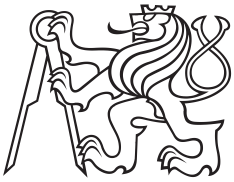
Bachelor Project



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Measurement

Sensor Network for Smart Agriculture

Jiří Maňák

Supervisor: prof. Ing. Radislav Šmíd, Ph.D.
May 2024

Acknowledgements

I would like to express gratitude to those who helped and supported me during the creation of this thesis.

First and foremost, I would like to thank my supervisor, Prof. Radislav Šmíd, for his insights, positive attitude, ideas and the material support.

Also my partner, Bc. Diana Varšíková, for her support, counsel of the visualizations and the final review pass.

Lastly, huge thanks goes to Ing. Tomáš Kořínek and Ing. Jan Spáčil from the department of electromagnetic field for their willingness, provided access to the instruments and overall great company.

Declaration

I hereby declare that I have independently authored the submitted thesis and have cited all the sources used.

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze 21. května 2024

Abstract

This thesis examines the development of a wireless sensor for soil moisture sensing, crucial for efficient water resource management in agriculture, horticulture, and domestic settings. Outcome of the work are two hardware components, a general purpose LoRa module and the soil moisture sensor, based on capacitive volumetric water content measurement. The module, built on the STM32WLE System on a Chip, features integrated non-volatile memory for data logging and firmware updates, compact size, low power consumption, and versatile I/O support, its firmware is implemented in the Rust language. Field tests confirmed the module's stable communication over distances exceeding one kilometer, demonstrating its effectiveness and potential for broader applications.

Keywords: Sensor Network, LoRa, The Rust Language

Supervisor: prof. Ing. Radislav Šmíd, Ph.D.
Praha, Technická 1902/2, místnost: A3-324

Abstrakt

Tato bakalářská práce se zabývá vývojem bezdrátového senzoru pro měření vlhkosti půdy, který je užitečný pro efektivní hospodaření s vodními zdroji v zemědělství, zahradnictví a domácnostech. Výsledkem práce jsou dva hardwarové komponenty, univerzální modul s podporou LoRa rozhraní a senzor vlhkosti půdy, založený na kapacitním měření objemového obsahu vody. Modul, postavený na čipu STM32WLE, obsahuje integrovanou nevolatilní paměť pro záznam dat a aktualizace firmwaru, je kompaktní, má nízkou spotřebu a univerzální podporu I/O. Firmware modulu je implementován v programovacím jazyce Rust. Experimentem byla oveřena stabilní komunikace modulu na vzdálenosti přesahující jeden kilometr.

Klíčová slova: Senzorová síť, LoRa, Jazyk Rust

Překlad názvu: Senzorová síť pro Smart Agriculture

Contents

Figures

Tables

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Maňák Jiří** Personal ID number: **499229**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Measurement**

Study program: **Cybernetics and Robotics**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Sensor network for Smart Agriculture

Bachelor’s thesis title in Czech:

Senzorová síť pro Smart Agriculture

Guidelines:

Design and implement a module for data acquisition from sensors used in smart agriculture with a LoRa interface based on a microcontroller from the STM32 family, including PCB. Implement Over-The-Air firmware update capability, design a transmission protocol suitable for prolonged node inactivity, implement a sensor network with multiple nodes, and evaluate the transmission parameters. Experimentally evaluate the possibility of powering the module from a solar cell for outdoor use.

Bibliography / sources:

1. firemní literatura ST microelectronics
2. LoRa® Devices: Smart Agriculture Real World Solutions, Semtech 2023

Name and workplace of bachelor’s thesis supervisor:

prof. Ing. Radislav Šmíd, Ph.D. Department of Measurement FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **16.02.2024** Deadline for bachelor thesis submission: _____

Assignment valid until:
by the end of summer semester 2024/2025

prof. Ing. Radislav Šmíd, Ph.D. Supervisor’s signature	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt	Student’s signature

Chapter 1

Introduction

With the upward trend in global temperatures, the world is witnessing an increased scarcity of water resources. This environmental shift poses a significant challenge to traditional farming practices, which are heavily reliant on predictable weather patterns and stable water supply.

"*Over 60% of European soils are unhealthy, and scientific evidence shows that soils are further degrading due to unsustainable management of the land, sealing, contamination, and overexploitation, combined with the impact of climate change and extreme weather events. [...] The loss of these essential soil ecosystem services costs the EU at least 50 billion euros per year.*", Soil health - European Commission [18].

In the face of these challenges, the adoption of smart agriculture technologies emerges as a vital strategy. These technologies offer the potential to transform agricultural practices by optimizing water usage, improving crop yields, and ensuring sustainable farming operations.

This project aims to develop a sensor network that is not only reliable and easy to maintain but also versatile enough to be applied across various contexts - from small–scale home gardens to medium–sized agricultural fields and regional deployments.

The technical challenges will be the primary focus of this work, and since many of the requirements for a reliable remote sensing solution are common amongst other applications, it will also venture into those, especially in the initial architectural and design stages.

1.1 Goals

Many options exist when it comes to transferring sensor data over long distances. For this work, the LoRa technology [38, 46] was selected for its unique blend of long-range (orders of 1 to 10 km in favorable conditions), albeit relatively slow (0.3 to 50 kbit [38]), data transfer, with very low power consumption (5 mA receiving [38]), global availability and general fit for the purpose [51].

Furthermore, an over–the–air (OTA) firmware update capability will be explored, a feature not commonly found in LoRa nodes. In the case of soil moisture sensors, an update can bring support for additional soil types, improve the accuracy or prolong the battery life.

OTA crucially allows for the update to happen without any operator intervention and transforms a one-shot solution, which once obsolete is replaced completely, into one that is, over time, able to support more sensors and features with minimal downtime [28, 3, 9].

A custom hardware solution will need to be developed in order to support OTA updates. Chapter 3, in particular, will make a point as to why this is necessary by compiling the hardware requirements and assessing the existing solutions currently available on the market.

This hardware solution will consist of two pieces - a general-purpose LoRa module (the "LoRa module" from now on) capable of OTA updates and a soil moisture sensor, which integrates the LoRa module into the specialized application.

To implement the sensing, its configuration, status reporting, and the update capability, a custom protocol will be developed atop the LoRa physical layer. This will make the system self–sufficient, but may limit the deployment options. However, the system could also be made compatible with the majority of commercial LoRaWAN networks in the future if needed.

On the note of autonomy and minimal maintenance, individual nodes should be power–efficient enough to be powered by a small built–in battery augmented by a solar panel, to achieve a "set and forget" deployment.

Lastly, the system will be evaluated based on its performance in the field by conducting a range test in actual operating conditions, and its sensor's ability to be powered by solar panel will be tested on a prototype unit.

Chapter 2

Prior Art

There are many commercially available solutions for soil moisture sensing. Offers range from generic hobby-grade to professional research-grade sensors of different form factors and utilize various methods of measurement. Following is a list of offerings approximately sorted by price and capabilities

- TEROS 54 and TEROS 10 from METER Group [26, 24]
- BL-5311 biSensor by Baseline [7]
- generic capacitive soil-moisture sensor available from various sources [10]

(a) : METER TEROS 10 **(b) :** Baseline BL-5311 **(c) :** Generic sensor

Figure 2.1: Examples of commercially available soil moisture sensors [24, 7, 10].

The Baseline BL-5311 is constructed from an FR4 material and a plastic box housing the electronics. It is designed to be buried 10's centimeters below the surface near the root zone. The sensor measures volumetric water content (measured volume is ≈ 0.08 L for the Compact variant and ≈ 0.21 L for the Original) using Time Domain Transmissometry (TDT); thus the sensor is galvanically isolated from the soil. The sensor communicates using a two-wire interface, by which it also receives power and is able to measure 5% to 100% soil saturation with water while also measuring the temperature of the soil.

2.1 LoRa Wireless Interface

LoRa (Long Range) is a low-power wireless communication technology and is one of the leading technologies used in the rapidly expanding field of the Internet of Things (IoT), particularly in applications requiring devices to send small amounts of data over long distances while conserving battery life.

Its modulation technique is known as LoRa modulation, which is derived from chirp spread spectrum (CSS) modulation [38]. It operates in the sub-gigahertz radio frequency bands, and it is typically used in a star–of–stars network topology in conjunction with LoRaWAN (Long Range Wide Area Network), which defines the network's communication protocol and system architecture.

LoRa is particularly noted for its excellent penetration in dense urban environments and indoor connectivity, compared to other technologies like Wi–Fi and cellular networks [46, 38, 37]. This capability, combined with its low power requirements and long-range, provides a distinct advantage in scenarios where alternative networking technologies might fail or be too costly.

2.1.1 Modules Implementing LoRa

Communications modules are a way to integrate the desired communications interface into an application. These modules usually offer a layer of abstraction above the interface they implement, if not also being used as the main processing unit.

The NUCLEO-WL55JC could be considered for this use-case [44], but it is perhaps too versatile to be practical. Mainly, it includes many options for connecting additional devices, increasing its footprint, and additionally integrating features, which are only necessary during development, potentially increasing power consumption.

2.1.2 LoRa Network Coverage

The LoRaWAN® specification is a Low Power, Wide Area (LPWA) networking protocol designed to wirelessly connect battery operated "things" to the internet in regional, national or global networks, and targets key Internet of Things (IoT) requirements such as bi-directional communication, end-to-end security, mobility and localization services. - What is LoRaWAN | LoRa Alliance. According to the LoRa Alliance, there are three network definitions.

- Network Operator is any LoRaWAN network aimed at openly monetizing connectivity or end-to-end services to third parties.
- LoRaWAN Open Communities are developer communities.
- LoRaWAN Private Networks are used by smart cities and businesses that roll out their own LoRaWAN Networks.

There are many providers that allow customers to connect and transfer data through their network. The model is similar to that of cellular networks, where each device receives a Unique Identifier (ID) and needs to conform to data limits. Global network operators include Orange, Meshify, Chirp Stack, and The Things Network (TTN) [51]. Amazon Sidewalk is only available in the US [4]. Czech CRA offers plans for customers starting at 200 CZK per month for 10 devices with a limit of 10,000 messages total [35].

This project chooses to host a private network since the network coverage is mostly limited to urban areas, brings many restrictions [49, 50] and an ongoing cost associated.

2.2 Soil Water Content

Soil is made up of a solid phase of minerals and organic matter, and the pores in-between the solids, which hold gasses and water [32]. The total amount of moisture is the sum of the moisture contained inside the solids (in intra-aggregate pore space) and the space around the solids (inter-aggregate pore space) [30]. This work will not distinguish between the two for simplicity.

2.2.1 Definition

Soil water or moisture content is a ratio that ranges from 0, meaning completely dry, to the value of material porosity at saturation [1]. It expresses the quantity of water contained in the soil. We can measure it by mass (gravimetric method) or by volume, as depicted in Figure 2.2.

Volumetric content (VWC) can be expressed mathematically as

$$\theta = \frac{V_w}{V_s + V_w + V_a} \tag{2.1}$$

where V_w is the volume of water, and $V_s + V_w + V_a$ is the total volume of the soil sample, including the contained air.

Likewise, gravimetric water content (GWC) is defined as

$$u = \frac{m_w}{m_s} \tag{2.2}$$

where m_w is the mass of the water and m_s is mass of all solids in the sample [14].

2.2.2 Methods of Measurement

Drying the Soil

Drying the soil sample in a drying oven is a direct method of measurement and is used as the reference method [1]. By weighing the sample and measuring its volume, then doing that

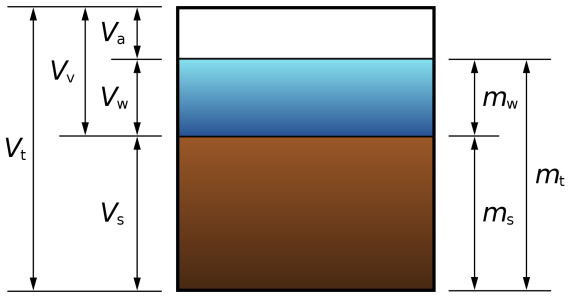


Figure 2.2: Soil composition by volume and mass [2].

again after drying, it is possible to very accurately measure both the volumetric and the gravimetric water content simultaneously.

The procedure involves gathering a known sample quantity, ranging from 30 grams to 5 kg, depending on the fines of the particles, heating it, and drying it at 65 to 110 degrees Celsius until the weight stops decreasing [11, 30, 32].

Since direct methods of measurement of soil moisture content are impractical for field use, we will focus on indirect methods from now onwards.

Geophysical Methods

Geophysical methods exploit other properties of water contained in the soil to approximate the VWC, such as its conductivity, dielectric constant, or interaction with neutrons. These methods are thus indirect and subject to inaccuracy if wrong assumptions are used [1]. However, if applied correctly, these methods allow continuous water content monitoring without human intervention.

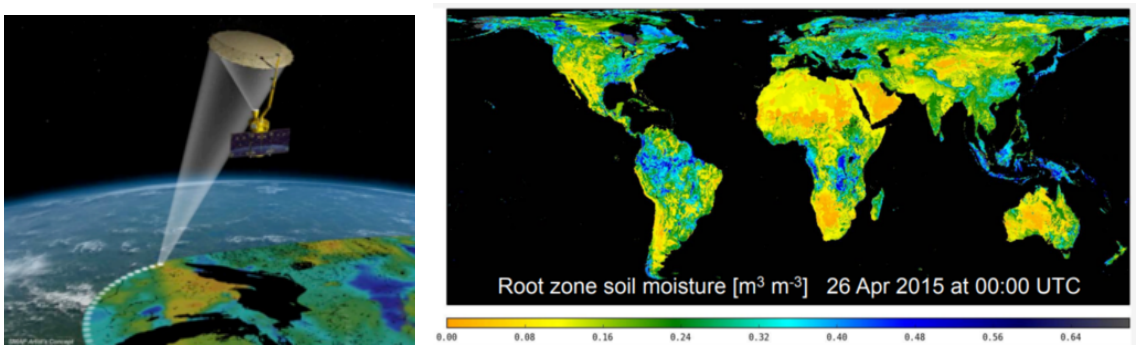
Satellite Remote Sensing Method

Thanks to recent and ongoing large-scale deployments of Synthetic Aperture Radar satellites, it is possible that this method will become much more widespread for global–scale soil moisture content estimation. It also relies on the large contrast in dielectric properties of wet and dry soil and generally exhibits too low resolution (>3 km [36, 16]) for most applications outside of large farming fields.

2.3 Capacitive Soil Moisture Sensors

Capacitive soil moisture sensing is an indirect, geophysical method of measurement, which makes it suitable for this application, as discussed earlier.

The working principle of capacitive soil moisture sensors is based on the dielectric permittivity of the soil. Water has a high dielectric constant compared to other soil components



(a) : SMAP Artist's Concept of the satellite

(b) : Global root zone soil moisture

Figure 2.3: Examples of commercially available soil moisture sensors [24, 7, 10].

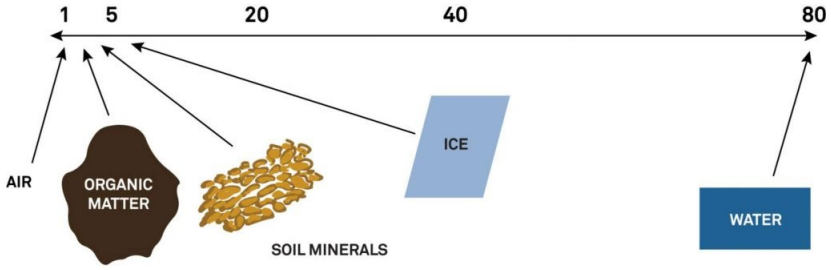


Figure 2.4: Dielectric constant ϵ_r of various materials. Source [25].

such as air, minerals, and organic matter, as illustrated by Figure 2.4. This property makes the presence of water in the soil significantly alter its dielectric permittivity.

A typical capacitive soil moisture sensor consists of two conductive plates, which function as electrodes. These plates are separated by a non–conductive material or are directly exposed to the soil, depending on the sensor design. The plates and the soil form a capacitor. When an alternating current is applied to the electrodes, an electric field is created between them.

As the volumetric water content in the soil increases, the relative dielectric permittivity of the soil also increases due to the high dielectric constant of water. This change in permittivity affects the capacitance between the plates. Specifically, the capacitance increases as the water content rises according to the equation

$$C = \frac{\epsilon_r \epsilon_0 S}{d} \tag{2.3}$$

where $\epsilon_0 = 8.85 \cdot 10^{-12} \frac{F}{m}$ permittivity of the vacuum, the surface area S and distance between the electrodes d are constant, the capacitance is proportional to the relative permittivity ϵ_r.

In practice, the dielectric constant is also largely dependent on temperature, measuring frequency, and saturation with dissolved ions [25, 36]. This makes precise absolute measurements almost impossible without prior calibration against known references for particular soil types.

To summarize, the fact that it can be employed without ongoing operating interventions by technicians and engineers, at the cost of providing only relative measurements and the need for initial calibration, this measuring method is a good compromise given the application requirements. Sections 3.4.1, 4.2.1, and 4.5, where the particular measuring method is designed and validated, expand upon this.

Chapter 3

System Architecture

3.1 Analysis of the System

As discussed in Chapter 1, the main goal of this work is the soil moisture sensor hardware with the accompanying firmware and a proof–of–concept application. With that being said, one can easily discern separate sub–tasks within this broader goal - mainly the fact that the communication aspect can be separated from the sensor itself (see Figure 3.1), a practice commonly seen in the industry. While this design does not specifically follow the IEEE 1451 [23], the common pattern is present nevertheless.

This design will go a step further and also separate the communication interface and the sensor on the hardware level, which will bring many advantages - future sensor implementations can be made with only effort being put into the sensor itself: the wireless part, and most of the development and testing overhead can be solved once, not repeated for every sensor type. Also, once a sensor compatible with the interface exists, it can be made compatible with future versions of the interface.

While this thesis is mainly concerned with the soil moisture sensing application, having a LoRa-compatible unit, which is capable of OTA updates and has enough processing power to handle most sensing and simple control tasks while being power–efficient enough to be battery-powered, is an interesting sub–goal of this work. The following sections will go through the process of finding requirements for such hardware and explain the compromises made.

3.2 LoRa Module Requirements

3.2.1 Over–the–air update support

To re-iterate the Introduction 1: *An update can bring support for additional soil types, improve accuracy, or prolong the battery life. It also, crucially, allows this to happen without any operator intervention and transforms a one-shot solution into one that is, over time, able to support more sensors and features with minimal downtime.*

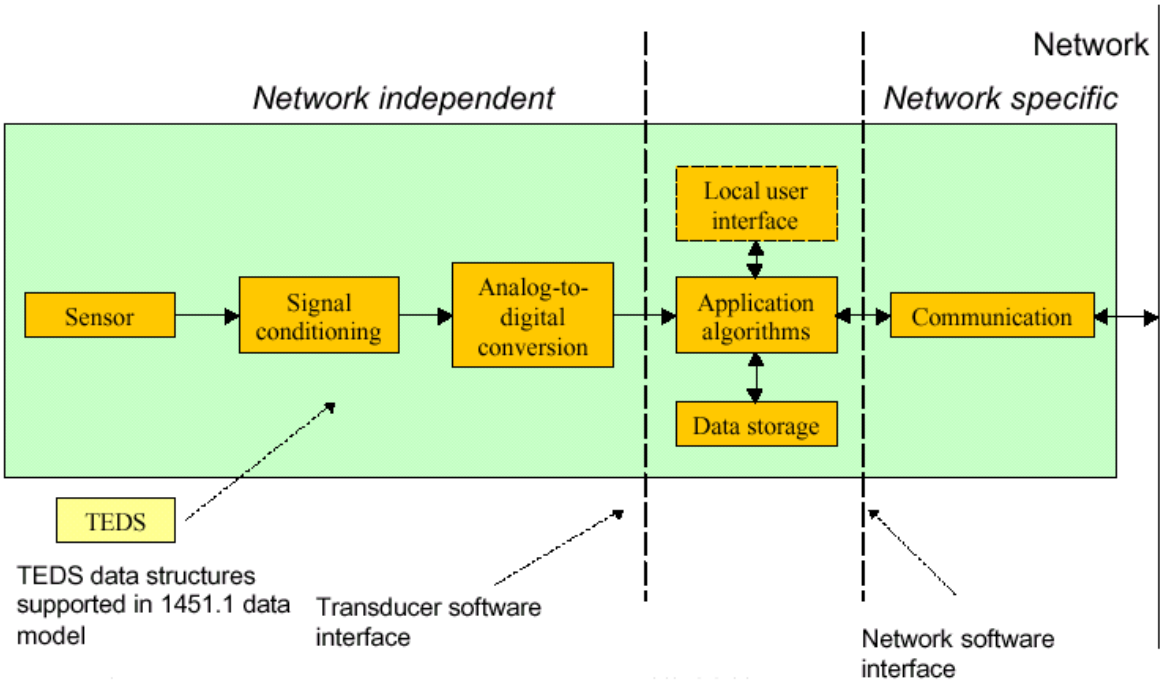


Figure 3.1: Logical high-level building blocks of smart sensors, as defined by IEEE 1451.

There are two general approaches to implementing OTA updates [9, 3]. The first approach is to split the firmware into a so-called Bootloader, a binary that runs before the main application and implements a way to communicate with the master node such that it is able to overwrite the application. The application usually also needs to communicate with the master node in order to transmit sensor data and accept commands, which leads to duplication of all the driver and protocol logic required to use the communication interface.

The first option provides some level of separation between the application and the Bootloader, which could be useful in cases where the application is in some ways "untrusted", either due to security measures limiting the potential impact of bugs or because it is actually provided by a different vendor and so on. This could be especially valid for cases where the processor provides privileged execution modes etc.

But it also cuts the other way since this approach hinders the upgradeability of the device itself because now this relatively large Bootloader with all the error-prone interface handling is locked in and unable to be updated remotely, unless, of course, the application also implements a way to update the Bootloader, but this defeats any potential security benefits unless a sophisticated authentication method is employed in both the Bootloader and the Application, which leads to even more duplication.

