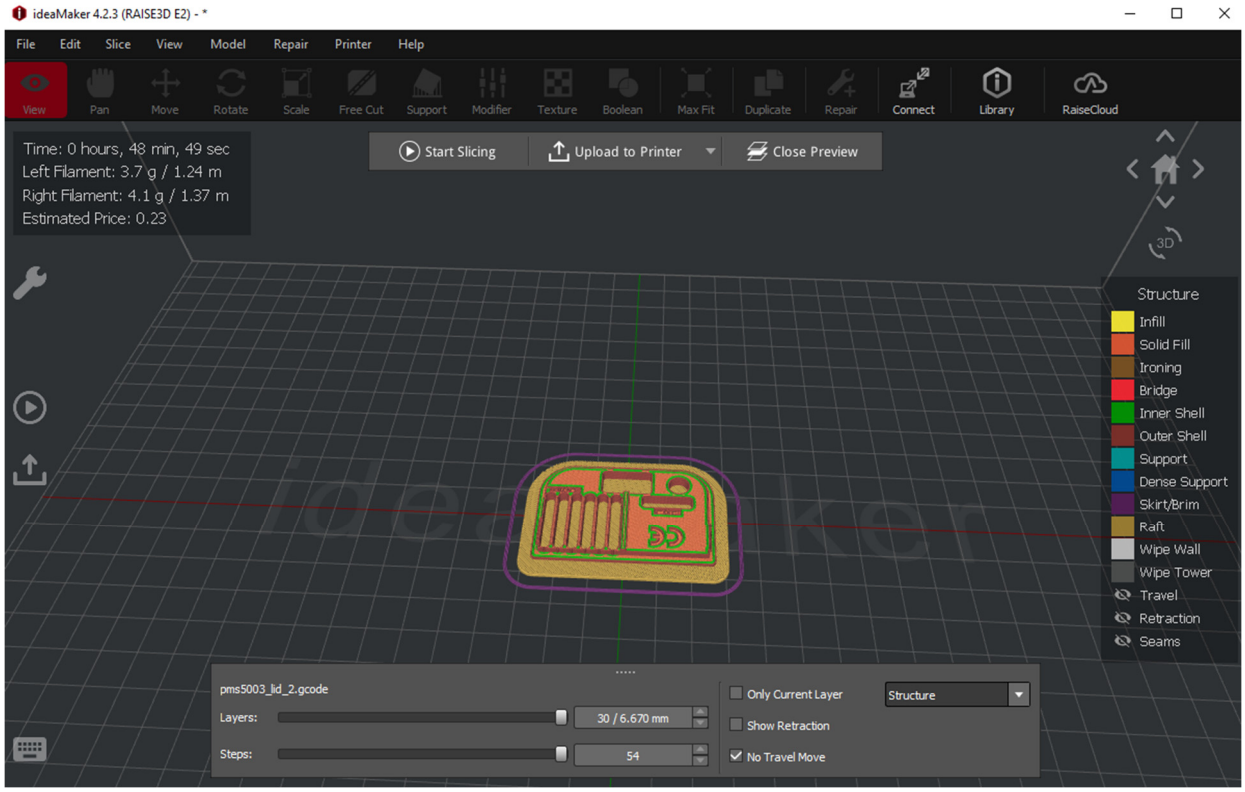


***Figure 7.13**. Slicer for the sensor box cover*

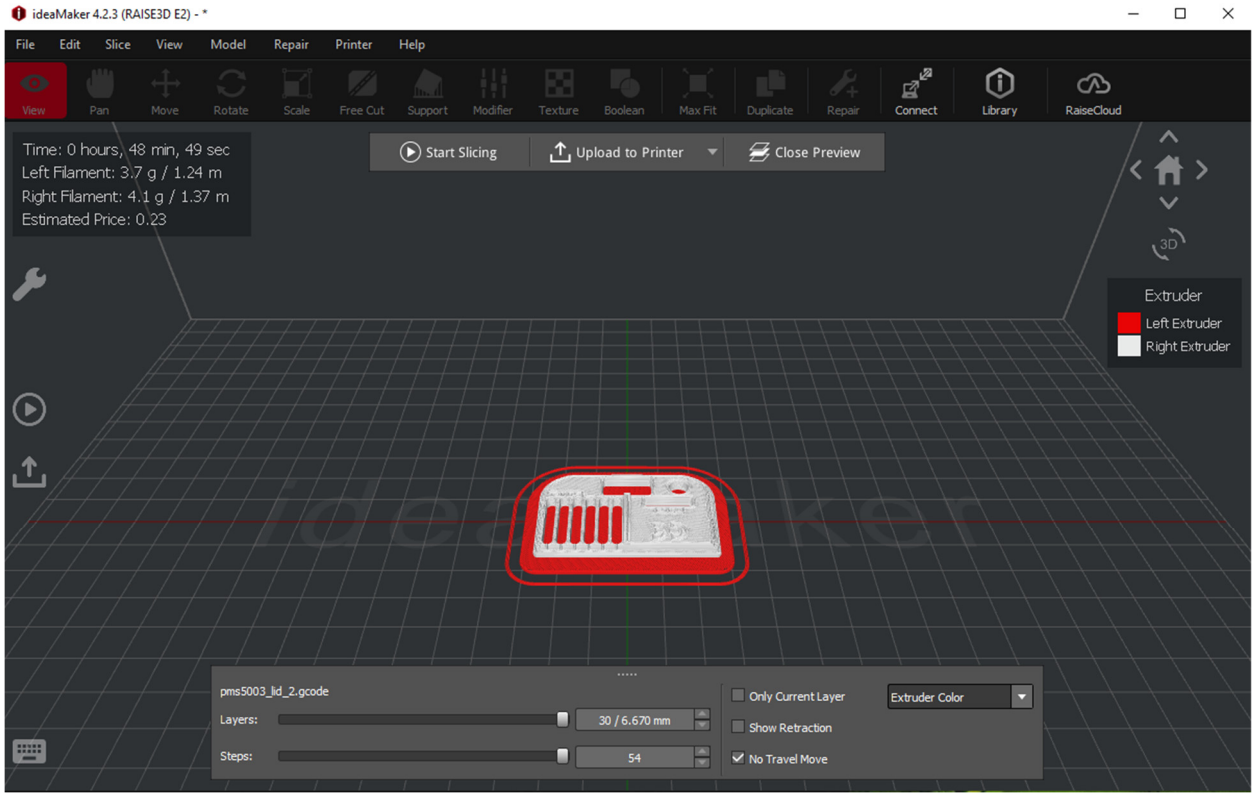


***Figure 7.14**. Idem figure 13. We used white PLA for the actual body of the 3D piece, with red backing*

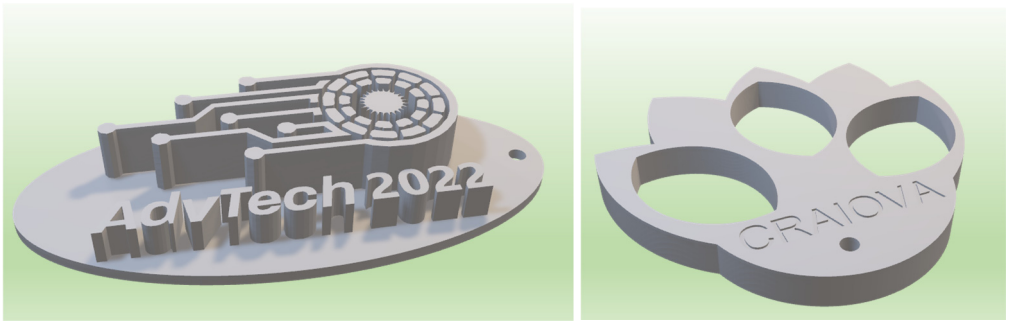


***Figure 7.15**. Two 3D models projected at the Erasmus+ Summer School (from 2022)*

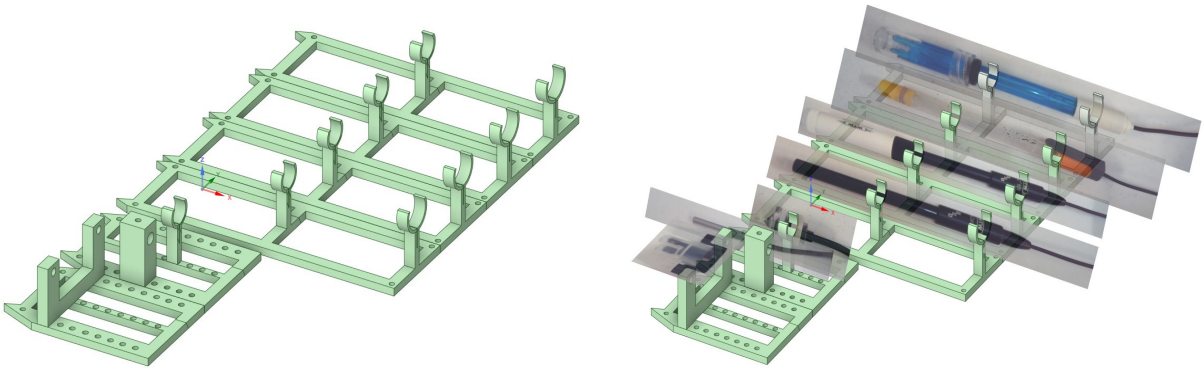


***Figure 7.16**. 3D printed holder for a set of sensors. On the left side, we have the 3D holder. On the right side, we have the sensor images superimposed over the standard base (from left-down to right-up, we have the following sensors: turbidity, temperature, level, conductivity, oxygen, additional sensor, pH)*

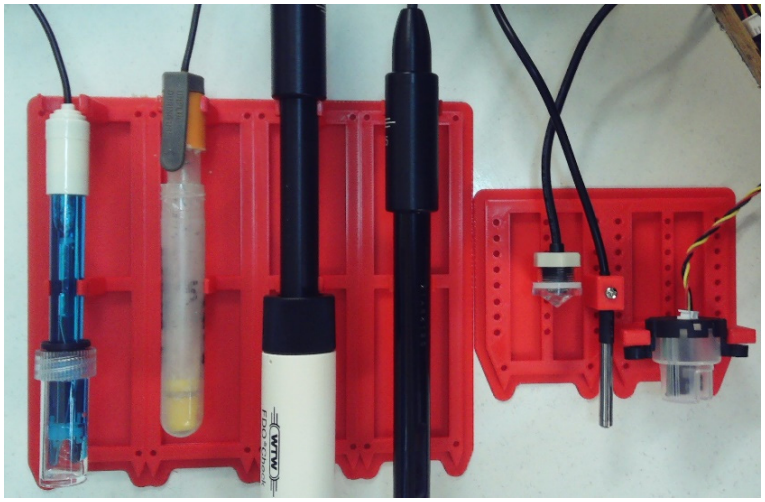


***Figure 7.17**. 3D printed holder for a set of sensors and the sensors (actual image). We focused our photography on the 3D-printed part.*

References

[1] S. Singh, Beginning Google Sketchup for 3D Printing, Publisher: Apress, 2010.

[2] G. Fisher, Blender 3D Printing Essentials, Publisher: Packt Publishing, 2013.

[3] H. Lipson, M. Kurman, Fabricated: the new world of 3D printing, Publisher: Wiley, 2013.

[4] L.W. Kloski, N. Kloski, Getting Started with 3D Printing: A Hands-on Guide to the Hardware, Software, and Services That Make the 3D Printing Ecosystem, Publisher: Make Community, LLC, 2021.

[5] C. Anandharamakrishnan, Jeyan A. Moses, T. Anukiruthika, 3D Printing of Foods, Publisher: Wiley, 2022.

[6] C. Zoccali, P. Ruggieri, F. Benazzo, 3D Printing in Bone Surgery, Publisher: Springer, 2022.

[7] S. Ehsani, P. Glauner, P. Plugmann, F.M. Thieringer, The Future Circle of Healthcare: AI, 3D Printing, Longevity, Ethics, and Uncertainty Mitigation, Publisher: Springer, 2022.

[8] Joanna Izdebska-Podsiadły, Polymers for 3D Printing: Methods, Properties, and Characteristics, Publisher: William Andrew, 2022.

[9] B. Rangel, A.S. Guimarães, J. Lino, L. Santan, 3D Printing for Construction with Alternative Materials, Publisher: Springer, 2023.

[10] M. Doddamani, H. S. Bharath, P. Prabhakar, S. Gururaja, 3D Printing of Composites, Publisher: Springer, 2023.

[11] Ram K. Gupta, 3D Printing: Fundamentals to Emerging Applications, Publisher: CRC Press, 2023.

CHAPTER 8. ACQUISITION OF THE DATASETS COLLECTED BY SENSORS

This chapter was written by Mihaela Tinca Udristioiu
from the University of Craiova, Romania

8. Theory

8.1. Sensor Network Description

The PM Smoggie sensors network made in the framework of the Erasmus+ project by students (during the summer schools), represents an expansion of the network that was realized in another three volunteering projects (Clear Air Craiova, Clear Air Oltenia, and Prevent), implemented in the recent years (2020–2023), at the University of Craiova. At Oltenia region level, the network includes an A3 sensor, an ionizing radiation sensor, a radon sensor, and 33 Smogie PM sensors. The network www.clearairoltenia.ro is part of a larger network, www.uradmonitor.com, which contains sensors of several types, including Smogie CO_2, Smogie Gas, Model Industrial, and Model City.

In the following image, you can see the uradmonitor.com sensors from Romania. Most of the sensors are of the Smogie PM type, the network being developed with the involvement of local communities who wanted to verify the data provided by local environmental protection agencies in the context of poor communication by local authorities. The map shows the sensors that measure PM2.5 concentration. The sensor's color is related to the pollution level where the sensor is placed (green means excellent air quality, and red awful air quality). From a utility point of view, the network is simple and intuitive.

In the framework of the Erasmus+ project Applying some advanced technologies in teaching and research, in relation to air pollution (contract no 2021-1-RO01-KA220-HED-000030286), the network was extended to Bulgaria, Slovakia, and Turkey, in the area of the partner universities.

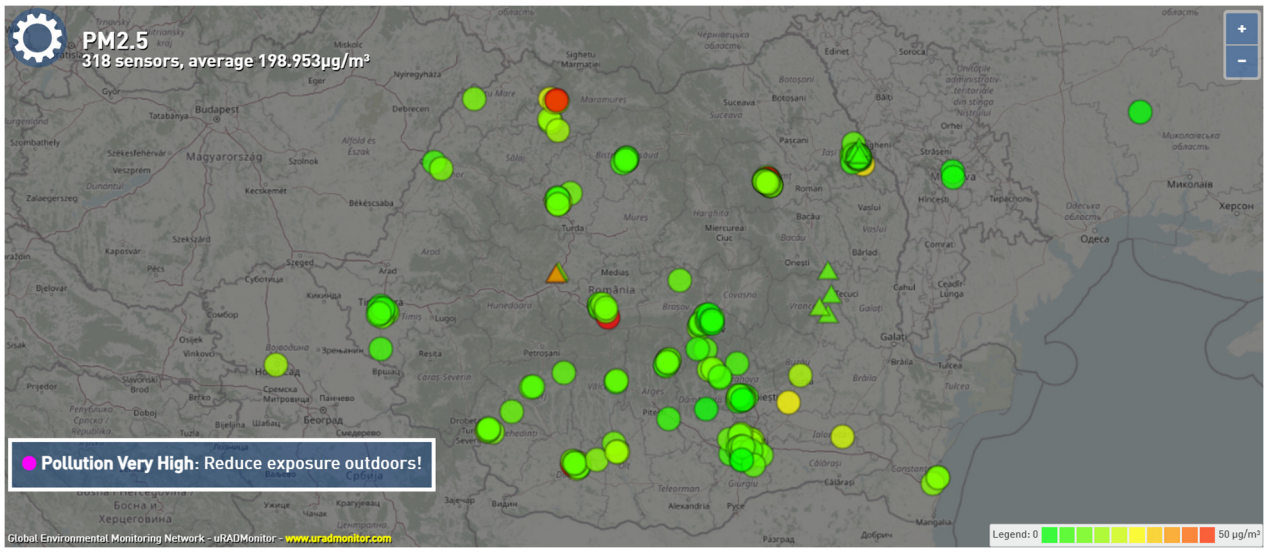


Figure 8.1. *The network of uradmonitor sensors in Romania*

8.2. Laboratory setup: Reading information from sensors

The network can be accessed by anyone, anytime, anywhere. The first step, obtaining information from the current day at the level of Romania, a region, or a city, is to select the sensor on the sensor map and then the desired parameter. For this purpose, the desired parameter is chosen, as shown in the following images.

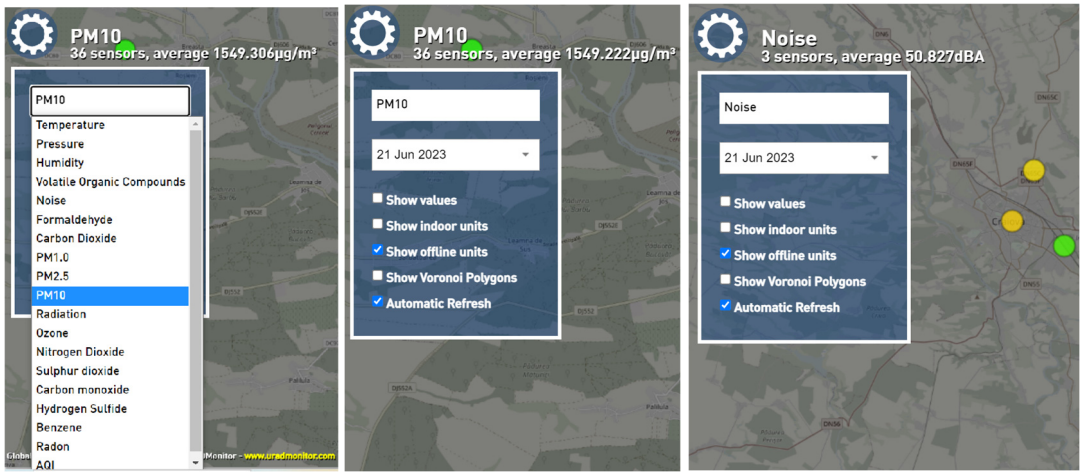


***Figure 8.2**. Choosing the parameter about which information is desired*

To obtain instant information from a particular sensor, a simple click on the sensor in a specific area (whose border turns blue) is enough, and immediately, information appears for a time interval between 0:00 and the current time about the selected parameter in the form of a graph (for easier visualization).

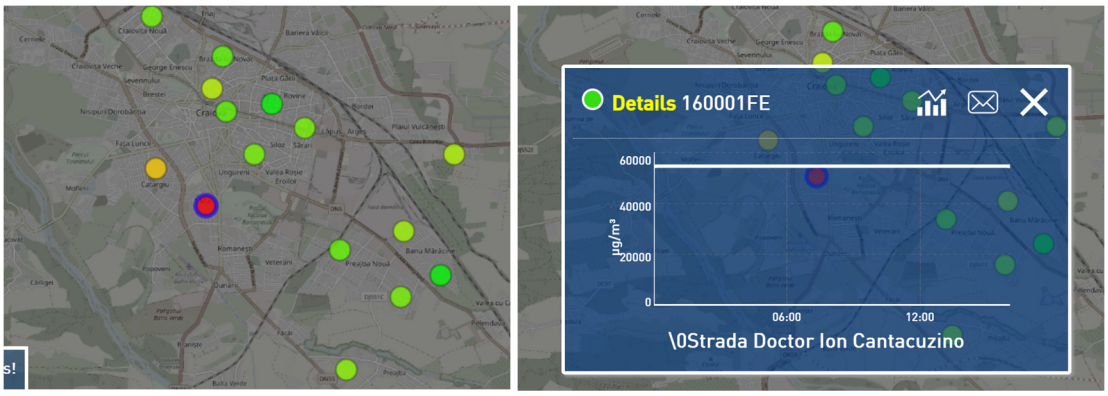


***Figure 8.3**. Selecting a parameter from a specific sensor and the information displayed*

When you click on a particular sensor, its measurements from that day to the current time appear. In the case of the PM Smoggie sensor, at the time and location where the sensor is located, the temporal representation of each parameter looks like the following figure:

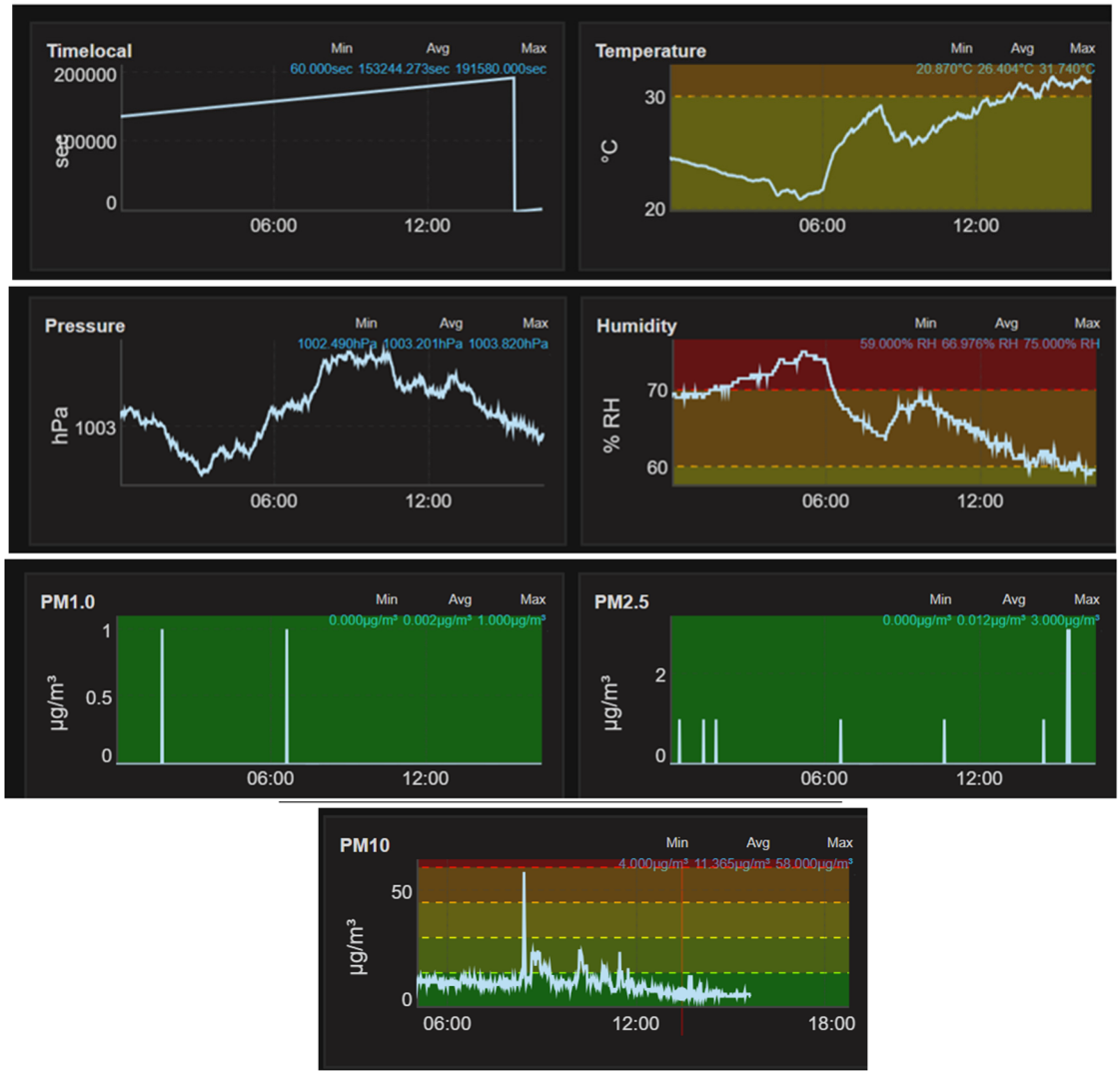


***Figure 8.4**. Graphical representation of meteorological parameters and concentrations of particles in the air*

Sensors in the network provide data over more extended periods in the form of type files CSV (Comma Separated Values). For a person who wants to obtain data sets longer than one day, it is necessary to submit a request by email to the owner of the uRADmonitor network, in which he mentions the sensor and the period for which he wants data. People who have purchased sensors can download data directly from their accounts. The PM Smoggie sensors measure three meteorological parameters (temperature, pressure, relative humidity) and three concentrations of particles suspended in the air (PM1, PM2.5, PM10).

8.3. Tasks

8.3.1. Datasets downloading

To download data, the user goes to the uRADmonitor website and clicks on the Dashboard button. The next step is to log in (Login) with your username and password, as in the following image.

