Stefan Ertl, BSc

Development and Application of a LoRa-based Particle Measurement Platform

Master's Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme:
Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor
Univ.-Prof. Mag.rer.nat. Dr.rer.nat. Alexander Bergmann

Institute of Electrical Measurement and Sensor Systems

Co-Supervisor
Dipl.-Ing. Markus Knoll, BSc

Graz, April 2021

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date	Signature

Abstract

In the past few decades, air pollution has become a severe problem for humans and the environment, resulting in millions of premature deaths. In addition to living organisms including vegetation, air pollutants such as particulate matter can also affect materials. Therefore, the SensMat project aims to develop preventive solutions for museums to preserve artifacts of cultural heritage. This includes the use of a sensor network in museums to record environmental characteristics. This work presents a wireless sensor node that measures particulate matter along with other environmental quantities and uses the LoRa wireless technology to upload data to a server for further processing. Low-power electronic components, a rechargeable Li-ion battery, and power-saving firmware enable operating times of more than four months at hourly measurements. Using a modern software framework for embedded systems, the microcontroller firmware is developed in C++ and uses object-oriented design patterns for a structured software architecture. Measurements next to a reference station showed that the used optical particle sensor measures the particulate matter concentration of particles with a diameter of $<2.5\,\mu m$ ($PM_{2.5}$) with an accuracy of 56–64 %, while its part-to-part variance is minimal.

Contents

List of Figures

1 Introduction

Air pollution became a severe problem for human beings and the environment in the past few decades. According to the World Health Organization (WHO), only one person out of ten lives in a city that complies with the WHO's air quality guidelines (World Health Organization, 2006). In many cities, the limit values for air pollutants in the ambient air are exceeded many times over. As a result of the high air pollution, every ninth premature death in 2012 was caused by air-pollution-related conditions. About 3 million of those deaths can be attributed solely to outdoor air pollution (World Health Organization, 2016). The European Environment Agency (EEA) has identified the main sources of air pollutants in Europe: transport; residential, commercial, and institutional; energy supply; manufacturing and extractive industry; agriculture and waste. Emissions of those sectors contribute different pollutants to the ambient air. Particulate matter (PM), nitrogen dioxide (NO_2) and ground-level ozone (O_3) are considered to be most harmful to humans' health (European Environment Agency (EEA), 2020). However, air pollution also affects vegetation and the earth's atmosphere. Heavy metals and acid rain can damage crops and lead to forest dieback. A well-known consequence of greenhouse gases and air pollution is the ongoing climate change (Vallero, 2008). To assess the air quality, track the trend and set countermeasures to reduce air pollution, it is necessary to quantify the pollutants in the ambient air.

The traditional way to measure the air quality is to use stationary, highly-accurate, and calibrated equipment to measure various air pollutants at well-chosen locations. This approach is expensive, complex, and inflexible. Such a reference measuring instrument for one measurand can cost from €5000 to €50000. Only specific institutions like government authorities, research institutions, or companies can afford such high-grade equipment (Castell et al., 2017). The needed infrastructure is another disadvantage because many conventional measurement approaches use a power supply line and a wired internet connection. Due to the size, complexity, and high price of the equipment, it is impracticable to measure the air quality at a high spatial resolution (Knoll et al., 2018).

In the last few years, a growing amount of commercial low-cost air-quality sensors were brought to market. Prices for a single air pollution sensor range from 5 to

several hundred euros. That is only a fraction of what a high-quality measuring device costs. Low-cost sensors are small, easy-to-use, and accurate enough for many use cases. However, their accuracy depends strongly on the analyte, and many low-cost sensors have a high cross-sensitivity to other gases or particles or show a high drift of the measured values. These effects can make them less suitable or useless for specific applications. Nevertheless, for applications where the limited accuracy is acceptable, low-cost sensors allow government authorities and research institutions to increase the spatial density of measurement points in cities. Further, individuals and companies can use low-cost sensors to measure indoor air quality in households and offices. The evolution of the Internet of Things (IoT) adds more opportunities to air quality measurement systems (Snyder et al., 2013; Castell et al., 2017). Combining the low-cost sensors with the wireless technologies evolving with the IoT makes it possible to create fully wireless air quality measurement platforms. Indoor sensor nodes often use Wi-Fi, Bluetooth, or ZigBee to report their measured data. For wireless sensor networks distributed over whole cities or large buildings, Low Power Wide Area Networks (LPWANs) suit better. They support wireless ranges of up to several kilometers. Since the sensor nodes are often battery operated, low power consumption for wireless communication is also essential. The three most commonly used LPWAN technologies are LoRa, Sigfox, and NB-IoT. They allow creating wireless sensor networks with hundreds of sensor nodes distributed over a whole city or building (Kumar et al., 2015; Knoll et al., 2018).

This thesis proposes a way to develop a LoRa-based wireless sensor node to report indoor air quality data in real-time. The attached sensors measure particulate matter, temperature, humidity, and ambient pressure. In the first part of the work, low-cost components are selected and evaluated to design the hardware for a small, power-efficient, and easy-to-use device. A stable framework and established software design patterns are the basis for the software to provide a reliable operation for a long time.

The wireless sensor node is used in the course of the SensMat project to monitor the indoor air quality and ambient conditions of museums. The following chapter, Chapter 2, provides more information on the SensMat project.

2 Background

This chapter provides background information about the thesis and the motivation behind it. Since the work was carried out as part of the SensMat project, SensMat will be introduced in the following (Section 2.1). Afterward, the requirements and constraints are presented, which were derived from the project requirements. Finally, a short literature review is given to show state-of-the-art approaches.

2.1 SensMat

SensMat is a project of the European Union's Research and Innovation program Horizon 2020. The name of the project, SensMat, is derived from its full title *Preventive solutions for **Sens**itive **Mat**erials of Cultural Heritage*. The fact sheet of the project describes the project objective as follows:

> "Preventive conservation (PC) has emerged as an important approach for the long-term preservation of sensitive cultural heritage (CH), notably for mobile artefacts, those displayed or stored in harsh environments and for small and medium-sized museums. SensMat aims to develop and implement effective, low cost (<20 – 30€ for basic platform), eco-innovative and user-friendly sensors, models and decision-making tools, as well as recommendations and guidelines[,] to enable prediction and prevention of degradation of artefacts as a function of environmental conditions. SensMat is user-driven (inclusion of 19 museums in the project plus survey of 100 more), and the consortium has solid existing results and a strong capacity to mature the sensors, models and decision-making solutions to [Technology Readiness Level] TRL 7 during the project. Based on multiscale modelling, data management systems, collaborative platforms and sensor communication networks (IoT), museums stakeholders will be informed in real-time of possible dangers to their artefacts, thus reducing degradation risks and costly

> conservation treatment. Demonstration of the platform in 10 representative case studies in museums, historical buildings, storage sites and workshops will prepare rapid uptake after the project. Knowledge transfer, training, and recommendations of best practices will facilitate standardization, strategy implementation, new policy definition, and wide-scale adoption of the new solution by cultural heritage sites immediately after the project." (European Comission, 2020)

One of the 17 participating institutions working on the SensMat project is the *Institute of Electrical Measurement and Sensor Systems (EMS)*. It is responsible for developing a wireless sensor node that measures particulate matter. Because the sensor node is only a small part of a sensor network and needs to be integrated into that network, several requirements must be satisfied to ensure interoperability.

2.2 Requirements and constraints

The key requirements to enable a seamless integration of the sensor node into the museum environment and the wireless sensor network established with the SensMat project are explained in the following.

Operational environment

The sensor node is intended to operate indoors, mainly in museums and storage rooms. Thus, weatherproof housing is not needed, and the sensor node must not withstand harsh outdoor weather conditions. To guarantee a compact and solid mechanical design and to avoid touching and possible damage to the electrical components, a suitable housing is required to protect all components of the sensor node.

Particulate matter measurement

The measurement of particulate matter concentration is an essential requirement of the sensor node. The particle sensor must have a measurement range capable of measuring the level of air pollution that may occur indoors.

Low-cost components

For the development of low-cost sensor nodes, the selection of suitable low-cost components is of significance. Reducing the sensor nodes' costs allows them to be deployed in a wide range of museums that can afford the costs.

Wireless communication

All sensor nodes in the network have to send their measurement data to internet-connected gateways, which upload the data to a cloud server. A communication protocol for the wireless communication between sensor nodes and gateways was developed in an earlier project phase. Every sensor node needs to implement this protocol according to the specification in order to communicate with the gateways. As wireless technology, the LoRa technology is used, which is described in Section 3.1.

Size

The sensor nodes must be as small as possible to be inconspicuous within a museum. The small size makes them handy and easy to place.

Battery operation

Battery operation is an important requirement because power lines are not available everywhere in a museum or storage room. Wireless sensor nodes can be placed anywhere, as long as it is within the gateway's range. A sensor node should operate for several weeks or months until the battery needs to be recharged or replaced.

Offline data logging

A micro SD card slot is required to allow the logging of measurement data. This enables the sensor node to be used in environments where no gateway is present to upload data. Another advantage of adding a non-volatile data storage to the sensor node is that it can log events and errors to facilitate troubleshooting in the application software.

Configuration interface

In order to configure, test, and control the sensor node with a PC, the sensor node requires a USB interface. Via this interface, arbitrary functions can be implemented. For example, setting measurement parameters or configuring the real-time clock (RTC). Without such an interface, it would be necessary to change, recompile and flash the software for every changed parameter or setting.

2.3 Literature review

This section takes a look at existing approaches to develop a wireless particle measurement platform with similar requirements compared to those described in Section 2.2.

One approach that uses low-cost sensors and LoRaWAN to monitor the air quality on a university campus is presented by Park et al. (2019). Therefore, they use a star-of-stars network topology which is typical for LoRaWAN. Whereas the LoRa specification only specifies the physical layer, LoRaWAN is a networking protocol designed to connect wireless devices to the internet (LoRa Alliance, 2021). Park et al. (2019) use NUCLEO-L073RZ development boards (STMicroelectronics, Geneva, Switzerland) with attached SX1272MB2DAS LoRa modules (Semtech, Camarillo, CA, USA) as wireless end devices. The low-cost particulate matter sensor PMS5003T (Plantower, Beijing, China) is connected to the development board to measure particulate matter, temperature, and humidity. Battery packs with a capacity of 10 200 mAh provide the necessary energy for the end devices, which makes them completely wireless. Several end devices send their measurement data to internet-connected gateways, which upload the data to a cloud server. By using LoRaWAN as the communication protocol, a single gateway can handle data from several hundred end devices. This is only possible if the gateway uses a high-performance chip, which can handle several LoRa packets on different frequencies simultaneously. In summary, Park et al. (2019) have shown a way to build a reliable air quality monitoring network with hundreds of sensor nodes. The development time for the described end nodes is minimal because they used off-the-shelf components and the well-established LoRaWAN protocol. However, the minimal hardware design effort is reflected in the size of the end devices. They are quite large and therefore not well suited for installations in museums. Another downside of the setup is the used particulate matter sensor PMS5003T, which does not accurately distinguish between PM_1, $PM_{2.5}$, and PM_{10} size fractions, according

to Kuula et al. (2020). Even if it is important to keep costs low, the accuracy of the sensors should still be reasonably high in order to draw reliable conclusions from the measured values.

Liu et al. (2016) propose a very similar air quality monitoring system to the one from Park et al. (2019). The main advantage is the photovoltaic cell, which makes the sensor node energy self-sufficient. Unfortunately, this approach only works outdoors in direct sunlight. Indoors, the light intensity is too low to power a sensor node with a photovoltaic cell.

For a slightly different application, Wu et al. (2018) present a wearable IoT sensor node to monitor harmful environmental conditions. Their sensor node also uses LoRa as wireless technology to send measurement data to an internet-connected gateway. They chose LoRa among several other wireless technologies because it has the highest receiver sensitivity and low power consumption. The sensor node's main components are the battery, a high-efficiency buck-boost converter, a microcontroller unit (MCU), a commercial LoRa module, a load switch for the sensor's power supply, and several sensors. They paid attention to choose only components with low current consumption and high efficiency while also keeping the costs low. Combining the hardware with a software algorithm to further optimize the power consumption could lower the current consumption down to 24.71 µA in sleep mode and a maximum of 33 mA in transmit mode. The connected sensors can measure temperature, relative humidity, pressure, carbon monoxide, carbon dioxide, and ultraviolet light. Although the presented sensor node does not include a particulate matter sensor, it satisfies most of the requirements described in Section 2.2. A disadvantage of this approach is the limited operation time. Because the sensor node is intended to be worn on the body, it must be as lightweight as possible. Thus, they chose a coin battery as the power source, which is small but has little capacity. Although the sensor node's power consumption is minimal, an operating time of several weeks or months is not achievable with this sensor node.

In addition to research on LoRa-based air quality monitoring systems, several companies offer wireless sensor nodes to monitor the air quality in outdoor or indoor areas. Many sensor nodes use Wi-Fi or Bluetooth to upload their measurement data because it is broadly available in offices or households. However, the range of Wi-Fi or Bluetooth networks is limited, and in larger buildings or outdoor areas, wireless technologies like LoRa are used instead. Companies like (TalkPool AG, 2021), GlobalSat WorldCom Corporation (GlobalSat WorldCom Corporation, 2021) and netvox (Netvox Technology Co., Ltd., 2021) offer commercial particulate matter sensors which use LoRaWAN to upload the measurement data to cloud servers.

The cost of a node ranges from €200 to €600. Most of them require an external power supply, which increases infrastructure costs and limits flexibility. One of a few commercial sensor nodes which are battery-operated is the R72616A (Netvox Technology Co., Ltd., 2021). Nevertheless, none of the examined nodes can be re-programmed with custom firmware to implement the communication protocol used by all sensor nodes of the SensMat project to communicate with the gateways. This is one of the main reasons why commercial sensor nodes cannot be used directly for the SensMat project. Instead, an in-house development allows hardware and software to be tailored to all requirements and is also more cost-effective than using commercial products.

3 Theory

This chapter gives a theoretical introduction to three essential contents of the thesis. The first section is about LoRa, a low-power wireless technology specially developed for wide area networks and the IoT. Section 3.2 provides an overview of selected PM measurement methods. The last section of this chapter presents theoretical background about battery management, with focus on lithium-ion (Li-ion) batteries.

3.1 LoRa technology

LoRa (**Lo**ng **Ra**nge) is a proprietary LPWAN technology from Semtech which gets increasingly popular in IoT applications. In contrast to short-range radio technologies such as Wi-Fi, Bluetooth Low Energy, and IEEE 802.15.4, LoRa can cover long ranges of up to several kilometers in rural areas and hundreds of meters in indoor deployments. The achievable range and reliability of radio links depends strongly on the environment. Indoors, factors such as the room layout, the building material, the interior of a room and the location of the wireless communication devices can affect the quality of the radio link (Bor and Roedig, 2017; Augustin et al., 2016). To still achieve a long communication range, the signal-to-noise ratio (SNR) at the receiver must be as high as possible, which can be achieved by two methods. The first one is to narrow down the bandwidth to reduce the receiver's noise level. Sigfox, for example, takes advantage of this method by using ultra-narrow bands with only 100 Hz bandwidth. The second method is to spread the signal's energy over a wide frequency range to reduce its spectral power density and make it more resilient to jamming and interference (Cattani et al., 2017). LoRa uses the second method and therefore specifies a proprietary modulation scheme which is a derivative of the chirp spread spectrum (CSS) modulation (Semtech Corporation, 2015).

3.1.1 LoRa spread spectrum modulation

CSS was already developed back in the 1940s for military applications and got increasingly popular in the past 20 years because of its low transmission power requirements and its robustness to effects like multipath fading, Doppler, and in-band jamming interferers. To spread the signal's energy over a broad spectrum, CSS modulated signals consist of chirps, which are sinusoidal signals whose frequency continuously increases or decreases over time (Semtech Corporation, 2015).

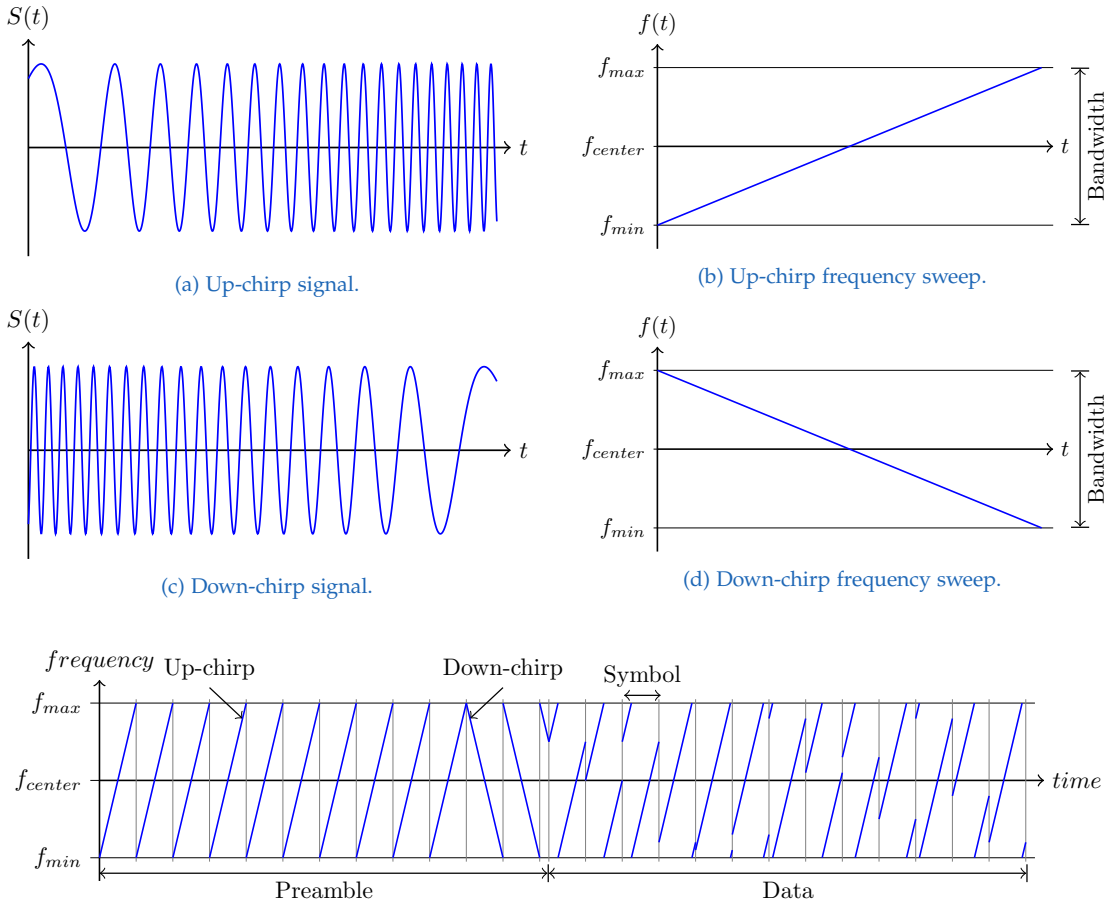
Figure 3.1 shows the two possible chirp variants, up-chirps and down-chirps. The signal bandwidth and the center frequency define the chirps' frequency range. Chipping the data signal and modulating it onto the chirps adds information to the chirps. Figure 3.1e depicts the frequency over time of a LoRa packet. No data is modulated onto the chirps in the preamble segment, which is visible on the uninterrupted chirp from f_{min} to f_{max}. However, in the packet's data segment, the chirps are cyclically shifted to encode data within these frequency shifts. This modulation technique makes it possible to demodulate the signal even if the signal power is up to 20 dB below the noise floor (Cattani et al., 2017; Semtech Corporation, 2015). However, if multiple LoRa devices operate at the same frequency, they can interfere with each other.