This brings us to the second option, which also consists of a Bootloader and the Application, but here the Bootloader is much simpler, it only implements the bare minimum that is required to take a newer version of the application and overwrite the currently active one, see Figure 3.2. However, if the Bootloader is unable to communicate with the master node, this means that the Application needs to download the new binary and store it somewhere locally on the end–node.

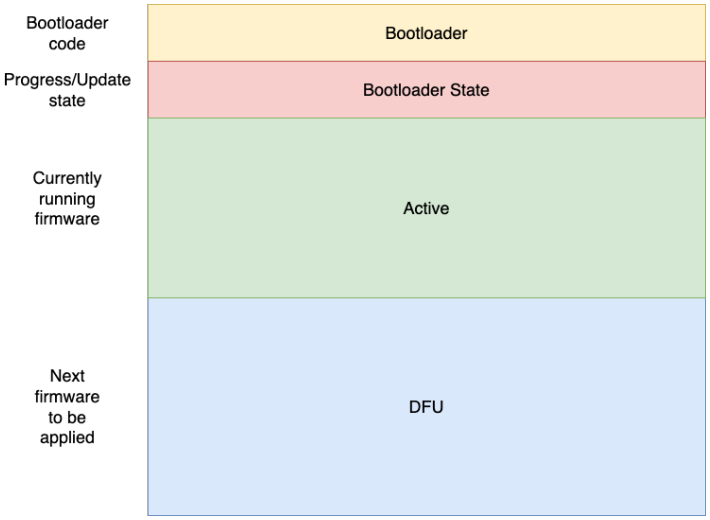


Figure 3.2: The bootloader divides the storage into 4 main partitions.

This is usually a challenge for embedded devices, where the built–in FLASH memory, in which the application is stored, is generally just about able to fit the Bootloader and the Application and rarely would be able to accommodate two copies of the binary.

Integrating additional memory increases the cost, but it can be used for other purposes besides OTA updates, such as storing configuration, calibration data, and logs, or if the application integrates some sort of graphical display, may be required to store the assets, such as images and fonts.

The most important benefit is that considering proper Bootloader support [13, 15], the application binaries can be swapped instead of deleting the old binary, which opens the possibility of rolling back unsuccessful updates. Such a possibility could be considered a crucial feature in some applications and would not be practical to implement using the first OTA approach presented above.

To summarize, the LoRa module should have enough non–volatile memory available to perform updates and stay relevant for future iterations of the firmware. Section 4.3 will describe the implementation of the update OTA protocol.

3.2.2 Potential LoRa Module Applications

In order to find the optimal boundary between the sensor implementation part and the Interface and Processing part, as defined in Figure 3.1, it is useful to look at the possible applications of the proposed module.

Generic use-case of the LoRa module - Indoor environment sensors

Let us consider this basic, typical use–case for such a communication module. This application can implement the following sensors: a thermometer, hygrometer (relative humidity sensor),

human presence detector, air quality sensor (CO^2e, smoke, ...), and a light sensor.

We can omit some of the listed sensors in the actual application, but the generic LoRa module should be able to support the full configuration without any kind of co–processor. The main limiting factor will probably be the number of communication peripherals and general-purpose input-output (GPIO) pins.

The thermometer is usually an integral part of any hygrometer measuring relative humidity [1], which is also suitable for this application. These sensors are frequently found in fully integrated solutions with a digital interface of some sort, usually I2C [8]. The same is true for any modern light sensor, which will also be able to measure intensities of different wavelengths of light [41], [48]. Thus more than half of the sensors listed only require a single I2C port to control them comfortably.

Traditionally, PIR sensors are used to detect motion, thus the presence of humans in the vicinity of the sensor, but this might not work reliably for indoor applications. Thus, nowadays, the use of radar-based systems [20] or IR ranging sensors [43] is a lot more prevalent for human presence detectors. Such a sensor might expose a digital interface, such as I2C or SPI, or simply output an analog signal, which can be sampled using an ADC.

Other environmental sensors, such as air quality sensors, also implement similar interfaces - I2C or SPI or an analog output. Notably, these sensors usually exhibit relatively high power draw ($>$ 100 mW) and slowest startup times of all the other sensors of this application (orders of 10s of seconds to minutes) [5], so they are not suitable for battery-powered applications.

On the note of power draw, this application may wish to be battery-powered or remain mains-powered; this will affect the capabilities and the end use case. When running on battery, the active–time is limited to a periodic sampling of the environment a handful of times per hour. Being able to power down all sensors can prove useful in this application to greatly improve the battery life. On the other hand, if the application aims at fast reaction times, switching on the lights when presence detected for example, and the inclusion of all the sensors listed, it will need to be mains powered to be practical.

Generic module use-case - Light dimmer

For this generic application of the LoRa module, only a timer peripheral capable of generating PWM of sufficient frequency and resolution on a handful of channels is needed. Such peripheral exists on most modern microcontrollers.

If local control is also required, a rotary encoder, for example, can be sampled using a digital input interrupt or a dedicated peripheral designed to handle encoders.

Generic module use-case - Soil moisture sensor

The defining features of this application are outdoor use, battery power with the possibility of including a solar panel for zero–maintenance operation, long range, and low dynamic duty–cycle.

Generic module use-case - Gateway

The module should be versatile enough to be also able to act as a communication interface for a host computer to connect to and manage the network of sensors, though other more specialized hardware could also be used for this use–case.

3.2.3 Final Requirements

The following list of specs was derived from the above use-cases of the generic LoRa module. These requirements should ensure that the LoRa module will be able to handle a broad range of tasks, including the soil moisture sensor implementation.

- 2.8–3.3 V nominal voltage range - the lower the minimum threshold, the better - for being able to harvest as much energy as possible from ie. a coin–cell battery
- low power design - support for switchable power rails for standby modes, low duty cycle operation, low power standby of the module itself
- target the 865–923 MHz (EU868, US915, IN865, ...) frequency range
- wide temperature range for outdoor applications
- support for a wide range of use–cases - minimize the amount of specialized hardware on the module
- support for OTA updates, which requires large enough internal storage
- integrated RF - ideally a built–in antenna or some means to connect one
- host communication interface
- minimal footprint
- low cost

Existing Hardware Satisfying These Requirements

SeedStudio Wio-E5-LE [46, 37] is a cost effective LoRa module integrating the STM32WLE5JC SOC, however it lacks the additional FLASH memory and is thus unsuitable.

3.3 LoRa Module Architecture and Parts Selection

Solderable PCB modules are a standard way of integrating existing solutions into custom ones. Modules providing wireless connectivity in particular are very common, see Figure 3.3.

This approach allows us to separate the usually complex and more expensive multi–layer board layouts required by modern SOCs, along with their power delivery and any other

(a) : Omega 2S [31] **(b) :** RN4871 [27] **(c) :** MAX F10s [52]

Figure 3.3: Examples of common wireless modules.

supporting circuitry, from the less complex end–application consisting of local power regulation, battery management, connectors, and other mechanical features.

In order to satisfy the requirements 3.2.3, the module should provide means for analog and digital signal acquisition, digital communication interfaces, and sufficient internal storage along with implementing wireless connectivity.

3.3.1 Microcontroller

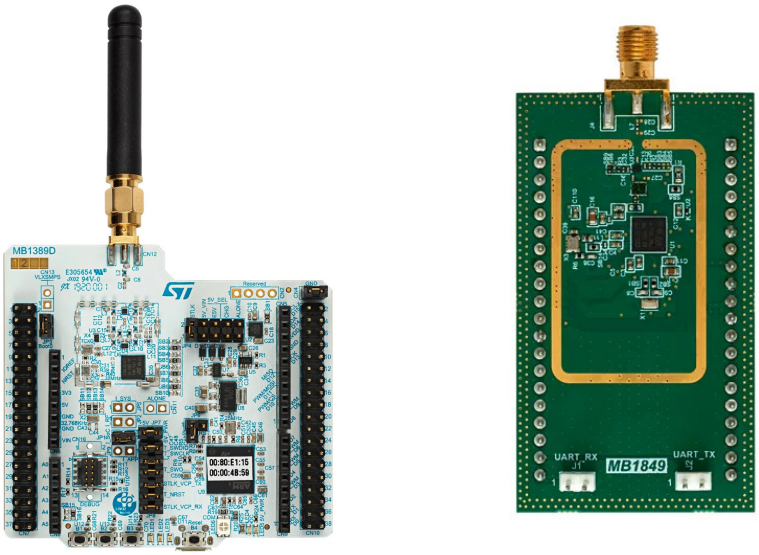
Given the requirements for minimal footprint a fully integrated System–on–a–Chip (SOC) solution is preferred to a separate MCU and an RF solution. STM32WL series offers such a SOC, which also satisfies the requirement of low power consumption by being based on the STM32L4, a well-known ultra-low power family of microcontrollers [40].

The manufacturer also offers a development board, the NUCLEO–WL55JC (3.4a), and a plethora of reference designs, the STDES–WL5xxxxx series, where the STDES–WL5U4ILH (3.4b) overlaps well with our requirements.

Each of the designs focuses on optimizing different parameters depending on the application priorities and their codification follows table 3.5, from which the defining features are apparent. For this work, application footprint was one of the top priorities, so the lower–power 15 dBm version with IPD was selected.

IPD stands for Integrated Passive Device; it consists of a balun and a harmonic filter. This circuitry is usually realized using passive components, which is cheaper but takes up more board space (see Figure 3.6 for comparison) and is also more prone to design mistakes and tuning mismatch. This particular IPD was specifically designed for STM32WL line of microcontrollers in the configuration defined by the reference design [42].

One downside of using an IPD in this case is that it requires the use of an RF switch because ST's IPDs are designed to work with separate receive and transmit paths. Again, this is a tradeoff between complexity, cost, and board space required. Fortunately only a Single Pole Double Throw switch is required, thus a simple switch such as the BGS12SN6 from Infineon [19] is sufficient.



(a) : NUCLEO-WL55JC [44] **(b) :** STDES-WL5U4ILH [45]

Figure 3.4: Nucleo development kit and the reference design board.

A 4 layer reference design was selected because it is anticipated it will allow a higher-density layout, shrinking the module dimensions even further. UFQFPN package is preferred over UFBGA to stay compatible with lower-cost manufacturing solutions. It is expected the design will use most of the available pins on the package; a complex fanout would be required if we were to go with the BGA variant; perhaps some space savings could be had at the cost of losing access to most of the pins, making any modifications to the design difficult to impossible, should the need arise.

In conclusion, the MCU choice is a culmination of tradeoffs, where the prevention of mistakes, efficiency, and familiarity were more important than the absolute performance and cost. This should not in any way hamper further improvements in succeeding versions of this hardware while allowing the completion of Proof–of–Concept stages of this project.

3.3.2 Power Regulation and Standby Mode Solution

The selected MCU supports a wide input voltage of 2–3.6 V thanks to its internal voltage regulation. It supports two modes - LDO, a linear regulator, which does not require any additional external component at the cost of lower efficiency, and the SMPS mode, utilizing a synchronous buck regulator, which is more efficient - this application will, therefore, attempt to implement the latter.

A separate, switchable power–rail is a feature deemed necessary by some considered applications in 3.2.2. This can be achieved using a built–in MOSFET - such a power rail could be used to save power even by powering off some parts of the module itself.

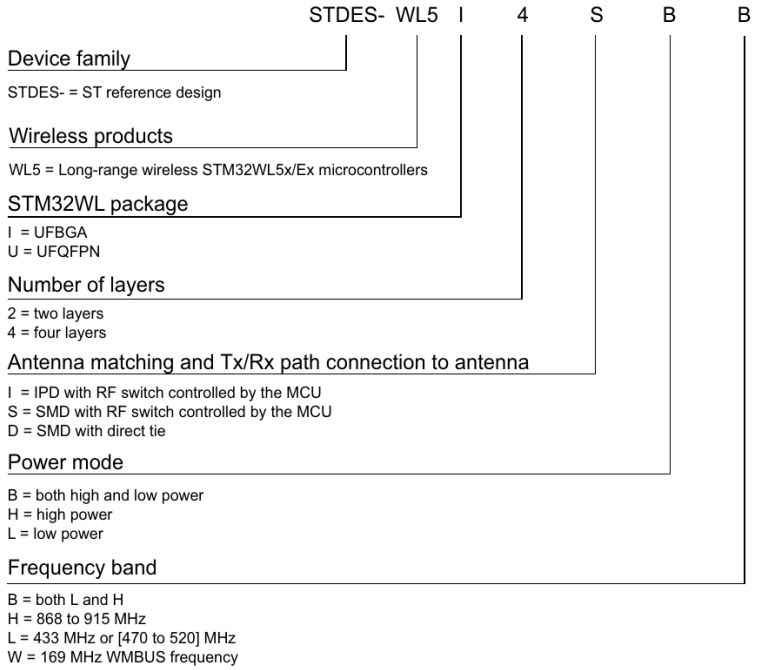


Figure 3.5: STM32WL5x and STM32WLEx reference designs codification.

3.3.3 Non–volatile Memory

As outlined in the OTA update support Section 3.2.1, external memory is needed to store the application binary during the download process. This memory does not need to be non–volatile, but since it is comparatively cheaper than RAM and provides a backup in case of a power loss, the choice is apparent.

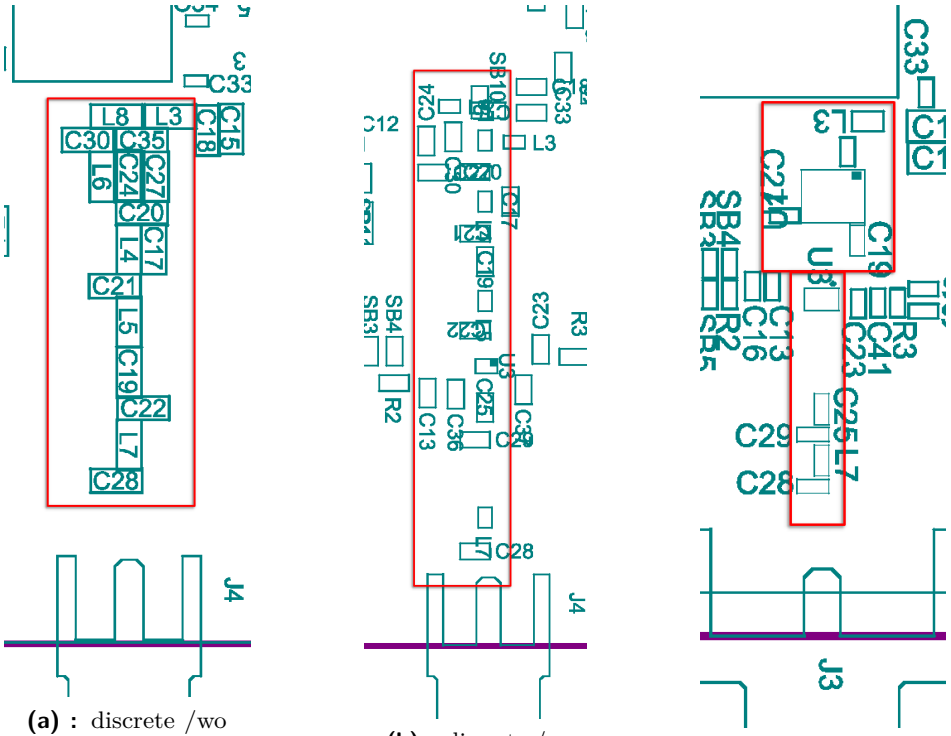
LoRa data rates are in the realm of 100–1000 bytes per second, even less in real–world applications, thus basically any available interface can be used. The capacity needs to be at–least equal to the built–in FLASH memory of the MCU 3.3.1, which is 256 kB.

The SST26VF080A 8 Mb FLASH memory was selected. It is available in the WDFN–8 package, which is space–efficient, features a standard SPI interface, and supports a wide input voltage range of 2.3–3.6 V, which was an important consideration for this use–case.

3.3.4 Antenna

In the initial module requirements 3.2.3 it was deemed preferable to integrate the antenna onto the module itself. This should not, however, compromise the usability and performance of the module.

In general, there are two approaches for integrated antennae - a trace antenna constructed using the PCB directly or an antenna in the form of a solderable component, such as a ceramic chip antenna. If the board space is limited, the performance is insufficient or the use of an integrated antenna is prohibited by any other limitation, the application must resort to



(a) : discrete /wo RF switch (variant D)

(b) : discrete /w RF switch (variant S)

(c) : IPD (variant I)

Figure 3.6: Frontend layout and part selection comparison, components on the RF path highlighted. Reference designs selected for comparison were picked based on their similarity with the STDES–WL5U4ILH design.

an external antenna connected to the RF circuitry through a coaxial cable. The table 3.1 provides a summary.

Antenna types	**Advantages**	**Disadvantages**
PCB antenna	Low cost, No assembly, Good performance achievable, Small size at high frequency	Difficult to design small and efficient antenna, Large size at low frequency
Chip antenna	Small size, Off-the-shelf solution	Medium performance, Medium cost
Whip (external) antenna	Good performance, Off-the-shelf solution	High cost, Large size

Table 3.1: Summary of advantages and disadvantages of different antenna solutions according to [6].

The following sections elaborate on each antenna type in more detail in the context of this particular application.

At this point in the design process, the module proportions, derived from the individual component footprints, are not expected to surpass 500 mm^2. It is also expected the design will need to conform to the PCB parameters specified in Table 3.2. Since the first prototypes will only operate in Europe, the aim is mainly the EU868 band.

Parameter	Value
Stackup thickness	1.6 mm
Substrate dielectric constant	4.5 [–]
Substrate loss tangent	0.02 [–]

Table 3.2: PCBWay manufacturer stackup parameters [33].

Printed Circuit Board Antenna

Due to the limited board space, this option is expected to not be feasible. Nevertheless, some types of PCB antenna solutions were calculated to support this intuition.

A microstrip patch antenna is one of the simplest types of PCB antenna designs [54, 53]. This antenna consists of the patch itself and its feed–line with an optional inset for impedance matching, see 3.7a.

The approximate parameters presented in Table 3.3 were obtained using [54]; the input impedance could be tuned using the inset and a matching circuit, but more importantly, this simple antenna design is not practical because of its size, which already has roughly 15 times the surface area of the module components.

Another, more space–efficient option, is to use the inverted–F design, see Figure 3.7b for its simplest form. We can calculate its length L given the target frequency f

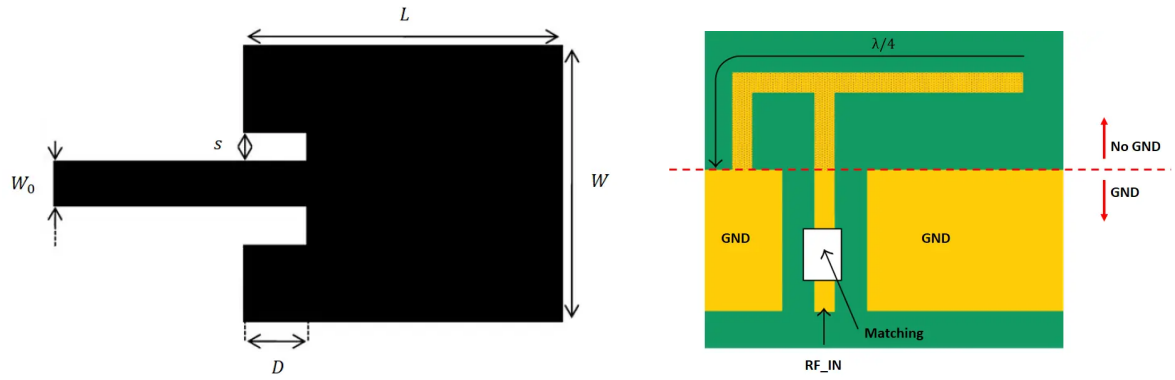
$$L = \frac{\lambda}{4} = \frac{c_0}{4f} = \frac{c_0}{4 \cdot 868 \cdot 10^6} \approx 86.3 \text{ mm} \tag{3.1}$$

which is a great improvement compared to the patch antenna.

This design could perhaps be optimized such that it would fit the size constraints, but such work might be enough to write another thesis focused on just this detail. Dimensions based on the layout in Figure 3.7b were included in Table 3.3 for comparison sake.

Parameter	Patch antenna	Inverted-F antenna
Input impedance	306 Ω	not calculated
Bandwidth	3.92 MHz	not calculated
Width (W)	104 mm	≈ 70 mm
Length (L)	80 mm	≈ 18 mm
Minimum footprint	8320 mm^2	1260 mm^2

Table 3.3: Approximate microstrip patch antenna parameters and dimensions (as described in 3.7a) and a notion of the dimensions of an inverted–F antenna (corresponding to 3.7b), not including the ground–plane and matching network.



(a) : Microstrip patch antenna dimensions including the feed-line [54]

(b) : Basic inverted-F antenna design [34]

Figure 3.7: Simple Printed Circuit Board antenna designs.

These preliminary calculations led to the conclusion that a PCB antenna at this frequency with these size constraints is not viable in this case. These calculations are not even beginning to factor in all the other components required, not to mention the even larger ground plane that such an antenna would require.

This could be made possible with significant design effort and expertise or by licensing an existing design, but the overall goal of creating this hardware would need to shift as a result.

Chip Antenna

As outlined in 3.1, the chip antenna is another logical step in the search of the right solution. It is a common choice seen in commercially available hardware, especially in cases where the space is at a premium - as can be seen on the RN4871 3.3b Bluetooth module.

There is a large selection of these antennae on the market, mainly targeted at the very common bands, such as 2.4 GHz and 5 GHz, but solutions for the 868 MHz frequency range are also not hard to come by [12, 29].

The search was narrowed by focusing on omnidirectional and linearly polarized antennae of small size and straightforward implementation guidelines. Cost also played a factor since many solutions, although good, would make the design unfeasible. The following Table 3.4 features the main candidates that were picked. All have 50 Ω impedance and satisfy the above criteria.

