INSTITUTO
UNIVERSITÁRIO
DE LISBOA

SoilIoT - Smart Sensing and IoT for Precision Agriculture - Soil Characteristics Monitoring

Oleksandr Kobelyuk

Master in Telecommunications and Computer Engineering

Supervisor:
PhD Octavian Adrian Postolache, Full Professor
Iscte – Instituto Universitário de Lisboa

Co-Supervisor:
MSc Bruno Miguel Gonçalves Mataloto, Invited Assistant Professor,
Iscte – Instituto Universitário de Lisboa

September, 2024

TECHNOLOGY
AND ARCHITECTURE

SoilIoT - Smart Sensing and IoT for Precision Agriculture - Soil Characteristics Monitoring

Oleksandr Kobelyuk

Master in Telecommunications and Computer Engineering

Supervisor:
PhD Octavian Adrian Postolache, Full Professor
Iscte – Instituto Universitário de Lisboa

Co-Supervisor:
MSc Bruno Miguel Gonçalves Mataloto, Invited Assistant Professor,
Iscte – Instituto Universitário de Lisboa

September, 2024

To you, whoever you may be.

Acknowledgements

I would like to thank Professor Octavian Postolache and Professor Bruno Mataloto for their supervision and support throughout the writing of this dissertation. I would also like to thank Iscte - Instituto Universitario de Lisboa and Instituto de Telecomunicações for the technical support and knowledge given during the whole period of my dissertation.

And most of all, I would like to thank my family and my friends, both here and abroad, who accompanied me throughout these 5 short years, which flew by in the blink of an eye, and have seen me grow as a person, thank you so much from the bottom of my heart. I wouldn't be the person I am today without you.

Resumo

Com a população da terra sempre a crescer, a produção agricola mundial também tem de crescer adequadamente, ao mesmo tempo a agricultura sustentável com preservação das caracteristicas do solo para atingir grandes volumes de colheitas também é importante. Existem e são aplicadas diversas estratégias de gestão do solo, e a agricultura de precisão é uma das mais importantes.

No tópico da agricultura de precisão e da preservação das caracteristicas do solo, podemos dizer que os macronutrientes, nomeadamente o Nitrogénio, o Fósforo e o Potássio, são extremamente importantes quando queremos atingir colheitas volumosas. Neste contexto, monitorizar estes macronutrientes no solo e reagir às suas mudanças de forma adequada é uma necessidade para conseguir atingir uma agricultura sustentável, e isto é extamante o que é a agricultura de precisão. Soluções à base da Internet das Coisas têm surgido como boas opções para atingir este tipo de agricultura, combinando *hardware*, *software* e protocolos de transmissão de dados para atingir uma monitorização contínua do solo, com dados visíveis numa interface gráfica conveniente para o utilizador.

Esta dissertação propõe uma implementação de um sistema de monitorização contínua do solo com nós localizáveis geográficamente, na qual os dados são enviados para uma interface gráfica na qual podem ser monitorizados. O *design* e a validação do sistema, incluíndo uma análise dos dados, estão incluídos na dissertação.

Palavras-Chave: *Internet das Coisas*, *agricultura de precisão*, *LoRaWAN*, *MQTT*, *NPK*

Abstract

With earth's population ever increasing, the output of the agricultural field needs to be increased accordingly, at the same time sustainable agriculture with preservation of soil characteritics for higher yields is an important objective. Different management strategies are reported and applied nowadays, and an important one is precision agriculture.

Regarding precision agriculture and soil characteristics preservation, it can be mentioned that soil macronutrients, more specifically Nitrogen, Phosphorus, and Potassium, are of utmost importance when aiming for high yields. In this context, monitoring the soil for these macronutrients and reacting acordingly is a necessity in order to achieve sustainability, and this is what is encompassed by precision agriculture. Internet of Things (IoT) based solutions arise as prime candidates to achieve this type of agriculture, by combining hardware, software and data transmission protocols, continuous and precise soil monitoring can be achieved, with the collected data being displayed on a convenient user interface.

This dissertation proposes an IoT based soil monitoring solution built on geo-localized nodes that transmit their data to a user interface where data can be monitored. The design and validation of the system, including the analysis of experimental data is included in the dissertation.

KEYWORDS: *Internet of Things*, *precision agriculture*, *LoRaWAN*, *MQTT*, *NPK*

Contents

List of Figures

List of Tables

List of Acronyms

AES: Advanced Encryption Standard

AFH: Adaptive Frequency Hopping

API: Application Programming Interface

AWS: Amazon Web Services

BLE: Bluetooth Low Energy

CRC: Cyclic redundancy check

GPIO: General-purpose Input/Output

GSM: Global System for Mobile Communications

IoT: Internet of Things

JSON: JavaScript Object Notation

LED: Light-Emitting Diode

LMIC: LoraWAN-MAC-in-C

LPWAN: Low-power Wide-area Network

MOSFET: Metal-oxide-semiconductor field-effect transistor

NPK: Nitrogen, Phosphorus and Potassium

OTAA: Over-the-Air-Activation

TDR: Time Domain Reflectometry

TTN: The Things Network

UI: User Interface

WPAN: Wireless Personal Area Network

CHAPTER 1

Introduction

1.1. Motivation and context

Farming has always been a core part of human society, be it nomadic tribe pastoralism or sedentary tribe crop cultivation. It is believed by researchers that the first evidence of humans farming crops goes as far back as 23000 years ago, to the shore of the Sea of Galilee in Israel where a small-scale cultivation of "proto-weeds" took place [1].

The world population has been steadily increasing since the dawn of humanity until the 1950s, when an unprecedented population boom started to happen due to the wide availability of public health measures in less developed regions [2], and in the span of 70 years the Earth has grown from 2.5 billion people to 8 billion [3]. As a consequence, humanity has had to increase its agricultural production drastically over the previously mentioned timespan, from 205 million tons to 1.2 billion tons [4]. Such a drastic increase wouldn't be possible without the means to monitor the soil and plants for various factors like humidity, temperature and macronutrients such as NPK and perform an optimal management of the soil characteritics for maximum efficiency and yield, especially nowadays with climate change and prolonged droughts being an ever increasing issue for farmers [5], [6]. One of the strategies employed to achieve improved production sustainability is called Precision Agriculture, in which the previously mentioned factors are monitored and reacted upon to increase yield and reduce waste.

The NPK macronutrients are extremely important for crop yield [7, Section 2.1.1.4]: Nitrogen (N) is crucial for the production of proteins, nucleic acids and chlorophyll which drives photosynthesis, Phosphorus (P) is necessary for root growth, formation of fruits and seeds and blooming of flowers, and Potassium (K) helps with moving water and nutrients in the plant. A deficiency of these elements can lead to stunted growth, reduced yields and necrosis [8].

The measurement of macronutrients concentration in soil is an expensive and/or time consuming endeavour: easy-to-use solution-based soil tests have limited uses and thus require continous buying and sample collection, laboratory tests are both time consuming due to soil sample gathering and expensive as reliable portable laboratories are not cheap, finally standalone capacitive sensors are sold without the required additional equipment needed to make them work.

With the previously mentioned factors in mind, this disseration will focus on an easy-to-use IoT ecosystem prototype designed for soil NPK concentration monitoring.

