

# **Automatic sampling of water content in the soil using lysimeters**

**Pedro Miguel Coelho Fernandes**

Thesis to obtain the Master of Science Degree in

## **Electrical and Computer Engineering**

Supervisor(s): Prof. Doctor João Nuno de Oliveira e Silva  
Doctor Maria Alexandra Soares Gomes Cardoso de Oliveira

### **Examination Committee**

Chairperson: Prof. Teresa Maria Sá Ferreira Vazão Vasques  
Supervisor: Prof. Doctor João Nuno de Oliveira e Silva  
Member of the Committee: Prof. Martijn Kuipers

**September 2021**



## **Declaration**

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.



Dedicated to my family.



## **Acknowledgments**

Firstly, I would like to thank Prof. João Silva and Maria Alexandra Oliveira for giving me the opportunity of working on this project and for all the guidance and help provided.

I would also like to thank Altice for providing the SIM card, as it was a fundamental component to this project, and Companhia das Lezírias for allowing the installation of a monitoring site on their property.

To my friends that helped and supported me throughout these last years, thank you. These journey was much more enjoyable thanks to you.

Lastly, a special thanks to my family and girlfriend that always supported and encouraged me. I would not be here without you.





## Resumo

À medida que maior atenção é dada aos efeitos da poluição atmosférica nos ecossistemas e na saúde humana, novas medidas e leis são estabelecidas, com o objetivo de melhor compreender estes efeitos e legislar a emissão de gases poluentes. Um grande número destas medidas é focado no estudo e monitorização da concentração de poluentes nos ecossistemas, nomeadamente, nas plantas, água e solo. Os lisímetros são dispositivos frequentemente utilizados na amostragem da água do solo, dada a sua acessibilidade e facilidade de utilização para monitorizar os solos. No entanto, são normalmente operados manualmente, o que muitas vezes é inconveniente, uma vez que o utilizador precisa de esperar por condições ótimas de amostragem antes de extrair uma amostra, o que pode ser difícil de prever. Além disso, países e áreas com baixas taxas de precipitação têm apresentado dificuldades em manter frequências de monitorização e amostragem semelhantes às indicadas nos manuais de monitorização referência indicados pela Comissão Europeia. Isto acontece devido à falta de água no solo, perpetuada pela diminuição dos níveis de precipitação, que é esperada agravar devido ao aquecimento global. Esta tese propõe dois dispositivos capazes de simplificar o processo de amostragem de água no solo. Um dispositivo desenvolvido em torno de um lisímetro, capaz de detetar condições ótimas de amostragem da água do solo e de extrair uma amostra automaticamente, necessitando de intervenção mínima do utilizador. Uma vez extraída uma amostra, o utilizador é alertado e procede à recolha da amostra. O segundo dispositivo é uma iteração mais simples que o primeiro, onde o utilizador é alertado quando são detetadas condições ótimas de amostragem da água do solo, e o mesmo procede à extração e recolha manual da amostra. A comunicação dos dispositivos com o utilizador é baseada em tecnologias IoT, e através de uma plataforma é possível gerir e interagir com os dispositivos instalados. Estes dispositivos, junto com a plataforma, facilitarão significativamente o processo de amostragem, aumentando a frequência da mesma, e permitirão ao utilizador gerir facilmente um grande número de dispositivos espalhados por uma grande área.

**Palavras-chave:** Poluição, Monitorização de Solos, Amostragem de água, IoT, Lisímetro



## Abstract

As more awareness is given to the effects of air pollution on ecosystems and human health, new measures and laws are established in order to better understand these effects and legislate emission ceilings. A great number of these measures are focused on studying and monitoring the concentration of pollutants in the ecosystems, namely, in plants, water and soil. Lysimeters are devices often used in soil water sampling given their accessibility and ease of use for monitoring soils. Nevertheless, they usually are operated manually which can be often inconvenient, as the user needs to wait for optimal sampling conditions before extracting a sample, which may be hard to predict. Furthermore, countries and areas with low precipitation rates have shown difficulty in keeping monitoring and sampling frequencies similar to those indicated in reference manuals provided by the European Commission. This happens due to lack of water in the soils, perpetuated by diminishing precipitation levels, which are expected to further decrease due to global warming. This research project proposes two devices capable of simplifying the sampling extraction process. One device built around a lysimeter, capable of automatically detecting optimal soil water sampling conditions and extracting a sample with minimal user intervention. Once a sample is extracted, the user is alerted and proceeds to collect the sample. The second device is a simpler iteration of the first one, where the user is alerted, when optimal soil water sampling conditions are detected, and proceeds to extract and collect the sample manually. The device communication is based on IoT technologies, and through a platform it is possible to manage and interact with the devices installed. These devices, together with the platform developed, will significantly ease the sampling procedure, increase sampling frequency, and allow the user to easily manage a large number of devices spread throughout a large area.

**Keywords:** Pollution, Soil Monitoring, Water Sampling, IoT, Lysimeter



# Contents

Acknowledgments . . . . .	vii
Resumo . . . . .	ix
Abstract . . . . .	xi
List of Tables . . . . .	xvii
List of Figures . . . . .	xix
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problems . . . . .	2
1.3 Proposal . . . . .	3
1.4 Thesis Outline . . . . .	4
<b>2 Background and Related Work</b>	<b>5</b>
2.1 Legislation . . . . .	5
2.1.1 European Legislation . . . . .	5
2.1.2 Implementation of the Directives in Portugal . . . . .	6
2.2 Sampling of Soil Water Content . . . . .	7
2.2.1 Suction Lysimeters . . . . .	9
2.2.2 Automated Lysimeters . . . . .	10
2.2.3 Water retention in different types of soil . . . . .	11
2.3 Internet of Things (IoT) technologies . . . . .	12
2.3.1 Internet of Things (IoT) wireless technologies . . . . .	13
2.3.2 IoT Application Layer Protocols . . . . .	15
2.3.3 IoT mechanisms applied to the Natural Environment . . . . .	17
2.4 Micro-Controller and Sensor Technologies . . . . .	18
2.4.1 Micro-Controllers . . . . .	18
2.4.2 Soil Humidity Sensors . . . . .	20
<b>3 Design</b>	<b>21</b>
3.1 Requirements . . . . .	21
3.2 Architecture . . . . .	22
3.2.1 Sampling Hardware . . . . .	23

3.2.2	Power Supply . . . . .	24
3.2.3	Operation Control . . . . .	24
3.2.4	Communication . . . . .	25
3.2.5	Web Application . . . . .	25
<b>4</b>	<b>System Implementation</b>	<b>27</b>
4.1	Hardware . . . . .	27
4.1.1	Sampling Hardware . . . . .	27
4.1.2	Sensors . . . . .	29
4.1.3	Communication Module . . . . .	30
4.1.4	Micro-Controller . . . . .	31
4.1.5	Final Setup . . . . .	33
4.2	Communication . . . . .	34
4.2.1	Communication Module . . . . .	36
4.2.2	Communication Protocols . . . . .	37
4.3	Software . . . . .	40
4.3.1	Arduino Pro Mini . . . . .	40
4.3.2	Procedures . . . . .	42
4.3.3	Server . . . . .	43
4.3.4	Deployment . . . . .	49
4.4	Testing Device . . . . .	50
<b>5</b>	<b>System Evaluation</b>	<b>51</b>
5.1	Device Calibration . . . . .	51
5.1.1	Pressure Sensor . . . . .	51
5.1.2	Soil Humidity Sensors . . . . .	52
5.2	Soil Humidity Threshold Study . . . . .	53
5.2.1	Experiment Setup . . . . .	53
5.3	Battery . . . . .	57
5.3.1	Device Power Consumption . . . . .	57
5.3.2	Battery Performance . . . . .	58
5.4	Cost . . . . .	59
5.5	Laboratory Experiment . . . . .	59
5.6	Field Tests . . . . .	61
5.6.1	Communication . . . . .	61
<b>6</b>	<b>Discussion</b>	<b>63</b>
6.1	Requirements . . . . .	63
6.2	Communication . . . . .	64
6.3	Hardware . . . . .	64

6.3.1	Notificator Device . . . . .	64
<b>7</b>	<b>Conclusions and future work</b>	<b>67</b>
	<b>Bibliography</b>	<b>69</b>





# List of Tables

2.1	Climatic information in the monitoring sites based on data from the Digital Climatic Atlas of the Iberian Peninsula, collected from 1950 to 1999 [7] . . . . .	6
2.2	Measured effective saturation degree for each soil textural group [22] . . . . .	12
2.3	IoT wireless technologies comparison . . . . .	15
2.4	IoT Application Layer Protocols comparison . . . . .	17
2.5	Micro-Controllers price and power consumption . . . . .	19
4.1	Legend of the components in the figure 4.1 . . . . .	28
4.2	BC66-TE-B Interfaces Description [59] . . . . .	31
4.3	Arduino Pro Mini Technical Specs [61] . . . . .	32
4.4	Legend of the components in the figure 4.12 . . . . .	35
4.5	Events and their corresponding message parameters . . . . .	39
4.6	Device requests and corresponding server responses . . . . .	39
4.7	HTTP Server endpoints and their description . . . . .	49
5.1	Minimum and maximum sensor output values obtained during the calibrations of three sensors . . . . .	53
5.2	Power consumption per component . . . . .	57
5.3	Device and its materials prices . . . . .	60
6.1	Notifier Device and its materials prices . . . . .	66



# List of Figures

1.1	Lysimeter setup currently in use on the Portuguese sampling sites with a vacuum syringe [9] . . . . .	2
1.2	Trends in annual (left) and summer (right) precipitation across Europe between 1960 and 2015 [10] . . . . .	3
2.1	Location of monitoring sites [7] . . . . .	7
2.2	Standard Installation setting of a Suction Lysimeter [8] . . . . .	10
2.3	Alternative Installation setting of a Suction Lysimeter . . . . .	10
2.4	Wireless communication technologies for Internet of Things [26] . . . . .	13
2.5	Data rate and Power Consumption Analysis for different Wireless Communication Technologies [28] . . . . .	14
3.1	Full Collector Hardware Architecture Diagram . . . . .	23
3.2	Exemplification of the device extraction process using an electric vacuum pump and a suction lysimeter . . . . .	23
3.3	Full Collector Device Architecture Diagram . . . . .	26
4.1	Implementation of the sampling hardware . . . . .	27
4.2	30cm Soil Solution Access Tube . . . . .	28
4.3	Installation of lysimeters at different depths . . . . .	28
4.4	Soil Humidity Sensor Recommended Installation Depth [56] . . . . .	29
4.5	Soil Humidity Sensor isolated with epoxy glue . . . . .	29
4.6	BC66-TE-B Arduino Interface Definition [59] . . . . .	30
4.7	Grove Base Shield V2.0 [63] . . . . .	32
4.8	Grove Connector [65] . . . . .	32
4.9	Pro Mini UNO Shield Adapter Board, from OPEN-SMART [64] . . . . .	33
4.10	Driver circuit for controlling the vacuum pump and valves . . . . .	34
4.11	Final device implementation diagram . . . . .	34
4.12	Implemented device . . . . .	35
4.13	The Fundamental Principle of QuecOpen Software Architecture [68] . . . . .	36
4.14	BC66-TE-B Software Architecture . . . . .	38
4.15	Configuration Procedure Diagram . . . . .	40

4.16 Arduino Pro Mini Software Architecture . . . . .	42
4.17 Sleep Mode Procedure Diagram . . . . .	44
4.18 Server Software Architecture Diagram . . . . .	44
4.19 Database Entity Relationship Diagram . . . . .	46
4.20 Testing device circuit diagram [71] . . . . .	50
5.1 Barometer negative pressure values as a function of MPX4250DP sensor output voltage .	52
5.2 Sensor output evolution during the calibration process, with the sensor installed placed on air and submerged on water . . . . .	52
5.3 Representation of the container used and its cylinder equivalent . . . . .	54
5.4 Volume of water added, water leaked, water in the container and water extracted accord- ing to the water level of the corresponding iteration of the organic soil experiment . . . . .	55
5.5 Volume of water added, water leaked, water in the container and water extracted accord- ing to the water level of the corresponding iteration of the sand experiment . . . . .	55
5.6 Variation of the soil humidity and vacuum pressure throughout the iterations of the organic soil experiment, with water levels equal to 0, 15 and 30mm . . . . .	56
5.7 Variation of the soil humidity and vacuum pressure throughout the iterations of the sand experiment, with water levels equal to 0, 5 and 10mm . . . . .	56
5.8 Battery Level Evolution during the first test . . . . .	59
5.9 Battery Level Evolution during the second test . . . . .	59
5.10 Server logs created during the laboratory experiment . . . . .	59
5.11 MEO Cellular Network Coverage . . . . .	61
6.1 Notificator Device Architecture Diagram . . . . .	65
6.2 Notificator Software Architecture Diagram . . . . .	65

# Chapter 1

## Introduction

This chapter briefly describes the legislation and directives, based on pollution monitoring and soil water sampling, that compose the thesis motivation. The problems that this thesis aims to solve are also described, mainly the difficulty in extracting soil water samples in mainland Portugal. Lastly, two devices capable of solving said problems, by simplifying and automating the extraction process, and introduced.

### 1.1 Background

In order to better the air quality and reduce the pollution impact on human health and ecosystems, the European Union established the Directive EU 2016/2284 [1]. This Directive states that the Member States should reduce anthropogenic atmospheric emissions of polluting gases. Furthermore, each nation needs to implement national air pollution control programmes, so that those emissions are monitored and reported, as well as their impacts in the ecosystems.

In the case of Portugal, the implemented program and its measures are defined in the Portuguese decree-law nº 84/2018 (23rd October 2018) [2]. One of the measures delineated in the decree-law is collecting and analyzing several chemical components in the soil, vegetation and water within the soil, in several sampling sites throughout the Portuguese territory.

The sampling sites stationed in Portugal are equipped with manual sampling systems composed of lysimeters. Lysimeters are devices that allow the extraction of water from the soil for analytical purposes [3]. There are several types of lysimeters with different structures and extraction methods, depending on the purpose for which they were designed [4–6]. The ones installed in the Portuguese sampling sites are Suction Lysimeters [7], depicted in figure 1.1. Lysimeters of this type can extract water when under vacuum pressure and, normally, require the operation to be performed by an in-situ operator. This process preserves the sample quality which allows for accurate methods of measuring the soil salinity, nutrients, pH, etc. [8].

In Portugal the sampling sites are installed in relatively isolated areas where the water source is precipitation. So the probability of a lysimeter being able to extract a sample successfully depends heavily on the occurrence of precipitation. Given the type of lysimeters used, to collect a sample, the

operator needs to proceed to the sampling site after a precipitation event, to collect the sample. These events are often hard to time and it is not guaranteed that the soil is humid enough for a sample to be extracted. These factors make extracting samples a very complicated process resulting in sampling frequencies much lower than the stipulated values.

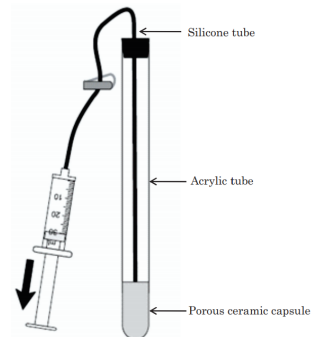


Figure 1.1: Lysimeter setup currently in use on the Portuguese sampling sites with a vacuum syringe [9]

## 1.2 Problems

As stated before, in Portugal, the water source for the sampling sites is precipitation. Due to the global warming, high water deficit (evapotranspiration minus precipitation) is expected to become a frequent phenomenon. This phenomenon, among other important issues, affects the sampling rate in monitoring sites of the effect of air pollution in terrestrial ecosystems, which should be of two weeks, maximum, ideally shorter. According to the European Environment Agency [10] and depicted in the figure 1.2, since 1960 the North-Eastern and North-Western Europe have shown an increasing trend in annual precipitation, with increases up to 70 mm per decade. On the other hand, Southern Europe has shown decreases up to 90 mm per decade. The Iberian Peninsula, especially central Portugal have shown large declines, of up to 90 mm per decade, as well. On top of this, mean summer precipitation has been decreasing an additional 20 mm per decade in most Southern European countries [10].

The already low, and decreasing, precipitation rates have provided increasing difficulties in collecting samples at the recommended frequency, as the soils are not saturated enough to perform soil water extractions [7].

As already stated, the current systems in place requires regular travels to the sampling sites, right after precipitation events, where many times the water collection is unsuccessful due to low water content in the soil. At times, the precipitation level is not enough to saturate the soil to acceptable humidity values, other times soil saturation decreases within the necessary time for the operator to reach the monitoring site, rendering water sample extraction using manual suction lysimeters impossible. This is not cost-effective, since sampling sites are spread throughout Portugal, with large distances between them, and plans to increase its quantity are under way, following the European Commission suggestions. Besides, it is not possible for the operator to know which precipitation level or soil humidity levels allow for extractions nor to measure the soil humidity remotely. Another important factor is that trips to the

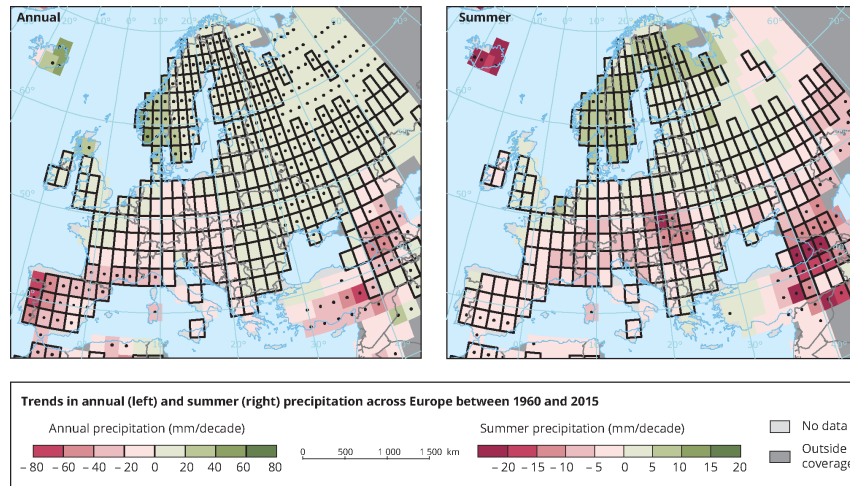


Figure 1.2: Trends in annual (left) and summer (right) precipitation across Europe between 1960 and 2015 [10]

sampling locations come with costs, in time and money, so unsuccessful samplings can unnecessarily increase these costs.

This fact is especially evident in areas with low precipitation throughout the year. Another important aspect that impacts the extraction process is the soil texture (proportion between sand, silt and clay). Different types of soil retain different amounts of water for different time periods, which influences the time window for when a sample is available for extraction. This added complexity hampers the scheduling of the extraction even further. Seeing that the optimal water content in the soil, for a successful sampling, depends on the amount of precipitation in the region, it is hard to schedule the sample extraction. Thus, most times, the extraction success highly depends on luck.

### 1.3 Proposal

The main objective of this research project is to ease the sampling process explained beforehand, solving the problems listed above. This was achieved through two devices developed throughout this project:

- **Full Collector:** Electro-mechanical device that collects water samples from the soil after raining, when the water content in the soil is optimal. A Soil Humidity Sensor will measure the soil water content allowing the system's controller to detect its optimum level. This device automates the majority of the extraction process, so when a sample is collected the system will store it and alert the operator that a sample is ready for collection. In order to pull the water out of the soil, a Suction Lysimeter is used in conjunction with an Electric Vacuum Pump. The device messages a central server which alerts the operator through an email message, where the operator will only need to collect the extracted sample and reset the device.
- **Notificator:** This device is a simpler iteration of the first one. It only detects when the soil water content is adequate for sampling extraction and alerts the user, of this event. In this case, the

technician will need to travel to the sampling site promptly, while the soil conditions still allows for extraction, and manually extract the sample.

Both devices need to be of low maintenance, meaning the only physical interactions with the user should be limited to collecting a sample and/or changing the battery, ideally only once or twice a year. They will also provide useful data of the sampling site's soil water content progression throughout the year.

The operation of these systems also needs to take into account the soil texture, because different types of soil require different humidity threshold values in order to successfully collect the required quantity of water for laboratory analysis. In order to calibrate these levels, a third device was introduced. This device will assist the operator in calibrating the humidity threshold levels. The operator will try to manually extract water samples from a soil sample collected upon installation of a device in the monitoring site, testing different humidity levels, and observing which range of values allow for extraction of the required water quantities (approximately 50 ml). These values will then be inputted into the two systems mentioned above, before their installation, according to the soil profile in the installation site, or remotely changed during the device operation.

## **1.4 Thesis Outline**

Chapter 1 briefly described the thesis motivation and the problems that it aims to solve, mainly the difficulty in extracting soil water samples in mainland Portugal. It also briefly describes two devices capable of simplifying and automating the extraction process. In chapter 2, the European and Portuguese legislation behind the thesis motivation is explained, including the reasons that lead to the creation of said legislation and the current measures implemented in European and Portuguese territory. This chapter also introduces multiple mechanisms capable of performing sampling of soil water content, how they work and their advantages and disadvantages. In addition, it also describes current devices, available in the market, and studies that are based on available mechanisms. Lastly, Chapter 2 also describes several Internet of Things technologies that may be used in the proposed devices communication. These technologies are studied and analyzed in the project context, and their architecture, advantages and disadvantages are discussed. Several implementations of these technologies are described in order to verify their behavior. In Chapter, 3 the requirements of the devices are presented, as well as a proposed architecture that complies with the predefined requirements. Chapter 4 specifies an implementation of the architecture of one of the devices previously defined. All the required hardware and software components are described, as are their advantages and limitations. Chapter 5 presents the results of several experiments undertaken using the implemented device. These experiments include sensors calibration, soil humidity measurement and analysis and device performance. In Chapter 6 the problems that emerged during this thesis are detailed and possible solutions are discussed. Chapter 7 describes the conclusions drawn from this thesis and multiple future improvements that can be applied to the work developed during this thesis.



## Chapter 2

# Background and Related Work

The following chapter explains the legislation behind the thesis motivation. Multiple mechanisms capable of performing sampling of soil water content are described, as well as current devices and studies that are based on available mechanisms. Lastly, several Internet of Things technologies that may be used in the proposed devices communication are introduced.

## 2.1 Legislation

### 2.1.1 European Legislation

On December 2016 the European Parliament approved the Directive (EU) 2016/2284. It states that in order to reduce the levels of air pollution and its effect on human health and ecosystems, every Member State needs to establish national air pollution control programmes. These programmes are primarily focused on reducing the anthropogenic atmospheric emissions of sulphur dioxide ( $\text{SO}_2$ ), nitrogen oxides ( $\text{NO}_x$ ), non-methane volatile organic compounds (NMVOC), ammonia ( $\text{NH}_3$ ) and fine particulate matter ( $\text{PM}_{2.5}$ ) [1].

Over the past decades, unsustainable land use, emissions of pollutant gases, use of toxic or hazardous substances in households, factories and agriculture, has lessened soil quality. These events also have negative impacts on ecosystems, habitats for several species and biodiversity, thus implementing measures around soil sustainability is a very important step towards accomplishing said objectives [1, 11]. The effectiveness of national emission reductions is, in part, evaluated by monitoring its impacts on terrestrial and aquatic ecosystems. The Directive (EU) 2016/2284, also named the NEC (national emission ceilings) directive, specifically legislates the establishment of ecosystems monitoring networks to collect a set of defined indicators, that include soil solution chemical analysis [1].

According to the International Co-operative Programme on Assessment and Monitoring of Air Pollution Effects on Forests (ICP Forests) [12], in order to get coherent soil water samples, the sampling period should be no longer than a month. If this sampling frequency is not possible, a monthly sampling could be performed, depending mainly on climate. As for the sample size, a volume of about 50 ml is sufficient for measuring the required concentrations using the analytical techniques available at the

national environmental agency (Agência Portuguesa do Ambiente). It is mandatory to sample at fixed selected depths: 0-20 cm, 20-40 cm and 40-80 cm.

## 2.1.2 Implementation of the Directives in Portugal

The Portuguese decree-law nº84/2018 [2], implemented on 23rd October 2018, marks the transposition of the Directive EU 2016/2284 by Portugal and the implementation of the said National air pollution control programme. The required monitoring is described in the National Report 2019 for Ecosystem Monitoring in Portugal [7].