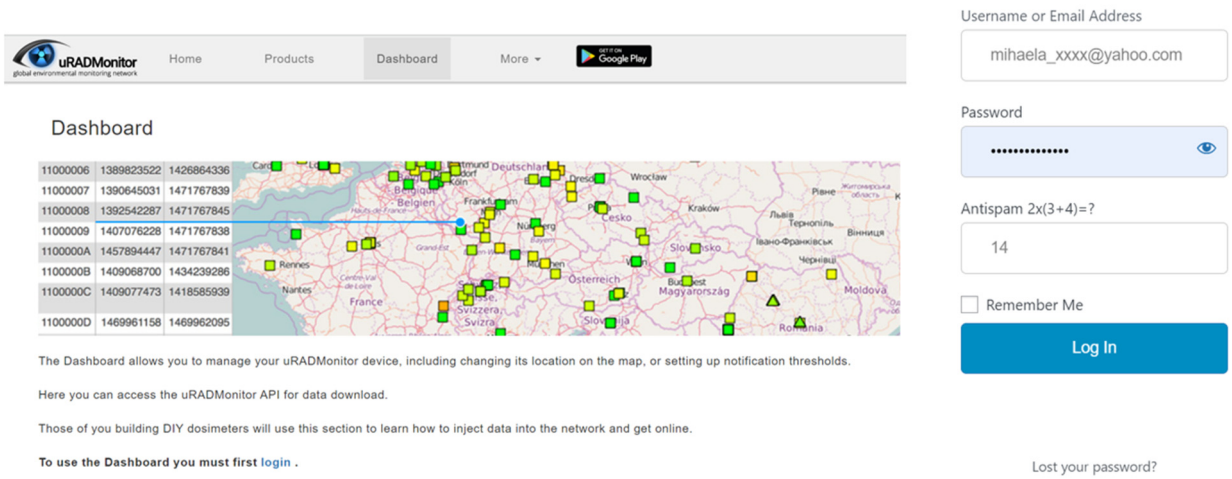


Figure 8.5. *User's login*

After authentication, the user can see in the Dashboard the IDs of the sensors owned by the user, their GPS location, the status (online or offline), the city, and the country where each sensor is located.

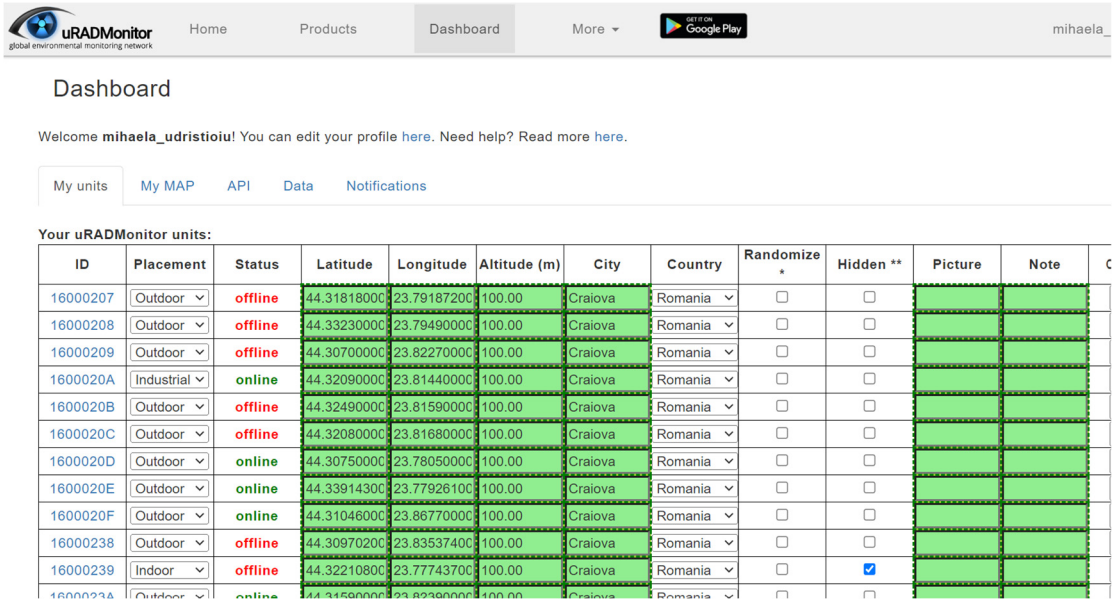


Figure 8.6. *How user's sensors appear in the Dashboard*

By clicking the My Map button, the user can see their Sensors on the map, as in the following image.

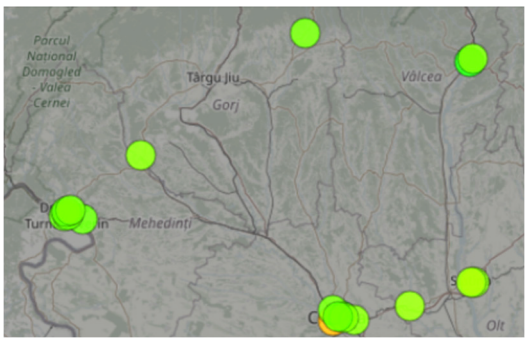


Figure 8.7. *User's map*

The API button provides information about how the user is identified.

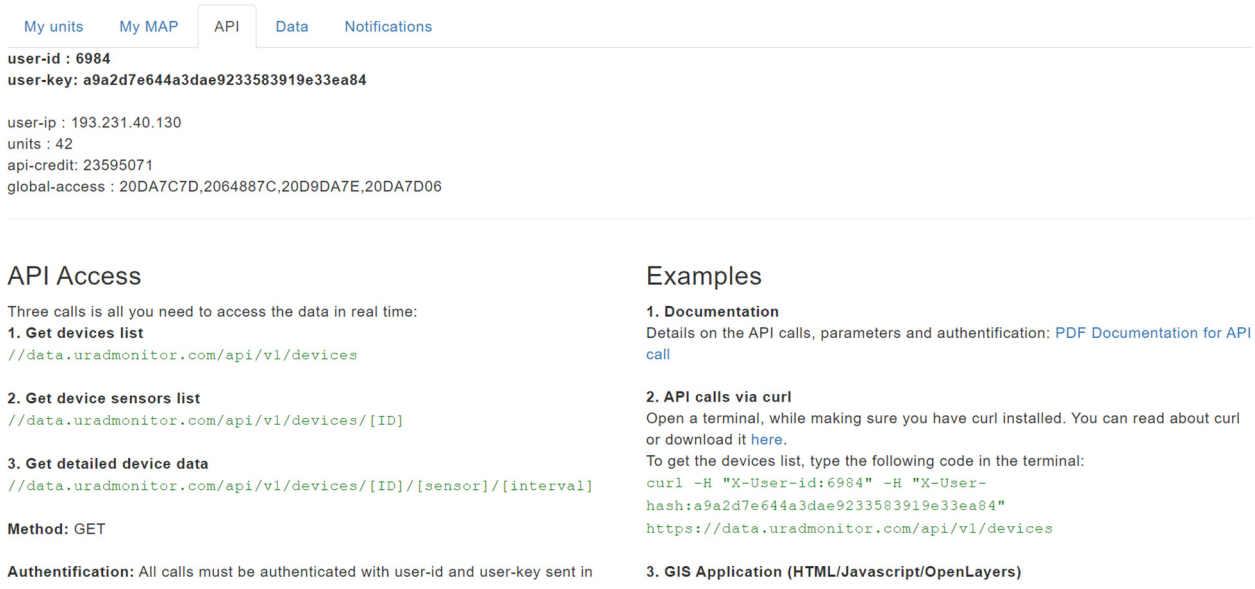


Figure 8.8. *API Information*

Clicking the Data button provides a tutorial on exporting data in JSON/CSV format. Also, information related to obtaining the real-time where a particular measurement was made can also be obtained. Next, select the time interval you want to analyze.

Figure 8.9*. Print screen of how the tutorial on getting real-time in the dataset appears*

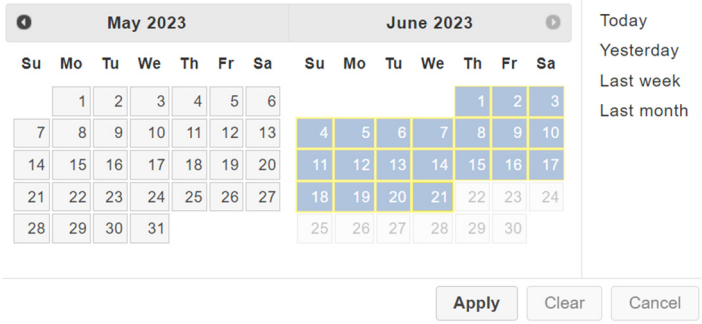


Figure 8.10*. Selecting the time range for which the user will download data*

To download data from a specific sensor in the Dashboard, press the Data button, select a specific time interval, then go to the sensor ID and select the parameter you want information about. It is downloaded parameter by parameter for a maximum of two months.

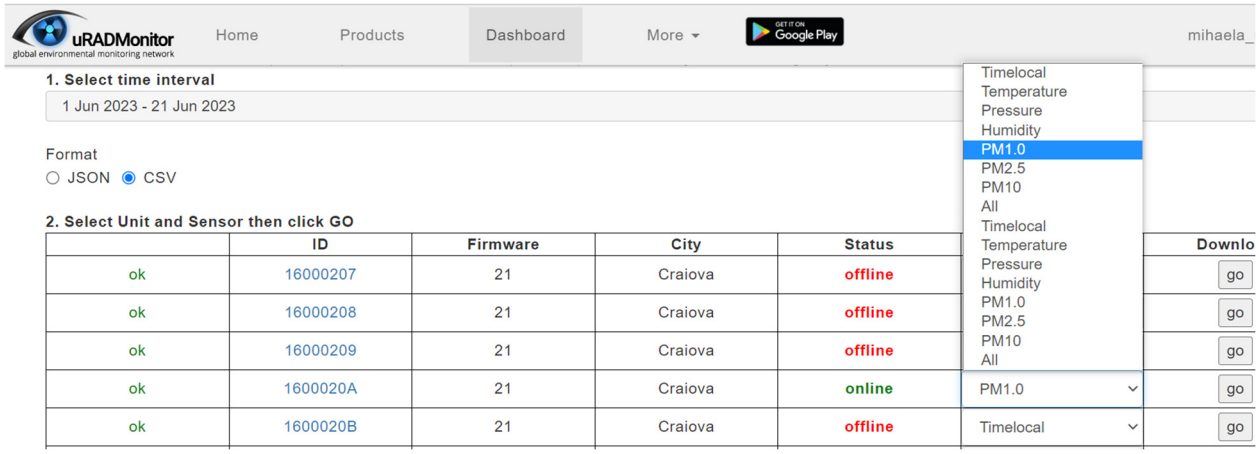


***Figure 8.11**. Selection of the parameter to be downloaded for a time interval*

After clicking on the "go" button, the user downloads the measurements acquired in the chosen time interval. You can view the loading process and its completion (the number of lines with information about the selected parameter).

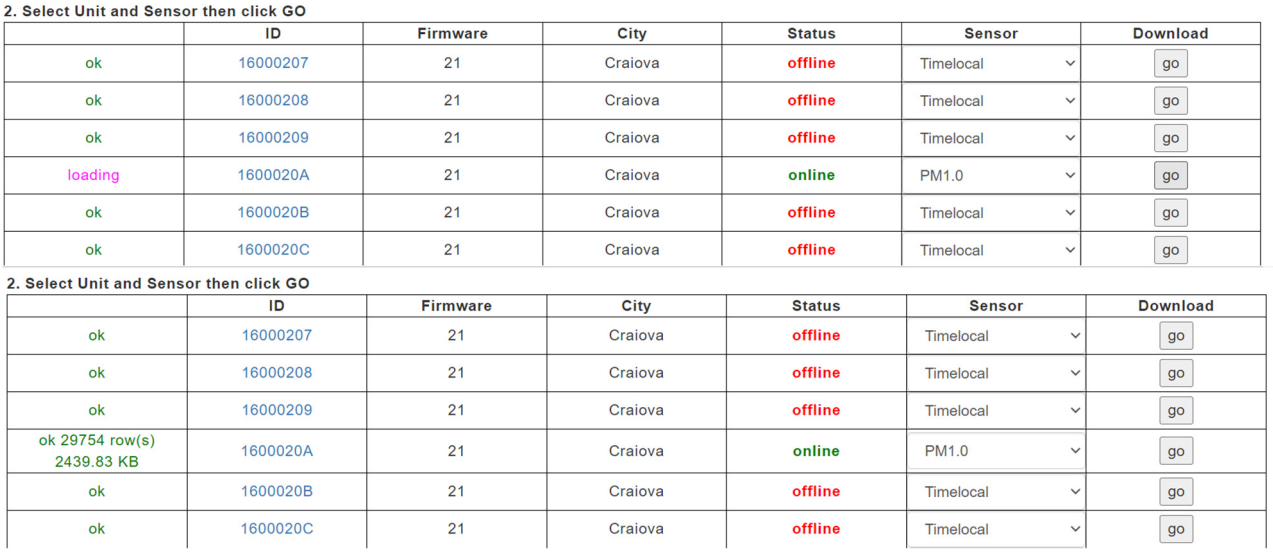


***Figure 8.12**. Download measurements of a specific parameter*

The data in .csv format appears in the following image. For the space-time evaluation of the data, insert the "real-time" column in that file and use one of the following formulas in the Excel file, adjusted according to your document

=A2/(60*60*24)+"1/1/1970"

Or

=DATE(1970,1,1)+A2/86400.

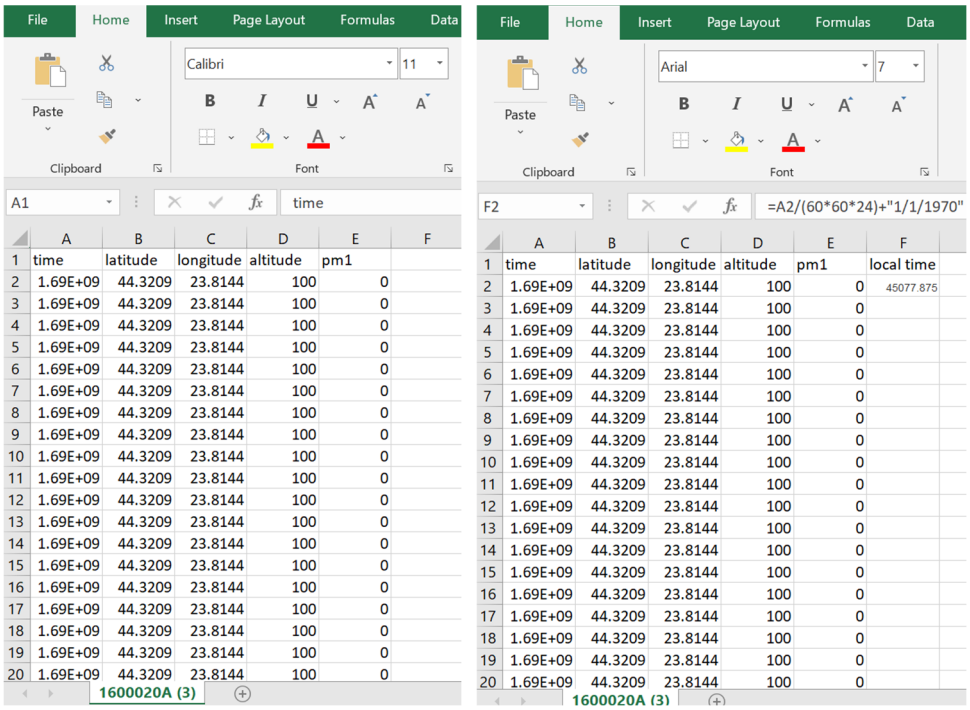


***Figure 8.13**. Insertion of the "real-time" column in the downloaded document*

To include the date and time of each measurement by the user, it is necessary to indicate the format of the cells, usually choosing a format of the type dd/mm/yyyy hh:mm. Measurements are made by the sensor at every minute, so the dataset obtained is vast.

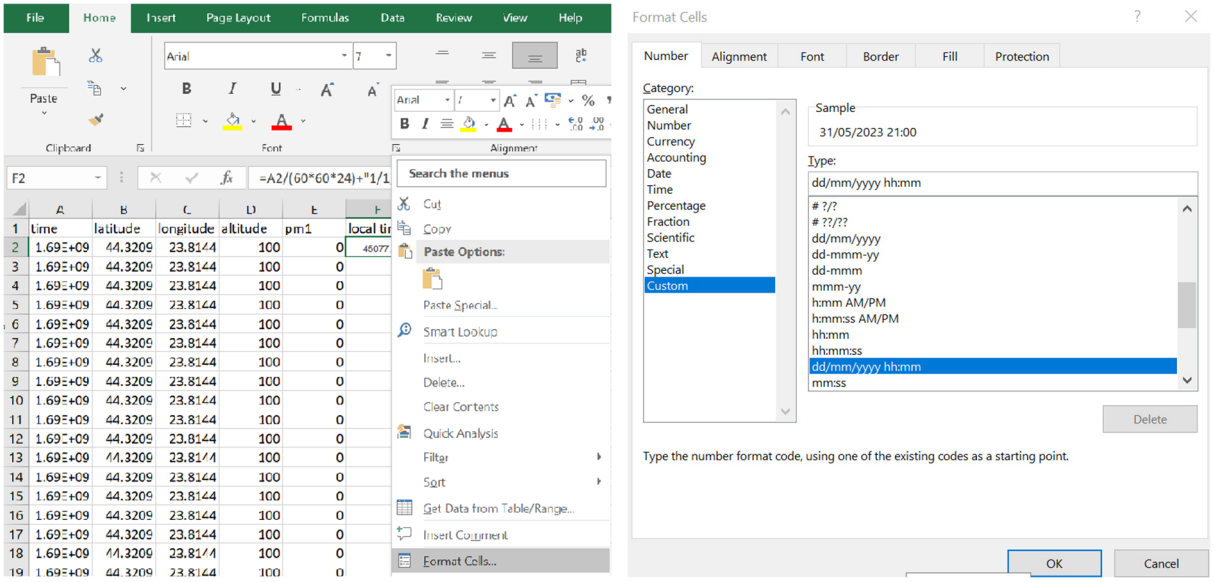


***Figure 8.14**. Choosing the cell format*

Double-clicking on the corner of the first cell under "local time," the time parameter will appear in the entire column.

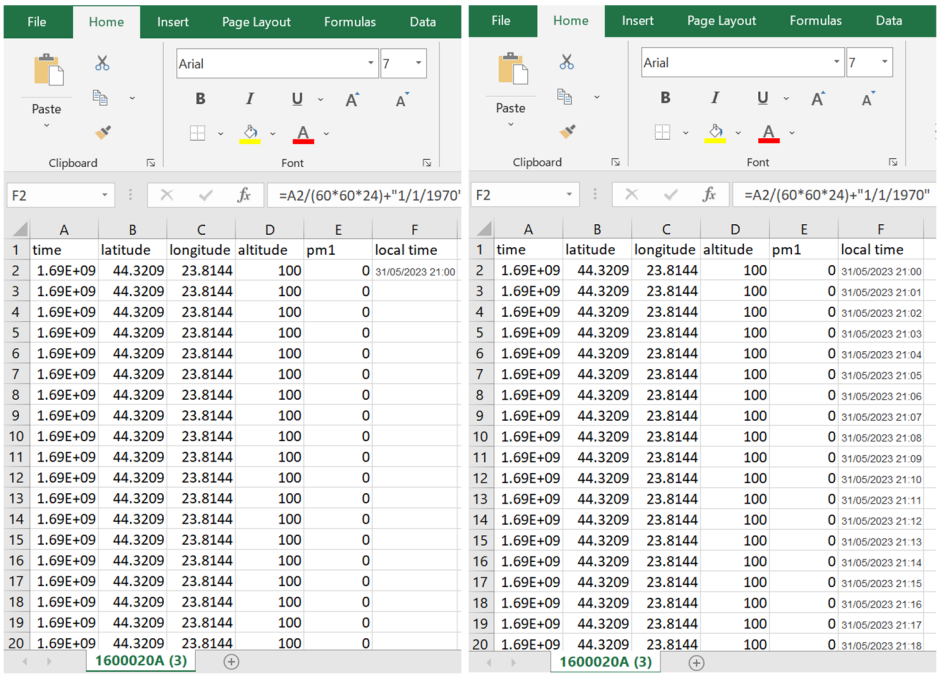


***Figure 8.15**. Local time indication for the entire set of measurements*

8.3.2. Data organization

Data related to each parameter can be downloaded for two months. The download continues parameter by parameter. Before starting data processing, a matrix is created with all downloaded parameters (copying and pasting, column by column) for the considered time interval.

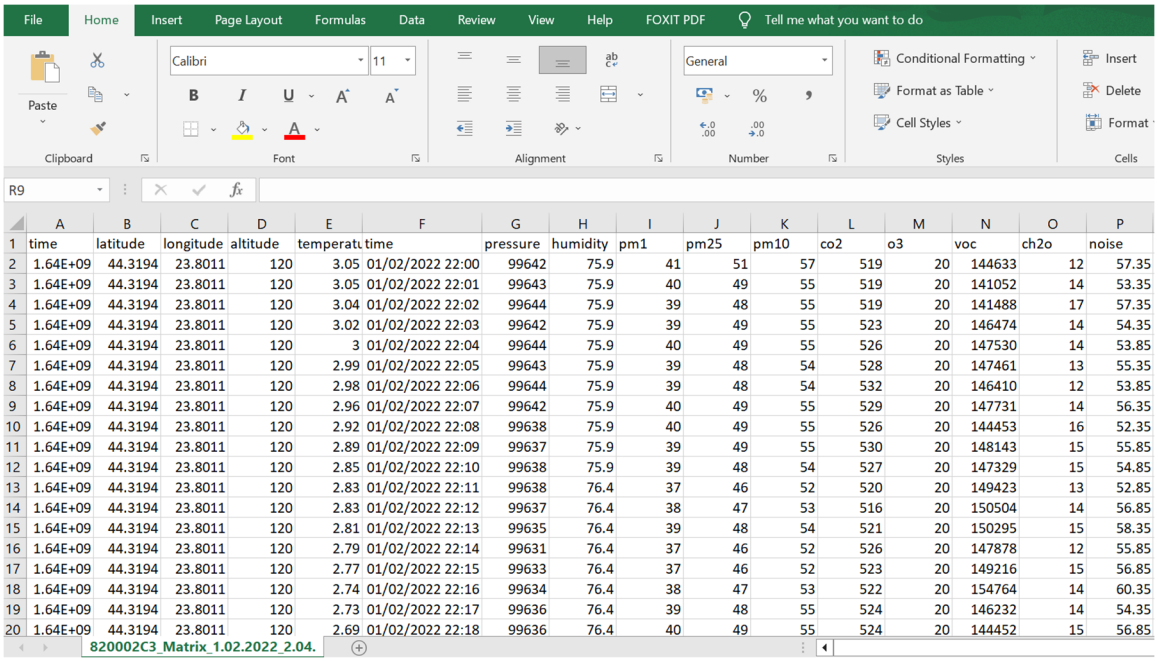


***Figure 8.16**. The matrix of data obtained from a sensor for a particular time interval*

References

www.uradmonitor.com (accessed August 10, 2023)
www.clearairoltenia.ro (accessed August 10, 2023)

CHAPTER 9. PROCESSING AND CORRELATIONAL ANALYSIS OF SENSOR DATA ACQUISITION

This chapter was written by Ion Buligiu from the University of Craiova, Romania

9.1. Theory

9.1.1. Import Data from CSV Files into Excel

Temperature, pressure, humidity, and particulate matter concentration sensors in the air provide data in CSV (Comma Separated Values) files. The analysis of these data can be performed through the Excel calculation processor to determine the average values, establishing correlations between parameter variations, as well as the synthesis and graphical interpretation of the data series collected from sensors.

The structure fields of the data tables will be as follows:

- time reference (serial number),
- latitude,
- longitude,
- altitude,
- temperature (degrees Celsius), humidity (hPa), relative pressure (%), PM 1, 2,5, and 10 μm (μg/m3),
- local time.

	A	B	C	D	E	F
1	time	latitude	longitude	altitude	temperature	local time
2	1652389214	44.3209	23.8144	100	21.16	12/05/2022 21:00
3	1652389274	44.3209	23.8144	100	21.13	12/05/2022 21:01
4	1652389334	44.3209	23.8144	100	21.08	12/05/2022 21:02

***Figure 9.1**. Data structure provided by sensors*

The data must be imported into Excel workbooks for proper analysis and interpretation.

Excel uses an import tool from the **Data** tab, **Get & Transform Data** command group, **From Text/CSV** button.

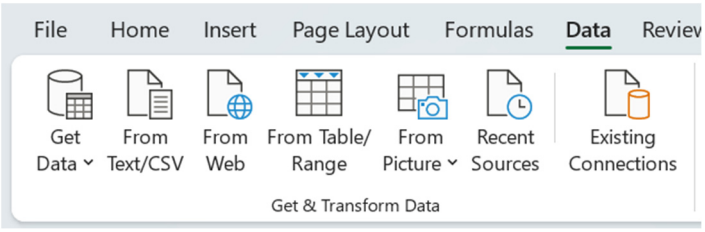


***Figure 9.2**. Group of commands for importing data into Microsoft Excel*

The users navigate to the CSV file if they want to import and select it.

In the popping-up dialog box (fig. 9.3.), select the correct separator (comma for CSV files) and preview the data.

Press the "Import" button to complete the import process if all settings are correct.

After completing these steps, the data from the CSV file will be imported into Excel and displayed in a table. You can save this table as an Excel file or perform any other operations or analysis you want with the imported data.