1.2. Research questions

The following list contains the research questions that motivate this dissertation project:

- How does IoT benefit agriculture?

- Which are the effective methods to perform in-situ NPK measurements?
- What is the best wireless communication protocol for an ecosystem deployed in farming fields?
- How to determine the best sensor deployment layout?

1.3. Goals

The main goal of this disseration is to develop an IoT ecosystem which performs soil characteristic measurments with standalone nodes, with a focus on NPK macronutrients concentration measurement capabilities. Included in the main goal is the choice of an appropriate microcontroller and wireless communication protocols. As these nodes may run on battery power, some battery optimizations through both software and hardware methods will be discussed.

Aditionally, for the purpose of data visualization, a user interface component, accessible via many device types, with the following capabilities will be developed: displaying the geographic position of each node on a map, transmitted data storage, in order to be able to consult historical data, and graphical representation to track NPK concentration evolution.

CHAPTER 2

State of the art

The following information was compiled and summarized with the purpose of introducing relevant information and related works that exist at the time of writing.

This chapter will explore some of the applications of IoT in agriculture while also highliting the benefits in Section 2.1, then in Section 2.2 NPK measurement methods will be discussed by evaluating their accessibility, accuracy, time consumption and cost effectiveness, Section 2.3 will introduce and compare some of the available board brands, Section 2.4 will introduce a number of communication protocols geared towards IoT, Section 2.5 will discuss some of the currently available user interface solutions for IoT projects, Section 2.6 will talk about the distribution of NPK in soil, amd finally Section 2.7 will introduce works that motivate this dissertation.

2.1. IoT in Agriculture

Ever since its inception back in 1999 by Kevin Ashton for inventory tracking [9], IoT expanded beyond its initial scope and now is used everywhere in our lives [10], [11], one of the notable sectors is agriculture where it is used, for example, for crop monitoring, livestock tracking, enviromental monitoring and automated irrigation.

Certain companies have emerged that specialize in providing the previously mentioned applications as services, like ChrysaLabs [12] who provide solutions to measure up to 35 soil parameters, however manually, with real time analysis or GeoPard Agriculture [13] who provide, for example, crop monitoring via satelite imagery and macronutrients monitoring in soil. As stated by these companies, by employing IoT solutions for soil and crop monitoring farmers can, in turn, increase yield, reduce the utilization of fertilizers and increase sustainability. Although, the expense incurred by employing the products of these companies may be unjustified for small to medium scale farms, these entities could, as an alternative, set up their own monitoring ecosystems by utilizing easily acquirable sensors and microcontrollers.

2.2. NPK concentration level monitoring in soil

When it comes to testing soil for NPK three methods stand out: chemical tests, portable laboratories, which are also chemical tests however more advanced and testing via soil sensors and finally sensor-based solutions.

Chemical test have the benefit of ease of use and only requiring the purchase of widely available kits [14], however these kits have limited uses and require manual sample collection. As macronutrients are not distributed equally [15], [16] many tests may be required, this scales with the size of the measured area, that in turn not only decreases time efficiency, but also increases the overall cost as the reagents used to test the soil get used up quicker, requiring the

purchase of new ones. These kits also do not provide exactly values as results instead providing concentrations in relative values as can be seen in Figure 2.1.

FIGURE 2.1. A chemical test kit with its components [14].

When it comes to the question of how these tests work, an example could be [17]:

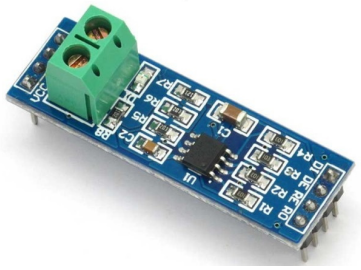
- N - Zinc reduces Nitrate to Nitrite and Chromotropic Acid reacts proportionally to the amount of Nitrite to form a red color
- P - Ammonium Molybdate reacts with Phosphorus to form a phospho-molybdate complex, it is then reduced to a blue complex by Ascorbic Acid the color of which is proportional to the amount of Phosphorus in the extract
- K - Sodium Tetraphenylboron reacts with Potassium to form a white precipitate in proportion to the Potassium concentration

Portable laboratories (Figure 2.2) used to test soil for macronutrient concentrations have a steep initial investment, divided between the machine, the accessories and chemicals required for the reactions [18]. Just like chemical kits, these laboratories require replacement reagents once the initally bought ones run out and samples have to be manually collected, thus making easy continuous testing impossible in both cases. However, unlike chemical tests, these devices provide exact numbers for the measured macronutrients in various units, like kg/ha (kilograms per hectare) or mg/kg (milligrams per kilogram).

FIGURE 2.2. A portable NPK testing laboratory [18].

Time Domain Reflectometry (TDR) sensors (Figure 2.3a) use parallel rods as transmission lines, voltage is applied to the rods which is then reflected back by the soil to the sensor, by analyzing the reflected waveform (magnitude, duration and shape) the sensors can determine, among many things, macronutrient concentrations. They measure these in mg/kg and, generally, are sold without the hardware that is needed to make use of their capabilities, thus requiring further purchases in the form of a microcontroller, to send and receive data, and, generally, a RS-485 to TTL module (Figure 2.3b) to enable communication between the two over the RS-485 standard.

(A) Capacitive NPK sensor [19]



(B) RS-485 to TTL module [20]

FIGURE 2.3. Soil sensor and communications module

These sensors are cost efficient as the components need to be purchased only once, additionally, when combined, the previously mentioned components can be turned into nodes spread across fields that communicate their readings, thus making concentrations readily available at any moment without having to collect and test a number of soil samples manually, allowing for continuous monitoring. This way of collecting data also has the added benefit of being easily expandable with other sensors. The downside of this solution is the requirement of having an understanding of how to assemble circruits and work with microcontrollers.

Aside from these 3 common methods, NPK concentration in soil can also be measured with optical sensors [21], these work by applying the Beer-Lambert Law which establishes a relationship between light absorbance (A) and the concentration (C) of the measured element in the sample: $A = \log \frac{I_0}{I_t}$ where I_t is the intensity of the transmitted light and I_0 is the intensity of the incident light, and $C = \frac{A}{\varepsilon \ell}$ where ε is the molar absorptivity and ℓ is the optical path length. The downside of this method is having to constantly compensate for the light conditions.

2.3. Microcontrollers and Boards

Microcontrollers are characterized as small low-power computers on an integrated circuit containing a processing unit, memory, various peripherals, like programmable General-purpose Input/Output (GPIO) [22], and may also have wireless capabilities like, and not limited to, Wi-Fi or Bluetooth [23], they are designed for embedded aplications thus making them well-suited for IoT applications. These microcontrollers are then embedded into development boards for accessibility and ease of use. These boards, in some cases, provide some basic functionality like, commonly, a Light-Emitting Diode (LED) as can be seen in Figure 2.4.