External Antenna

It was deemed necessary to rely on an external antenna instead of an integrated on–module antenna. This choice was made primarily because of the ground plane requirements of the integrated solutions, which is a downside of using lower frequencies in general.

Size of the ground–plane has a large affect on the antenna performance and its tuning. Due

Parameter	Linx ANT868[22]	Kyocera M6207[21]	Taoglas ILA08[47]
Construction	Wire wound	Ceramic chip	Ceramic chip
Rated frequency	862–876 MHz	863–870 MHz	863–870 MHz
Peak gain	5.6 dB	0.3 dB	−0.5 dB
Average gathering	−1.1 dB	−5.0 dB	−4.0 dB
Maximum VSWR	2.2	1.7	1.9
Dimensions [mm]	25.4 × 15.3 × 8.9	6.0 × 2.0 × 1.0	5.0 × 3.0 × 0.5
Ground plane [mm]	84.0 × 38.0	100.0 × 40.0	80.0 × 40.0

Table 3.4: Comparison of the three selected chip antennae.

to the nature of a communications module, which could be used in different applications, these requirements would need to be closely followed by the application; failing that could cause the module to radiate outside of the intended frequency range.

External antenna has the major advantage of being self-contained–thus, thus the implications on the circuit board design are minimal, and the responsibility is shifted onto the placement of the antenna, providing enough clearance and using RF–transparent materials.

Another advantage of its use in soil moisture sensing is that the antenna can be mounted higher above the ground level, which could improve its performance significantly if long-range transmission is required.

3.3.5 Conclusion

During the component selection phase, the requirements set in 3.2.3 were continually assessed and successfully met. One major compromise was made in the antenna selection, where the design needed to resort to an externally connected antenna due to the size and application constraints. This choice, however, should ensure that the potential for problems arising during prototype stages should be limited.

3.4 Sensor Architecture

Given that the primary point of this work is to create a proof–of–concept sensing solution, which utilizes the LoRa module envisioned in 3.3, the sensor should be suitable for small–scale testing and prototype work, it should also be simple to manufacture and maximize the potential for success in the first iteration, due to the limited time allocated for the whole project.

A single board construction is suitable, as it is simple, self–contained and can be made through standard PCB manufacturers. It is also commonly seen in hobby grade and some commercial soil moisture sensors.

3.4.1 Calculating the Sensor's Capacitance

The FR4 material, from which the sensor will be built, itself has a non–negligible dielectric constant of around 4.5 [33], which means that the measurement of the PCB should be minimized to avoid it being the majority of the measured capacitance. For this reason, the capacitor plates should be placed only on one side of the board, with all other layers devoid of copper.

To get a *very* rough estimate of the capacitance using equation 2.3 the dimensions are needed. The sensor is expected to have an active area of about 150 mm by 30 mm divided into 4 zones. Each zone thus has 1125 mm^2, assuming an interleaved structure of the one–sided capacitor with 5 tabs for each electrode, the average distance could be 2 mm, which translates to some 10–500 pF[1].

3.4.2 Measuring Method

It was decided to use the simplest method possible - measuring the time constant of an RC circuit. As discussed in Section 2.3, in order to meaningfully measure soil moisture, it is necessary to employ advanced methods of capacitance measurement and integrate them into a small and power-efficient circuit.

Not only that, but it needs to be stable across a very large temperature range, to measure at high frequencies (>50 MHz [25]) and be able to do so at multiple depths. Because of these reasons, methods based on capacitor value comparison were rejected, and any kind of method that relies on generating a reference sine wave was also dropped.

The proposed solution is to use the GPIO of the MCU to control a simple charge–discharge circuit muxed into the active zones of the sensor, combined with a comparator to trigger the stop of a time measurement.

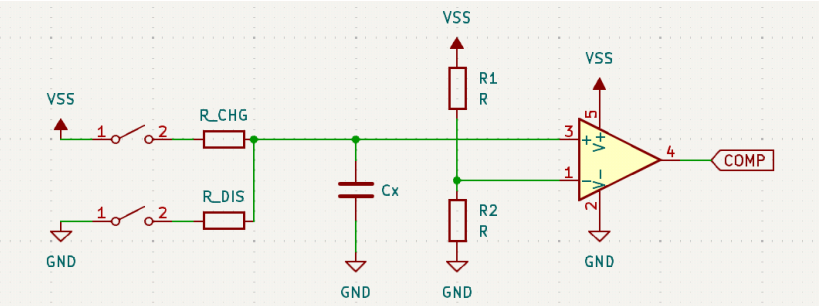


Figure 3.8: The proposed capacitance measuring solution. This circuit is able to estimate the capacitance of C_X by measuring its time constant, given a known R_{CHG} and the comparator $V-$ voltage, set by resistors R_1 and R_2. The R_{DIS} is used for discharging the capacitor prior to measurement. Switches are to be controlled by the MCU.

[1] $C \approx \frac{0.15/4*0.03*4.5*8.85*10^{-12}}{0.002} = 22$ pF at $\epsilon_r = 4.5$, $C \approx \frac{0.15/4*0.03*80*8.85*10^{-12}}{0.002} = 400$ pF at $\epsilon_r = 80$, according to Equation 2.3, which is normally used for calculating the capacitance of two parallel plates. Here, it is used to provide an orders-of-magnitude estimate for designing the circuit.

Chapter 4

Results

4.1 LoRa Module Design

As discussed in section 3.3, where the LoRa module was architected, and its main parts were selected, the schematic and board layout was created. These outputs are included in image form as Appendix B, along with more details, and available at github.com/manakjiri/lora-module-hw[1].

The Open source KiCad Electronic Design Automation (EDA) software was used throughout the project to create these designs.

4.1.1 Schematic

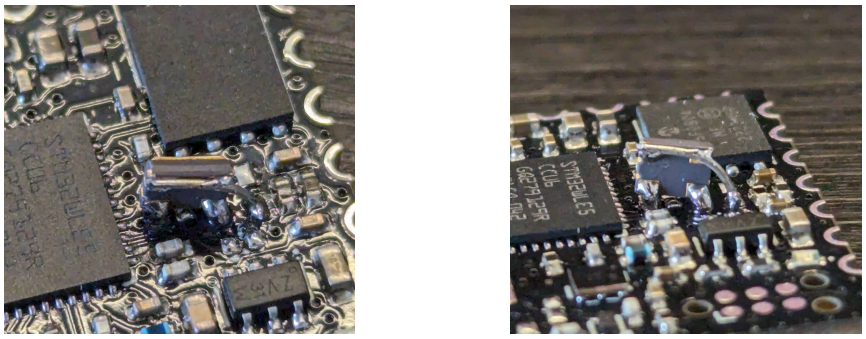
While the 3.3 focused on the fundamental parts of the design, many details were left up to the later development stages once the overall system implementation was clear. This section will focus and expand on these parts of the design.

Of note is the selection of the main clock source for the MCU 3.3.1. In this case, the only two options are to use a Crystal oscillator (XO) or a Temperature Compensated XO (TCXO). Application note AN5646 (STMicroelectronics [39]) summarizes the main differences as

- An XO is more efficient on power consumption, startup time, and BOM cost.
- A TCXO is more efficient on frequency accuracy and frequency variation over temperature changes. It also removes layout constraints.

This aspect was not considered carefully enough during the parts selection, the recommended NX2016SA series crystal oscillator was selected for the power savings and reduced cost. However, the selected programming framework, Embassy, and the library `lora-rs` in particular (expanded upon in Appendix A), only correctly supported the use of a TCXO, at the time of the prototype bring–up. This led to an ad hoc modification, documented in Figure 4.1, of the module and the addition of the Abracon ATX–11–F series TCXO to fix this issue.

[1] https://github.com/manakjiri/lora-module-hw/releases/tag/v0.1



(a) : top view **(b) :** side view

Figure 4.1: The ad hoc TCXO modification on the LoRa module.

All other aspects of the MCU integration were executed according to the application notes and the datasheet [40, 39], while closely following the implementations of the reference design and the nucleo development kit [45, 44]. This includes the selection of decoupling capacitors, the SMPS circuitry, and the reset handling.

Another consideration was the implementation of the switchable power rail VDD_SW. This rail taps off the main power rail of the rest of the module; any disruption could cause a glitch or trigger the BOR protection circuitry. It is thus necessary to limit the inrush current caused by this rail's switch–on and subsequent charging of any local capacitances. An RC circuit was designed to slow–down the gate voltage rise–time to about 10 ms.

4.1.2 Printed Circuit Board Layout

A 4-layer board stackup was proposed in Section 3.3.1, which indeed ended up being used in this design. The purpose of each layer is described in table 4.1 and clearly observable in the final renders B.5.

Layer name	**Primary purpose**
F. Front layer	Components and local connections
1. First inner layer	Ground
2. Second inner layer	Power
B. Back layer	Signal and markings

Table 4.1: Module board layer signal and power assignments.

The board is only populated on the front side, see Figure B.6, to allow its use as a solder–able module. Still, the final dimensions of the module are 20.32×22.48 mm, which is better than the initial optimistic estimate given in Section 3.3.4.

To stay compatible with low–cost manufacturing options, conservative parameters were picked when it comes to minimal clearance, trace width and drill size. No density–increasing technologies, such as blind or buried vias, via–in–pad, micro–via, etc. were employed either.

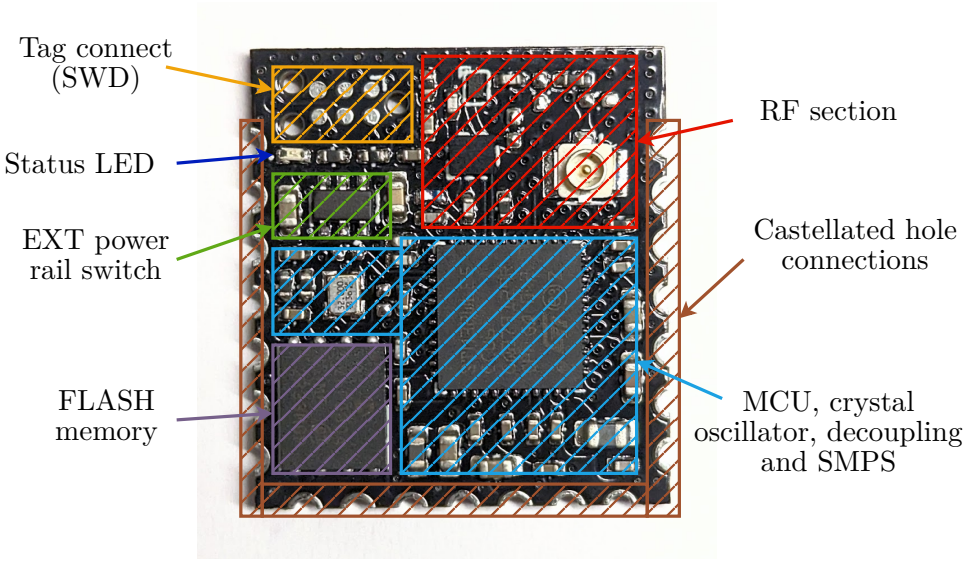


Figure 4.2: Image of the manufactured and fully assembled module with overlay highlighting its functional parts.

These parameters are summarized in the following Table 4.2 and were enforced by the Design Rule Checker (DRC) throughout the project.

Parameter	Dimension
Minimum trace clearance	0.15 mm
Minimum trace width	0.15 mm
Minimum via width	0.3 mm
Hole to trace clearance	0.3 mm
Hole to hole clearance	0.5 mm
Board edge to trace clearance	0.15 mm

Table 4.2: Board layout physical limits.

Being only 4 layers, the traces needed to be routed in a densely and well thought-out manner, attempting to minimize the number of relatively large vias required. For this reason, the module's external connection signal assignments were decided only near the end of the design stage, conforming mostly to the existing pin locations on the MCU itself. The final pad assignments are included in Table 4.4.

4.1.3 Finalized LoRa Module Specification

Figure 4.2 shows the manufactured and fully assembled module with overlay highlighting its functional parts; its technical specifications are captured by Table 4.3.

Supply voltage range	2.3–3.5 V
Maximum supply current (excluding EXT)	65 mA
Operating temperature range	−40–85 °C
Output RF power	15 dBm
Operating band	868 MHz
RF connector	U.FL
Module connection type	Castellated hole (0.1 inch pitch)
Supported interfaces	UART, SPI, I2C
Programming interface	ARM Serial Wire Debug (SWD)

Table 4.3: Final module specification.

IO	Pin	TIM[a]	ADC	I2C	SPI	UART	Other
1	PA7	17_1, 1_1N		3_SCL	1_MOSI		CMP2_OUT
2	PA6	16_1			1_MISO		
3	PA4	L1,2_OUT					RTC_OUT2
4	PA2	2_3				2_TX	CMP2_OUT
5	PA1	L3_OUT, 2_2			1_SCK		
6	PA0	2_1					WKUP1
7	PB8	16_1, 1_2N		1_SCL			
8	PB7	L1_IN2, 17_1N		1_SDA		1_RX	
9	PB6			1_SCL		1_TX	
10	PB5	L1_IN1			1_MOSI		CMP2_OUT
11	PB4		3	3_SDA	1_MISO		CMP1,2_INP
12	PA11	1_4	7	2_SDA	1_MISO		CMP1,2_INM
13	PB3	2_CH2	2		1_SCK		SWO, WKUP3
14	PA13		9				SWDIO
15	PA14	L1_OUT	10				SWCLK
16	PH3						BOOT0

Table 4.4: Module pin legend including feature summary.

[a]Refer to each column following “Pin” (excluding “Other”) as [Column header][Column-row contents], such as “SPI1_MOSI” and so on. Some features were omitted for clarity, for the complete list refer to the MCU manufacturer’s documentation

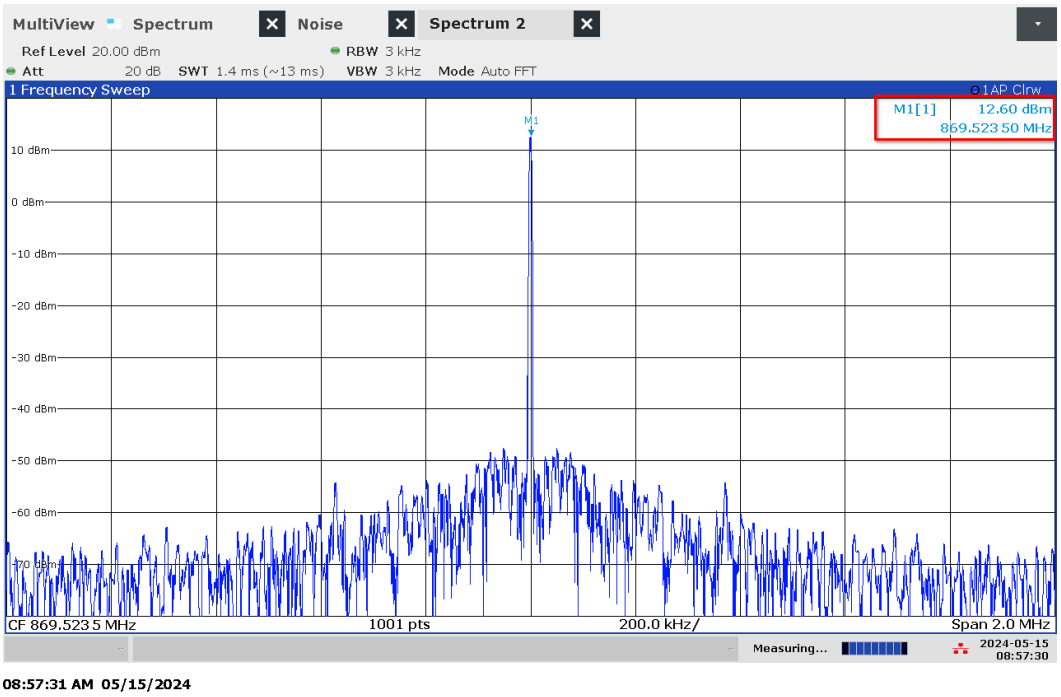


Figure 4.3: Measurement of the module RF output in "continuous wave" mode at 869.525 MHz with power of 15 dBm

4.1.4 Radio Performance

The output RF characteristics were validated using a spectrum analyzer connected directly to the RF output of the module through an U.FL pigtail.

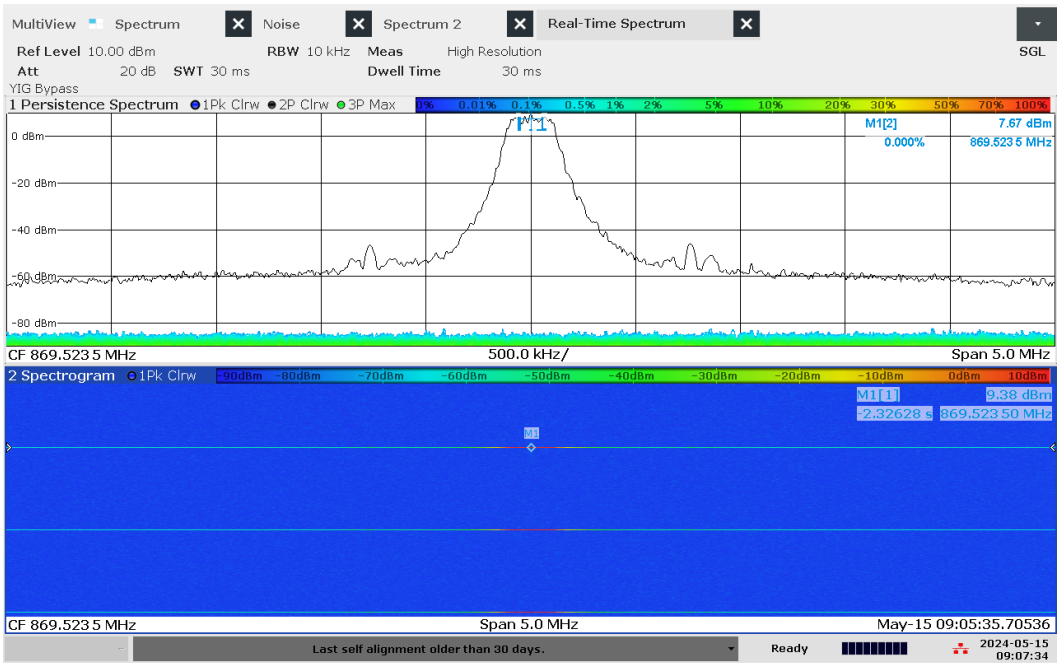
The module was set up in "continuous wave" mode [38] at 869.525 MHz with a power of 15 dBm, which produced the spectrogram visible in Figure 4.3. Here, we can observe that the peak power was measured at 12.60 dBm, and the frequency is off by 150 kHz.

Next two measurements in Figure 4.4, are captures of the module transmitting 8 bytes of data at SF5, bandwidth 250 kHz and coding rate 4:5, the rest of the parameters is identical as in Table 4.5.

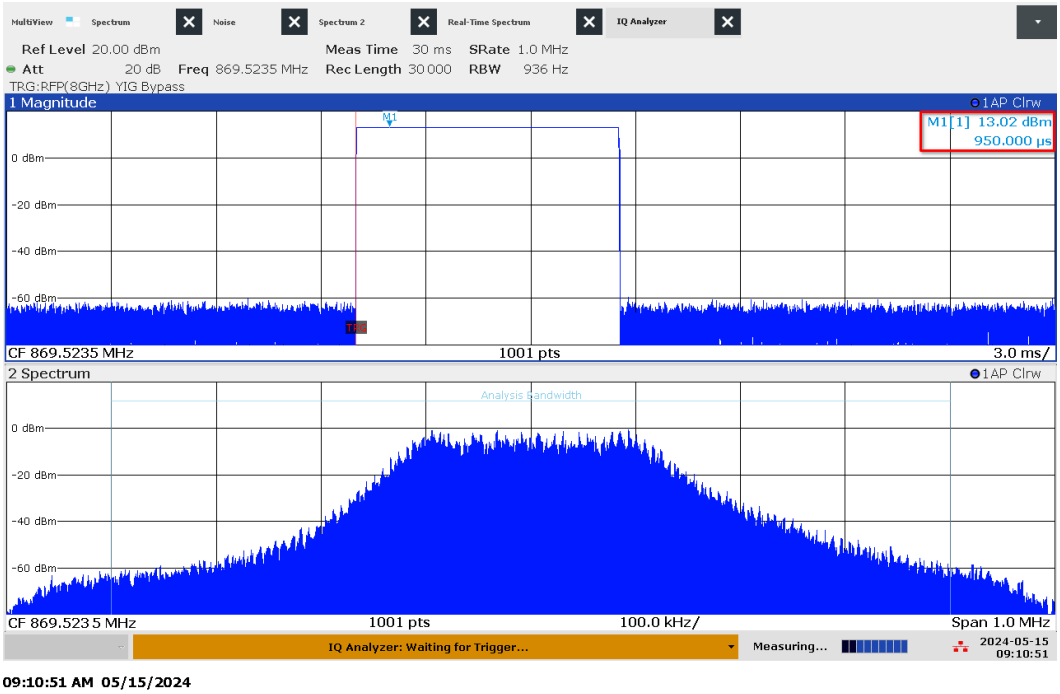
These measurements show that the output power stays stable at 13 dBm once modulating and that there are smooth transitions and power ramp-up, suggesting good power supply and power management design. The mask visible in the spectrum is uniform, without any unexpected artifacts.