Three terrestrial and three freshwater sampling sites are situated throughout Portugal, with the terrestrial sites being located in Mata Nacional de Leiria, Área Florestal de Sines e Mata Nacional Terras da Ordem. These sites locations are identified in the figure 2.1, where each of them has different temperature and precipitation profiles [7]. Currently, there are no lysimeters installed in Mata Nacional Terras da Ordem (site code 89351 in figure 2.1), since the soil is poorly developed and it doesn't reach the required sampling depths. Therefore, taking the IPC Forest recommendations into account, several lysimeters were installed in two locations in mainland Portugal, with 2-3 additional monitoring sites in planning stage (location still not defined).

Table 2.1: Climatic information in the monitoring sites based on data from the Digital Climatic Atlas of the Iberian Peninsula, collected from 1950 to 1999 [7]

Site code national	Monitored Ecosystem	Average Temperature [°C]	Precipitation [mm]
625087	Mata Nacional de Leiria	15.3	841
236130	Área Florestal de Sines	17.1	651
89351	Mata Nacional Terras da Ordem	17.6	588

Table 2.1 reports the location name, temperature and precipitation statistics of the sampling sites. Figure 2.1 illustrates elevation, temperature and precipitation maps, where the sampling locations are marked. There is a clear tendency for the temperature to rise and precipitation to decline, while approaching southern territory. As stated before, lack of precipitation may hinder the rate at which samples are available for extraction, thus the Área Florestal de Sines site may provide fewer samples than the Mata Nacional de Leiria one.

Another important factor to consider from the sampling locations is the prominent soil type in which they are installed. Mata Nacional de Leiria develops on silica-rich sand, while Área Florestal de Sines monitoring site is located over an extensive hummocky dune field, with sandy textures but higher clay content [7].

Given the differences in climate and soil type in each sampling location, it is expected that each location provides unique sampling patterns and characteristics. Differences in sampling frequency and sample volume being the most relevant for this work.

As stated before, the number of sampling sites is expected to increase, with one of the new proposed locations being Companhia das Lezírias.

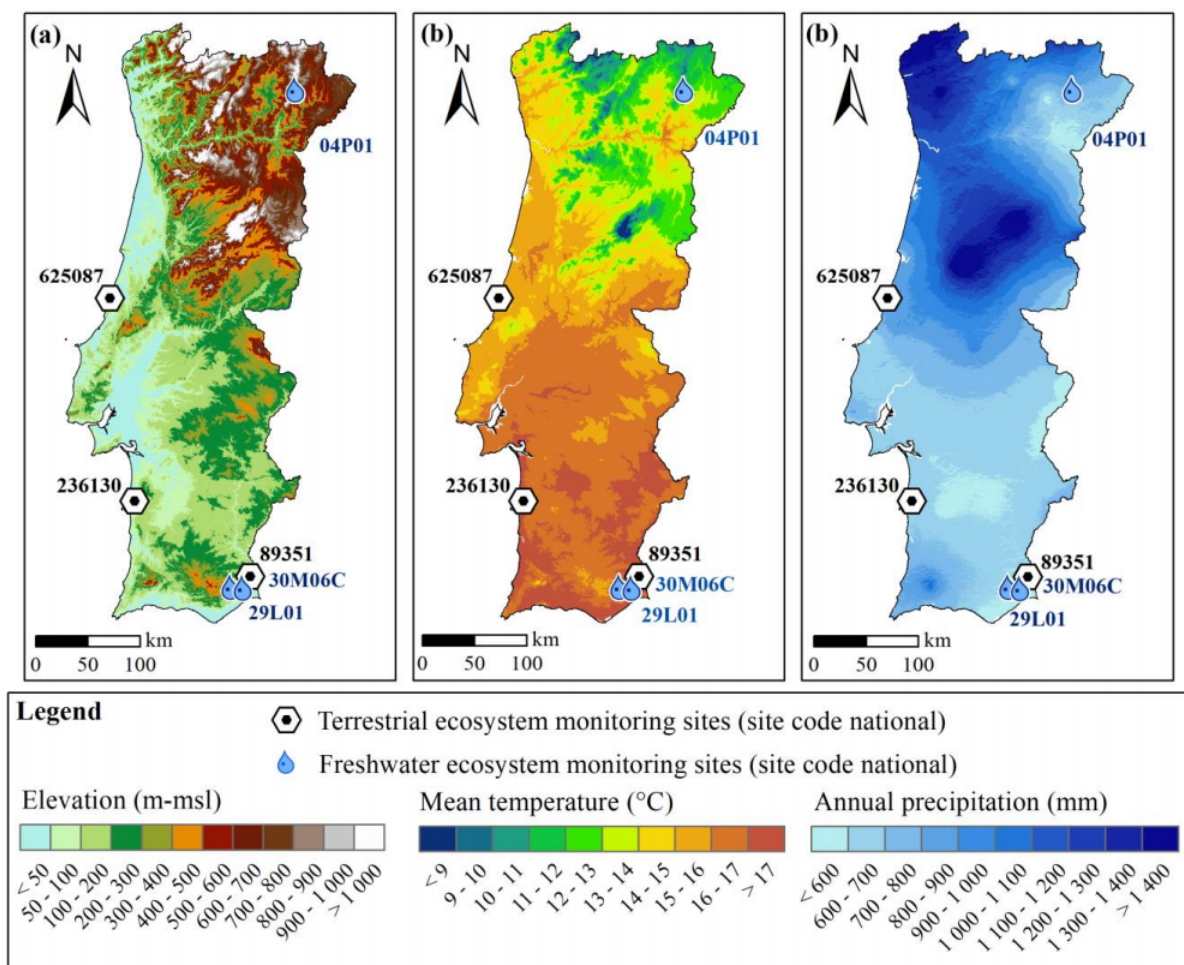


Figure 2.1: Location of monitoring sites over: (a) Elevation above mean sea level (msl); (b) average annual temperature; (c) Annual precipitation map. The ASTER Global Digital Elevation Model is a product of METI and NASA. Temperature and precipitation data from the Digital Climatic Atlas of the Iberian Peninsula [7]

As previously stated, at maximum, the sampling frequency should be monthly. However, given the low levels of precipitation in mainland Portugal, no sample has been collected since the installation of lysimeters in the two monitoring sites, due to low water contents, even after rainy periods. In fact, in every attempt to manually extract soil water after a precipitation event, the precipitation level was not enough to saturate the soil and, consequently, to extract a sample. This means that the timeframes when it is possible to extract a sample are rare and need to be fully exploited. Although this issue was already reported to the European Commission, the effects of air pollution in terrestrial ecosystems are still at an early stage of reporting, and, so far, no solution has yet been identified, neither by the Commission, nor by participating member states.

## 2.2 Sampling of Soil Water Content

Soil water sampling is an essential process not only in the context of this thesis, but in other important areas, such as agriculture, to assess leaching of nutrients due to excess fertilization [13].

In order to assess ecosystem chemical balances, soil, plant and soil water chemical analysis is paramount. Specifically, for soil water, physical and chemical parameters, such as pH, conductivity, nitrogen, phosphorous, potassium, calcium magnesium, aluminum, sulfur, dissolved organic carbon, and others, are important in determining nutrient balance in the soil and soil acidity [12]. These elements in the soil water, together with chemical characterization of soil and leaves, are required to assess the status of ecosystems. As the effect of air pollution in the chemical balance in ecosystems is very important, the need for fast and reliable soil water sampling methods were created, especially automated ones, able to maximize higher frequency of sampling in drylands.

In situ soil water sampling systems can be divided into two categories, active sampling and passive sampling. In situ systems are characterized by being installed directly at a certain depth within original soil, so the samples maintain their quality and characteristics allowing for accurate measures and data [14]. Passive systems, are normally formed of containers that define a boundary for water collection [4]. In these systems, the water in the sampling locations drains directly to the container. These systems are easier to automate since they don't require active devices that displace water, as the collection is done by itself, but they can disrupt the soil and change the natural field conditions. On the other hand, active sampling systems, such as those used in this work, are not very intrusive to the soil but require an external force to extract water samples [14, 15].

Singh et al. [16] analyzed over 300 research articles on water sampling and collection, enumerating and describing the most common methods and devices available. Methods like drainage lysimeters, pan lysimeters, ion exchange resin bags and membranes being the most common passive samplers, while wick lysimeters, suction lysimeters and suction plates appear as the most common active samplers.

### **Passive water sampling**

Drainage lysimeters, also referred as water columns, are the oldest and most accurate method of water sampling. Unexpectedly, they are also the most popular method in measuring water content and water fluxes. They are described as large containers installed into the soil where water drains to. Usually, they are equipped with an weighing balance or some type of water sensor that measures water content. Measuring the balance and sensor data over long periods of time can help determine water fluxes and evaporation rates. This type of lysimeter is usually used in crops [3–5, 16].

Pan lysimeters are another passive water sampling alternative. This type of lysimeter is a low-cost alternative to drainage lysimeters, with smaller containers, and specialized in capturing gravitational flows of water, like precipitation. Given their low cost of production, this lysimeters are used in many environmental studies. Pan lysimeters are commonly used in forests and cropping systems [5, 16]. Its main disadvantages is the soil disruption required to create a collection chamber, siting lower than the sampling depths, as the water is collected into separate recipients by gravity.

Another type of passive water samplers are ion exchange resin bags and membranes. Resin bags possess an anion and cation beads, while membranes possess a thin film made with anion and cation membranes. These materials absorb the nutrients from the water, so in reality this devices can be defined as water solute samplers. These devices are also installed into the soil, so that water infiltrates

into contact with the anion and cation, where its ions are absorbed. Ion exchange bags and membranes are a common use in forest ecosystems studies and agricultural systems, as they are suitable for long studies in remote locations [16]. Unfortunately, these methods are not indicated in the reference manuals for monitoring the effects of air pollution in terrestrial ecosystems.

### **Active water sampling**

Suction lysimeters are specially prominent given their accessibility, and are the most studied water samplers in reviewed research articles [16]. Suction lysimeters, in junction with soil moisture measures, may provide feasible approaches for automated soil water extraction during drainage periods, with great transferability [14]. Advanced systems focused on automated in situ water sampling, based on suction lysimeters, are currently installed in Berlin [14].

Suction plates are very similar to suction lysimeters in its operating principles. The main differences rely on the shape and size of the plates. The suction plates occupy a larger area which can boost sample's volume, but also increase the system cost. Given their characteristics they are harder to automate and install in sampling systems [16].

Wick lysimeters are an intermediate device between the passive and active scope. They don't require an external devices to power the water collection, but use an additional wick material. This material, by having low water potential energy, directs the water in the involving soil to a container connected to said material. Although this method works for most cases, the device is still more expensive to construct than similar types of lysimeters, like the pan lysimeter. Despite this, wick lysimeter systems are easily scalable in size and adaptable to many uses [16].

#### **2.2.1 Suction Lysimeters**

As stated before, suction lysimeters are the water sampling systems installed in the several terrestrial ecosystem monitoring sites, in mainland Portugal. Therefore, and given that this works objective was to automate soil water sampling, these are the lysimeters used in this project. For this particular type of lysimeter, which is also often referred as Suction Cup [15], the sample extraction happens when vacuum pressure is exerted in a tube inside the lysimeter with a porous segment at the end, that allows the water to pass through its pores [8].

Suction lysimeters are, normally, attached to a tank or container, so when a sample is extracted it is placed in the container. Once the sample is inside the container, it won't be under pressure anymore, thus remaining in place until its manual collection.

Figure 2.2 shows the standard installation setting of a suction lysimeter. Inside the main tube, connected to the porous cup, there are two smaller tubes. One is connected to a vacuum pump which will provide pressure to the lysimeter and another tube transfers the collected water to the container. The porous cup can be covered with a silica mixture or a slurry, made out of water and soil from the installation depth, that helps the absorption of water from the soil, thus bringing water into contact with porous cup. Once the pores from the cup are substantially filled, the vacuum can pull the water from the

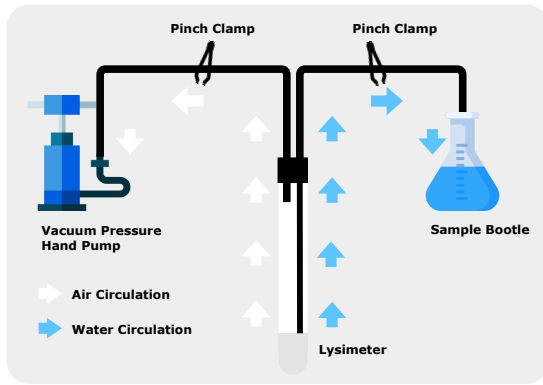


Figure 2.2: Standard Installation setting of a Suction Lysimeter [8]

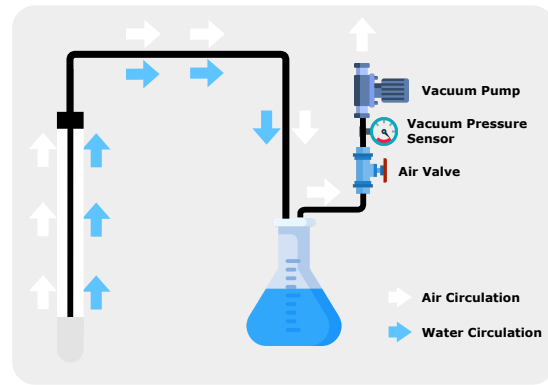


Figure 2.3: Alternative Installation setting of a Suction Lysimeter

cup to the tube which raises to the container, using the same vacuum pressure, similarly to the use of a straw [8]. When the soil humidity level is not high enough it is possible that the pressure built inside the lysimeter leaks through the porous cup and the evolving soil. When the soil is sufficiently humid the porous cup is better isolated to pressure leaks.

An alternative lysimeter installation setting is shown in the figure 2.3, where the sample bottle is installed between the pump and lysimeter, which allows a more controlled flow of water.

The manual operation has a price of 341.03€ including the three lysimeters, the manual vacuum pump and other extraction equipment, such as clamps and tubes. The price can also be considered of 224.49€, if the lysimeters are not considered. Although, the vacuum pump and other extraction equipment can be shared between different monitoring sites, lowering the average value of the operation.

Q. Wang et al. [15] analyzed the characteristics of conventional lysimeters, like drainage and weighing lysimeters, and suction lysimeters, identifying its advantages and disadvantages. In contrast to the other types of lysimeters, suction lysimeters are easier to install and operate. These traits allow for undemanding incorporation of lysimeters in autonomous systems, at low capital cost.

## 2.2.2 Automated Lysimeters

Several devices were implemented with the objective of automating the process of soil water sampling through lysimeters. Currently, there are several commercial automated lysimeters referred as smart lysimeters. From the options available in the market, two stand out: the Smart Field Lysimeter from Meter [17] and the Eijkelkamp Smart Lysimeter [18]. There are also other devices, based specifically on suction lysimeters, named Automated Vacuum Lysimeters (ASL), although their market is limited [19].

The Smart Field Lysimeter is built around a weighing lysimeter, with a height of 30 to 90cm, and provides additional features. One of those features is a bi-directional pump that automatically maintains true field conditions within the lysimeter. By using tensiometers and soil moisture sensors, the device detects if the soil outside of the lysimeter is drier or wetter than inside and pumps water in the direction that balances the field conditions. Other features include a data logger, an optional GPRS (General Packet Radio Service) modem for data transfer and remote access, solar panel power and others [17].

The Eijkelkamp Smart Lysimeter is also based on a weighing lysimeter that uses a system of weighing cells to measure water content changes in a soil column. Several sensors provide additional data such as soil moisture, matric potential, temperature and electrical conductivity. A telemetric system provides data reporting through a web portal [18].

These devices provide accurate and diverse data but are mainly used for measuring precipitation, evapotranspiration, and deep drainage, and not for water sampling. Besides, since they are based on weighing lysimeters, their installation is hard and disturbs the existing soil profile, despite the Smart Field Lysimeter being capable of simulating true field conditions.

ASL use electronic controllers as well as a set of vacuum sensors to keep the vacuum applied to suction lysimeters or suction plates [19]. ASL use tensiometers, or other methods, vacuum sensors and electronic controllers to measure soil-water tension and estimate the optimal level of vacuum that a suction sampler needs in order to extract a sample. This process makes ASL the most accurate method of soil water sampling. There are multiple types of automated suction lysimeters, such as the Controlled Suction Period Lysimeter (CSPL).

CSPL is the current most accurate ASL method. It measures the soil-water tension above the suction plate which allows the vacuum pressure exerted to be optimized and removes the need for calibration. However, its implementation and power efficiency are poor given its complexity.

ASL can operate and record data for long periods of time, with minimal supervision, while also being easy to replicate. However, given their complexity, it is hard to control multiple lysimeters simultaneously, as well as their power requirements, which make ASL hard to install in remote areas. In addition, the current market is very limited in cost-effective ASL solutions, despite the fact that technology capable of solving these problems is currently available [19].

A. Farsad et al. [20] developed "An Automated Suction on Lysimeter for Improved Soil Water Sampling", to measure post-harvest nitrate leaching from a rye cover crop field during the falls and winters of 2007 to 2009. The system was based on an suction plate, referred as suction lysimeter in the publication, and other components such as: tensiometers, vacuum pump, solenoid vacuum valves, vacuum sensors, an electronic controller, a data logger memory, and others. Furthermore, the ASL was tested in both laboratory and field environments. In the laboratory tests, the device was installed in a containerized soil column equipped with a precise water supply and drainage system. Each sampling period lasted 5 hours and the sampler extraction rates and drainage rates were recorded every minute. For the field tests, nine different ASL were installed in different terrain plots. Each ASL had two suction plates, which provided a wider variety of data. It was verified that, in general, the device was fairly reactive to the precipitation events, that occurred during the test phase, although the humidity sensors data discrepancy grew considerably after rain events.

### **2.2.3 Water retention in different types of soil**

Water retention and availability depend heavily on the type of soil. Characteristics such as grain size and chemical composition affect the moisture content and water potential of a soil, as well as its ability to

retain water [21]. Water potential refers to the water potential energy. As other types of potential energy, water potential quantifies the energy retained by a pool of water and its tendency to move.

In the case of a lysimeter, water retention of the soil directly affects its ability to extract water. In order for a lysimeter to extract water, the surrounding soil needs to be saturated, or close to, in water content at field capacity [8]. The soil saturation is directly affected by the water potential and water content. A soil is considered to be saturated when the water potential is equal to, or near, 0 and its water content is maximum. In this situation, since the potential is nonexistent, the water movement is largely reduced and very abundant around the lysimeter. These characteristics are optimal for the sampling procedure.

F. Meskini-Vishkaee et al. [22], predicted the soil moisture retention curve for different textures of soil. The prediction model used is based on the particle size for each soil type and provides accurate predictions. The obtained results are expressed in the table 2.2.

Table 2.2: Measured effective saturation degree for each soil textural group [22]

Soil Texture	Effective saturation degree $[L^3 L^{-3}]$
Clay	0.51
Clay loam	0.58
Loam	0.45
Silt loam	0.44
Silty clay	0.51
Silty clay loam	0.43
Loamy sand	0.40
Sand	0.37
Sandy clay loam	0.36
Sandy loam	0.46
<i>Average</i>	0.42

The effective saturation degree corresponds to a relation between maximum obtainable volume of water and volume of wet material. Which means, soils with higher saturation degrees are capable of accumulating larger quantities of water for the same soil volume [23]. Analyzing the table 2.2, is possible to conclude that clayish soils retain the most water. On the other hand, sandier soils retain less water, while loamy soils roam in the middle of the two. When a soil is saturated, the more water content there is, the easier it is to extract. Although, since soils with lower saturation degrees require less water to saturate, they may reach saturation faster than soils with higher saturation degrees, and provide faster extractions.

## 2.3 Internet of Things (IoT) technologies

Internet of Things, IoT, is a concept born through the integration of multiple processes, such as sensing, networking, and computation. This idea allowed all kinds of electrical and electronic devices to



be connected to the Internet, facilitating the interaction between them and the users. There are countless IoT applications but it is possible to group them in several major areas, such as health, traffic, logistics, retail, agriculture, smart cities, smart metering, remote monitoring, process automation, and many more [24] [25].

### 2.3.1 Internet of Things (IoT) wireless technologies

The fast growth in the IoT sector has been powered by the extensive spectrum of wireless technologies [25]. These technologies vary in many aspects, from infrastructure to the area that they cover. Figure 2.4 shows the most common wireless communication technologies used in the IoT panorama. These technologies are grouped in several types according to their coverage area [26, 27]:

- **Proximity:** Allows very short-range device identification and communication.
- **Wireless Personal Area Network (WPAN):** Used in inter-communications between devices in a personal work-space, like computers and smartphones.
- **Wireless Local Area Network (WLAN):** Allows communication between multiple devices inside a limited area such as schools, offices, and others.
- **Wireless Metropolitan Area Network (WMAN):** Used in wider areas and geographical regions allowing the connection of several WLANs inside those regions, for example.
- **Wireless Wide Area Network (WWAN):** Used to connect a large number of devices on a national and global scales. Built on top of cellular networks, satellite or other large infrastructures.

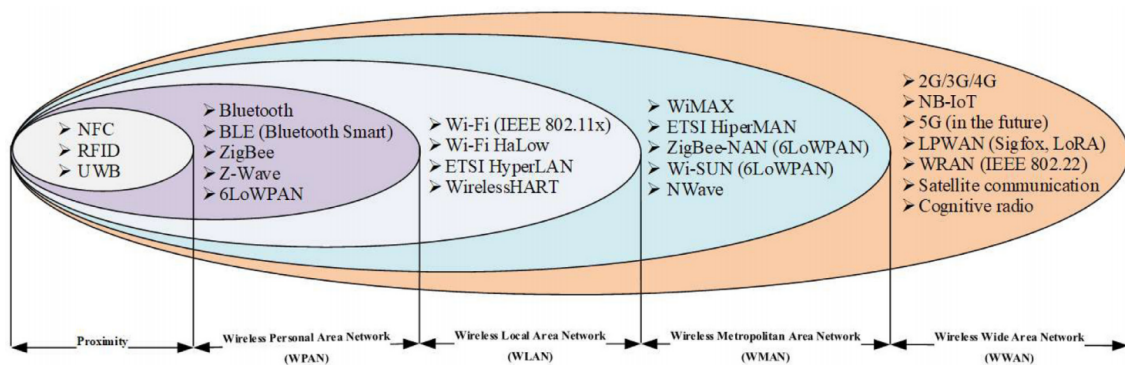


Figure 2.4: Wireless communication technologies for Internet of Things [26]

With a large number of IoT devices requiring long range transmission and low energy consumption, neither short-ranged technologies such as Wi-Fi and Bluetooth, or Cellular Communication, which consume large amount of power, could meet these requirements. In the WWAN category, some technologies can be referred as LPWAN (Low-Power Wide-Area Network) technologies. These technologies are designed to allow long range communications with lower data rate and latency requirements, while being adapted to battery powered devices. Figure 2.5 shows a comparison of some of the IoT wireless technologies, where the vertical axis represents the data rate and power consumption, while the range

is represented in the horizontal axis. The cost of each technology is also represented through different colors. It is possible to conclude that the LPWAN technologies provide the power efficient alternatives to long range communications with lower prices [27].

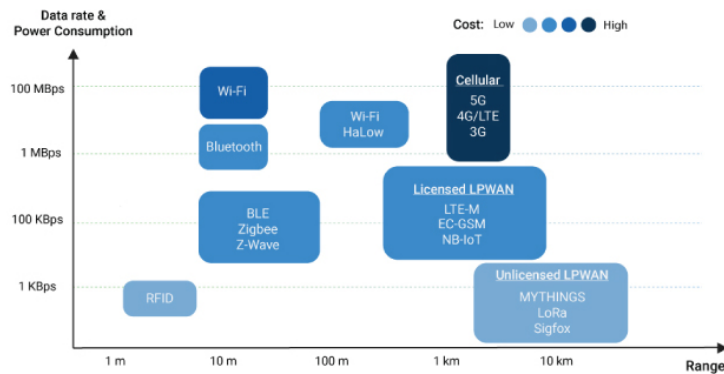


Figure 2.5: Data rate and Power Consumption Analysis for different Wireless Communication Technologies [28]

Since the terrestrial ecosystem monitoring sites are spread throughout hundreds of kilometers, LPWAN technologies may provide sufficient coverage to comply with this requirement, while allowing for low power consumption. Thus, a further description of several LPWAN technologies, such as LoRaWAN, SIGFOX, NB-IoT, is provided below.