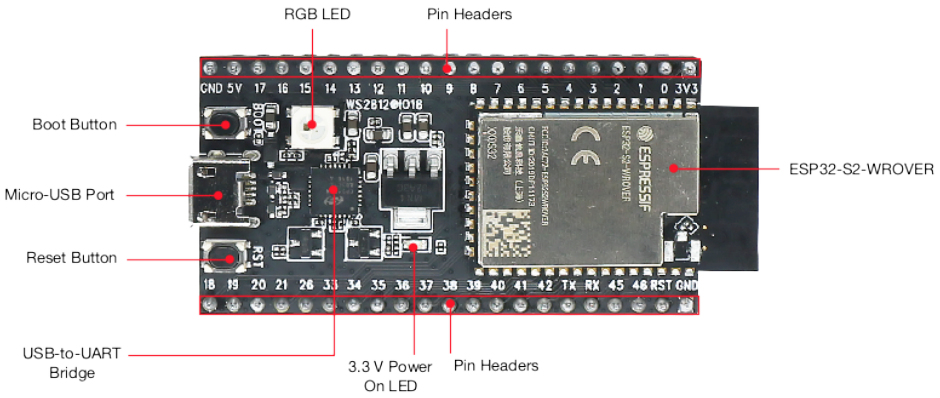


FIGURE 2.4. ESP32-S2-Saola-1 board with components description [24].

There are many companies specialized in producing development boards, notable examples include Espressif Systems (ESP board series), Arduino and Raspberry Pi. The products available from the aforementioned brands, generally, provide similar or the same functionality. However Arduino boards are, in most cases, the most expensive and may include extra built-in sensors, which may not be needed, or lack funcionality in some instances, thus when seeking basic functionality and cost efficiency for an IoT project it is better to acquire boards from the other brands. For instance, the cheapest Arduino board, the Arduino Nano Every, lacks Wi-Fi while, at time of writing, costing 12.50 euros [25] while a product with the equivalent form factor from Raspberry Pi with Wi-Fi and better specifications, the Raspberry Pi Pico W, costs, at the time of writing, 7.49 euros [26]; power consumption between these boards cannot be compared as Arduino Nano Every's datasheet does not state these numbers [27, Section 2.2]. Espressif Systems provide a wide variety of microcontrollers, what they call modules, and development boards [28] depending on the requirements: wireless connectivity, support for IoT protocols, like ZigBee, operating temperature, peripherals, memory and processing power [29].

Keeping the microcontroller awake between each operating cycle (data acquisition, processing and transmission for example) comes at a cost in terms of autonomy, thus the need arises for special modes of operation where power consumption is reduced to the bare minimum. This need is met by power modes, which offer diferent levels of power consumption depending on which components remain enabled during the elapsed sleep time. As a pratical example, the ESP microncontrollers allow for power consumptions as low as 10 μA during deep sleep [30, Table 3-2], this power mode also has the added benefit of limited data storage between each sleep cycle.

2.4. Wireless Communication in IoT

IoT is characterized, among many things, by the wide choice of communcation protocols supporting a variety of ranges, topologies, transmission speeds, frequency bands and power consumptions. Some examples, that will be discussed, include: ZigBee (2005), Z-Wave (1999), Bluetooth Low Energy (BLE) (2006/2009), Sigfox 0G (2010) and LoRaWAN (2014).

ZigBee is a Wireless Personal Area Network (WPAN) [31, Section 1.3] with ranges spanning from 10 to 100 meters [31, Figure 1.1], under perfect conditions, and is able to run on battery power for years [31, Section 1.1]. Three topologies are supported, namely Star, Mesh and Tree [31, Figure 1.6-1.8] and a typical network is comprised of three types of devices [31, Section 1.8]: end devices which gather and send data and spend most of their lifetime in a sleeping state, routers which relay messages in the network and a coordinator which manages the network. It supports 3 frequency bands, namely 868 MHz in Europe with 1 channel, 915 MHz in the USA and Australia with 10 channels and 2.4 GHz virtually worldwide with 16 channels [31, Table 3.1], each allowing, respectively, maximum data rates of 20, 40 and 250 kbit/s [31, Table 1.1]. A 2.4 GHz ZigBee network may get interference from or interfere with Wi-Fi unless the channels are configured so as to not overlap, as can be seen in Figure 2.5. Transmitted data is encrypted using Advanced Encryption Standard (AES), a symmetric-key algorithm with a 128-bit key shared by all devices [32, AnnexB B.1.1].

FIGURE 2.5. Possible channel configuration to avoid interference [33].

Z-Wave has two specifications/generations: Z-Wave with ranges of up to 100 meters [34] and on-battery autonomy of up to 1.5 years [35] and Z-Wave Long Range with ranges up to and beyond 1000 meters with up to a decade of on-battery autonomy [34]. The first uses the mesh topology [34], an example of which can be seen in Figure 2.6, and the latter uses a combination of mesh and star where the devices are connected via a mesh to the controller and it, in turn, transmits the received data to a gateway that serves multiple coordinators, forming a star [36].

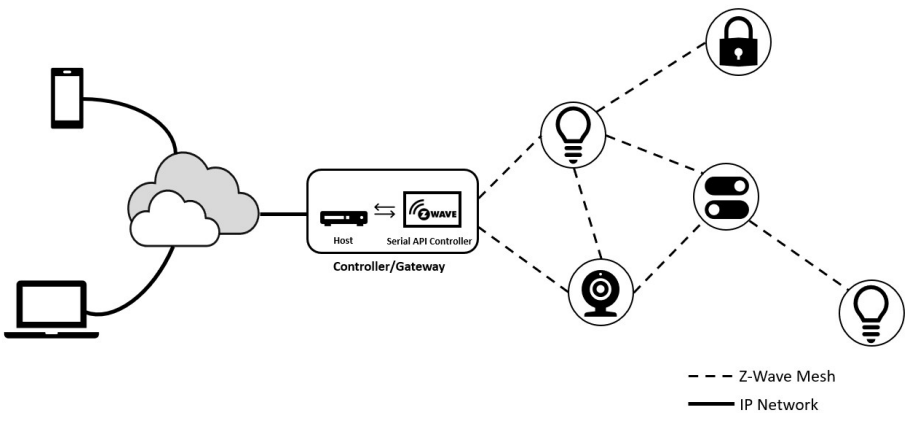


FIGURE 2.6. Possible Z-Wave Network configuration [37].

A given Z-Wave network contains a controller and end nodes, the controller is responsible for setting up and managing the network while end nodes transmit and relay data [38]. Communications occur on sub-1GHz frequencies, namely between 865.2 MHz and 926.3 MHz, depending on the region/country [39], with data rates of 9.6, 40 and 100 kbit/s [37], making Z-Wave impervious to interference from Wi-Fi. Transmitted data is encrypted using AES with 128 bit keys shared by all devices [38, Section 4.2.6.4.3].

BLE was designed for very low power consumption [40], with a variable range that depends on the environment and the specifications/hardware of the devices used, but the effective range can be anywhere between less than a meter and more than a kilometer [41]. It supports three topologies, namely point-to-point, broadcast and mesh (Figure 2.7).