4.1.5 Power Consumption

Because of the improvised replacement of the crystal oscillator with a TCXO, which exhibits much higher power consumption (2–3 mA) and an inability to connect its power supply to the pin PB0, which is designed to power this oscillator and gate its power supply whenever it

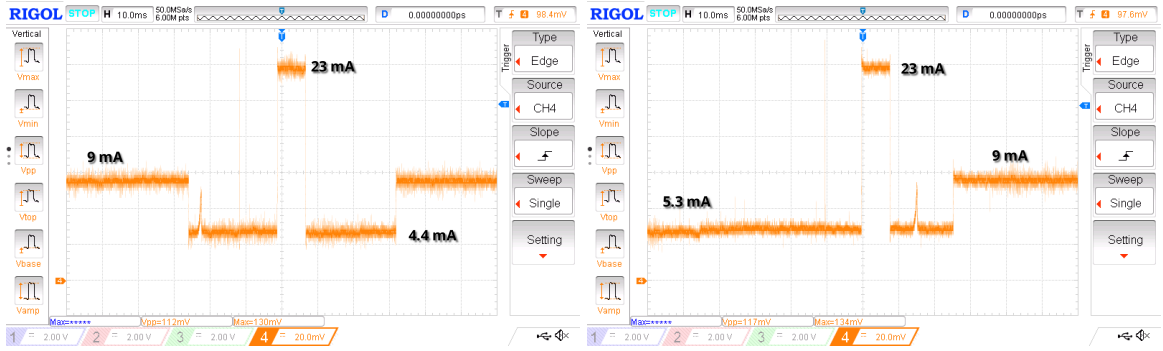


(a) : Spectogram of the transmitted signal (upper), waterfall graph of the spectrum (lower)



(b) : Packet in the time domain (upper), packet spectrum (lower)

Figure 4.4: Capture of the module transmitting 8 bytes of data at SF5, bandwidth 250 kHz and coding rate 4:5, the rest of the parameters is identical as in Table 4.5



(a) : Continuous receive with transmit in the middle **(b) :** Idle, transmit, receive acknowledge

Figure 4.5: Current waveform of the module measured through a 5.6 Ω resistor in different operational scenarios.

is not needed, the module exhibits higher than expected power draw of around 9 mA while receiving and 6 mA while idle. The fix is documented in Section B.1.

The idle power draw could be optimized to reach 10–100s of μA with the TCXO properly connected and significant time investment into the firmware development. The current consumption waveform can be examined in Figure 4.5.

4.2 Soil Moisture Sensor Design

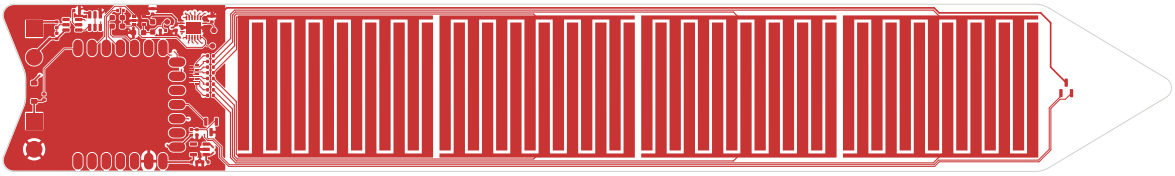


Figure 4.6: Printed Circuit Board design of the top layer of the soil moisture sensor board, where the 4 capacitive sensing zones are distinctly visible. These capacitors form what is later referred to as the "active area" of the sensor. More is available in Appendix C.

To summarize, the soil moisture sensor board contains

- the capacitance measuring circuit (as described in Section 3.4.2) with two ranges,
- a dual single-pole quadruple throw (Dual SP4T) mux chip for switching between the 4 sensing zones contained in the active area of the sensor,
- 8 channel TVS diode array to clamp the voltage on the capacitor electrodes and means of isolating the sensor elements from the measuring circuits,
- a 2.8 V linear low-dropout (LDO) regulator,

- footprint for the LoRa module,
- two analog temperature sensor ICs and
- single cell rechargeable lithium battery protection circuitry.

4.2.1 Capacitance Measuring Circuit

Of note is the performance of the capacitance measuring circuit, which was validated using an oscilloscope connected to the test-points on the sensor board, see Figure 4.8.

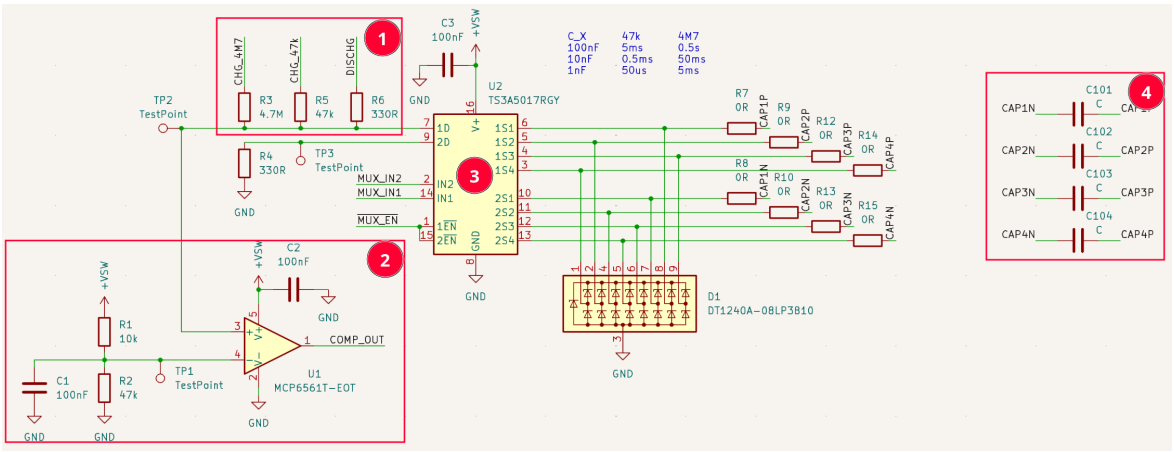


Figure 4.7: The capacitance measuring circuit functions by charging the unknown capacitors C_X (4) through a known resistor R_{CHG} (1) and measuring the time it takes to reach a certain voltage derived from a voltage reference (2), as proposed by Figure 3.8. The muxing (3) and protection circuitry (D1) are also included. Resistors R7–15 are there to facilitate a complete isolation of the sensor from the measuring circuit for measuring the capacitor elements externally, if the need be, without damaging the board. The full schematic is available in Appendix C.

The capacitance can be calculated from readings obtained in Figure 4.8. Given the charging voltage $U = 3.3$ V (it may not seem so from the Figure due to the charging resistor being disconnected as soon as the threshold voltage is reached). The capacitance RC time constant is defined as

$$\tau = RC \quad \rightarrow \quad C = \frac{\tau}{R} = \frac{7.6 \cdot 10^{-6}}{47 \cdot 10^{3}} = 161 \text{ pF}. \tag{4.1}$$

This result goes in line with the estimate given in Section 3.4.1. The reading would triple when submerged fully in water.

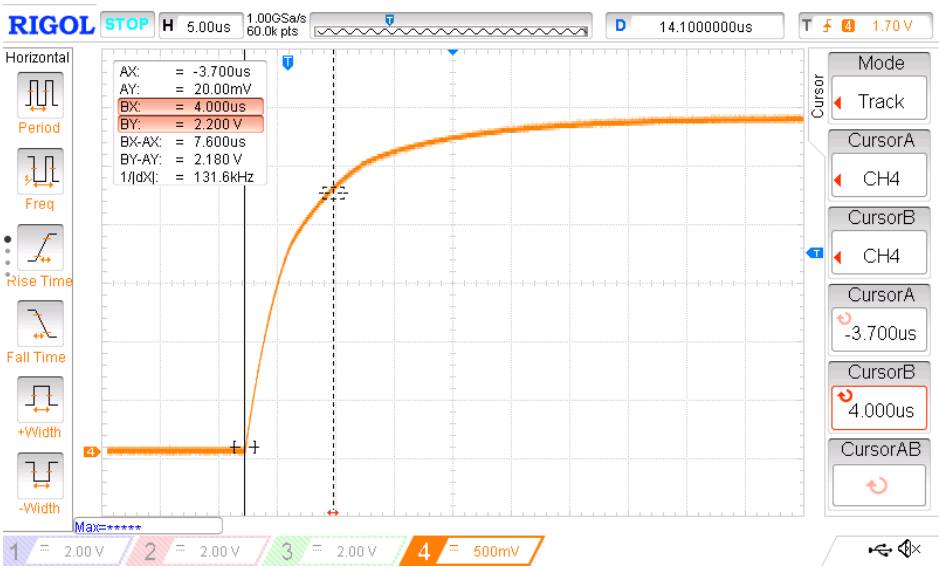


Figure 4.8: Sensor capacitor (in air) voltage rise time measurement when charged through a 47kΩ resistor, which was later replaced by a different value to increase the measuring time.

The original 47 kΩ and 4.7 MΩ resistors were replaced by 100 kΩ and 1 MΩ resistors to improve the measuring performance. These resistors were designed with this possibility in mind, and a larger footprint was used to make this process easier. the 4.7 MΩ resistor was not able to charge the circuit up to the detecting threshold of the comparator because the rest of the components incurred parasitic resistance to ground of about 15 MΩ, forming a voltage divider.

This setup provides repeatable and stable measurements in free air and when the capacitor is completely submerged in water. It is sensitive enough to reliably detect close proximity (1 cm) or touch of a human hand.

The following code is responsible for each sample conversion; refer to the comments for principle explanation.

```
pub async fn sample_current_channel(
        &mut self, range: SoilSensorRange
    ) -> SoilSensorResult {
    /* 1. Handle the range selection by setting the required charge
    pin as output and the unused pin as high-impedance input. */
    match range {
        SoilSensorRange::Low => {
            self.chg_1m.set_as_input(Pull::None);
            self.chg_100k.set_as_output(Speed::Low);
            self.chg_100k.set_low();
        },
        SoilSensorRange::High => {
            self.chg_100k.set_as_input(Pull::None);
            self.chg_1m.set_as_output(Speed::Low);
            self.chg_1m.set_low();
        },
    }
    /* 2. Bridge the connection between the selected capacitor
    and the rest of the measuring circuit (the channel selection
    happens before this function) and discharge the capacitor. */
    self.dischg.set_low();
    self.mux_nen.set_low();
    Timer::after_micros(100).await;
    /* Note: the DISCH pin is configured as open-drain output. */
    self.dischg.set_high();
    /* 3. Start charging the capacitor through the selected resistor. */
    match range {
        SoilSensorRange::Low => {
            self.chg_100k.set_high();
        },
        SoilSensorRange::High => {
            self.chg_1m.set_high();
        },
    }
    let start = Instant::now();
    /* 4. Await either the interrupt from the COMParator input
    or a timeout in case a wrong range was selected. */
    let ret = match select(
        self.comp.wait_for_high(),
        Timer::after_millis(2)
    ).await {
        Either::First(_) => {
            /* 5a. Calculate the measured time to be returned
            in case of successful conversion */
            let mut elapsed = start.elapsed();
            match range {
```

```
47                  SoilSensorRange::Low => {
48                      elapsed *= 10;
49                  },
50                  SoilSensorRange::High => {
51                      elapsed *= 1;
52                  },
53              }
54              SoilSensorResult::Ok(elapsed.as_micros() as u16)
55          },
56          /* 5b. Timeout */
57          Either::Second(_) => SoilSensorResult::Timeout,
58      };
59      /* 6. Return all pins to default state */
60      self.chg_100k.set_as_input(Pull::None);
61      self.chg_1m.set_as_input(Pull::None);
62      self.dischg.set_low();
63      self.mux_nen.set_high();
64      ret
65  }
```

4.3 Over–the–air Update Implementation

It is the responsibility of the update process to securely and reliably transfer the binary image of the firmware from the host computer through the Gateway to the designated Node (sensor) of the network.

This task is complicated by the limited resources available on embedded devices, as discussed in Section 3.2.1, and the slow and sometimes unreliable nature of the wireless connection in general. The following sections describe the protocol solution to this aspect of the OTA update process.

The performance testing was done along with the range test described in the following Section 4.4.

4.3.1 Data Fragmentation

The binary needs to be fragmented in order to be transmitted piece by piece. To guarantee the assembly of these fragments on the Node, an acknowledge mechanism must be present along with a way to detect errors in the transferred data. While the LoRa physical layer provides forward error correction and also up to a 16-bit CRC checksum, the error rate is, from experience, still too high to be practical. To tackle this, a custom 32-bit CRC was used to protect each fragment together with a SHA256 hash of the whole assembled binary, which is calculated as the last step in the update process.

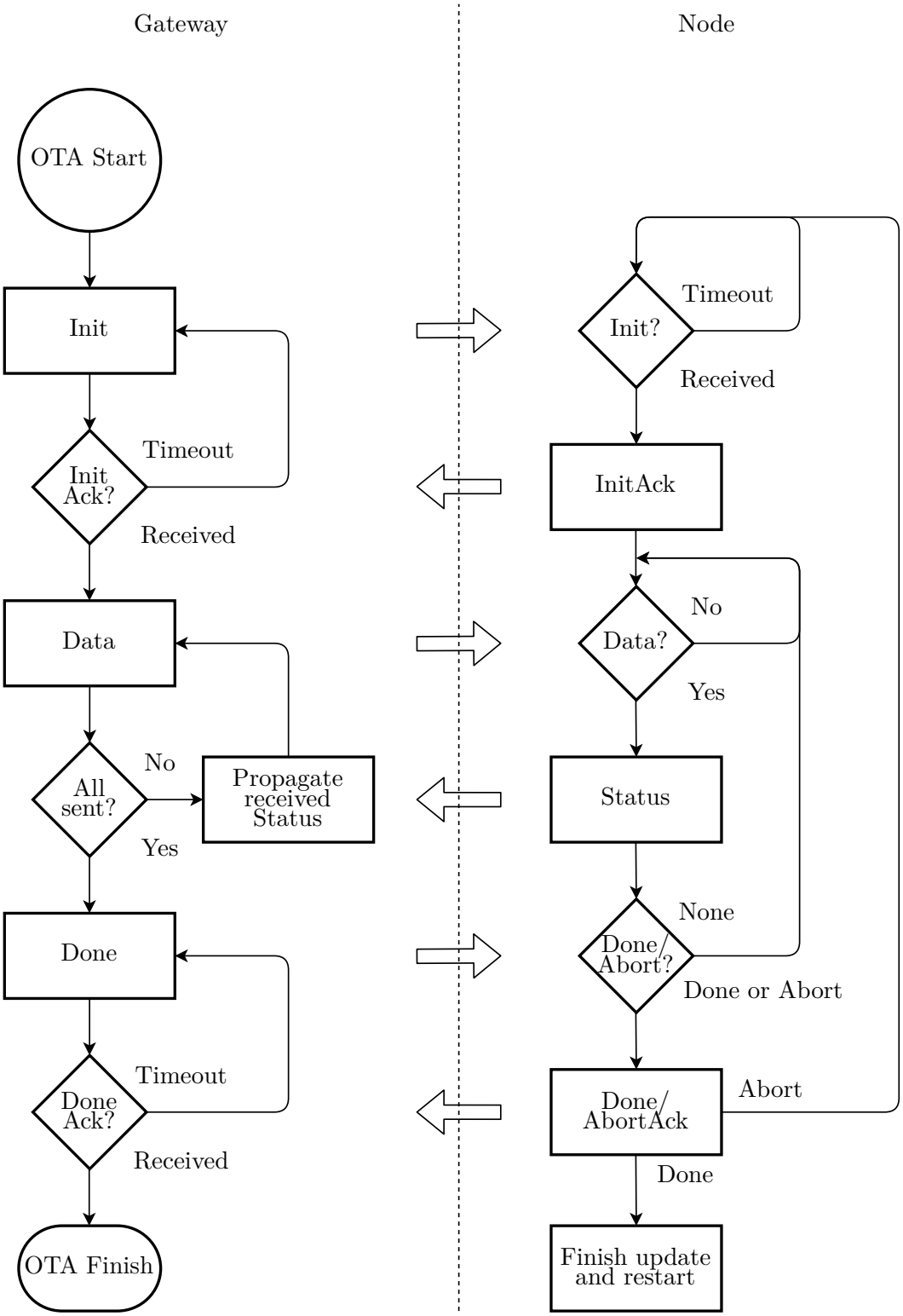


Figure 4.9: Simplified flowchart depicting the over–the–air update communication process between the Gateway and the Node.

Figure 4.9 provides a basic overview of the OTA update process. The communication with the host is left out for brevity but mostly mirrors the operations that happen on the Gateway. Names in the cells correspond to packet types defined in the module-runtime/src/ota/common.rs[2] file as `enum OtaPacket`. Packets are serialized and deserialized using the `postcard`[3] Rust package. LoRa driver is provided by the `lora-rs`[4] package.

4.3.2 Acknowledge Mechanism

The Automatic Repeat Request (ARQ) mechanism is inspired by Selective Repeat ARQ. The receiver is able to accept frames that are out of order and selectively requests missing or corrupted blocks. In contrast to standard ARQ, here the acknowledge (ACK) packet contains up to 32 indexes of blocks that were successfully received instead of just one.

Likewise, the transmitter does not expect to receive ACK packets consistently and continues to transmit in order for up to 16 unACKed frames. The receiver is thus able to relax the rate of ACK packets to reduce the air time usage. Since the transmitter is not attempting to use the maximum available channel capacity, it transmits about one packet per second; it does not need the immediate response to measure latency.

4.3.3 Bootloader

Once the data was transferred to the target Node to be updated, the Node needs to perform the update by swapping the old firmware image with the new one, as described in Section 3.2.1. The Embassy Bootloader was used for this purpose, as it is easy to integrate and features a small memory footprint of just 8 KBytes, along with a rollback feature [15]. The rollback is initiated after the device reset if the application fails to signal to the bootloader that it starts up successfully by writing a special value in the state section of the FLASH.

4.4 LoRa Module Range and Link Speed Test

In order to determine the real-world performance, a range test was conducted. Primary points of interest are

- maximum practically usable range of the solution and connection quality deterioration as the distance and occlusion increase,
- performance of the designed module compared with the Nucleo development board as reference,
- the effect of proximity of the antenna to the ground and
- the effect of the spreading factor (SF) modulation parameter.

[2] https://github.com/manakjiri/lora-module-fw/blob/thesis/module-runtime/src/ota/common.rs
[3] https://docs.rs/postcard/latest/postcard/
[4] https://github.com/lora-rs/lora-rs

All testing so far was done in a home environment with Nodes in close proximity, and no major packet loss could be observed. Given the relatively low experience in RF design, however, it is expected the module will perform worse than the Nucleo board.

With an increasing spreading factor, the symbol time increases, and the link budget goes with it [38], as can be seen in Table 4.10. But longer symbol time relies on more stable and precise frequency reference.

Spreading Factor (SF)	5	6	7	8	9	10	11	12
2^SF (Chips / Symbol)	32	64	128	256	512	1024	2048	4096
Typical LoRa® Demodulator SNR [dB]	-2.5	-5	-7.5	-10	-12.5	-15	-17.5	-20

Figure 4.10: Range of Spreading Factors (SF).

Bandwidth variation exhibits similar traits, but to stay within the legal limits of the EU868 band, it is necessary to stay within 125–500 kHz [17, 49]. This is the reason for selecting SF as the variable while keeping the Bandwidth and the coding rate (CR) constant. CR also has a very predictable effect on the connection quality.

4.4.1 Hypothesis

It is expected the proximity to the ground will have a high impact on the usable range of the sensor, meaning Nodes mounted with their antenna closer to the ground should perform worse. Higher SF should yield a longer range. We are not expecting to surpass the distance of 1 km.

4.4.2 Prerequisite

The maximum achievable SF needed to be determined. This was done experimentally, starting with SF5 and incrementing util it was no longer possible to transfer data. This test was conducted using Nucleo acting as the initiator of connection with the LoRa module. Devices were situated on a desk about 1.5 meters apart, with their antennae positioned orthogonally in respect to each other to avoid overwhelming the receiver. All other parameters are identical to those used in the latter experiment.

The connection was stable at SF11, but at SF12 the LoRa module stopped responding, thus the experiment was done at SF5 and SF11, which corresponds to a theoretical delta of 15 dB [38].

4.4.3 Methodology

All LoRa modulation and packet parameters were set in firmware (commit 76acdcd[5]) identically for all devices used, a summary is provided in Table 4.5.

Frequency	869.525 MHz
RF power output	15 dBm
Bandwidth	250 kHz
Coding rate	4 : 8 (2x overhead)
Bandwidth	250 kHz
Preamble length	32 b
Implicit header	No
LoRa CRC	No
Inverted IQ	No
Transmit boost	No
Receive boost	No

Table 4.5: LoRa parameters for range testing.

LoRa modules used the MOLEX 105262–0003 antenna and Nucleo boards used their stock antenna, all were close to vertical orientation. A pole with two LoRa modules and one Nucleo (see Figure 4.11) was constructed to act as the stationary set of Nodes to test against. Another Nucleo board was attached to a car (see Figure 4.12) to act as the Gateway; see Table 4.6 for more details.

Device	**Height**[a] **[m]**	**Address**[b]
Nucleo (Gateway)	1.65	1
LoRa module (Node)	0.19	2
Nucleo (Node)	0.55	4
LoRa module (Node)	0.82	3

Table 4.6: Devices used for range testing.

[a] Measured distance from the root of the antenna to ground level
[b] Data presented will use these addresses to distinguish between the different Nodes used in the experiment

A location was picked to represent the final use–case, a field with sections of direct line–of–sight (LOS) and sections obstructed by hills. This place is situated near Průhonice municipality in the Czech Republic. The pole with Nodes attached was installed at 50.0012006N, 14.5308961E[6] (WGS84).

A set of measurements was taken through the Gateway at different locations, progressing further from the stationary pole with the Nodes attached. These measurements consisted

[5] https://github.com/manakjiri/lora-module-fw/tree/76acdcd7b31f259c88f1808ed79886dc26295b4e
[6] https://en.mapy.cz/letecka?q=50.0012006N%2C%2014.5308962E&x=14.5381677&y=50.0011824&z=16

Figure 4.11: Close-up of the pole with Nodes attached for range testing.