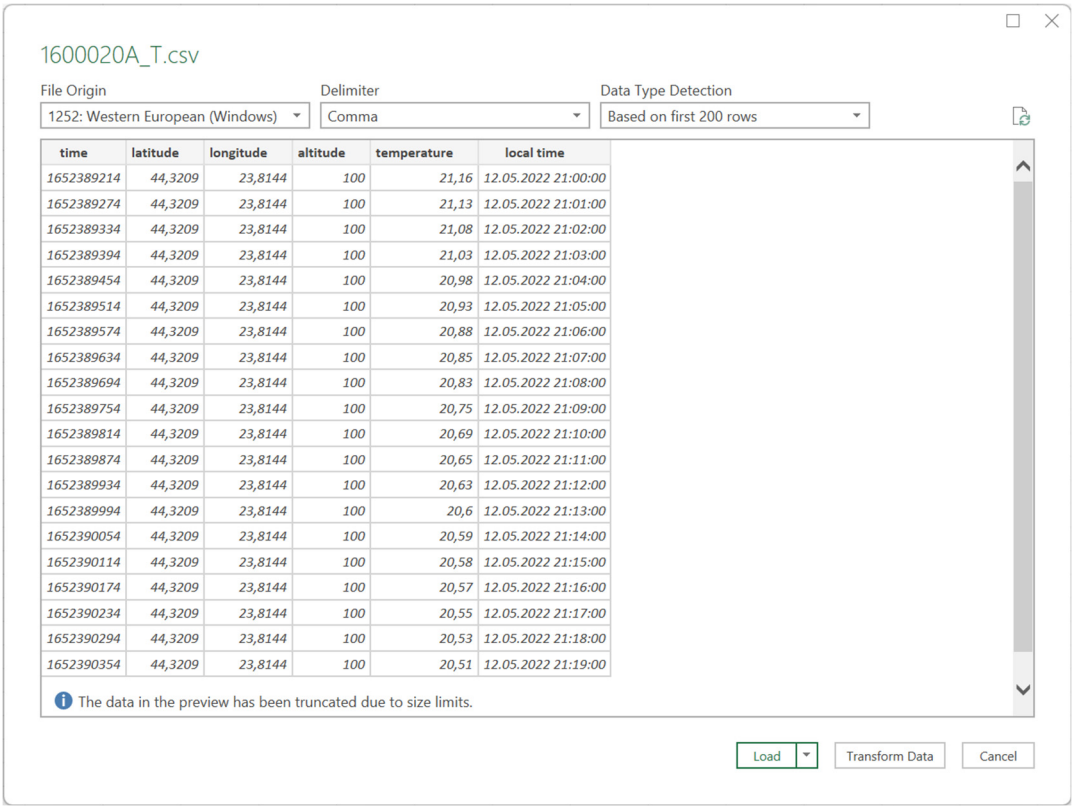


***Figure 9.3**. Import data from the CSV file*

9.1.2. Data Centralization

Starting from separate structural data files, we need to centralize all temperature, pressure, humidity, and particulate matter concentration sensor data in the air into a single data table.

The procedure is simple, addressing external data from each parameter in the centralized data table.

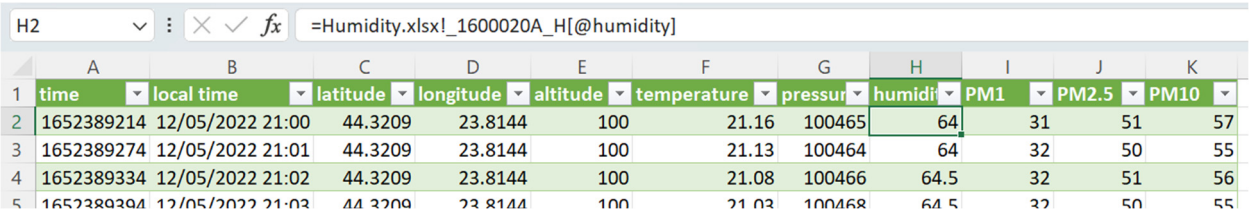


***Figure 9.4**. Using external addresses to centralize data*

Using external addresses in the cells of the summary spreadsheet is more convenient than copy-paste due to the size of the data table, and more importantly, the centralized data will be linked to the data sources. So, any change in data sources will be updated automatically by Excel with the imperative condition that workbooks and data sources are opened when you access the centralized data table.

9.1.3. Summarize Data with the PivotTable Tool

The next step is to synthesize all data sensors because we have many records (tens of thousands of data lines). Each sensor reports parameter values every minute, so we need to calculate an average daily value for a more straightforward data evaluation.

The right tool that synthesizes our data is the **Pivot Table**, which can be found in the **Insert tab** in the Tables group of commands.

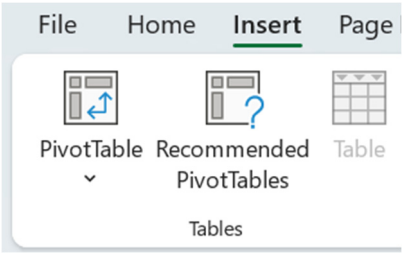


***Figure 9.5**. PivotTable Synthesis Tool*

It's vital that before inserting the pivot table into the workbook, you must select a cell within the data series (any cell in the data table)!

The first step to creating a new pivot table is to specify the table's data source (our case) or the scope from which the data is extracted, using absolute addressing.

You can choose where to place the Pivot Table, as a **New Worksheet** or in an existing worksheet in your workbook (Fig. 9.6.).

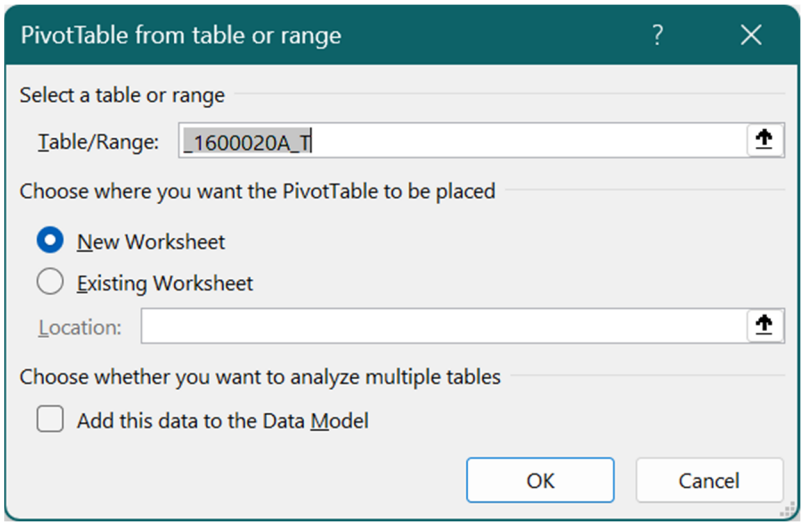


***Figure 9.6**. Selecting the data source to pivot*

In the PivotTable design interface, the report fields for our analysis should be specified as follows (Fig. 9.7):

- Filters: months (local time)
- Rows: days (from local time)
- Values: fields of all parameters (temperature, pressure, humidity, particulate matter 1, 2.5, and 10 µm).

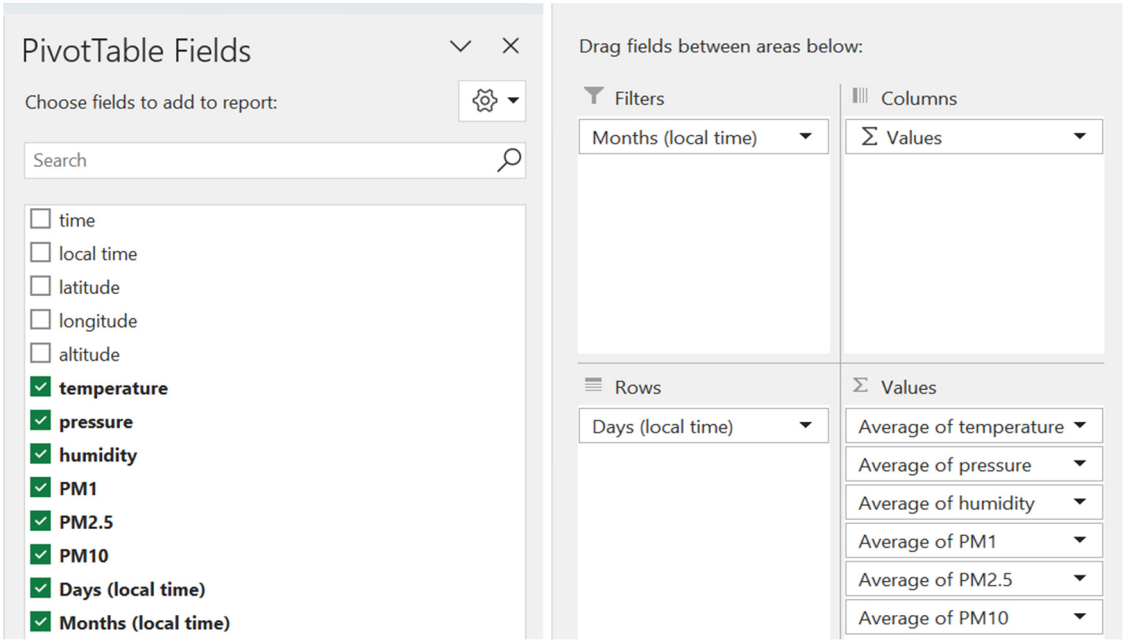


***Figure 9.7**. Select fields for the pivot table*

With these settings, we can synthesize our data rows at the detail level to days (we can always change the level of detail to months).

For each **Values** field parameter, we need to specify additional settings (Fig. 9.8), such as changing the *Summarize value field by* setting it to *Average* to get an average value for each day.

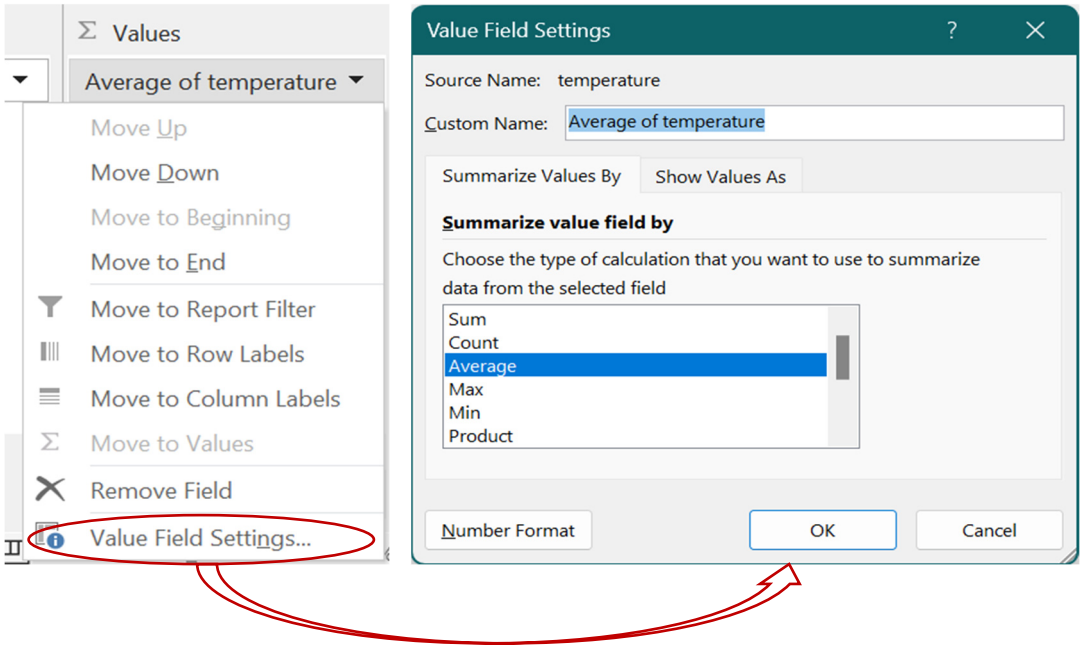


***Figure 9.8**. Set the arithmetic mean calculation for pivot table values*

In the pivot table report (Fig. 9.9), we can observe in the first column the time reference (per day) and then the average values of each parameter (temperature, pressure, humidity, and PM).

In the first line of the report, we can filter data series by month or months one by one or select multiple values from the list of available months.

	A	B	C	D	E	F	G
1	Months (local time)	(All)					
2							
3	**Row Labels**	**Average of temperature**	**Average of pressure**	**Average of humidity**	**Average of PM1**	**Average of PM2.5**	**Average of PM10**
4	12-May	18.99	100447.20	67.67	12.72	17.99	20.75
5	13-May	23.79	100229.11	62.77	6.61	7.85	8.54
6	14-May	23.90	100031.60	64.82	5.45	6.39	6.86
7	15-May	21.43	100465.52	67.70	5.84	6.64	7.01
8	16-May	22.59	100433.05	64.03	8.56	10.45	11.18
9	17-May	22.29	100111.09	68.04	5.84	6.70	7.20
10	18-May	15.80	100713.73	75.01	2.04	2.42	2.71
11	19-May	17.22	101366.24	66.87	3.53	4.83	5.41
12	20-May	20.76	100980.39	61.70	4.23	6.09	7.03
13	21-May	23.28	100085.90	62.14	2.70	3.11	3.50
14	22-May	21.94	99671.31	59.90	1.31	1.57	1.90

***Figure 9.9**. PivotTable Report*

9.1.4. Plot Chart Generation and Data Flow Interpretation

The data series generated in the Pivot Table report are already synthesized. We have reduced many data records into a time series with average values calculated for each sensor parameter. Still, it is difficult to observe and compare variations in numerical values.

The solution is to plot the data series to understand each parameter's evolution over time better, and we can approximate a primary correlation between our data.

We choose a type of plot by lines or lines with markers, and then we use a chart combined with a secondary axis for pressure because the values are much too large for the representation on the same axis but the temperature.

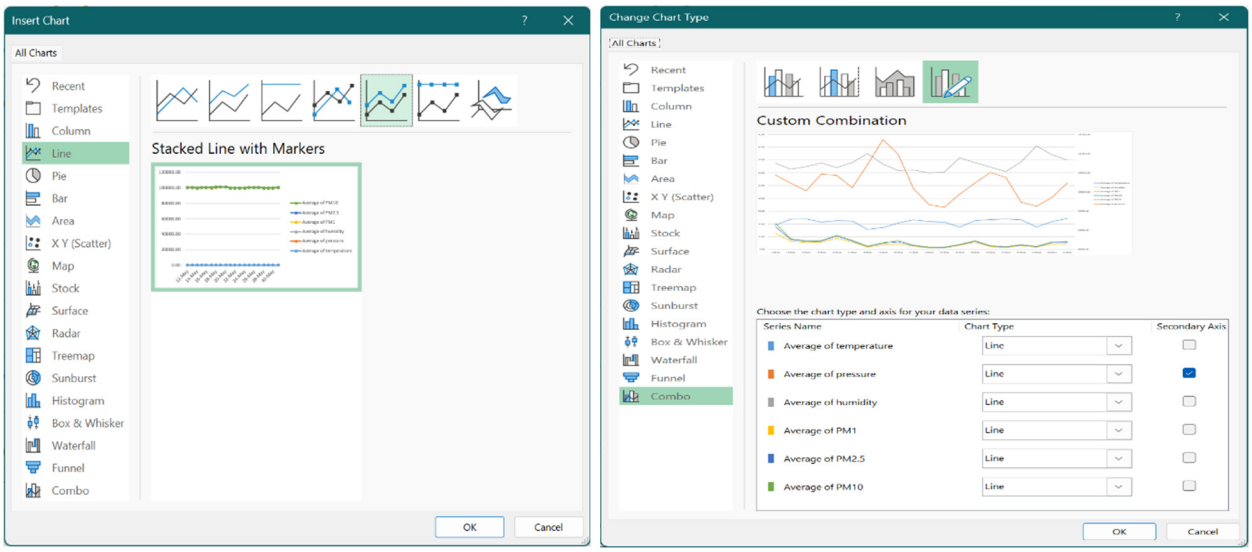


***Figure 9.10**. Choose chart type for correlation analysis*

We will thus obtain a diagram to perform the correlational analysis of the tracked parameters.

From the diagram's study, a synchronous evolution of data on quantities of particulate matter with dimensions of 1, 2.5 and 10 µm can be seen. At the same time, average correlations between temperature and the set of particulate matter indicators can be distinguished, as well as an average inverse correlation between temperature and humidity and a weak inverse correlation between temperature and pressure.

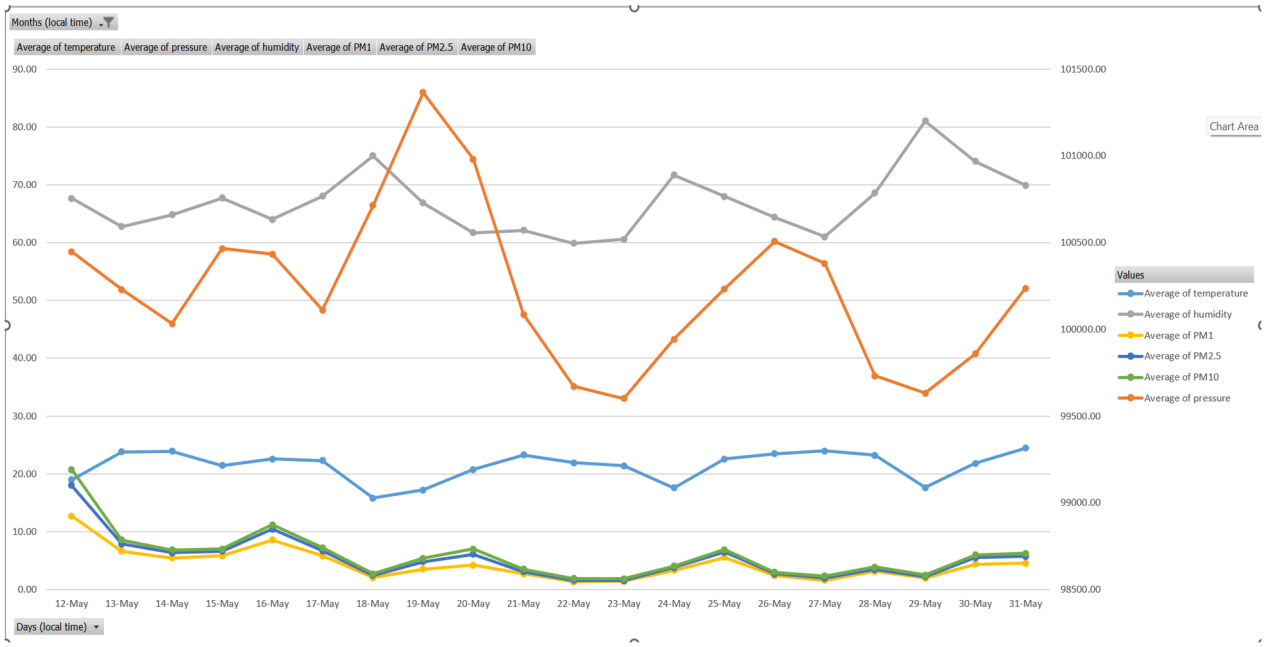


***Figure 9.11**. Level of correlation for each parameter*

However, the comparative analysis by studying representation diagrams is approximate, so it is necessary to identify exactly the level of correlation and its type by calculating the correlation coefficients of the tracked parameters.

9.2. Correlation Analysis

9.2.1. Calculation of Correlation Coefficients Using Analysis Toolpak

The correlation coefficient tells us how strongly two related variables are and can take values ranging from -1 to +1.

Microsoft Excel makes the CORREL function or Analysis Toolpak add-in available to users to calculate the correlation coefficient between two variables.

A *correlation coefficient with a +1 value* indicates a **perfect positive correlation** (Fig. 9.12). As variable X increases, variable Y increases. As variable X decreases, variable Y decreases.

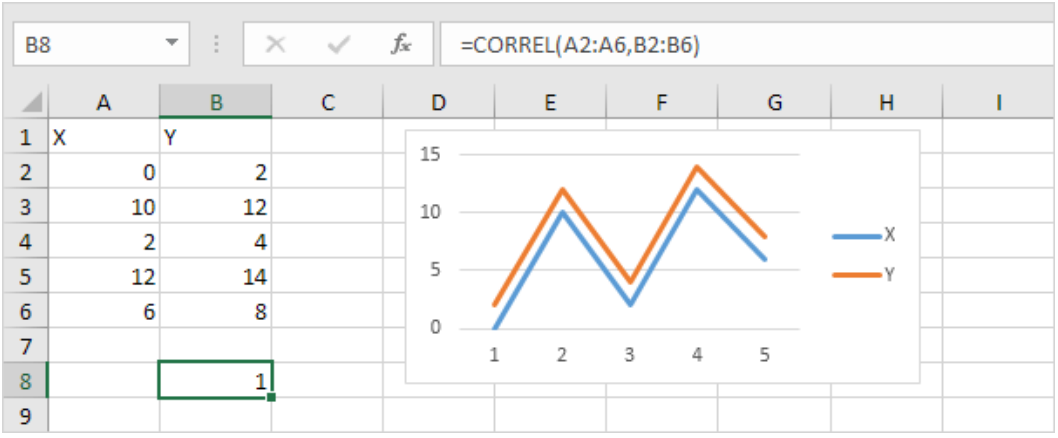


***Figure 9.12**. Perfect positive correlation between variables X and Y*

A *correlation coefficient with the value -1* value indicates a **perfect negative correlation** (Fig. 9.13). As variable X increases, variable Z decreases. As variable X decreases, variable Z increases.

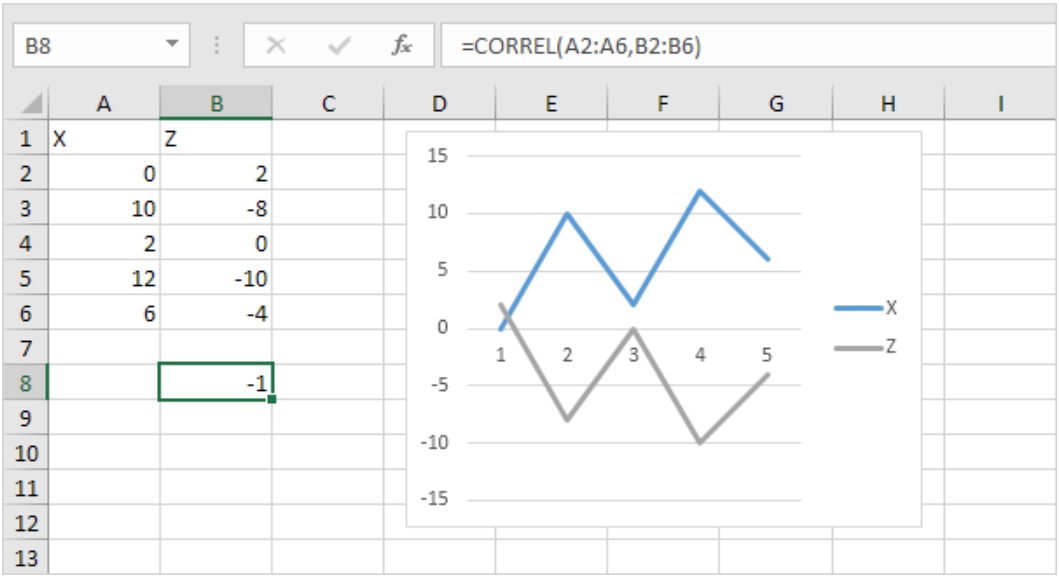
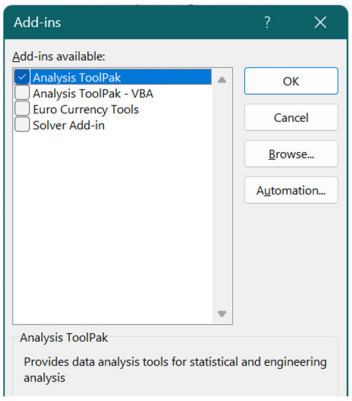


***Figure 9.13**. Perfect negative correlation between variables X and Y*

A correlation coefficient close to 0 indicates no correlation.

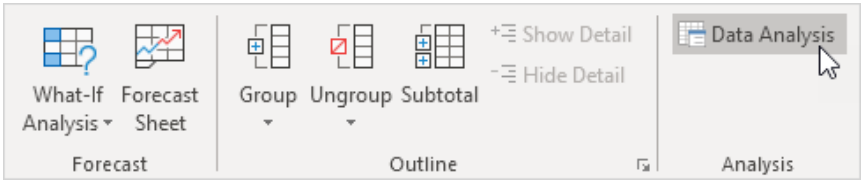


The Analysis Toolpak data analysis tool allows quick calculation of correlation coefficients between several variables.

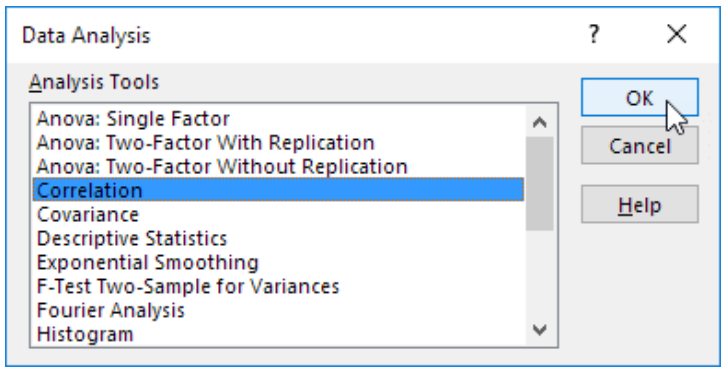
Analysis ToolPak can be enabled in Excel with the following sequences of commands: **File** – **Options** – **Add-Ins**, then press the button **Go...**, and from the list of tools, enable **Analysis Toolpak**, then tap **OK.**

The steps to be taken are as follows:

1. On the **Data**, commands group **Analysis**, press **Data Analysis** button**:**



2. In the dialog that appears, choose from the list of tools **Correlation** and press **OK:**



3. In this example, select range A1:C6 as the input range.
4. Enable the validation box **Labels in the first row**.
5. Select cell A8 as the representation location (**Output range**).
6. Click OK.