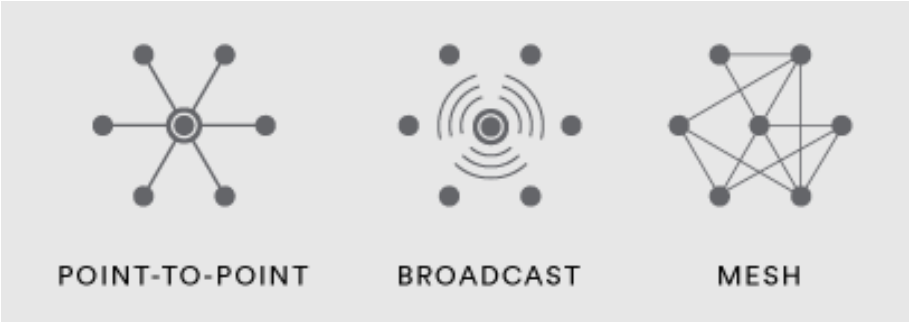


FIGURE 2.7. Bluetooth Low Energy Toplogies[40]

BLE operates in the 2.4GHz band with 40 channels, 3 of which are reserved for adveritising, and supports data transmission rates of 125, 500 kbit/s, 1 and 2 Mbit/s. While it operates in the same band as Wi-Fi, BLE employs Adaptive Frequency Hopping (AFH) to hop to an available channel determined by an algorithm [42] to avoid interference. The encryption algorithm depends on the type of device pairing [43, Section 5.1].

Sigfox 0G is a Low-power Wide-area Network (LPWAN) [44] with a range of up to 10 kilometers in an urban setting and up to 40 kilometers in rural setting [45] and is able to run on battery power for years [44]. The network employs the star topology to connect devices to base stations, however a device is not tied to one base station and messages are on average received by three stations to guarantee high quality of service [46, Section 3.3]. It operates between 862 MHz and 928 MHz depending on the region/country and with transmission rates between 100 and 600 kbit/s [47]. There is not encryption by default, however it can be enable by using an existing solution or by implementing a custom one [46, Section 5.1.3]. It is also important to note that using Sigfox 0G incurs a network utilization fee that varies per country [48] and daily messages are limited to 140 uplinks and 4 downlinks per day [49].

LoRaWAN is also an LPWAN [50] with a range lower range of up to 3 kilometers in urban environments and up to 10 kilometers in rural ones, and is able to run on battery power for up to 10 years [51]. It uses a star-of-stars topology where end devices communicate to gateways and the gateways relay the data to a central server and vice-versa. Communications occur in sub-1GHz frequencies between 486.3MHz and 927.5MHz depending on the country/region [52] with data

rates between the 0.3 and 50 kbit/s [50]. End devices are categorized into classes depending on the downlink communication capabilities and, as a consequence, power consumption [50]:

- Class A devices consume the lowest amount of power allowing them to run on batteries, however they can only be reached after an uplink communication for 2 short downlink windows, meaning the device must first wake up and send data to be reachable by the network server, meaning server-to-device communications must be buffered.
- Class B devices share Class A's downlink windows but are also synchronized to the network server allowing for scheduled downlink ping slots, providing the ability to send downlink communications every defined period of time, the additional power consumption is low enough to still be viable for battery power.
- Class C devices can receive downlink communications at any time, not having to wait for an uplink message or for a ping slot, however this comes at the expense of considerably more power consumption meaning these devices are only suited for cases where a continuous source of power is available. Battery powered devices are able to switch to Class C for tasks like firmware updates over the air.

Encryption is achieved using AES with 128-bit keys [50]. The most popular network server is The Things Network (TTN), a free and open global network, fair usage policies apply per device with a cap on daily messages in both directions, with a maximum cumulative uplink airtime of 30 seconds and 10 downlink messages per 24 hours [53]. As an example, if a message takes 41 ms of airtime then it can be sent $\lfloor \frac{30}{0.041} \rfloor = 731$ times per day.

While not strictly wireless it is also important to mention MQTT, a topic based publish-subscribe messaging protocol that runs over the internet. An MQTT deployment is made out of two types of devices: MQTT Clients which may both publish data to topics and subscribe to them in order to receive data, and the MQTT Broker which receives published data and routes it to the subscribers [54]. For instance, a TTN application can make use of MQTT to further transmit the data received from LoRa devices [55].

Table 2.1 provides a general overview of each presented protocol for an easier comparison between each.

	ZigBee	Z-Wave	BLE	SigFox 0G	LoRaWAN
Range	10-100m	Up to 100m	Depends	Up to 40km	Up to 10km
Battery	Years	Years	Years	Years	Years
Topologies	Star, Mesh, Tree	Mesh	Point-to-point, Broadcast, Mesh	Star	Star-of-stars
Frequencies	Sub-1GHz, 2.4GHz	Sub-1GHz	2.4GHz	Sub-1GHz	Sub-1GHz
Data Rate	20-250 kbit/s	9.6-40 kbit/s	125 kbit/s - 2Mbit/s	100-600 kbit/s	0.3-50 kbit/s
Encryption	AES with 128 bit key	AES with 128 bit key	Depends	User defined	AES with 128 bit key
Network usage cost	None	None	None	Depends	Depends*
Message limit (per day)	None	None	None	140 uplinks, 4 downlinks	Depends

*TTN offers a free tier

TABLE 2.1. General overview of each presented protocol.

2.5. Data visualization in IoT

There are various plataforms that support data visualization and may also offer IoT protocol bridges, some examples include openHAB, Node-RED, Grafana and ThingSpeak. They all offer distinct dashboard solutions and ways to build them.

Although, the main use for openHAB is smart house automation, it can be used for general IoT applications. It offers three ways to visualize data and control devices: MainUI, HABPanel and BasicUI. MainUI is, as the name may suggest, the main UI solution, offering a modern look, HABPanel is block based and particularly suited for tablets and BasicUI is based on Material Design Lite from Google and useful for simple functional UIs, an example of which can be found in Figure 2.8 which shows a simple segmented dashboard for a smart house.

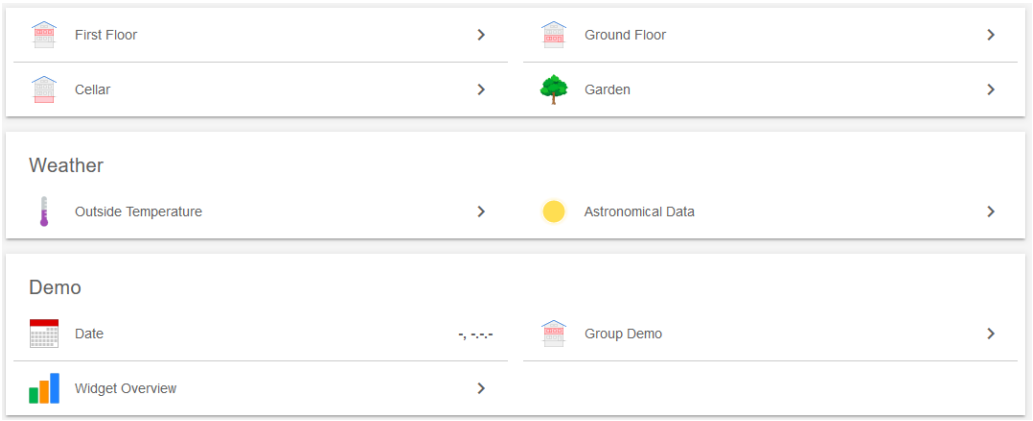


FIGURE 2.8. openHAB BasicUI demo [56].

openHAB has a rich plugin ecosystem where plugins for various protocols and technologies can be found, for example, weather acquisition via OpenWeatherMap. The platform supports many ways of scripting, one of which being rule-based scripting [57]. Letting users define actions based on rules which monitor events transmitted from connected devices, allowing, for example, the creation of an intrusion alarm based on motion and door state sensors. One downside of openHAB is that since the wiki is not regularly updated, it may contain incomplete or outdated data.