## LPWAN Technologies

LoRaWAN is a network protocol used in many battery powered devices. Using gateways the devices are able to communicate with each other, via Internet, while a Network Server routes data from the devices to the Application Server. The communications are secured using symmetric cryptography. This protocol is built on top of an infrastructure that uses LoRa (Long Range) modulation, which uses unlicensed spectrum. The communication's frequency channels and data rates vary according to the network conditions, ranging from 0.3 kbps to 50 kbps. This allows the devices to save power. The devices can exchange messages bidirectionally with scheduled or continuous receive slots, depending on the device class [27, 29].

SIGFOX is a proprietary network of the SIGFOX company with similar characteristics to LoRa. One important characteristic being that SIGFOX gateways communicate through a Cloud platform owned by the company itself. This technology also makes use of the unlicensed spectrum. Since the network depends on public infrastructure, some locations may not have access to it [27, 30].

Narrowband Internet of Things (NB-IoT) is a radio technology proposed by the 3GPP (3rd Generation Partnership Project). This technology is integrated with the existing Long-Term Evolution (LTE) standard. It was standardized with the Release 13 and named Cat-NB1. The main objective of this technology was to cover an important scenario in the IoT panorama, with the following characteristics: extended coverage, support of a massive number of low throughput devices, high allowed latency, ultra-low device cost, low device power consumption and optimized network architecture [27, 31].

Similarly to NB-IoT, LTE-MTC Machine Type Communication (LTE-M), was also standardized by the 3GPP. It was created with the main objective of reducing the cost and power consumption, simplifying the hardware used and use narrow band operation, while maintaining the original LTE design [27].

## Technologies Comparison

R.S. Sinha et al. [32] stated that, since LoRa uses unlicensed spectrum, it can't compete with NB-IoT in terms of Quality of Service (QoS). This happens because the NB-IoT licensed spectrum allows for a slotted synchronous protocol. Regarding battery lifetime, Sinha et al. [32] state that, since NB-IoT devices require regular synchronization, they spent more power than LoRa based ones. However, possess lower latency and higher data rates. In regard to the network coverage, since NB-IoT devices depend on existing infrastructure, the coverage may be more restricted, whereas LoRa provides greater coverage flexibility.

## Portugal Coverage

Most of these technologies are already implemented in Portuguese territory. The operators Altice [33] and NOS [34] already announced full NB-IoT coverage. Vodafone announced coverage in certain areas around Cascais. Sigfox [35] has also provided full coverage. Currently, LoRaWAN [36] coverage is still growing but there are already 80 gateways installed throughout Portugal, including Madeira and Azores, that provide coverage to most of the territory. At the time of this thesis, no LTE-M coverage has been announced.

Table 2.3: IoT wireless technologies comparison

Technology	Coverage	Range [m]	Data Rate [b/s]	Cost
LoRaWAN	Medium	<15km	0.3-38.4k	Low
SIGFOX	High	<40km	100	Low
NB-IoT	High	<35km	Up to 1000k	Medium
LTE-M	None	-	Up to 100k	Medium

## 2.3.2 IoT Application Layer Protocols

Upon defining a Network Infrastructure and its wireless technology, its important to format the transmitted data according to a protocol. Different kinds of protocol can be applied to different situations, such as gathering the data from RFID tags, sensors etc., protocols for Machine to Machine (M2M) communication, Device to People (D2P) communication, and Device to Device (D2D) communication protocols to send the data to the server.

Currently, there are four widely accepted messaging protocols used in IoT systems: MQTT, CoAP, AMQP and HTTP [37], further detailed below.

## **MQTT**

Message Queuing Telemetry Transport Protocol (MQTT) is a publish/subscribe protocol, used in lightweight M2M communications. In this protocol, a MQTT Client publishes messages to a MQTT Broker. These messages can be subscribed by other clients and retained for future subscriptions. The messages can be separated in different addresses, named topics, allowing clients to subscribe to multiple topics and receive the corresponding messages. This protocol uses the TCP transport protocol with TLS/SSL security mechanisms and provides three levels of Quality of Service (QoS) for personalized reliability in message delivery. MQTT is mostly used in large networks of small devices [37–39].

## **CoAP**

Constrained Application Protocol (CoAP) is another lightweight M2M protocol. It supports two architectures: request/response and resource/observe. In the request/response context, CoAP depends on REpresentational State Transfer (REST), where different methods and endpoints can be used, which allows easy interoperability with HTTP and the RESTful Web through simple proxies. In the resource/observe architecture, the publisher sends data to the server through a Universal Resource Identifier (URI) using an extended GET method. This URI can be compared to the topics used in MQTT. Upon receiving an updated value the server notifies all subscribers with the new value. Unlike MQTT, CoAP uses the UDP transfer protocol and DTLS for security. Since the clients and server communication is connectionless, it may be less reliable. Therefore, CoAP implements its own reliability mechanism using “confirmable messages” and “nonconfirmable messages” [37–39].

## **AMQP**

Advanced Message Queuing Protocol (AMQP) is a messaging protocol that also supports both request/response and publish/subscribe architectures. It was designed for reliability, security, provisioning and interoperability between clients and message middleware servers. AMQP also offers multiple features such as a reliable queuing, topic-based publish/subscribe messaging, flexible routing and transactions. In AMQP communications, an “exchange” is created by publishers or consumers. This “exchange” has a name that is broadcast, through the network, allowing publishers and consumers to find each other. The incoming messages to the consumers are attached to a queue responsible for the exchange. AMQP is built on top of the TCP protocol and uses TLS/SSL and SASL for security. It also provides two levels of QoS in message delivery [37, 39].

## **HTTP**

Hyper Text Transport Protocol (HTTP) is a web messaging protocol that similarly to CoAP supports a request/response RESTful Web architecture, allowing caching. HTTP supports different endpoints and the same methods as CoAP, thus both protocols are easy to integrate. It also uses a Universal Resource Identifier (URI) instead of topics, from where clients and servers exchange messages. HTTP uses the TCP transport protocol and TLS/SSL for security. It is the global accepted web messaging standard

protocol, offering a wide range of features such as persistent connections, request pipelining, chunked transfer encoding, among others [37].

A simple comparison between the protocols is shown in table 2.4.

Table 2.4: IoT Application Layer Protocols comparison

Technology	Architecture	Protocol	Security
MQTT	Publish/Subscribe	TCP	TSL/SSL
CoAP	RESTful Request/Response & Resource/Observer	UDP	DTLS
AMQP	Request/Response & Publish/Subscribe	TCP	TLS/SSL
HTTP	RESTful Request/Response	TCP	TLS/SSL

### 2.3.3 IoT mechanisms applied to the Natural Environment

Throughout recent years, IoT technologies have added significant value to several sectors, through the use of different sensors and communication methods. Industrial plants, transportation, health, urban monitoring, are some of the sectors that have grown due to automated monitoring and data collection. Sectors related to the natural environment have also seen several changes due to the rise in data collection methods. Forestry studies benefited from the increase in biological and environment variables monitoring, like weather, soil temperature, water availability, and others. While in the agriculture sector, monitoring systems that control planting, irrigation, fertiliser application and harvesting are already in use [40].

Several different IoT technologies have been used in monitoring forest environments. X. Feng et al. [41] state that NB-IoT, ZigBee and LoRa wireless communication technologies are the most suitable for Precision Agriculture scenarios. While ZigBee is the most power efficient technology, LoRa and NB-IoT provide the most coverage. Plenty of projects and studies based on these technologies were performed, with some being described below.

A new networking based intelligent platform to monitor forests, built around ZigBee, has been developed and studied [42]. This system used a layered decomposition, where a data collecting subsystem is responsible for managing sensors and their data. Several of these subsystems interact with a central server system through ZigBee nodes. ZigBee based networking technologies have several advantages favorable in isolated locations, such as low power dissipation, low data rate and high-capacity transportation, which make this system reliable in forest environments.

Various technologies have been used in automated agriculture systems. Usually a Wireless Sensor Network (WSN) is built around the crops. Different aspects like temperature, light, humidity, soil moisture, and others, are the most common in these systems. The main differences rely on the communication methods. In some cases, ZigBee or Wi-Fi modules may be used to communicate with a central node or base station, that collects sensor data and controls the overall network [43]. Other technologies, like GSM, have been used in the implementation of greenhouse control systems. In this system, the interaction between the users and the controller is done through Short Message Service (SMS) messages

[44].

G. Valecce et al. [45] carried out several field experiments on the NB-IoT performance in rural areas, in comparison with GPRS. The tests results showed good NB-IoT coverage performance in most environments and especially good results in underground areas where the technology's high penetration provides significant advantage over GPRS. An interesting result was also verified, as the MQTT protocol, based on TCP communication, caused significant retransmissions and delays. The authors claim that CoAP based experiments could provide a means of comparison with other technologies and confirm the obtained results.

## **2.4 Micro-Controller and Sensor Technologies**

### **2.4.1 Micro-Controllers**

One of the most important hardware components in this project is the Micro-Controller Unit (MCU), as it is responsible for reading sensor values, execute actions through actuators, in this case the motor and valves, and controlling the overall behaviour of a device, through the developed code. Micro-controllers are often installed in general-purpose boards which add useful features and functionalities for not only prototyping projects but also sophisticated tools. The project requirements define the need for an affordable and accessible MCU. Some of the most popular micro-controllers, that are suitable for the project, and some of their features are described below. These micro-controllers are highly accessible and provide distinct hardware specifications [46].

#### **ATmega328**

The ATmega328 is an 8-bit AVR micro-controller. It is a MCU used in several Arduino development boards, like the Arduino UNO, Arduino Nano and Arduino Pro Mini, which make it one of the most popular micro-controllers with excellent community support and easiness of programming through the Arduino development platform [46].

Its operating voltage ranges between 1.8 to 5.5 V and features 23 general purpose I/O lines, 32 general purpose registers, 3 timers, internal and external interrupts, digital communication peripherals (1 UART, 2 SPI and 1 I2C ports), power saving modes and others. It also includes a 32 kB Flash memory, a 1 kB EEPROM and a 2 kB SRAM [47].

#### **ATmega32u4**

The ATmega32u4 is a low-power Microchip 8-bit AVR RISC-based micro-controller, with similar characteristics to the ATmega328, also including several GPIO lines, general purpose registers, 4 timers, internal and external interrupts, 1 UART, 2 SPI and 1 I2C ports, and six power saving modes. The ATmega32u4 has the same memory characteristics as the ATmega328 and its operating voltage ranges from 2.7 to 5.5 V. One of the main characteristics of this MCU is that it includes a USB Device peripheral that provides high data transfer rates [48].

This MCU is found in several well-know boards such as the Teensy 2.0, Arduino Pro Micro, Arduino Leonardo and others [46].

## ESP8266

The ESP8266 is another very popular MCU and one of the first micro-controllers with integrated communication capabilities. It includes a L106 32-bit RISC microprocessor core and an Wi-Fi microchip with micro-controller capabilities, 17 GPIO pins, digital communication peripherals, such as SPI, I2C (only software implementation), I2S and 2 UART (1 transmit only) ports, and others [49].

There are over one hundred development boards based on the ESP8266, with some of the more popular being the NodeMCU DevKIT and SparkFun ESP8266 Thing. The ESP8266 is also compatible with the Arduino development platform, also providing a large community support [46].

## ESP32

The ESP32 can be considered an upgrade on the ESP8266, that not only has integrated Wi-Fi capabilities but also a dual-mode Bluetooth. It contains a 32-bit LX6 microprocessor, with two cores, and also provides additional peripheral interfaces comparing to the ESP8266, with 34 GPIO pins, digital communication peripherals, such as 4 SPI, 2 I2S, 2 I2C and 3 UART ports, and power management features, such as deep sleep mode, wake up from GPIO or timer interruptions, among others [50].

Similarly to the ESP8266, a great number of development boards using the ESP32 are available. Some examples are the NodeMCU-32S and ESP32 Thing [46].

Table 2.5 shows some development boards that use the micro-controllers referred above, and their respective price.

Table 2.5: Micro-Controllers price and power consumption

Board	Micro-controller	Power Consumption [mA]		Price [€]
		Active	Sleep	
Arduino UNO	ATmega328	98.4	27.9	20.00
Arduino Nano	ATmega328	22.1	4.8	20.00
Arduino Pro Mini	ATmega328	14.6	3.2	8.42
Teensy 2.0	ATmega32u4	27.3	0.04	18.85
Arduino Pro Micro	ATmega32u4	-	-	15.19
Arduino Leonardo	ATmega32u4	-	-	18
NodeMCU DevKIT	ESP8266	-	-	6.94
SparkFun ESP8266 Thing	ESP8266	60	0.077	14.34
NodeMCU-32S	ESP32	64.6	4.7	14
ESP32 Thing	ESP32	41	4.43	18.57

## 2.4.2 Soil Humidity Sensors

A great part of the available soil humidity sensors can be divided into two categories: resistive or capacitive sensors. Each type measures the soil humidity in its own way and has different accuracy values. From these sensors, low-cost versions have widely emerged in the e-commerce space and have been used in different applications such as automation of agricultural irrigation systems [51].

Resistive humidity sensors use two probes that are inserted into the soil and force an electric current to pass through them and the soil. The soil resistance is then measured, which can be equated to a humidity value through a calibration process.

Given that water is a better conductive material than soil, when the soil water content is elevated the resistance value drops significantly, while when the soil is drier the resistance rises.

Capacitive humidity sensors contain a pair of electrodes that are in contact with the soil, which acts as the dielectric medium. The sensor then measures the charge time of the medium, that acts as a capacitor, which is inversely proportional to the capacitance of the soil.

Higher soil water content provides less charging time and, therefore, higher capacitance values. These values can also be equated to humidity values through calibration processes.

Given the operation of this type of sensors, less corrosive material can be used in their fabrication, comparing to resistive sensors. However, capacitive sensors are susceptible to substances present in the water that may affect its conductivity. These sensors are also more expensive than resistive ones.

S. Adla et al. [52] studied the performance of several low-cost capacitive and resistive soil humidity sensors. In this study two capacitive and two resistive sensors were submitted to several performance tests, in different soil and temperature settings. The capacitive sensors were also submitted to additional performance tests in liquids of known dielectric constants, to measure their accuracy in different conductivity values.

It was observed that the capacitive sensors provided better precision scores than resistive sensors and that, overall, the more expensive sensors are more precise. S. Adla et al. [52] conclude that low-cost capacitive sensors can match the performance of the standard sensors if soil-specific calibration is used.



# Chapter 3

## Design

### 3.1 Requirements

As stated before, the main objective of this research project is to ease and automate the soil water sampling process through a device capable of extracting water samples using a suction lysimeter, and notifying the operator responsible for the extraction that a sample was collected. A second device was also projected. This device is a simpler version of the first one, where the operator is notified when a sample is ready for extraction but the extraction mechanism is omitted. Thus, the operator is also responsible for manually extracting the sample. Both devices functionality should follow the ensuing steps:

- **Assembly and Calibration:** A comprehensive user manual should be provided to the technician where the components list and assembly instructions are succinctly described, allowing the replication of both devices. A third simpler device will assist the technician in finding the specific threshold humidity values and calibrate the other devices.
- **Installation:** Should be easily installed by the technician on the sampling site.
- **Sample detection:** Once it starts raining and the soil moisture level is high enough, the devices should start extracting the sample and/or alert the technician.
- **Alert the user:** When the Full Collector device extracts a sample it should send a message to the technician informing that a sample was extracted.
- **Sample collection:** The technician arrives on the sampling site, collects/extracts the sample and/or resets the device to look for new samples.

These devices are planned to be used at a national scale, by several different people, in several different locations with different types of environment and soil. Given these characteristics, the system must comply to several requirements:

1. **Low cost:** Since the system will be replicated throughout several locations, it is important to build an affordable and easily reproducible system. The final cost should be in the same order of the current operation costs.
2. **Efficient use of power:** In most cases, the systems will be located in isolated places, where access to power networks is not possible, needing to resort to batteries. Thus, it is necessary that the system spends its power in the most efficient way possible, in order to extend its battery lifetime. The battery should require minimum maintenance, only needing to be recharged by the operator once per year, at most. The use of a solar panel could also greatly increase the battery lifetime.
3. **Reachable:** Since the devices can be installed in remote locations, they need to be able to maintain communications even in areas with low network coverage.
4. **Ease of use:** Most users of the systems may not have an engineering background or knowledge about the system structure. Therefore the user interface must provide a friendly method for the user to interact with the system. Specially when collecting the extracted sample from the container.
5. **Adaptable to multiple soils:** The devices will be installed in many different types of soil. So its essential that the person installing the equipment is able to calibrate it in accordance to the site-specific soil profile. Sample volume needs to be at least 50 cc, thus a wrong calibration could lead to unusable samples. As more experiences and analysis are done in the soils, the operator may need to configure the soil humidity threshold on an already installed device. Therefore the devices need to be capable of real-time remote configuration.
6. **Reliable:** The devices need to be able to deal with unexpected errors and conditions and the sensors need to be accurate. This is an important requirement since an erroneous user notification or sensor measurement may lead to samples not being extracted or the operator being alerted in false conditions.

## 3.2 Architecture

It is possible to separate the Full Collector device in four sections: sampling hardware, power supply, operation control and communication. The sampling hardware is responsible for the sample extraction, the power supply powers the device, the operation control includes the hardware and software that define the device behaviour, while the communication allows the device to alert the operator. The Notificator device, while similar to the Full Collector, doesn't include the sampling hardware.

All the devices installed communicate with a server. This server is responsible for connecting the devices to the operator, controlling certain device parameters and storing relevant device data.

Figure 3.1 shows the Hardware Architecture Diagram, where it is possible to discern the different hardware components and their respective sections.

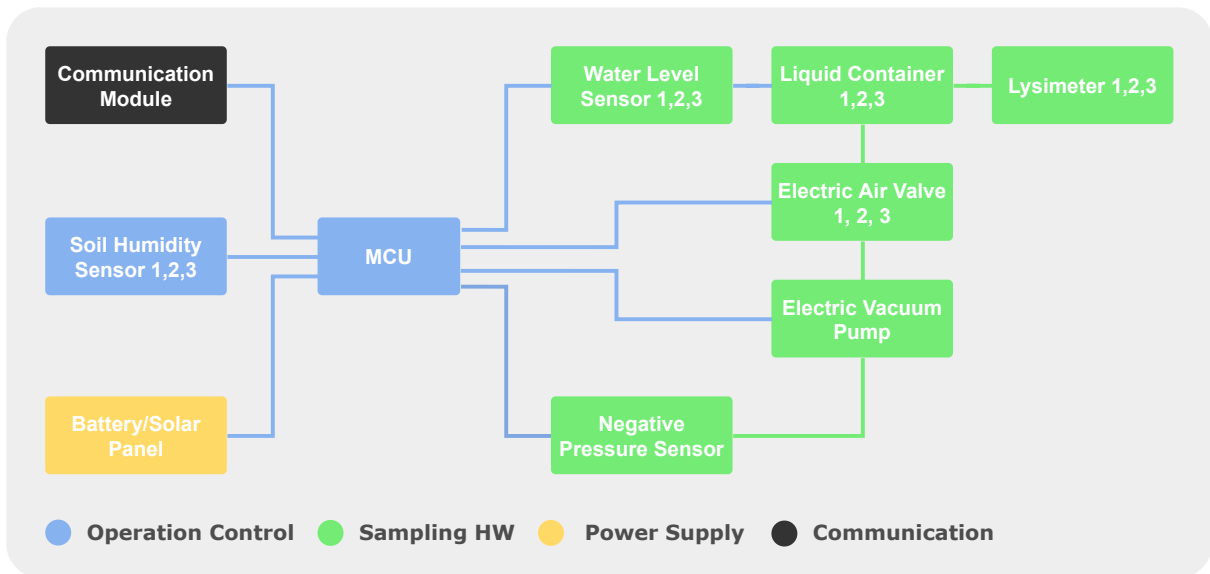


Figure 3.1: Full Collector Hardware Architecture Diagram

### 3.2.1 Sampling Hardware

This section contains the mechanical components used in the extraction of water samples through the lysimeters. This process is shown in figure 3.2, where an Electric Vacuum Pump exerts negative pressure, or vacuum, in the tubing system. A set of normally closed Electric Solenoid Air Valves blocks the negative pressure from being exerted on the corresponding lysimeter and container and allows each lysimeter to be pressurized independently. When the valve is activated and opened, the vacuum is propagated to the corresponding lysimeter and container. After some time, the lysimeter cup's pores will be filled with water that will start flowing through the lysimeter tubes. Once the water starts reaching the container, its flow will be stopped and the sample will be kept in the container.

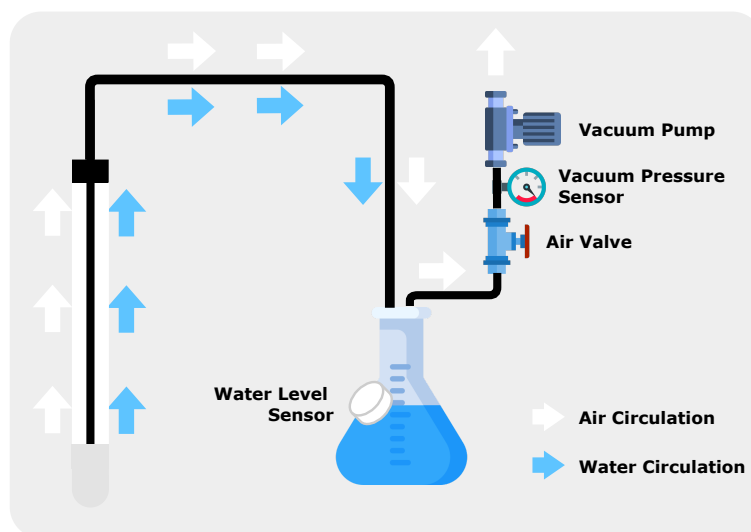


Figure 3.2: Exemplification of the device extraction process using an electric vacuum pump and a suction lysimeter

If the sample starts overfilling the container it may flow to the tubes connected to the valves and the

other electrical components, although the valves should be able to contain the water. To prevent this, a water level sensor should be equipped in the container, that detects when the water sample is sizable enough and stop the extraction process. This behaviour will also optimize the extraction time, saving battery. This sensor should be contact-less, meaning it can be attached on the outside of the container which prevents sample contamination.

Another important concern is the amount of negative pressure created by the pump. If this pressure rises above the lysimeter limit, it could break it. To keep this from happening, a negative pressure sensor should be added to the tubing, allowing the system to turn off the pump before reaching that limit.

### 3.2.2 Power Supply

As stated in the requirement 2, the devices will need to be powered by batteries. These batteries need to be powerful enough to support not only the electronics but also the electric pump and valves, in the Full Collector devices. The capacity of the battery should also allow the device to have a battery lifetime large enough to avoid frequent battery changes.

Solar panels may also be used as a way to prolong the devices battery life, by charging the batteries. Most batteries are susceptible to over-charging when connected to solar panels, which may add the need for charge controllers to be installed. Charge controllers protect batteries from overcharging and overheating, however they are generally recommended for batteries with over 20Ah capacity. On the other hand, solar trickle chargers, if chosen correctly, provide a reliable way of maintaining a battery level without overcharging risk and need for additional circuitry.

As most electric pumps, micro-controllers and sensors have different voltage requirements, the devices need to be able to regulate the battery voltage to different levels. Thus, external circuits, like voltage regulators, need to be added to the devices. Since the electric pumps require larger voltage levels, the batteries used need to be powerful enough to support them, while the voltage regulators drop the battery voltage level to ranges accepted by the electronics.

### 3.2.3 Operation Control

The Operation Control section is responsible for controlling the device behaviour, from the sampling hardware to the device communication.

#### Hardware

The operation of both the Full Collector and Notificator devices requires two main hardware components:

- **MCU:** the Micro-Controller (MCU) is the main hardware component of this section, as all other components are connected to it. The MCU reads the sensors data, signals the valves and pump, in the Full Collector device, and exchanges messages with a Communication Module.

- **Soil Humidity Sensor:** These sensors measure the soil humidity. They should be installed at the required depths and relatively close to the lysimeters, in order to obtain accurate measurements. They should also be calibrated to obtain a relative percentage value.