Since different applications have different requirements, LoRa offers several parameters to adjust the communication performance for a specific application. The following parameters can be set:

Bandwidth

The bandwidth is the frequency range over which the LoRa chirps are spread. Increasing the bandwidth allows packet transmission with higher data rates and leads to a shorter air time, which in turn increases the energy efficiency. However, higher bandwidth and shorter air time also reduce the receiver sensitivity and thus the robustness and communication range. A lower bandwidth increases the receiver sensitivity but lowers the data rate and requires a more accurate crystal. Typical values for the bandwidth are 125 kHz, 250 kHz and 500 kHz (Bor and Roedig, 2017). With the given bandwidth BW, the chip rate R_C can be computed as:

$$R_C = BW \qquad chips/s. \tag{3.1}$$



(a) Up-chirp signal.

(b) Up-chirp frequency sweep.

(c) Down-chirp signal.

(d) Down-chirp frequency sweep.

(e) The time characteristic of the frequency of a transmitted packet.

Figure 3.1: LoRa packets consist of chirps, which are frequency sweeps with either an increasing (up-chirp) or a decreasing (down-chirp) frequency over time (Augustin et al., 2016)

Spreading factor

To increase the receiver sensitivity, each symbol is spread over many chips. The spreading factor (SF) defines the number of chips representing a symbol as a power of 2. Thus, the symbol-rate R_S can be expressed as:

$$R_S = \frac{R_C}{2^{SF}} = \frac{BW}{2^{SF}} \qquad symbols/s, \tag{3.2}$$

and the modulation bit rate, R_M as:

$$R_M = SF \cdot \frac{BW}{2^{SF}} \qquad bits/s. \tag{3.3}$$

Older LoRa transceivers support spreading factors from 7 to 12, whereas the newest transceivers also allow SF5 and SF6. A lower spreading factor increases the symbol rate and thus increases energy efficiency because of shorter air times. A higher spreading factor instead increases the SNR and thus the sensitivity and range (Cattani et al., 2017; Bor et al., 2016).

Coding rate

For fault-tolerant packet transmission, LoRa uses a forward error correction (FEC) at the bit level. By changing the coding rate (CR), the amount of redundant information within the transmitted data can be adjusted. Possible values for the coding rate are $4/5$, $4/6$, $4/7$ or $4/8$, whereas $4/8$ equals 100% redundancy. Combining the formula for the modulation bit rate R_M and the coding rate CR, the resulting bit rate R_B of LoRa can be computed as:

$$R_B = R_M \cdot \frac{4}{4+CR} = SF \cdot \frac{BW}{2^{SF}} \cdot \frac{4}{4+CR} \tag{3.4}$$

A higher CR offers better resilience to interference bursts but requires higher air time due to the redundancy overhead.

Transmission power

Increasing the transmission power is an easy way to increase the communication range, but it also significantly raises the power consumption. Even if power consumption is not a concern, the transmission power cannot be arbitrarily increased due to legal regulations and hardware limitations. One of these legal requirements stipulates that the maximum radio duty cycle must be limited by 1% for transmission powers above 17 dBm (Cattani et al., 2017).

Carrier frequency

The carrier frequency is the center frequency (f_{center}) as shown in Figure 3.1e which can be adjusted within the sub-GHz range. For example, the LoRa transceiver chip Semtech SX1262 (Semtech Corporation, 2019) supports a frequency range from 150–960 MHz, settable with a resolution of 0.95 Hz when using a 32 MHz crystal. Because a large part of the sub-GHz frequency band is licensed, most LoRa devices and networks use the unlicensed 433 MHz, 868 MHz (Europe) and

915 MHz (North America) industrial, scientific and medical (ISM) radio bands. In Europe, LoRa devices can communicate on ten channels with different bandwidths in the 868 MHz ISM band (Cattani et al., 2017).

3.1.2 LoRa packet format

As Figure 3.2 illustrates, every LoRa packet starts with a configurable number of preamble symbols to synchronize the receiver with the incoming signal. The recorded LoRa packet from Figure 3.1e shows a preamble sequence with $(8 + 4.25)$ preamble symbols, which is the minimum. If the explicit header mode is selected, a header and a cyclic redundancy check (CRC) sum over the header are transmitted after the preamble. The header includes the payload length, the coding rate, and the information whether a CRC is used for the payload. When using the implicit header mode, both the transceiver and the receiver must know in advance the payload length, the coding rate, and whether a payload CRC is sent or not. The following payload includes 0–256 bytes of data, whose integrity can be ensured by using an optional payload CRC (Semtech Corporation, 2019).

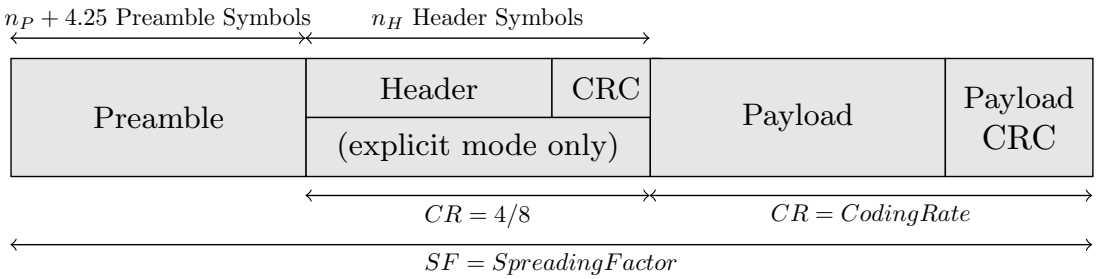


Figure 3.2: The LoRa packet format (Semtech Corporation, 2019).

3.2 Particulate matter measurement

Measurement of particulate matter is essential to understand the effects, characteristics, and behavior of particles, which can affect humans health and our environment. It is also necessary to comply with emission limits set by authorities to guarantee air quality or emission standards (European Environment Agency (EEA), 2017; Giechaskiel et al., 2018). Many methods, instruments and sensors have been developed to measure various characteristics of particles like particle number, mass, or size. These measurement methods use different physical or

chemical effects and, depending on the application, one or the other method is more suitable. Particle number or mass concentration and particle size are the most commonly measured characteristics. The physical methods can be divided into PM size distribution measurement methods and PM concentration measurement methods (Amaral et al., 2015). PM concentration can be measured with gravimetric methods, microbalance methods, electrical charge-based or optical methods (Giechaskiel et al., 2013). A selection of these methods is presented below, focusing on the optical method employed in this work.

3.2.1 Gravimetric methods

In gravimetric analysis, a constant stream of air deposits particles onto a filter. The filter is weighed before and after the sampling period to calculate the particle mass concentration. Weighing the filter mostly needs to be done by hand after the sampling period, limiting the sampling period and the possibility of determining the particle concentration in real-time. An advantage of this approach is that the particles can be analyzed visually and chemically after sampling. This method is often used to measure the particle mass concentration emitted by combustion engine vehicles (Amaral et al., 2015).

3.2.2 Microbalance methods

Instruments based on the microbalance method collect particles on an oscillating element and measure the changing resonance frequency to determine the particle mass. The Tapered Element Oscillation Microbalance (TEOM) uses a sampling filter attached to the tip of a tapered quartz wand. Particles collected by the sampling filter influence the resonance frequency of the quartz. In contrast to TEOM, Quartz Crystal Microbalance (QCM) accumulates particles on a thin quartz crystal resonator through electrostatic precipitation. An increasing number of accumulated particles leads to a decreasing resonance frequency. This effect is used to calculate the particle mass. Microbalance needs a proper decoupling from the environment because vibrations and changing relative humidity or temperature can introduce measurement errors (Giechaskiel et al., 2013).

3.2.3 Optical methods

Optical methods are those methods that use light and optical components to determine the particle concentration. Thereby, a beam of light illuminates particles. The light is partially scattered in all directions when hitting the particles and partially absorbed and transformed into other forms of energy. Adding up *scattering* and *absorption* results in *extinction*. These three principles are the basis for many instruments measuring particle concentration in real-time.

Light *absorption* is used to measure black carbon (BC) because BC strongly absorbs light. Giechaskiel et al. (2013) identified the three most common methods to measure aerosol absorption. The first one computes the absorption from the difference of extinction and scattering. The second method uses a filter to collect particulate matter and measures the light attenuation caused by the particles. The last one is based on the photoacoustic effect, where particles are thermally excited by the incoming light and the resulting pressure wave is measured with an acoustic sensitive device.

An Opacity meter is an instrument using the *extinction* of light to determine the particle concentration. It measures the light transmitted through a specified volume of polluted air. Subsequently, light extinction is calculated by subtracting the portion of transmitted light from the incident light. Nowadays, this method is not well suited to measure PM concentrations as the required sensitivity can not be reached (Giechaskiel et al., 2013).

Light *scattering* is the third optical method used to measure particle concentration. An instrument counting the particle number in a defined aerosol flow is called optical particle counter (OPC). Many low-cost particulate matter sensors are OPCs, also the SPS30 (Sensirion AG, 2020a) (Sensirion AG, Stäfa, Switzerland), which is used in this work. Therefore, the technology behind OPCs is described in more detail below.

Particle counting by means of light-scattering

An OPC needs a constant stream of air passing through a small specified area, which is illuminated by a beam of light. Figure 3.3 shows the basic design and components of an optical particle counter. A coherent and monochromatic laser is used to generate the light beam, whose light is absorbed and scattered as soon as it interacts with particles. A photodetector mounted at a certain angle to the light beam captures a fraction of the light scattered by the particle. Pattern and intensity

of scattered light strongly depend on the ratio between particle size and wavelength of incident light (Giechaskiel et al., 2013). The photodetector converts the incident light into an electrical signal proportional to the light intensity. Every electrical pulse is counted as particles flowing through the illuminated area. This process acts as the basis to count particles. Because only a small part of the particles within the aerosol flow can be counted, the particle number concentration is derived by extrapolating the counted particles.

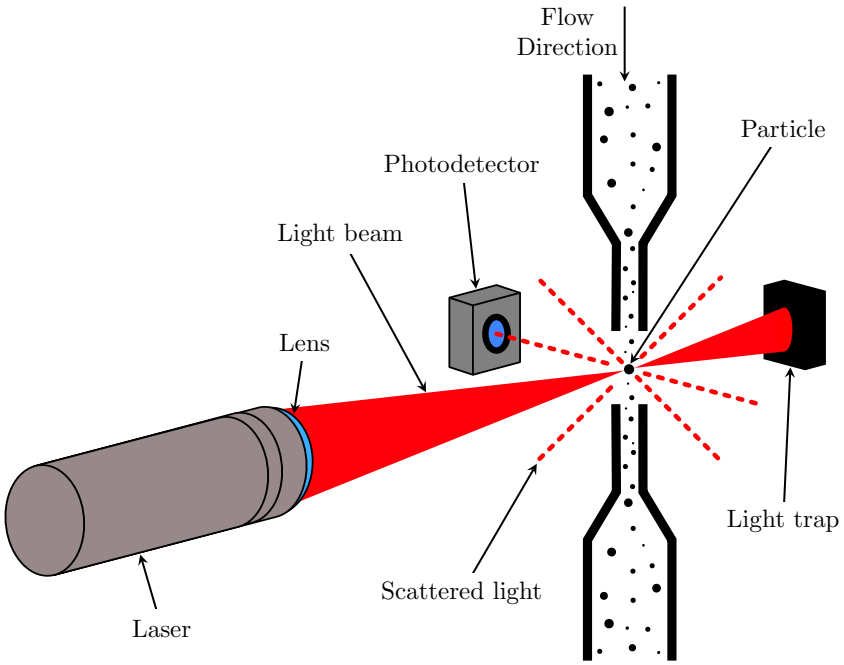


Figure 3.3: Measuring principle of an optical particle counter (Sensirion AG, 2020b).

Merely counting the particles is not enough to compute the particle mass concentration. Therefore, the particle size is needed to estimate the counted particles' mass. With an OPC, particle size can also be estimated because the power of a particle's scattered light is a function of the particle's size, refractive index, and shape (Baron and Willeke, 2005). In practice, the refractive index and the shape of a particle are unknown. Determining the particle size is done by measuring the pulse height from the photodetector's electrical signal and applying a calibration curve. The calibration curve is obtained by measuring the pulse height of spherical particles of known size and morphology. Even after calibration, uncertainty in the refractive index can lead to high variability in the computed particle size distribution (Giechaskiel et al., 2013). The accuracy of OPCs heavily depends on the algorithm converting the photodetector's signal into particle concentration (Sensirion AG, 2020b). OPCs can detect particles with diameters >300 nm (Giechaskiel

et al., 2013). In addition, OPCs support sampling rates down to 1 Hz, making it possible to detect rapid changes in particle concentration (Sensirion AG, 2020a).

3.3 Battery management

Batteries are composed of electrochemical cells utilizing a chemical reaction to generate electrical energy. In rechargeable batteries, the chemical reaction can also be reversed to store electrical energy. The cell type provides information about the materials used to achieve this storage effect. Depending on the cell type, the voltage provided by a single cell varies and changes with the battery's state of charge. Additionally, different cell types require different charging processes to allow a long battery life. Due to the cells' peculiarities, a battery-operated device needs special battery management according to the used cell type.

At this time, Li-ion cells are prevalent and power many portable devices because of their high energy density, high cell voltage (3.6 V), and low self-discharge rate (Horowitz and Hill, 2015). However, due to their chemical composition, Li-ion cells require protection against overcharge, over-discharge, overcurrent, and short circuits; otherwise, thermal runaway and cell damage may occur. Several semiconductor manufacturers offer individual protection integrated circuits (ICs) for this purpose. Most battery packs with one Li-ion cell or several cells in parallel or series connection already have such protection ICs built in (Cadex Electronics Inc., 2018).

Charging Li-ion batteries also requires special care, as unsuitable charging conditions can reduce the capacity and service life of Li-ion batteries. Figure 3.4 shows the battery voltage (V_{bat}) and charge current (I_{chrg}) during a typical charging cycle as performed by many available battery charging ICs. The presented charging technique is known as constant current constant voltage (CC-CV) (Chen and Rincon-Mora, 2006).

If the battery is discharged below the minimum charge voltage V_{min}, modern battery charging circuits start the charging cycle with a pre-charge mode, also called the pre-conditioning phase. Thereby, the battery is charged with a small charge current I_{pre} until the battery reached V_{min}. The pre-conditioning phase is also used to detect faulty batteries. If V_{bat} does not reach V_{min} within a certain amount of time, the battery is considered faulty, and the charging cycle is terminated (Chen and Rincon-Mora, 2006).

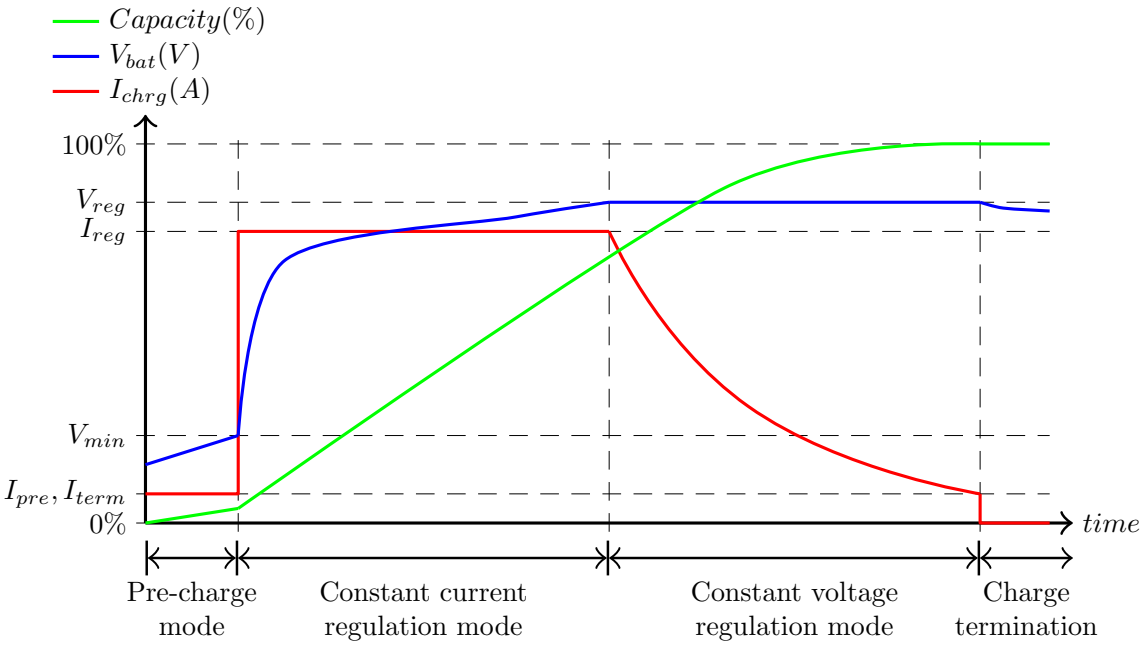


Figure 3.4: Typical charge cycle for a deeply discharged Li-ion cell (Chen and Rincon-Mora, 2006).

Afterward, in constant current regulation mode, the battery is charged with the regulation current I_{reg} derived from the charge rate, also called C-rate. The battery manufacturers provide information about the recommended charge rate, which defines the charge current in relation to the battery's capacity. For instance, if a battery has a capacity of 2 Ah and its recommended charge rate is 0.8 C, the battery should be charged with $I_{reg} = 1.6$ A. Typical charge rates are between 0.5 C and 1 C (Chen and Rincon-Mora, 2006; Analog Devices Inc., 2003).