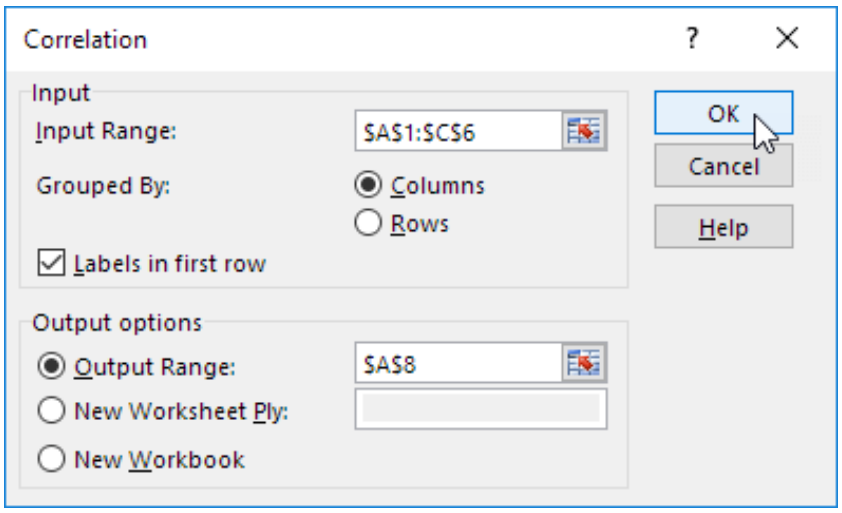


***Figure 9.14**. Setting parameters for determining correlations*

From the results obtained (Fig. 9.15), we can see that variables A and C are positively correlated (0.91); variables A and B are poorly correlated (0.19). Variables B and C also have a low correlation (0.11). You can also verify these conclusions by studying the graph.

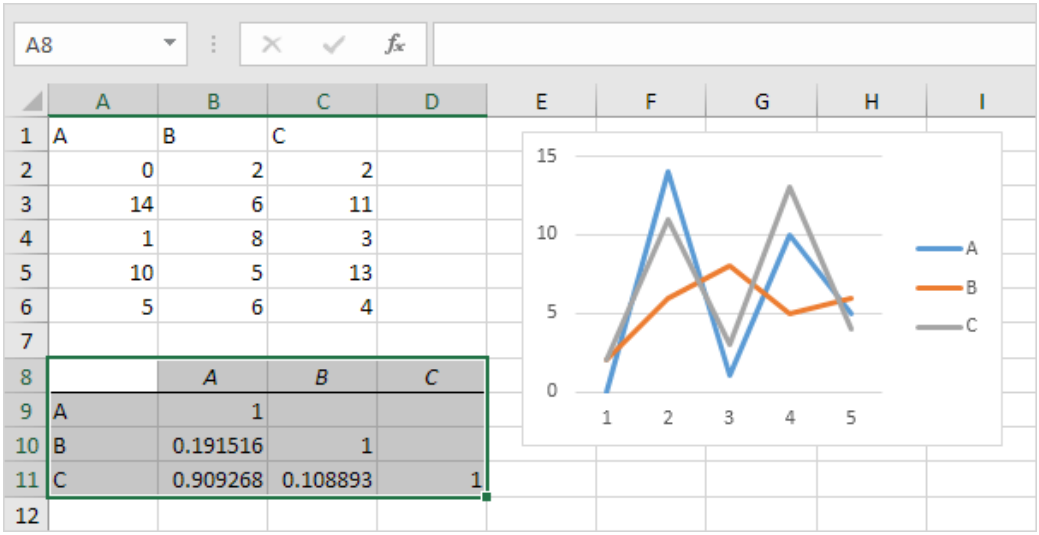


***Figure 9.15**. Analysis of correlation coefficients between variables A, B and C.*

9.2.2. Correlational analysis between CO_2, noise, CH_2O and O_3 parameters

In the following, we will study the correlational analysis for CO_2, noise, CH_2O, and O_3 parameters, following the steps described above, importing data from CSV files, centralizing data, and synthesizing them. The results obtained for these parameters are in the following table.

Table 1. *Correlational table between CO_2, noise, CH_2O and O_3 parameters*

	Average of co2	*Average of noise*	*Average of ch2o*	*Average of o3*
Average of co2	1			
Average of noise	-0,602455913	1		
Average of ch2o	0,213288816	-0,149072181	1	
Average of o3	-0,792741937	0,475328544	-0,127393812	1

A negative correlation (-0.602455913) between CO_2 and noise is observed in the study data range, a weak positive correlation (0.213288816) between CO_2 and CH_2O, and a better negative correlation between CO_2 and O_3 (-0.792741937).

At the same time, noise is a weak negative correlation (-0.149072181) with the CH_2O parameter and a mean positive correlation (0.475328544) between noise and O_3.

The CH_2O parameter is in weak negative correlation (-0.127393812) with parameter O_3.

Correlation diagrams between CO_2, noise, CH_2O and O_3 parameters:

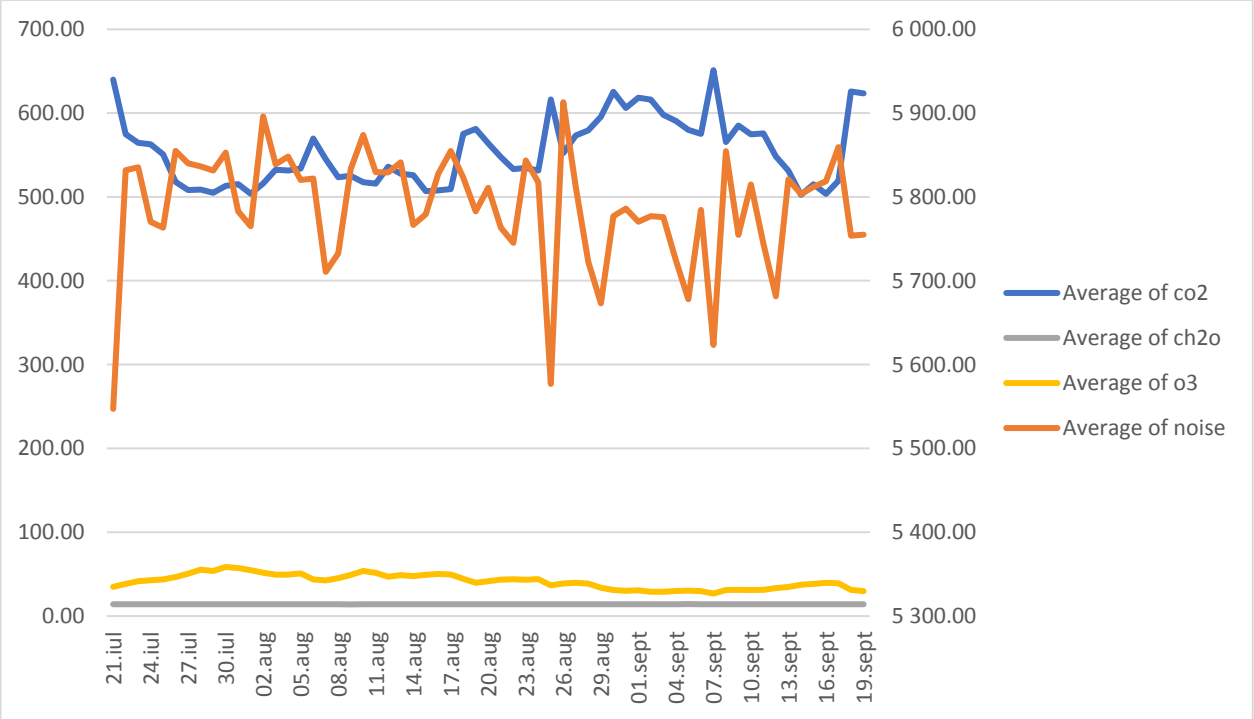


Figure 9.16*. Correlational graph CO_2 – Noise*

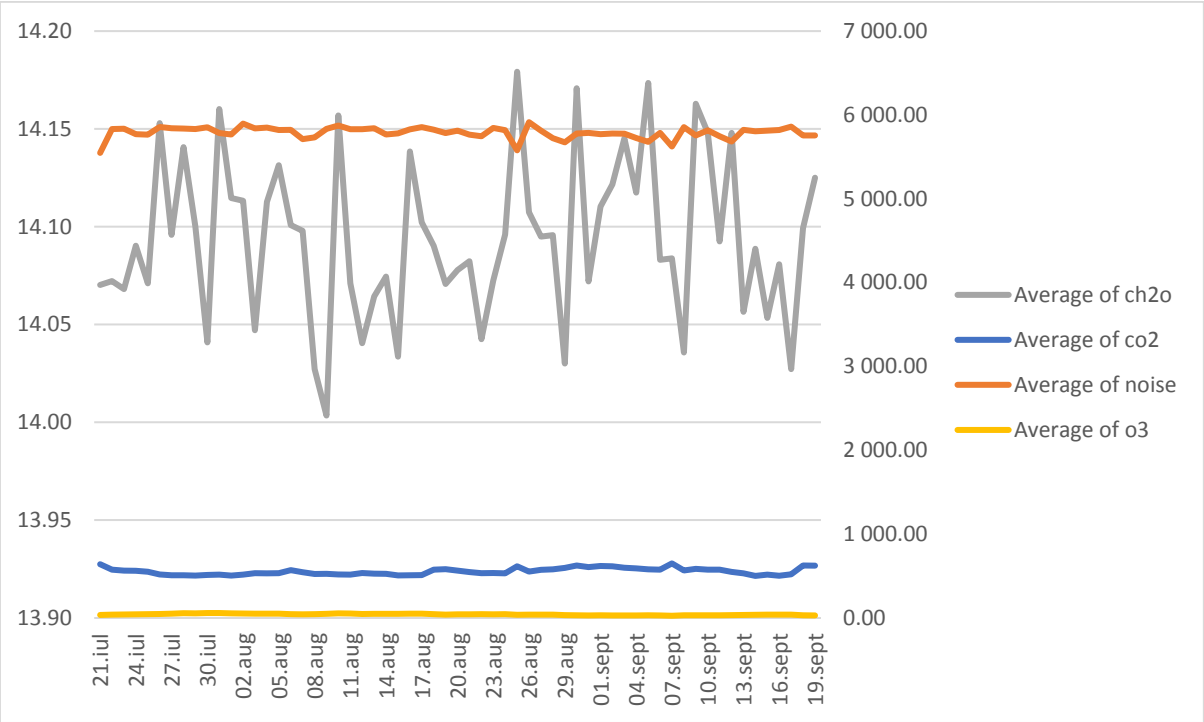


***Figure 9.17**. Correlational graph Noise – CH_2O*

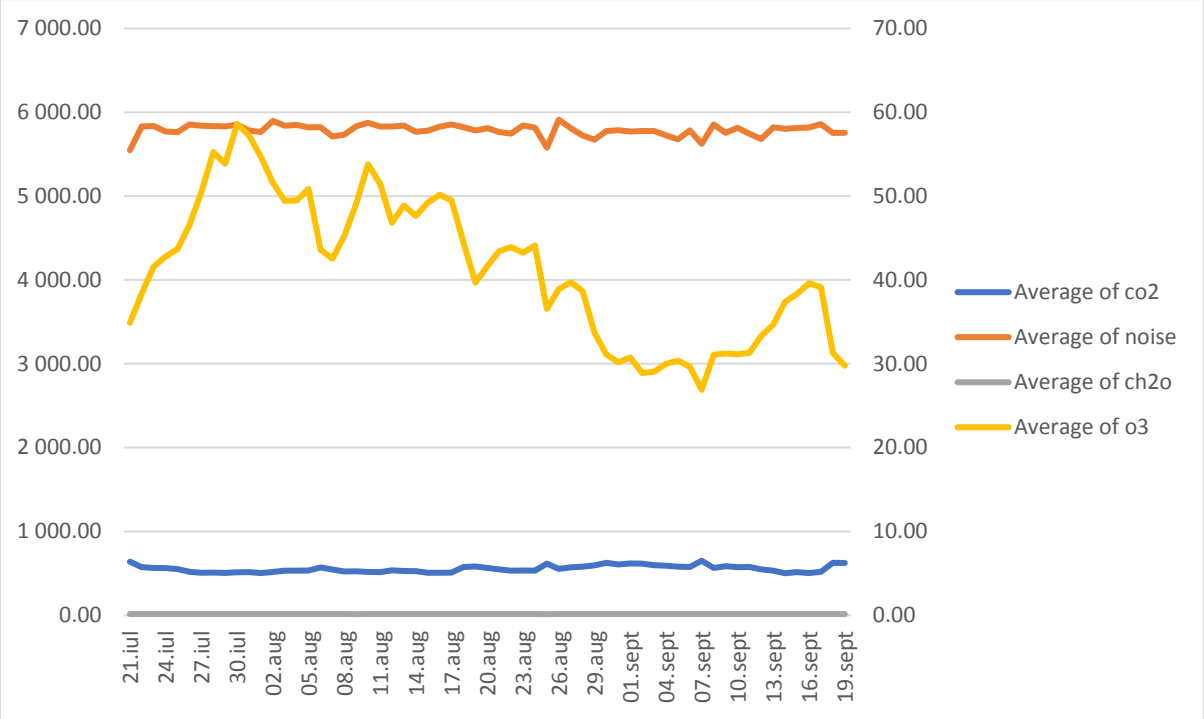


***Figure 9.18**. Correlational graph Noise – O_3*

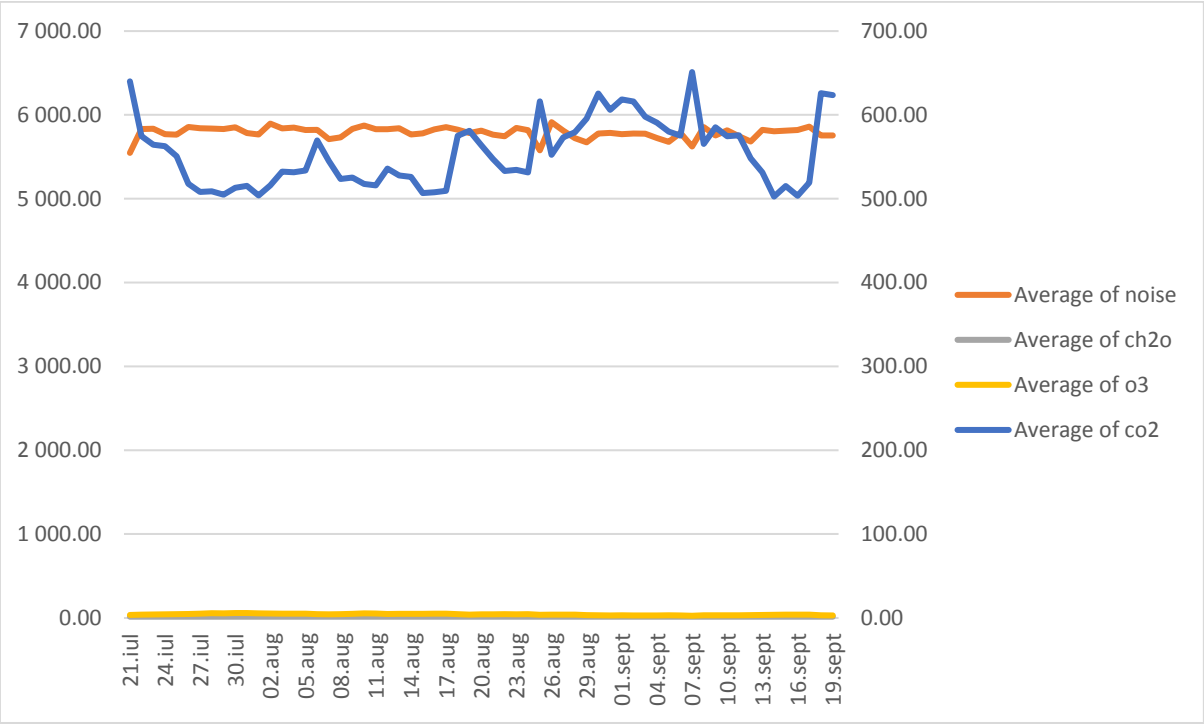


***Figure 9.19**. Correlational graph Noise – CO_2*

Increasing the degree of granularity of the data, we obtain the following correlational diagrams between CH_2O parameters and noise, between O_3 and noise, and between CO_2 and noise:

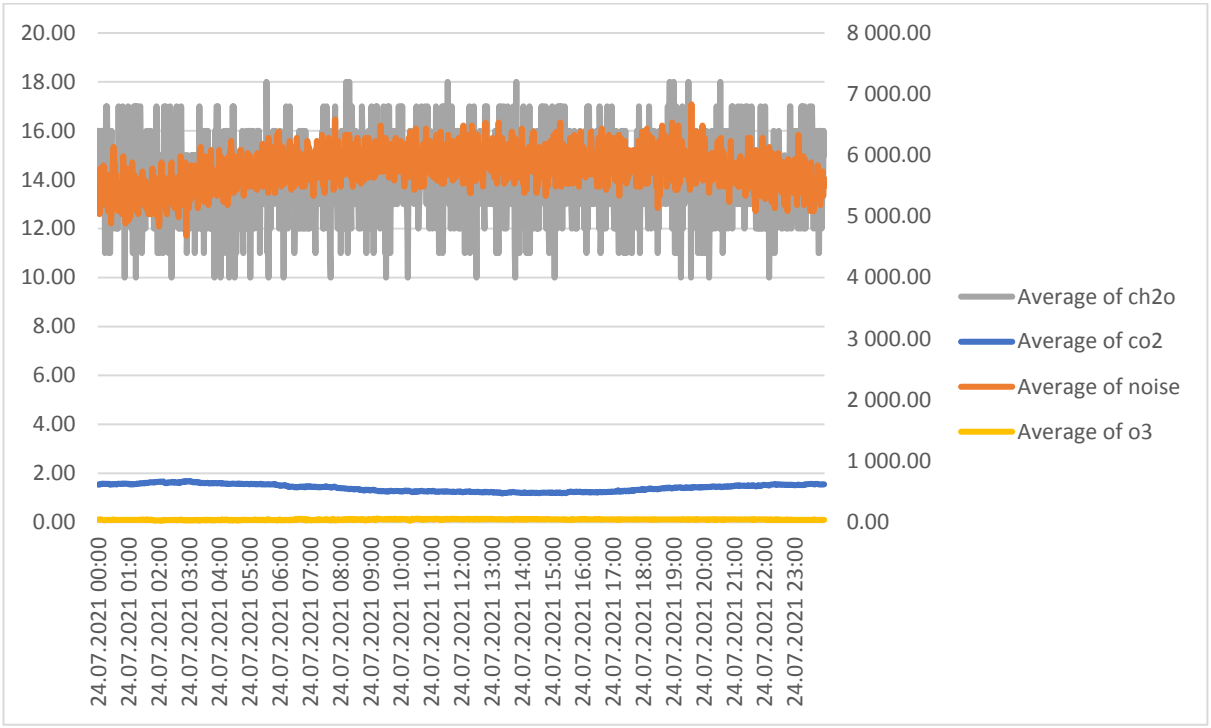


***Figure 9.20**. Correlational diagram of higher granularity CH_2O and noise*

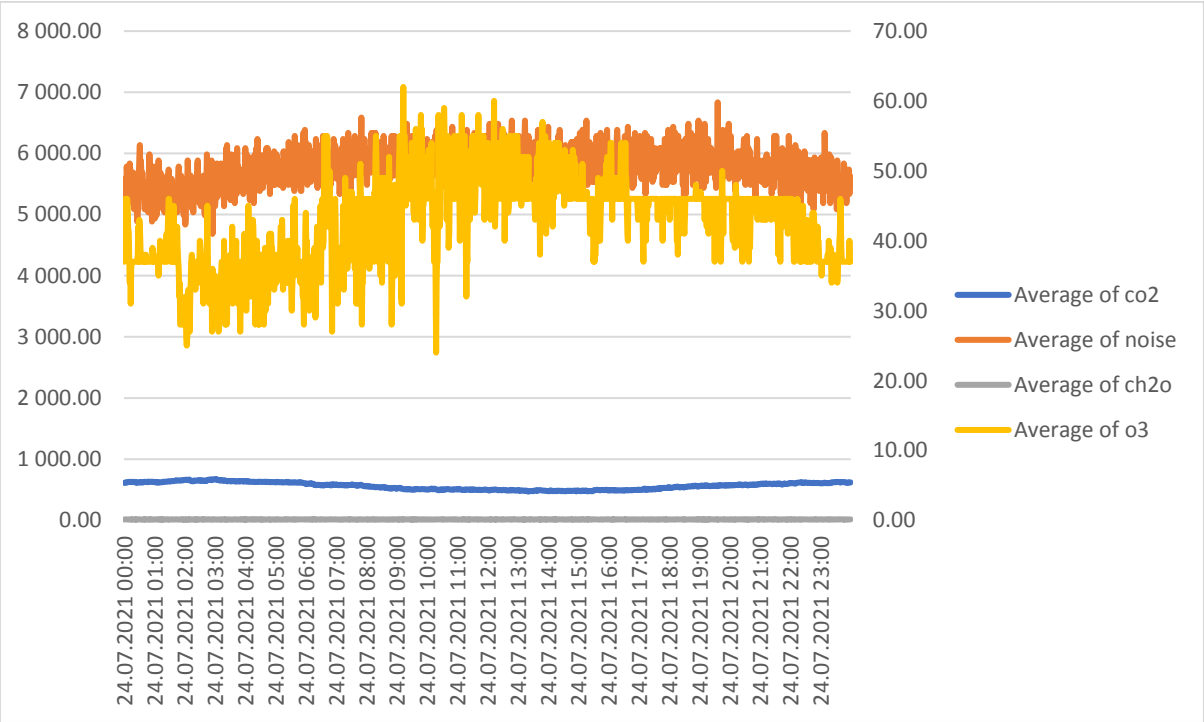


***Figure 9.21**. Correlational diagram of higher granularity O_3 and noise*

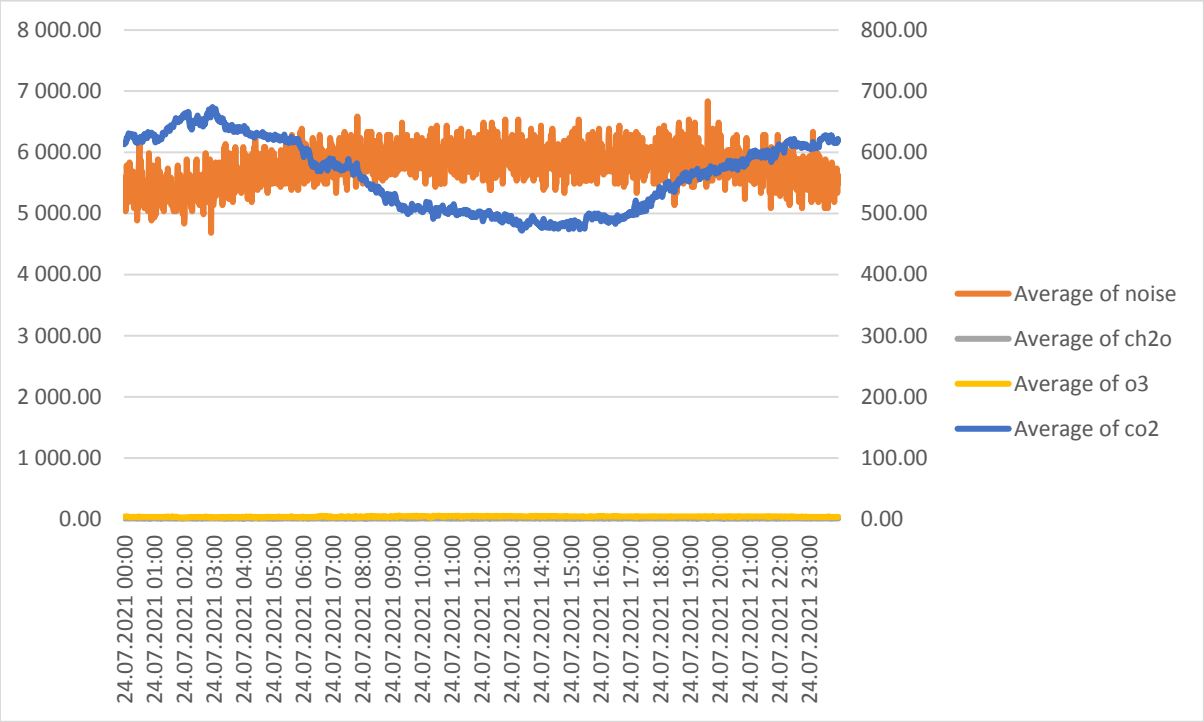
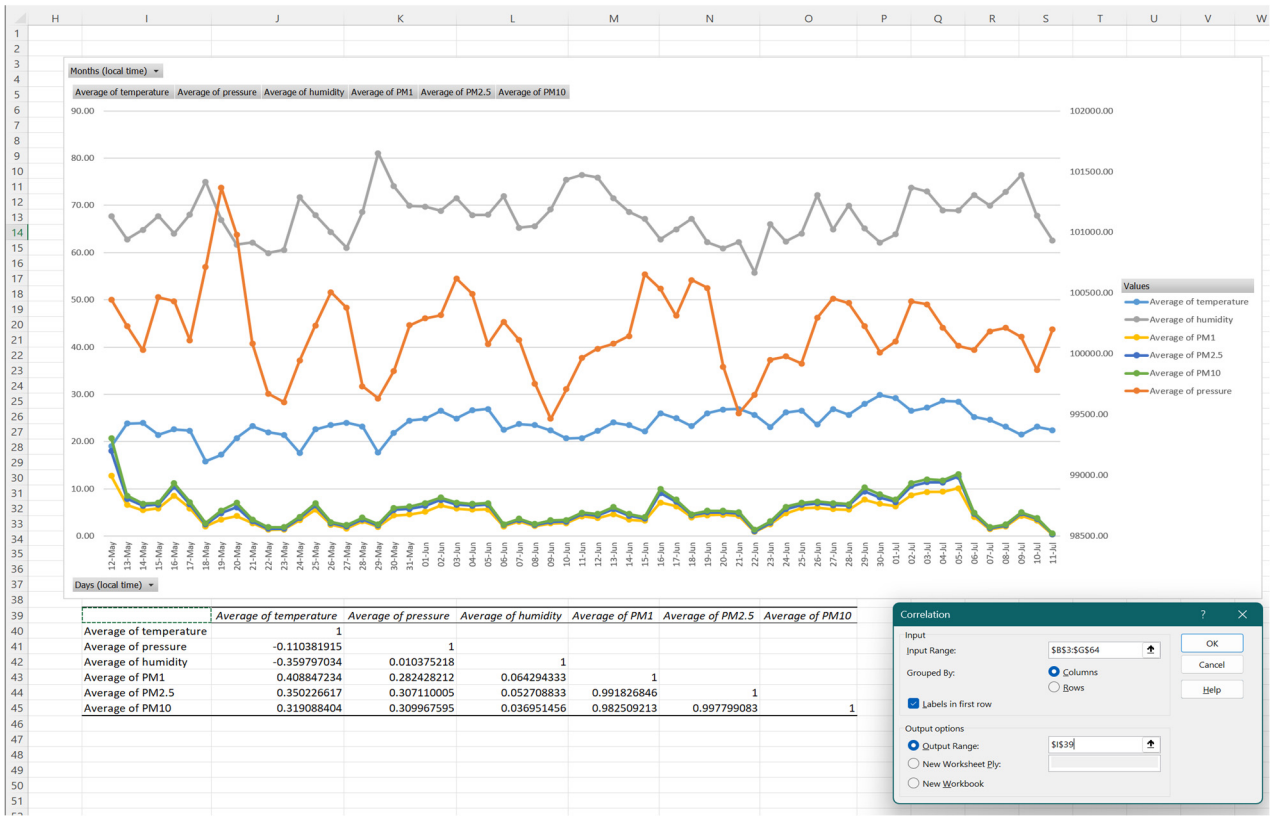


***Figure 9.22**. Correlational diagram of higher granularity CO_2 and noise*

9.3. Tasks

Using **the Analysis ToolPak** tool, realize a correlational analysis between temperature, pressure, humidity, and PM1, PM2.5, and PM10 levels.