Node-RED [58], while not stricly built for IoT or data visualization, can be used for such [59] due to its capability to wire together hardware, Application Programming Interfaces (APIs) and online services using low-code programming, as in, mostly, without writing code. It offers a dashboard solution similar to openHAB's BasicUI via a plugin [60] and is programmed by using flows made out of nodes that are wired together. Allowing data to be wired to visualization nodes and action nodes, like buttons, to be wired to actions as can be seen in Figure 2.9. If more complex logic is required, Node-RED offers function nodes where JavaScript code can be written. Just like openHAB, support for various protocols and extra features is obtained via the rich plugin ecosystem.

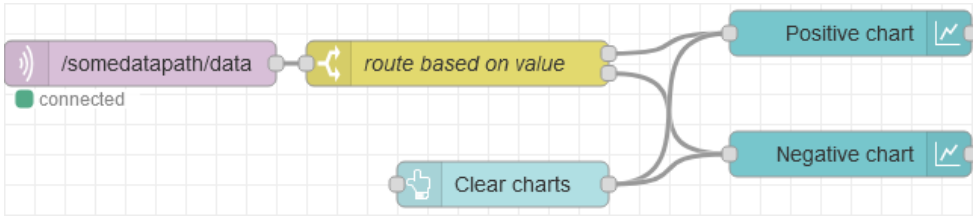


FIGURE 2.9. A basic flow that feeds data from MQTT to the dashboard.

Grafana [61] is purely a data visualization platform best used when integrated with other plataforms or tools to produce graphs, for instance: a Node-RED instance could receive, process and store data in a database while a Grafana instance could pull this data and visualize it requiring no code to be written in the process, aside from database queries.

FIGURE 2.10. Example Grafana dashboard with various elements.

ThingSpeak [62] characterizes itself as a cloud-based IoT analytics platform with the capabilities of data aggregation, visualization and analysis. Providing, among many things, libraries for mainstream development boards to easily communicate with the cloud, data sharing, integrations with other apps and support for protocols like LoRaWAN. However, whereas the previous platforms are free and self-hostable, ThingSpeak runs on a freemium model where the service can be used free of charge with diminished or locked features, and to unlock the full capabilities a yearly fee must be paid, the price of which depends on how the service is being used [63]. It is important to note that data preprocessing, visualization and analysis is done using the MATLAB language. Figure 2.11 shows a basic example of ThingSpeak's data displaying capabilities taken from a public channel.

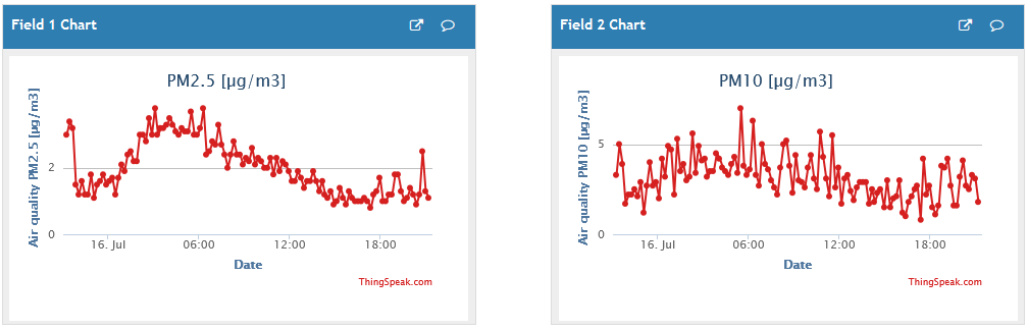


FIGURE 2.11. Example ThingSpeak UI elements [64].

2.6. NPK distribution

NPK concentrations are not distributed linearly or equally, instead a given macronutrient can have zones where it is abundant and zones where it is lacking as can be seen in Figure 2.12 which shows nitrogen distribution in a given plot of soil.

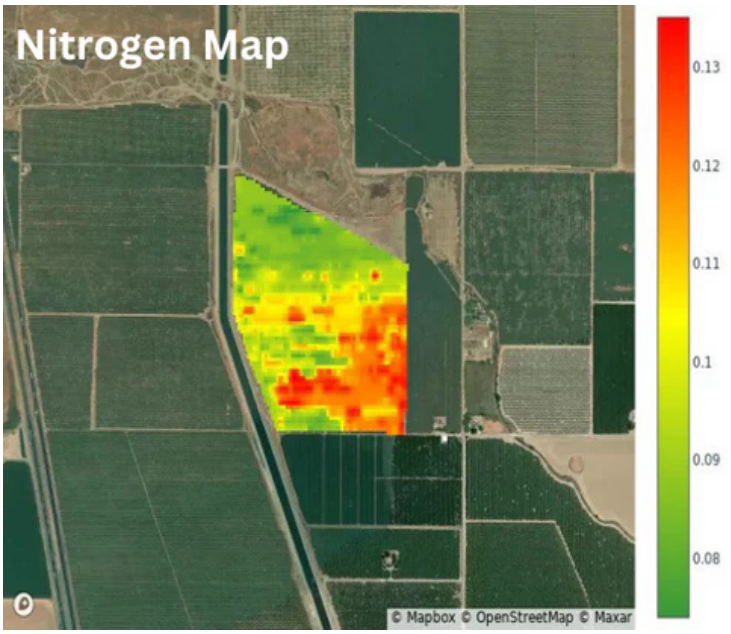


FIGURE 2.12. Nitrogen concentration variation graph [65].

These concentration zones are also not the same between the many macronutrients, a zone with a high concentration of one macronutrient may have a low concentration of another as can be seen in Figure 2.13 which compares macronutrient concentration distributions on a larger scale.

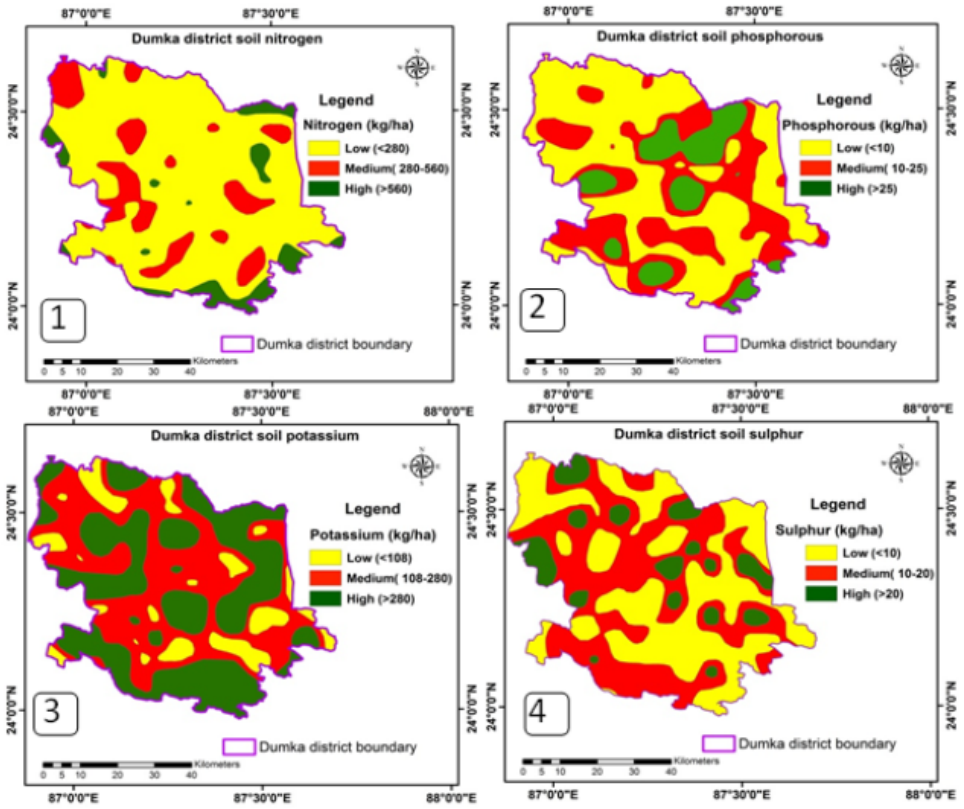


FIGURE 2.13. Spatial concentration variation graphs for NPK and sulphur [66].