Once V_{bat} reaches the regulation voltage V_{reg}, battery charging continues in constant voltage regulation mode by applying a constant V_{reg}. As the charge level increases, I_{reg} exponentially decreases until it falls below the termination current I_{term}, which is typically 10 % of I_{reg}. If this happens, the charging cycle is terminated, and the battery has its full capacity. Charge termination is important because overcharging of Li-ion batteries leads to plating of metallic lithium and compromises safety. To prevent thermal runaway, most battery charging ICs have a thermistor input to stop charging if the battery temperature exceeds 50 °C (Chen and Rincon-Mora, 2006; Cadex Electronics Inc., 2018).

Discharging a Li-ion battery with constant current leads to a decrease of the battery voltage V_{bat} over time as illustrated in Figure 3.5. After a rapid drop of V_{bat} at the beginning of the discharge cycle, it decreases linearly for a majority of the

time. Towards the end of capacity, V_{bat} again quickly decreases, which needs to be limited at a certain cut-off voltage V_{cut} to avoid over-discharging the battery. Therefore, Li-ion battery protection ICs have a cut-off function to disconnect the cells from the load if V_{bat} falls below V_{cut}.

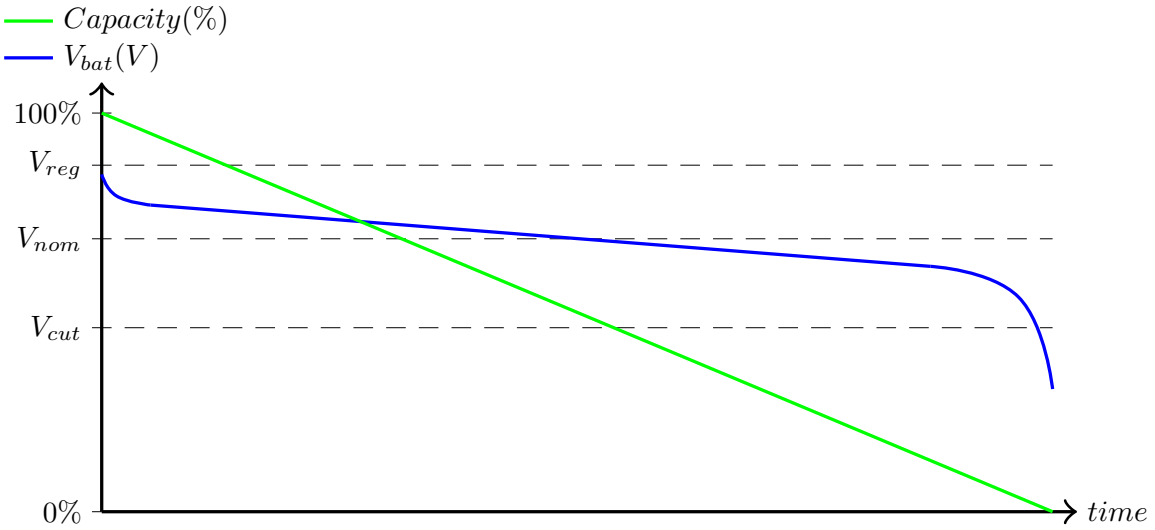


Figure 3.5: Discharge cycle of a Li-ion battery when a constant current is drawn (Chen and Rincon-Mora, 2006).

Although Li-ion cells are generally difficult to handle, it is possible to operate them safely by following standard rules for charging and discharging and using appropriate protection circuits to avoid life-limiting or damaging effects.

4 Sensor System Design

In this work, a wireless sensor node was developed based on the requirements described in section 2.2. This involved selecting suitable components, designing a printed circuit board (PCB), implementing microcontroller firmware, and designing the mechanical construction. Figure 4.1 depicts the developed sensor node.

Figure 4.1: The fully assembled sensor node with the lid open.

4.1 Electronics design

4.1.1 System overview

The block diagram in Figure 4.2 shows the main components of the sensor node, which are divided into three sections around the core, the microcontroller. The microcontroller runs the firmware to interact with the sensors and provide the required functionality for the interfaces. The power supply consists of a rechargeable battery, a battery charging circuit, and voltage regulation for the used electronic components. The sensor node integrates two different sensors for measuring particulate matter and environmental quantities, which are both handled by the microcontroller. Several interfaces connected to the microcontroller provide functionality for data storage, wireless communication, and human interaction.

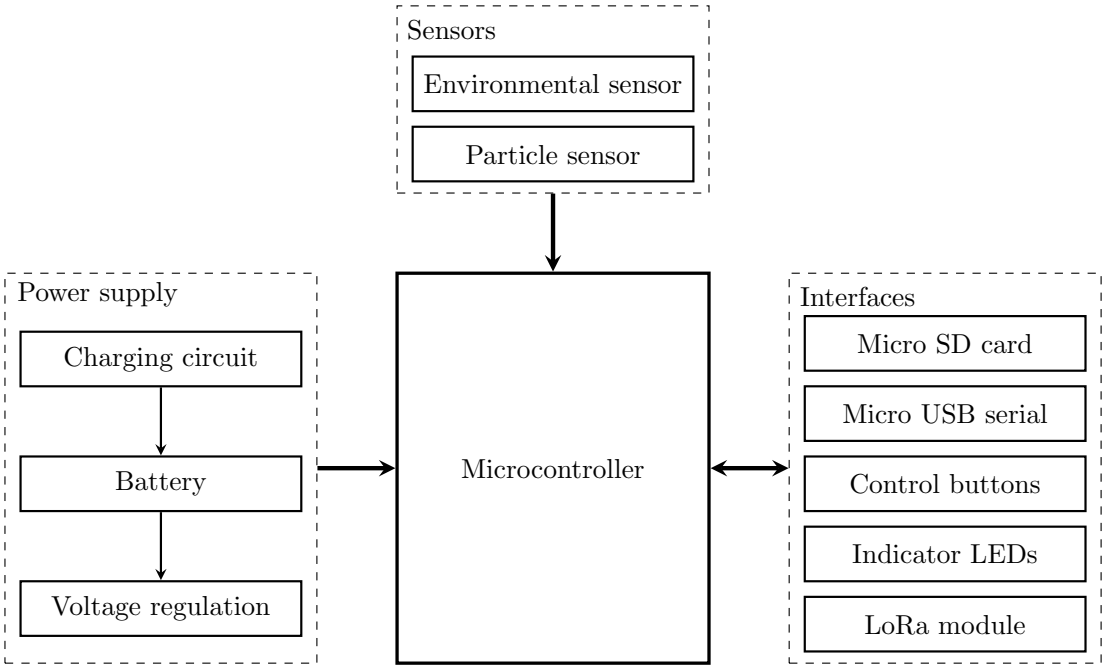


Figure 4.2: Overview of the sensor node's components and the internal connections.

4.1.2 Components

When selecting the electronic components for the sensor node, special care was taken to use low-power components to maximize operation time. As far as possible,

surface mount components were used to minimize PCB size and ease PCB production. Using a solder paste and a pick and place machine, the PCB production could be significantly accelerated. For easier reworking and to allow hand-soldering of the components, mostly components with leads were chosen. The components, their properties, and their selection are described in more detail below.

Battery

To select the right battery for the sensor node, it is necessary to know the required capacity, the battery voltage, and the maximum discharge current. Since the maximum current drawn is relatively low (<200 mA), this is not the most crucial limitation. Furthermore, the nominal battery voltage must be several volts higher than the maximum supply voltage required for the components if no step-up voltage regulator is used. Almost all components of the sensor node require 3.3 V as supply voltage, only the particle sensor requires 5 V. Thus, the battery must provide at least slightly more than 5 V to utilize the entire battery capacity, regardless of the battery state of charge. The required battery capacity must be determined by estimating the power consumption of the components, as described in a later section (Section 4.1.3).

Another important decision concerns the battery chemistry used. For portable devices, nickel-metal hydride (Ni-MH) or Li-ion batteries are most commonly used. Ni-MH batteries are best suited for applications where high currents are drawn. They are easy to charge and do not require special safety mechanisms because their chemistry is quite stable. The disadvantages of Ni-MH batteries are their high self-discharge rate and low power density compared to Li-ion batteries (Horowitz and Hill, 2015). For this reason, most portable devices with limited space and low power consumption use Li-ion batteries as power supply. Hence, the sensor node also uses a Li-ion battery pack. It contains two series-connected Li-ion cells in the widely used 18650 format, resulting in a nominal battery voltage of 7.2 V, sufficient to provide the required 5 V. The battery pack has a built-in safety board that protects the cells from over-voltage, under-voltage, over-current, and short circuit. Therefore, no additional protection circuitry is required for the battery.

Voltage regulation

Voltage regulation is required to regulate the battery voltage down from 7.2 V to the 3.3 V and 5 V required by the electronic components and sensors. The efficiency of

a voltage regulator has a major impact on battery life. Linear voltage regulators are the least efficient voltage regulators because they dissipate the voltage difference between input and output by converting it to heat. This is only acceptable if the voltage difference is marginal. Switching regulators dissipate much less energy than linear regulators. They regulate voltage by rapidly switching the voltage, which is known as pulse width modulation. A capacitor and inductor are used as an energy buffer to provide a constant voltage at any current drawn.

Two switching voltage regulators are used for the sensor node, providing the required 3.3 V and 5 V. Their inputs are both connected to the battery voltage. The TPS6205x series (Texas Instruments, 2015) voltage regulators have a suitable input voltage range of 2.7–10 V and provide a maximum of 800 mA, which is sufficient for this application. In addition, the quiescent current is 12 µA and the efficiency can reach 95 %, which minimizes power dissipation. The TPS62051, used for the 3.3 V supply, is an enhanced version of the TPS62050 and provides low-battery functionality that shuts down the regulator when the input voltage drops below a certain threshold. This shutdown voltage and the reactivation voltage can be adjusted with four external resistors.

The 5 V regulator powers the particle sensor. Via an enable pin, the microcontroller activates the voltage regulator only when the particle sensor is needed. This saves additional power, although the particle sensor consumes little power in sleep mode.

Battery charging circuit

The sensor node has an integrated battery charging circuit, so the battery does not need to be disconnected and removed for charging. Instead, a power adapter is connected to the sensor node to charge the Li-ion battery. The chosen battery charging IC is the LTC4002 (Analog Devices Inc., 2003). This is a standalone Li-ion switch mode battery charger, meaning it does not dissipate the voltage difference between the input and battery voltage, as it would be the case with a linear battery charger. The LTC4002-8.4 charges two Li-ion cells connected in series using the CC-CV charging method presented in section 3.3. Furthermore, it supports input voltages from 8.9 to 22 V and the maximum charging current was set to 1 A with an external sense resistor. In addition to end-of-charge termination, which measures the charge current to detect whether the battery is fully charged, the IC also has a charge timer that terminates charging after three hours. To protect the battery from overheating, a thermistor is connected to the charging IC to stop charging when the temperature exceeds 50 °C. Compared to other switching battery chargers, the

LTC4002 requires only a few external components and comes in an 8-lead package, minimizing the PCB area required for the battery charging circuit.

Microcontroller

The sensor node's firmware runs on the ultra-low-power STM32L476RG (STMicroelectronics, 2019) microcontroller with 1 MB flash memory and 128 kB SRAM, enough to run a real-time operating system (RTOS) on it. An Arm 32-bit Cortex-M4 CPU with integrated floating point unit (FPU) is the core of the microcontroller. With a frequency of up to 80 MHz, the MCU is fast enough to drive the connected peripherals. To simplify development around the STM32L476RG, a NUCLEO-L476RG development board is available, helping to prototype applications quickly around this MCU.

The microcontroller has several built-in functions, such as the RTC, which is used for logging the time of log entries for events and measurements. In addition to the usual interfaces such as Universal Asynchronous Receiver Transmitter (UART), Inter-Integrated Circuit (I^2C) and Serial Peripheral Interface (SPI), the microcontroller also has a USB 2.0 interface to establish a serial connection to a PC. Another useful feature is the hardware CRC unit, which calculates the CRC checksum of given data in hardware, which is much faster than doing it in software. Several analog inputs with 12-bit analog-to-digital converters (ADCs) can be used to measure voltages, for instance, the battery voltage, when using an appropriate voltage divider.

Particle sensor

For the measurement of particulate matter, the SPS30 (Sensirion AG, 2020a) was selected. It is a small ($42 \times 42 \times 12$ mm) particulate matter sensor that is MCERTS-certified by the Environment Agency of the UK (Sira Certification Service, 2020) for $PM_{2.5}$ measurements and particulate matter concentrations below 75 $\mu g/m^3$ (Sira Certification Service, 2020). Evaluations of the sensor confirm that it has low intra-model variability and precision above 92 % for $PM_{2.5}$ measurements, even under various temperature and relative humidity combinations. However, the SPS30 tends to overestimate $PM_{2.5}$ concentrations compared to a reference instrument (South Coast Air Quality Management District, 2019).

With a short cable (100 mm), the SPS30 is connected to the sensor node board and further to the microcontroller. Commands and measured values are exchanged

between the microcontroller and the particle sensor via the UART interface. Although the SPS30 also supports I^2C, UART was chosen because it is more suitable and robust for connecting off-board peripherals.

The SPS30 consumes the most power of the sensor node components due to the integrated fan generating a constant airflow and the laser illuminating the particles. After a brief increased current consumption when the fan starts, it typically consumes about 55 mA according to the datasheet.

Environmental sensor

Apart from a particle sensor, the sensor node contains a BME280 (Bosch Sensortec, 2020) sensor for measuring the temperature, absolute pressure, and relative humidity of the ambient air. Since it comes in a small surface-mounted device (SMD) package, it is soldered directly onto the board and connected to the microcontroller via SPI.

LoRa module

For wireless communication, a separate LoRa module, the LAMBDA62 (RF Solutions, 2021), is soldered onto the PCB. Using a prefabricated module for this application has several advantages compared to using a transceiver only. Using a module eliminates the need to calculate and design the matching network, which requires detailed knowledge and costs time. In addition, the LoRa module selected is already CE-certified for the 868 MHz ISM band, saving certification costs.

Most available LoRa modules include a microcontroller with a LoRaWAN stack, but no LoRaWAN network is used for this work. Therefore, the LAMBDA62 module was chosen because it contains only the SX1262 (Semtech Corporation, 2019) transceiver IC and an antenna matching network. Via SPI and additional control pins, the LoRa module is connected to the microcontroller.

The antenna is attached to the U.FL connector, which is soldered onto the LoRa module. As antenna, a flexible 868/915 MHz ISM standalone antenna (Molex, 2014) is utilized, which is glued on the lid of the sensor node. This saves space by not having to mount a rod antenna on the outside of the box.

User interfaces

Interfaces that are required during operation and for regular maintenance, such as buttons, LEDs, and connectors, are visible and accessible from the outside, as depicted in Figure 4.3.

(a) Front. (b) Back.

Figure 4.3: User interfaces of the sensor node: power switch (1), button to start single measurement (2), button for status indication (3), status LED (4), battery charging LED (5), battery level LEDs (6), micro SD card slot (7), micro USB socket (8) and DC socket.

The power switch on the front of the sensor node is used to turn the sensor on and off by connecting and disconnecting battery power to the voltage regulators. Two pushbuttons, a bi-color status LED, and three green battery LEDs are connected to the digital inputs and outputs of the microcontroller. Pressing the button next to the power switch triggers a single measurement, mainly for testing purposes. Pressing the second button displays the sensor node's current status on the status LED next to the button and the current battery level on the three battery level LEDs. Pressing the right button for more than three seconds triggers its second function, which puts the sensor node into a state where it can be configured using a PC and the USB interface. The battery charging LED is driven by the battery charger and indicates whether the battery is charging when a power supply is connected to the DC jack on the back of the sensor node. A micro SD card slot and a micro USB socket are also accessible on the back. The micro SD card slot has a push-pull mechanism. Measurement data and log files are stored on the inserted micro SD card during operation.

In addition to the interfaces that are accessible from the outside, there are two more interfaces on the board that are only accessible when the lid is open. These interfaces are two pin headers for flashing and debugging the firmware, a serial wire debug (SWD) interface, and a UART interface.

4.1.3 Power consumption estimation

Derived from the requirements, the sensor node is designed to have an operating time of several months without recharging it. To determine the battery capacity required to achieve this operating time, the sensor node's power consumption was estimated using the electrical characteristics from the datasheets of the selected components. Table 4.1 lists the estimated current consumption of each component and its supply voltage. For components that are directly connected to the battery voltage, a nominal battery voltage of 7.2 V is assumed. The average power consumption is calculated as a function of the duty cycle per hour since some components are only in operation for a few seconds per hour. The duty cycle per hour is estimated for four measurements per hour. While the SPS30 sensor is not in use, the 5 V regulator is switched off to save power. The maximum values from the datasheets were taken for the current consumption of the components. Since the voltage regulators have a limited efficiency that depends on the load current, their efficiency is included in the calculated average power consumption. Summing the average power consumption of all components gives a total power consumption of 13.6 mW and an average current consumption of 1.8 mA if a nominal battery voltage of 7.2 V is assumed. With this current draw, a battery capacity of 2643 mAh is required for 60 days of operation. As shown in Table 4.1, the SPS30 sensor clearly consumes the most power, almost 88 %. Thus, the operating time can be significantly increased by increasing the measurement interval. If measurements are taken only once an hour, an operating time of approximately 200 days is obtained with the same battery capacity, which is quite a large gain. Based on the presented calculations, the battery pack was selected, and an approximate operating time could be estimated.

4.2 Mechanical design

In addition to the electronics design, it was necessary to develop a mechanical design for the sensor node that allows convenient handling of the device, protects the electrical components from mechanical impact, and at the same time looks inconspicuous and be as small as possible, since visual aspects are important in cultural applications. Therefore, several available standard housings were investigated to see if they are suitable and if all components fit into the housing. For this purpose, a CAD program was used, and various arrangements of the 3D models of the largest components were tried out. Since the PCB was designed after an enclosure was selected, it was necessary to estimate the PCB size and keep enough

Table 4.1: Power consumption estimation for all major components of the sensor node.