## Software

The MCU contains the operation control software, which can be divided into two main sub-modules:

- **Humidity Sensors Control:** This module is responsible for measuring the soil humidity sensors values and converting them into relative humidity values. If the relative humidity value is superior to the defined threshold, the sample extraction is triggered. Given the low frequency of precipitation in the sampling zones and the time for the soil to absorb and retain water, the moisture sensor will not need to be continually sampling. A discrete behavior will allow for lower power consumption while maintaining current and accurate measures without compromising the device effectiveness. Therefore, the device should be asleep for most of the time and wake up sporadically to send the sensors data or events.
- **Sample Extraction Control:** This module is responsible for controlling the sampling hardware, when the sample extraction process is activated. Since the MCU is connected to the pump, valves and pressure and water level sensor, it is able to implement the process explained in the section 3.2.1.

When an alert needs to be sent to the operator, such as samples being ready for collection, low battery level, or others, a message should be sent to the server, that in turn alerts the operator. This is achieved through a communication module present in the device.

### 3.2.4 Communication

As stated above, all device communications are made through a communication module. This module exchanges messages between the server and the device through IoT wireless technologies.

From the available IoT wireless technologies, LPWAN technologies provide an extensive array of modules and protocols that can be used in the devices. Most of LPWAN technologies are compliant with the device requirements 2 and 3, since they are used in many similar long range and battery powered devices.

The technology chosen needs to be prominent in mainland Portugal, capable of reaching the defined sampling locations and affordable, in order to comply with requirement 1.

### 3.2.5 Web Application

The Web Application consists in a server responsible for receiving requests from all the devices installed, store the device's and user's data, alert the users through email messages and provide a user interface for the user to register and configure devices. The figure 3.3 illustrates the software and hardware components, and their interactions.

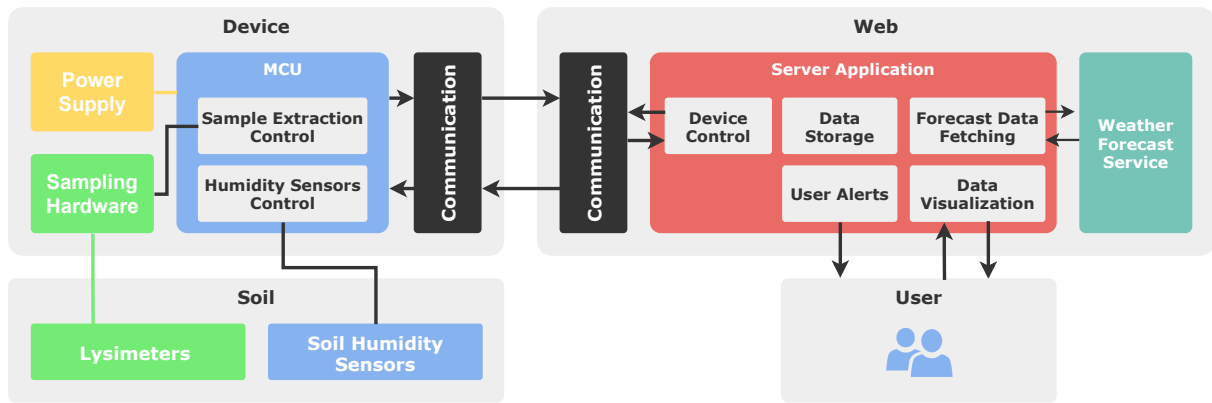


Figure 3.3: Full Collector Device Architecture Diagram

The server can be divided in five sub-modules: the device control, the data storage, data visualization, user alerts and forecast data fetching.

The user/operator interacts with the server, through the data visualization module, where it is possible to register new devices, configure them in real-time and view the data sent by them.

When registering a new device, its coordinates and other useful information should be stored in the server. When a device is installed, it sends data to the server, such as the sensor's values and events like an extraction starting or ending, low battery level and others, which are also stored in the server. All the data is stored in the data storage module.

When the server receives an event it alerts the user device, through the user alerts module, so that the user can collect or extract the sample. The server stores the received data and responds to the devices with information such as configuration data. The configuration data can include new humidity threshold values, changed by the user, that need to be updated in the device, and the device wake up time. Which data is sent to the device is defined in the device control module. This module receives the device messages and according to the message content and the data stored in the server, decides which data to send back to the device.

By having the device wake up during raining periods, the device effectiveness and battery usage will be more efficient. This can be achieved by having the server request meteorology predictions for the device location, through an external service, obtain the nearest raining period and send it to the device as a timestamp in the configuration data. The forecast data fetching is responsible for fetching forecast data in the external service and process it to an adequate timestamp.

## Chapter 4

# System Implementation

In this chapter, an implementation of the Full Collector device is studied. As previously stated, the Notificator device is a simpler version of the former, therefore the data and results gathered from this implementation, are also used in the analysis of the Notificator device.

This implementation makes use of three suction lysimeters installed at the defined depths, 20, 40 and 60 cm.

### 4.1 Hardware

In this section the components of the sampling hardware and sensors used are described.

#### 4.1.1 Sampling Hardware

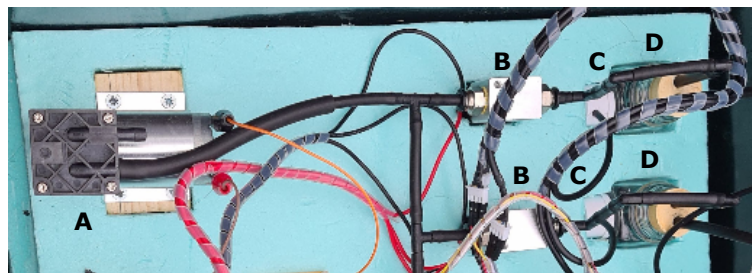


Figure 4.1: Implementation of the sampling hardware

The figure 4.1 shows the sampling hardware section of the implemented device. The legend of the figure is shown in the table 4.1. The components used in this section are described below.

#### Lysimeters

The lysimeters used in this implementation are the Soil Solution Access Tubes (SSAT) from the Irrometer Company. These lysimeters are sized according to their installation depth, meaning they are 20, 40 and 60 cm in size, respectively. A 30cm SSAT is shown in figure 4.2. Therefore, the top part of

Table 4.1: Legend of the components in the figure 4.1

Component	Description
A	Electric Pump
B	Solenoid Air Valve
C	Water Level Sensor
D	Container

the lysimeters is placed directly on the soil level, with the lysimeter extending until the defined depth, as shown in figure 4.3. According to the lysimeters operating manual [53], the recommended vacuum pressure for sampling extraction ranges from 70 to 80 kilo-pascals (kPa).



Figure 4.2: 30cm Soil Solution Access Tube

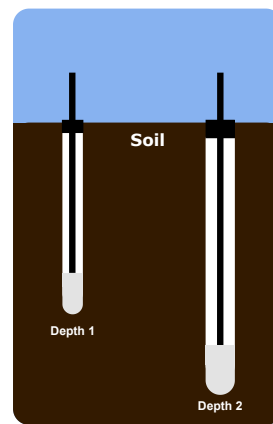


Figure 4.3: Installation of lysimeters at different depths

## Pump

It was used the D2028 Electric Air Pump from Airpon [54]. This pump is powered through a 12 V voltage and exerts negative (vacuum) and positive pressure ranging from -70 to 250 kPa, meaning the maximum vacuum pressure achievable is 70 kPa.

## Solenoid Air Valves

As for the valves, three Normally Close 12V Electric Solenoid Air/Water Valves, from Hoypeyfiy [55], were used.

## Containers

Three 50 ml glass containers were acquired. These containers can store the exact amount of samples required, and are compatible with the contact-less water level sensors used.



## 4.1.2 Sensors

### Soil Humidity Sensors

Three soil humidity sensors were acquired. The sensors chosen were the Capacitive Soil Moisture Sensor v1.2 from DF ROBOT [56]. These are analog sensors that measure by capacitive sensing. They are made of corrosion resistant materials that gives the sensors extended lifetime. An integrated voltage regulator allows the sensors to be powered by voltages in the 3.3 to 5.5VDC range, while the output voltage ranges from 0 to 3VDC and the sensor's power consumption sits around 5mA. As seen in figure 4.4, the sensors are supposed to be installed in a depth smaller then the red warning line. Since the sensors acquired need to be installed in deeper depths, the sensor's electronics were isolated with epoxy, as shown in figure 4.5.

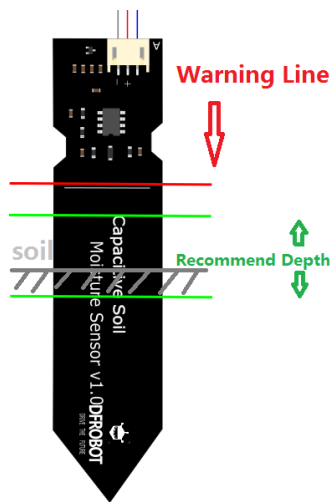


Figure 4.4: Soil Humidity Sensor Recommended Installation Depth [56]

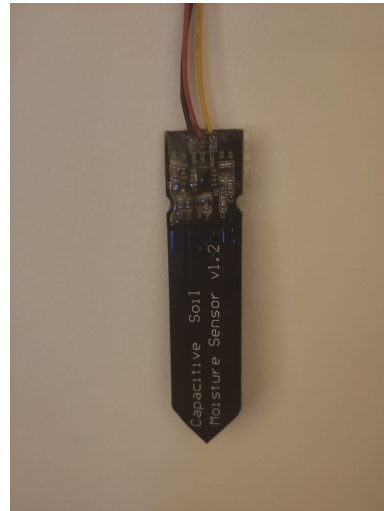


Figure 4.5: Soil Humidity Sensor isolated with epoxy glue

There is an inverse ratio between the sensor output value and soil moisture, therefore the sensors need to be calibrated, in order to obtain a relative soil humidity value. By placing the sensors in the air and water, reading their respective values, that correspond to 0 and 100%, it is possible to interpolate other values between them and obtain their percentage value.

### Vacuum Pressure Sensor

Given the maximum vacuum pressure reachable by the pump, the sensor acquired should be able to hold and measure pressure in that range, while being powered by low enough voltages to be compatible with the micro-controller. The MPX4250DP sensor [57] supports a maximum pressure of 250 kPa and has a supply voltage of 5.1V which complies with the previous requirements, while having a fast response time (1 ms) and a linear relation between the applied pressure and output voltage, which makes it easier to calibrate. This sensor power consumption varies between 7 to 10 mA.

## Water Level Sensor

The DF Robot XKC-Y25-T12V Non-contact Digital Water/Liquid Level Sensor [58] allows non-contact liquid level detection. This sensor has an operation voltage of 5 to 24V and consumes 5 mA. It is a digital sensor, meaning it only sends two signals, low when no liquid is detected and high when liquid is detected. It detects liquids using capacitive sensing, thus it only works on non-metal surfaces such as glass, plastic, ceramic, etc..

### 4.1.3 Communication Module

The communication module used was the BC66-TE-B. The reasons for the choice of this module are explained in the communication section 4.2.

The BC66-TE-B [59] is a NB-IoT module that incorporates the BC66 module as its core. The BC66 [60] is a high-performance multi-band LTE Cat NB1 module that supports several protocol stacks and frequency bands. This BC66-TE-B is a testboard that besides the NB-IoT capabilities, also provides additional features for easier integration with other devices and micro-controllers, while maintaining important characteristics like low power consumption and high durability.

One interesting fact of the BC66 module is that it can be programmed and used as an independent micro-controller. However, its hardware specifications are not compatible with the device requirements, therefore another more capable micro-controller should be used.

The BC66-TE-B provides several interfaces, as shown in figure 4.6 and described in table 4.2.

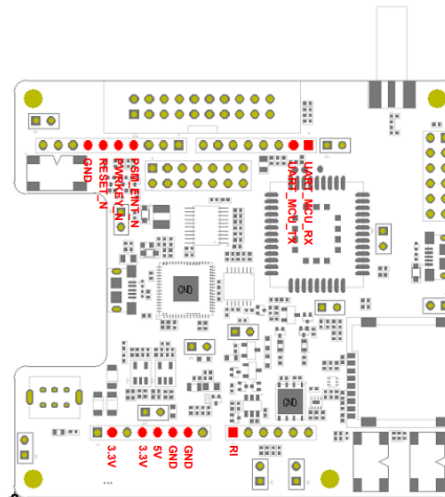


Figure 4.6: BC66-TE-B Arduino Interface Definition [59]

The module also provides three operating modes [60]:

- **Active:** In active mode, all module functionalities are available.
- **Idle:** In idle mode, network connection is maintained but only paging messages can be received. This mode can transition to active or PSM modes.

Table 4.2: BC66-TE-B Interfaces Description [59]

Interface	Description
PWRKEY Button	Used to turn on BC66 module
RESET Button	Used to reset BC66 module
PSM Wakeup Button	Used to wake up BC66 from PSM mode
UART_MCU_RX	Main UART port RX
UART_MCU_TX	Main UART port TX

- **Power Saving Mode (PSM):** In PSM, only the 32kHz RTC is activated. The module can only exit this mode when the RTC times out or by pulling down the PSM\_EINT pin.

#### 4.1.4 Micro-Controller

The micro-controller used in this implementation was the 5V 16 MHz Arduino Pro Mini [61], which is a well documented and easily acquirable micro-controller. The Pro Mini includes the ATmega328p micro-controller and is a more cheap and power efficient alternative to other Arduino micro-controllers. This Arduino provides 6 analog input pins, from A0 to A5, which are enough to support all the analog sensors: the three soil humidity sensors and the pressure sensor. The A4 and A5 pins correspond to the data, SDA, and clock, SCL, lines that support the I2C communication protocol. The 14 digital I/O pins available, 0 to 13, are able to support the other requisites: the three digital water level sensors, the serial communication with the BC66-TE-B module and the electric pump and valves control. The Arduino also provides the SoftwareSerial library, which permits the implementation of serial communication using two digital pins, as the receiver and transmitter. With this library, the UART chip is replaced by software, which allows the Arduino embedded UART, available in the pins 0 and 1, to be used for debugging through the USB port.

An important requirement is that the MCU enters sleep mode and wakes up on a predefined timestamp. This Arduino only has 8 or 16 bit timers which are not precise enough to measure long waiting times. Therefore, a more precise external clock is needed. The DS3231 clock [62] is a real-time clock (RTC) capable of keeping the current date and time information. At the end of the month the date is automatically adjusted according to the number of days in the current month, including corrections for leap years. It also provides two programmable time-of-day alarms and a programmable square-wave output which are able to wake up external devices, as the Arduino, at a determined time through external interruptions. The DS3231 supports I2C communication, its voltage supply can vary from 3.3 to 5VDC and its power consumption ranges from 170 to 300  $\mu$ A, depending if it is in active or stand-by mode, which makes it compatible with the Arduino used, and a very reliable clock.

Another limitation from the Arduino Pro Mini is that it only has two Vcc, 5V, output pins and three ground pins. Since all the sensors require Vcc and GND connections, a circuit that splits the Vcc and GND lines is required. The chosen alternative was the Grove Base Shield V2.0 [63], seen in figure 4.7, from Seeed Studio. This shield provides several grove connections that are attached to some of the

Table 4.3: Arduino Pro Mini Technical Specs [61]

Microcontroller	ATmega328P
Board Power Supply	5 - 12 V
Circuit Operating Voltage	5V
Digital I/O Pins	14
PWM Pins	6
UART	1
SPI	1
I2C	1
Analog Input Pins	6
External Interrupts	2
DC Current per I/O Pin	40mA
Flash Memory	32KB of which 2 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Arduino analog and digital pins. As shown is figure 4.8, each grove connection includes a Vcc line, a GND line, a main line, Pin 0, that is connected to an Arduino Pin and a secondary line, Pin 1, connected to another Arduino pin. The shield provides in total: 4 analog connections, from pin A0 to A3, four I2C connections, attached to the A4 and A5 pins, one UART connection, attached to the Rx and Tx pins, and 7 digital connection, from pin 2 to 9. However, the Grove Base Shield was made to be attached to specific Arduino micro-controllers, such as the Uno, Mega, Due, and others. Consequently, the Pro Mini UNO Shield Adapter Board [64], from OPEN-SMART, was used. This board allows the Arduino Pro mini to be connected with shields that possess the Arduino Uno pin definitions. It also comes with other important features, such as an on-board 5V and 3.3V voltage regulators and DC jack socket, that allows an external battery to be connected to the board, as long as the voltage is between 7 to 15V, and also an on-board 500mA resettable fuse that protects the board from the external power supply's current spikes.

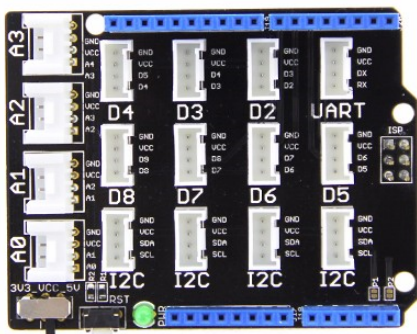


Figure 4.7: Grove Base Shield V2.0 [63]

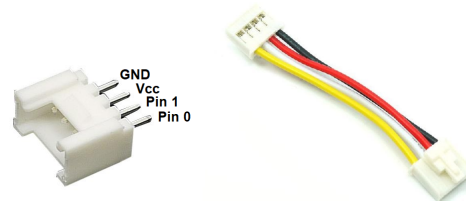


Figure 4.8: Grove Connector [65]

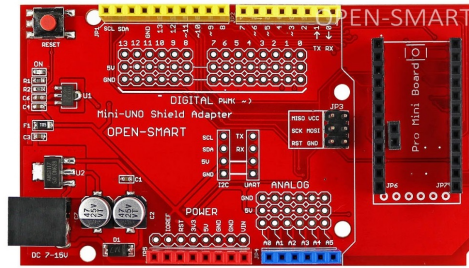


Figure 4.9: Pro Mini UNO Shield Adapter Board, from OPEN-SMART [64]

## Power Supply

In order to power the pump and valves, a 12 V power source is required, while the other electronics only require 5V. Given the power requirements, a 12V battery is enough to run the device for a significant amount of time. A 12V sealed lead acid (SLA) battery provides a good battery lifetime to cost ration and is a widely available option. The SLA battery used the BTX9-BS that has a capacity of 8.4Ah and a max current of 135A.

In order to greatly boost the device lifetime and battery duration, a solar panel could be added connected to the battery. A solar battery trickle charger provides a low cost and efficient solution. Normally, its generated power is low, less than 5W, thus, it can be connected in parallel to the battery without a charger controller, since it is not enough power to overcharge or damage the battery. The charger used generates 2W of power and is able to reach 167mA, at maximum.

As the Arduino doesn't provide enough power to activate the pump or the valves, an external driver circuit is needed. In this circuit, the Arduino is able to switch the pump and valves individually while they are powered directly from the battery. The circuit shown in figure 4.10 is a driver circuit that uses a N-channel MOSFET to switch the pump or valve, while a rectifier diode protects the transistor and the Arduino from reverse voltages. The R1 resistor limits the Arduino output current to the gate, while the R2 is used as a pull-down resistor. This circuit is similar to the Basic Drive Circuit designed by Toshiba in the MOSFET Gate Drive Circuit Application Note [66].

The transistor used was the IRF540N [67] since its Gate Threshold Voltage, 2V to 4V, is lower than the Arduino output voltage, 5V, and, according to its output characteristic, at a 5V  $V_{gs}$  input signal, it is able to pass over 11A, which is enough to power the pump and valves. The diode used was the 1N4006 since it holds currents up to 1A and voltages up to 800V, which is sufficient power for the pump and valves.

### 4.1.5 Final Setup

The final setup configuration of the device is represented in the figure 4.11.

The soil humidity sensors are connected to the pins A0, A1 and A2 of the Arduino, while the pressure sensor is connected to A3. The water level sensors are connected to the digital pins 3, 4 and 9. Since the Base Shield is being used, all the connections include the Vcc and GND lines, necessary to power the sensors. The digital pins 5, 6, 8 and 11 are connected to the driver circuit where they control the

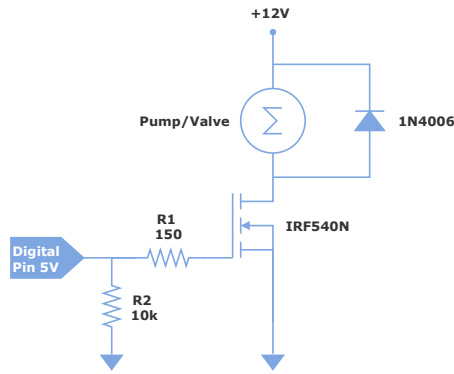


Figure 4.10: Driver circuit for controlling the vacuum pump and valves

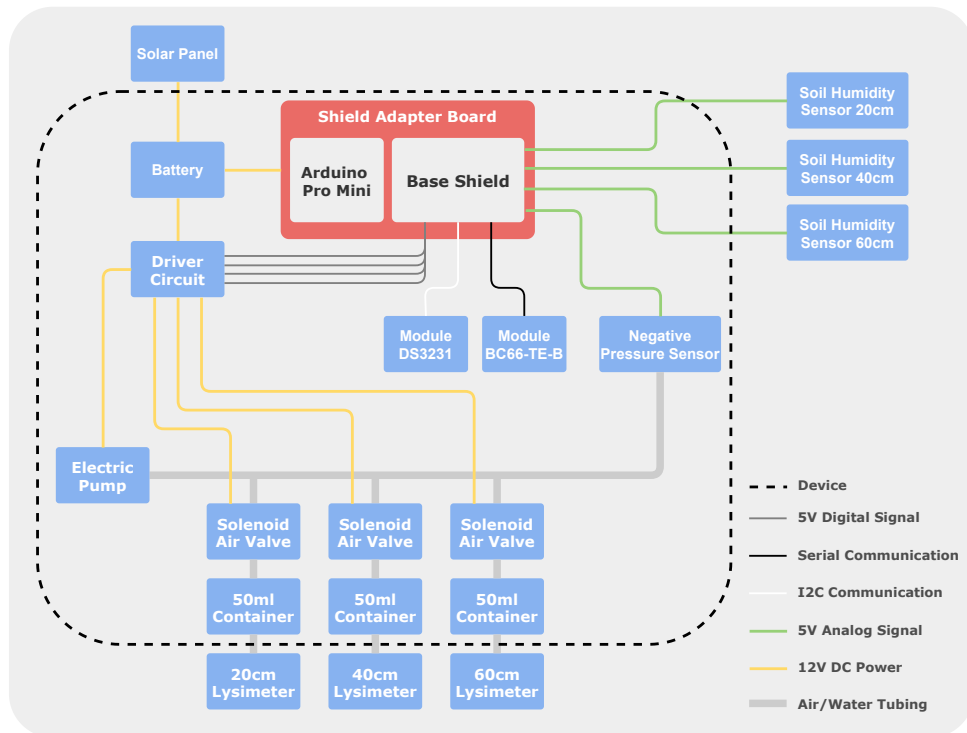


Figure 4.11: Final device implementation diagram

pump and valves, respectively. Lastly, the pins 12 and 13 implement the serial communication, through the SoftwareSerial library, with the BC66-TE-B UART interface. The PWRKEY and PSM\_EINT pins, from the BC66-TE-B module, are connected to the Arduino pins 7 and 10, respectively. An auxiliary Arduino UNO is connected to the Arduino Pro Mini in order to connect it to a USB connection for logging and downloading code. This Arduino is not connected to the device during its operation. The final implemented device is shown in figure 4.12 and the legend of its components in the table 4.4.