The easiest way to perform a distribution study is with the chemical tests mentioned in the beginning of Section 2.2 as just relative results are enough to reach a conclusion regarding distributions. This in turn also answers the question regarding the effective methods to perform in-situ NPK measurements, TDR sensors are very effective for continuous monitoring as they are easy to use, can be automated, output numeric values and are low cost when compared to the other methods of getting numeric values, like portable laboratories, whereas chemical tests via kits are effective for distribution studies where relative results can provide a quick answer to how to distribute the nodes in space. Although, it is possible to do a distribution study with a TDR sensor, one must pay attention to the moisture of soil.

2.7. NPK IoT reported solutions

The combination of IoT and NPK sensors to quantify, transmit and display soil characteristics is a relatively new field in academia.

This dissertation draws inspiration from Postolache et. al [67] who proposed and implemented a system to monitor NPK and moisture in garden soil, however communications occur over Wi-Fi and thus are limited in range, data can only be accessed via a mobile application and the available data visualization options are rather limited, not allow tracking the evolution of data over time with, for example, charts or by accessing previous data.

Cheruvu et. al. [68] implements data charting, however it is limited to a number of points, not allowing the plotting of previous data beyond the points currently being displayed on the chart, additionally communications occur over Wi-Fi and the data is stored and displayed on a

proprietary platform, meaning the user doesn't have full control and is limited to the capabilities of the platform, finally there's no mention of running multiple nodes.

Aliparo et. al [69], similarly to the previous cases, uses Wi-Fi for data transmission and doesn't mention running multiple nodes at once.

Pyingkodi et. al. [70] explicitly mentions the use of AC mains for power and Global System for Mobile Communications (GSM) for data transmissions, neither being ideal for a farming solution due to the limited scope of deployment and data cost.

Sundari et. al. [71] does not implement a graphical user interface, instead displaying the data on an LCD screen mounted on the nodes, meaning there's no easy to way to monitor readings as every screen would need to be checked individually every time.

Madhumath et. al. [72] uses Wi-Fi for communications and the data visualization application is mobile-only while only allowing simple numeric visualization for only one node, additionally Amazon Web Services (AWS) is used which translates to extra monthly costs.

CHAPTER 3

System Architecture

In order to develop this system a typical IoT architecture was used, divided into hardware/firmware, communication and data visualization. This chapter will go over the development of this architecture in the context of this system with the following division: Section 3.1 will go over and justify the chosen hardware while also providing some alternatives, Section 3.2 will go over the the chosen communication procotols, Section 3.3 will give an overview of the firmware while also explaining further the most important parts of our solution and finally Section 3.4 will give an overview of the implemented graphical interface.

As a general overview of what comes in the next sections, all the elements above (hardware, firmware, commmunication protocols and visualization platform) combine to form the flowchart found in Figure 3.1 which shows how everything is connected and how the data flows between the elements.

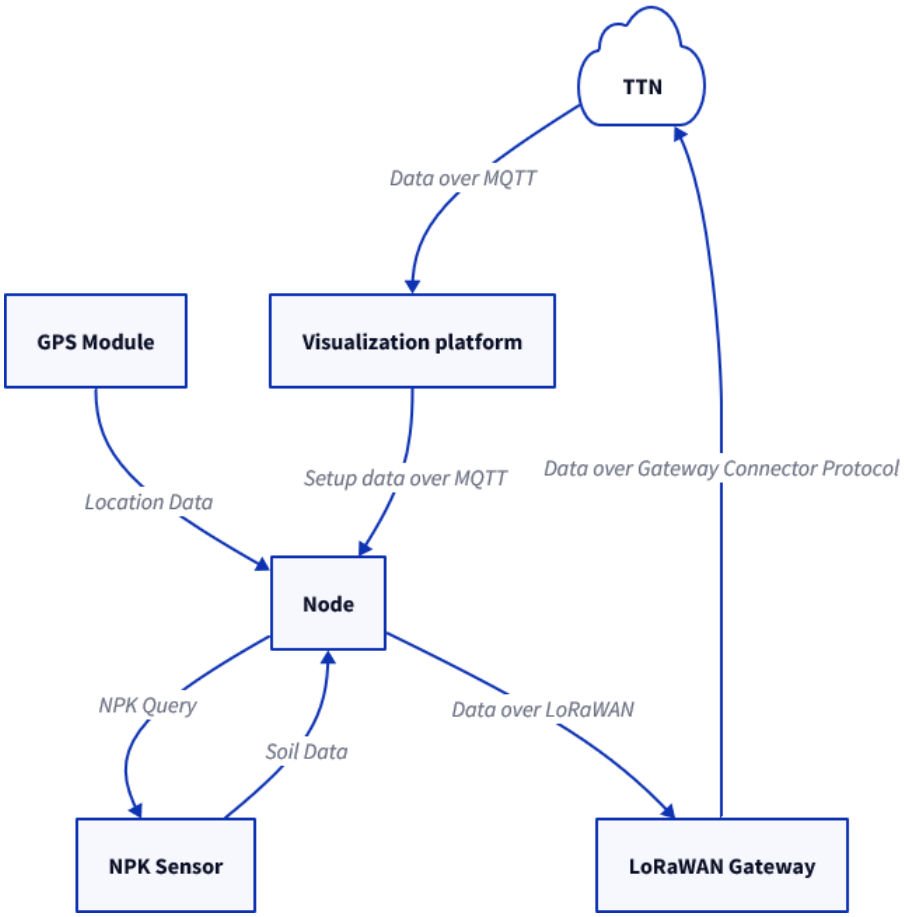


FIGURE 3.1. System overview flowchart.

3.1. Hardware

There is a number of mandatory components required to build a working measuring node, these include: a development board with wireless capabilities, a NPK sensor, an UART TTL to RS485 module, a step-up module, a GPS module, a power relay module, a battery and a booster for it (although these 2 can be easily replaced).

Figure 3.2 represents the wiring diagram when using the components mentioned above. The power lines are colored red, ground lines black and data lines green for readability.