Component	Operation mode	Voltage in V	Current in µA	Voltage regulator efficiency in %	Duty cycle per hour in s	Average power in µW
LTC4002-8.4 Battery charger	Sleep mode	7.2	20	100	3,600	144
TPS62050 5 V regulator	Active mode	7.2	20	100	120.8	4.8
	Shutdown mode	7.2	5	100	3,479.2	34.8
TPS62051 3.3 V regulator	Active mode incl. input voltage divider	7.2	16	100	3600	115.2
BME280 Sensor	Measurement mode	3.3	442	80	0.4	0.2
	Sleep mode	3.3	0.3	75	3,599.6	1.3
SPS30 Sensor	Fan start	5	80,000	91	0.8	97.7
	Measurement mode	5	65,000	91	120	11,904.8
LAMBDA62-8D LoRa module	TX mode with 14 dBm TX power	3.3	90,000	91	8	725.3
	RX mode	3.3	5,300	90	8	43.2
	Standby mode	3.3	0.6	80	3,584	2.5
STM32L476RG Microcontroller	Run mode at 80 MHz	3.3	9,600	90	8	78.2
	Low power sleep mode at 2 MHz	3.3	96	75	3,592	421.5
						13,573.4

space for it in the housing. It was also essential to find an arrangement of the components to not impede the functionality of the particle sensor, meaning that particle inlet and outlet ports have to be properly connected to the environment. Figure 4.4 shows an exploded view of the 3D model with the selected housing and the designed PCB. The SPS30 particle sensor and the battery pack are placed side by side on the bottom of the housing, and on top is the PCB with spacers in between.

The components on the PCB that must be accessible from the outside were placed on the edges of the PCB and are accessible through cutouts in the housing. Based on the 3D model of the sensor node, a technical drawing was created so that the cutouts could be milled in the right place. On one side of the sensor node, LEDs, a power switch, and two buttons were placed, which are needed by the user to control the sensor node. All interface components were placed on the opposite side, namely a micro SD card slot, a micro USB socket, and a DC socket. Cutouts

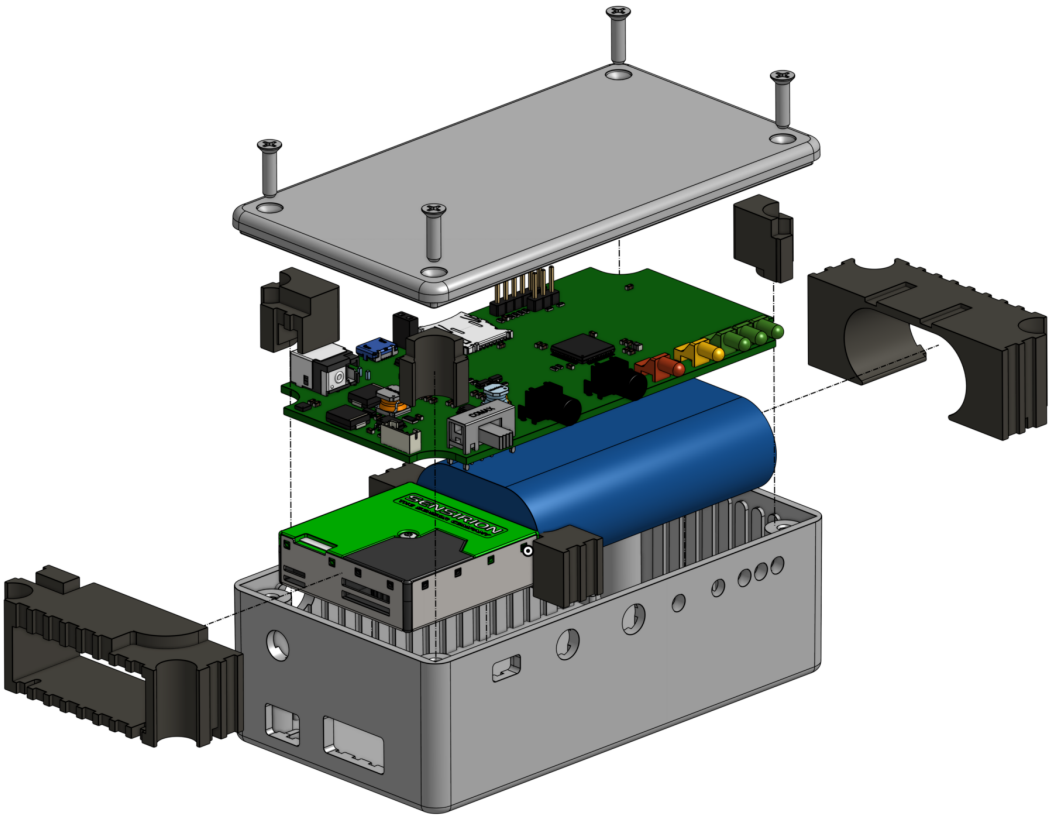


Figure 4.4: 3D model of the sensor node showing the individual components, the housing and the holders for fixing all components.

for the inlet and outlet of the SPS30 sensor were milled on the shorter side, and a hole was drilled above them for the BME280 sensor, which is mounted directly on the PCB behind this hole. The indicator LEDs are SMD parts and soldered directly to the PCB, so their light is basically not visible from the outside. Therefore, light pipes were mounted above them to redirect their emitted light through the holes to be visible to the user.

To hold battery pack, particle sensor, and circuit board in place, holders were designed and 3D printed to fix the components in the housing. Since the selected 1591CSBK housing has vertical rails, it was possible to design holders that allow the sensor node to be assembled without screws, except for those that fix the lid. With the presented assembly, the sensor node has dimensions of $120 \times 65 \times 40\,\text{mm}$ and can be simply placed on horizontal surfaces. For wall mounting, the housing manufacturer also offers flanges to mount the sensor node with screws.

4.3 Software design

To implement the firmware for the microcontroller, a rich framework and existing libraries were used as far as possible, which significantly reduces the development time. In order to create a structured software architecture, several well-known design patterns for object-oriented software development were applied. This helps in principle to make software reusable, maintainable, and open for extensions (Gamma et al., 1995).

4.3.1 Arm Mbed OS

The firmware for the microcontroller was developed based on Mbed OS (Arm Limited, 2021), an open source real-time operating system for Arm Cortex-M-based microcontrollers. It was chosen because it provides good support for the STM32L476RG microcontroller and includes RTOS features, useful libraries and drivers for peripherals. Furthermore, Mbed OS has a good documentation, is actively developed, and is open-source with a large community.

Mbed OS abstracts the hardware to some extent, which makes it easy to develop applications that are portable to a wide range of devices. While it is possible to create applications that use the Mbed OS RTOS, for applications that do not require multithreading support, a bare-metal profile with only the device drivers can also be used. One advantage of the RTOS profile used in this work is the power management, which automatically puts the MCU into the lowest possible power mode.

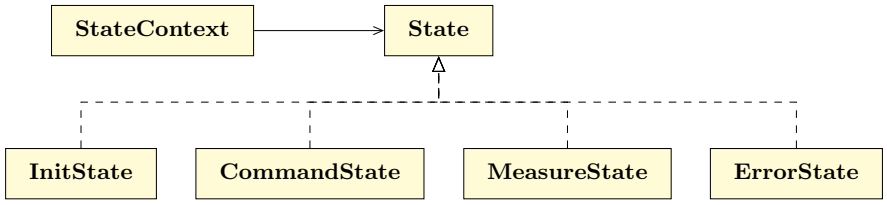
Whereas Mbed OS itself is developed in C++ and C, the applications built on top of it can be developed in pure C++, allowing to apply object-oriented design patterns and writing well-structured code. Common pitfalls with arrays, strings, or pointers in C can also be avoided using the C++ data structures and functions.

Mbed's library structure is very modular, so modules can be included in an application as needed, which avoids including unused source code in the application. To manage the modules, their dependencies, the build toolchain, and the source code, Mbed provides a command-line tool that handles all these tasks. Finally, debugging Mbed applications on the microcontroller can be done with the GNU debugger (GDB) (Free Software Foundation, 2020) and a programmer connected to the SWD interface.

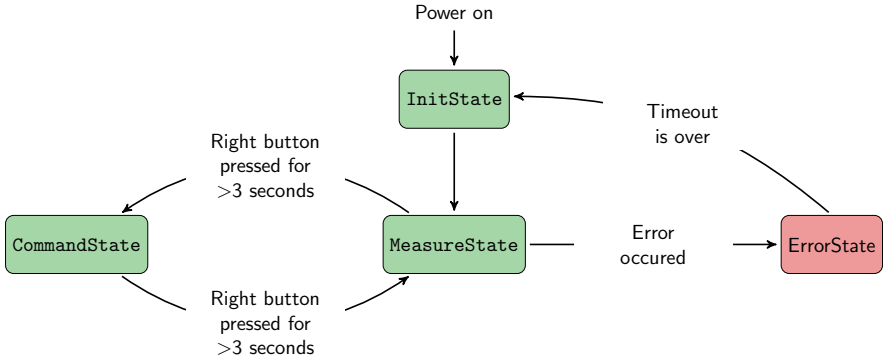
4.3.2 Control flow

The developed sensor node firmware uses two threads, one for LoRa communication (communication thread) and one for all other functionality (main thread). This decoupling of the LoRa communication allows the communication between sensor node and gateway to be handled independently of the measurement task so that measurements are not stopped or paused if no radio connection to a gateway could be established.

The main thread implements a state machine using the state pattern, which is an object-oriented design pattern. The main advantage of this pattern is that an object can change its behavior depending on its internal state. By creating a separate class for each state, the states are independent of each other. Hence, states can be easily inserted, removed, or replaced, and tested independently. Also, state transitions can be changed without modifying the state classes (Gamma et al., 1995). In this work, four concrete states were defined, all implementing the abstract base class *State*, as shown in Figure 4.5a. In the class *StateContext*, the state machine's logic is implemented by defining the state transitions. When a state transition is required, the *StateContext* determines which state is set active next depending on the result of the previous state. Figure 4.5b shows the four states and the possible state transitions.



(a) Class diagram of the state machine.



(b) The four states of the state machine and their transition conditions.

Figure 4.5: The main thread implements a state machine using the state pattern.

InitState

The first state that is active after power-on is the *InitState*, which initializes the SD card driver, the SPS30 sensor, and the BME280 sensor and loads configuration parameters from the flash memory. Depending on whether LoRa communication was enabled in the loaded configuration, the communication thread (see Section 4.3.3) is started or not. Before exiting the state, the restart counter is reset if no error has occurred.

MeasureState

After initialization, the sensor node automatically switches to the *MeasureState*, which handles the measurement process and is active until it is manually switched to another state or an error occurs. The measurement process uses an event-driven approach. Therefore, the application has an event-handling loop that waits for an event flag to be set. Once a flag is set, the event handling loop handles the event that occurred. The possible flags that can be set and the subsequent actions are explained below:

- *FLAG_BUTTON_MEASURE*: This flag is set when the left button of the sensor node is pressed, which triggers a new measurement by immediately setting the *FLAG_DO_MEASUREMENT*.
- *FLAG_BUTTON_STATUS*: Pressing the right button of the sensor node sets this flag. This measures the battery voltage and displays the current battery voltage level and the application status on the corresponding LEDs.
- *FLAG_BUTTON_COMMAND*: If the right button is pressed for more than three seconds, this flag is set, and the application exits the *MeasurementState* and enters the *CommandState*.
- *FLAG_DO_MEASUREMENT*: A ticker sets this flag at regular time intervals. When the flag is set, a new measurement is initiated by starting the SPS30 particle sensor if it is not already starting or running.
- *FLAG_SPS30_STARTUP_DONE*: This flag is set 30 seconds after the SPS30 sensor is started, as the sensor is now fully started and provides reliable measurements. The sensor readings are then read from the sensor, stored on the micro SD card, and sent to the communication thread.
- *FLAG_CREATE_NEW_FILE*: Every 24 hours, this flag is set by a ticker, which causes the application to create a new file for the measurement data on the SD card.

The use of event flags and an event-handling loop to handle occurring events avoids permanent polling for events. Since the RTOS manages the event flags, it knows when all threads are waiting for an event and can then put the microcontroller into deep sleep mode in the meantime.

ErrorState

If an error occurs in any of the other states, the *ErrorState* is entered, causing the status LED to turn red for three seconds and then restarting the sensor node. In case the error persists even after four consecutive restarts, the sensor node sleeps for one hour and then restarts again. This delay prevents the battery from being rapidly discharged by repeated restarts of the sensor node.

CommandState

As shown in Figure 4.5b, it is possible to switch between the *MeasureState* and the *CommandState* by pressing the right button for longer than three seconds. If the *CommandState* is active, the sensor node provides a serial command interface via the micro USB interface. A detailed description of the command interface follows in section 4.3.4.

4.3.3 Communication protocol

Besides the main thread, the second thread handles the wireless communication between the sensor node and the gateway. Since all sensor nodes developed for the SensMat project must use the same communication protocol, a protocol specification was elaborated that defines the LoRa parameters used, the message structure, and the order in which messages are sent. Figure 4.6 illustrates the communication thread control flow as implemented in the sensor node firmware.

To establish a connection between a sensor node and a gateway, the sensor node must send an authentication request that contains information about the sensor node. As illustrated in Figure 4.7, the request contains a fixed authentication identifier (ID) ($0xFF$) at the beginning, followed by an initial randomly generated node CRC sum. The subsequent hardware ID is needed to identify the sensor node type. For example, the particle sensor node has the hardware ID $0x4220$. Other sensor nodes that measure different characteristics have a different hardware

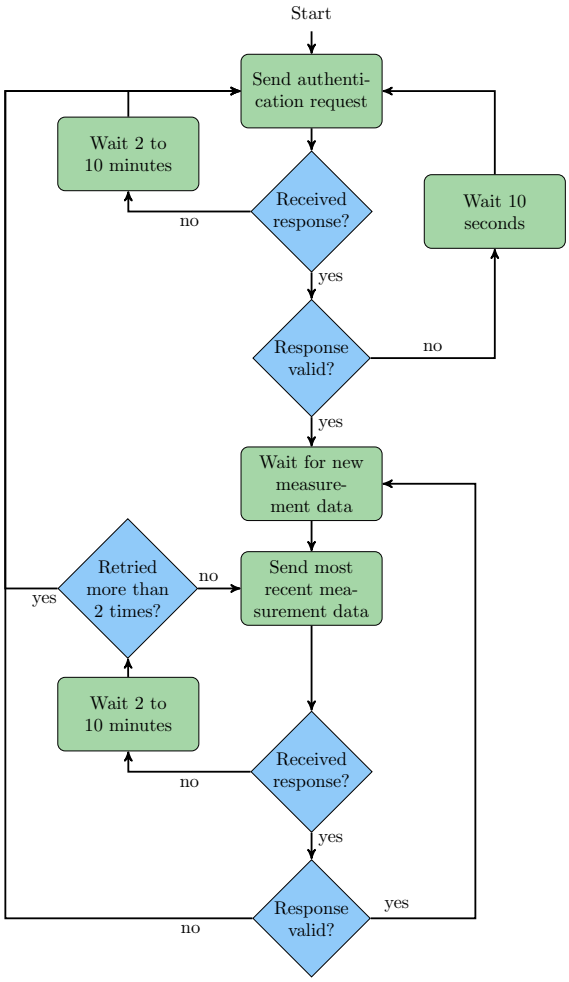


Figure 4.6: Control flow of the LoRa communication thread.

ID. The hardware IDs are defined for all types of sensor nodes in advance. The hardware unique identifier (UID) is derived from the microcontroller's serial number and is unique among all sensor nodes. A software UID is used to identify the firmware version running on the microcontroller. To ensure the integrity of the message, a 16-bit CRC sum over the preceding data is appended to the end of the authentication request.

When the gateway receives a valid authentication request, it sends an authentication response to the sensor node that contains a new node ID assigned by the gateway. It is used in each subsequent message to identify the sensor node. The base CRC following the node ID is generated by the gateway and is used to ensure correct packet ordering. The CRC sum for this message and all subsequent messages is

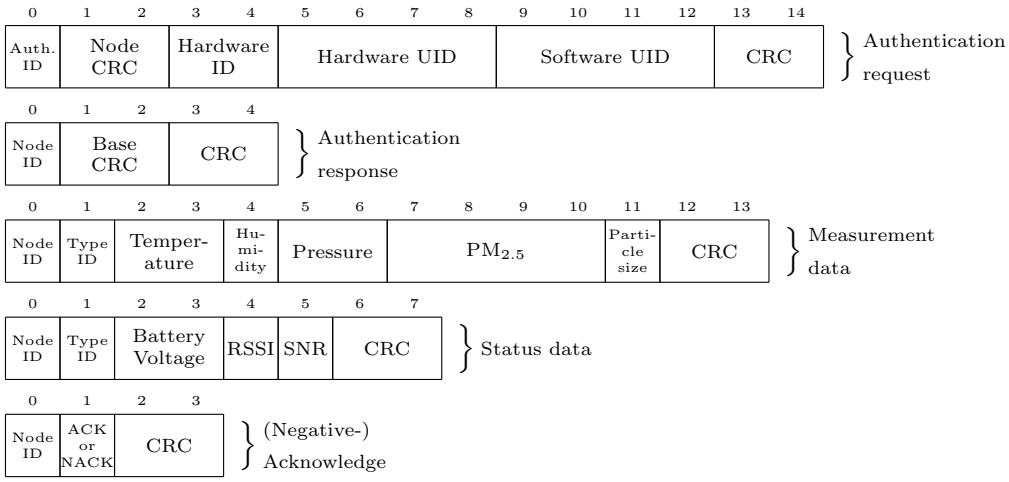


Figure 4.7: Format of the packets exchanged between sensor node and gateway.

now calculated over the preceding data in the message, the base CRC, and the node CRC.

If the sensor node does not receive an authentication response, it sends a new authentication request after a random timeout between two and ten minutes. If the authentication response is invalid, due to an incorrect CRC sum, for example, then the sensor node sends a new authentication request after ten seconds.

Now the sensor node is bound to a gateway and can send measurement data to the gateway. Therefore, the communication thread on the microcontroller waits for an event flag to be set by the main thread as soon as new measured values are available.

A data packet sent to the gateway contains a type ID after the node ID to identify which data is sent by the node to be decoded correctly on the server. Subsequently, the data is sent in a format corresponding to the type ID. The sensor node sends two different types of data packets. A measurement data packet contains the most recent values for temperature, relative humidity, pressure, $PM_{2.5}$ and the typical particle size. The sensor node can also send a status message at fixed time intervals. It includes the battery voltage as information about the battery charge state and the received signal strength indicator (RSSI) and SNR of the last received message for information about the radio link quality.

The gateway responds to a data packet with either an acknowledgment for a valid packet or a negative acknowledgment for an invalid packet. As shown in Figure 4.6, the sensor node retries to send data after a random timeout between two and ten

minutes if it did not receive a response. After three attempts, or if the response is invalid, the sensor node must re-authenticate.