## 4.2 Communication

From the available IoT wireless technologies, the LPWAN spectrum provides adequate options for the device communications, given the requirements 2 and 3. Inside this spectrum, NB-IoT is a suitable

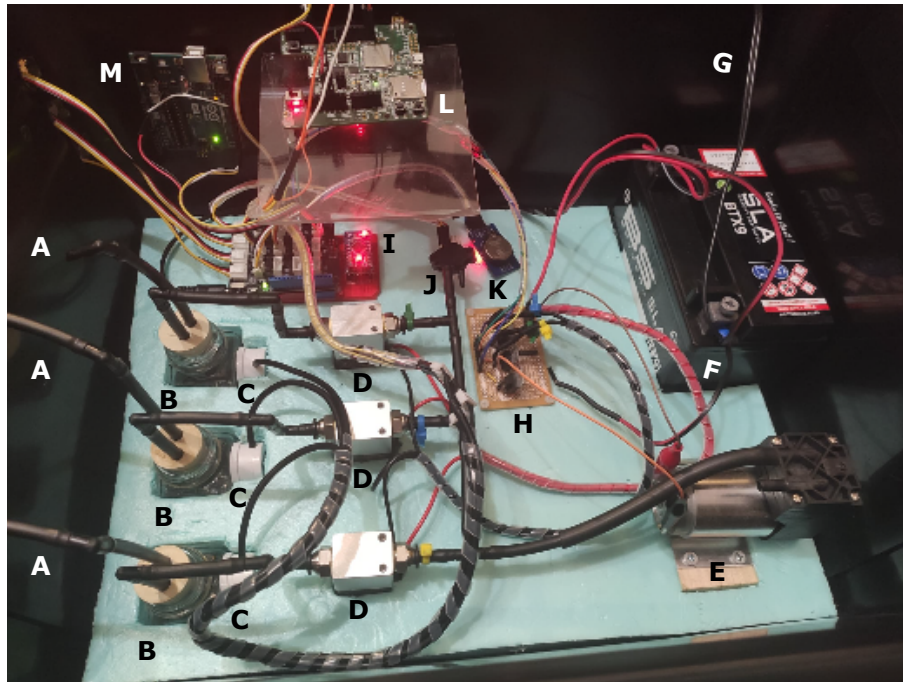


Figure 4.12: Implemented device

Table 4.4: Legend of the components in the figure 4.12

Component	Description
A	Lysimeter Tubing Connection
B	Container
C	Water Level Sensor
D	Solenoid Air Valve
E	Electric Vacuum Pump
F	Battery
G	Solar Panel Connection
H	Driver Circuit
I	Arduino + Base Shield
J	Vacuum Pressure Sensor
K	DS3231 RTC
L	BC66-TE-B Module
M	Auxiliary Arduino UNO

option, has full coverage in Portuguese territory and provides an array of several hardware modules compatible with the technology.

Taking this into consideration, a collaboration was established with Altice and a test SIM card was provided for this project. This SIM card allows the device to be connected to the Altice NB-IoT network. Through the Altice IoT Connect platform it is possible to view the SIM card communication records and other data. The current plan provides a plafond of 28501 Mb that can be spent monthly on data communications. One of the compatible modules with the Altice NB-IoT SIM cards and network is the



BC66 from Quectel.

## 4.2.1 Communication Module

As stated before, the BC66 module can be used and programmed as a micro-controller. This is made possible through the QuecOpen platform [68]. QuecOpen is an embedded development solution for M2M applications, where Quectel modules, including the BC66, can be designed as the main processor, without any external MCU. QuecOpen greatly simplifies application development and it has been widely used in M2M field, such as smart home, smart city, tracker and tracing, automotive, energy, etc. The QuecOpen platform was built over the FreeRTOS operating system which provides characteristics of micro-kernel, real-time, multi-tasking, etc.

As seen through the QuecOpen Software Architecture, in figure 4.13, several user API's are available, which allow the access to hardware resources, radio communications resources, or external devices. The RIL API simplifies the use of AT commands [69], which are important in the communication's and device's configuration and usage. The NVDM API provides functions to read and write to the module RAM, while the HW API provides functions that interact with hardware interfaces like the UART and other GPIO ports. The System API contains additional API's like Power Management and Time API's, etc..

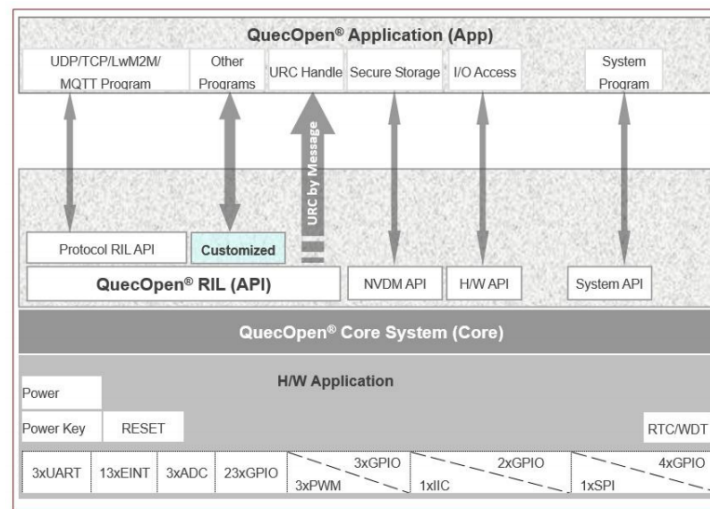


Figure 4.13: The Fundamental Principle of QuecOpen Software Architecture [68]

In this implementation, the BC66-TE-B module was programmed with the Arduino Quectel BC66 development platform. This platform was developed by Georgi Angelov, and provides a higher-level implementation of the Arduino functions based on the QuecOpen API's, which greatly simplifies the code and integration of other libraries.

## IoT Protocol

Given the device requirements and architecture, and considering the NB-IoT protocols available, the communication protocol chosen was CoAP. The CoAP request/response architecture is fitted to the



device architecture, with the BC66-TE-B module being a CoAP client that communicates with a CoAP server, besides being easy to integrate with HTTP servers, as stated in section 2.3.2.

The CoAP protocol was implemented through the CoAP Simple Library, developed by Hirotaka Nisato. This library provides a lightweight CoAP client and server implementations to Arduino devices, thus is also compatible with the Arduino Quectel BC66 platform.

However, one of the main issues of this implementation is that neither the BC66 module nor the CoAP library supports DTLS security over the protocol, at the time of this thesis. According to a Quectel Forum post, CoAP DTLS support may be added in future versions. In addition, the library only supports the PUT and GET methods.

## Architecture

The BC66-TE-B module was programmed according to the block diagram 4.14. The program starts by registering the SIM card. If the SIM card is registered successfully, the program will wait for the module to be registered in the NB-IoT network. When the module is registered in the network, the CoAP protocol is initialized and the module sends a "ready" message to the Arduino, confirming that it is ready to send messages. Then, it will continuously wait for a message to be received in the UART port. If the message received is equal to "sleep" the device will enter the sleep mode procedure, any other message is sent to the CoAP server. Before sending a message, the module checks if it is registered in the network and waits for registration if it is necessary. After sending a message, the module will continuously wait for the server's response. If the response doesn't arrive after a certain amount of time, which can occur due to network problems or other factors, the module will resend the message. The message will continuously be resent until a response is detected or a maximum number of tries is reached.

### 4.2.2 Communication Protocols

#### Message Format

All messages sent between the devices and the server follow the same format. This format was defined with the idea that it is more efficient to send a message with all the necessary data than several messages with parts of said data.

All messages sent from the device use the CoAP PUT method and are directed to the same server endpoint. This format uses JSON as the base format and all messages are required to contain the following parameters:

- **id:** Identifier of the device, which allows the server to associate a message with their device.
- **msg:** Since a message can be sent multiple times, in cases where the server response doesn't reach the device, by having an identifier of the own message the server can identify repeated messages and not store the same data or alert the users multiple times.
- **h20:** Current soil humidity level at 20cm depth, rounded to one decimal place.

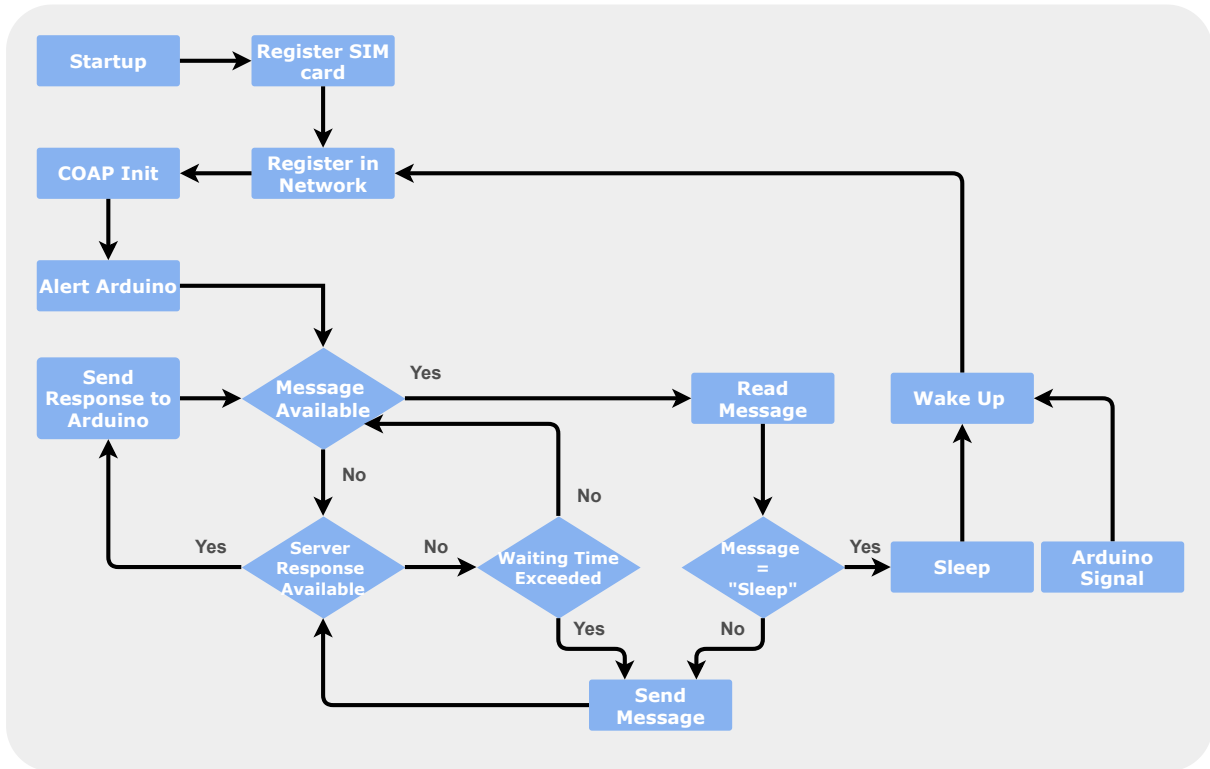


Figure 4.14: BC66-TE-B Software Architecture

- **h40**: Current soil humidity level at 40cm depth, rounded to one decimal place.
- **h60**: Current soil humidity level at 60cm depth, rounded to one decimal place.
- **p**: Current pressure level on the device, rounded to one decimal place.

Therefore a simple device message would look like:

```
{"id":1,"msg":1,"h20":10.0,"h40":10.0,"h60":10.0,"p":1.0}.
```

When an event occurs and the server needs to be alerted, the message sent also contains the sensors data besides the event flag represented as one of the parameters present in table 4.5.

The server's responses also follows the JSON format. In the response a status parameter, represented as "st", is always present which indicates the type of the response. So a server response looks like:

```
{"st":"ok","t":1626110377}.
```

## Messaging and Configuration Protocols

The device messages can vary depending on the state of the device. The server responses are also dependent on the request that it receives. The table 4.6 shows all device requests and their corresponding responses.

As stated before and described in the table 4.6, device configuration changes are contained in the request/response messages between device and server. When the user changes a device parameter,

Table 4.5: Events and their corresponding message parameters

Parameter	Value	Event
ini	ok	Program initialized
l20	start	Extraction started on the 20cm lysimeter
	end	Extraction ended on the 20cm lysimeter
	error	Error extracting on the 20cm lysimeter
l40	start	Extraction started on the 40cm lysimeter
	end	Extraction ended on the 40cm lysimeter
	error	Error extracting on the 40cm lysimeter
l60	start	Extraction started on the 60cm lysimeter
	end	Extraction ended on the 60cm lysimeter
	error	Error extracting on the 60cm lysimeter
b	low	Low battery detected

Table 4.6: Device requests and corresponding server responses

Request	Response
<code>{"id":1,"msg":1,"h20":10.0,"h40":10.0,"h60":10.0,"p":1.0}</code>	<code>{"st":"ok","t":1626110377}</code> <code>{"st":"c20","l":30}</code>
<code>{"c20":"ok"}</code>	<code>{"st":"ok","t":1626110377}</code> <code>{"st":"c20","l":30}</code>
<code>{"id":1,"msg":1,"h20":10.0,"h40":10.0,"h60":10.0,"p":1.0,"l20":"start"}</code>	<code>{"st":"ok"}</code>
<code>{"id":1,"msg":1,"h20":10.0,"h40":10.0,"h60":10.0,"p":1.0,"l20":"end"}</code>	<code>{"st":"pck"}</code>
<code>{"id":1,"msg":1,"h20":10.0,"h40":10.0,"h60":10.0,"p":1.0,"l20":"error"}</code>	<code>{"st":"pck"}</code>
Message with a msg_id already used	<code>{"st":"ack"}</code>
Wrong or Invalid Message	<code>{"st":"err"}</code>

namely a humidity threshold value, the server stores that the value was changed and the said value. When the next device request happens, and if this request does not contain any event, the server will respond with the new value. When the device receives the value, it should store it in memory to keep it permanently. To confirm that the device received the new value, it needs to send a confirmation message. When the server receives the confirmation message, it stores that the value is updated and stops sending said value to the device. If the confirmation message doesn't reach the server, the server will keep sending the new value until it receives the message. In this implementation, it was only needed to update the humidity threshold, but this mechanism can be expanded to other parameters. This process is shown in the diagram 4.15.

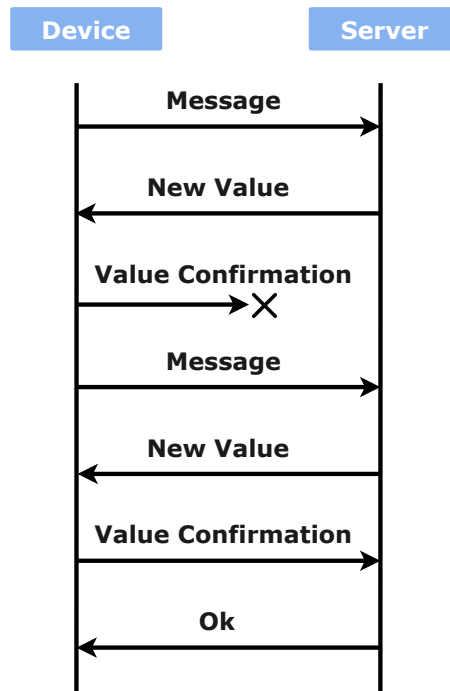


Figure 4.15: Configuration Procedure Diagram

## 4.3 Software

### 4.3.1 Arduino Pro Mini

#### Architecture

The Arduino is responsible to implement the majority of the device's software. The most important software component of this section is a state machine that keeps track of the device's state. Each lysimeter/depth has its own state, which can be one of possible four values: Normal, Collecting, Collected and Error. The Normal state corresponds to when the soil is not humid enough to extract a sample and the device is still waiting for the soil humidity to become ideal for extraction. When the humidity level at a lysimeter's depth surpasses the threshold value, the state of that lysimeter changes from Normal to Collecting and the sample extraction mechanism is triggered. Once the sample reaches a certain volume in size and the water level sensor is triggered, the state changes from Collecting to Collected. During the Collecting state, the device will pressurize the corresponding lysimeter periodically, in order to keep the lysimeter pressurized throughout the entire process. This step is important since the lysimeters may lose pressure if the soil is not humid enough, as stated in section 2.2.1. If, during the pressurization, the pump is not able to reach the pressure threshold, it is possible that the air is leaking or there is a problem with the pump or valves. In this case, the state will change to Error. If a certain time has passed since the extraction begun and no sample was collected the device returns to Normal mode. However, if the humidity value is still above the threshold but the device couldn't extract the sample in the given time frame, probably there is a problem with the mechanical components or a problem with the sensors readings. In this case the state changes to the Error value.

When a state is changed for a certain lysimeter, the device sends a message to the server with a flag, that identifies which lysimeter changed and its state value, as well as, the sensors data. When the device messages the server, the Arduino sends the message in question to the BC66-TB-E module, which sends it to the server, and awaits for the module to send back the server's response. When a lysimeter changes to the Collected state and alerts the server of this event, if the server responds successfully, the device receives confirmation that the user will come pickup the sample.

When the Arduino sends a message to the BC66-TE-B module, it waits for the server's response for a limited amount of time. If the Arduino is not able to receive the response, it will try to resend the message. If, after a certain amount of tries, the Arduino wasn't able to receive the response, the program continues its execution. Since the BC66-TE-B module may keep trying to send the message, it is possible for the Arduino to receive a response of a previous message or multiple messages, which can be processed sequentially.

If all the lysimeters are in the collected state and the device received confirmation that the user was alerted that a sample is ready, it will enter the sleep mode indefinitely. If the device hasn't received the confirmation, it will periodically message the server until it receives the confirmation and, consequently, enter sleep mode indefinitely. When the user comes to collect the sample(s), the device is reset, all the states return to normal and the programs execution starts over. The operator can easily reset the device through the reset button present in the Shield Adapter Board.

When all lysimeters are in the Normal state, the device is not expecting any event besides a sample being ready for collection. In this situation, the device enters sleep mode, in order to save power. This is achieved by sending a message to the server, to which it responds with a timestamp corresponding to the wake up time. Besides the timestamp, this server response can contain the configuration changes to the device, like new humidity threshold values, if the device configuration was changed in the server. This data is stored in the Arduino EEPROM, in order to be kept after a reset or shutdown, where each configuration parameter has a defined EEPROM address. If the device is not able to receive the server's timestamp it will sleep for an hour and try again.

After waking up and during the extraction process, the device will check the battery status. If a low battery level is detected, the device messages the server and enters sleep mode for a long time period, in order to save power and try to raise the battery level. The user will also be alerted of this event.

This entire process is described in the diagram 4.16.

## **Battery Level**

Since the Arduino Pro Mini digital operating voltage is 5V, it cannot measure the battery level directly, which is around 12V. So to measure the battery voltage, a voltage divider or other external circuit is needed, which add extra energy consumption. An alternative method was found using a 1.1 V reference voltage inside the ATmega328P micro-processor. It is possible to compare the Arduino Vcc voltage with the reference voltage and obtain an exact value of the Vcc. It was verified that when the battery voltage drops significantly, the 5 V voltage regulator present in the board can drop the Vcc level to around 4.8 V. So, in order to detect low battery levels, the Arduino checks if the Vcc voltage is close to 4.8 V.

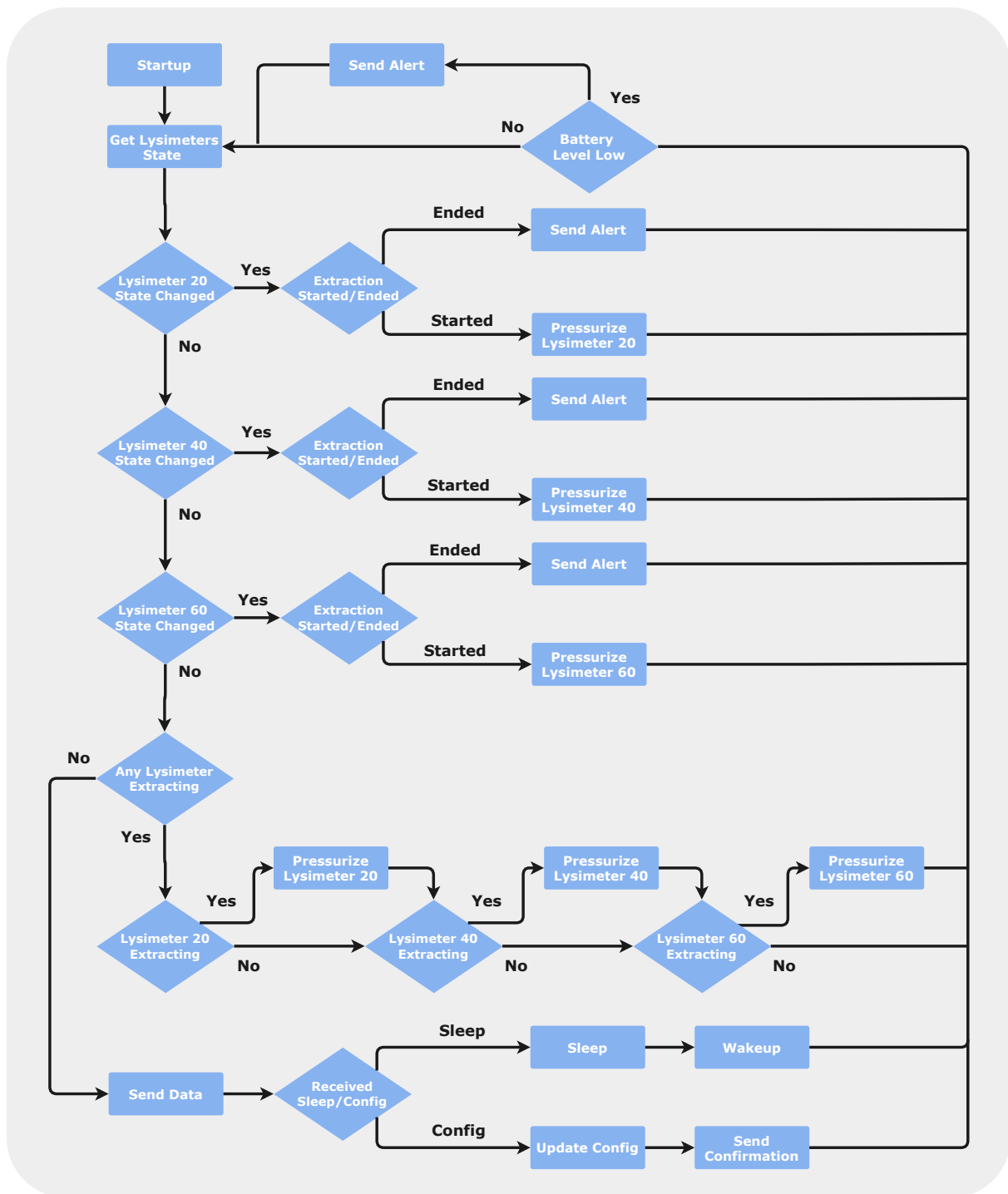


Figure 4.16: Arduino Pro Mini Software Architecture

## 4.3.2 Procedures

### Startup Procedure

During the Arduino program startup, the serial communication with the BC66-TE-B, all the digital pins and the DS3231 RTC are initialized. The DS3231 RTC is initialized and its alarms are reset, so that the Arduino wake up interruption is not triggered.

Then, it will start up the BC66-TE-B module, by signaling the PWRKEY pin. When the module is

registered in the network and ready to send messages, the Arduino receives a "ready" message and proceeds its program.

The last step of the startup routine is to send a initialization message to the server. This message allows the device and the user to confirm that the server is online and functioning. It will also allow the server to know that the device is starting up.

### **Sleep Mode Procedure**

Before the device enters sleep mode, it is necessary to perform some operations. Firstly, the Arduino will send a "sleep" message to the BC66-TE-B module.

When the module receives a "sleep" message from the Arduino, it enables the PSM and responds with an "ok" message to the Arduino, to confirm that it is entering PSM.

When the Arduino receives the sleep confirmation message from the module, it sets the DS3231 alarm to the wake up time, attaches the wake up interruption routine to the pin connected to the alarm and enters deep sleep mode.

The Arduino deep sleep mode is implemented through the `avr/sleep` library. This library implements several sleep modes where different Arduino functionalities are turned off.

In the `SLEEP_MODE_PWR_DOWN` mode, most Arduino functionalities are deactivated and the only way to wake it is with either a watchdog timer interrupt, a level interrupt on pin 2, or a Pin Change interrupt.

Once the Arduino wake up interruption is fired, the deep sleep mode is deactivated and the wake up interruption is detached from the alarm pin. The Arduino will then signal the BC66-TE-B PSM\_EINT pin to wake it up from PSM mode.

When the module wakes up, the wake up routine is executed. In this routine the device registers itself in the network, if it was not yet registered, and send a "ready" message to the Arduino, indicating that it is ready to message the server.

When the Arduino receives the "ready" message, both the BC66-TE-B module and the Arduino itself are confirmed to be active, so the program execution continues.

The diagram depicted in figure4.17 illustrates this process.

## **4.3.3 Server**

### **Architecture**

The server is divided into four applications: the CoAP server, the HTTP server, the database and the OpenWeatherMap service, as shown in figure 4.18.

The CoAP server functions as a proxy between the devices and the HTTP server, where all the requests received from the devices are forwarded to the HTTP server and the responses from the server are forwarded back to the respective devices.