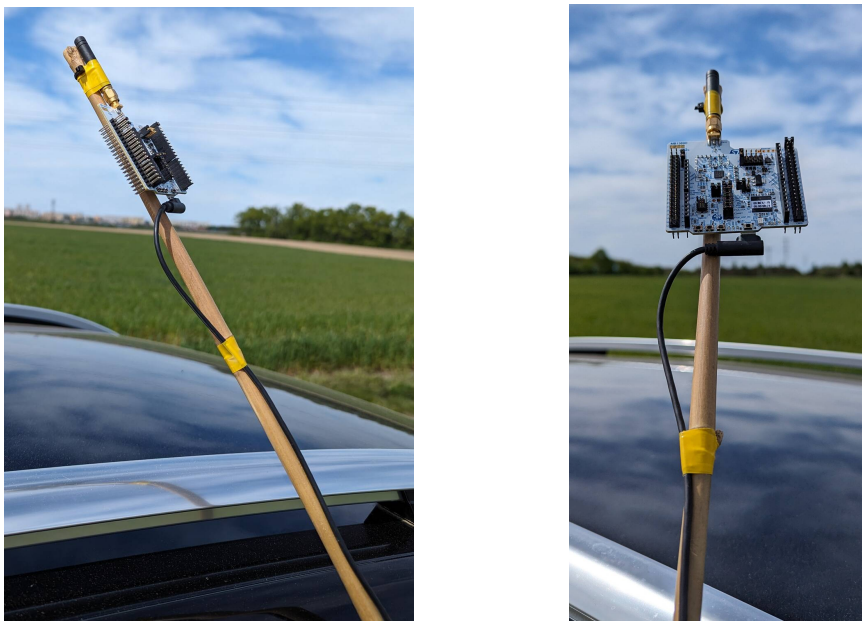


Figure 4.12: Close-up of Nucleo attached to the car acting as the Gateway for range testing.

of downloading a random binary 1 KB file to each of the Nodes consecutively. This file was fragmented on the fly by the OTA update protocol to 16 blocks of 64 bytes each.

This means that effectively 16 different measurements of the signal quality were taken at each location for each of the three Nodes. Measurements were recorded by the system in a CSV file containing the time since the download started, the number of packets sent, and the last block that was acknowledged - thus, a success rate can be calculated. In case the download fails to initiate, nothing gets recorded, and that is interpreted as 0% success rate.

4.4.4 Results

As this is an experiment where direct line–of–sight cannot be relied upon to interpret the results, it was deemed necessary to somehow capture the environment and the terrain and present it along with the obtained data.

Altitude data was obtained from the Seznam.cz elevation API[7] in 5 m resolution of interpolated coordinates of lines connecting the Nodes and Gateway locations. This location data is also available in raw form as part of the Appendix D.

Elevation Deviation from the Straight Line

The following graphs describe the terrain in terms of Elevation Deviation from the Straight Line (EDSL) connecting the Node's location (spot 0) with the furthest point at which the last measurement was taken. Units of EDSL proportionally translate to meters since the effect is similar to subtracting the offset signal in data.

The same data without the EDSL interpretation is included in Appendix D for completeness. It also demonstrates the need for such data manipulation because the features of the terrain are a lot less apparent in the original data.

Effect of the Spreading Factor

Lower SF decreases the packet air time at the expanse of the lower range reached. SF5 averaged 0.125 s, while SF11 averaged 1.125 s round-trip time (excluding packet-loss) with these payloads.

Using SF5, it was possible to maintain a stable connection with Nodes (3 and 4) positioned higher above the ground level even when occluded by the terrain. Connection with Node 2 was lost as soon as any occlusion was introduced. It can be expected that this setup would be feasible in very flat environments but could be susceptible to dropouts with slight changes, such as growing crops.

The SF11 configuration exceeded expectations in terms of distance reached, and a stable connection was maintained with all Nodes without difference in packet loss, which is perhaps

[7] https://api.mapy.cz/v1/elevation

most surprising since some degradation was expected, at least for the Node 2 position closest to the ground level.

While the range was higher than with SF5, the connection quality is somewhat worse in the stable regions (less than 500 m distance, direct line-of-sight) for SF11. This suggests that a stability limit was indeed reached, as outlined in the Prerequisite Section 4.4.2.

Maximum Distance

Finding the limit at FS11 proved to be more difficult than anticipated. The surrounding environment either contained too favorable or absolutely unfavorable conditions at further distances. Two additional data points displayed in Figures 4.17 and 4.18 were gathered at locations outside of the planned path.

The first Figure 4.17 shows the only flawless connection with the Nucleo Node 4, which must have been facilitated thanks to a signal reflection from some other feature in the environment. While the connection with Node 3 was initiated and it acknowledged 2 packets, the connection later timed out after the Node failed to respond in 30 more attempts.

The second location, captured by Figure 4.18, exhibited flawless transmission of the payload for all 3 Nodes at a distance of 1.2 km with occlusion.

The nature of LoRa is that it performs well within its large link budget, but once that is exhausted, the performance drops off rapidly. The attenuation of the medium through which the electromagnetic wave travels changes with temperature and saturation with humidity and particulate, the position of the antennae may change slightly due to wind and the onboard oscillators can drift with temperature and power supply.

Most of these factors remained constant throughout this experiment since it was done within a 1.5-hour window, but they must be considered for long-term deployments.

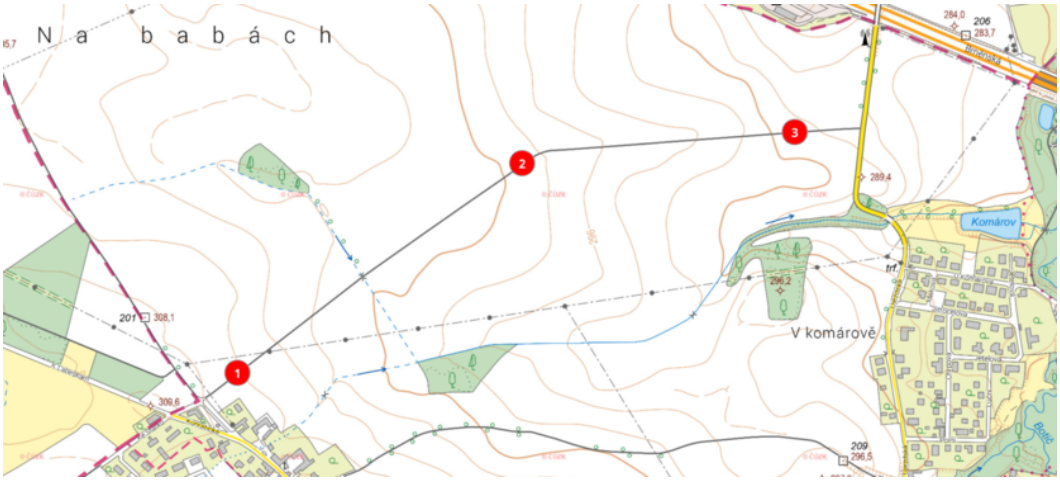
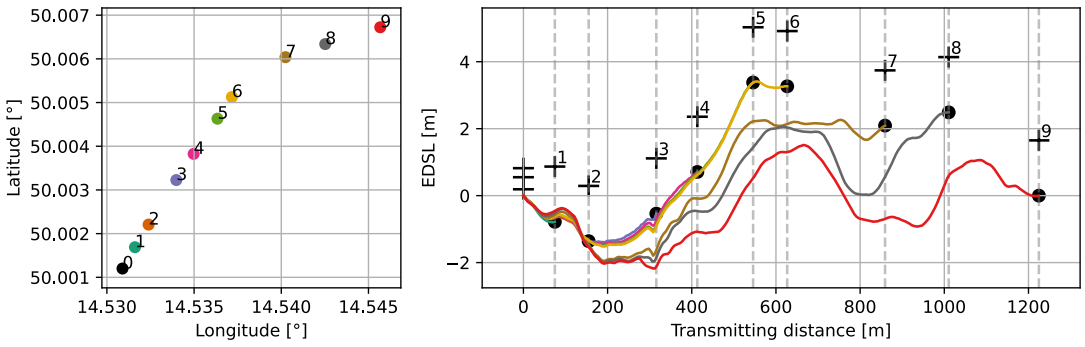
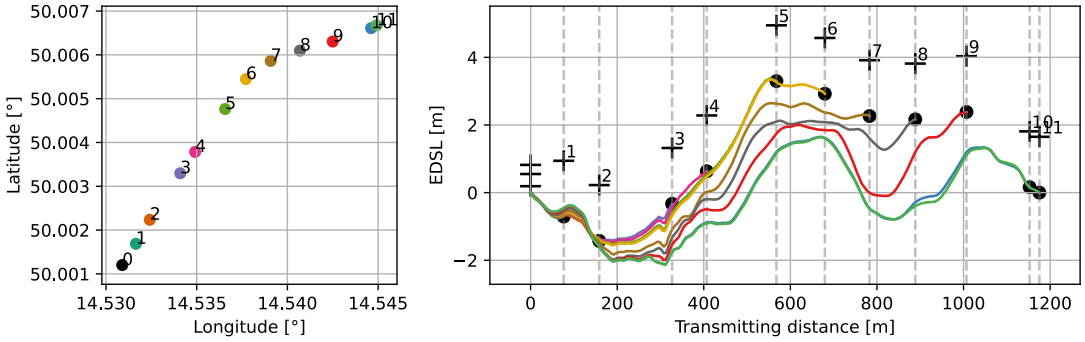


Figure 4.13: Map of the range test site. Correlation with Figures 4.14 and Table 4.16 follows: point 1 corresponds to the Nodes location (spot 0), point 2 is the spot 5 in both SF5 and SF11 measurements, point 3 is the spot 9 and spot 11 for SF11 and SF5 respectively. Source: Geoprohlížeč ČÚZK.

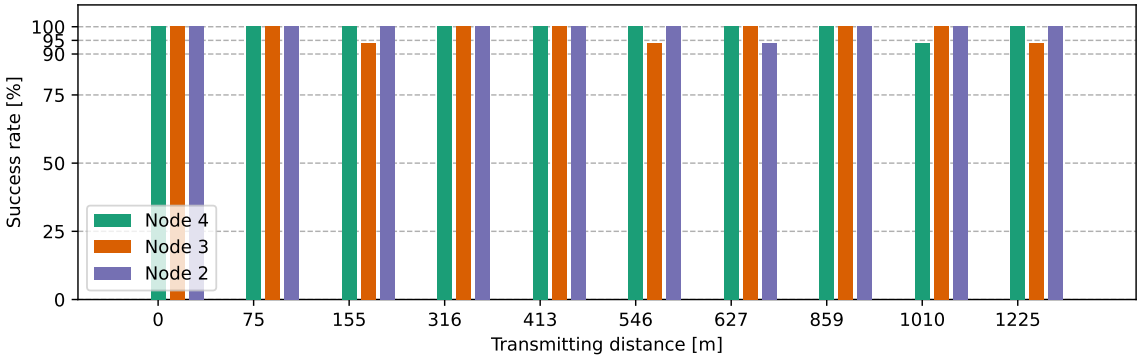


(a) : SF11

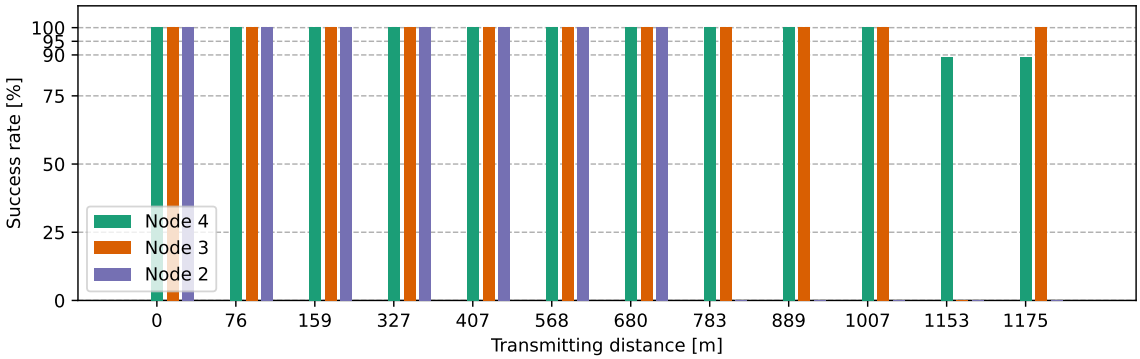


(b) : SF5

Figure 4.14: Graph of the range test locations, their distance from the pole with Nodes, and the relief in the path of the signal described as Elevation Deviation from the Straight Line 4.4.4, which compensates the downward slope of the terrain and brings attention to the features of the terrain. Black crosses signify the position of the antennae of each device for each spot number.



(a) : SF11



(b) : SF5

Figure 4.15: Graph of the packet transfer success rate for each Node with respect to the transmission distance.

SF11

Spot	Node 4	Node 3	Node 2
0	100%	100%	100%
1	100%	100%	100%
2	100%	94%	100%
3	100%	100%	100%
4	100%	100%	100%
5	100%	94%	100%
6	100%	100%	94%
7	100%	100%	100%
8	94%	100%	100%
9	100%	94%	100%

SF5

Spot	Node 4	Node 3	Node 2
0	100%	100%	100%
1	100%	100%	100%
2	100%	100%	100%
3	100%	100%	100%
4	100%	100%	100%
5	100%	100%	100%
6	100%	100%	100%
7	100%	100%	0%
8	100%	100%	0%
9	100%	100%	0%
10	89%	0%	0%
11	89%	100%	0%

Figure 4.16: Success rates for each node and location at spreading factors 5 and 11.

SF11-FAR1

Spot	Node 4	Node 3	Node 2
0	100%	100%	100%
10	100%	6%	0%

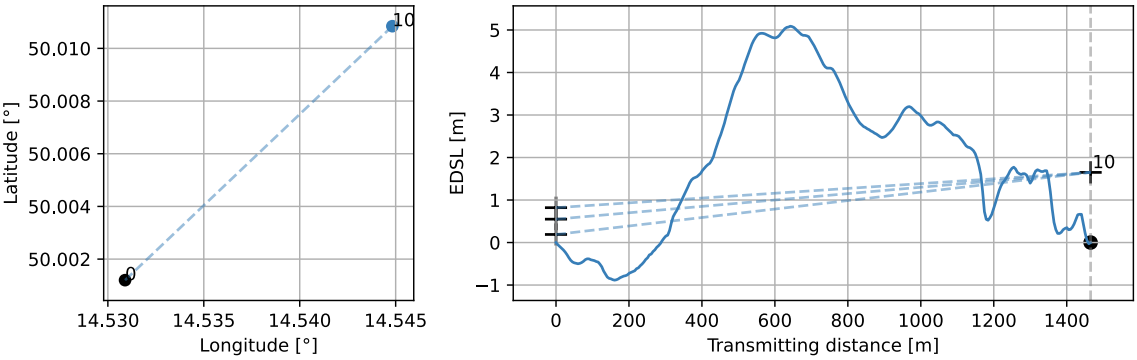


Figure 4.17: Table of the success rates and graph of the first long-range test location, distance from the pole with Nodes, and the relief in the path of the signal described as Elevation Deviation from the Straight Line 4.4.4, which compensates the downward slope of the terrain and brings attention to the features of the terrain. Black crosses signify the position of the antennae of each device for each spot number. Dashed lines highlight the direct line-of-sight for the antennae.

SF11-FAR2

Spot	Node 4	Node 3	Node 2
0	100%	100%	100%
11	100%	100%	100%

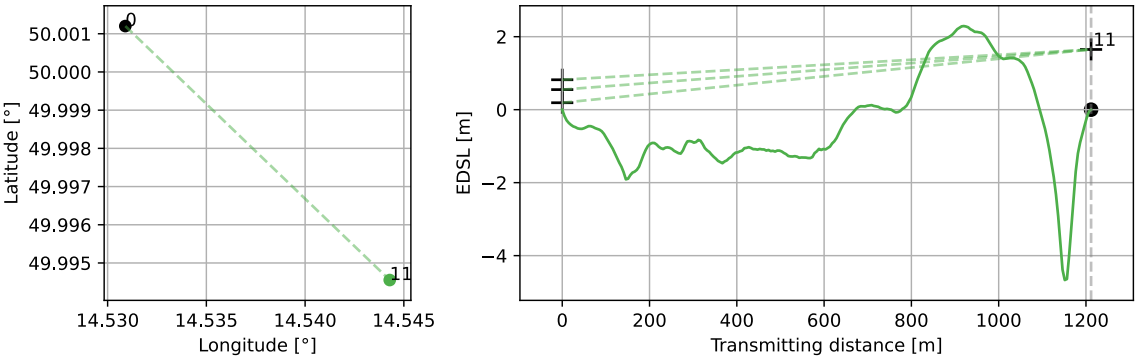
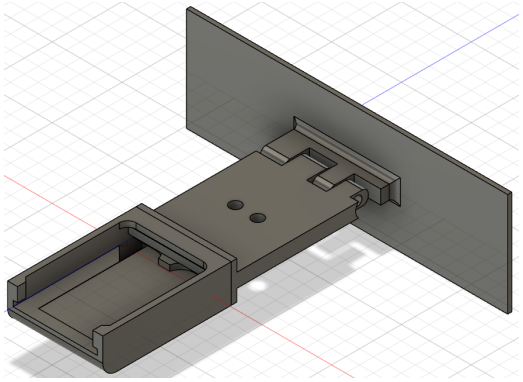
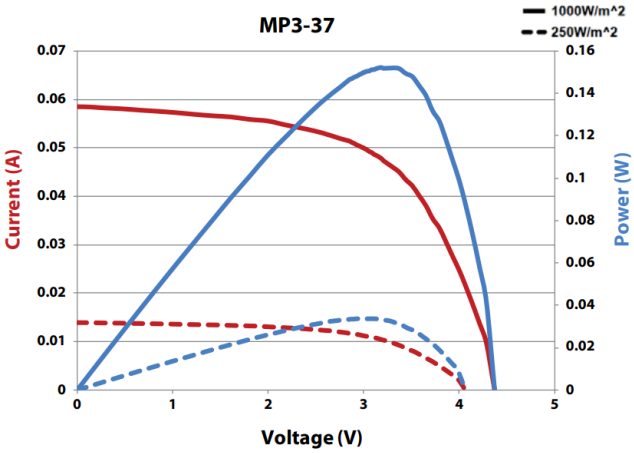


Figure 4.18: Table of the success rates and graph of the second long-range test location, distance from the pole with Nodes, and the relief in the path of the signal described as Elevation Deviation from the Straight Line 4.4.4, which compensates the downward slope of the terrain and brings attention to the features of the terrain. Black crosses signify the position of the antennae of each device for each spot number. Dashed lines highlight the direct line-of-sight for the antennae.



(a) : CAD design render of the soil moisture sensor case (bottom left) including the hinge providing the swivel adjustment of the panel (center) and the panel plate part.



(b) : IV Curve of the MP3-37 solar panel manufactured by PowerFilm, which was used for the soil moisture sensor validation.

Figure 4.19: Soil moisture sensor case (left) and the IV curve of the solar panel (right). Both parts were used during the soil moisture sensor validation.

4.5 Soil Moisture Sensor Validation

A practical experiment was conducted to validate the functioning of the soil moisture sensor itself.

4.5.1 Hypothesis

The sensor should detect soil moisture based on the experiments done during sensor validation. It is also expected that the sensor will be able to charge its battery when enough sunlight is available.

4.5.2 Prerequisite

The sensor is meant to be made self–sufficient thanks to a solar panel. The main constraint is the power generated and the panel's dimensions. Since the solar panel is charging the battery directly, it needs to generate sufficiently high voltage to do so. The MP3–37 from PowerFilm was picked because its open circuit voltage is 4.6 V, and according to 4.19b, it should produce 50 mA at 4.1 V, which is ideal for this application.

A sensor housing needed to be constructed to support the solar panel in the correct orientation. The housing was designed in Fusion 360 CAD software and 3D printed, as visible in Figure 4.20. The housing has a support arm for the solar panel, which allows it to swivel, which is better viewed in Figure 4.19a.

(a) : Soil moisture sensor stuck in the pot (center) on a window sill facing south. The lithium battery is connected externally through the RaspberryPi Pico development board (bottom), acting as a data logger of the voltage and current.

(b) : Close–up picture of the sensor and its housing. The LoRa module is clearly visible, along with the connections to the solar panel, the battery, and the antenna.

Figure 4.20: Soil moisture sensor proof–of–concept deployment scenario used for validating the solar panel performance and the sensor's measuring capabilities.

Lastly, the sensor's scale needed to be calibrated. This was done by taking 10 measurements in free air (Dry) and fully submerged in water (Wet). The resulting coefficients are included in Table 4.7. These coefficients are used for calculating the soil moisture saturation using the following equation

$$\theta' = \frac{T - T_{Dry}}{T_{Wet} - T_{Dry}} \tag{4.2}$$

4.5.3 Methodology

The sensor was stuck in the existing planter of a citrus tree plant. Sensor readings are gathered using the LoRa interface every 15 seconds. Power logger readings using Raspberry Pi Pico are also gathered every 15 seconds.

The power logger is measuring the battery voltage and current. Thus, any power consumed by the sensor itself during charging is not visible but is assumed to be constant throughout the experiment, as the reading interval is fixed.

Zone	Depth [mm]	Dry value [μs]	Wet value [μs]
1	20	276	728
2	50	277	728
3	90	277	581
4	120	277	498

Table 4.7: Soil moisture sensor zone calibration coefficients.

The planter with the sensor is placed behind a glass window facing south. The panel is inclined by the swivel mechanism at about 30 degrees.

4.5.4 Results

The sensor was started on an external power supply the day prior to the test to gather the initial measurements visible in Figure 4.21 up to 12.5 00:00. After enough readings were gathered, the sensor was connected to its 330 mAh Lithium Polymer battery, which has been discharged completely prior to the experiment. The soil was mostly dry, as can be seen in the referenced Figure.

Watering occurred at 12.05 10:10, at which point the relative moisture saturation started rising into the 60–80 percent range in all but the control zone. The soil moisture was than gradually dropping throughout the following 7 days. Notably, the surface zone (zone 2, brown in Figure 4.21) returned to the level of soil moisture saturation before the watering, whereas the root-level zones were drying slower, signaling that there is still enough moisture.