4.3.4 Command interface

Since it is very tedious to recompile and flash the firmware every time a single parameter needs to be changed or the time needs to be adjusted, a command interface has been added to the sensor node's firmware to handle these and other tasks. The command interface can be accessed by connecting the sensor node to a PC via the micro USB interface. Then the sensor node needs to be put into *CommandState* and a serial terminal must be opened on the PC. Afterward, the user can send commands to the sensor node, which also provides feedback on the commands, similar to a command-line interface.

For the implementation of the command interface, the command pattern was applied, which is a behavioral software design pattern. Its main purpose is to decouple the invocation of a command from the actual execution, which means that an invoker of the command does not need to know how to execute it. This can be achieved by creating a concrete subclass of an abstract *Command* class for each command, as shown in the simplified class diagram in Figure 4.8. The *CommandHandler*, representing the caller, receives a command string over the USB serial port and determines which command to execute (White, 2011). The command is then processed in the respective *Command* class. Another advantage of the command pattern is that a new command can be added easily with minimal integration effort.

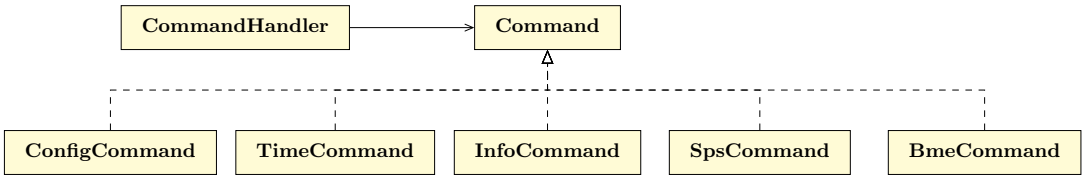


Figure 4.8: Class diagram of the command interface implementation.

The command interface provides a help command that prints all supported commands or the usage information for a particular command. Each of the five implemented commands is described in more detail below.

ConfigCommand

This command is used to display and change various configuration parameters. Examples of configurable parameters are sampling time, calibration values, log level, or whether measurement data is logged on the micro SD card or not. Since the configuration parameters should persist even if the sensor node is switched off, they are stored in a specific area of the flash memory. The flash memory retains its data even when it is not powered, and the mentioned memory area is not overwritten when flashing the firmware. When the sensor node is initialized after power-up, the configuration is loaded from the flash memory. A CRC sum is calculated over the parameters and written to the flash memory to detect unintentional changes to the configuration parameters. When loading the parameters, a CRC check is performed, and if the CRC sum does not match, the default configuration is used. The configuration parameters can also be reset by pressing both push buttons while switching on the sensor node.

TimeCommand

To log measurement data with correct timestamps, the RTC of the microcontroller must match the real time. Therefore, the *TimeCommand* provides the functionality to read the RTC and set it to a given time. Furthermore, this command is used to calibrate the RTC, described in more detail in section 4.3.5.

InfoCommand

The *InfoCommand* prints the firmware version, software UID and hardware UID. This allows the unique identification of each sensor node based on the hardware UID and provides information about the currently running firmware version.

SpsCommand

For testing the SPS30 particle sensor, this command provides functionality for controlling and testing the SPS30. It can be started to subsequently read and display the measured particle values at an interval of one second. In addition, the sensor supports a self-cleaning mode, which can be initiated manually using this command.

BmeCommand

Like the *SpsCommand*, this command allows direct reading of measured values from the BME280 environmental sensor at one-second intervals.

4.3.5 RTC calibration

An RTC, such as the one inside the STM32L476RG microcontroller, only provides the real time if it is set and calibrated to the correct time because RTCs have a certain time drift. This drift is caused by the inaccuracy of the crystal oscillator clock from which the RTC derives the time. The microcontroller requires a 32.768 kHz crystal oscillator for the RTC. Its accuracy depends on the load capacitance connected to the terminals of the crystal oscillator and the used crystal itself (STMicroelectronics, 2020b).

As shown in Figure 4.9, the load capacitance is composed of the two external load capacitors C_{L1} and C_{L2} and the stray capacitance of the PCB and connections (C_S).

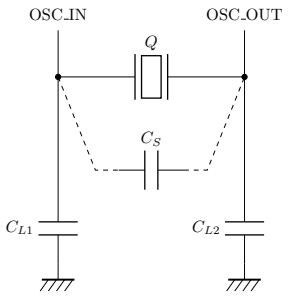


Figure 4.9: The crystal oscillator circuit connected to the microcontroller with *OSC_IN* and *OSC_OUT* (STMicroelectronics, 2020b).

Hence, the load capacitance C_L is calculated as:

$$C_L = \frac{C_{L1} \times C_{L2}}{C_{L1} + C_{L2}} + C_S, \tag{4.1}$$

where C_S must be measured or estimated. For this work, C_S was estimated to be 5 pF. The required load capacitance C_L for a given crystal oscillator is given in the datasheet. For the crystal oscillator used in this sensor node, C_L should be as close as possible to 12.5 pF. Substituting these values into equation 4.1 gives $C_{L1} = C_{L2} = 15$ pF (STMicroelectronics, 2020b).

Since C_S could only be estimated and the 15 pF capacitors have some tolerance, the clock of the crystal oscillator is still not exactly 32.768 kHz, which leads to a time drift at the RTC. Therefore, the RTC has two registers to correct small timing differences between −487.1 to 488.5 ppm (STMicroelectronics, 2020a).

For convenient RTC calibration, a command-line tool has been developed to measure the time drift. It uses the *TimeCommand* of the command interface, which provides two functions that can be used with the command-line arguments `tick` and `calibrate`. Sending the command `time tick 60` via the command interface causes the microcontroller to send a message every time the seconds-digit of the RTC changes. This is done for 60 seconds, as defined in the second argument of the command sent. The RTC tool stores a timestamp when it receives a message from the microcontroller. After capturing all timestamps, the RTC tool calculates the differences between the first ten and the last ten timestamps and subtracts the real time difference (50 seconds). By calculating the mean value of the ten resulting time-shift values and scaling it to one day, the time drift per day is obtained. This value is sent to the microcontroller with the command `time calibrate 5483` if the calculated time drift is 5483 ms/day. Afterward, the microcontroller calculates and sets the two register values according to the received time drift. The register values define after how many clock cycles an additional clock cycle is inserted, or a clock cycle is skipped to correct the timing drift.

This method allows only a calibration with limited accuracy because several factors can influence the time measurements. Uncertainties can come from the USB driver, the baud rate, the varying CPU utilization, or the buffering of messages, for example. However, by collecting the timestamps over a longer period and calculating the time drift over a longer period, the influence of these uncertainties can be reduced. The developed RTC tool was used to calibrate the RTCs of the microcontrollers to keep the maximum time drift within an acceptable range of $<$1 s/day.

4.3.6 Drivers

Accessing peripherals with a microcontroller often requires driver software. Since writing drivers for each peripheral is a time-consuming task, existing drivers have been integrated into the firmware. Low-level drivers are usually implemented generically, without making assumptions about how the peripheral will be used. Hence, these drivers provide all the functionality that is supported by the pe-

ripheral. However, this often means that more code is required to implement applications if the low-level driver functions are used directly.

To reduce redundant code, the original drivers can be abstracted by wrapping them in adapter classes. This method is also known as the adapter pattern, another design pattern for object-oriented software. As a result, the adapter class provides a higher-level interface to the peripherals with only functions needed by the application. Furthermore, the adapter classes can be customized to better fit into the software structure.

The adapter pattern was also used in this work for the implementation of driver classes. For the SPS30 sensor and the BME280 sensor, the low-level drivers are provided by the sensor manufacturers. Since the drivers are written in C, and the application is mostly written in C++ with an object-oriented structure, a separate adapter class was implemented for each sensor based on the low-level driver. Thus, the adapter class provides only those functions of the sensor that are actually required by the application.

For SD card access, Mbed OS includes the *SDBlockDevice* class for raw block device access and the *FATFileSystem* class for file system access. Both classes were used to implement the *SDCard* class, which handles all low-level operations and any errors that occur. The only public methods through which the micro SD card can be used are provided by the *SDCard* class.

As for the micro SD card, Mbed includes a driver for the SX126x LoRa transceiver ICs, which has also been wrapped in an adapter class. This class provides four functions: *init()*, *send()*, *receive()* and *sleep()*, resulting in a neater interface compared to the low-level driver.

Simple driver classes were implemented for the trivial peripherals, namely the LEDs, the pushbuttons, and the battery voltage divider. The class *LedIndicator* provides a method to display a certain status color on the status LED for a specified duration and a method to display the current battery voltage level on the battery level LEDs. For the latter method, the *Battery* class is used, which activates the switchable battery voltage divider and reads the current battery voltage from an analog input. The *Buttons* class captures and debounces the button input signals. It also measures how long a button is pressed because one of the buttons has two different functions depending on the duration the button was pressed. When a button is released, a corresponding event flag is set for another part of the firmware to react to.

5 Measurements and Results

The individual components of the sensor node were evaluated for their functionality. These evaluations encompassed the LoRa module, power supply parts, and the used sensors. This chapter presents the methods used for the evaluations and the results obtained from them.

5.1 LoRa module

The LAMBDA62 LoRa module datasheet contains a test report describing a range test in an outdoor environment with two LAMBDA62 modules exchanging ping messages. Thereby, the packet reception rate was 100 % at a line-of-sight (LOS) distance of 12 km. However, in an urban area with a non-LOS distance of 3 km, several packets were lost. For these tests, they soldered the LAMBDA62 modules on a breadboard and used simple wires as antennas. They state that with an improved PCB design and antenna, higher ranges could be achieved (RF Solutions, 2021). Indoors, the range is significantly reduced compared to outdoors. Due to building structure and furnishings, the signal is attenuated, resulting in a weaker signal at the receiver side (Neumann et al., 2016).

To verify that the performance of the LoRa module is also sufficient in an indoor environment and how different antennas affect the communication range, a test setup was developed to measure the performance of the sensor node's wireless communication.

5.1.1 Antenna comparison

The LoRa module was evaluated before the design phase to determine if the LoRa module is suitable for this application. Therefore, the LoRa module was plugged onto a NUCLEO-L476RG development board along with an adapter board. A simple whip antenna was connected to the U.FL connector of the LoRa

module. A trivial application was developed using two of these end devices to test the communication between them. The measurements showed that reliable communication over a sufficiently long distance is possible with this setup.

However, the antenna used for this test is a whip antenna, which requires a through-hole mounting on the housing of the sensor node and would thus protrude 87 mm from the housing. Since the sensor node is intended to be inconspicuous, a whip antenna is unsuitable. Instead, an adhesive antenna was chosen, which is affixed to the inside of the housing lid and is thus invisible from the outside.

In order to evaluate whether this adhesive antenna performs better or worse than the whip antenna, a test was carried out to compare the two antennas. For this purpose, a NUCLEO-L476RG board with a LoRa module and a whip antenna served as the base station. Two fully assembled sensor nodes, one with a whip antenna and one with a adhesive antenna, were used as endpoints. Figure 5.1 illustrates the test setup.

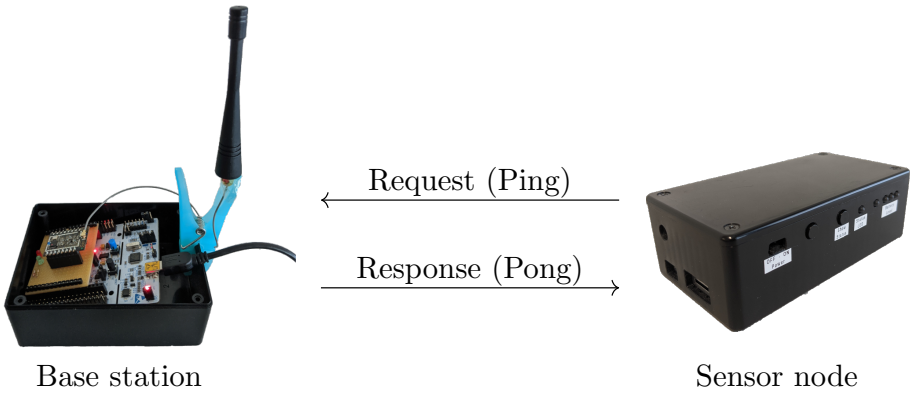


Figure 5.1: The test setup for the antenna and range test consists of a base station and a sensor node that exchange ping-pong messages. Here, the sensor node uses the adhesive antenna.

The two devices were placed at a LOS distance of 1 m from each other and 1 m above the ground. For the test, the sensor node sent a ping message to the base station, which responded with a pong message upon reception. This sequence was repeated 100 times to obtain statistically meaningful results. Table 5.1 lists the LoRa parameters used for this test. They were set to the values defined in the SensMat communication protocol specification.

Figure 5.2 shows the distribution of RSSI values determined by the LoRa modules at the base station and at the sensor node when receiving the ping message and the pong message, respectively. The RSSI values at the sensor node were about 5 dBm higher with the adhesive antenna compared to the whip antenna. The RSSI values

Table 5.1: LoRa parameter settings used for the measurements.

LoRa parameter	Value
Carrier frequency	868 MHz
Bandwidth	125 kHz
Spreading factor	12
Code rate	4/5
Transmit power	14 dBm

at the base station were also slightly higher when using the adhesive antenna. These results show that both the receive and transmit signal strength could be increased with the adhesive antenna. However, these results are only an estimate since different antennas have different and mostly non-spherical radiation patterns, so the signal strength depends not only on the distance between the antennas but also on their orientation.

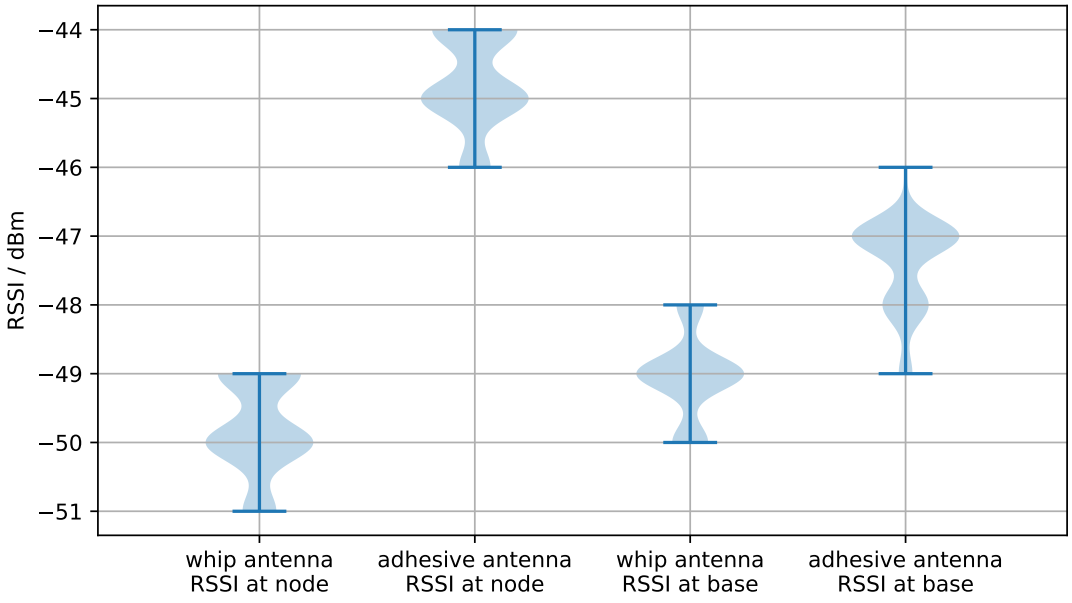


Figure 5.2: RSSI values measured at the base station and at the sensor node, whereby two different antennas were tested at the sensor node.

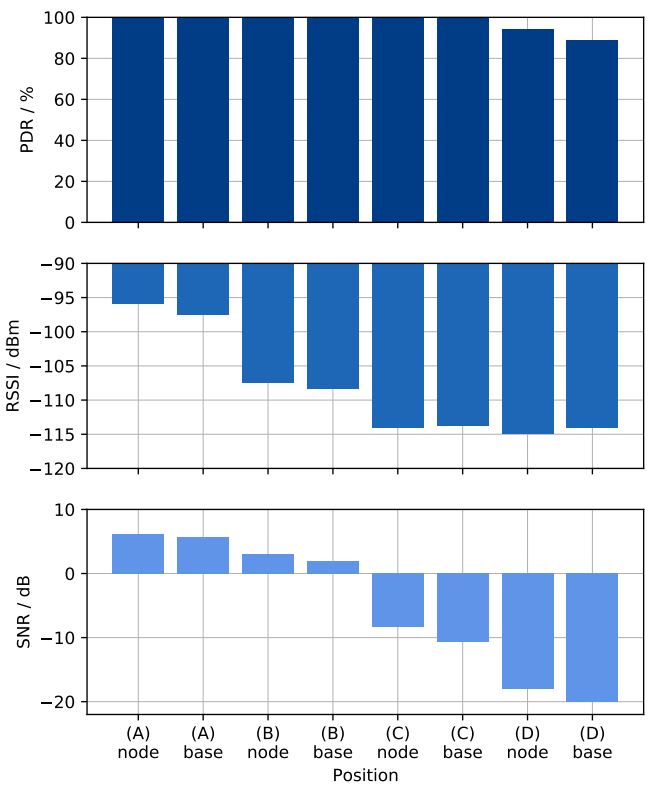
5.1.2 Range and reliability

Since the sensor nodes are used indoors, the range measurements were also performed indoors. As in the measurements from the LoRa module datasheet, the communication range in environments without LOS between two end devices is expected to be significantly lower than in measurements with free LOS. In indoor environments, walls and floors between the communicating devices are the main cause of signal degradation as they attenuate the signal. Additionally, the signal is reflected by walls and floors, which may cause interferences with the original signal.

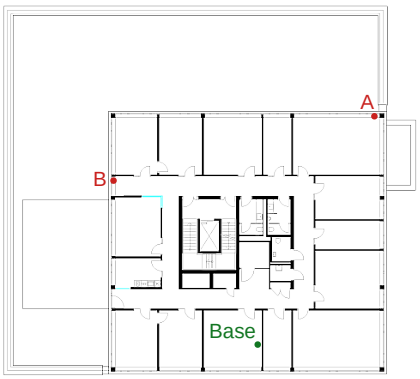
For the range and reliability tests, the same equipment with only the adhesive antenna (see Figure 5.1) and identical LoRa parameters (see Table 5.1) as for the antenna comparison measurements were used. To measure the reliability and signal strength of wireless communication for different spatial arrangements, the base station was placed in an office room, and the sensor node was placed at different locations in the same building. Figure 5.3b and Figure 5.3c show the test locations of the sensor node spread over two floors.