The HTTP server exposes a Website where the users can access and manage the devices, read their data history, manage other users, etc.. It also exposes a API where the CoAP server sends the

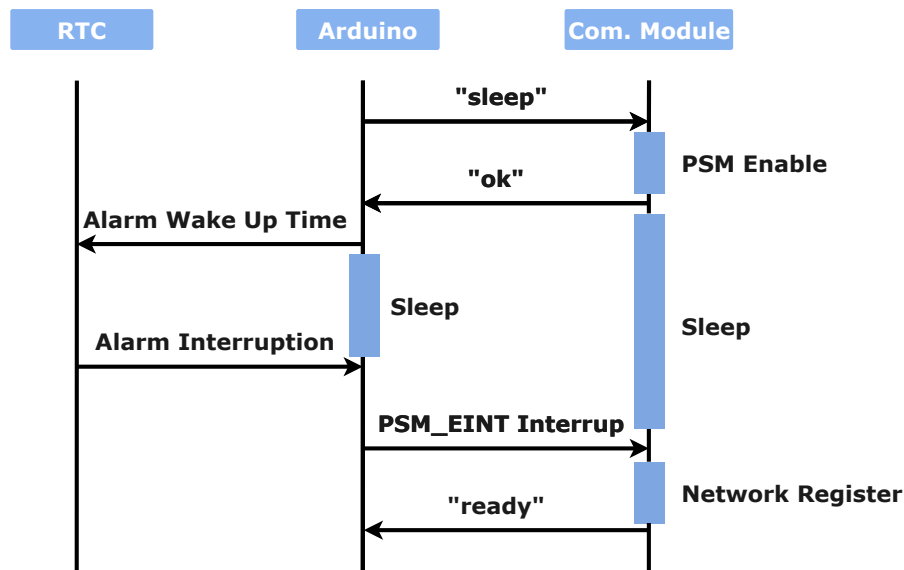


Figure 4.17: Sleep Mode Procedure Diagram

valid device requests to and where it is possible to access all the devices data.

The HTTP server uses a database application where it stores all the device and user data. This database is exclusive to the HTTP server as the CoAP server only functions as a proxy with no storing needs.

Lastly, the OpenWeatherMap is an online service that provides global meteorology data, including current weather data, forecasts, nowcasts and historical weather data for any geographical location. This data can be accessed through the OpenWeatherMap One Call API [70], an HTTP API, and will be used by the HTTP server to obtain the next raining time in each device location.

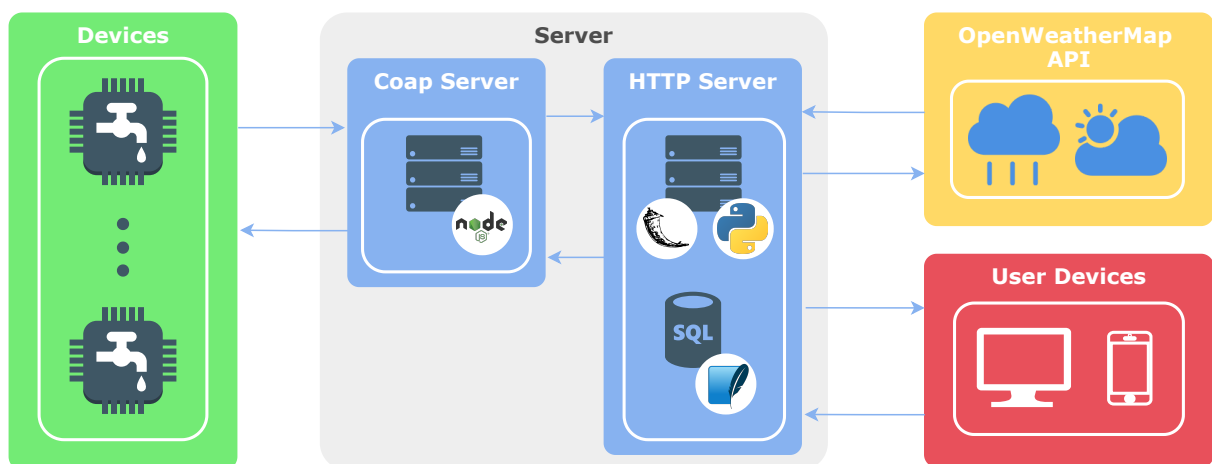


Figure 4.18: Server Software Architecture Diagram

## CoAP Server

This server was implemented in Node.js using the node-coap package. This package provides a CoAP client and server library. The server implements an API where two main objects are available:



request and response. In the request it is possible to access its method, endpoint, payload and other data. While in the response object it is possible to customize its payload and payload format.

When the server receives a request it checks if the method and endpoint are correct and if the payload, where the message is stored, is in JSON format. If one of these conditions is not verified, the error message is sent to the device. Otherwise, the request payload is sent to the server through the HTTP API and its response is sent back to the device.

## Database

The database was implemented in SQLite. SQLite is a relational database management system that contains relational databases that follow the SQL syntax. SQLite databases are integrated directly in applications, in this case, the HTTP server. In this implementation there are five database tables, as shown in the diagram represented in figure 4.19:

- **Users:** Table that represents the users registered in the server. Besides the user basic information the email is also stored so that the alerts are sent to the email.
- **Locations:** The location table stores the devices coordinates. A location can be related with several devices, in case several devices are installed in the same location.
- **Soils:** It is possible to register different soil types in the server, where each soil can have different humidity threshold values.
- **Device:** Each device is related with a location and three soil types, for each depth, since the soil composition may vary with depth, even in the same location. This table also stores the device status and whether a humidity threshold value needs to be updated. The type parameter allows the server to distinct Full Collector Devices from Notificator devices.
- **Readings:** A reading is created when a device sends a message to the server. Each reading stores the soil humidity values, the pressure value, if a sample was collected and if an extraction is occurring. Every reading is associated with the respective device.

## HTTP Server

The HTTP server was implemented in Python, using the Flask web framework. Flask is a micro web framework, meaning it doesn't require any extra libraries or utilities. However, it supports a great deal of extensions that expand the framework base functionalities. In this implementation, several extensions were used, the main ones being briefly described below:

- **Flask SQLAlchemy:** This extension adds SQLAlchemy support to the server app. SQLAlchemy is an object-relational mapper (ORM), which allows the database to be access through Python objects. This method provides an abstraction layer above the database which simplifies operations such as Read, Write and Delete.

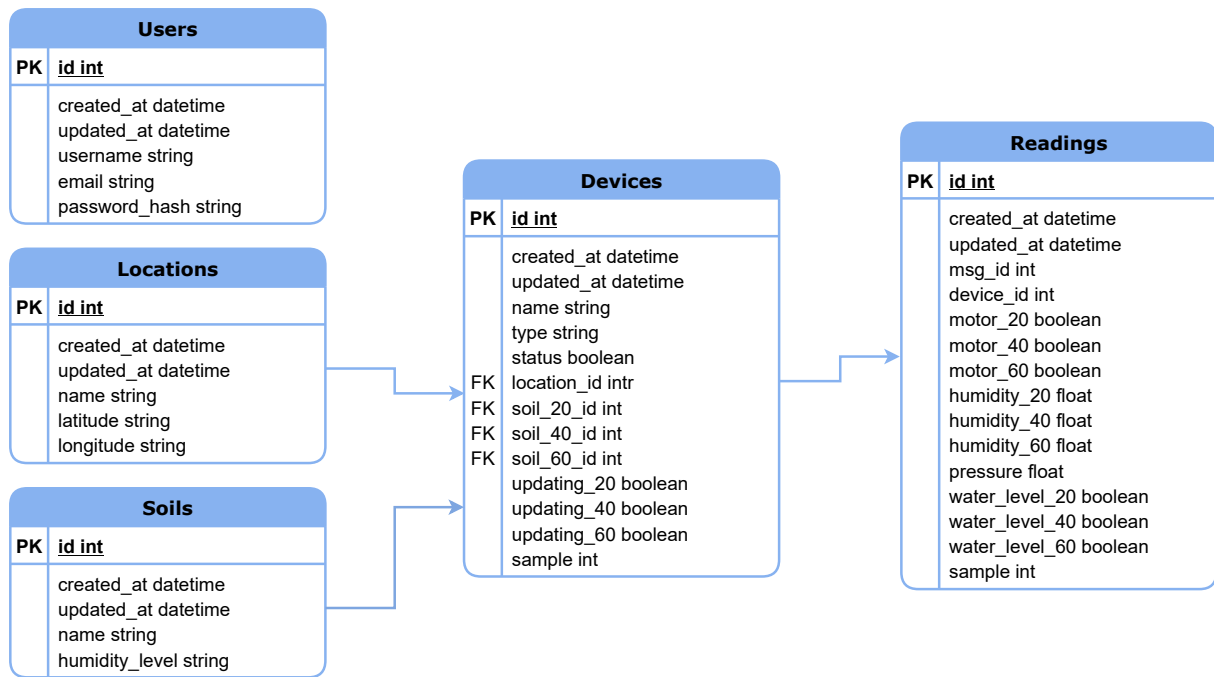


Figure 4.19: Database Entity Relationship Diagram

- **Flask Login:** This extension greatly simplifies the users login logic. It keeps track and manages all the users' login sessions. It provides important features such as securing the user's session and can also restrict certain web pages to certain users.
- **Flask Mail:** The Flask Mail extension provides a simple interface between the server app and SMTP servers. This simplifies the sending of email messages, which are used to alert the users.

The HTTP Server is separated in several API's: the Webpage API, the Info API, the CoAP API and the Device Check API.

The Webpage API is used to return the Webpages through where the users interact. The Webpage API endpoints can only be accessed if the user is authenticated. In this API, users can create and delete new locations, soil types and devices, and can also register new users in the server. This API also contains an endpoint that provides a CSV file with a device's data for easier data access.

The Info API is also only available to authenticated users and provides the device's and location's data in JSON format instead of Webpages. This API may simplify the Server integration with other applications and projects.

The CoAP API contains the endpoint that is used by the CoAP Server and implements the device server-side logic. This endpoint is protected by an API key that is present in the request's headers. This API key is only known by the HTTP and CoAP server, so that no other applications are able to access the endpoint and create false data.

The Device Check API includes an endpoints that checks all active devices and verifies that they are still functioning. This verification is done by checking if a device's last received message date is over the maximum sleep time, that is defined in the section 4.3.3. If a device is verified to be inactive, the device's database status is updated and an email message is sent to all users informing that a device that was

active has stopped working. This endpoint is repeatedly called, with a frequency equal to the maximum device sleep time, through a cron job.

All the API's endpoints are describes in the table 4.7.

### **OpenWeatherMap One Call API**

This API [70] provides the weather forecast for any geographical coordinates. It presents three modes that show forecast data with different frequency and duration:

- **Minute Forecast** that shows data every minute for an hour;
- **Hourly Forecast** that shows hourly data for 48 hours;
- **Daily Forecast** that shows daily data for 7 days.

Given the scope of the project, the hourly forecast presents the best solution. Besides having a finer data granularity, the 48 hours range are also enough for the device needs.

The API response for the hourly forecast contains an array of 48 elements with different parameters, as shown below.

```
[
  {
    "dt":1626102000,
    "temp":302.26,
    "feels_like":304.37,
    "pressure":1013,
    "humidity":60,
    "dew_point":293.7,
    "uvi":4.24,
    "clouds":78,
    "visibility":10000,
    "wind_speed":1.65,
    "wind_deg":182,
    "wind_gust":2.04,
    "weather":[
      {
        "id":500,
        "main":"Rain",
        "description":"light rain",
        "icon":"10d"
      }
    ],
    "pop":0.55,
    "rain":{
      "1h":0.17
    }
  },
  ...
]
```

These parameters vary from temperature values to wind speeds. In the server context, only the "dt" value and "rain" are taken into account. The "dt" value corresponds to the UNIX timestamp of the forecast date. When it is predicted to rain the "rain" parameter includes a rain volume prediction, "1h", for respective hour.

When the server needs to calculate a device wake up timestamp for a certain device, it requests the One Call API with the device's coordinates. The server then parses the response array looking for a forecast with a rain prediction and returns the respective timestamp value, "dt", to the device. If the forecast doesn't include a rain prediction during the 48 hours, the server returns the timestamp of the 48<sup>th</sup> hour.

### 4.3.4 Deployment

The server applications used in this implementation, including the HTTP server, CoAP server and database, were deployed in the same server and the HTTP server Website can be accessed through the URL `http://146.193.41.162:9001/`.

To deploy a new device, firstly it is necessary to register a new device in the Website, which will generate an ID. If the new device locations and soil types are not registered, they need to be registered before registering the device. The generated ID needs to be inserted in the Arduino code and then can be uploaded to the Arduino. To upload the code to both the Arduino and BC66-TE-B, the Arduino IDE tool can be used, as long as the WizIO package is installed. This package contains the Arduino Quectel BC66 platform libraries and boards used to program the BC66-TE-B module.

Table 4.7: HTTP Server endpoints and their description

API	Endpoint	Method	Description
Webpage API	/home	GET	Returns the homepage
	/login	GET/POST	Handles the login authentication
	/logout	GET	Handles the user logout
	/users	GET	Lists all the users
	/user	GET/POST	Creates a new user
	/user/<used_id>/delete	POST	Deletes a user
	/locations	GET	Lists all the locations
	/location	GET/POST	Creates a new location
	/location/<location_id>/delete	POST	Deletes a location
	/location/<location_id>/devices	GET	Lists all the devices of a location
	soils	GET	Lists all the soil types
	soil	GET/POST	Creates a new soil type
	/soil/<soil_id>/delete	POST	Deletes a soil type
	/device	GET/POST	Create a new device
	/device/<device_id>/delete	POST	Deletes a device
	/device/<device_id>/edit	GET/POST	Edits a device
	/device/<device_id>/csv	GET	Returns a CSV file with all the device data
Info API	/api/locations	GET	Lists all location's data
	/api/location/<int:location_id>	GET	Lists a location data
	/api/device/<int:device_id>	GET	Lists a device data
CoAP API	/api/reading	POST	Creates a new reading and Implements the Messaging Protocol 4.2.2
Device Check API	/device-check	POST	Checks if the active devices are functioning

## 4.4 Testing Device

With more devices being installed in new locations, it is expected that these locations present other soil types beyond those studied previously. Therefore, new studies will need to be performed in those soil types, to obtain their threshold values, where the person responsible for them may not have technical knowledge of the device. Given this, the technician needs to have access to a simpler device that is capable of assisting him performing the study. This device can also help the technician in calibrating the soil humidity sensors.

The device consists of an Arduino Uno connected to a soil humidity sensor and a 16x2 display LCD. The Arduino is programmed to periodically read the humidity sensor output voltage and humidity value, and display those values in the LCD. This will allow the technician to perform the soil humidity threshold study and register the humidity values manually without the need for extra components. A manual vacuum pump and a lysimeter may be used to manually extract samples. Although this device is not as automated as the one used in the laboratory experiment 5.2.1, it is much simpler and easier to replicate by technicians without previous device knowledge.

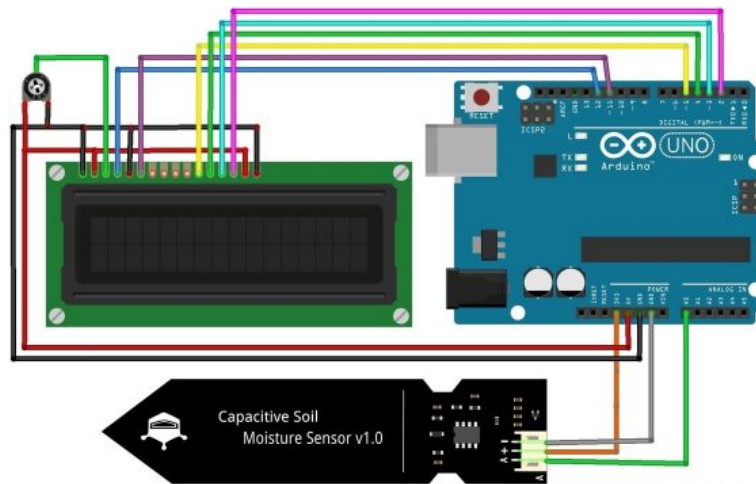


Figure 4.20: Testing device circuit diagram [71]

The figure 4.20 shows the device circuit diagram, where the LCD and sensor are directly connected to the Arduino, while a 100k potentiometer is used to adjust the LCD luminosity.

## Chapter 5

# System Evaluation

After assembling the Full Collector device according to the chapter 4, the device was submitted to several tests in order to evaluate the operation of each component, calibrate several device parameters and, ultimately, install the device in a monitoring site. Each test is detailed in the following sections.

### 5.1 Device Calibration

As stated in the section 4.1.2, the analog sensors need to be properly calibrated before installing a device. Each sensor can behave differently, thus every sensor should pass through the following calibration processes.

#### 5.1.1 Pressure Sensor

According to the MPX4250DP datasheet [57], and as stated in section 4.1.2, the relation between the negative pressure applied on the sensor and the output voltage is linear. Therefore, a manual vacuum pump with a barometer was attached to the pressure sensor. Increasing levels of vacuum pressure were exerted in the sensor and its output voltage was compared with the barometer reads, in kPa. The manual pump is hard to control which caused the values to be spaced irregularly. The output voltage levels are mapped to an integer value between 0 and 1023.

The graphic in the figure 5.1 shows the output voltage in the horizontal axis and the barometer values in the vertical axis. The linear relation between the negative pressure and the sensor value is verified, and is described as:

$$y = 0,2863x - 7,2073$$

Where the x corresponds to the sensor output value, mapped between 0 and 1023, and the y to the negative pressure value, in kPa. The high correlation coefficient confirms the linear relation between the pressure and output voltage, and confirms that the sensor is very reliable and provides accurate values.

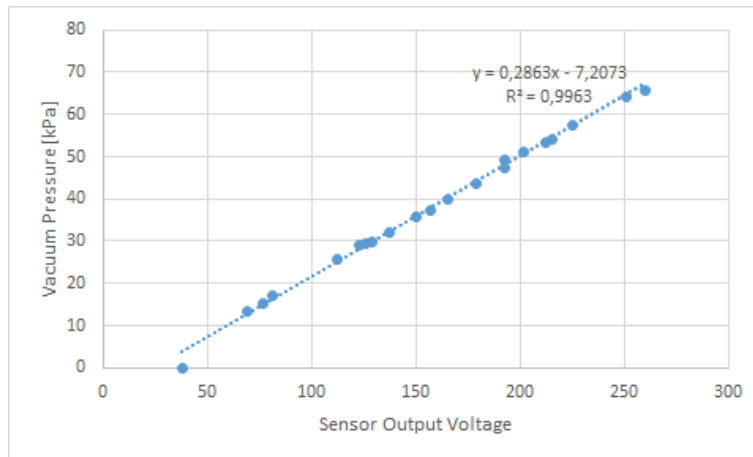


Figure 5.1: Barometer negative pressure values as a function of MPX4250DP sensor output voltage

## 5.1.2 Soil Humidity Sensors

As explain in the section 4.1.2, in order to calibrate a humidity sensor, it is necessary to obtain the sensor's voltage output when the sensor is placed in the air, 0% humidity, and underwater, 100% humidity. With these two values it is possible to estimate humidity percentages through the sensor's output voltage, using interpolation.

In this calibration, the sensor voltage output, mapped between 0 and 1023, is measured repeatedly, second by second, during 2 minutes. This process is done with the sensor fully submerged underwater and repeated with the sensor placed on air. The values obtain from the measurements of one sensor are presented in the figure 5.2. Since there is an inverse ratio between the sensor output value and the soil moisture, the output value is higher when the sensor is placed on air then on water. Therefore, this sensor's upper value is the maximum value of 887, while the lower value is the minimum value of 585.

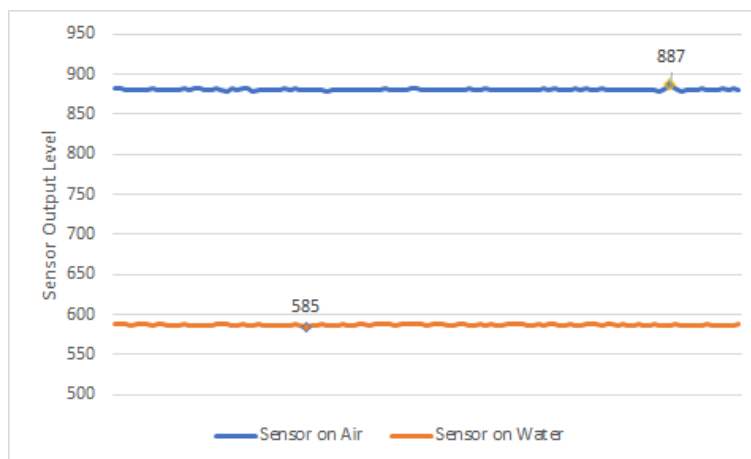


Figure 5.2: Sensor output evolution during the calibration process, with the sensor installed placed on air and submerged on water

This process was repeated with other two sensors and the final results are described in the table 5.1. The results are consistent throughout the three lysimeters, which indicates that thy can be used together. It also shows that the device described in 4.4 can be used to calibrate the sensors, given the



simplicity of the process.

Table 5.1: Minimum and maximum sensor output values obtained during the calibrations of three sensors

Sensor	Minimum Value	Maximum Value
Sensor 1	887	585
Sensor 2	895	585
Sensor 3	892	580

## 5.2 Soil Humidity Threshold Study

In order to calibrate the device parameters to the soil profile of the installation site, a study should be performed. This study is necessary to obtain the soil humidity thresholds and discern the extraction process in said soil. It will also verify if the sampling hardware used in the device submitted to the study is able to extract a sample.

This study consists of several iterations, where a 30cm lysimeter and a soil humidity sensor are installed in a container, with known dimensions, that is filled with the soil collected at the installation site. By pouring water in the container, it is possible to mimic a precipitation event. Knowing the volume of the container and the wanted precipitation level, allows the volume of the poured water to be calculated. The water level is the height of the volume of poured water, considering the surface area of the container. This value corresponds to the total precipitation level, measured in millimeters, accumulated during a certain period of time which can be related to the soil humidity level, measured by the sensor. By attaching the lysimeter to the device and continuously try to extract the sample, it is also possible to relate the soil humidity level with the amount of water extracted, and, consequently, with the precipitation level.

Each iteration lasts at most four hours and consists in pouring a predefined volume of water evenly in the container and exerting the required pressure for extraction in the lysimeter. Every minute, the soil humidity value is logged. At the end of the iteration the volume of the extracted sample is also logged. In the next iteration, the water level is increased, the collected sample is ditched and the process is repeated. New iterations are performed with increasing water levels until the collected sample reaches 50cc in volume or the sample size stabilizes.

### 5.2.1 Experiment Setup

During this thesis, this study was performed in two different soil types: organic soil and sand offering contrasting textural characteristics. The main objective was to assess if soil humidity values required to extract 50cc of water varied significantly for contrasting soil types. The equipment used was a simplified version of the Full Collector device, with only one lysimeter and no valves. The software was also adapted, so that the device is continuously maintaining the same pressure level and sending the sensor data every minute. Since the experiment was performed in a laboratory with Wi-Fi access, the device

was connected to a Raspberry Pi that was programmed to send the device logs to the HTTP server directly.

A plastic vase was used as the container. This container has the shape of a truncated cone, with circular bases, where the bottom base area is different from the top base area. To calculate the volume of water that corresponds to a water level, the containers needs to have a constant area throughout its vertical axis. Since the container area varies between the top and bottom areas, the container shape was approximated to a cylinder with an area equal to the average between the top and bottom areas, as seen in figure 5.3.

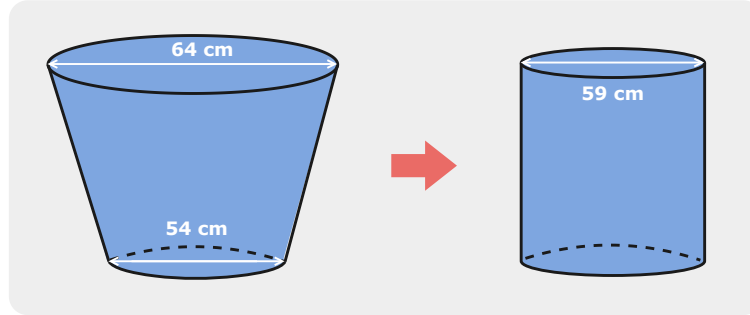


Figure 5.3: Representation of the container used and its cylinder equivalent

The container has a bottom base diameter of 54cm and a top base diameter of 64cm, which results in a 59cm average diameter and a  $2734\text{cm}^2$  area, approximately. The height is equals to 55cm, and is considered the same in the cylinder approximation. Therefore the poured water volume can be calculated through the cylinder volume definition

$$Volume [ml] = Base Area [cm^2] \times Water Level [cm] = 2734\text{cm}^2 \times 0.1 Water Level [ml]$$

where a water level of 1mm corresponds to 273ml. The container used has holes at the bottom that allow the water to flow to a different container. The volume of leaked water was also measured which allowed the calculation of the current amount of water in the container.