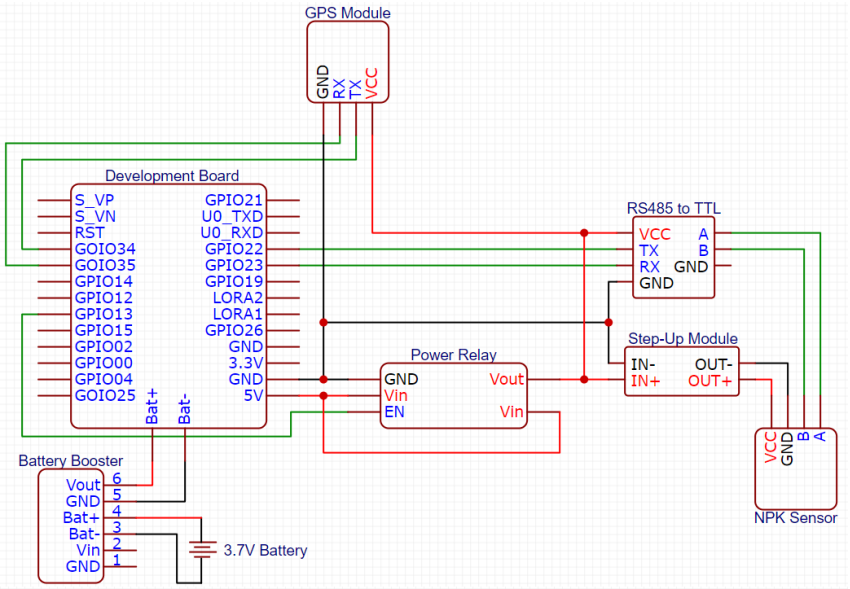


FIGURE 3.2. Wiring diagram for mentioned components.

The following hardware choices were made regarding the components found in the diagram:

- The LilyGO TTGO T3 LoRa32 (Figure 3.3) development board was chosen due to its various integrated peripherals which include an OLED screen, LoRa antenna ports, the SX1276 LoRa chip and a dedicated port and cable for battery power. Another important factor is that the ESP32-PICO-D4 used on the board provides very low power consumption in deep sleep, as mentioned previously.

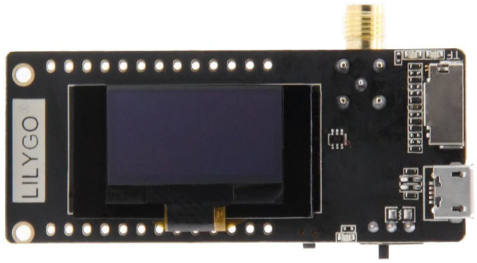


FIGURE 3.3. LilyGO TTGO T3 LoRa32.

- The 7-in-1 JXBS-3001-TR-RS (Figure 3.4) TDR sensor was chosen as the NPK sensor due to the robustness of data that it can collect, not only NPK but also soil humidity, temperature, pH and conductivity, meaning the expansion of collected data is a trivial

task. As stated previously, these sensors use RS485 to communicate and receive commands. It is also important to mention that it is rated for 12 volts and has an accuracy of ±2% for NPK measurements.

FIGURE 3.4. JXBS-3001-TR-RS 7-in-1 sensor.

- As the NPK sensor uses the RS485 standard, a RS485 to TTL conversion module is required. There are two types of UART TTL to RS485 Two-way Converters, with and without a CD4069BM chip (Figure 3.5, chip on the left). A converter with the chip was chosen due to general convenience of implementation in both hardware and firmware.

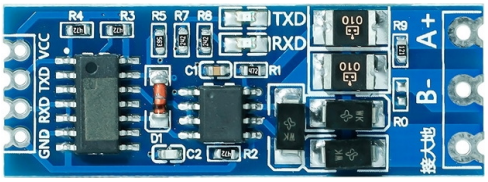


FIGURE 3.5. UART TTL to RS485 Two-way Converter.

- A generic XL6009E1-based step-up module (Figure 3.6) was chosen to get the recommended 12 volts for the NPK sensor.

FIGURE 3.6. XL6009E1 Step-Up Voltage Regulator.

- When it came to acquiring the geographic position, the choice fell onto the Air530 statelite positioning module (Figure 3.7) due to its relatively low positioning error of 10 meters and support for all the major satellite positioning systems, namely GPS, GALIEO, GLONASS and BEIDOU.

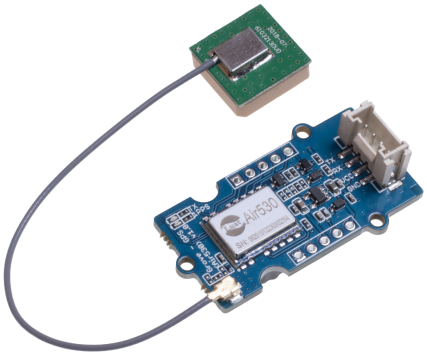


FIGURE 3.7. Air530 satellite positioning module.

- A SLA-05VDC-SL-A RobotDyn DC 5V (Figure 3.8) was chosen due to its activation curves, during deep sleep all GPIO pins on a given board are set to low and this relay happens to cut power while its enable pin receives a low signal. This module is necessary in order to prevent the sensors from constantly using current.

FIGURE 3.8. SLA-05VDC-SL-A RobotDyn DC 5V.

- As batteries are generally rated under 5 volts, they needs to be boosted in order to prevent development board brownouts (these happen when voltage drops below a certain threshold and the board shuts itself down). For this purpose, a DFR1026 (Figure 3.9) was chosen as it can boost its input to 5 volts, it also has a low quiescent current while being used of 50 μA. Additionally, it provides a charge circuit out of the box.

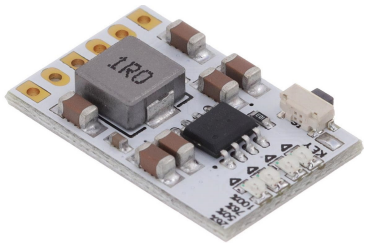


FIGURE 3.9. DFRobot DFR1026.

Some hardware alternatives include the Heltec WiFi LoRa 32(V3) for the development board or the NEO-6M for the GPS module. Other alternatives include a different type of RS485 to TTL converter (without a CD4069BM chip), a different voltage regulator, power relay module or any other battery booster. In both cases, alterations in the firmware or circuit may be required. Finally, as an alternative to the battery and its booster, a powerbank could be be used instead as it is easy to integrate, due to the USB port on nearly all development boards, and recharge.

There are a number of strategies that can be employed to maximize battery lifespan on a battery powered system, some via firmware and some via hardware. In the case of hardware, the most direct and simple solution is simply cutting power to the sensors when they are not necessary anymore. This strategy can be achieved via a power relay module, preferably the normally open type, which can be used to cut power to the power lines used by the sensors once all the necessary data has been acquired. An extra optimization would be removing the LEDs from both the relay module and the battery booster as they are always active.

The very first iteration of the node was made on a most commonly available type of breadboard, however it was quickly abandoned as it was causing a lot of power transmission problems, especially for the voltage regulator which was outputting values beyond its specification at, sometimes, 80 volts while being configured for 12 volts and rated for a maximum of 35 volts.

Figure 3.10 represents the second prototype of assembled node, while it worked it had a very big disadvantage in that everything on the board itself was soldered, meaning that switching hardware was difficult. This issue first manifested itself when, due to a short circuit, both the RS485 to TTL module and the voltage regulator broke down, replacing them proved to be non trivial with just a soldering iron. Another issue, and related to the previous one, was that eventually causing a short circuit after soldering was unavoidable due to the proximity of the holes and metalic coating on both sides of the board.