For positions A to D, the measured packet delivery rates (PDRs), RSSI values, and SNRs are shown in Figure 5.3a. The values shown are the averages of all measurements at a position. For positions E and F, none of the packets sent by the sensor node was received by the base station. The results show that for positions A and B, which are on the same floor as the base station, the RSSI value and SNR are high enough to achieve a PDR of 100 %. When comparing the results of positions A and B, it is noticeable that the RSSI and SNR for B are lower than for A, although B is closer to the base station. This can be attributed to the higher attenuation caused by the walls between the base station and the sensor node at position B.

At position C, on the first floor, the PDR is 100 %, but the received signal strength and SNR are already significantly reduced. Position D is also on the first floor and is a few meters further away than position C. There, 16 of 100 packets were lost during the ping-pong round trip. Comparing the RSSI values for positions C and D show that they are very similar. However, the SNR at position D is only -20 dB, almost 10 dB below the SNR at position C. The reduced PDR means that the received signal power is at the detection limit of the receiver. This is consistent with LoRa characteristics, which state that LoRa transceivers can detect signals with a power level down to 20 dB below the noise floor (Voigt et al., 2016).



(a) Comparison of PDR, RSSI and SNR for different indoor locations.



(b) First floor.

(c) Ground floor.

Figure 5.3: Results (a) and locations (b, c) for the indoor range measurements performed in a university building. The green dot marks the position of the base station and the red dots mark the positions of the sensor node.

5.2 Power supply

For battery-powered devices, electrical power supply handling is essential. Therefore, this part of the work first evaluates the electrical characteristics of the Li-ion battery that serves as the power source. Second, the low-voltage handling of the battery is considered. And third, the power consumption in different operating states of the sensor node is measured and examined.

5.2.1 Li-ion battery

The selected Li-ion battery pack consists of two Li-ion cells in series and delivers a nominal voltage of 7.2 V. To measure the real battery voltage over a discharge cycle, the battery was discharged through a 37 Ω resistor while the battery voltage was measured. Figure 5.4 shows the captured voltage curve. If the charging voltage (8.4 V) is removed when the battery is fully charged, the voltage will drop a little. So, in this measurement, the discharge cycle starts at 8.26 V and not at 8.4 V. During the first half of the discharge cycle, the battery voltage decreases approximately linearly to the nominal battery voltage before dropping at a slower rate. Towards the end, the voltage drops faster and faster until the over-discharge detection (cut-off) voltage of the battery protection circuit is reached. The battery capacity calculated from the discharge curve is 2568 mAh, which is between the minimum (2500 mAh) and nominal battery capacity (2600 mAh) specified in the datasheet (Ansmann AG, 2020).

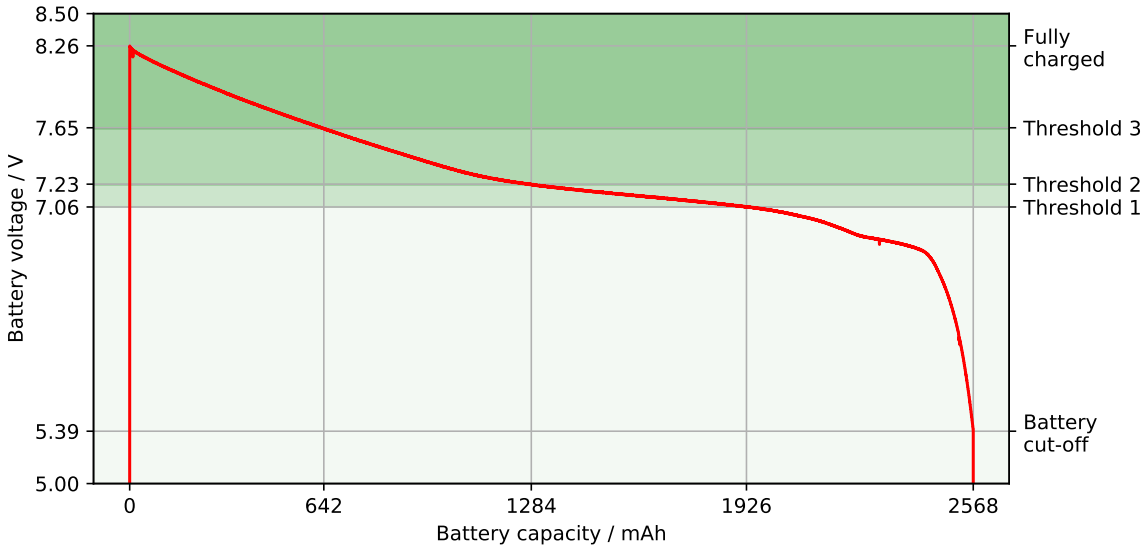


Figure 5.4: Measured battery voltage for one discharge cycle.

The discharge curve was also used to determine the switching thresholds for the three battery status LEDs on the sensor node's front panel. Therefore, the calculated capacity was divided into four equal parts. The corresponding threshold voltages at the boundaries of these parts were then determined and taken as the threshold values, which are also shown in Figure 5.4.

5.2.2 Low battery handling

To protect the battery from over-discharge, which could damage the battery, the protection board included in the battery pack disconnects the battery cells from the load when the voltage of a cell drops below 2.4 V. In the discharge test, it was observed that the battery pack cuts off the voltage at 5.39 V already, which is slightly lower than the standard discharge voltage of 5.5 V specified in the datasheet. Since the cut-off voltage varies between different battery packs, the low voltage function of the 3.3 V regulator was used to set the cut-off voltage to a higher voltage level.

The cut-off voltage and the release voltage of the 3.3 V regulator can be adjusted by choosing suitable values for four connected resistors. Using a calculation sheet provided by the manufacturer, the resistor values were calculated to set the cut-off voltage to 6.25 V and the release voltage to 6.6 V. To verify that the actual threshold voltages match the calculated ones, the battery voltage and the drawn current were measured during operation towards the end of a discharge cycle. Voltage and current curves are shown in Figure 5.5. Here, the sensor node was configured to measure every 30 seconds, as reflected by the repeating current peaks in the current waveform. These are caused by the increased current consumption when sending the measurement data. The measured cut-off voltage is about 6.32 V, and the measured release voltage is about 6.65 V. Thus, the threshold values are higher than the calculated ones, which can be attributed to resistance tolerances and measurement inaccuracies.

The voltage difference between cut-off voltage and release voltage is the hysteresis, which should be 350 mV, while the measured hysteresis is about 330 mV. This hysteresis is necessary to prevent the regulator from being switched on after an under-voltage cut-off, which can happen because the battery voltage rises again a bit after the load is removed. This is also visible in Figure 5.5. At the end of the voltage curve, the battery voltage increases by a few millivolts.

Due to the low-voltage handling of the 3.3 V regulator, the supply voltage for the electrical components is switched off at the specified cut-off voltage, which is higher than the cut-off voltage of the battery's protection circuit. This is necessary because a particular scenario has been observed that may result in damage to the battery charging circuit. More precisely, this scenario can occur if the under-voltage protection of the battery protection circuit has been triggered and the power adapter is then connected for charging. This can damage the charging IC or the corresponding switching transistor leading to a malfunctioning charging circuit.

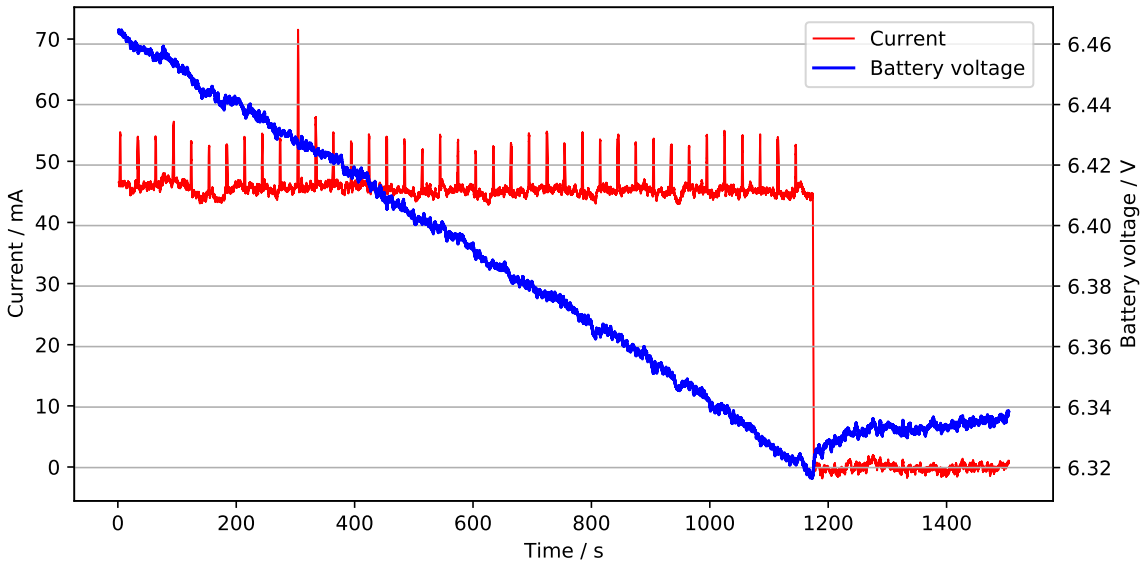


Figure 5.5: Captured voltage and current curve at the end of a discharge cycle, showing the 3.3 V regulator cut-off event.

5.3 Power consumption

In order to estimate the operating time of the sensor node, it is necessary to know the current consumption in all operating states. Therefore, the current consumption was measured while a specific sequence of operations was performed. Thereby, the sensor node was configured to measure once per minute. The sequence was generated with the following steps:

1. Switch on the sensor node.
2. Wait until two measurements are completed.
3. Press status button.
4. Wait until another measurement is completed.
5. Switch sensor node into command state.
6. Send a command (`time`) via the USB serial interface.
7. Switch sensor node back into measurement state.
8. Wait until one measurement is completed.
9. Switch off the sensor node.

Figure 5.6 shows the current consumption during this sequence, with the following important events and sections marked with numbers:

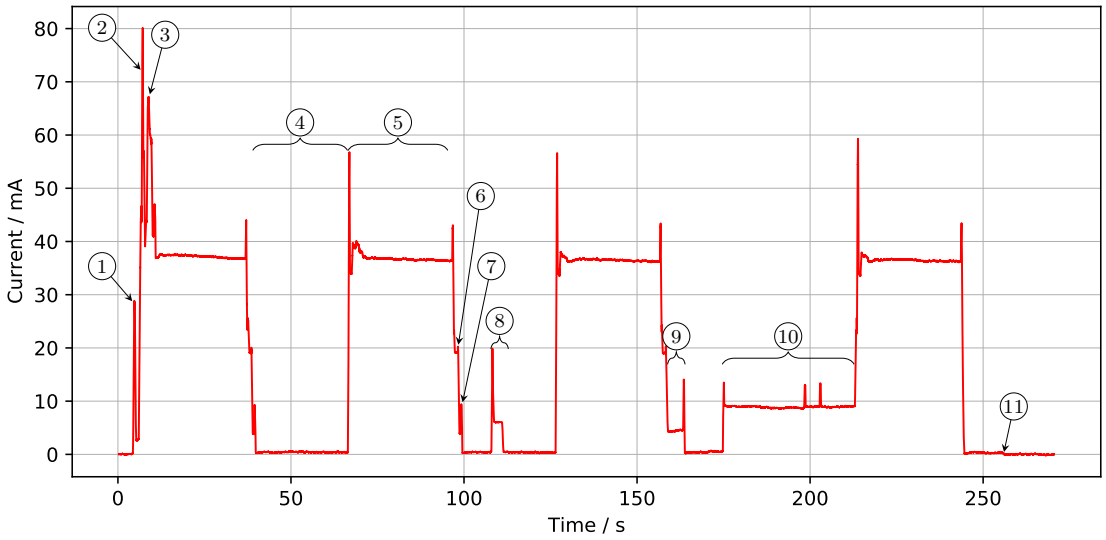


Figure 5.6: Measured current consumption of the sensor node in different operating states.

1. The first current peak after powering up the sensor node occurs during initialization while loading the configuration, initializing the SD card, and checking the communication with the sensors.
2. This peak is caused by the start-up of the SPS30 particle sensor, which is started 30 seconds before the particle measurement data is read. With a maximum of approximately 80 mA, this peak is higher than those of the following SPS30 start-up events because the sensor node exchanges authentication messages with the gateway at the same time, which consumes additional power.
3. After the authentication messages, the sensor node immediately sends status data to the gateway, resulting in another current peak.
4. In this section, the microcontroller is in deep sleep mode, where the average current consumption is 410 µA. This is the operating mode in which the sensor node spends most of its time, so it is crucial that the current consumption in this section is minimal.
5. While the SPS30 measures for 30 seconds, the current consumption is about 37 mA. The microcontroller is in deep sleep mode during this time.
6. After reading the measurement data from the sensors, the data is sent to the gateway, which causes a current consumption of 19 mA for the packet transmission time, which in turn depends on the LoRa settings. In this case, the time-on-air is about 1.16 s.
7. After sending the measurement data, the sensor node is in receive mode

and receives a response from the gateway, which requires less energy than sending.

8. While the sensor node was sleeping, the status button was pressed, after which the sensor node displayed the application status and battery status on the LEDs. This takes three seconds, and running all LEDs consumes 5 mA on average.
9. Here, the sensor node again waits for the gateway's response, but unlike before (at 7.), the gateway does not send a response, resulting in a timeout after three seconds.
10. In this section, the sensor node is in the command state, connected to a PC. Since receiving commands via the USB interface is not possible in deep sleep mode, the sensor node is only in sleep mode while waiting. As a result, the current consumption is significantly higher (9 mA). When a serial terminal connects or a command is received and processed, the microcontroller is in run mode and consumes additional power. Thus, the first peak (at 198 s) indicates when the serial terminal connected and the second peak (at 203 s) indicates when the *time* command was received and processed.
11. This marks the time when the sensor node was switched off, resulting in a current drop from the deep sleep current to 0 A.

Since most of the current is consumed when the SPS30 sensor is running, the operating time depends mainly on the measurement interval. However, as the measurement interval increases, the influence of the deep sleep current on the operating time increases.

Using the recorded current consumption from Figure 5.6, five main operating modes were identified that are repeatedly executed when the sensor node is in the measurement state. These operating modes and the current consumed in these modes are listed in Table 5.2.

Table 5.2: The current consumption of the sensor node for one hour operating time. Listed are the different operating modes, their current consumption and operating duration.

Operating mode	$\mathbf{I_{measure}}$ / mA	$\mathbf{V_{measure}}$ / V	$\mathbf{t_{active}}$ / s	$\mathbf{I_{average}}$ / mA
SPS30 startup	55	7.45	0.8	0.013
SPS30 running	37	7.63	120	1.307
LoRa transmit	19	7.81	5.4	0.031
LoRa receive	5	7.95	5.4	0.008
Deep sleep	0.41	8.00	3468.4	0.439
			Sum:	**1.798**

Thereby, $I_{measure}$ is the measured current in each operating mode. t_{active} is the time in seconds that an operating mode is active per hour, assuming four measurements per hour. The measured supply voltage $V_{measure}$ depends on $I_{measure}$ since the current was measured across a series resistor of 10 Ω. In normal operating mode, the supply voltage does not depend on the consumed current.

By normalizing the consumed current $I_{measure}$ to the nominal battery voltage $V_{bat\_nominal}$ (7.2 V) and multiplying with the duty cycle D, the average current consumption $I_{average}$ in each operating mode is calculated as

$$I_{average} = \frac{I_{measure} \cdot V_{measure}}{V_{bat\_nominal}} \cdot D = \frac{I_{measure} \cdot V_{measure}}{7.2\,\mathrm{V}} \cdot \frac{t_{active}}{3600\,\mathrm{s}}. \tag{5.1}$$

With a usable battery capacity $Q_{battery}$ of 2500 mAh and the summed average current consumption of all operating modes, the expected operating time of the sensor node $t_{operating}$ is

$$t_{operating} = \frac{Q_{battery}}{\sum I_{average}} = \frac{2500\,\mathrm{mAh}}{1.798\,\mathrm{mA}} = 1390\,\mathrm{h} = 57.95\,\mathrm{d}. \tag{5.2}$$

Long-term measurements showed that the calculated operating time is consistent with the actual operating time. With a measurement interval of 15 minutes, the sensor node was in operation for 57 days, which is slightly less than the calculated 57.95 days. With a shorter measurement interval of 36 seconds, the calculated operating time is 73.7 hours, and the measured operating time was 72.4 hours on average.

Thus, it can be assumed that with a measurement interval of one hour, as used in the final deployment of the sensor nodes, the operating time is approximately 131 days.

5.4 Sensors

Particulate matter is the main characteristic measured by the sensor node, which is done with the SPS30 particulate matter sensor. In addition, temperature, relative humidity, and absolute pressure are measured with the BME280. To gain more insight into the accuracy, stability, and reproducibility of the measured characteristics, the sensors were evaluated in different environments by performing various measurements.

5.4.1 Accuracy

Although each SPS30 sensor is factory calibrated with a reference device, the accuracy was measured and evaluated in this work. For this purpose, the sensor nodes were placed near an environmental measurement station in Weiz (Styria, Austria), which serves as a reference. The sensor nodes were configured to measure every 5 minutes, and the reference station provides measurements every 30 minutes. Figure 5.7 shows the measured values of the sensor nodes and the reference station for one week.