The result should appear in the following form:



	Average of temperature	Average of pressure	Average of humidity	Average of PM1	Average of PM2.5	Average of PM10
Average of temperature	1					
Average of pressure	-0.110381915	1				
Average of humidity	-0.359797034	0.010375218	1			
Average of PM1	0.408847234	0.282428212	0.064294333	1		
Average of PM2.5	0.350226617	0.307110005	0.052708833	0.991826846	1	
Average of PM10	0.319088404	0.309967595	0.036951456	0.982509213	0.997799083	1

Figure 9.23. *Correlational analysis of temperature, pressure, humidity parameters, and particulate matter levels PM1, PM2.5 and PM10*

CHAPTER 10. A STATISTICAL ANALYSIS OF THE DATA GIVEN BY AN AIR QUALITY MONITORING STATION: ADANA CASE

This chapter was written by Tuğçe Pekdoğan
from the Adana Alparslan Türkeş Science and Technology University, Turkey

10.1. Theory

10.1.1. Outdoor air pollution description

Outdoor air pollution is a severe problem affecting the health of people around the world. The sources of outdoor air pollution are diverse. Natural sources of outdoor air pollution include dust, pollen, forest fires, etc. Non-natural sources of outdoor air pollution include emissions from power plants, factories, transportation, and construction activities.

Outdoor air pollution often severely affects ambient air quality in urban areas. High levels of pollutants such as particulate matter, nitrogen oxides, sulfur dioxide, and carbon monoxide can significantly affect human health. These pollutants can cause respiratory diseases, cardiovascular problems, disorders, and other health complications.

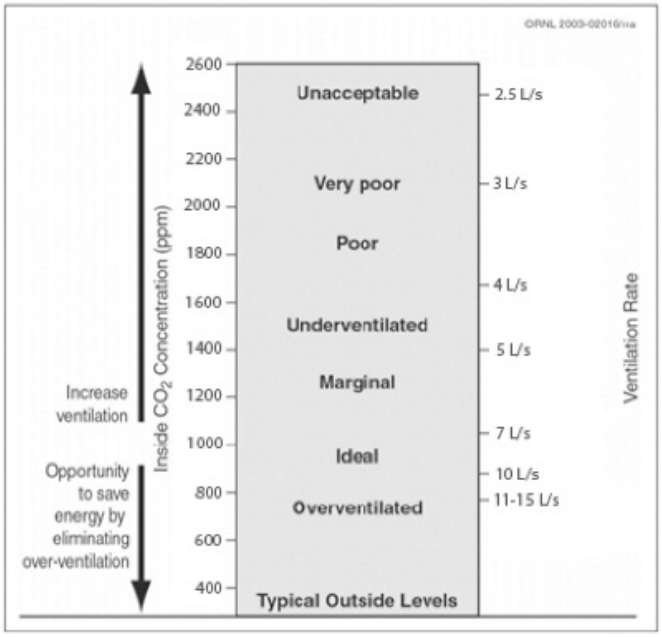


***Figure 10.1**. The relationship between CO_2 and ventilation rates (converted to SI unit) [4].*

The amount of CO_2 in the atmosphere is about 0.03%. Environmental factors estimate that the amount of CO_2 in outdoor air is between 300 and 500 ppm [1]. By inhaling and concentrating inward, people release CO_2. Human metabolic rate, influenced by size and physical activity level, determines how quickly CO_2 is produced in the body [2]. Although the quality of indoor air in buildings can be affected by outdoor air pollution, some other factors adversely affect indoor environments and the health of occupants of the building. Optimum carbon dioxide (CO_2) concentrations for indoor air quality vary from country to country, with a predicted level of 1000 ppm generally accepted. CO_2 sensing technologies are often used

with ventilation systems to maintain a healthy indoor environment. The relationship between CO_2 levels and ventilation is shown in Figure 10.1; here, it is believed that the application of demand control based on carbon dioxide can save energy by more accurately meeting the air intake requirements of buildings [3]. CO_2 concentrations ranging from 400 to 500 ppm are suitable for outdoor settings. A ventilation rate of about 7 L/s is sufficient to ensure a good fresh air flow, even if the CO_2 level outside is 700 ppm. According to the American Society of Heating, Refrigeration, and Air Conditioning Engineers, the recommended ventilation rate per person in ASHRAE Standard 62.1 is 7–10 L/sec [1].

The maximum CO_2 limits for contaminants related to indoor air quality have been set in various nations worldwide. The allowable CO_2 concentration limits set by ASHRAE and various international governments [5] and standards agencies are compared in Table 1.

Table 1. *CO_2 occupational exposure limits for select countries [3]*

Country	Agency/Standard	Maximum Level
USA	ASHRAE 62.1-2013 Appendix C	700 ppm above outside levels
	Occupational Safety and Health Administration (OSHA)	5,000 ppm
Australia	Safe Work Australia; Workplace Exposure Standards for Airborne Contaminants (2011)	5,000 ppm
Canada	Federal-Provincial Advisory Committee on Environmental and Occupational Health	3,500 ppm
Germany	DIN 1946-6 DIN 1946-2	1,000 ppm (recommended) and 1,500 ppm (upper limit)
Japan	Japan Society for Occupational Health (2004)	1,500 ppm
Sweden	Occupational Exposure Limit Values, AFS 2011:18	5,000 ppm
UK	Health and Safety Commission	5,000 ppm
	The UK. Building Bulletin 101	1,500 ppm

Particulate matter, or PM for simple terms, is the term for small particles in the air that come from various sources, including transportation, industry, and natural sources like dust and forest fires. PM consists of particles that are 10 microns in diameter or smaller (PM10), 2.5 microns in diameter or smaller (PM2.5), and 1 micron in diameter (PM1). Outdoor air PM concentrations are typically measured using a particle monitor device. This tool uses a filter that captures and weighs airborne particles, and the data is expressed as micrograms PM per cubic meter of air.

Maximum values for PM levels vary from country to country. Still, the World Health Organization recommends that PM2.5 levels not exceed 10 micrograms per cubic meter of air per year averaged and 25 micrograms per cubic meter of air averaged daily [6]. According to National Ambient Air Quality Standards (NAAQS) for PM, The United States Environmental Protection Agency sets an annual standard of 10–12 micrograms per cubic meter of air and a daily standard of 35 micrograms per cubic meter of air for PM2.5.

10.1.2. Outdoor Air Pollution Requirements

Several regulations and standards have been put forward by organizations to prevent outdoor air pollution. These requirements help protect public health and the environment. Overall, these transparent, publicly available standards and regulations can help ensure that the air we breathe is safe and healthy for everyone.

Table 2 shows 1990 to 2019, attributed to outdoor air pollution – from ambient particulate matter and ozone – as a risk factor. According to the table, it is seen that the death rate due

to air pollution has increased worldwide since 1990. It is observed that this rate is especially high in Asian countries. In Nepal, Yemen, and Bangladesh, the death rate due to air pollution is relatively high.

In European countries, it is seen that death rates due to air pollution have decreased in general, but it is still a significant problem. Factors such as the level of development, technological infrastructure, and environmentally friendly policies can be cited as the reason for this situation. For example, in countries such as Sweden and Finland, the death rate due to air pollution is lower than in other European countries. The implementation of environmentally friendly policies and the development of technological infrastructure in these countries have been effective.

However, there is still a significant increase in death rates due to air pollution in some developed countries. For example, in countries like the USA and Germany, the death rate due to air pollution is still high. In this case, implementing environmentally friendly policies and technological development is especially important in these countries.

Table 2. *The death rate due to air pollution worldwide from 1990-2019 [7].*

Country	1990	2019	Relative Change
Finland	2.67%	0.73%	-73
Sweden	2.86%	0.81%	-72
Estonia	3.54%	1.01%	-71
Norway	3.70%	1.10%	-70
Scotland	4.39%	1.31%	-70
Ireland	4.81%	1.68%	-65
United Kingdom	6.41%	2.41%	-62
Switzerland	5.87%	2.34%	-60
Denmark	6.18%	2.61%	-58
United States	4.81%	2.01%	-58
Germany	6.88%	3.05%	-56
Luxembourg	5.37%	2.47%	-54
Netherlands	6.80%	3.15%	-54
Canada	3.19%	1.50%	-53
Iceland	1.85%	0.90%	-51
Lithuania	6.38%	3.13%	-51
Belgium	6.68%	3.33%	-50
Austria	6.28%	3.23%	-49
France	4.53%	2.41%	-47
Portugal	3.79%	1.99%	-47
Australia	1.99%	1.10%	-45
Russia	7.22%	3.99%	-45
New Zealand	1.70%	0.96%	-44
Czechia	9.82%	5.58%	-43
Italy	7.58%	4.29%	-43
Slovenia	7.16%	4.19%	-41
Slovakia	10.34%	6.27%	-39
Spain	4.33%	2.68%	-38
Malta	6.64%	4.20%	-37
Malaysia	9.31%	6.08%	-35
Ukraine	8.85%	5.81%	-34
Greece	7.35%	4.89%	-33
Israel	7.71%	5.17%	-33

Country	1990	2019	Relative Change
Croatia	8.50%	5.98%	-30
Singapore	8.34%	5.82%	-30
Bulgaria	10.05%	7.13%	-29
Poland	9.60%	6.78%	-29
Hungary	7.73%	5.57%	-28
Greenland	1.95%	1.45%	-26
Romania	7.36%	5.43%	-26
Cyprus	6.69%	5.08%	-24
Moldova	6.56%	5.06%	-23
Bahamas	4.52%	3.63%	-20
Montenegro	10.01%	7.99%	-20
Argentina	4.52%	3.73%	-17
Japan	3.70%	3.07%	-17
Cuba	6.59%	5.54%	-16
Mexico	6.18%	5.34%	-14
Virgin Islands	2.62%	2.29%	-13
Qatar	17.05%	15.14%	-11
Bahrain	16.42%	14.86%	-10
Serbia	9.50%	8.58%	-10
South Korea	7.91%	7.12%	-10
North Macedonia	11.55%	10.68%	-8
United Arab Emirates	13.61%	12.56%	-8
Georgia	6.88%	6.45%	-6
Jordan	10.45%	9.86%	-6
Brazil	3.47%	3.31%	-5
Panama	3.58%	3.41%	-5
Chile	5.17%	5.15%	0
Monaco	3.09%	3.13%	1
Taiwan	6.10%	6.37%	4
Albania	6.17%	6.52%	6
Syria	11.04%	11.65%	6
Venezuela	6.18%	6.65%	8
Colombia	4.86%	5.35%	10
Turkmenistan	9.78%	10.87%	11
Costa Rica	3.54%	3.97%	12
South Africa	4.85%	5.42%	12
Kazakhstan	6.70%	7.70%	15
Turkey	8.29%	9.51%	15
Armenia	9.67%	11.22%	16
Thailand	5.46%	6.51%	19
Egypt	13.04%	15.70%	20
Iran	9.10%	11.11%	22
Peru	4.83%	5.94%	23
Dominica	3.50%	4.33%	24
Algeria	8.65%	10.77%	25
Paraguay	2.64%	3.33%	26
Lebanon	7.75%	9.87%	27
Kyrgyzstan	6.45%	8.39%	30
Iraq	11.48%	15.01%	31
Bosnia and Herzegovina	7.19%	9.47%	32

Country	1990	2019	Relative Change
Libya	8.23%	11.09%	35
Philippines	3.56%	5.08%	43
Azerbaijan	6.91%	10.06%	46
World	**5.31%**	**7.80%**	**47**
Uzbekistan	8.66%	12.99%	50
Papua New Guinea	1.27%	1.94%	53
Jamaica	3.13%	5.00%	60
China	8.52%	13.74%	61
Maldives	2.46%	4.08%	66
Sri Lanka	3.14%	5.38%	71
North Korea	5.95%	10.53%	77
Palestine	6.31%	11.45%	81
Niger	1.14%	2.08%	82
Indonesia	3.13%	6.11%	95
Saudi Arabia	7.45%	14.85%	99
Senegal	1.97%	4.23%	115
Afghanistan	1.99%	4.31%	117
Kenya	1.04%	2.36%	127
Guinea-Bissau	1.37%	3.14%	129
Fiji	1.82%	4.20%	131
Vietnam	2.55%	6.06%	138
Ghana	3.02%	7.22%	139
Mali	0.86%	2.06%	140
Madagascar	0.70%	1.72%	146
Morocco	4.76%	11.75%	147
Cambodia	1.37%	3.45%	152
Dominican Republic	2.16%	5.46%	153
Myanmar	2.51%	6.38%	154
Nigeria	2.07%	5.40%	161
Laos	1.37%	3.59%	162
India	4.74%	12.57%	165
Pakistan	3.46%	9.42%	172
Zambia	1.11%	3.03%	173
Mozambique	0.35%	0.96%	174
Congo	2.22%	6.11%	175
Tanzania	0.72%	2.07%	187
Nepal	4.18%	13.43%	221
Djibouti	2.15%	6.97%	224
Bangladesh	2.78%	9.89%	256
Ethiopia	0.59%	2.12%	259
Yemen	2.14%	8.00%	274
Bhutan	2.14%	8.34%	290
Uganda	0.60%	2.51%	318
Sudan	2.24%	9.75%	335

Outdoor air quality requirements vary from country to country and are typically based on measured pollutants. Each country has set its limits and standards for outdoor air quality. It is seen that different countries have different limits for air pollutants depending on their geographical location, topography, climate, and economic activities. The standards according to the selected countries are given below.

USA:

The United States Environmental Protection Agency (EPA) [8] National Ambient Air Quality Standards (NAAQS) set standards for six common outdoor air pollutants, including PM2.5, PM10, NO_2, SO_2, ozone, and lead.

- Ozone: 70 parts per billion (ppb) above the 8-hour average,
- Fine Particulate Matter (PM2.5): 12 micrograms per cubic meter (µg/m^3) above the annual average,
- Carbon Monoxide (CO): 9 parts per million (ppm) on an 8-hour average,
- Sulphur Dioxide (SO_2): 75 ppb in 1-hour average,
- Nitrogen Dioxide (NO_2): 53 ppb above annual average

European Union:

The European Union has developed ambient air quality standards, Ambient Air Quality and Cleaner Air for Europe, Directive 2008/50/EC, [9] that are mandatory for all member states. These limit values aim to provide good air quality and protect human health. Primarily measured pollutants include PM10, PM2.5, NO_2, SO_2 and benzene.

- Ozone: 120 µg/m^3 for a maximum period of 8 hours,
- Fine Particulate Matter (PM2.5): 25 µg/m^3, above the annual average,
- Carbon Monoxide (CO): 10 ppm above the 8-hour average,
- Sulphur Dioxide (SO_2): 350 µg/ m^3 over 24 hours,
- Nitrogen Dioxide (NO_2): 40 µg/m^3 above the annual average

China:

Under Ambient Air Quality Standards (National Standard GB 3095-2012), [10] China has set limits for outdoor air pollutants, including PM10, PM2.5, NO_2, SO_2 and CO.

- Ozone: 160 µg/m^3 in 8-hour average,
- Fine Particulate Matter (PM2.5): 35 µg/m^3 above the annual average,
- Carbon Monoxide (CO): 20 ppm on an 8-hour average,
- Sulphur Dioxide (SO_2): 150 µg/m3 over 24 hours,
- Nitrogen Dioxide (NO_2): 40 µg/m^3 above the annual average

India:

The Central Pollution Control Board (CPCB) [11] sets India's air quality standards with the Central Pollution Control Board (CPCB), Ministry of Environment and Forest (MoEF), and Bureau of Indian Standards (BIS) regulatory bodies. The CPCB has set limits for PM10, PM2.5, NO_2, SO_2 and lead.

- Ozone: 100 µg/m^3 in 8-hour average,
- Fine Particulate Matter (PM2.5): 40 µg/m^3 above the annual average,
- Carbon Monoxide (CO): 4 ppm above the 8-hour average,
- Sulphur Dioxide (SO_2): 80 µg/m^3 over 24 hours,
- Nitrogen Dioxide (NO_2): 80 µg/m^3 above the annual average

These limits and standards may vary by region or city in each country. Outdoor air quality requirements aim to protect public health and the environment.

10.2. An example of working with a dataset

10.2.1. Climatic Conditions

The sensors in Adana have a Mediterranean climate classified under the Köppen Climate Classification system. The region experiences long, hot summers and short, mild winters. Adana's average dry bulb temperature varies between 18.1°C and 28.7°C throughout the year (Fig. 10.2). In summer, temperatures usually peak in late July and August, with daytime temperatures exceeding 35°C. On the other hand, winter temperatures usually hover around 10-15°C. Average monthly relative humidity in Adana typically ranges from 49% in August to 81% in January. Air pollution is a common problem in densely populated areas. The area experiences many summer days, moderate air pollution, and relatively low humidity.

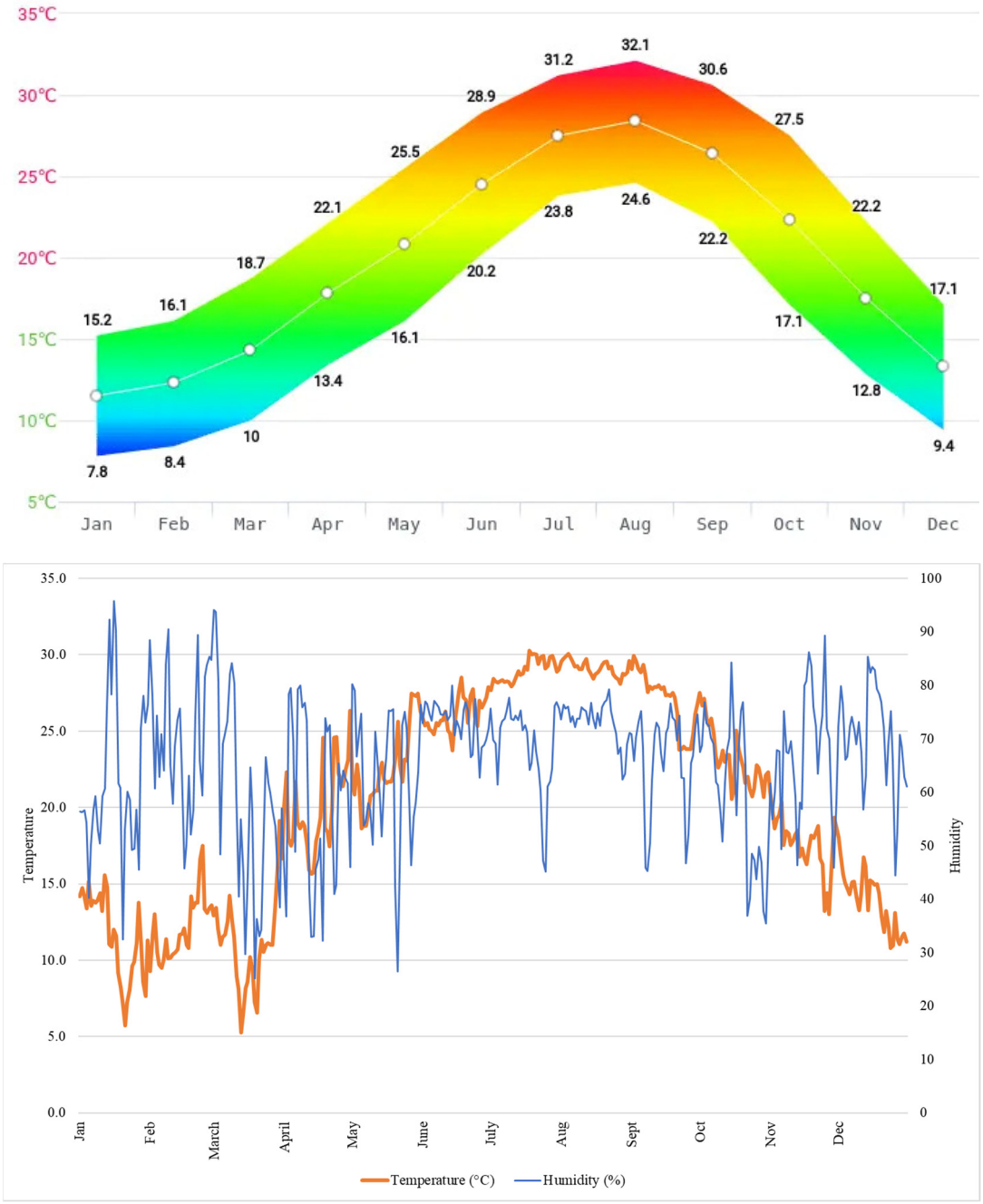


Figure 10.2. *Adana/Turkey's average monthly temperatures*

10.2.2. Sensors' location

The following figures unveil the geographic placement of three air pollution meters within the city of Adana, situated in the Southeastern region of Turkey, known for its bustling streets and industrial activities (Fig. 10.3). These innovative scientific devices are armed with cutting-edge plug-and-play technology, specifically designed to measure atmospheric PM2.5, PM1, and PM10 particles and monitor Carbon Dioxide, temperature, and air humidity. These advanced air quality monitoring stations aim to provide practically immediate, precise, and reliable data on the city's air quality, operating seamlessly through the installation of Wi-Fi connectivity. The first sensor is located at 20FDDC62 (1), 37.011 latitudes and 35.280 longitudes, in the Yuregir region. The sensor with code 20FD2908 (2) is in an apartment above 37.0320 Latitude and 35.302 longitudes in the Karaisalı area of Adana city. The last sensor, located at 20FD51B8 (3), 37.061 latitudes and 35.384 longitudes, is in the Seyhan region.