The organic soil is typically used for house plants and contains over 60% of organic matter and mineral fertilizer. It has a granulometry that varies between 0 to 15mm, although, it is observable that it is mostly composed of very small particles, with a texture similar to clay where the granulometry ranges between 0 to 0.002 mm, according to the International Standard ISO 14688-1:2017 [72].

The type of sand used in the second experiment was coarse sand. Since no information is known about the granulometry, it is estimated that it varies between 0.063 to 2 mm, according to the International Standard ISO 14688-1:2017 [72].

The experiments results for the different soil types are described in the following sections.

## Water Volume Evolution

Figures 5.4 and 5.5 show the variation of the volume of water added, water leaked, water currently in the container and water extracted throughout the experiment iterations. The volume of water added,

water leaked and water in the container is shown in the left vertical axis, while the volume of water extracted is shown in the right vertical axis. In the horizontal axis, the water levels of each iteration are represented.

In the figure 5.4 it is shown that, in the organic soil, water samples can be extracted with water levels above 15mm, while in figure 5.5, the repeated experiment for sandy soil, water samples were only extracted levels above 4mm. In the first experiment, 15 iterations were performed, with the added water level reaching 34mm and the volume of extracted samples stabilizing around 44ml. During the second experiment, 10 iterations were performed, reaching a water level of 10mm and a volume of samples around 38 ml.

In both figures it is possible to observe that, after a certain water level, the volume of water leaked increases, until reaching the volume of water added, indicated that the soil is saturated. In the organic soil this occurs at the 22mm water level mark, while in the sand at the 8mm mark. After saturation is reached, there was no need to continue the experiment, as it will show little influence in the soil humidity values (shown below) or in the extraction of 50cc of water using the suction lysimeter.

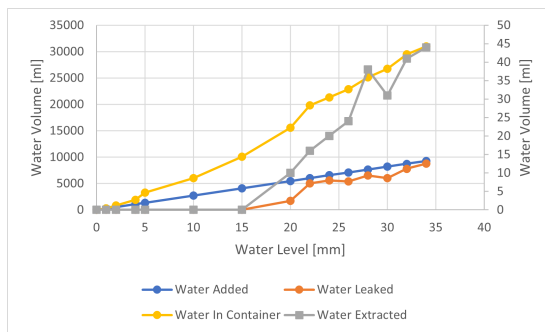


Figure 5.4: Volume of water added, water leaked, water in the container and water extracted according to the water level of the corresponding iteration of the organic soil experiment

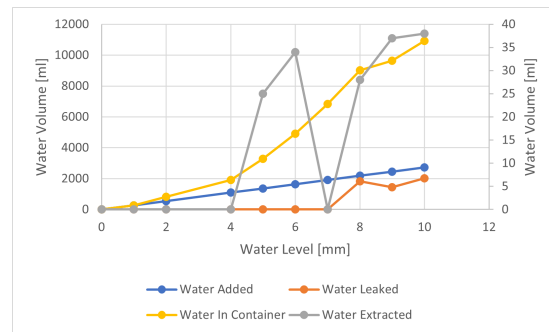


Figure 5.5: Volume of water added, water leaked, water in the container and water extracted according to the water level of the corresponding iteration of the sand experiment

## Sensor Reading

Figures 5.6 and 5.7 show the variation of the soil humidity and pressure in the lysimeter throughout the entire iteration. The values from three iterations, that correspond to the start, middle and end of the experiment, are shown in the graphs. The horizontal axis corresponds to the time of the iteration, while the soil humidity is represented in the left vertical axis and the lysimeter pressure in the right vertical axis.

For the organic soil, the water levels chosen were 0, 15 and 30 mm, where the 15mm level is the minimum level that allows for extractions, as seen above. The figure 5.6 shows that water levels above 15mm correspond to humidity levels between 50 and 80%. It is also possible to observe that the pressure of the lysimeter drops over time, in the first iteration, but is constant, around 50kPa, in the other iterations.

In the sandy soil, the water levels chosen were 0, 5 and 10 mm, where 5 mm corresponds to a water level close to the minimum level required for extraction. Water levels equal or higher than the 5mm

mark, allow sample extraction and correspond to humidity values in the 30 to 75% range, as seen in the figure 5.7. The lysimeter pressure is kept at around 40kPa throughout the entire experiment. It is also important to highlight that, after saturation (at 22mm level for organic soil and 8mm level for the sandy soil), soil humidity stabilizes around 70%.

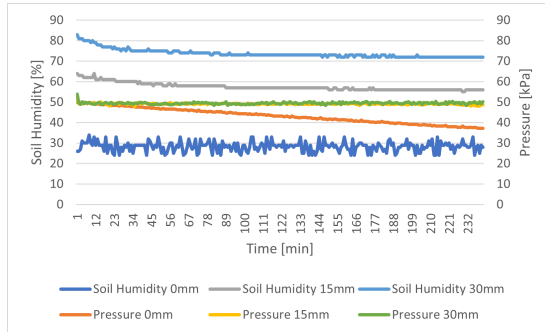


Figure 5.6: Variation of the soil humidity and vacuum pressure throughout the iterations of the organic soil experiment, with water levels equal to 0, 15 and 30mm

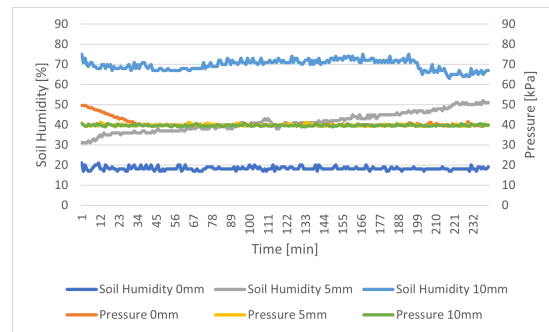


Figure 5.7: Variation of the soil humidity and vacuum pressure throughout the iterations of the sand experiment, with water levels equal to 0, 5 and 10mm

## Conclusions

Firstly, given that samples with significant size were extracted in both types of soil, with different granulometries, it is possible to confirm that the selected hardware, in particular the vacuum pump, is able to extract samples through the lysimeter, while the soil humidity sensors provided accurate measures for both soil profiles, complying with the requirement 6. It can also be assumed that the implemented device will be able to work in most soils within this spectrum of granulometry, such as silts, fine and medium sands and other types, complying with the requirement 5.

The results also show that extractions in sand require much less soil water content than in organic soil, since significant volumes of extracted water were possible for lower water and humidity levels. Another contributing factor that was verified is that the sand soil reached saturation with lower volumes of added water. This implies that coarser soils may require lower humidity threshold values. Regardless, if threshold values for a specific soil type are unknown, it's safe to assume that a humidity value from saturated soil of 70% always guarantees extraction of 50cc of water. Given that in neither of the soils the volume of the extracted sample didn't reach 50cc in four hours, the maximum extraction time should be set to a value higher than four hours.

It was also verified that the pressure levels stayed approximately constant throughout the experiments. In the first iteration of the organic soil experiment, the pressure dropped significantly which may have been a result of an air leakage or another factor. This shows that the device can maintain the required pressure when installed in different soil types.

## 5.3 Battery

Once the implemented device was fully assembled, it was submitted to two performance tests in order to measure the components power consumption and the device battery lifetime.

### 5.3.1 Device Power Consumption

Before performing the tests, the device and its components power consumption were measured. This values are shown in the table 5.2.

Table 5.2: Power consumption per component

Component	Mode	Power Consumption [mA]
Solenoid Air Valve	Operating	530
	Idle	0
Electric Pump	Operating	730
	Idle	0
Arduino Pro Mini	Active	17
	Sleep	3
Shield Adapter Board	-	18
BC66-TE-B	Active	6
	Sleep	2
	Sending Data	110
Pressure Sensor	-	5-10
Water Level Sensor	-	5
Soil Humidity Sensor	-	5
<b>Total Sleep</b>	-	<b>63</b>
<b>Total Active Not Extracting</b>	-	<b>81</b>
<b>Total Active Extracting</b>	-	<b>1341</b>

The total sleep power consumption was calculated by adding the power consumption of the sensors, which are always active, to the BC66-TE-B module and Arduino in sleep mode. The other components are in idle mode with no power consumption. Since the device will spend most of its time in sleep mode, it is possible to estimate its theoretical maximum battery lifetime through it. Given the battery size of 8400 mAh and a consumption of 63 mA, the device has a maximum battery lifetime of approximately 133 hours. The solar panel, with a maximum power current of 163 mA, provides a higher power input than the device sleep mode consumption, thus it should be able to greatly extend the battery lifetime.

## 5.3.2 Battery Performance

### First Test

The first test main objective is to verify the device battery evolution and estimate its battery lifetime. This test lasted 24 hours, in order to verify the behaviour of the solar panel and its influence in the battery level, throughout a full day. The test was also meant to verify the device sleeping mode and message exchange procedures.

In this test, the server was configured to always return a wake up timestamp of one hour after the request time and the device lysimeters were kept at the normal state. With this configuration, the device sends a message to the server, receives the response and sleeps for one hour, then wakes up and repeats this behaviour. This test was performed two times, one time with the device connected to the solar panel, and another without the solar panel. During both times, the device battery voltage level was measured hourly.

It is possible to observe, in figure 5.8, the impact that the solar panel has in the device battery lifetime. Without the solar panel the battery voltage continuously drops throughout the test, while with the solar panel the battery voltage also drops during night time, but from sun rise to the end of the test it is continuously recharging. Although the battery didn't start with maximum capacity, in the end of the test it was fully charged.

### Second Test

The second test was performed in order to verify the device power consumption when in extracting mode and measure the solar panel influence in that mode. The test is executed until the device battery level is low, starting in the morning, in order to expose the solar panel to the maximum amount of sunlight. The battery voltage level was measured hourly.

In this test, the device configuration was changed to always be in extracting mode. In this configuration, each lysimeter is pressurized during 10 seconds, one at a time, with a 30 seconds rest after the pressurizations are complete. This test was maintained until the device, that was not connected to the solar panel, reached a low battery level and repeated for the same duration with the solar panel. The battery was recharged between the tests.

The second test only lasted 8 hours, since the battery level dropped significantly during that time. In the figure 5.9, it is possible to verify that the battery level dropped rapidly. Comparing both repetitions, it can be concluded that the solar panel didn't have much effect on the power loss, as expected.

### Conclusions

These results show that the solar panel is capable of supporting the device during the majority of its operation, complying with requirement 2. Although, extended periods of cloudy weather or long extraction processes could diminish the solar panel effect and lower the battery level significantly. In these cases the device should be able to alert the server.

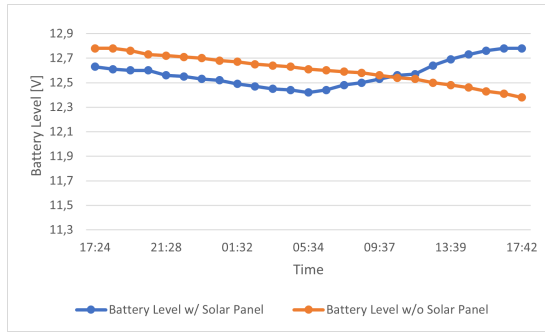


Figure 5.8: Battery Level Evolution during the first test

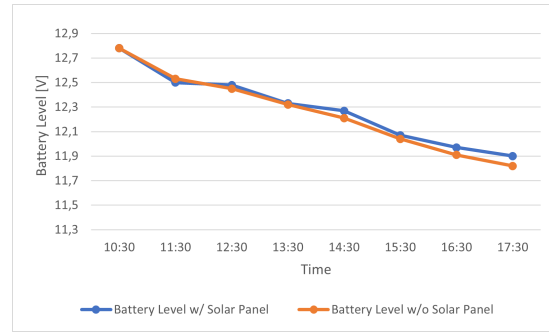


Figure 5.9: Battery Level Evolution during the second test

## 5.4 Cost

The table 5.3 shows all materials necessary to replicate the device and their price. The total price of the device can be estimated by adding all the materials price. The total amount was separated in before and after including the lysimeters since the lysimeters are already part of the extraction sites and may not be considered part of the device. Therefore the device cost is around 243.32 €, not considering the lysimeters, or 359.86 €, considering the lysimeters.

Given the manual operation price of 341.03, referred in section 2.2.1, the total device price, of 359.86 €, is considered to be compliant with the requirement 1.

## 5.5 Laboratory Experiment

The implemented device was installed in a laboratory setting, so that its overall operation could be checked and thoroughly tested. The device was assembled as described in the section 4.1.5, with the solar panel being attached to the battery and connected to three lysimeters. The lysimeters and humidity sensors were individually placed on different containers, in order to be controlled separately and the humidity threshold value was set to 50%, on the three sensors. The communication module was active, and device exchanged messages with the server through the NB-IoT network. The server was configured so that the device sleep time varied between 48 hours, 24 hours and 1 hour. Figure 5.10 shows the server logs created during June 20 to June 26.

Id	Created At	Humidity 20cm	Humidity 40cm	Humidity 60cm	Pressure	Motor 20cm	Motor 40cm	Motor 60cm	Water Level 20cm	Water Level 40cm	Water Level 60cm
1	2021-06-20 10:04:23	2.1	3.3	0.2	3.5	-	-	-	-	-	-
2	2021-06-22 10:05:02	2.3	3.0	0.8	3.5	-	-	-	-	-	-
3	2021-06-24 10:05:54	94.3	3.3	1.7	48.5	ON	-	-	-	-	-
4	2021-06-24 10:33:24	95.1	2.7	1.3	48.5	ON	-	-	Collected	-	-
5	2021-06-24 10:33:56	9.3	4.1	2.3	3.5	-	-	-	Collected	-	-
6	2021-06-25 10:34:22	2.3	3.1	2.3	3.5	-	-	-	Collected	-	-
7	2021-06-26 10:34:53	3.1	4.1	3.3	3.5	-	-	-	Collected	-	-
8	2021-06-26 11:35:17	3.3	4.6	3.4	3.5	-	-	-	Collected	-	-
9	2021-06-26 12:35:45	3.6	3.8	2.7	3.5	-	-	-	Collected	-	-
10	2021-06-26 13:36:23	2.4	95.2	2.2	48.5	-	ON	-	Collected	-	-
11	2021-06-26 13:36:55	2.4	95.2	98.1	48.5	-	ON	ON	Collected	-	-
12	2021-06-26 14:06:44	1.6	95.4	98.5	48.5	-	ON	ON	Collected	Collected	-
13	2021-06-26 14:08:13	2.1	95.4	98.3	48.5	-	-	ON	Collected	Collected	Collected

Figure 5.10: Server logs created during the laboratory experiment

Table 5.3: Device and its materials prices

Material	Price [€]
Shield Adapter Board	2.05
Base Shield	4.05
Arduino Pro Mini	7.75
Battery	30.25
Solar Panel	22.99
Electric Pump	24.35
3 x Soil Humidity Sensors	22.14
Negative Pressure Sensor	20.61
BC66-TE-B	56.50
DS3231	4.39
3 x Air Solenoid Valves	28.88
3 x 50ml Containers	3
4 x 10k resistor	0.56
4 x 150 resistor	0.40
4 x IRF540N transistor	5
4 x 1N4006	0.40
Cables/Tubing and Other Materials	≈ 10
<b>Sub-Total</b>	<b>243.32</b>
Lysimeters 20cm	35.36
Lysimeters 40cm	40.59
Lysimeters 60cm	40.59
<b>Total</b>	<b>359.86</b>

Initially the server was configured to return sleep times of 48 hours. On the fourth day the container with the humidity sensor corresponding to the 20cm lysimeter was filled with water. Since the soil humidity measured was around 94%, the extraction was triggered and the server was notified. After roughly 28 minutes the sample was collected and the server was notified. After this extraction the sleep time was changed to 24 hours, and past two days it was changed to one hour. After a few hours, water was inserted in remaining two containers which started their extraction. With the collection of the remaining samples, the extraction stopped and the server was notified.

This experiment showed that the device communication and extraction procedures are well defined. The devices was able to notify the server and receive the respective sleep timestamps correctly. It was also able to enter sleep mode and wake up at the defined times, with minimal delays. During the experiment, the device was also able to extract several samples simultaneously, while the sensors provided accurate measures.



## 5.6 Field Tests

During the final month of this thesis, beginning in July of 2021, the implemented Full Collector device was installed in the test site, at Companhia das Lezírias estate (latitude 38°49'12.22"N, longitude 8°49'31.32"W). This test installation was done with the objective of gathering sensors data in real conditions and to try to extract a sample in case of a precipitation event. Unfortunately, due to the occurrence communication problems, it was not possible to gather data nor collect samples. However, during the installation of the device it was possible to collect some data regarding the NB-IoT communication.

### 5.6.1 Communication

Figure 5.11 shows that the device is installed in an area with low network coverage, however it is close to strong 3G and 4G signals which provided NB-IoT coverage in the device area, since the same infrastructure is used for both NB-IoT and cellular networks. The device registered signal strengths between -88 and -93 dBm and was able to communicate with the server with latency up to 5000 ms. From the messages sent by the device, more than 90% reached the server and received response, which indicates reasonably low packet loss rates.

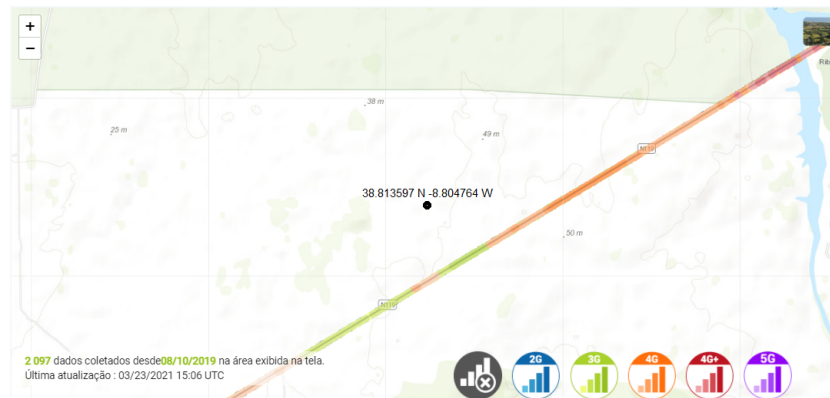


Figure 5.11: MEO Cellular Network Coverage

The battery performance tests 5.3.2 were performed in Lisbon with access to high cellular signal where the device signal strength ranged between -67 to -87 dBm and the server latency was at most 2000 ms.

The device had better communication performance during the tests, and field conditions also provided acceptable values considering the requirement 3. The main problem with the communications during field installation was the frequent lost of access to the NB-IoT network for long time periods. This occurrence was verified as the device stopped being able to send messages and the network registration duration reached values up to 30 minutes.

Due to site access limitations and weather conditions it was not possible to conduct further field tests, which limited the amount of data collected. Therefore, it was not possible to perform a more complex statistical analysis to the communication data.



# Chapter 6

## Discussion

As stated in the previous chapter, the main problems related to the device operation are hardware or communication related. In the following section these problems and possible solutions are analyzed. The validation of the requirements defined in the section 3.1 is also discussed below.

### 6.1 Requirements

The results obtained in the chapter 5 confirmed the validation of several of the established requirements.

The estimated cost of the implemented Full Collector, calculated in the section 5.4, was close to the value of the materials used in the current monitoring sites, validating the Low Cost requirement, 1.

The battery performance tests performed on the implemented device, explained in the section 5.3.2, showed that the device behaviour is power efficient and, using the solar charger, its battery level can be maintained for long durations, complying with the Efficient use of power requirement, 2.

The field tests performed in Companhia das Lezírias, described in the section 5.6.1, showed that the device is able to connect to the NB-IoT network, even in locations with low signal, validating the Reachable requirement, 3. However, depending on the location conditions, some communication difficulties may surge.

The soil humidity threshold study, described in the section 5.2 performed on two very different soil profiles, confirmed that the device can operate in a large range of soil profiles, complying with the Adaptable to multiple soils requirement, 5.

The sensors provided accurate values during the calibration and experiments performed. In addition, the device behaviour during the threshold study and laboratory experiment, confirmed its reliability, verifying the requirement 6.

However, the hardware problems that appeared during the assembly and testing phases indicate that the system complexity could be a hinder to its reproducibility and easiness of use, referred in the requirement 4.

## 6.2 Communication

The main problem with the NB-IoT communications was sudden drops in access to the network. The implemented device architecture, that allows the device to resend messages if the server response doesn't reach the device, and is not able to solve or mitigate this problem, since the BC66-TE-B module is not able to register itself with the network in a period that does not compromise the device operation. This problem only occurred at the field site, as the NB-IoT signal was significantly lower than in the tests location, which raises the question whether it is possible to reproduce this device to other remote locations.

Since the NB-IoT network signal strength is defined through the currently installed infrastructure, it is not possible to guarantee the device correct operation in locations with low NB-IoT signal strength. One possible solution, is to develop an alternative version of the device that uses communication based on other wireless technologies, such as other LPWAN or cellular networks, depending on the coverage conditions, although, based on the current LPWAN coverage in Portuguese territory, described in section 2.3.1, only cellular or SIGFOX may provide reliable alternatives.

## 6.3 Hardware

Another significant challenge faced during the device implementation and installation derived from hardware malfunctions. These malfunctions were mostly related to the sampling hardware section components. Malfunctions such as individual components breaking, like the electric pump or valves, tubing leakage or poorly isolated humidity sensors were fairly common and affect the device robustness. The device robustness is important since it needs to be transported to remote locations that may be hard to access, while being easily reproducible.

The idea of developing an alternative device (Notificator device) arose from these challenges as a way to greatly simplify the previous device that also fulfills the objective of easing the sampling process while complying with all the requirements defined in 3.1. The Notificator device is described below.

### 6.3.1 Notificator Device

As stated in section 1.3, the Notificator device is a simpler iteration of the Full Collector device that does not contain the sampling related hardware and control software, as shown in the device architecture. For this reason, a manual vacuum pump needs to be used to extract the samples.

Hardware-wise, all the other modules remain, specifically the operation control, power supply and communication. Software-wise, the device behaviour will also be similar to the Full Collector, but simpler. In this device, the lysimeter possible states are Normal or Collecting. The device is initialized with all lysimeters in Normal state, until the soil humidity surpasses the threshold value, where the state is changed to Collecting, indicating that a sample can be extracted in the respective lysimeter. When a state changes the user is alerted in the same way as the Full Collector. The state is kept until the operator comes to collect the sample and reset the device. In cases where the operator can't proceed

to the device, if sufficient time has passed since the end of the precipitation event, the soil humidity may decrease to a lower value than the predefined threshold. In this case the state reverts back to Normal and the operator is alerted. The startup, sleep mode and messaging and configuration protocols, described in the section 4.3.2, are kept in this device as well, although the pressure parameter "p" is not part of the message format. The device and software architectures are depicted in the figures 6.1 and 6.2, respectively.