Figure 4.22 starts at the moment of power–on of the sensor in the morning the following day. The power consumption of the sensor was too high and the charging efficiency too poor for the sensor to maintain stable operation, as it discharged after midnight the following day at 2.5 V, at which point the sensor's battery protection circuitry disconnected it. The same event is repeated at the same voltage all following days and is thus left out. It was overcast with occasional full sun the day of 12^{th} of May; following two days offered full sun throughout.

At this point, a problem was noticed by observing the data, where it was apparent that the sensor stopped charging even though there was full, unobstructed sun hitting the solar panel. This was rectified in the morning the following day (14^{th} of May) by bypassing the charging diode D2 (see Figure C.3). This caused a temporary disruption in the logged results, which manifested by the peak visible at roughly noon that day.

After this modification, the battery managed to reach higher voltage and the sensor stayed working longer than any previous day. The rest of the data is cyclical in the same way, thus not shown. This poor battery performance is caused by a combination of many factors:

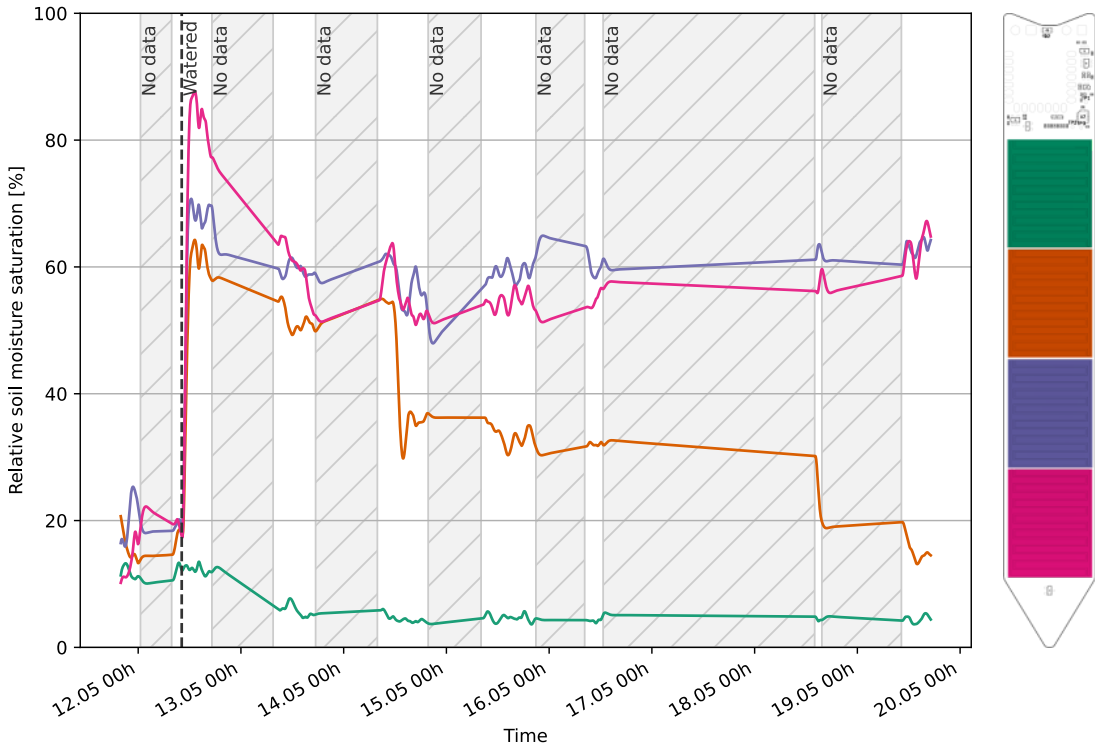


Figure 4.21: Filtered readings from the individual sensor zones. Zone 1 (green) is in free air acting as control (as can be seen in Figure 4.20), zones 2 and 3 below it (brown and purple) are completely submerged in the soil. Zone 4 (magenta) is in the boundary layer of soil and Ceramsite at the bottom of the pot. The sample was watered 7 days before the marked point of watering (sharp increase), after which a decline in the measured capacitance can be observed, corresponding to the sample drying out. The sensor run out of power during the night, these areas are highlighted in grey.

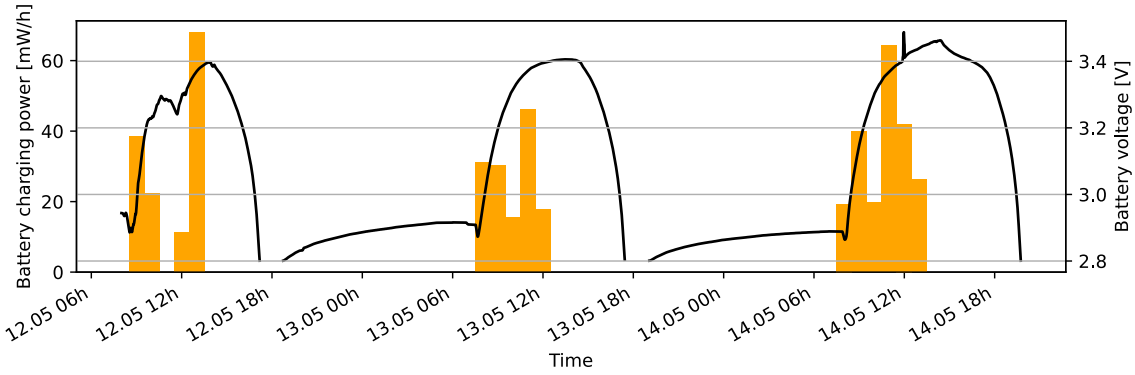


Figure 4.22: Battery voltage (black) and charging power (yellow), excluding the power consumed by the sensor and the module itself. The power is provided solely by the 150 mWp solar panel. The experiment was started with the battery completely empty. Voltages below 2.8 V are left out; the under–voltage protection of the sensor is always triggered at 2.5 V

- Due to the hardware design mistake involving the crystal oscillator on the LoRa module 4.1, the TCXO cannot be switched off, which causes a constant current draw of 2.5 mA. The MCU has the capability to start the oscillator only when needed, which is implemented in the following revision of the LoRa module.
- The measuring interval is rapid to bring more data into understanding the sensor performance, which is not necessary for final deployment, where a measuring interval of 10 minutes or longer would suffice.
- The MCU is not able to run at a low enough clock speed because of limitations in the firmware
- The solar panel is behind a double-glazed window with a low solar zenith angle (summer solstice), causing some energy loss in the glass. Because of this, the solar panel may not be receiving enough radiation to reach peak performance, which would be expected given the clear sky.
- The battery is directly charged by the solar panel without any energy harvesting optimizations as Maximum Power Point Tracking implemented. However, this was a conscious decision during the sensor design to make it simple at the cost of some efficiency.
- Lastly, the solar panel is obstructed by building and window features, which limit the irradiation time.

Many of these factors are solved by the next iteration, see B.1, and with improvements in the firmware of the module. It is even possible to update this firmware remotely, thanks to the OTA update capability.

4.6 Proof–of–concept application

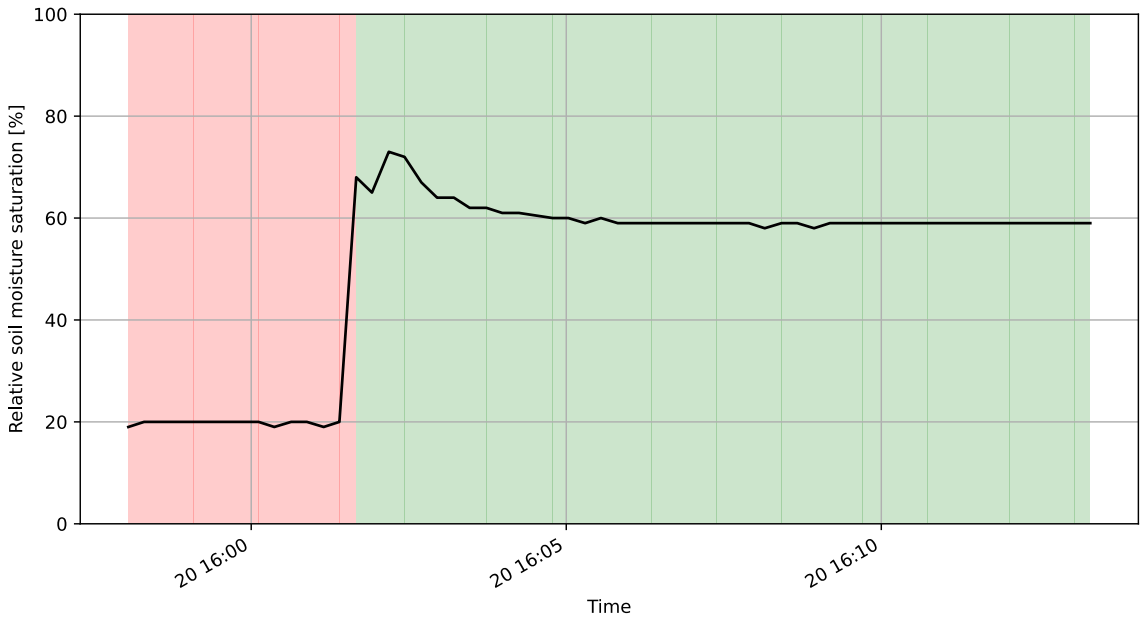


Figure 4.23: Output from the demo application showing the measured soil moisture saturation (black) and the watering request depicted as red when watering is requested or green once the request has been fulfilled.

For demonstration purposes, a simple proof-of-concept backend application was created and tested. This application uses multiple different inputs, the main input being the sensor-measured soil moisture saturation. Other inputs include the weather forecast for the next 6 hours and the current time of day, as plants tend to prefer being watered during the day. For the weather forecast, if there was a probability of precipitation in the next 6 hours, which was higher than 25 %, the watering was postponed. Considering the probability is enough, as the main focus is not to over–water the plants, if the downpour was not sufficient, the application would request watering the next day.

The whole decision process is documented in the code below. This function is responsible for computing the watering request. The configuration is stored in a json file. The weather information is fetched from OpenWeather API[8] and processed before this function. The decision process could be further ameliorated, as hinted by the comments in the code.

[8] https://openweathermap.org/api/one-call-api

```
fn figure_out_watering(
        config: &Config, moisture: [u16; 4], pop: f64
    ) -> WateringResult {
    /* 1. Apply the calibration constants to the raw readings
    from each zone of the soil moisture sensor */
    let moisture = moisture
        .iter()
        .zip(config.sensor_cal_low.iter()
        .zip(config.sensor_cal_high.iter()))
        .map(|(m, (low, high))|
            ((*m).clamp(*low, *high) - *low) / (*high - *low))
        .collect::<Vec<f64>>();
    /* 2. Average all zones
    Note: this could be the main point of improvement,
    but to implement this correctly, the program would
    need to know a lot more context about the plant,
    the soil type and the needed schedule */
    let moisture = moisture.iter()
        .fold(0.0, |acc, m| acc + m) / moisture.len() as f64;
    /* 3. Apply the rules regarding day length, moisture
    and probability od precipitation */
    let hours = Local::now().hour();
    WateringResult {
        watering: moisture < (config.moisture_threshold / 100.0)
            && pop < (config.precipitation_threshold / 100.0)
            && hours >= config.day_start_hour
            && hours < config.day_end_hour,
        moisture,
    }
}
```

JSON configuration file example for the backend application. The application itself outputs its findings in a CSV file, which could be read by other systems providing the watering automation. The CSV file also provides a history, which could be provided to the user.

```
{
    "latitude": 50.0755,
    "longitude": 14.4378,
    "sensor_cal_low": [276, 277, 277, 277],
    "sensor_cal_high": [728, 728, 581, 498],
    "moisture_threshold": 20,
    "precipitation_threshold": 50,
    "day_start_hour": 6,
    "day_end_hour": 18
}
```

Chapter 5

Conclusions

In the first part of this thesis, a generic microcontroller–based module with LoRa communication, Over–the–air update capability, sensor interfaces, and solar powering was developed. In the second part, a wireless sensor node for use in soil moisture sensing applications, useful for managing water resources in agriculture, horticulture, and home environments, was realized.

The communications module is based on the STM32WLE5JC System on a Chip supporting the LoRa wireless interface operating in the EU868 band. The module was designed, manufactured, and validated. Its biggest differentiator against other similar hardware is the integrated 1 Mbyte non–volatile memory, which can be used for data logging and configuration, but its main purpose is to facilitate safe and efficient Over–the–air updates of the firmware running on the module. This module is versatile enough to be useful in other applications outside of this work with its small footprint of 20.32×22.48 mm, low power consumption (9 mA receiving) and 16 Input/Output pins supporting interfaces such as UART, I2C and featuring 5 ADC channels.

For the firmware and host–side software implementation the Rust programming language was used. STM32 support and the async–await executor were provided by the Embassy project, together with the Bootloader, while the `lora-rs` library was used to integrate LoRa. The firmware was split into the

- generic `module-runtime` library code, which includes hardware initialization, the OTA implementation and other utilities,
- the `module-gateway` and `module-node` applications, which implement the Gateway (communication with the host computer) and the Node (moisture sensing application), respectively.

The soil moisture sensor is a piece of hardware containing the LoRa module, 4 capacitive sensing zones, 2 temperature sensors, measuring and charging circuitry, and is designed to be stuck into the soil to measure its moisture content. The active sensor area can measure the volumetric moisture level at 20, 50, 90, and 120 millimeters below the surface; the temperature is measured near the surface and at 150 millimeters below the surface. The sensor is able to charge its 330 mAh lithium cell from an integrated 150 mWp solar panel and stay operational

throughout the day currently, however with more work it could function for weeks on a single charge, without much sunlight.

Range test was conducted in a typical deployment scenario, where the LoRa module proved to maintain a stable connection at a distance of over 1 kilometer, with its antenna positioned only 190 millimeters above ground level. This was achieved at 15 dBm transmit power with the spreading factor set to 11, yielding an average data rate of 300 bits per second (including protocol overhead and dead time). It is even possible to far exceed this range in more favorable conditions, or increase the transfer speed significantly (up to 9 times) at the expanse of some range.

Appendix A

Using the Rust Programming Language for Embedded Applications

Rust is increasingly recognized in the field of embedded systems for its promise of memory safety and concurrent programming without the overhead of a runtime or garbage collector. Originally designed for systems programming, Rust offers deterministic performance and fine-grained control over hardware, akin to C, but with a stronger emphasis on safety and modern programming features.

As of recent years, the Rust language has seen growing adoption in embedded development, supported by a robust toolchain and a vibrant ecosystem. Rust's compiler, Cargo package manager, and integration with LLVM provide a seamless development experience, from writing high-level application logic to low-level hardware interfacing. The availability of crates, Rust's libraries, for various hardware abstractions and middleware, further contributes to Rust's suitability for embedded applications.

The traditional language of choice in this field has been C, thus it is customary to compare these two languages. Rust is a much younger language and as a result, it provides native support for many of the now ubiquitous features, such as error handling, package management, and generics, that programmers expect from any language, but the fundamentals are the most important aspects for embedded

- Rust's ownership model ensures memory safety at compile time, virtually eliminating common bugs such as buffer overflows and null pointer dereferences.
- Rust's approach to concurrency, based on the ownership and borrowing principles, allows developers to write inherently safe concurrent code without the typical risks of data races. While primarily developed for multi-threaded applications, this checking is also applicable to interrupts.

Additionally, it provides tools for code and library management, which traditionally needed to provided separately. The use of Cargo for dependency management and builds, along with Rust's built-in testing and documentation tools, modernizes the embedded development process.

However, being a new language is the source of its main objective drawbacks as well, including

- Rust's strict compiler and its concepts of ownership and borrowing can be challenging for new users, particularly those familiar with the more forgiving nature of C.
- While growing rapidly, the ecosystem for embedded development in Rust is less mature than C's, which has decades of accumulated libraries and tools.

A.1 Frameworks in Rust for Embedded Systems

Rust's ecosystem includes several frameworks designed to leverage its safety features and performance in embedded contexts. Notable among these are RTIC, which offers a real-time concurrency model, and Tock, an operating system for microcontrollers. These frameworks demonstrate Rust's capability to support both bare-metal applications and more complex operating system environments on embedded devices.

Embassy stands out as an advanced async/await executor tailored for embedded systems, facilitating the writing of asynchronous, non-blocking applications. It supports a variety of hardware platforms, among others including STM32 and NRF families, and peripherals, making it a versatile choice for developers looking to leverage modern Rust features in their embedded projects.

- Embassy utilizes Rust's async/await syntax, simplifying the management of complex asynchronous operations and improving code clarity and maintainability.
- By enabling non-blocking programming models, Embassy helps optimize resource usage and power efficiency, which is crucial for battery-powered or energy-sensitive applications. All this is possible without the overhead of threads, since async/await is cooperative, much like many applications written in C for embedded.
- The design of Embassy allows it to scale from small, single-core microcontrollers to more complex multi-core processors, supporting a wide range of application requirements.

Appendix B

Module v0.1 Design Files

Following pages include the outputs from the design stage of the LoRa module.

B.1 A Word About the v0.2

Due to the use of incompatible crystal oscillator in this version of the LoRa module a v0.2 was also designed. It includes the TCXO modification, as discussed in 4.1.1, an optimized pinout and some other minor edits. The v0.2 was not manufactured because the tuning of the front-end was not realized within the deadline of the project.

The modifications needed for the v0.1 to function had little impact on the results and the additional resources that would need to be invested in manufacturing a second version with other possible defects, that could be resolved and tested on the v0.1, was not justified.

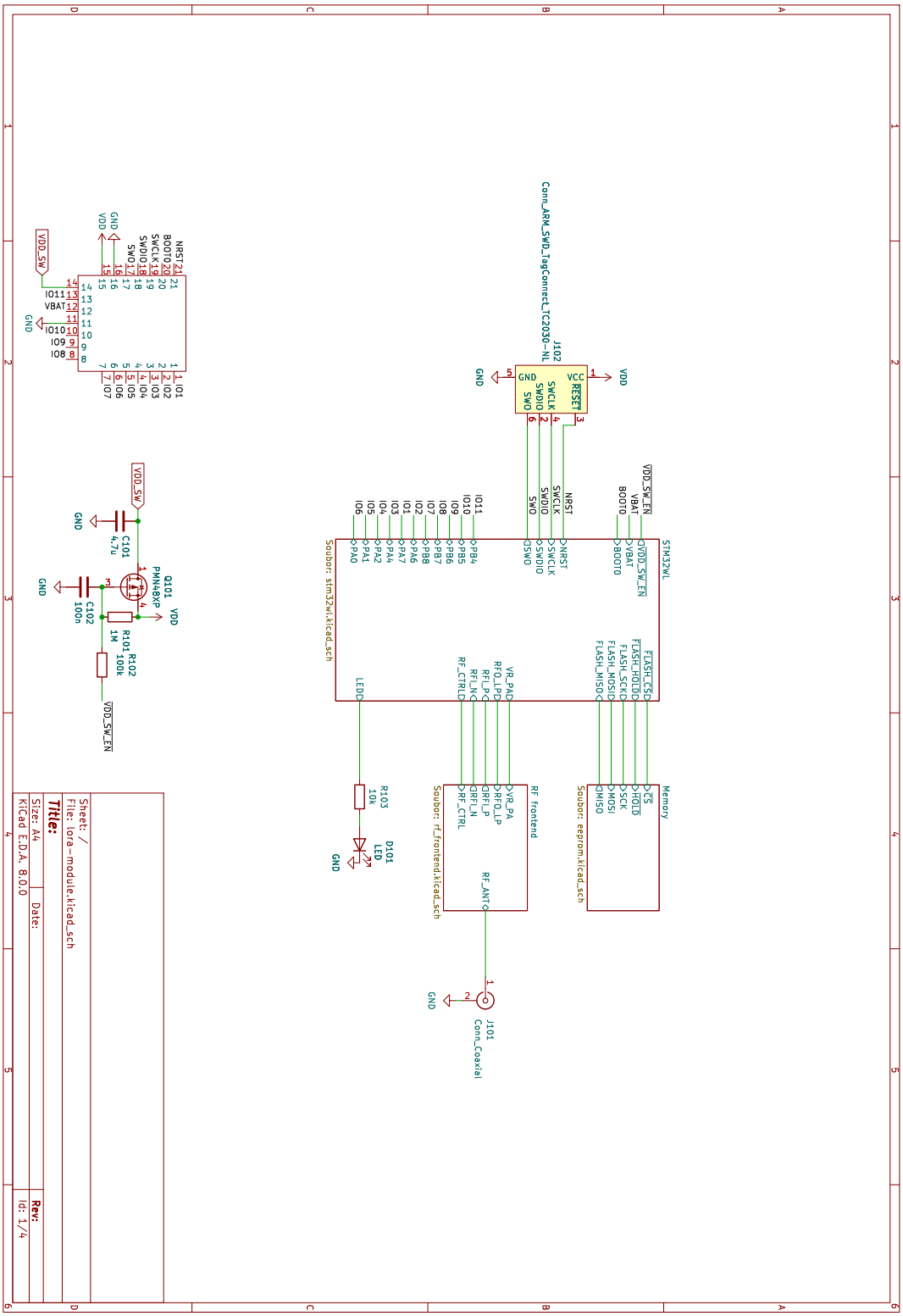


Figure B.1: Top level schematic sheet of the LoRa module schematic.