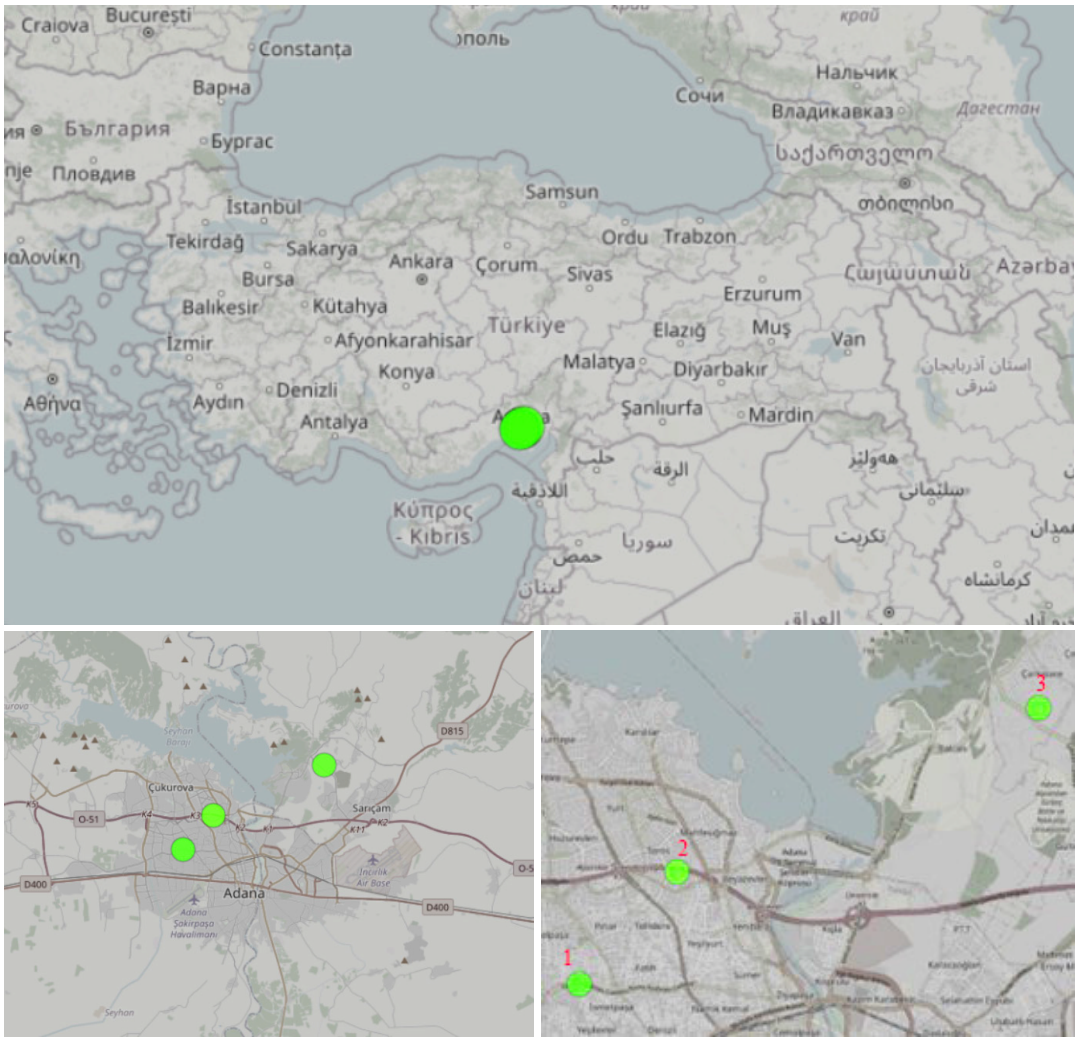


***Figure 10.3**. The location of air pollution sensors*

10.2.3. Data Collection

URAD (Urban Network of Air Quality Devices) monitor was used for these measurements. URAD Monitor is a portable air quality monitoring device designed to measure air pollution levels in urban areas. It is a small box-like sensor mounted on a wall or placed on a flat surface. The sensors located here are on the balconies of some apartments in different locations at different times, and

measurements are taken. The identified urban areas include commercial, residential, and working areas, and each has a different population density and, therefore, environmental pollution.

Using the sensor is very simple. The most basic requirement for the sensor to be used is to connect it to a Wi-Fi network. In this way, the device collects air quality data and sends it to the cloud, which can be accessed via a mobile app or web-based dashboard.

The sensor uses a combination of sensors to collect data. It includes a particulate matter (PM) sensor, carbon dioxide (CO_2) sensor, temperature sensor, and humidity sensor. These sensors are designed to detect and measure the levels of various air pollutants in the environment.

The URAD monitor comprises several components: the air pollution sensor, sensors, microprocessor, Wi-Fi module, and power supply. The data transfer of the sensors takes place in 3 stages. The sensors collect data about the levels of pollutants in the air, and this data is then processed by the microprocessor and sent to the cloud via the Wi-Fi module.

Figures 4, 5, and 6 show the received results of sensors S1-S2-S3, respectively, marked on the map. The World Air Quality Index project hosts a website to access the results of air pollution meters worldwide. This website has an interface available worldwide to instantly read the air quality values. It creates a graph from the day the measurement starts and presents the results. The graphics shown here were obtained from this website. Adana has 11 sensors in total, and the uRAD monitor provides three, made by students during their summer school.

The specific air quality levels shown by colors in the graphs are described in table 3. The value indicated in dark green here is between 0-50, and the air quality is defined as good. Moderate between 51-100, and air quality is acceptable. On the other hand, 101-150 is described as unhealthy for sensitive groups, and 151-200 as unhealthy. Between 201-300 is determined as very unhealthy, and health effects are seen. If it is 300 and above, it is a hazardous level. This scale has been determined according to the US-EPA 2016 standard, and the definitions of the values measured by the three sensors shared here are in colored Table 3 [12].

Table 3. *Air Quality Index Scale and Color Legend [12]*

AQI	Air Pollution Level	Health Implications	Cautionary Statement (for PM2.5)
0–50	Good	Air quality is considered satisfactory, and air pollution poses little or no risk	None
51–100	Moderate	Air quality is acceptable; however, for some pollutants, there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.	Active children, adults, and people with respiratory diseases, such as asthma, should limit prolonged outdoor exertion.
101–150	Unhealthy for Sensitive Groups	Members of sensitive groups may experience health effects. The public is not likely to be affected.	Active children, adults, and people with respiratory diseases, such as asthma, should limit prolonged outdoor exertion.
151–200	Unhealthy	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects	Active children and adults, and people with respiratory diseases, such as asthma, should avoid prolonged outdoor exertion; everyone else, especially children, should limit prolonged outdoor exertion

201–300	Very Unhealthy	Health warnings of emergency conditions. The entire population is more likely to be affected.	Active children, adults, and people with respiratory diseases, such as asthma, should avoid all outdoor exertion; everyone else, especially children, should limit outdoor exertion.
300+	Hazardous	Health alert: everyone may experience more serious health effects	Everyone should avoid all outdoor exertion

Sensor S1 has made measurements for 122 days, and the measurements are continuing. Here, the relative humidity value is 55.87% on average for 122 days, with a minimum of 43.67 and a maximum value of 67.47%. The measurement started in the winter months for Adana, so the average temperature is 14.21°C, the minimum temperature is 9.9°C, and the maximum temperature is 19.2°C. When PM1 values are considered, the highest value is 39.4 in this region, while the lowest PM1 value is 3.3 µg/m^3. The average is 12.45 µg/m^3. The average PM10 value is 22.7 µg/m^3, but on some days, it reaches very high levels, with a maximum of 85.2 µg/m^3 and a minimum of 6.25 µg/m^3. While the value of PM2.5 is good on average, the maximum is 75, which is acceptable but a moderate value. Moreover, the lowest value is 5.6 µg/m^3.

Sensor S2 has made measurements for 73 days, and the measurements continue. Here, the relative humidity value is 47.26% on average for 73 days, with a minimum of 36.1 and a maximum value of 57.30%. On the other hand, the temperature value is 17.6°C on average, the minimum temperature is 13.3°C, and the maximum temperature is 21.2°C for the 73 days of measurement. When PM1 values are considered, the highest value is 52.1 µg/m^3 in this region, while the lowest PM1 value is 3.5 µg/m^3 on average. The average is 14.61 µg/m^3. The average PM10 value is 25.8 µg/m^3, but on some days, it reaches very high levels, with a maximum of 88.1 µg/m^3 and a minimum of 6.75 µg/m^3. While the PM2.5 value is good on average, a maximum of 36.5 µg/m^3 is seen. Moreover, the lowest value is 4.02 µg/m^3.

According to the results of Sensor S3, which took measurements for 134 days, the average relative humidity value is 52.70%, the minimum value is 39.35%, and the maximum value is 46.14%. The average temperature is 18.9°C, the minimum temperature is 15.3°C, and the maximum is 24.4°C. When PM1 values are considered, the highest value is 15.6 µg/m^3 in this region, while the lowest PM1 value is 1.8 µg/m^3 on average. The average is 5.7 µg/m^3. The PM10 value is 10.42 µg/m^3 on average, with a minimum of 3.5 µg/m^3 and a maximum of 29.6 µg/m^3. While the PM2.5 value is at good levels on average, the maximum value of 26.02 µg/m^3 is seen. Furthermore, the lowest value is 3.3 µg/m^3.

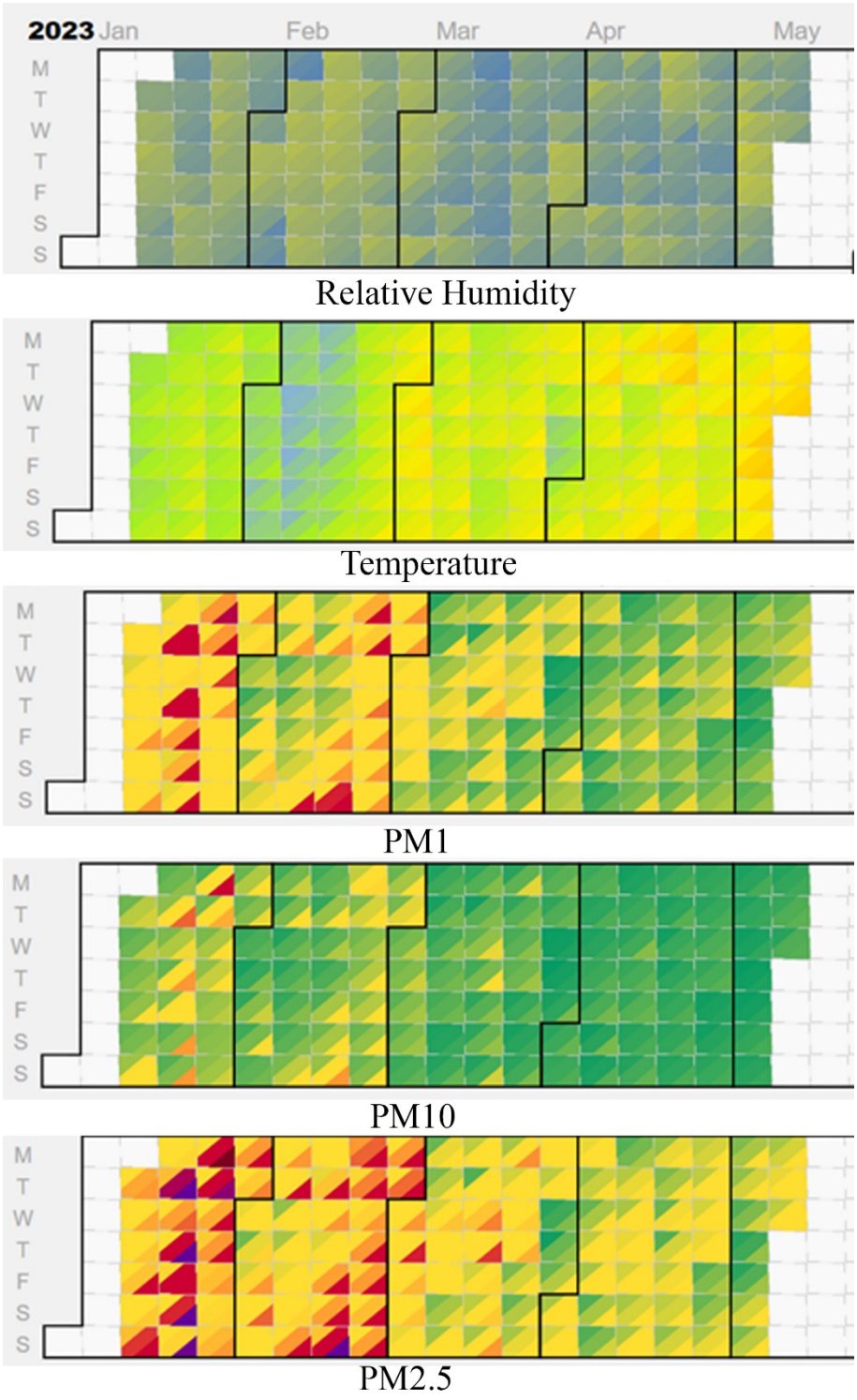


Figure 10.4. *Sensor S1 measurement results for 122 days*

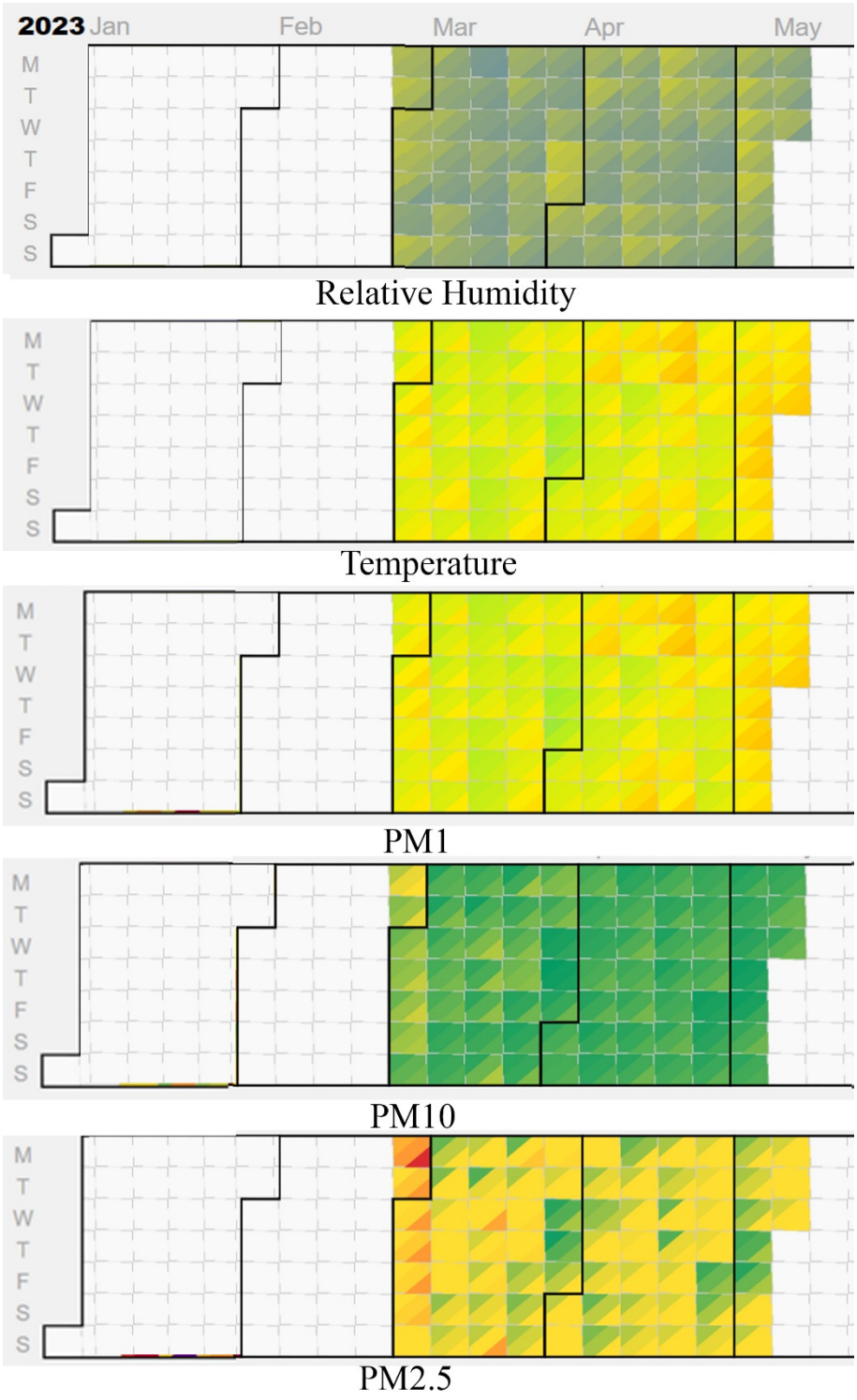


***Figure 10.5**. Sensor S2 measurement results for 73 days*

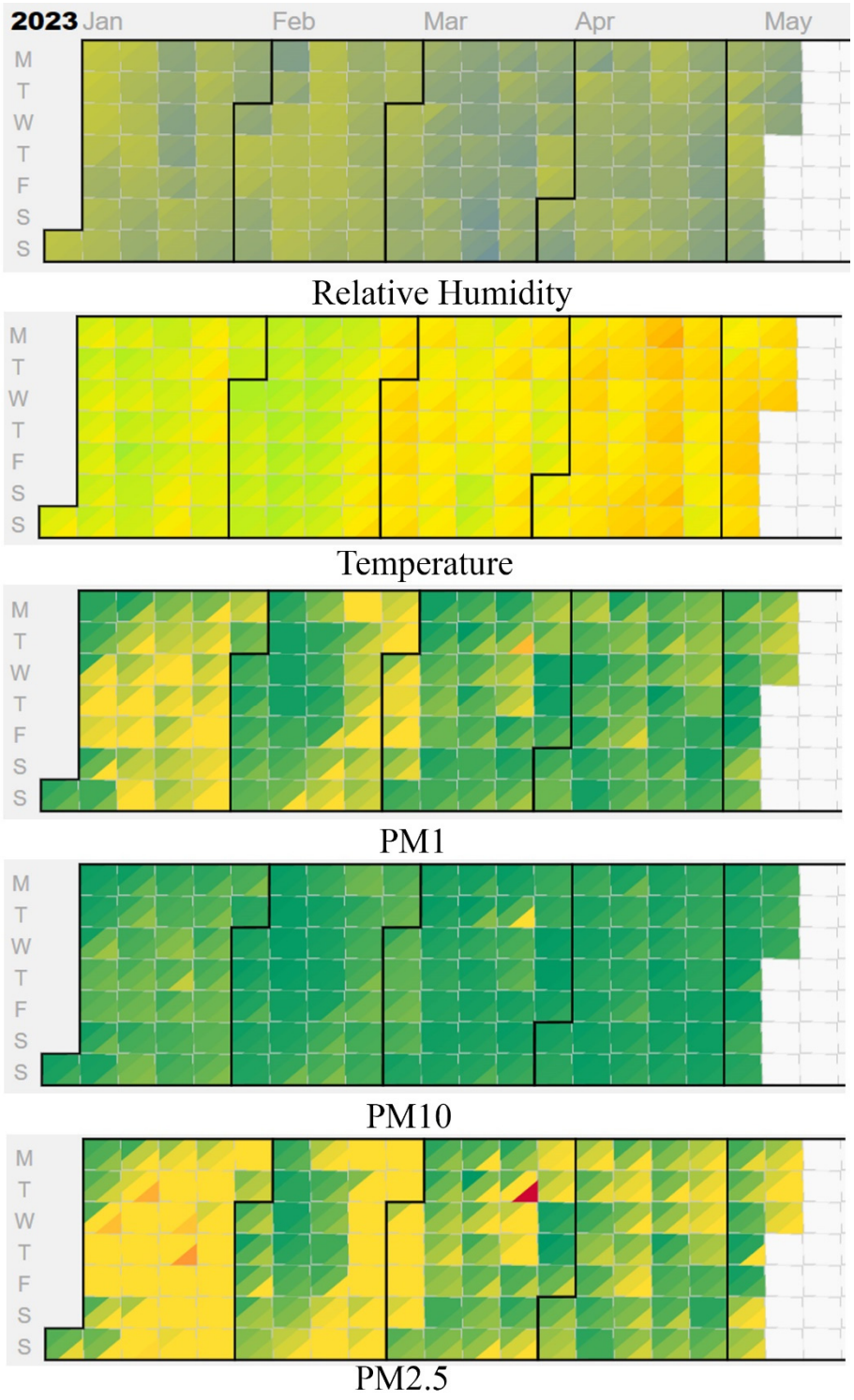


Figure 10.6. *Sensor S3 measurement results for 134 days*

10.3. Data Analysis

SPSS Software is a statistical program used in academic research and industrial data analysis. SPSS can be used in data collection, analysis, visualization, and reporting. This program also supports data mining methods such as cluster analysis, classification, regression, and consolidation analysis.

The analysis results will be shared by showing how the obtained data in this study will be processed in the SPSS program. Thus, this report shows how the obtained data can be analysed by the SPSS statistical program and what analyses can be made.

This guide explains how to use SPSS Software step by step. In addition, this program can be used in data collection, analysis, visualization, and reporting sections.

10.3.1. Data Collection

The data analysed below were created with the PM sensors. The PM sensor is a device that measures air quality and works connected to the internet together with the power cable. The PM sensor measures the air quality by following the steps below.

1. Take the PM sensor out of the box and plug it into power.
2. Wi-Fi settings must be adjusted to connect the sensor to the internet. For this, go to the setup page using the device's own Wi-Fi and connect the device to the home network.
3. Select a suitable location for the PM sensor to use. Make sure that the device is not covered and is outdoors.
4. The PM device saves the data and automatically transfers it to the website. These data are accessible via the uRADmonitor web interface.

Figure 10.7 shows the operating state of Sensor1. The sensor is located on an apartment balcony in Adana/Turkey and has always been used in the same location for data transmission. A green light indicates that data transfer is in progress.

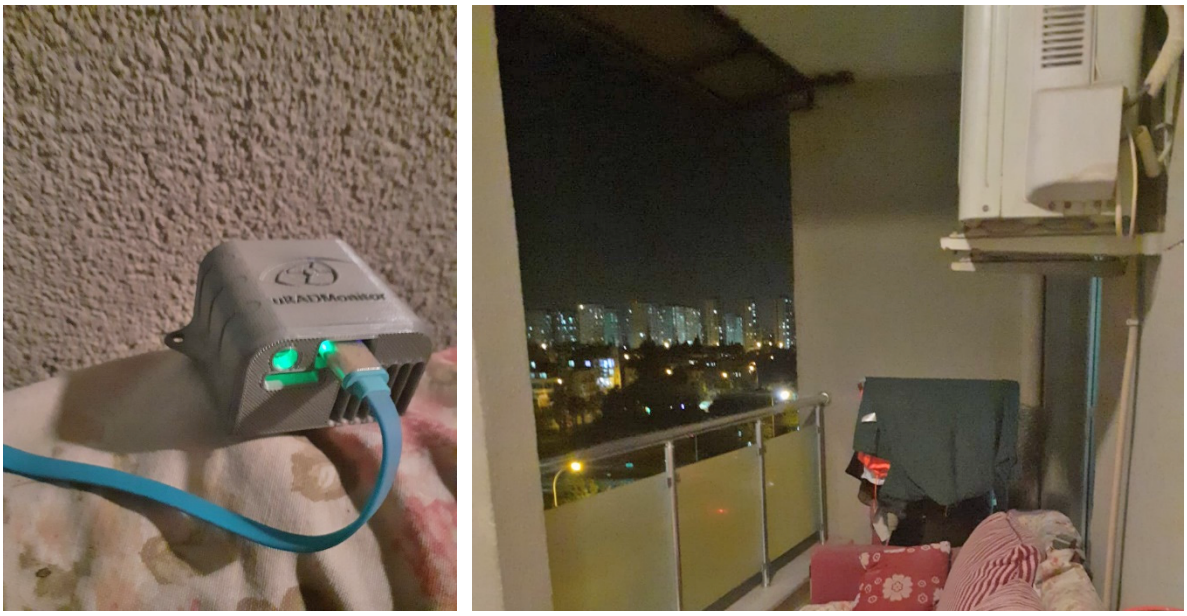


***Figure 10.7**. PM Sensor S1*

10.3.2. Data Upload

SPSS Software program offers straightforward usage of the data loading process. One of the most essential features of this program is that it supports different file formats. It is possible to open files in the program in XLSX, CSV, or SPSS format. After loading the data, the variables and their properties can be viewed by switching to the "Variable View" tab to ensure the data is loaded correctly.

- Open the program.
- Select “Open” from the “File” menu (Fig. 10.8).

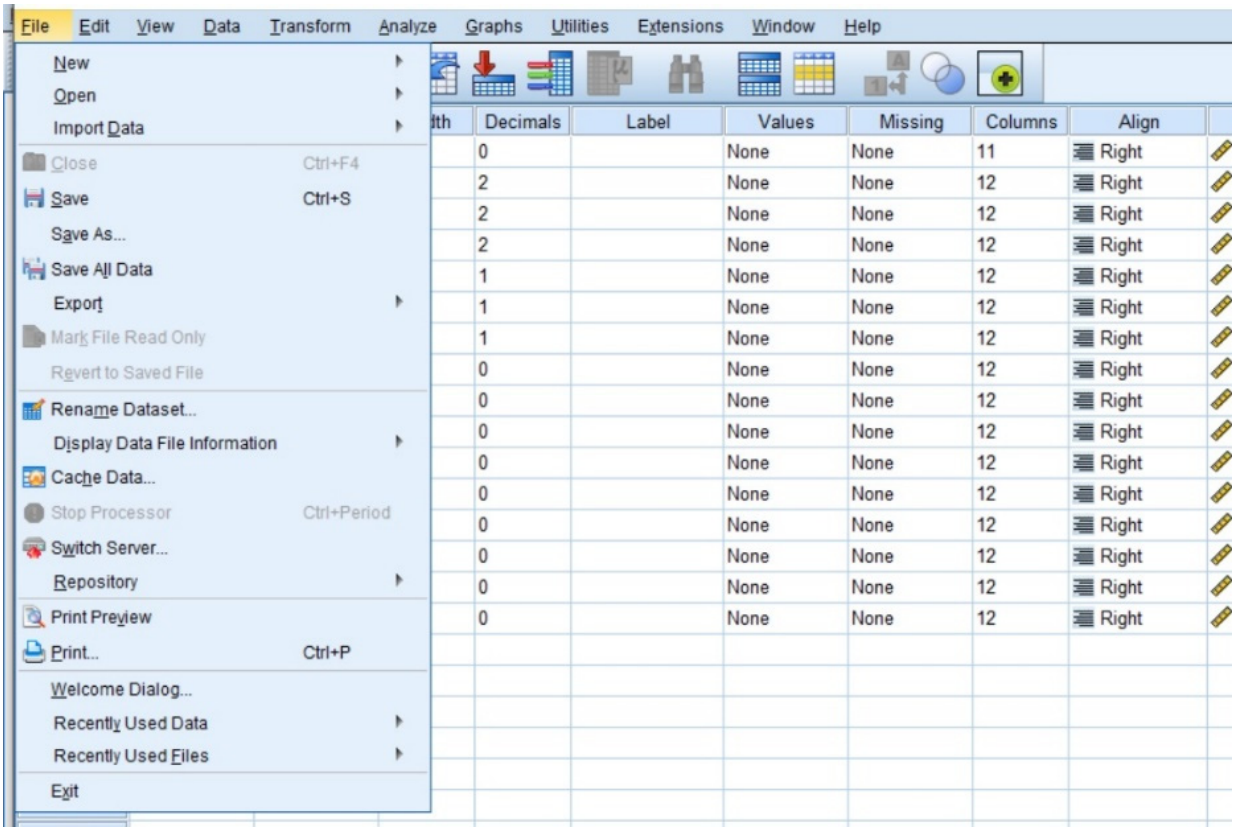


Figure 10.8. *SPSS opening dataset*

- Choose the data file.
- Select the data file format.
- Click the “Open” button.
- Switch to the “Variable View” tab to ensure the data is loaded correctly.