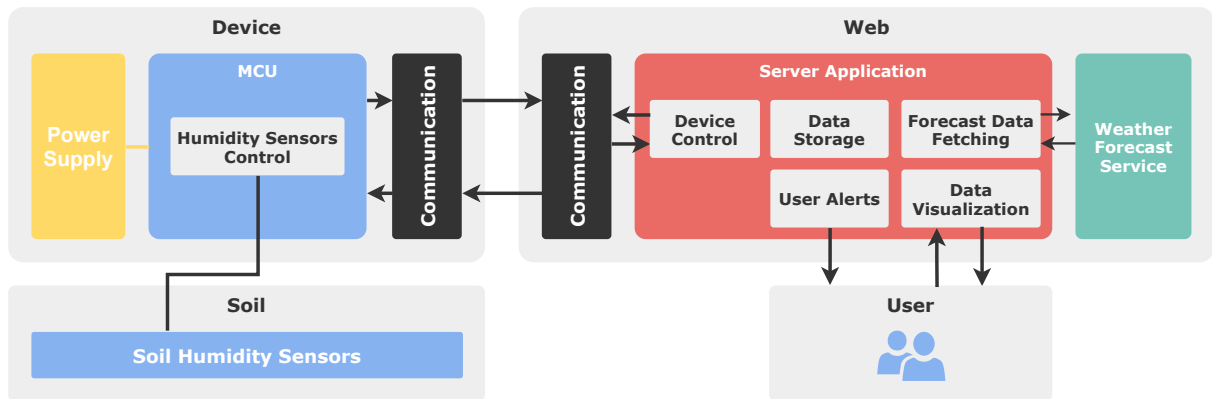


Figure 6.1: Notificator Device Architecture Diagram

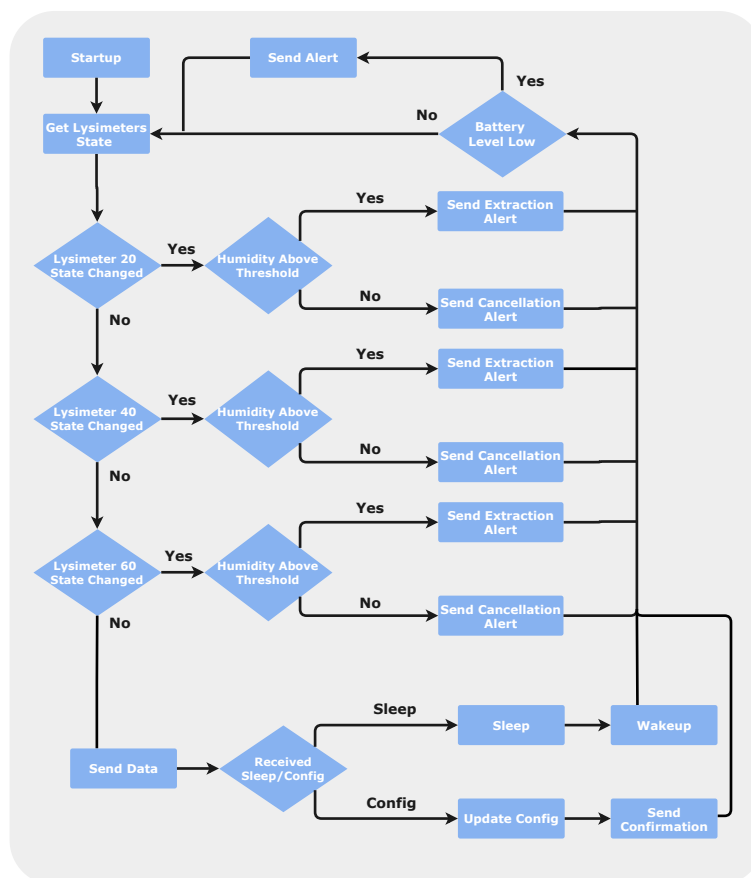


Figure 6.2: Notificator Software Architecture Diagram

This device's power requirements are similar to the Full Collector device, when the second is not

extracting samples, since the driver circuit used does not transfer power to the pump and valves while in idle mode. Therefore the battery performance tests 5.3.1, where the device was not extracting, are also valid with the Notificator device.

Ultimately, this device is not only easier to replicate and less susceptible to hardware problems but also a significantly cheaper alternative, costing 155.12€ compared with the Full Collector 243.32€, excluding lysimeters, as seen in the table 6.1. Additionally, even if the process is less automated than in the Full Collector device, the operator is required to proceed to the sampling site with both devices, spending the same dislocation costs.

Table 6.1: Notificator Device and its materials prices

Material	Price [€]
Shield Adapter Board	2.05
Base Shield	4.05
Arduino Pro Mini	7.75
Battery	30.25
Solar Panel	22.99
3 x Soil Humidity Sensors	22.14
BC66-TE-B	56.50
DS3231	4.39
Cables and Other Materials	≈ 5
<b>Total</b>	<b>155.12</b>

## Chapter 7

# Conclusions and future work

As European directives push for frequent analysis of soil water chemicals, areas with low precipitation rates, such as Portugal, have difficulty in collecting soil water samples using suction lysimeters, given the low soil water content. Current devices capable of automating this procedure and help countries like Portugal to reach acceptable sampling rates, are scarce or are based on other types of lysimeters, other than suction lysimeters. This thesis proposes two devices capable of automating and easing the sample extraction process, the Full Collector device and the Notificator device, which is a simplified version of the first.

Both devices are similar, and use soil humidity sensors to detect optimal soil water levels, that allow for the extraction of samples with the required size. When the soil humidity is optimal the Full Collector device automatically extracts a sample through an electric vacuum pump that pressurizes the lysimeter and alerts the operator that a sample was collected, while the Notificator device only alerts the operator that a sample can be extracted, leaving the extraction to the operator. These devices are battery and solar panel powered and have an important communication component responsible for communicating with a central server that alerts the operator through email messages. This communication is based on IoT LPWAN wireless technologies, more specifically NB-IoT, using the CoAP protocol proxied with an HTTP server.

An implementation of the Full Collector device was assembled and submitted to operation and battery performance tests. These tests showed that the device is adaptable and can extract samples in soils with very different textures, and that each soil requires a different soil humidity threshold that triggers the extraction process. The tests also showed that by using a solar panel the device battery lifetime can be significantly increased. However, it was verified that the hardware used in the extraction process of the Full Collector device highly increases the overall device complexity which makes it hard to reproduce and transport to other locations, since it is very susceptible to malfunctions. This challenge resulted in the Notificator device being designed, where the sampling hardware is removed from the device, greatly simplifying its architecture which boosts reproducibility and lowers the price. Additionally, given that the sampling sites are located in isolated places, with reduced access to the NB-IoT network, it is hard to guarantee the device reliability in those locations, due to communication errors.

In conclusion, the developed devices showed favorable results and met the most part of the proposed requirements. However there is still room for testing and improvement. Firstly, a thorough review on the availability of the NB-IoT network in Portuguese territory needs to be performed, in order to verify the reliability of the network in the current and future monitoring sites. Secondly, alternative wireless technologies should be studied and tested, in order to provide a wider range of available technologies from where the most adequate for a location site can be chosen. Further in-situ extraction tests should be performed on the field, in order to test the device in real environment conditions. This tests should also allow the operator to use and interact with the device.



# Bibliography

- [1] Directive (EU) 2016/2284 of the European Parliament and of the Council of 14 December 2016 on the reduction of national emissions of certain atmospheric pollutants, amending Directive 2003/35/EC and repealing Directive 2001/81/EC [2016] OJ L 344/1.
- [2] *Decreto Lei nº 204/2018 de 23 de outubro da Presidência do Conselho de Ministros*. Diário da República: I série, Nº 204 (2018).
- [3] T. A. Howell, A. D. Schneider, and M. E. Jensen. History of lysimeter design and use for evapotranspiration measurements. In *Lysimeters for evapotranspiration and environmental measurements*, pages 1–9. ASCE, 1991. URL <https://cedb.asce.org/CEDBsearch/record.jsp?dockkey=0072381>.
- [4] T. A. Howell. Lysimetry. In *Encyclopedia of Soils in the Environment*, pages 379–386. Elsevier, 2005.
- [5] T. Pütz, J. Fank, and M. Flury. Lysimeters in vadose zone research. *Vadose zone journal*, 17(1): 1–4, 2018. doi: 10.2136/vzj2018.02.0035.
- [6] G. Gee and J. Jones. Lysimeters at the hanford site: present use and future needs. Technical report, Pacific Northwest Labs., 1985.
- [7] M. A. Oliveira, C. Branquinho, H. Serrano, and P. Pinho. Ecosystem monitoring in portugal, national report. Technical report, cE3c - Centro de Ecologia, Evolução e Alterações Ambientais, FCUL - Faculdade de Ciências da Universidade de Lisboa, Portugal, 2019.
- [8] L. WILSON and J. ARTIOLA. Chapter 7 - soil and vadose zone sampling. In J. F. Artiola, I. L. Pepper, and M. L. Brusseau, editors, *Environmental Monitoring and Characterization*, pages 101–119. Academic Press, Burlington, 2002. ISBN 978-0-12-064477-3. doi: 10.1016/B978-012064477-3/50009-6.
- [9] D. A. d. Oliveira, A. Pinheiro, and M. d. Veiga. Effects of pig slurry application on soil physical and chemical properties and glyphosate mobility. *Revista Brasileira de Ciência do Solo*, 38:1421–1431, 2014. doi: 10.1590/S0100-06832014000500007.
- [10] E. E. Agency. Indicator assessment - mean precipitation, 2017. URL <https://www.eea.europa.eu/data-and-maps/indicators/european-precipitation-2/assessment>.

- [11] Decision N° 1386/2013/EU of the European Parliament and of the Council of 20 November 2013 on a General Union Environment Action Programme to 2020 'Living well, within the limits of our planet' [2013] OJ L 354/171.
- [12] Nieminen T. M., De Vos B., Cools N., König N., Fischer R., Iost S., Meessenburg H., Nicolas M., O'Dea P., Cecchini G., Ferretti M., De La Cruz A., Derome K., Lindroos A. J., Graf Pannatier E., "Part XI: Soil Solution Collection and Analysis", in *UNECE ICP Forests Programme Co-ordinating Centre (ed.): Manual on methods and criteria for harmonized sampling, assessment, monitoring and analysis of the effects of air pollution on forests*. Thünen Institute of Forest Ecosystems, Eberswalde, Germany, 20 p, 2016. Retrieved from <http://www.icp-forests.org/manual.html>.
- [13] L. Bergström. Use of lysimeters to estimate leaching of pesticides in agricultural soils. *Environmental pollution*, 67(4):325–347, 1990. doi: 10.1016/0269-7491(90)90070-S.
- [14] A. Reck, E. Paton, and B. Kluge. Advanced in situ soil water sampling system for monitoring solute fluxes in the vadose zone. *Vadose Zone Journal*, 2019. doi: 10.2136/vzj2019.01.0008.
- [15] Q. Wang, K. Cameron, G. Buchan, L. Zhao, E. Zhang, N. Smith, and S. Carrick. Comparison of lysimeters and porous ceramic cups for measuring nitrate leaching in different soil types. *New Zealand Journal of Agricultural Research*, 55(4):333–345, 2012. doi: 10.1080/00288233.2012.706224.
- [16] G. Singh, G. Kaur, K. Williard, J. Schoonover, and J. Kang. Monitoring of water and solute transport in the vadose zone: A review. *Vadose Zone Journal*, 17(1):1–23, 2018. doi: 10.2136/vzj2016.07.0058.
- [17] Meter. Smart field lysimeter user manual. URL [http://library.metergroup.com/Manuals/UMS/SmartField-Lysimeter%20User%20Manual\\_Version\\_05-2020.pdf](http://library.metergroup.com/Manuals/UMS/SmartField-Lysimeter%20User%20Manual_Version_05-2020.pdf). Last accessed on July 2021.
- [18] Eijkelkamp. Eijkelkamp smart lysimeters user manual. URL [https://www.eijkelkamp.com/download.php?file=M1681e\\_Eijkelkamp\\_Smart\\_Lysimeters\\_29c2.pdf](https://www.eijkelkamp.com/download.php?file=M1681e_Eijkelkamp_Smart_Lysimeters_29c2.pdf). Last accessed on July 2021.
- [19] A. Farsad, A. Sadeghpour, M. Hashemi, M. Battaglia, and S. Herbert. A review on controlled vacuum lysimeter design for soil water sampling. *Environmental Technology & Innovation*, 14: 100355, 2019. doi: 10.1016/j.eti.2019.100355.
- [20] A. Farsad, S. Herbert, M. Hashemi, and A. Sadeghpour. An automated suction lysimeter for improved soil water sampling. *Vadose Zone Journal*, 11(4):vzj2012–0003, 2012. doi: 10.2136/vzj2012.0003.
- [21] A. d. Costa, J. A. Albuquerque, A. d. Costa, P. Pértile, and F. R. d. Silva. Water retention and availability in soils of the state of santa catarina-brazil: effect of textural classes, soil classes and lithology. *Revista Brasileira de Ciência do Solo*, 37(6):1535–1548, 2013. doi: 10.1590/S0100-06832013000600010.

- [22] F. Meskini-Vishkaee, M. H. Mohammadi, and M. Vanclooster. Predicting the soil moisture retention curve, from soil particle size distribution and bulk density data using a packing density scaling factor. *Hydrology and Earth System Sciences*, 18(10):4053–4063, 2014. doi: 10.5194/hess-18-4053-2014.
- [23] R. K. Mobley. 13 - characteristics of compressed air. In R. K. Mobley, editor, *Fluid Power Dynamics*, pages 194–202. Newnes, Woburn, 2000. ISBN 978-0-7506-7174-3. doi: 10.1016/B978-075067174-3/50064-0. URL <http://www.sciencedirect.com/science/article/pii/B9780750671743500640>.
- [24] A. Čolaković and M. Hadžialić. Internet of things (iot): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, 144:17–39, 2018. doi: 10.1016/j.comnet.2018.07.017.
- [25] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking. A review on internet of things (iot), internet of everything (ioe) and internet of nano things (iont). In *2015 Internet Technologies and Applications (ITA)*, pages 219–224. IEEE, 2015. doi: <https://doi.org/10.1109/ITechA.2015.7317398>.
- [26] A. Čolaković and M. Hadžialić. Internet of things (iot): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, 144:17–39, 2018. doi: 10.1016/j.comnet.2018.07.017.
- [27] W. Yang, M. Wang, J. Zhang, J. Zou, M. Hua, T. Xia, and X. You. Narrowband wireless access for low-power massive internet of things: A bandwidth perspective. *IEEE wireless communications*, 24(3):138–145, 2017. doi: 10.1109/MWC.2017.1600298.
- [28] B. T. Inc. Leading types of iot wireless tech and their best use cases. URL <https://behrtechnologies.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases>. Last accessed on June 2021.
- [29] LoRa Alliance, Inc. Lorawan™ 1.1 specification. URL [https://loro-alliance.org/wp-content/uploads/2020/11/lorawantm\\_specification\\_v1.1.pdf](https://loro-alliance.org/wp-content/uploads/2020/11/lorawantm_specification_v1.1.pdf). Last accessed on June 2021.
- [30] R. Kalfus and T. Hégr. Ultra narrow band radio technology in high-density built-up areas. In *International Conference on Information and Software Technologies*, pages 663–676. Springer, 2016. URL [https://link.springer.com/chapter/10.1007/978-3-319-46254-7\\_54](https://link.springer.com/chapter/10.1007/978-3-319-46254-7_54).
- [31] A. D. Zayas and P. Merino. The 3gpp nb-iot system architecture for the internet of things. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 277–282. IEEE, 2017.
- [32] R. S. Sinha, Y. Wei, and S.-H. Hwang. A survey on lpwa technology: Lora and nb-iot. *Ict Express*, 3(1):14–21, 2017. doi: 10.1016/j.ict.2017.03.004.

- [33] Altice. Narrowband-iot is the future and altice portugal are bringing it. URL <https://www.telecom.pt/en-us/media/noticias/Pages/2018/novembro/NarrowBand-IoT-.aspx>. Last accessed on June 2021.
- [34] NOS. Nos lanÇa nova gama de tarifÁrios de nb-iot preparados para 5g. URL <https://www.nos.pt/institucional/PT/media/Documents/2020%2009%2024%20NOS%20LAN%C3%87A%20NOVA%20GAMA%20DE%20TARIF%C3%81RIOS%20DE%20NB-IOT%20PREPARADOS%20PARA%205G.pdf>. Last accessed on June 2021.
- [35] SigFox. Sigfox coverage. URL <https://www.sigfox.com/en/coverage>. Last accessed on June 2021.
- [36] T. T. N. Portugal. The things network. URL <https://www.thethingsnetwork.org/country/portugal/>. Last accessed on June 2021.
- [37] N. Naik. Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In *2017 IEEE international systems engineering symposium (ISSE)*, pages 1–7. IEEE, 2017. doi: <https://doi.org/10.1109/SysEng.2017.8088251>.
- [38] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan. Performance evaluation of mqtt and coap via a common middleware. In *2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)*, pages 1–6. IEEE, 2014. doi: <https://doi.org/10.1109/ISSNIP.2014.6827678>.
- [39] G. Choudhary and A. Jain. Internet of things: A survey on architecture, technologies, protocols and challenges. In *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pages 1–8. IEEE, 2016. doi: <https://doi.org/10.1109/ICRAIE.2016.7939537>.
- [40] B. Hock, M. Heaphy, M. Evans, A. Dunningham, B. Graham, et al. The internet of things for forestry: New concepts, new opportunities. *NZJ For*, 60:25, 2016. URL [http://nzjf.org.nz/free\\_issues/NZJF60\\_4\\_2016/906E11D1-DC1D-46ec-AC79-AB5825829AF8.pdf](http://nzjf.org.nz/free_issues/NZJF60_4_2016/906E11D1-DC1D-46ec-AC79-AB5825829AF8.pdf).
- [41] X. Feng, F. Yan, and X. Liu. Study of wireless communication technologies on internet of things for precision agriculture. *Wireless Personal Communications*, 108(3):1785–1802, 2019. doi: 10.1007/s11277-019-06496-7.
- [42] Z. Yu, L. Xugang, G. Xue, and L. Dan. Iot forest environmental factors collection platform based on zigbee. *Cybern. Inf. Technol*, 14(5):51–62, 2014. URL [http://www.cit.iit.bas.bg/CIT\\_2014/v14-s/6BJTU--06-md-Gotovo.pdf](http://www.cit.iit.bas.bg/CIT_2014/v14-s/6BJTU--06-md-Gotovo.pdf).
- [43] N. Gondchawar, R. Kawitkar, et al. Iot based smart agriculture. *International Journal of advanced research in Computer and Communication Engineering*, 5(6):838–842, 2016. URL <http://truevolts.com/wp-content/uploads/2017/11/47.pdf>.

- [44] A. Rahali, M. Guerbaoui, A. Ed-Dahhak, Y. El Afou, A. Tannouche, A. Lachhab, and B. Bouchikhi. Development of a data acquisition and greenhouse control system based on gsm. *International Journal of Engineering, Science and Technology*, 3(8):297–306, 2011. doi: 10.4314/ijest.v3i8.23.
- [45] G. Valecce, P. Petruzzi, S. Strazzella, and L. A. Grieco. Nb-iot for smart agriculture: Experiments from the field. In *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, volume 1, pages 71–75. IEEE, 2020. doi: <https://doi.org/10.1109/CoDIT49905.2020.9263860>.
- [46] Electronics-Lab. Top 10 popular microcontrollers among makers. URL <https://www.electronics-lab.com/top-10-popular-microcontrollers-among-makers>. Last accessed on July 2021.
- [47] Atmel. Atmega328p datasheet, . URL [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf). Last accessed on July 2021.
- [48] Atmel. Atmega16u4/atmega32u4 datasheet, . URL [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf). Last accessed on July 2021.
- [49] Espressif Systems. Esp8266ex datasheet, . URL [https://espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf). Last accessed on July 2021.
- [50] Espressif Systems. Esp32 series datasheet, . URL [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf). Last accessed on July 2021.
- [51] J. Teixeira and R. C. dos Santos. Exploring the applicability of low-cost capacitive and resistive water content sensors on compacted soils. *Geotechnical and Geological Engineering*, 39(4):2969–2983, 2021. doi: 10.1007/s10706-020-01672-0.
- [52] S. Adla, N. K. Rai, S. H. Karumanchi, S. Tripathi, M. Disse, and S. Pande. Laboratory calibration and performance evaluation of low-cost capacitive and very low-cost resistive soil moisture sensors. *Sensors*, 20(2):363, 2020. doi: 10.3390/s20020363.
- [53] Irrrometer. Ssat instalation and operation instructions. URL [https://www.irrometer.com/pdf/instruction-manuals/ssat/728\\_SSAT\\_Manual\\_WEB.pdf](https://www.irrometer.com/pdf/instruction-manuals/ssat/728_SSAT_Manual_WEB.pdf). Last accessed on July 2021.
- [54] Airpon. D2028 mini 12v dc electric air vacuum compressor. URL [https://www.alibaba.com/product-detail/Airpon-D2028-Mini-12v-dc-electric\\_354079276.html](https://www.alibaba.com/product-detail/Airpon-D2028-Mini-12v-dc-electric_354079276.html). Last accessed on June 2021.
- [55] Hoypeyfiy. 12v solenoid valve 1/4" normally close electric fast response air water solenoid valve. URL <https://www.amazon.co.uk/Solenoid-Valve-Normally-Electric-Response/dp/B07VWTM3XG>. Last accessed on June 2021.

- [56] DF Robot. Capacitive soil moisture sensor sku:sen0193. URL [https://wiki.dfrobot.com/Capacitive\\_Soil\\_Moisture\\_Sensor\\_SKU\\_SEN0193](https://wiki.dfrobot.com/Capacitive_Soil_Moisture_Sensor_SKU_SEN0193). Last accessed on July 2021.
- [57] NXP. Mpx4250d datasheet. URL <https://www.nxp.com/docs/en/data-sheet/MPX4250D.pdf>. Last accessed on June 2021.
- [58] D. Robot. Non-contact liquid level sensor xkc-y25-t12v sku sen0204. URL [https://wiki.dfrobot.com/Non-contact\\_Liquid\\_Level\\_Sensor\\_XKC-Y25-T12V\\_SKU\\_\\_SEN0204](https://wiki.dfrobot.com/Non-contact_Liquid_Level_Sensor_XKC-Y25-T12V_SKU__SEN0204). Last accessed on June 2021.
- [59] Quectel. Bc66-te-b user guide, . URL [https://www.quectel.com/download/quectel\\_bc66-te-b\\_user\\_guide\\_v2-1/](https://www.quectel.com/download/quectel_bc66-te-b_user_guide_v2-1/). Last accessed on July 2021.
- [60] Quectel. Bc66-opencpu hardware design, . URL [https://www.quectel.com/download/quectel\\_bc66\\_hardware\\_design\\_v1-3/](https://www.quectel.com/download/quectel_bc66_hardware_design_v1-3/). Last accessed on July 2021.
- [61] Arduino. Arduino pro mini. URL <https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardProMini>. Last accessed on June 2021.
- [62] M. Integrated. Ds3231 datasheet. URL <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>. Last accessed on July 2021.
- [63] S. Studio. Grove base shield v2.0 for arduino product details, . URL <https://www.seeedstudio.com/Base-Shield-V2.html>. Last accessed on July 2021.
- [64] OPEN-SMART. Pro mini uno shield adapter board. URL <https://pt.aliexpress.com/item/32779738543.html>. Last accessed on June 2021.
- [65] S. Studio. Grove system, . URL [https://wiki.seeedstudio.com/Grove\\_System/](https://wiki.seeedstudio.com/Grove_System/). Last accessed on July 2021.
- [66] Toshiba. Mosfet gate drive circuit application note. URL <https://toshiba.semicon-storage.com/info/docget.jsp?did=59460>. Last accessed on May 2021.
- [67] F. S. Corporation. Irf540n datasheet. URL <https://media.digikey.com/pdf/Data%20Sheets/Fairchild%20PDFs/IRF540N.pdf>. Last accessed on May 2021.
- [68] Quectel. Bc66&bc66-na-quecopen user guide, . URL [https://www.quectel.com/download/quectel\\_bc66-te-b\\_user\\_guide\\_v2-1/](https://www.quectel.com/download/quectel_bc66-te-b_user_guide_v2-1/). Last accessed on July 2021.
- [69] Quectel. Bc66&bc66-na at commands manual, . URL [https://www.quectel.com/download/quectel\\_bc66bc66-na\\_at\\_commands\\_manual\\_v2-2/](https://www.quectel.com/download/quectel_bc66bc66-na_at_commands_manual_v2-2/). Last accessed on July 2021.
- [70] OpenWeather. One call api. URL <https://openweathermap.org/api/one-call-api>. Last accessed on June 2021.
- [71] H. T. Electronics. Interface capacitive soil moisture sensor v1.2 with arduino. URL <https://how2electronics.com/interface-capacitive-soil-moisture-sensor-arduino>. Last accessed on July 2021.

- [72] ISO 14688-1:2017. Geotechnical investigation and testing — identification and classification of soil — part 1: Identification and description. Standard, International Organization for Standardization, Geneva, CH, Dec. 2017.