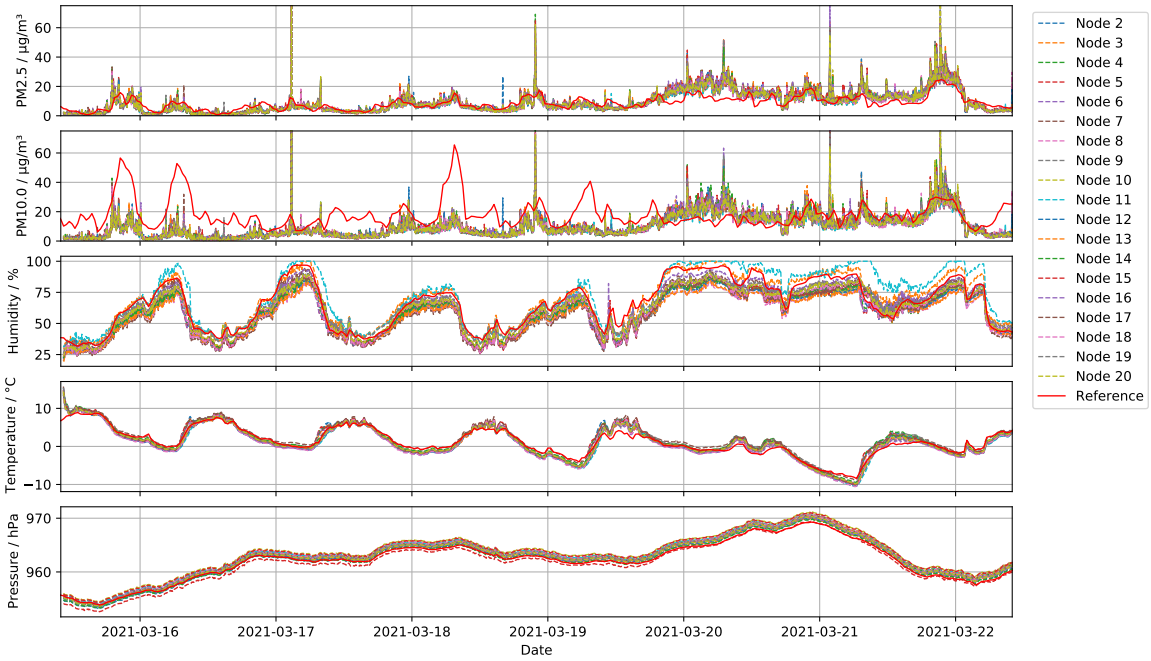


Figure 5.7: Measurement curves of the sensor nodes compared to the reference station.

Several high peaks were recorded in the $PM_{2.5}$ measurements. The reference station did not record these peaks because it averages the $PM_{2.5}$ concentration over 30 minutes, while the sensor node only measures the $PM_{2.5}$ concentration over 30 seconds at specific time intervals. This allows the sensor node to better detect short events with high or low $PM_{2.5}$ concentrations, but as a result, the measurements do not always reflect the long-term $PM_{2.5}$ concentration. Since the sensor node is intended to measure the long-term $PM_{2.5}$ concentration, it calculates the moving average over several samples to minimize the influence of high peaks in the measurements.

To quantify the accuracy of the measured characteristics, the mean accuracy $\bar{A}$ over n measurements was computed as

$$\bar{A} = 100\,\% \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} \left(1 - \frac{|X_i - R_i|}{R_i}\right), \tag{5.3}$$

where X_i and R_i are the measured values of the sensor node and the reference station, respectively (South Coast Air Quality Management District, 2019).

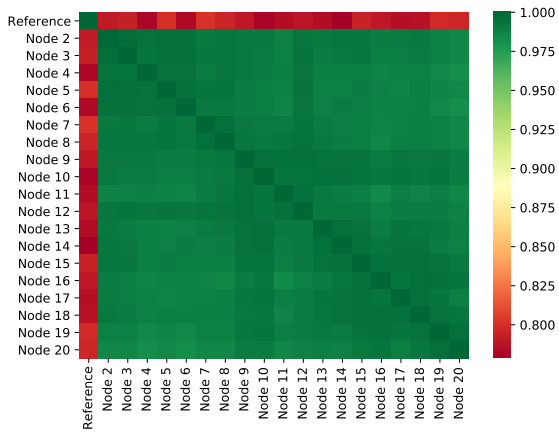
With measurement data from one week (Figure 5.7), the SPS30 sensors show an accuracy of 56–64 % for the $PM_{2.5}$ concentration. The correlation between the values of the sensor nodes and the reference station is shown in Figure 5.8a. It shows that the part-to-part variance of the SPS30 sensors is very low, which is consistent with measurements from other publications (Kuula et al., 2020; South Coast Air Quality Management District, 2019).

In contrast to the $PM_{2.5}$ concentration, the mean accuracy of the PM_{10} concentration measured by the SPS30 sensors is significantly lower with 44–48 %. Comparing the sensor node's $PM_{2.5}$ and PM_{10} values from Figure 5.7 and taking into account the low correlation to the reference values shown in Figure 5.8b, it appears that the SPS30 does not accurately measure larger particles.

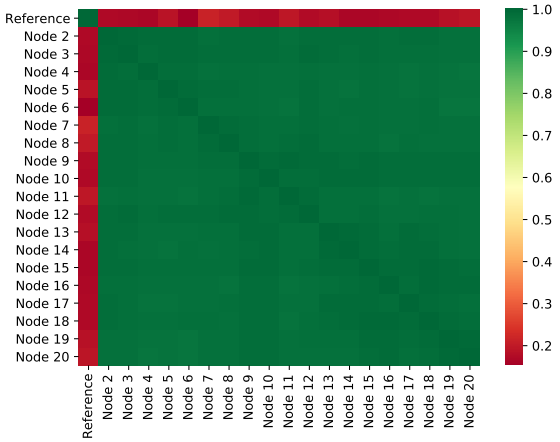
During the last two and a half days of measurement, humid weather and snowfall led to high relative humidity, visible in Figure 5.7. It could be observed that high relative humidity causes the SPS30 sensor to overestimate the particle mass concentration.

The relative humidity is measured with the BME280 sensor. While the correlation to the reference station is high, the BME280 sensor tends to measure lower values than the reference station, resulting in a mean accuracy of 85–93 %. The Pearson correlation from Figure 5.8c shows a worse correlation for sensor node 11 compared to the others. This is probably the result of a damaged BME280 sensor, which was therefore replaced after these measurements.

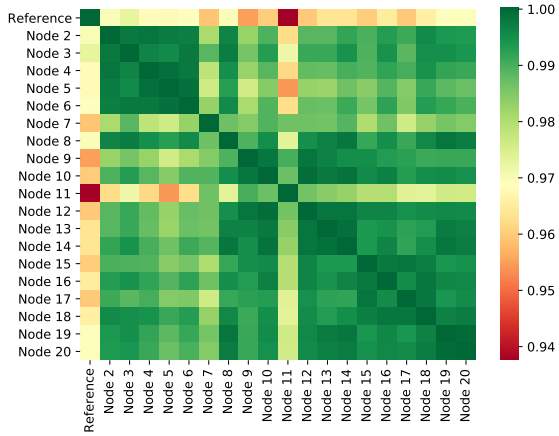
Temperature and pressure, measured by the BME280 sensor, show high accuracy and high correlation with the reference station readings (see Figure 5.8d and 5.8e). In previous measurements, the humidity and temperature showed higher deviations, whereby the temperature was too high, and the relative humidity was too low. Since this was only the case during strong daylight, the deviation can be attributed to the heating up of the black enclosure caused by the sunlight. To avoid this effect in the following outdoor measurements, the sensor nodes were covered with a white tarpaulin.



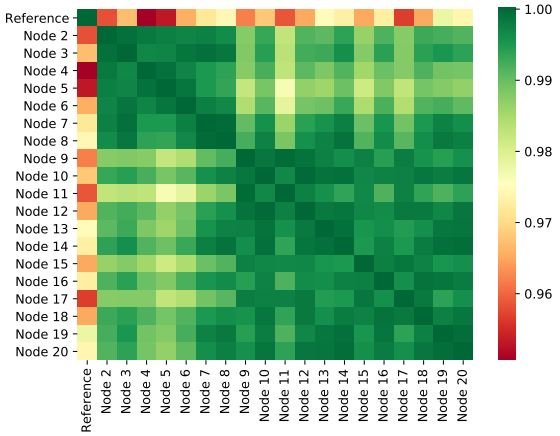
(a) $PM_{2.5}$



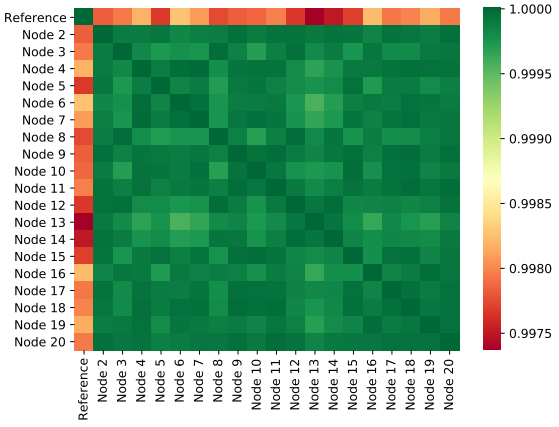
(b) PM_{10}



(c) Relative Humidity



(d) Temperature



(e) Absolute Pressure

Figure 5.8: The Pearson correlation coefficients between the measurement values of the sensor nodes and the reference station.

The SPS30 sensor needs at least 30 seconds after start-up until it delivers reliable measurement values. To investigate the progression of the measured $PM_{2.5}$ concentration in the first 180 seconds after sensor start-up, readings were acquired at one-second intervals. As shown in Figure 5.9, the measured $PM_{2.5}$ concentration fluctuates widely in the first few seconds and settles to the final value after about 40 seconds. The sensor node in the final deployment is configured to read the $PM_{2.5}$ concentration after 30 seconds, although the error to the final value is still slightly higher than after 40 seconds.

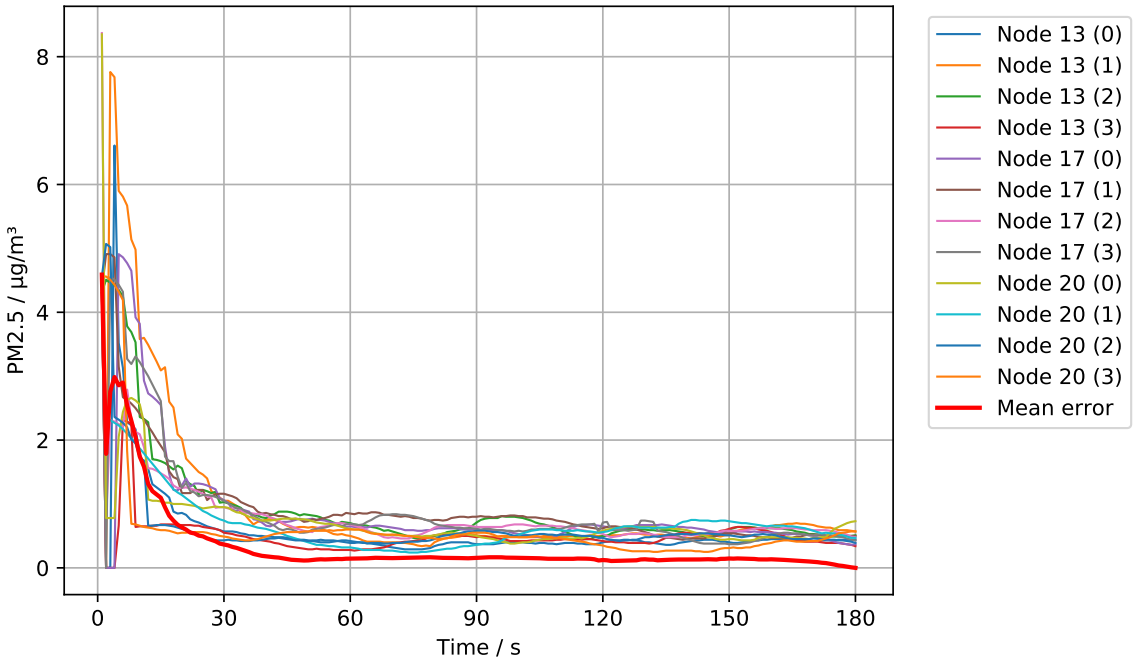


Figure 5.9: The measurement values for the first 180 seconds after starting the SPS30 sensor. The mean error was calculated in relation to the end value.

This error is acceptable for this application because the measurements from Figure 5.9 were taken in a room with very low $PM_{2.5}$ concentration, causing the SPS30 sensor to take a long time to count enough particles to estimate the $PM_{2.5}$ concentration. In an environment with higher $PM_{2.5}$ concentration, the SPS30 sensor provides reliable readings after less than 30 seconds.

5.4.2 Adjustment

Based on measurement data from ten days, different adjustment methods were evaluated to determine if they increase the accuracy of $PM_{2.5}$ measurements. Table 5.3 lists the analyzed methods and the corresponding functions. For the analysis, the measured data was divided into four parts. Thereof, one part was used as a training set to calculate the parameters for the functions. Another part was used as a test set to test the performance by applying the function with the calculated parameters. Then the result was compared with the reference station measurements by calculating the root-mean-square error (RMSE). In order to obtain statistically meaningful results, several combinations were evaluated by using different parts of the measurement data as the training set and the test set.

Table 5.3: These eight adjustment methods were evaluated by calculating the parameters for the functions with a training set and testing the performance with a test set.

Adjustment method	**Adjustment function for $PM_{2.5}$ (P)**
no adjustment	$P_{cal}(P) = 1 \cdot P$
linear without offset	$P_{cal}(P) = a \cdot P$
polynomial function (1^{st} degree)	$P_{cal}(P) = a \cdot P + b$
polynomial function (2^{nd} degree)	$P_{cal}(P) = a \cdot P^2 + b \cdot P + c$
polynomial function (3^{rd} degree)	$P_{cal}(P) = a \cdot P^3 + b \cdot P^2 + c \cdot P + d$
exponential function	$P_{cal}(P) = a \cdot e^{b \cdot P}$
power function	$P_{cal}(P) = a \cdot P^b$
humidity (H) and temperature (T) dependent	$P_{cal}(P, H, T) = a \cdot P + b \cdot H + c \cdot T$

The average RMSEs per sensor node and method are shown in Figure 5.10. It can be noted that the third-degree polynomial function and the exponential function do not improve the accuracy because the RMSE is too high. These methods tend to provide parameters that result in an overfitted function, increasing the error instead of minimizing it. Also, for a first and second-degree polynomial function or a linear function without offset, the RMSE is higher than for the non-adjusted data.

The last function from Table 5.3 additionally uses relative humidity and temperature as input since a correlation between high humidity and $PM_{2.5}$ concentration was observed. However, it turned out that this function leads to a higher RMSE on average and does not improve the measured values.

One evaluated method which uses a power function decreases the RMSE compared to the non-adjusted data. Further analysis of this method revealed that the calculated parameters are highly dependent on the training data. Thus, it is

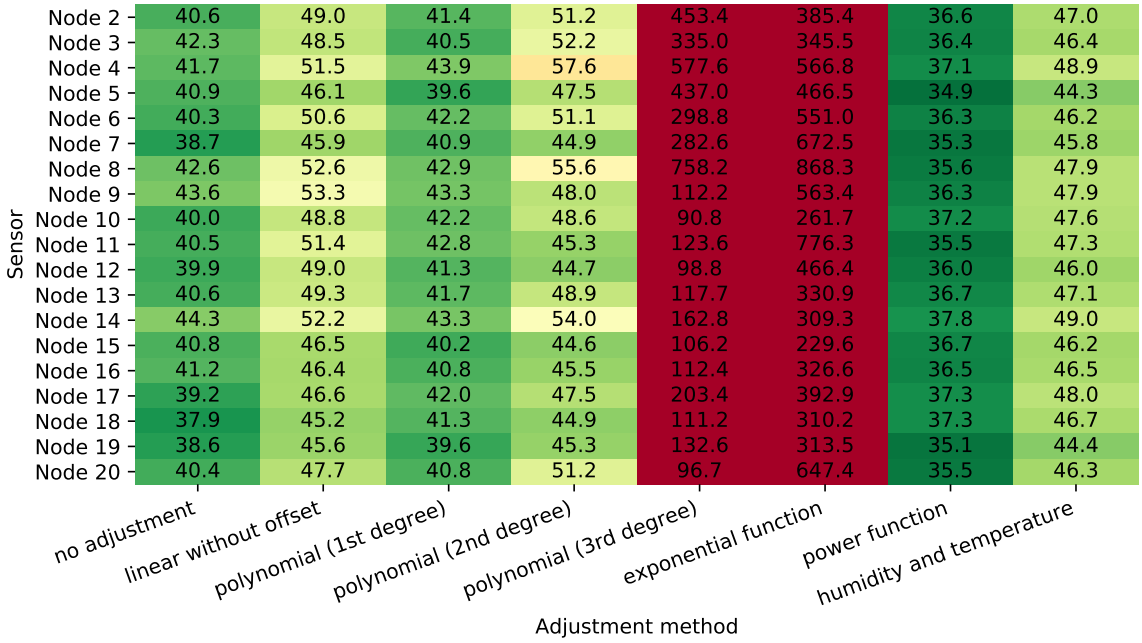


Figure 5.10: The average RMSE for the different functions.

difficult to determine a training set or respectively parameters that improve the measured values in all situations. Furthermore, by comparing the adjusted data with the non-adjusted data, it was observed that the power function flattens the measurement data, which is why it performs better on average. However, this reduces the sensitivity to short-term high or low $PM_{2.5}$ concentrations.

As a consequence of these findings on the different adjustment methods, it was decided not to adjust the $PM_{2.5}$ measurements because none of the evaluated methods significantly improved the measurement results in all situations.

5.4.3 Potential side effects affecting PM measurements

In addition to the factors already mentioned, multiple other effects can reduce the accuracy and precision of the particle sensor readings, resulting in errors between the measurements and the actual $PM_{2.5}$ concentration. Figure 5.11 shows possible causes that might affect the measurement deviations. Most causes are derived from observations and measurements already presented. The remaining causes are estimates that may also affect the measurement accuracy.

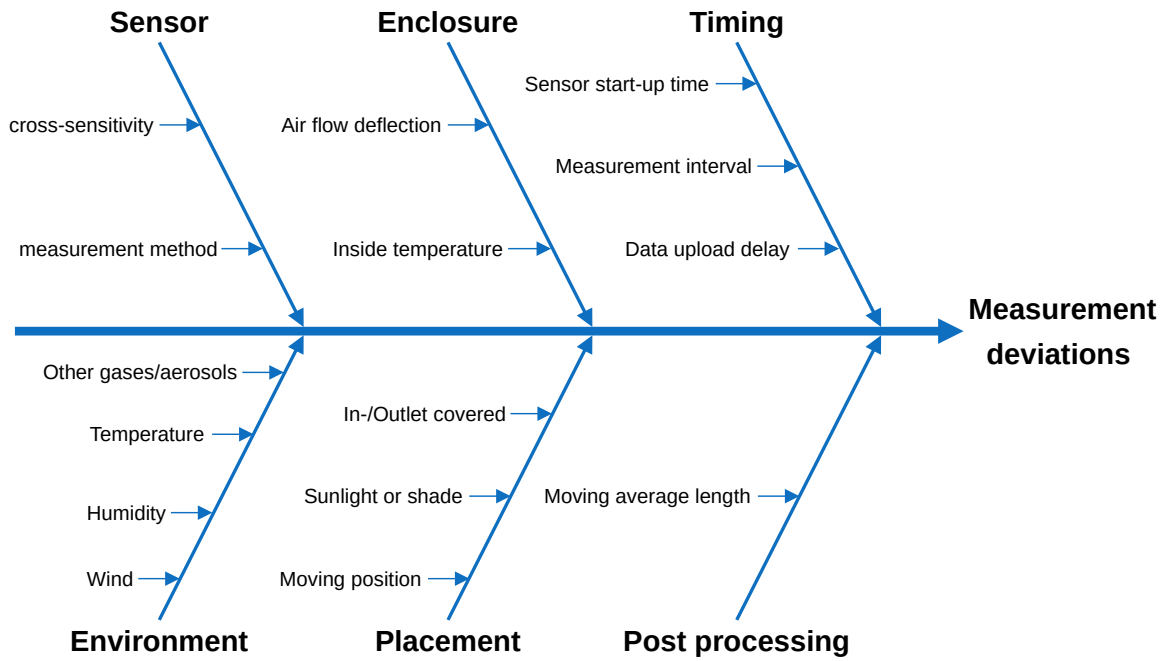


Figure 5.11: Causes that could affect the measurement deviations.