Table 4. *Sample data from Sensor S1, Sensor S2 and Sensor S3*

	S1_temp	S2_temp	S3_temp	S1_RH	S2_RH	S3_RH	S1_PM1	S2_PM1	S3_PM1	S1_PM2.5	S2_PM2.5	S3_PM2.5	S1_PM10	S2_PM10	S3_PM10
1.4.23 0:00	7.56	10.01	14.86	60	49	42	10	11	5	16	18	8	16	21	8
1.4.23 0:01	7.53	10.09	14.9	59.5	48.5	41.5	9	13	6	15	21	8	15	24	9
1.4.23 0:02	7.31	10.15	14.95	60	48.5	41.5	8	12	6	11	21	9	12	24	9
1.4.23 0:03	7.37	10.02	14.98	60	48.5	41.5	9	11	5	16	21	9	17	25	9
1.4.23 0:04	7.6	9.92	15	59.5	48.5	41.5	8	13	6	14	21	8	14	28	8
1.4.23 0:05	7.46	10	15.02	60	48.5	41.5	11	13	6	17	22	8	18	25	9
1.4.23 0:06	7.53	10.04	15.02	59.5	48.5	41.5	9	13	6	16	21	8	16	26	8
1.4.23 0:07	7.69	9.99	14.96	59.5	48.5	41.5	10	10	5	17	18	9	18	20	9
1.4.23 0:08	7.63	10.03	14.96	59.5	48.5	41.5	11	10	6	20	16	9	21	18	9
1.4.23 0:09	7.51	9.98	14.98	59	48.5	41.5	10	9	6	14	16	8	15	17	8
1.4.23 0:10	7.61	10.05	14.99	59	48.5	41.5	8	10	5	14	13	8	16	14	8
5.4.23 0:00	17.47	16.74	25.17	71	66.5	44.5	12	11	1	18	18	1	20	21	2
5.4.23 0:01	17.81	16.67	25.14	69.5	66.5	44.5	8	12	1	14	18	3	14	22	3
5.4.23 0:02	17.98	16.67	25.12	69.5	66.5	44.5	9	12	1	16	20	3	18	25	4
5.4.23 0:03	17.68	16.95	25.1	71	66	44.5	9	12	0	11	21	1	11	26	1
5.4.23 0:04	17.7	16.89	25.08	71	66	44.5	9	11	1	16	16	1	17	17	2
5.4.23 0:05	17.18	16.96	25.06	72.5	66	44.5	10	11	1	18	19	2	22	27	2
5.4.23 0:06	16.96	17.04	25.04	73	65.5	44.5	11	12	1	18	20	2	20	26	4
5.4.23 0:07	17.04	17.04	25.02	73	65.5	44.5	9	13	1	16	20	2	20	26	2
5.4.23 0:08	17.25	16.79	25	72	66	44.5	11	12	1	17	18	2	18	23	3
5.4.23 0:09	17.08	16.63	24.98	72.5	66.5	44.5	10	11	0	16	16	1	18	17	2
5.4.23 0:10	17.29	16.7	24.96	72	66.5	44.5	9	13	1	14	19	2	15	26	2
10.4.23 0:00	14.04	15.38	24.1	70	58.5	46.5	1	2	4	4	3	8	5	5	10
10.4.23 0:01	14.12	15.42	24.09	70	58.5	46.5	2	2	5	5	4	8	7	7	10
10.4.23 0:02	14.08	15.39	24.09	70	58.5	46.5	1	2	7	5	5	11	5	7	12
10.4.23 0:03	13.73	15.45	24.09	71	58.5	46.5	2	3	4	4	5	7	4	7	8
10.4.23 0:04	14.06	15.52	24.08	70	58.5	46.5	2	2	5	5	4	8	6	5	9
10.4.23 0:05	14.35	15.57	24.07	69.5	58.5	46.5	2	3	5	4	6	7	5	10	7
10.4.23 0:06	14.05	15.58	24.07	70	58.5	46.5	2	3	5	5	6	9	5	9	10
10.4.23 0:07	14.29	15.62	24.06	69.5	58.5	46.5	2	3	4	5	6	9	5	10	9
10.4.23 0:08	14.31	15.63	24.06	69	58	46.5	1	4	4	4	7	7	6	12	9
10.4.23 0:09	14.34	15.65	24.05	69	58	46.5	2	3	5	4	5	9	5	7	10
10.4.23 0:10	14.48	15.68	24.04	69	58	46.5	2	3	6	4	5	9	6	7	11

	S1_temp	S2_temp	S3_temp	S1_RH	S2_RH	S3_RH	S1_PM1	S2_PM1	S3_PM1	S1_PM2.5	S2_PM2.5	S3_PM2.5	S1_PM10	S2_PM10	S3_PM10
15.4.23 0:00	15.12	14.89	23.91	64.5	59.5	42.5	17	13	3	29	20	4	35	22	5
15.4.23 0:01	15.12	14.89	23.91	64	59.5	42.5	18	12	3	30	18	5	37	20	5
15.4.23 0:02	15.2	14.88	23.93	64	59.5	42.5	17	13	3	28	20	4	34	22	4
15.4.23 0:03	15.05	14.88	23.94	64.5	59.5	42.5	18	11	3	29	19	5	36	21	5
15.4.23 0:04	15.12	14.86	23.95	64	59.5	42.5	16	11	3	26	19	5	32	20	5
15.4.23 0:05	15.02	14.85	23.96	64.5	59.5	42.5	17	12	3	31	20	4	38	22	4
15.4.23 0:05	15.14	14.85	23.97	64	59.5	42.5	17	12	3	27	20	5	33	22	5
15.4.23 0:07	15.13	14.84	23.99	64	59.5	42.5	18	11	2	29	18	3	37	19	3
15.4.23 0:08	15.08	14.83	24	64.5	59.5	42.5	18	12	1	28	20	3	33	23	3
15.4.23 0:09	15.17	14.83	24.01	64	59.5	42.5	18	10	3	30	16	4	37	17	5
15.4.23 0:10	15.13	14.84	24.03	64	59.5	42.5	19	9	3	31	16	5	38	17	5
20.4.23 0:00	14.36	16.65	22.79	75.5	62.5	49.5	2	2	2	4	4	3	4	5	4
20.4.23 0:01	14.66	16.58	22.79	75	62.5	49.5	1	3	2	3	5	4	3	5	4
20.4.23 0:02	14.87	16.61	22.79	74.5	62.5	49.5	2	1	2	3	2	3	3	3	3
20.4.23 0:03	14.64	16.65	22.79	75	62.5	49.5	3	2	3	4	3	4	5	5	4
20.4.23 0:04	14.9	16.68	22.79	74.5	62.5	49.5	2	3	3	3	5	3	4	7	3
20.4.23 0:05	14.44	16.68	22.79	75.5	62	49.5	3	3	2	5	6	4	7	7	5
20.4.23 0:06	13.84	16.72	22.78	77.5	62	49.5	3	3	3	5	5	4	7	6	4
20.4.23 0:07	14.27	16.71	22.78	76	62	49.5	3	3	3	4	5	4	5	6	4
20.4.23 0:08	14.33	16.67	22.78	76	62	49.5	3	3	2	4	5	3	4	6	3
20.4.23 0:09	14.15	16.63	22.77	76.5	62	49.5	2	3	3	4	6	3	5	6	3
20.4.23 0:10	14.41	16.63	22.78	76	62.5	49.5	3	3	3	4	5	4	4	6	4
25.4.23 0:00	16.17	17.09	21.37	67.5	59.5	55	6	6	7	8	9	10	8	9	10
25.4.23 0:01	16.14	17.1	21.37	68	60	55	7	5	6	10	10	9	10	10	10
25.4.23 0:02	15.9	17.13	21.36	68	59.5	55	7	7	6	10	10	9	11	11	9
25.4.23 0:03	16.09	17.15	21.36	67.5	59.5	55	4	8	6	6	11	10	6	12	10
25.4.23 0:04	15.94	17.19	21.35	68	59.5	55	6	7	8	11	10	12	12	11	12
25.4.23 0:05	15.98	17.22	21.36	68	59.5	55	3	6	6	5	9	9	5	10	10
25.4.23 0:06	15.91	17.2	21.35	68	60	55	6	6	6	10	10	9	11	10	9
25.4.23 0:07	15.9	17.19	21.35	68.5	60	55	7	7	7	11	11	10	11	11	10
25.4.23 0:08	16.09	17.17	21.35	67.5	60	55	6	6	6	9	11	9	9	11	10
25.4.23 0:09	16.27	17.2	21.34	67.5	60	55	7	6	7	10	12	10	10	12	11
25.4.23 0:10	16.27	17.21	21.35	67	60	55	5	7	5	8	12	7	8	12	8

10.3.3. Descriptive Analysis

This program will show the basic statistical properties of the data. The SPSS Software program allows the analysis and examination of the data. This analysis menu is reached by selecting the "Descriptive Statistics" option from the "Analyze" menu to examine your data. This option shows the basic statistical characteristics of the data. These features include mean, standard deviation, median, and quartiles. By analysing these features, one can gain important insights about the data.

- Select "Descriptive Statistics" from the "Analyze" menu (Fig. 10.9).

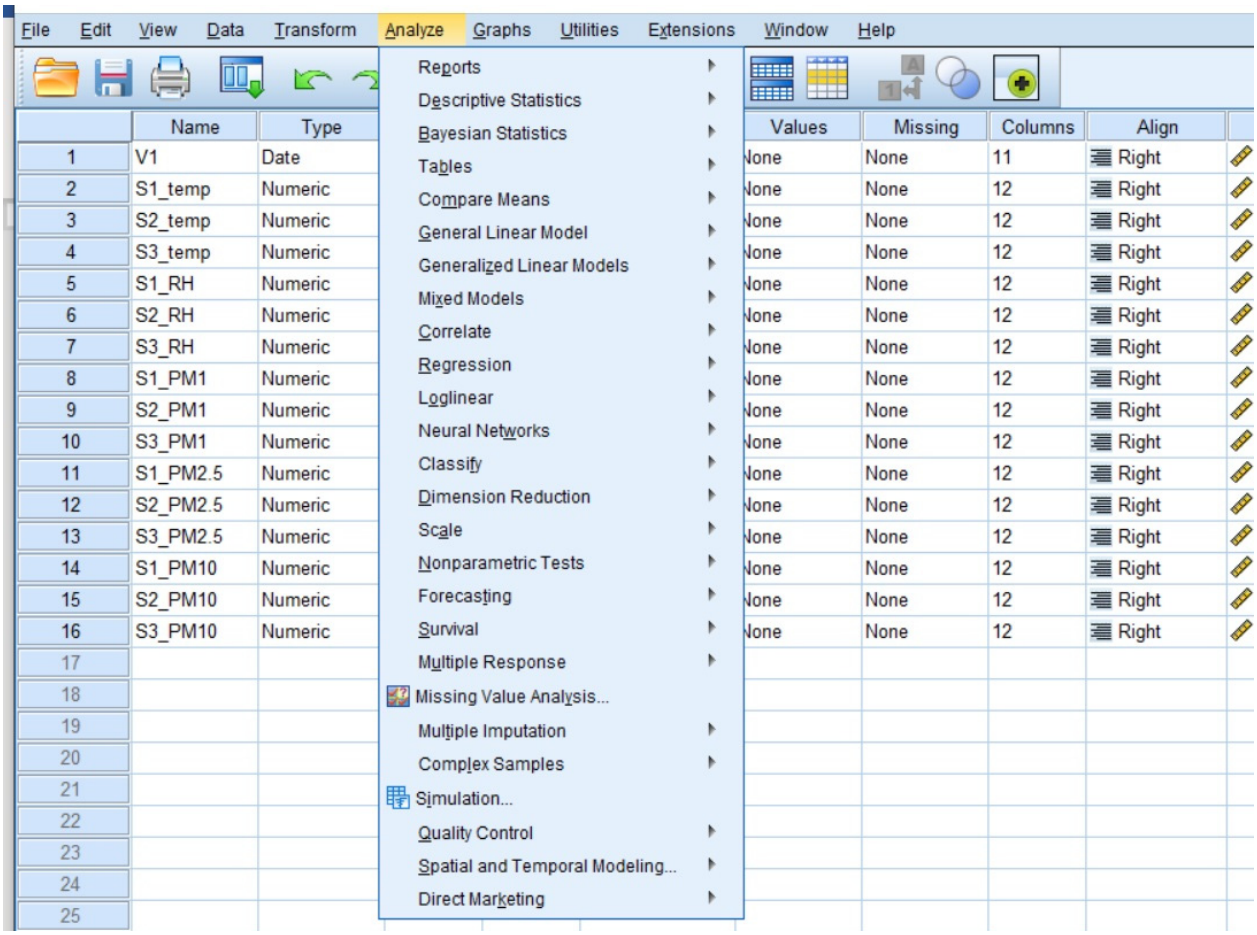


***Figure 10.9**. SPSS analyses menu*

- Select the relevant variables.
- Click the "OK" button.

Table 5 includes descriptive analysis and shows the temperature, humidity, and PM (particulate matter) levels measured by the different sensors. This table contains basic statistical features such as minimum, maximum, mean, and standard deviation.

Among the temperature measurements, the temperature values measured by the S3 sensor are higher compared to other sensors. There is not a huge difference between the S1 and S2 sensors. Among the humidity measurements, the humidity values measured by the S1 sensor are the highest, and the humidity values measured by the S3 sensor are the lowest.

Among the PM1, PM2.5, and PM10 measurements, the PM1 and PM10 values measured by the S2 sensor are higher than the other sensors. The PM2.5 values measured by the S1 sensor are the highest, and the PM2.5 values measured by the S3 sensor are the lowest.

The S3 sensor measured higher temperature values than other sensors, while the S1 sensor measured higher humidity values. Among PM measurements, the S2 sensor measured higher PM1 and PM10 values. The S1 sensor measured higher PM2.5 values. Considering the standard deviation values, it can be said that the values measured by the S3 sensor are generally less variable.

Table 5. *Descriptive analysis of the data*

	N	Minimum	Maximum	Mean	Std. Deviation
S1_temp	39900	5	29	17.63	4.014
S2_temp		8	33	18.52	4.166
S3_temp		11	36	22.52	3.382
S1_RH		31	83	58.19	10.632
S2_RH		24	69	50.57	10.019
S3_RH		31	75	49.27	6.721
S1_PM1		0	70	6.70	3.857
S2_PM1		0	59	7.45	3.905
S3_PM1		0	86	4.22	3.441
S1_PM2.5		0	150	10.93	6.279
S2_PM2.5		0	138	11.87	6.221
S3_PM2.5		0	176	6.78	5.442
S1_PM10		0	163	11.82	7.179
S2_PM10		0	172	13.60	7.438
S3_PM10		0	195	7.24	6.000

Table 5 presents air quality data: temperature, relative humidity, and particulate matter (PM1, PM2.5, and PM10) measurements from three different sensors (S1, S2, and S3). Data were analyzed for April in total. The minimum, maximum, mean, and standard deviation values for each parameter are presented in the table.

The conclusions from the data analysis are as follows. The average temperature for the S1 is 17.63°C with a standard deviation of 4.014. The temperature values measured by S2 and S3 are higher than measured by S1; the maximum temperature of S2 is 33°C, and the maximum temperature of S3 is 36°C. For Relative Humidity (RH), the mean RH for S1 is 58.19%, with a standard deviation 10,632. The RH values for S2 and S3 are lower than S1; S2 has a maximum of 69% RH, and S3 has a maximum of 75% RH. According to the PM concentrations data, mean PM concentrations for S1, S2, and S3 decrease in the order PM10 > PM2.5 > PM1.

10.3.4. Data Visualization

SPSS Software allows also the analysis of the data by visualizing. Data can be visualized by selecting the appropriate graph type from the "Graphs" menu. For example, by the "Bar Chart" (Fig. 10.11) can be compared the measurements of different sensors. The "Scatterplot"

option (Fig. 10.12) can create a graph showing the relationship between two sensors. These charts can be used to understand differences in data and visualize results.

- Select the appropriate graph options from the "Graphs" menu (Fig. 10.10).

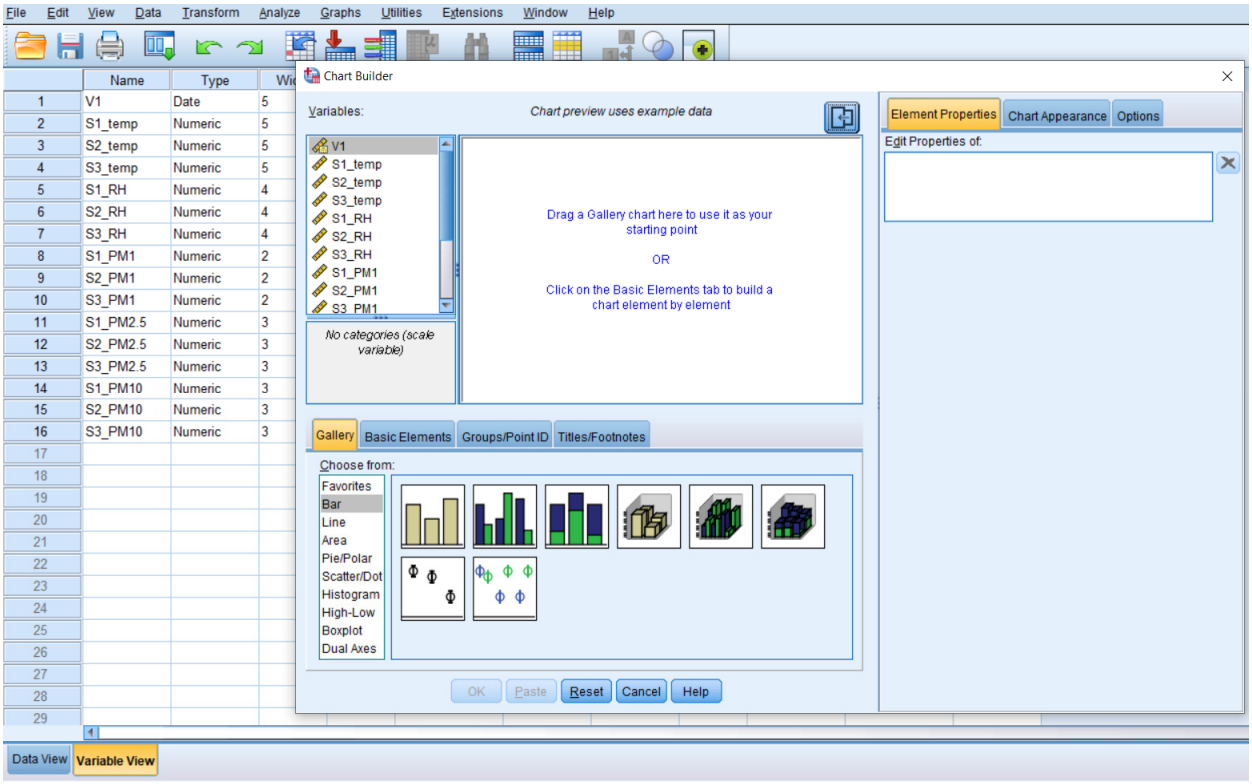


Figure 10.10. *SPSS Graph menu*

- Select the relevant variables.
- Click the "OK" button.
- The program will create the selected graph.

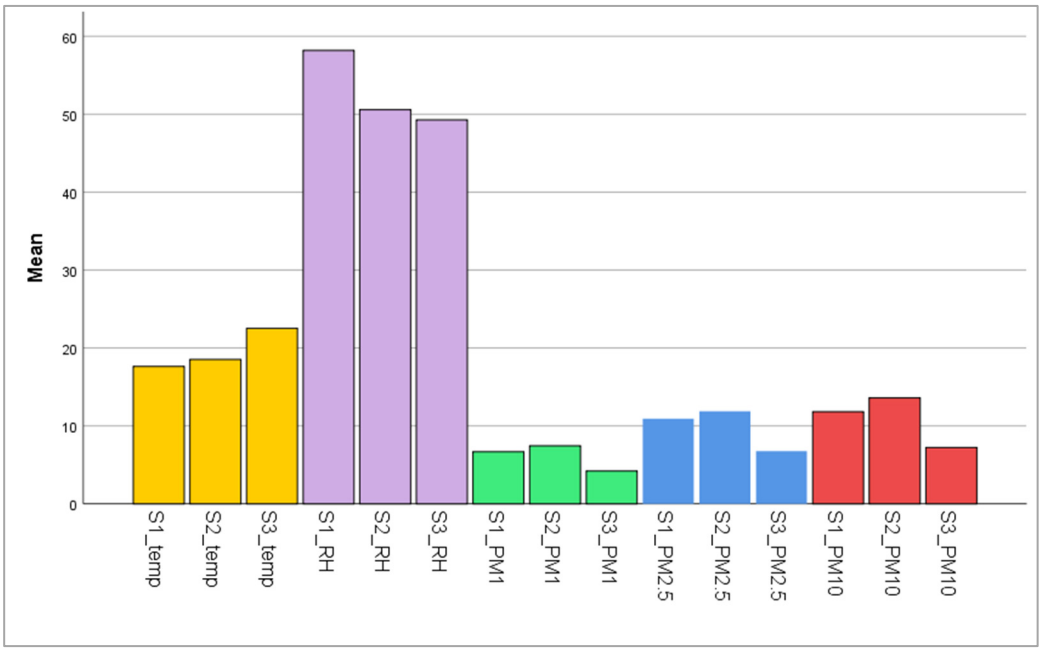


Figure 10.11. *Bar Chart of the data*

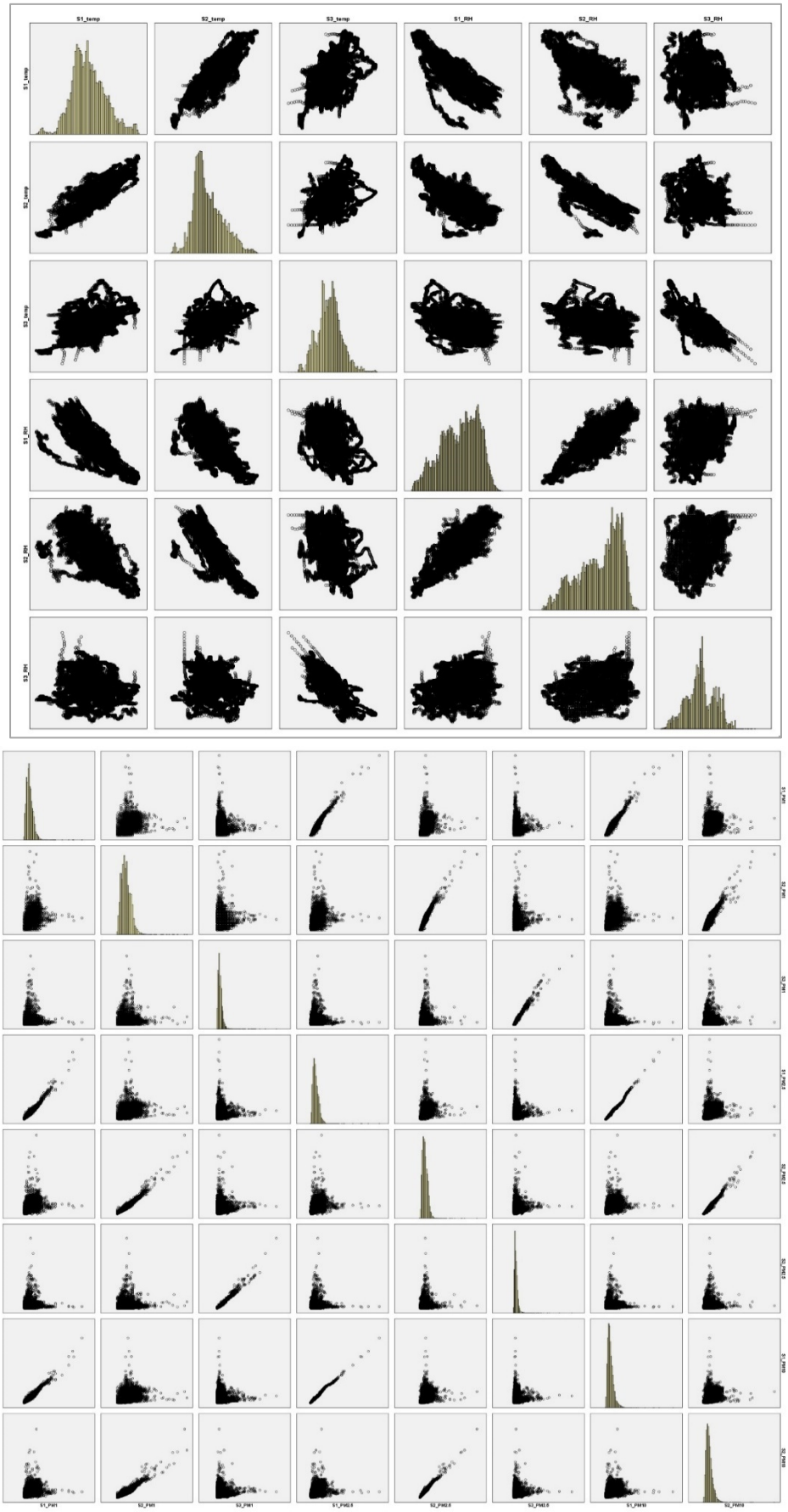


***Figure 10.12**. Scatterplot graphs of the data*

Pearson correlation coefficient formula:

$$r = \frac{N\Sigma xy-(\Sigma x)(\Sigma y)}{\sqrt{[N\Sigma x^2-(\Sigma x)^2][N\Sigma y^2-(\Sigma y)^2]}}$$

N = number of pairs of points
Σxy = sum of products of paired scores
Σx = sum of x scores
Σy = sum of y scores
Σx^2 = sum of squared x scores
Σy^2 = sum of squares of y scores

Scatter plots analyze whether there is a linear relationship between two variables. However, it is necessary to calculate the Pearson correlation coefficient to understand the relationship between the two data in this graph. This coefficient defines whether there is a large, moderate, or small correlation and positive or negative relationship between two data. Although the correlation coefficient takes values ranging from -1 to +1 ($-1 \leq r \leq +1$), values between 0.00 and 0.25 are 'very weak.' Values between 0.26 and 0.49 are 'weak' in correlation coefficients. A value between 0.50 and 0.69 is 'medium,' a value between 0.70 and 0.89 is 'high,' and a value between 0.90 and 1.00 is 'very high.' Therefore, it is necessary to calculate these coefficients to interpret this graph. Table 7 gives the correlation coefficient values. According to this table, it can be read that while there is a negative correlation between temperature and relative humidity, there is a positive correlation between PMs.