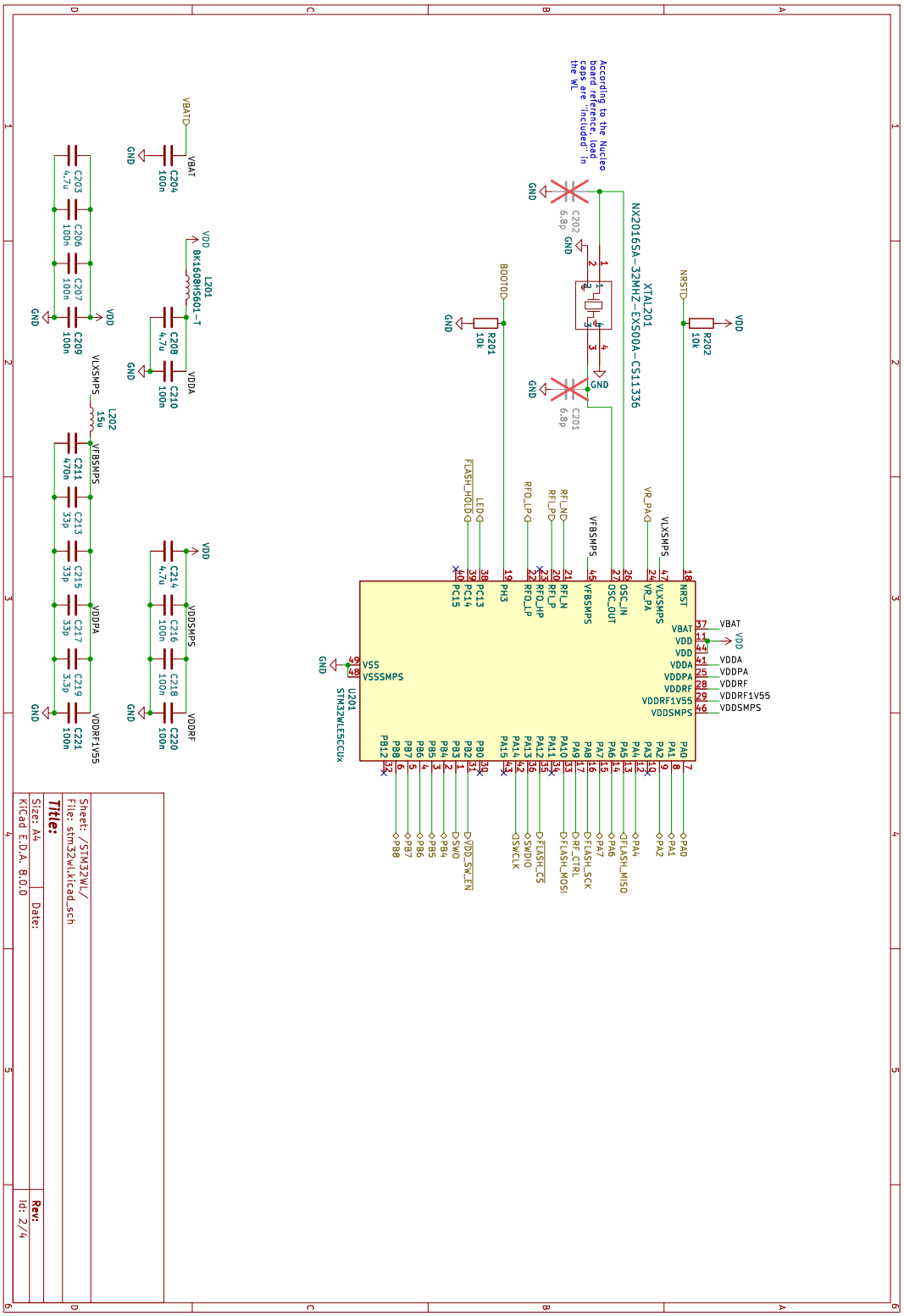


Figure B.2: STM32WLE5JC schematic sheet of the LoRa module schematic.

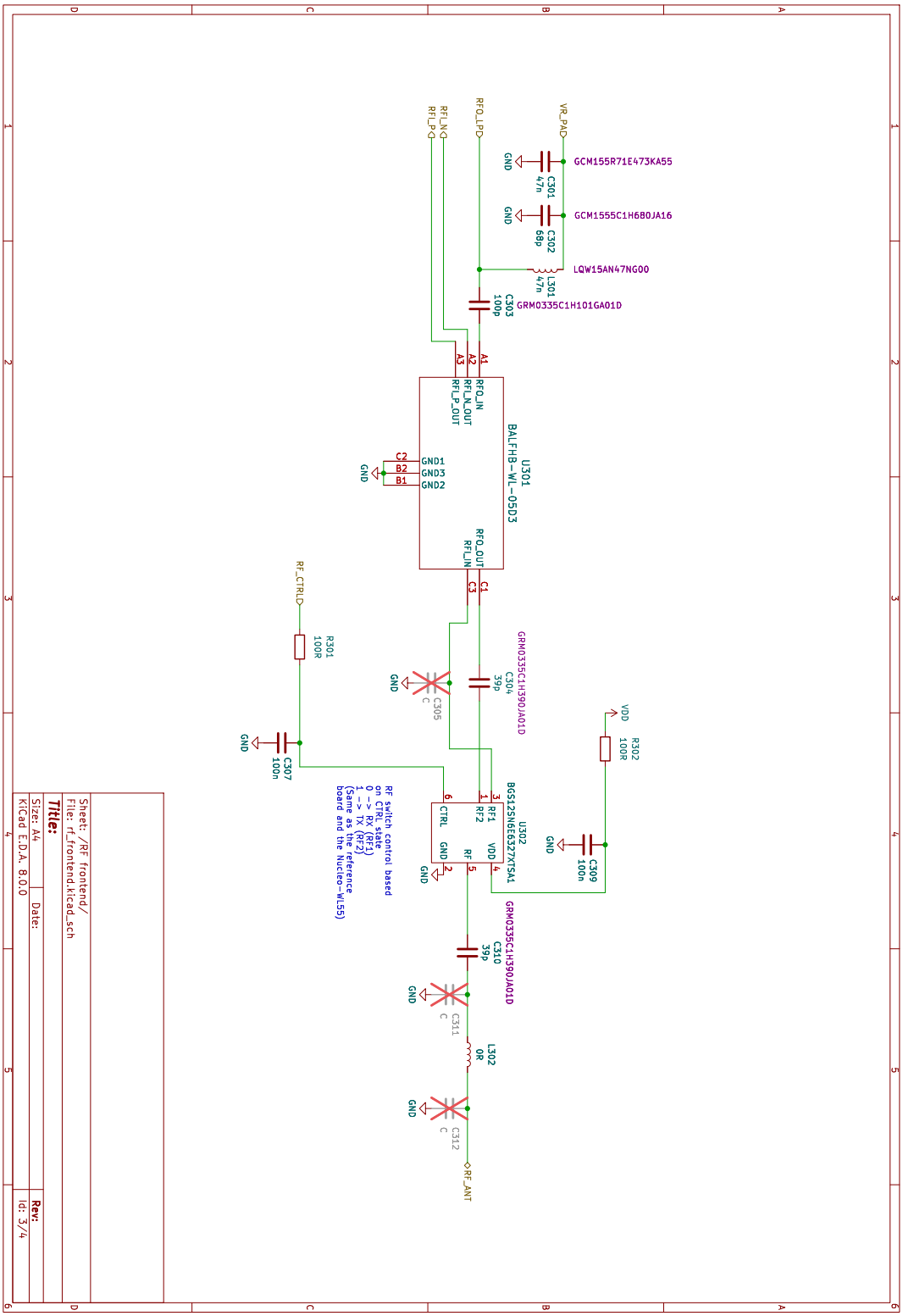


Figure B.3: RF frontend schematic sheet of the LoRa module schematic.

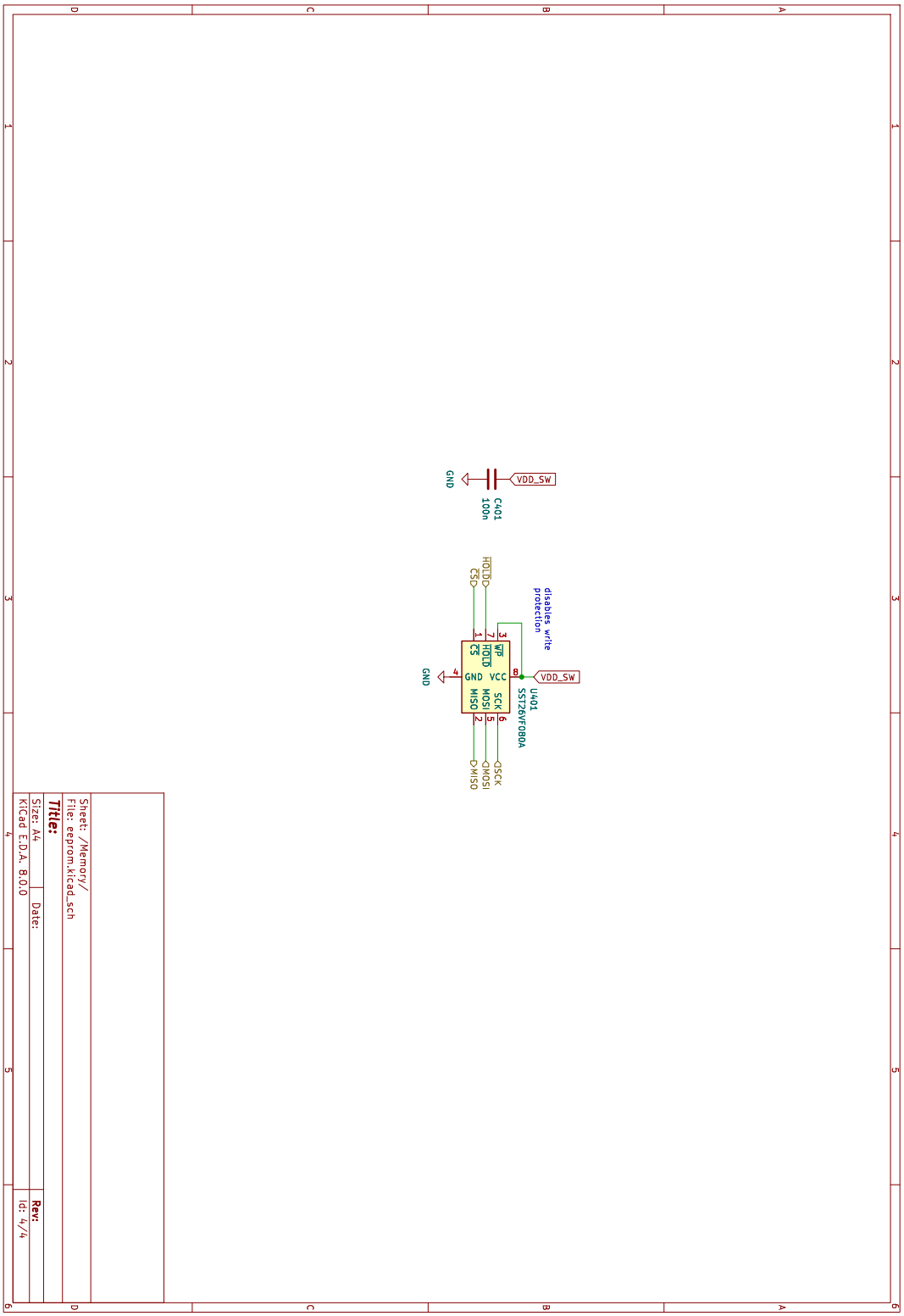
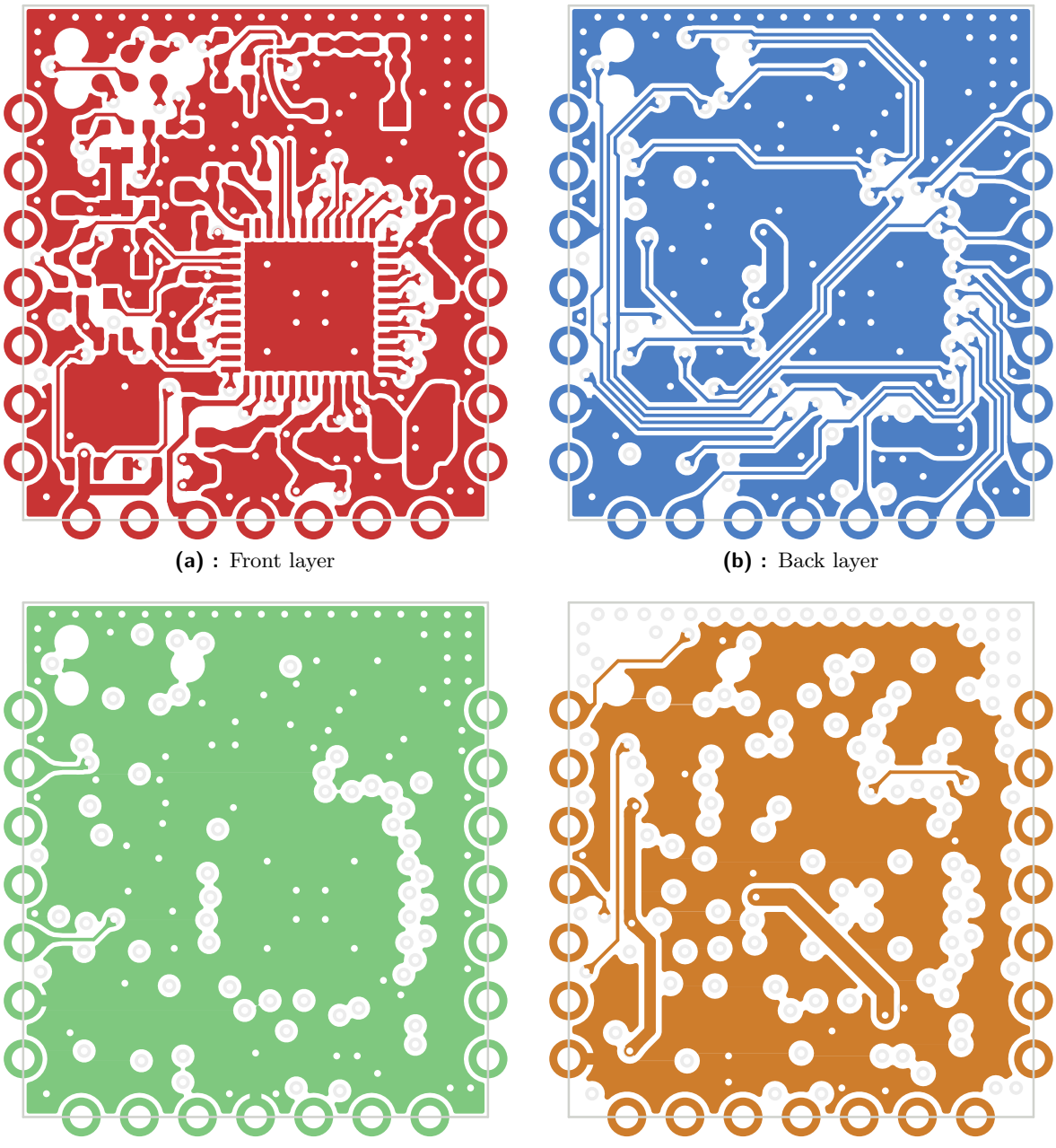


Figure B.4: Non-volatile memory schematic sheet of the LoRa module schematic.



(a) : Front layer

(b) : Back layer

(c) : Inner GND layer

(d) : Inner VCC layer

Figure B.5: Module v0.1 PCB layer design.

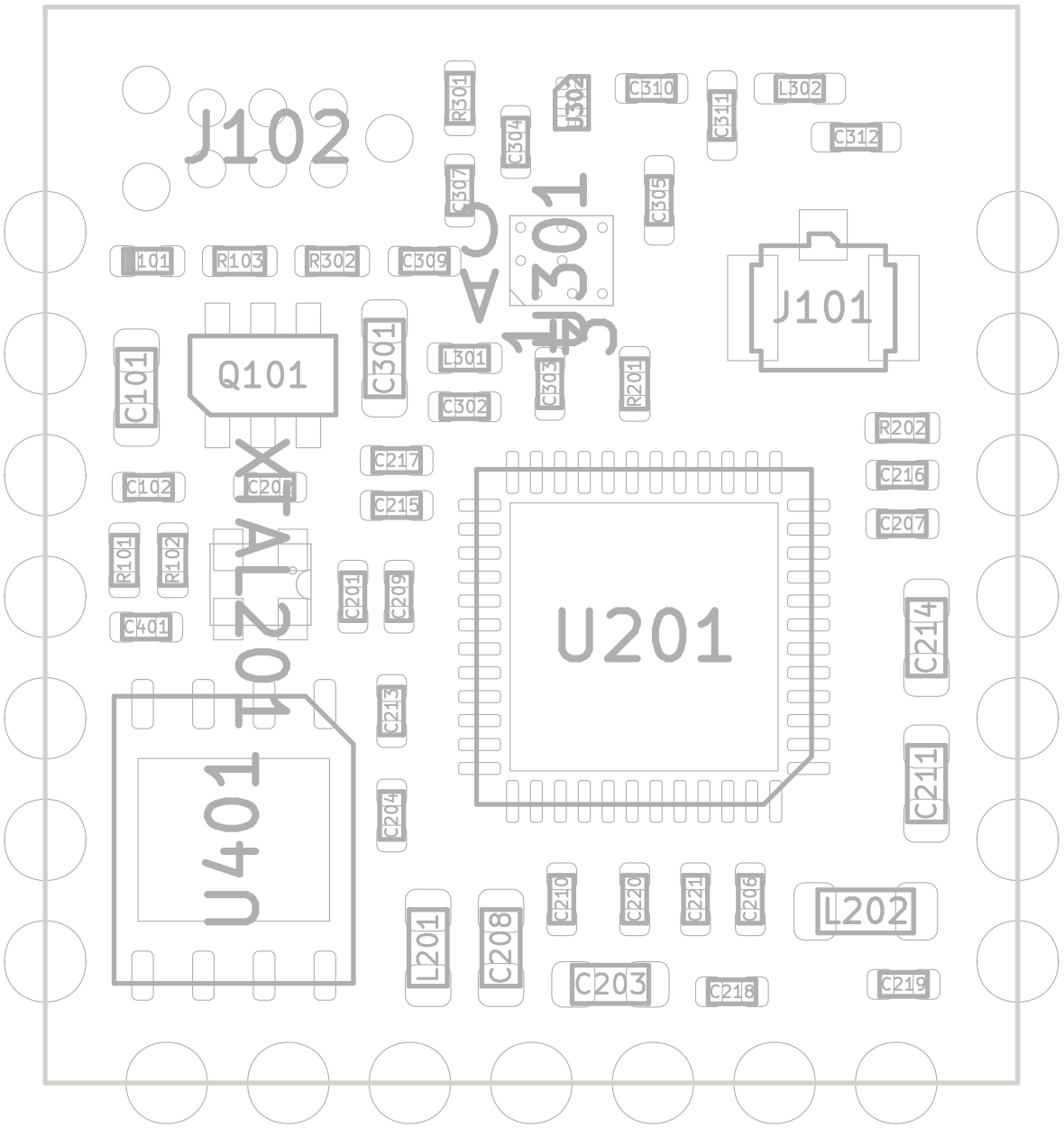


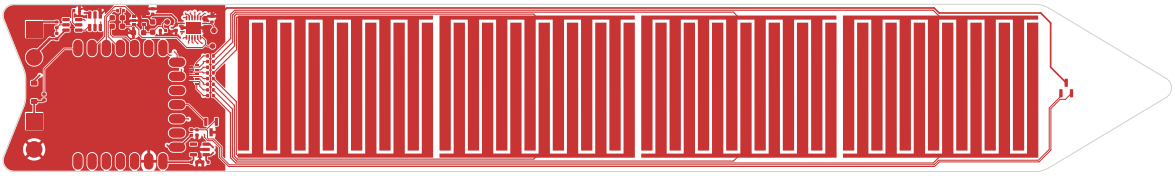
Figure B.6: Module v0.1 component position reference.

Qty	Reference(s)	Value
4	C101, C203, C208, C214	4.7u
4	C102, C204, C206, C207	100n
4	C209, C210, C216, C218	100n
5	C220, C221, C307, C309, C401	100n
2	C201, C202	DNP
1	C211	470n
3	C213, C215, C217	33p
1	C219	3.3p
1	C301	GCM155R71E473KA55
1	C302	GCM1555C1H680JA16
1	C303	GRM0335C1H101GA01D
2	C304, C310	GRM0335C1H390JA01D
3	C305, C311, C312	DNP
1	D101	B1861NB-05D000134U1930
1	J101	U.FL-R-SMT-1
1	L201	BK1608HS601-T
1	L202	MLZ2012M150W
1	L301	LQW15AN47NG00
1	L302	0R
1	Q101	PMN48XP
1	R101	1M
1	R102	100k
3	R103, R201, R202	10k
2	R301, R302	100R
1	U201	STM32WLE5CCUx
1	U301	BALFHB-WL-05D3
1	U302	BGS12SN6E6327XTSA1
1	U401	SST26VF080A
1	XTAL201	NX2016SA-32MHZ-EXS00A-CS11336

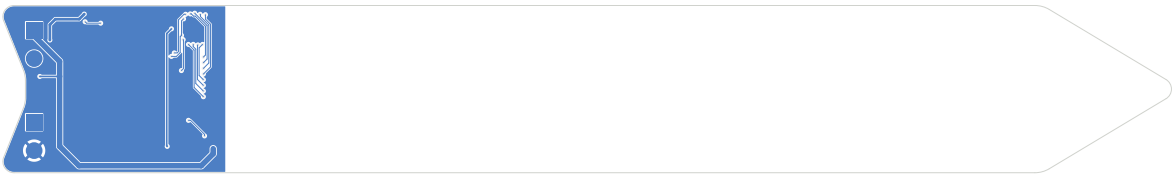
Table B.1: LoRa module Bill Of Materials (BOM).