Sensor

The particle sensor itself provides only limited accuracy, mainly due to its measurement method, which does not directly measure the particle mass concentration. Instead, it is derived from optical metrics as described in Section 3.2.3. Cross-sensitivity to other gases or aerosols is another critical factor to consider when interpreting the measurements.

Enclosure

Although no effects of the housing on the measured values were observed in tests, it cannot be ruled out that there are situations in which the housing can influence the airflow, resulting in distorted measurements. Furthermore, the temperature inside the housing can also play a role as it may not correspond to the ambient temperature.

Timing

As described in section 5.4.1, the SPS30 requires a certain start-up time to provide reliable measurements, which means that the start-up time also influences the measurement deviations. By sampling only at regular time intervals, the measurements are only snapshots, and the time between two measurements is not taken into account. Furthermore, the sensor node may not send the measurement data immediately after the measurement, but only after a few minutes delay. That causes the uploaded data to be assigned to a later timestamp at the server.

Environment

From observations, it can be said that relative humidity and temperature have the most significant influence on measurement deviations among the environmental conditions. However, other gases or aerosols can also lead to incorrect particle counts or incorrectly estimated particle sizes. In addition, external flow conditions can deflect the airflow, which can cause an increased or decreased airflow through the particle sensor, which should be as constant as possible.

Placement

When placing the sensor node, an unobstructed airflow through the inlet and outlet must be possible. To avoid heating the housing and the sensor node, it should not be placed in direct sunlight. Shifting the position during measurement or between measurements can also affect the measurement results.

Post Processing

Before uploading the measured data, the measured $PM_{2.5}$ concentration is averaged over a certain number of samples. As a result, the sensitivity to short events decreases with the size of the averaging window. For this work, the averaging is needed because the sensor node is not meant to measure short-term events, but it should capture long-term exceedances of limits.

6 Conclusion

In this work, a low-cost wireless sensor node was developed to measure particulate matter and other environmental quantities indoors. As part of the SensMat project, it is deployed in several museums and connected to a wireless network together with other sensors.

Based on the requirements originating from the SensMat project, the hardware, software, and mechanical design for the sensor node were developed. With the selection of low-power components, a rechargeable Li-ion battery as the power source, and a suitable electronics design, the sensor node can measure and upload data for several months, with the operating time depending on the measurement interval. By integrating the charging circuit for the battery into the sensor node and adding pushbuttons, switches, and LEDs to the front side, the sensor node is easy to use.

To minimize the development effort, an existing LoRa module was integrated into the sensor node PCB. This LoRa module and an unobtrusive adhesive antenna make it possible to transmit measurement data wirelessly across several rooms and floors to the gateway. Measurements have proven that LoRa is very well suited for low-power applications where long ranges need to be covered. In addition to its physical properties, LoRa is a flexible technology that is well suited as a basis for a customized communication protocol.

The microcontroller's firmware is based on the Arm mbedOS framework, which includes an RTOS, many drivers, and tools, facilitating the development of embedded application software in C++. The framework also allowed the use of well-known object-oriented design patterns to develop modular and structured firmware. By adding a serial command-line interface that can be accessed from a PC via a USB connection and a serial terminal, parameters can be adjusted to change the behavior of the firmware at runtime. It is also possible to calibrate the RTC and set its time via this interface using a desktop program developed specifically for this purpose.

Before deploying the sensor nodes, tests were conducted to evaluate the performance of the sensors. By placing the sensor node next to a federal air quality

monitoring station for several days, the measurements of the sensor nodes could be compared with those of a reference instrument. Aside from measurement variances caused by the different measurement methods of the sensor node and the reference station, the $PM_{2.5}$ concentration of the sensor nodes showed an average accuracy of 56–64 % compared to the reference station and minimal part-to-part variance.

However, the PM_{10} concentration of the used particle sensor does not seem to detect particles with diameters of 2.5–10 µm, resulting in an average accuracy of 44–48 %. Outdoor measurements further showed that the sensor node heats up in direct sunlight due to its black housing, leading to measurement deviations in temperature and relative humidity. Further observation of the measurement results showed that high humidity leads to a strong overestimation of the PM concentrations.

In order to optimize the measurement accuracy of $PM_{2.5}$ concentration, several adjustment methods were evaluated with calibration data from the reference measurements. It was found that none of the methods significantly improved the accuracy evaluated with the test data sets. Hence, the only post-processing performed on the measurements calculates the moving average over multiple samples to reflect the long-term $PM_{2.5}$ concentration better.

With the deployment of currently 18 sensor nodes in museums, further data will be collected. The deployment of the sensors should help to identify relations between measurements and potential degradation effects on historical artifacts. Hopefully, this will help to gain more knowledge on how to create preventive solutions to preserve cultural heritage for many generations to come.

Bibliography

Amaral, Simone et al. (Sept. 2015). "An Overview of Particulate Matter Measurement Instruments." In: *Atmosphere* 6.9, pp. 1327–1345. DOI: 10.3390/atmos6091327 (cit. on p. 14).

Analog Devices Inc. (2003). *LTC4002. Standalone Li-Ion Switch Mode Battery Charger*. Analog Devices Inc.: Wilmington, MA, USA. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/4002f.pdf (visited on 02/20/2021) (cit. on pp. 18, 23).

Ansmann AG (June 18, 2020). *Li-Ion Battery 7.2V 2600mAh with safetyboard*. URL: https://docs.rs-online.com/7607/0900766b815e8220.pdf (visited on 03/31/2021) (cit. on p. 46).

Arm Limited (2021). *Mbed OS*. URL: https://os.mbed.com/mbed-os/ (visited on 03/13/2021) (cit. on p. 30).

Augustin, Aloÿs et al. (Sept. 2016). "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things." In: *Sensors* 16.9, p. 1466. DOI: 10.3390/s16091466 (cit. on pp. 9, 11).

Baron, Paul A. and Klaus Willeke (2005). *Aerosol measurement: principles, techniques, and applications*. John Wiley & Sons, Inc. ISBN: 9780471784920 (cit. on p. 16).

Bor, Martin and Utz Roedig (June 2017). "LoRa Transmission Parameter Selection." In: *13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. DOI: 10.1109/dcoss.2017.10 (cit. on pp. 9, 10).

Bor, Martin, John Vidler, and Utz Roedig (2016). "LoRa for the Internet of Things." In: *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*. EWSN '16. Graz, Austria: Junction Publishing, pp. 361–366. ISBN: 9780994988607 (cit. on p. 12).

Bosch Sensortec (Nov. 2020). *BME280. Combined humidity and pressure sensor*. Rev. 1.9. URL: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf (visited on 03/07/2021) (cit. on p. 25).

Cadex Electronics Inc. (Apr. 24, 2018). *BU-409: Charging Lithium-ion*. URL: https://batteryuniversity.com/learn/article/charging_lithium_ion_batteries (visited on 02/19/2021) (cit. on pp. 17, 18).

Castell, Nuria et al. (2017). "Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates?" In: *Environment International* 99, pp. 293–302. ISSN: 0160-4120. DOI: https://doi.org/10.1016/j.envint.2016.12.007 (cit. on pp. 1, 2).

Cattani, Marco, Carlo Alberto Boano, and Kay Römer (June 2017). "An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication." In: *Journal of Sensor and Actuator Networks* 6.2, p. 7. DOI: 10.3390/jsan6020007 (cit. on pp. 9, 10, 12, 13).

Chen, Min and Gabriel A. Rincon-Mora (Nov. 2006). "Accurate, Compact, and Power-Efficient Li-Ion Battery Charger Circuit." In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 53.11, pp. 1180–1184. DOI: 10.1109/tcsii.2006.883220 (cit. on pp. 17–19).

European Comission (Oct. 5, 2020). *Preventive solutions for Sensitive Materials of Cultural Heritage*. URL: https://cordis.europa.eu/project/id/814596 (visited on 01/14/2021) (cit. on p. 4).

European Environment Agency (EEA) (Oct. 2017). *Air quality in Europe - 2017 report*. URL: https://www.eea.europa.eu/publications/air-quality-in-europe-2017 (visited on 03/18/2021) (cit. on p. 13).

European Environment Agency (EEA) (2020). *Air quality in Europe - 2020 report*. URL: https://www.eea.europa.eu/publications/air-quality-in-europe-2020-report/at_download/file (visited on 01/12/2021) (cit. on p. 1).

Free Software Foundation (Oct. 24, 2020). *GDB: The GNU Project Debugger*. URL: https://www.gnu.org/software/gdb/ (visited on 03/14/2021) (cit. on p. 30).

Gamma, Erich et al. (Dec. 1, 1995). *Design Patterns*. Prentice Hall. ISBN: 0201633612 (cit. on pp. 30, 31).

Giechaskiel, Barouch et al. (Sept. 2013). "Review of motor vehicle particulate emissions sampling and measurement: From smoke and filter mass to particle number." In: *Journal of Aerosol Science* 67, pp. 48–86. DOI: 10.1016/j.jaerosci.2013.09.003 (cit. on pp. 14–16).

Giechaskiel, Barouch et al. (Dec. 2018). "Particle number measurements in the European legislation and future JRC activities." In: *Combustion Engines* 174.3, pp. 3–16. DOI: 10.19206/CE-2018-301 (cit. on p. 13).

GlobalSat WorldCom Corporation (2021). *LoRaWAN Sensor Device LS-113G*. New Taipei City, Taiwan. URL: https://www.globalsat.com.tw/files/LS-113G_PM2.5_%20.pdf (visited on 02/01/2021) (cit. on p. 7).

Horowitz, Paul and Winfield Hill (Apr. 30, 2015). *The Art of Electronics*. 3rd ed. Cambridge University Press. 1220 pp. ISBN: 978-0-521-80926-9 (cit. on pp. 17, 22).

Knoll, Markus, Philipp Breitegger, and Alexander Bergmann (Oct. 1, 2018). "Low-Power Wide-Area technologies as building block for smart sensors in air quality measurements." In: *e & i Elektrotechnik und Informationstechnik* 135 (6), pp. 416–422. ISSN: 1613-7620. DOI: https://doi.org/10.1007/s00502-018-0639-y (cit. on pp. 1, 2).

Kumar, Prashant et al. (2015). "The rise of low-cost sensing for managing air pollution in cities." In: *Environment International* 75, pp. 199–205. ISSN: 0160-4120. DOI: https://doi.org/10.1016/j.envint.2014.11.019 (cit. on p. 2).

Kuula, Joel et al. (May 2020). "Laboratory evaluation of particle-size selectivity of optical low-cost particulate matter sensors." In: *Atmospheric Measurement Techniques* 13.5, pp. 2413–2423. DOI: 10.5194/amt-13-2413-2020 (cit. on pp. 7, 53).

Liu, Sujuan, Chuyu Xia, and Zhenzhen Zhao (Oct. 2016). "A low-power real-time air quality monitoring system using LPWAN based on LoRa." In: *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*. IEEE. DOI: 10.1109/icsict.2016.7998927 (cit. on p. 7).

LoRa Alliance (2021). *What is LoRaWAN Specification*. URL: https://lora-alliance.org/about-lorawan/ (visited on 01/27/2021) (cit. on p. 6).

Molex (Feb. 24, 2014). *868/915MHz ISM Standalone Antenna*. URL: https://www.molex.com/pdm_docs/ps/PS-105262-001.pdf (visited on 03/08/2021) (cit. on p. 25).

Netvox Technology Co., Ltd. (2021). *Wireless PM2.5/Temperature/Humidity Sensor R72616A*. Tainan City, Taiwan. URL: http://www.netvox.com.tw/um/R72616A/R72616Adatasheet.pdf (visited on 02/01/2021) (cit. on pp. 7, 8).

Neumann, Pierre, Julien Montavont, and Thomas Noel (Oct. 2016). "Indoor Deployment of Low-Power Wide Area Networks (LPWAN): a LoRaWAN case study." In: *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. DOI: 10.1109/wimob.2016.7763213 (cit. on p. 41).

Park, Junhyun et al. (Jan. 2019). "Low Cost Fine-Grained Air Quality Monitoring System Using LoRaWAN." In: *International Conference on Information Networking (ICOIN)*. IEEE. DOI: 10.1109/ICOIN.2019.8718193 (cit. on pp. 6, 7).

RF Solutions (2021). *LAMBDA62. +22dBm LoRa 868/915MHz Transceiver*. URL: https://www.rfsolutions.co.uk/downloads/1615221116DS-LAMBDA62-6.pdf (visited on 04/13/2021) (cit. on pp. 25, 41).

Semtech Corporation (2015). *LoRa Modulation Basics - AN1200.22, Revision 2*. Ed. by USA) Semtech Corporation (Camarillo CA (cit. on pp. 9, 10).

Semtech Corporation (June 2019). *SX1261/2. Long Range, Low Power, sub-GHz RF Transceiver*. Rev. 1.2. Semtech Corporation: Camarillo, CA, USA (cit. on pp. 12, 13, 25).

Sensirion AG (Mar. 2020a). *Datasheet SPS30. Particulate Matter Sensor for Air Quality Monitoring and Control*. URL: https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9.6_Particulate_Matter/Datasheets/Sensirion_PM_Sensors_Datasheet_SPS30.pdf (cit. on pp. 15, 17, 24).

Sensirion AG (Mar. 2020b). *Sensor Specification Statement. Howto Understand SpecificationsofSensirion Particulate Matter Sensors*. Version 1. Stäfa, Zürich, Switzerland. URL: https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9.6_Particulate_Matter/Application_Notes/Sensirion_Particulate_Matter_AppNotes_Specification_Statement.pdf (visited on 02/15/2021) (cit. on p. 16).

Sira Certification Service (Jan. 8, 2020). *MCERTS - Product Confirmity Certificate. SPS30 Particulate Matter Sensor*. CSA Group. URL: https://www.csagroupuk.org/wp-content/uploads/2020/01/MC20035001.pdf (visited on 04/13/2021) (cit. on p. 24).

Snyder, Emily G. et al. (2013). "The Changing Paradigm of Air Pollution Monitoring." In: *Environmental Science & Technology* 47.20, pp. 11369–11377. DOI: https://doi.org/10.1021/es4022602 (cit. on p. 2).

South Coast Air Quality Management District (2019). *Evaluation Summary. Sensirion SPS30*. Tech. rep. Diamond Bar, CA, USA: Air Quality Sensor Performance Evaluation Center (AQ-SPEC). URL: http://www.aqmd.gov/docs/default-source/aq-spec/summary/sensirion-sps30---summary-report.pdf?sfvrsn=8 (visited on 03/07/2021) (cit. on pp. 24, 53).

STMicroelectronics (June 2019). *STM32L476xx. Ultra-low-power Arm® Cortex®-M4 32-bit MCU+FPU, 100DMIPS, up to 1MB Flash, 128 KB SRAM, USB OTG FS, LCD, ext. SMPS*. URL: https://www.st.com/resource/en/datasheet/stm32l476rg.pdf (visited on 03/05/2021) (cit. on p. 24).

STMicroelectronics (Feb. 2020a). *AN2867 Application note, Rev. 13. Using the hardware real-time clock (RTC) and the tampermanagement unit (TAMP) with STM32 microcontrollers*. URL: https://www.st.com/content/ccc/resource/technical/document/application_note/c6/eb/5e/11/e3/69/43/eb/CD00221665.pdf/files/CD00221665.pdf/jcr:content/translations/en.CD00221665.pdf (visited on 03/16/2021) (cit. on p. 39).

STMicroelectronics (Dec. 2020b). *AN4759 Application note, Rev. 5. Oscillator design guide for STM8AF/AL/S, STM32 MCUs and MPUs*. URL: https://www.st.com/content/ccc/resource/technical/document/application_note/c6/eb/5e/11/e3/69/43/eb/CD00221665.pdf/files/CD00221665.pdf/jcr:content/translations/en.CD00221665.pdf (visited on 03/16/2021) (cit. on p. 38).

TalkPool AG (2021). *OY1700 LoRaWAN Particles meter*. Chur, Switzerland. URL: https://talkpool.com/oy1700-lorawan-particles-meter/ (visited on 02/01/2021) (cit. on p. 7).

Texas Instruments (June 2015). *TPS6205x 800-mA Synchronous Step-Down Converter*. Texas Instruments: Dallas, TX, USA. URL: https://www.ti.com/lit/gpn/tps62051 (visited on 03/04/2021) (cit. on p. 23).

Vallero, Daniel A. (2008). *Fundamentals of Air Pollution*. Fourth. Academic Press. ISBN: 9780123736154. DOI: https://doi.org/10.1016/B978-0-12-373615-4.50043-1 (cit. on p. 1).

Voigt, Thiemo et al. (Nov. 2, 2016). "Mitigating Inter-network Interference in LoRa Networks." In: arXiv: 1611.00688 [cs.NI] (cit. on p. 44).

White, Elecia (Dec. 2, 2011). *Making Embedded Systems*. O'Reilly UK Ltd. ISBN: 9781449302146 (cit. on p. 36).

World Health Organization (2006). *Air quality guidelines. Global update 2005.* ISBN: 9289021926. URL: https://www.euro.who.int/__data/assets/pdf_file/0005/78638/E90038.pdf (visited on 01/11/2021) (cit. on p. 1).

World Health Organization (2016). *Ambient air pollution: A global assessment of exposure and burden of disease*, pp. 15–17. ISBN: 9789241511353. URL: https://www.who.int/iris/bitstream/10665/250141/1/9789241511353-eng.pdf (visited on 01/08/2021) (cit. on p. 1).

Wu, Fan et al. (Feb. 2018). "WE-Safe: A wearable IoT sensor node for safety applications via LoRa." In: *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. IEEE. DOI: 10.1109/wf-iot.2018.8355234 (cit. on p. 7).