10.3.5. Regression Analysis

Regression analysis is a technique for determining the relationship between two or more variables. By choosing the "Regression" option from the "Analyze" menu, regression analysis may be run. Following this study, it will be possible to comprehend how the variables are related and use that relationship to make predictions for the future. Regression analysis, for instance, may be used to find the relationship between temperature and humidity by choosing these two variables (Table 6).

- Select "Regression" from the "Analyze" menu (Fig. 10.13).

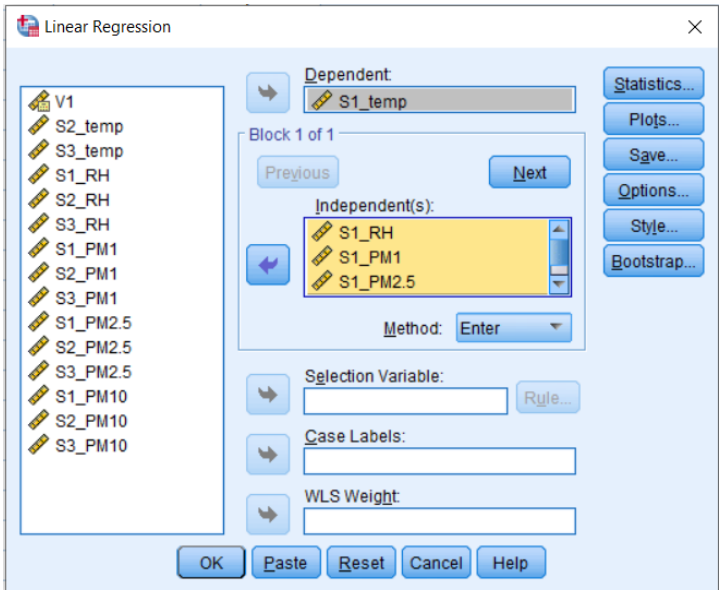


***Figure 10.13**. SPSS Linear Regression menu*

- Select the relevant variables.
- Click the "OK" button.
- The program will understand the relationship between the variables and allow the making of future predictions.

Table 6. *Regression Analysis sample for the relation between temperature and related values*

ANOVA[a]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	383605,314	4	95901,329	14766,635	,000[b]
	Residual	259096,509	39895	6,494		
	Total	642701,824	39899			
a. Dependent Variable: S1_temp						
b. Predictors: (Constant), S1_PM10, S1_RH, S1_PM1, S1_PM2.5						

ANOVA[a]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	483670,045	4	120917,511	23113,715	,000[b]
	Residual	208707,432	39895	5,231		
	Total	692377,477	39899			
a. Dependent Variable: S2_temp						
b. Predictors: (Constant), S2_PM10, S2_RH, S2_PM1, S2_PM2.5						

ANOVA[a]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	211562,893	4	52890,723	8618,332	,000[b]
	Residual	244835,694	39895	6,137		
	Total	456398,587	39899			
a. Dependent Variable: S3_temp						
b. Predictors: (Constant), S3_PM10, S3_RH, S3_PM1, S3_PM2.5						

Examining the ANOVA tables determines which predictors significantly affect the dependent variables and how strong this effect is. Here, ANOVA analysis is performed, and if the significance level of the values (sig.) is less than 0.05, it can be said that the regression model is significant for each data set. Accordingly, the 1st table is associated with S1_temp, the 2nd table with S2_temp, and the 3rd table with S3_temp data. Those tables show the results of the Analysis of Variance (ANOVA) for three different models, each with a dependent variable, the temperature, and four predictors of relative humidity and particulate matter.

Analysing 1st table, the regression section (383605,314) is much larger than the residual section (259096,509), indicating that it significantly affects predictors S1_temp. Also, the F-value (14766.635) is much larger than the critical value, indicating that the regression model is significant.

Similarly, the 2nd table shows that the regression model is significant for S2_temp, as the F-value (23113,715) is much larger than the critical value.

Finally, the third table shows that the regression model is significant for S3_temp, but to a lower degree than the other two models, as the F-value (8618,332) is still significant but not as high as the value of the other two models.

10.3.6. Correlation Analysis

Correlation analysis is a statistical method used to measure the relationship between two variables. This analysis shows the effect of one variable on another and is used to calculate a correlation coefficient that measures this effect.

The following steps are followed in the SPSS program to perform correlation analysis:

- Open SPSS and load the data.
- Select "Correlations" from the Data menu (Fig. 10.14).

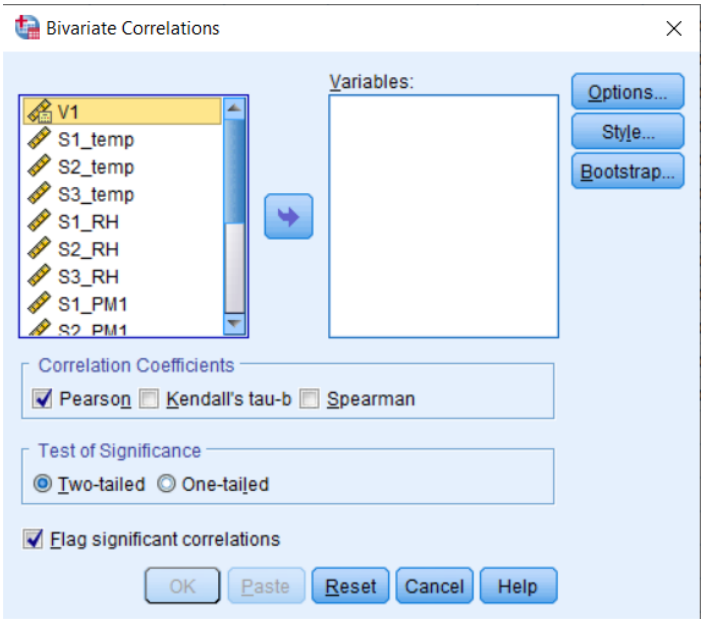


***Figure 10.14**. SPSS Correlation menu*

- Select the variables you want to perform correlation analysis on the screen that appears.
- When the "Pearson" correlation coefficient is selected in the analysis type option, the relationship measures the linear relationship between the variables.
- Examine the chart to determine which variable is most closely related to which other variable from the test results.
- Check the p-value to determine if the relationship is statistically significant. For example, $p < 0.05$ is considered statistically significant.

Based on the results, it can be determined whether the variables have a positive or negative correlation. In a positive correlation, the variables move together, while in a negative correlation, one variable increases while the other decreases.

Correlation analysis is a powerful statistical tool to evaluate the relationship between two variables. The SPSS program offers an easy and user-friendly interface to perform this analysis. By following these steps, correlation analysis can be done with SPSS, and the results can be interpreted.

Table 7 shows the results of the correlation analysis. The table shows the correlation of temperature, humidity, and various types of particles (PM1, PM2.5, and PM10) measured by different sensors. The values in the table were calculated using the Pearson correlation coefficient. Negative values in the table show an inverse relationship between the two variables, while positive values show a directly proportional relationship. In addition, high correlation coefficients (0.965** – 0.989**) were found between PM1, PM2.5, and PM10.

Table 7. Correlation analysis for the data

Correlations												
	S1_RH	S2_RH	S3_RH	S1_PM1	S2_PM1	S3_PM1	S1_PM2.5	S2_PM2.5	S3_PM2.5	S1_PM10	S2_PM10	S3_PM10
S1_temp	- .770**	- .695**	- .142**	- .085**	0.00	.058**	- .121**	- .037**	.047**	- .117**	- .025**	.058**
S2_temp	- .733**	- .833**	- .199**	- .181**	- .122**	.051**	- .210**	- .154**	.046**	- .206**	- .141**	.056**
S3_temp	- .400**	- .372**	- .675**	- .041**	0.01	0.00	- .051**	0.00	- .013**	- .047**	0.01	0.00
S1_PM1	.177**	.249**	.069**	1.00	.649**	.093**	.976**	.638**	.076**	.965**	.607**	.062**
S2_PM1	.081**	.196**	.058**	.649**	1.00	.102**	.626**	.975**	.079**	.606**	.947**	.066**
S3_PM1	0.00	- .011*	.116**	.093**	.102**	1.00	.085**	.095**	.977**	.083**	.086**	.967**
S1_PM2.5	.213**	.274**	.068**	.976**	.626**	.085**	1.00	.625**	.073**	.989**	.600**	.060**
S2_PM2.5	.116**	.224**	.055**	.638**	.975**	.095**	.625**	1.00	.078**	.609**	.980**	.065**
S3_PM2.5	0.01	-0.01	.128**	.076**	.079**	.977**	.073**	.078**	1.00	.073**	.073**	.992**
S1_PM10	.204**	.266**	.055**	.965**	.606**	.083**	.989**	.609**	.073**	1.00	.593**	.062**
S2_PM10	.111**	.214**	.038**	.607**	.947**	.086**	.600**	.980**	.073**	.593**	1.00	.064**
S3_PM10	-0.01	- .024**	.118**	.062**	.066**	.967**	.060**	.065**	.992**	.062**	.064**	1.00

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

When this correlation table is analysed, the relationships between temperature, relative humidity, and PM can be read. The values in the table represent the strength and direction of the relationship between each variable pair. Looking at the table, it is seen that there is a negative correlation between temperature and relative humidity, and there is also a negative correlation between temperature and PM concentrations. In addition, there is a strong positive correlation between PM measurements. This positive correlation between different PM measurements suggests that they may be affected by similar sources or processes in the environment.

10.3.7. SPSS Extensions

SPSS Software offers many methods that can be used to analyse the data and interpret the results. In addition, different analysis methods can be added from the "Extensions" menu of the program. These add-ons can be, for example, "GeoMap," which enables data to be visualized on maps, or "Modeler," which enables data to be analysed by machine learning methods. SPSS Modeler is a software tool for data mining and analysis. SPSS Modeler can be used to build prediction models. Time series forecasting is one of these models. Time series forecasting is a method for predicting future trends using historical data.

A step-by-step action plan for time series forecasting is as follows:

- Preparation of Data: The data must be prepared for time series forecasting. These data are data collected over past periods.
- Analysis of Data: Once the data is ready, a time series dataset is created in SPSS Modeler to analyze the data.
- Establishing the Forecast Model: In the SPSS Modeler, many forecasting model creation tools are available for time series forecasting. These tools contain many algorithms to be used to predict future trends.
- Making Predictions: Future trends can be predicted after the model has been validated.
- Measuring the reliability of the prediction results: Analyzing the prediction results is important to measure the accuracy and reliability of the model.

By following these steps, time series forecasting can be done with SPSS Modeler (Fig. 10.15 and 10.16).

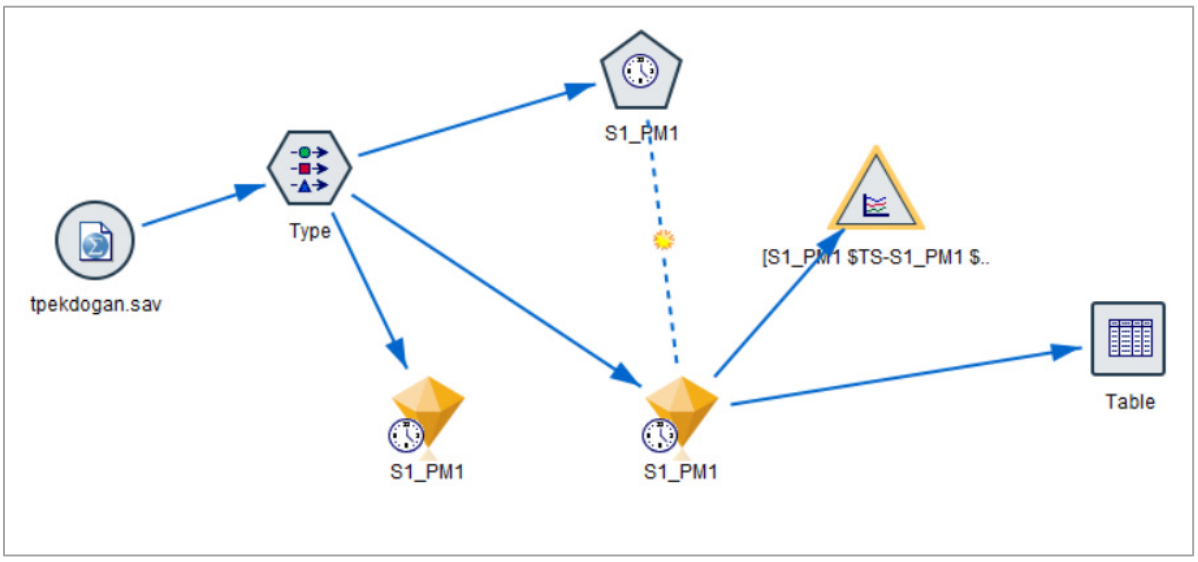


***Figure 10.15**. Visualization of the SPSS Modeler stream diagram*

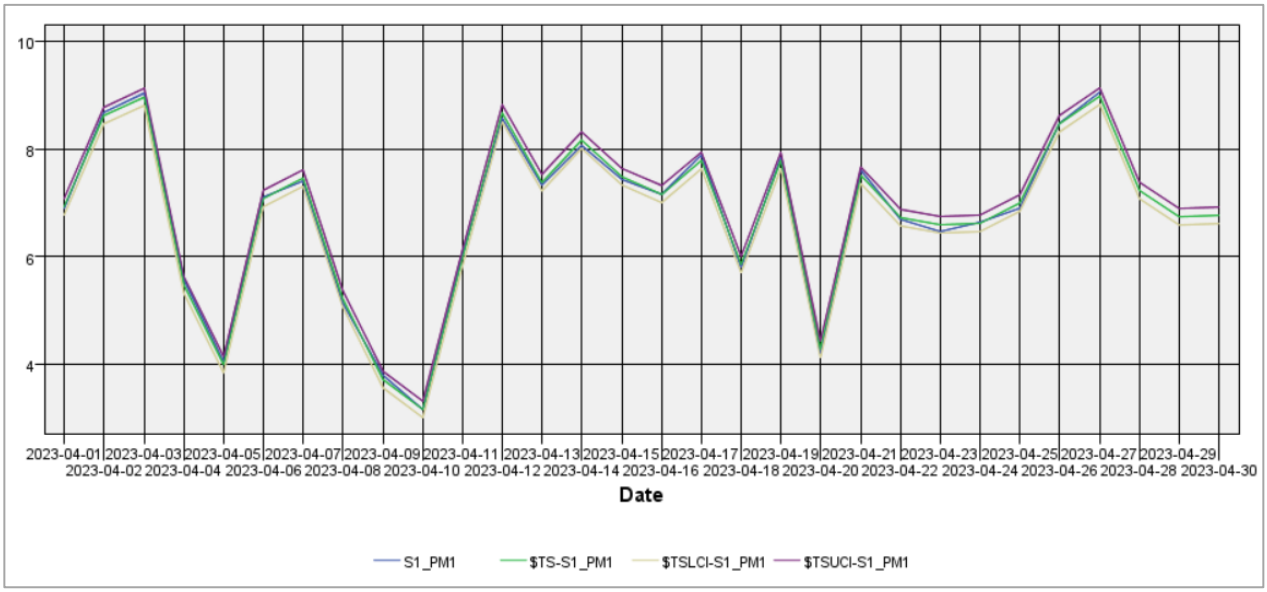


***Figure 10.16**. Time series forecast in IBM SPSS Modeler*

It finds predicted UCL and LCL values when asked to predict three days after this graph. UCL represents the upper control limit in the chart, and LCL represents the lower control limit. Compared to the values taken three days later, the average value of 7.14 PM1 for April 30 is the measured value, while the predicted value is 6.770, the LCL value is 6.616, and the UCL value is 6.924. It is seen that the difference between them is 5.5% on average.

10.4. Conclusions

In this chapter we analysed the measured data made in the project 2021-1-RO01-KA220-HED-000030286, "Application of some advanced technologies in teaching and research on air pollution," with the SPSS statistical program. This study provides a comprehensive discussion on the analysis of data collected from different sensors measuring temperature, humidity, and particulate matter levels. The analysis includes statistical methods such as descriptive statistics, data visualization, regression analysis, correlation analysis, and time series estimation using SPSS software.

Descriptive statistics in this report provide a better understanding of the characteristics of the data. The mean, median, standard deviation and quartile values were calculated at this stage. For example, the S2 sensor measured higher PM1 and PM10 values than the PM measurements, while the S1 sensor measured higher PM2.5 values.

During the data visualization phase, different graphs, such as bar graphs, scatter graphs, and box plots, were created with SPSS software.

Regression analysis is another method to determine the relationship between two or more variables.

Another analysis method is correlation analysis. Correlation analysis is used to measure the relationship between two variables.

For the final analysis the SPSS Modeler extension was used and was performed with another program installation. This program works as an extension of the SPSS program; many analyses and visualizations can be made from there. Time series forecasting is a method for predicting future trends using historical data.

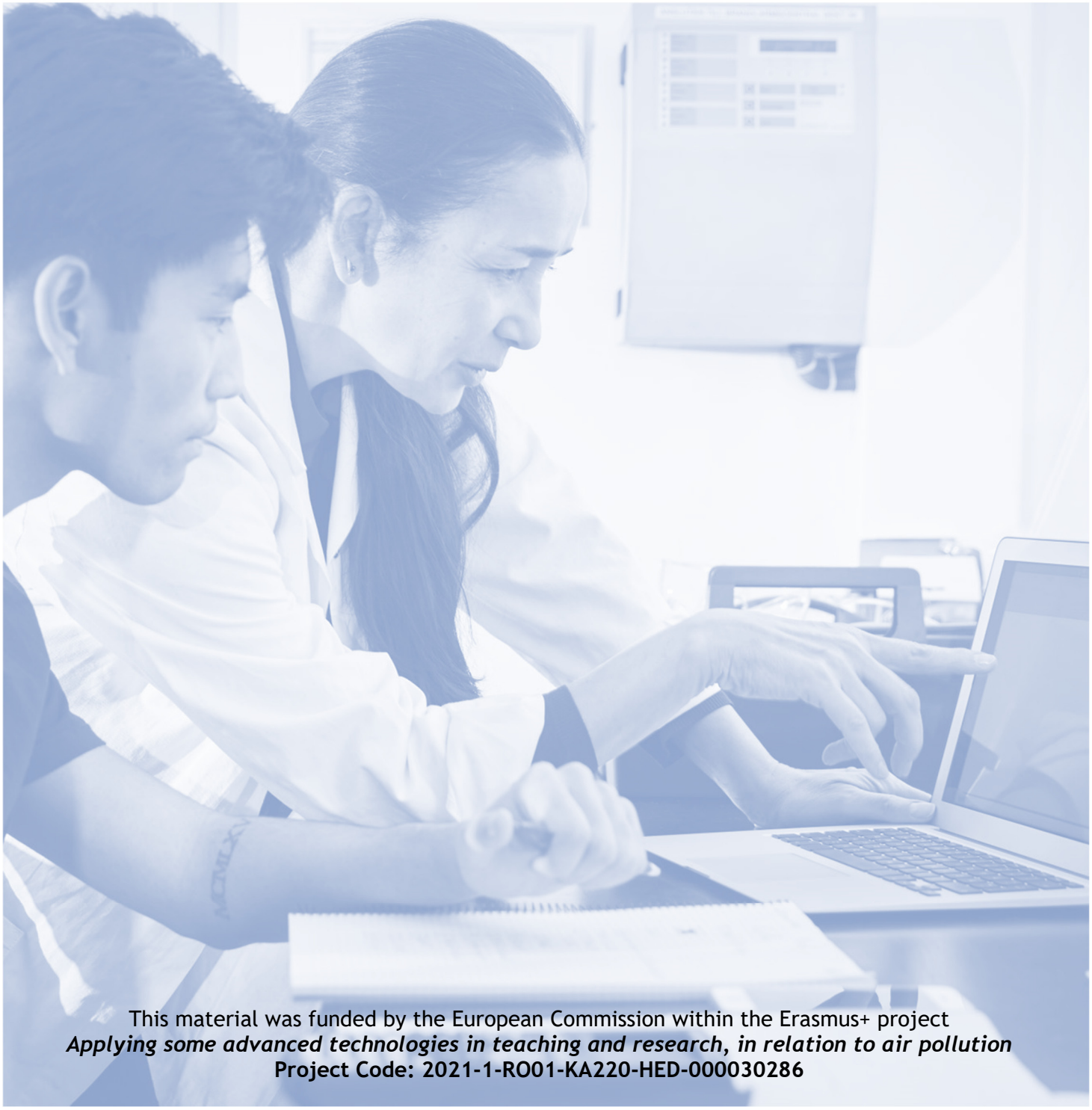
This chapter highlights the importance of monitoring and analysing air quality beyond monitoring. Analysing sensor data can provide important insights into factors affecting air

quality. Especially for comparing two data or predicting the future, SPSS software offers various methods to analyse the data and interpret the results. Following the steps outlined in this report makes it possible to understand data better and make informed decisions about air quality.

References

1. Schell, M.B.; Turner, S.C.; Shim, R.O. Application of CO2-Based Demand-Controlled Ventilation Using ASHRAE Standard 62: Optimizing Energy Use and Ventilation. ASHRAE Trans 1998, 104, 1213–1225.
2. Qi, M.W.; Li, X.F.; Weschler, L.B.; Sundell, J. CO2 Generation Rate in Chinese People. Indoor Air 2014, 24, 559–566.
3. Pekdogan, T. Experimental and Numerical Investigation of a Heat Recovery Ventilation Unit with Phase Change Material for Building Facades. 2021.
4. Bas, E. Indoor Air Quality: A Guide for Facility Managers.; 2004; ISBN 088173327X.
5. EU, EU Directive 2018/844/EU. Energy Performance of Buildings. Official Journal of the European Union 2018.
6. Anderson, J.O.; Thundiyil, J.G.; Stolbach, A. Clearing the Air: A Review of the Effects of Particulate Matter Air Pollution on Human Health. Journal of Medical Toxicology 2012, 8.
7. The death rate due to air pollution worldwide from 1990 to 2019.
8. Environmental Protection Agency. (n.d.). Criteria Air Pollutants. EPA. https://www.epa.gov/criteria-air-pollutants
9. European Council Directive 2008/50/EC of the European Parliament and of the Council of May 21, 2008 on ambient air quality and cleaner air for Europe. Off J 152:0044.
10. Ambient air quality standards (GB 3095-2012) [Review of Ambient air quality standards]. National Standard of the People's Republic of China.
11. Ministry of Environment, Forest, and Climate Change (Director). (n.d.). CPCB | Central Pollution Control Board. CPCB. https://cpcb.nic.in/
12. The World Air Quality Index Project, Integration of Education, No. 4, 2012, ISSN 2308-1058
13. Pekdoğan, T., Udristioiu, M. T., Yildizhan, H. and Ameen, A, From Local Issues to Global Impact: Evidence of Air Pollution for Romania and Turkey. Available at SSRN: https://ssrn.com/abstract=4569124 or http://dx.doi.org/10.2139/ssrn.4569124

AdvTech
AirPollution



This material was funded by the European Commission within the Erasmus+ project
Applying some advanced technologies in teaching and research, in relation to air pollution
Project Code: 2021-1-RO01-KA220-HED-000030286

Funded by the European Union

University of Craiova

Paisiy Hilendarski University of Plovdiv

Adana Türkeş Alparslan University of Science and Technology

Matej Bel University, Banská Bystrica