Appendix C

Soil Moisture Sensor Board Design Files

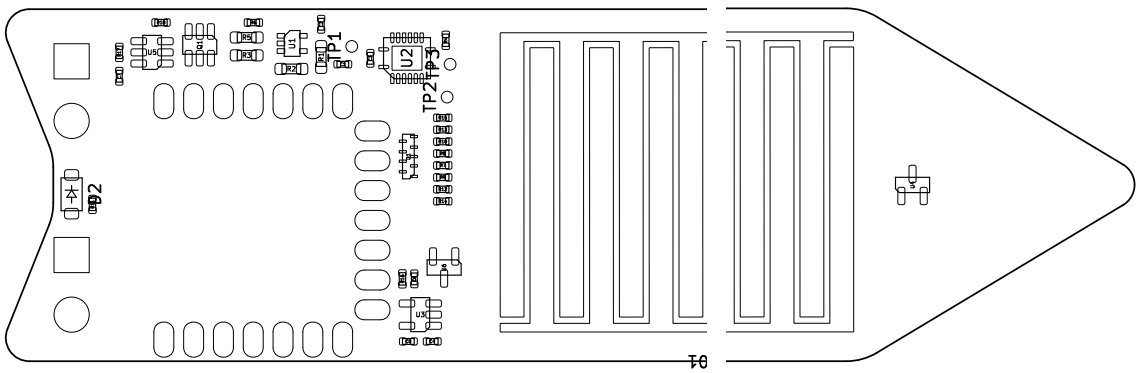


(a) : Front layer



(b) : Back layer

Figure C.1: Soil moisture sensor PCB layer design.



(a) : Electronics area **(b) :** Tip area

Figure C.2: Soil moisture sensor board component position reference.

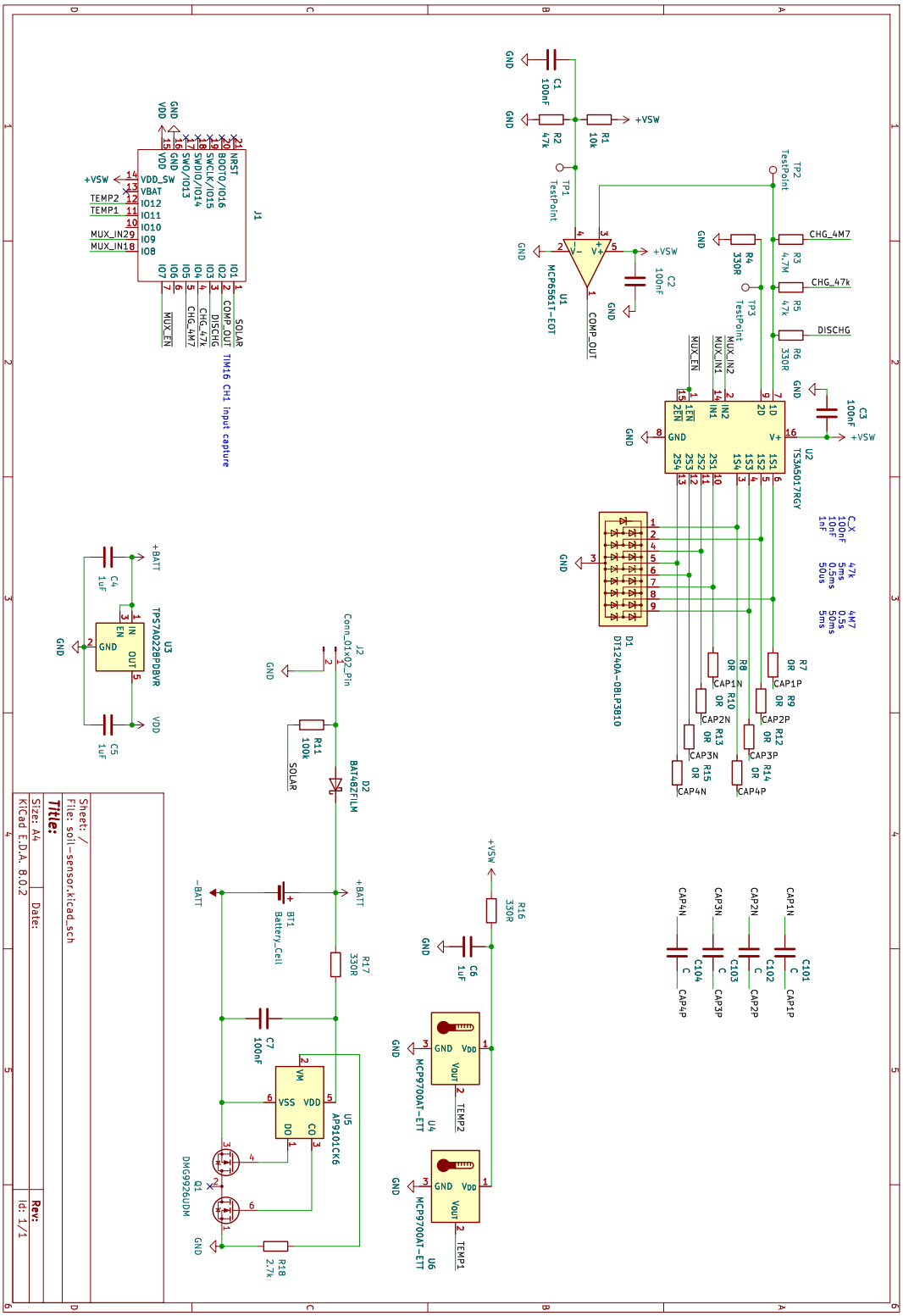
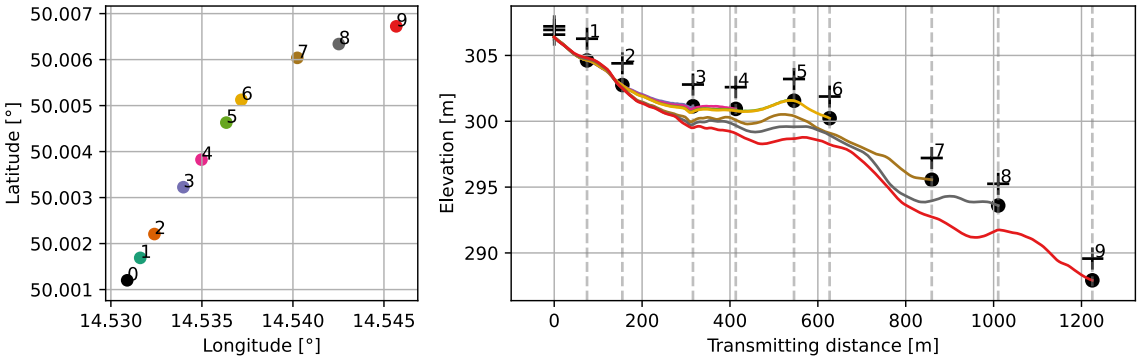


Figure C.3: Top level schematic sheet of the soil moisture sensor board.

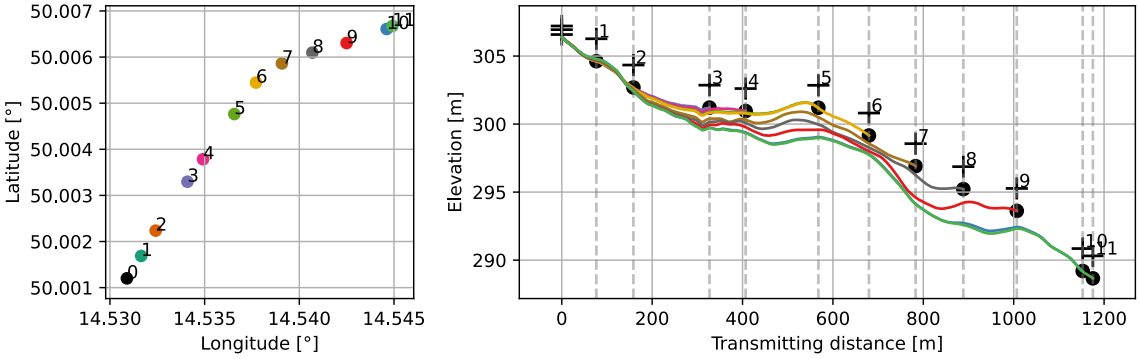
Appendix D

Additional Data from the Range Test

The script that generated these graphs can be found at thesis/data/range/process.py[1].



(a) : SF11



(b) : SF5

Figure D.1: Graph of the range test locations, their distance from the pole with Nodes, and the elevation in the path of the signal. Black crosses signify the position of the antennae of each device for each spot number.

[1]https://github.com/manakjiri/bcs-thesis/blob/main/thesis/data/range/process.py

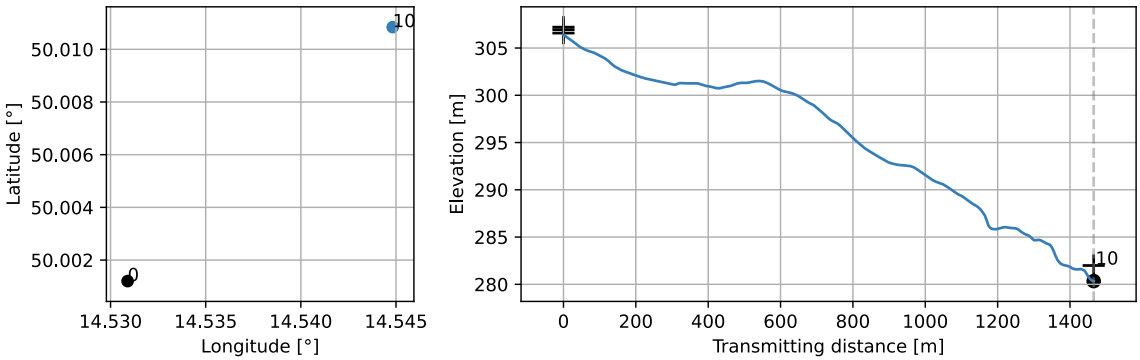
SF11

Spot	Latitude	Longitude
0	50.0012006	14.5308962
1	50.0016874	14.5316496
2	50.0022362	14.5324151
3	50.0032967	14.5340917
4	50.0037838	14.5349146
5	50.0047636	14.5365683
6	50.0054468	14.5377137
7	50.0058617	14.5390798
8	50.0060974	14.5406886
9	50.0063041	14.5424961
10	50.0066088	14.5446115
11	50.0066739	14.5449186

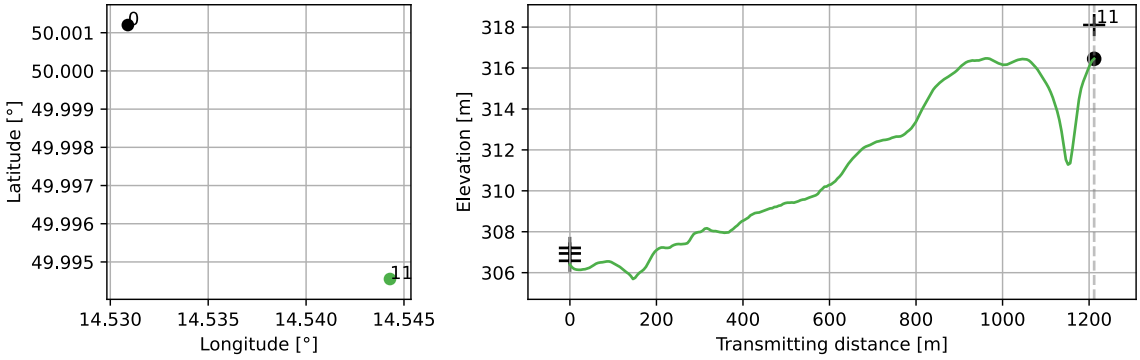
SF5

Spot	Latitude	Longitude
0	50.0012006	14.5308962
1	50.0016885	14.5316117
2	50.0022075	14.5323917
3	50.0032262	14.5339823
4	50.0038245	14.5349764
5	50.0046297	14.5363408
6	50.0051263	14.5371679
7	50.0060381	14.5402405
8	50.0063383	14.5425206
9	50.0067234	14.5456749

Figure D.2: Precise coordinates of each spot in the range test.



(a) : SF11-FAR1



(b) : SF11-FAR2

Figure D.3: Graph of the range test locations, their distance from the pole with Nodes, and the elevation in the path of the signal. Black crosses signify the position of the antennae of each device for each spot number.

Bibliography

[1] Humidity and Moisture Measurement. In *The Measurement, Instrumentation and Sensors Handbook*. CRC Press, Boca Raton, Fla, 1st edition edition, December 1998.

[2] Water content, May 2023. Page Version ID: 1154274797. URL: https://en.wikipedia.org/w/index.php?title=Water_content&oldid=1154274797.

[3] Android OTA updates, May 2024. URL: https://source.android.com/docs/core/ota.

[4] Amazon Inc. Amazon.com: Amazon Sidewalk: Amazon Devices & Accessories. URL: https://www.amazon.com/Amazon-Sidewalk/b?ie=UTF8&node=21328123011.

[5] Amphenol Inc. MiCS-VZ-89TE Integrated Sensor Module | Telaire, April 2024. URL: https://www.amphenol-sensors.com/en/telaire/524-voc/3285-mics-vz-89te.

[6] Audun Andersen. Selecting antennas for low-power wireless applications. *Analog Applications Journal*, 2008. URL: https://www.ti.com/lit/an/slyt296/slyt296.pdf.

[7] Baseline. Soil Moisture Sensors | Baseline Systems, May 2021. URL: https://www.hydropoint.com/baseline/products/soil-moisture-sensor/.

[8] Bosch Sensortec GmbH. bst-bme280-ds002.pdf, February 2024. BST-BME280-DS001-24. URL: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf.

[9] Benjamin Bucklin Brown. Over-the-Air (OTA) Updates in Embedded Microcontroller Applications: Design Trade-Offs and Lessons Learned | Analog Devices, May 2024. URL: https://www.analog.com/en/resources/analog-dialogue/articles/over-the-air-ota-updates-in-embedded-microcontroller-applications.html.

[10] CzechProject spol s.r.o. Půdní analogový vlhkoměr s antikorozní sondou v2.0 | dratek.cz, May 2024. URL: https://dratek.cz/arduino/174699-pudni-analogovy-vlhkomer-s-antikorozni-sondou-v2.0.html.

[11] Department of Sustainable Natural Resources. SOIL SURVEY STANDARD TEST METHOD, April 2024. URL: https://www.environment.nsw.gov.au/resources/soils/testmethods/mc.pdf.

[12] DigiKey. RF and Wireless category, April 2024. URL: https://www.digikey.com/en/products/filter/rf-antennas/875.

[13] Drogue IoT. Firmware updates, part 1: Bootloader, May 2024. URL: https://blog.drogue.io/firmware-updates-part-1.

[14] Edaphic Scientific Pty Ltd. How to convert gravimetric soil water content to volumetric soil water content, April 2024. URL: https://edaphic.com.au/soil-water-compendium/how-to-convert-gravimetric-soil-water-content-to-volumetric-soil-water-content/.

[15] Embassy Project Documentation. Bootloader :: Embassy Docs, May 2024. URL: https://embassy.dev/book/dev/bootloader.html.

[16] EOS Data Analytics. Remote Soil Moisture Monitoring For Precision Agriculture, December 2022. URL: https://eos.com/products/crop-monitoring/custom-solutions/soil-moisture-analytics/.

[17] ETSI. Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz; Part 2: Harmonised Standard for access to radio spectrum for non specific radio equipment, June 2018. URL: https://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.02.01_60/en_30022002v030201p.pdf.

[18] European Commission. Soil health, May 2024. URL: https://environment.ec.europa.eu/topics/soil-and-land/soil-health_en.

[19] Infineon Technologies. BGS12SN6 - Infineon Technologies, April 2024. URL: https://www.infineon.com/cms/en/product/rf/rf-switches-spxt-dpxt/bgs12sn6/.

[20] Infineon Technologies. Presence detection and sensing - Infineon Technologies, April 2024. URL: https://www.infineon.com/cms/en/applications/solutions/sensor-solutions/presence-detection/.

[21] Kyocera. ISM 868 & 915 MHz Embedded Ceramic Antenna. May 2024. URL: https://datasheets.kyocera-avx.com/ethertronics/AVX-E_M620720.pdf.

[22] Linx Technologies. HE Series 868 MHz Helical Antenna, May 2024. URL: https://cz.mouser.com/datasheet/2/418/9/ENG_DS_ant_868_hexx_A-3237873.pdf.

[23] John Mark. IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor Information Model. ISBN: 9780738117683. URL: https://ieeexplore.ieee.org/document/841361/, doi:10.1109/IEEESTD.2000.91313.

[24] METER Group. TEROS 10 - METER Group. Section: Uncategorized. URL: https://metergroup.com/products/teros-10/.

[25] METER Group. Soil moisture sensors—How they work. Why some are not research grade - METER Group, May 2023. Section: Measurement Insights. URL: https://metergroup.com/measurement-insights/soil-moisture-sensors-how-they-work-why-some-are-not-research-grade/.

[26] METER Group. TEROS 54 - METER Group, May 2024. URL: https://metergroup.com/products/teros-54/.

[27] Microchip Technology Inc. RN4871. URL: https://www.microchip.com/en-us/product/rn4871.

[28] Mobility Connected. What are OTA Updates?, January 2024. URL: https://connectedmobility.co/what-are-ota-updates/.

[29] Mouser Europe. 868 MHz Antenna catalog, April 2024. URL: https://eu.mouser.com/c/passive-components/antennas/?q=868.

[30] MyJoVE Corporation. Determination of Moisture Content in Soil | Environmental Microbiology | JoVE, April 2024. URL: https://www.jove.com/v/10011/determination-of-moisture-content-in-soil.

[31] Onion Corporation. Omega2S – Onion. URL: https://onion.io/omega2s/.

[32] Eldor Paul, editor. *Soil Microbiology, Ecology and Biochemistry*. Academic Press, Amsterdam Paris, 3rd edition edition, January 2007.

[33] PCBWay. Stackup for 4, 6, 8,..., 18 layers Multi-layer laminated structure - Custom PCB Prototype the Easy Way - PCBWay, April 2024. URL: https://www.pcbway.com/multi-layer-laminated-structure.html.

[34] Zachariah Peterson. Inverted-F Antenna Design For a PCB, February 2023. URL: https://resources.altium.com/p/inverted-f-antenna-design-pcb.

[35] PIXMAN s.r.o. Objednáváte připojení k IoT síti LoRa. URL: https://www.iotport.cz/objednavka-lorawan.

[36] Erika Podest and Amita Mehta. Applications of Remote Sensing to Soil Moisture and Evapotranspiration.

[37] seeedstudio. Wio-E5-LE Wireless Module (Bulk) - STM32WLE5JC, ARM Cortex-M4 and SX126x embedded, supports LoRaWAN on EU868 & US915, April 2024. URL: https://www.seeedstudio.com/Wio-E5-LE-Wireless-Module-Bulk-STM32WLE5JC-p-5751.html.

[38] Semtech Corporation. SX1261/2 Datasheet, May 2024. URL: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000Un7F/yT.fKdAr9ZAo3cJLc4F2cBdUsMftpT2vsOICP7NmvMo.

[39] STMicroelectronics. How to select a 32 MHz HSE oscillator for STM32WL5x/Ex MCUs. URL: https://www.st.com/resource/en/application_note/an5646-how-to-select-a-32-mhz-hse-oscillator-for-stm32wl5xex-mcus-stmicroelectronics.pdf.

[40] STMicroelectronics. STM32WLE5xx STM32WLE4xx Datasheet. URL: https://www.st.com/resource/en/datasheet/stm32wle5c8.pdf.

[41] STMicroelectronics. Ambient Light Sensors - STMicroelectronics, April 2024. URL: https://www.st.com/en/imaging-and-photonics-solutions/ambient-light-sensors.html.

[42] STMicroelectronics. BALFHB-WL-05D3 - Integrated filter matched balun to QFN STM32WL in low power mode, 862-928 MHz, using 4 layer PCB - STMicroelectronics, April 2024. URL: https://www.st.com/en/wireless-connectivity/balfhb-wl-05d3.html.

[43] STMicroelectronics. Human Presence Detection and Sensing - ICs, Design & Solutions - STMicroelectronics, April 2024. URL: https://www.st.com/en/applications/sensing/presence-sensing.html.

[44] STMicroelectronics. NUCLEO-WL55JC - STM32 Nucleo-64 development board with STM32WL55JC MCU, SMPS, supports Arduino and morpho connectivity - STMicroelectronics, April 2024. URL: https://www.st.com/en/evaluation-tools/nucleo-wl55jc.html.

[45] STMicroelectronics. STDES-WL5U4ILH - 4-Layer Reference design based on IPD "BALFHB-WL-05D3" with STM32WL UFQFPN MCU - High Band - Low Power - STMicroelectronics, April 2024. URL: https://www.st.com/en/evaluation-tools/stdes-wl5u4ilh.html.

[46] STMicroelectronics and seeedstudio. LoRa E5 Module - Module featuring ARM Cortex-M4 and SX126x embedded, supports LoRaWAN on EU868 & US915 - STMicroelectronics, April 2024. URL: https://www.st.com/en/partner-products-and-services/lora-e5-module.html.

[47] Taoglas. ILA.08 868MHz 5*3*0.5mm -0.5dBi Ceramic Loop Antenna. URL: https://cdn.taoglas.com/datasheets/ILA.08.pdf.

[48] Texas Instruments Inc. Light sensors | TI.com, April 2024. URL: https://www.ti.com/sensors/specialty-sensors/ambient-light-sensors/overview.html.

[49] The Things Network. EU863-870 MHz Band. URL: https://www.thethingsnetwork.org/docs/lorawan/regional-parameters/eu868/.

[50] The Things Network. Message Types. Section: lorawan. URL: https://www.thethingsnetwork.org/docs/lorawan/message-types/.

[51] The Things Network. The Things Network. URL: https://www.thethingsnetwork.org/.

[52] u-blox. MAX-F10S module, March 2024. URL: https://www.u-blox.com/en/product/max-f10s-module.

[53] Richard Wallace. AN058 – Antenna Selection Guide.

[54] Zachariah Peterson. Microstrip Patch Antenna Calculator for RF Designers, November 2022. URL: https://resources.altium.com/p/microstrip-patch-antenna-calculator-rf-designers.