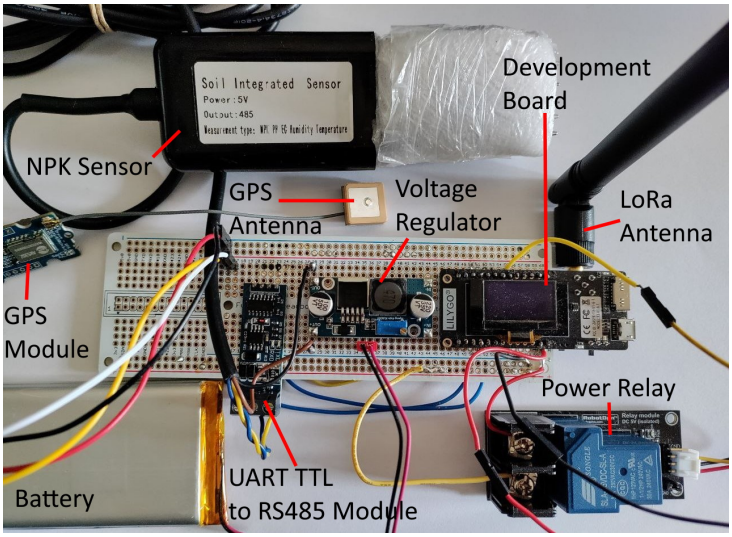


FIGURE 3.10. Second node iteration using a aolder-able breadboard.

The third and final iteration of the board, found in Figure 3.11 solves both the components switching problem and the short circuit problem. As for the former, all the hardware elements are mounted into their own custom female headers and the wires use a combination of both male and female headers for connections, on a case by case basis. As for the latter, the metalic coating is only applied undernear and it is thin enough where it can be easily cut in order to prevent any possible shorts or interference. This iteration also represents Figure 3.2 diagram in hardware form.

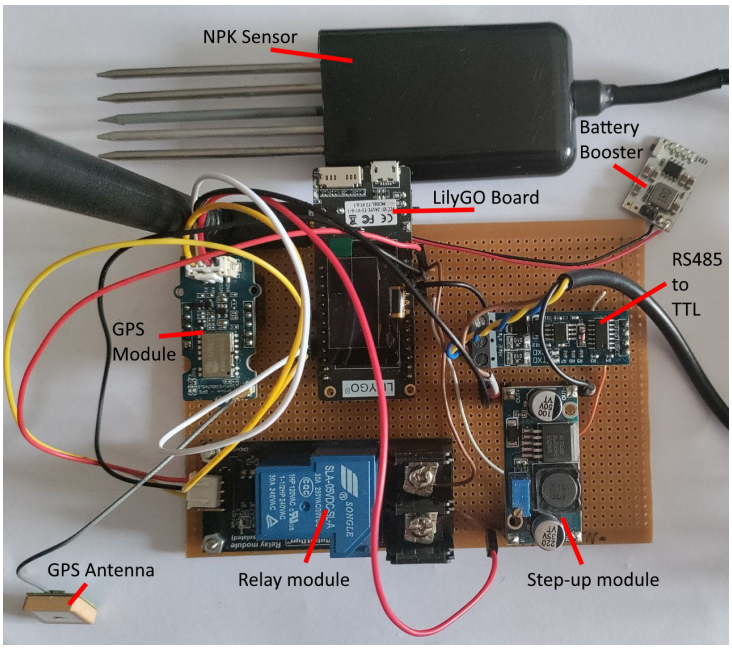


FIGURE 3.11. Final iteration of the custom hardware board, based on Figure 3.2.

3.2. Communication

There are 2 modes of operation for any given node: the setup mode and the data transmission mode. During setup, each node receives its own individual LoRaWAN keys and new identifier, this mode requires a fast short range protocol that can easily route data, with say, topics, to specific nodes, while the data transmission mode needs a protocol with a low power consumption and a relatively long range so as to be applicable in many environments. Taking the previous factors and Section 2.4 into consideration, the following choices were made:

- MQTT (over Wi-Fi) as the setup mode protocol due to its publish/subscribe architecture with data topics, being ideal for sending specific data targeted at a specific node. Each node listens to its own custom topic which follows the following format `/{wifi hostname}/lora` where `{wifi hostname}` is unique to each node.
- LoRaWAN as the protocol for the data transmission mode due to its low power consumption and long range. In addition to the previously mentioned factors, the TTN platform provides a free usage plan for this protocol and a MQTT intergration to redirect data to its real destination, for example, a user interface.

In order to send data, 2 types of payload formats are employed depending on the communications protocol. MQTT payloads have a theorectical maximum size of 256 megabytes meaning payload size is of no real concern, for this reason payloads sent over this protocol are formatted using JavaScript Object Notation (JSON), an example of which can be seen in Figure 3.18. LoRaWAN payloads have a rather limited maximum length of 256 bytes which is further reduced by the various connection paraments like the spreading factor and the frequency, meaning payloads should be kept as short as possible while still being easily decodable. Due to these factors data transmission mode payloads are formatted as strings of comma separated values, an example of which can be seen in Figure 3.18. Upon arriving at the TTN servers, the LoRaWAN payloads are

formatted into JSON objects via the Payload Formatter feature, the formatter used can be seen in Figure 3.12, it is assumed that the LoRaWAN payloads are properly formatted by default. Once formatted into a JSON object, the payload is sent to the visualization platform over MQTT.

```
function decodeUplink(input) {
  let payload = String.fromCharCode(...input.bytes).split(',');
  return {
    data: {
      node: payload[0],
      n: parseInt(payload[1]),
      p: parseInt(payload[2]),
      k: parseInt(payload[3]),
      lat: parseFloat(payload[4]),
      lot: parseFloat(payload[5])
    },
    warnings: [],
    errors: []
  };
}
```

FIGURE 3.12. TTN uplink payload formatter.

As discussed previously in Section 2.4 LoRaWAN uses a star-of-stars topology where end devices first send their payloads to a gateway which in turn forwards them to a network server like TTN. After careful consideration, the LPS8N gateway (Figure 3.13) was chosen due to its easy setup and management via a browser-based user interface, accessible from any device and due to the fact that Smart Agriculture is one of the suggested applications by the manufacturer. It is important to mention that the gateway needs an internet connection to communicate with the network server.

FIGURE 3.13. The LPS8N LoRaWAN Gateway [73].

3.3. Firmware

This section will go over the code implementation of the two possible states of any given node, namely the setup and the data transmission state. Additionally, this section will also discuss some difficulties encountered and how they were solved.

As mentioned previously, during the setup the node will receive its LoRaWAN keys over MQTT. To achieve this a number of steps have to be completed beforehand, namely connecting to the Wi-Fi and then connecting to the MQTT broker of choice. As ESP32 boards sometimes have issues connecting to a Wi-Fi network, the connection status is checked once per second for five seconds. If no connection is achieved, the board will restart and begin the process anew

until a connection is established. Information regarding this process is shown to the user on the OLED screen.

Once a Wi-Fi connection is established, using predetermined credentials, the process can continue to the next step which is handling the MQTT connection with the broker of choice in this case being HiveMQ's Public Broker, although it is also possible to host one locally. Once connected, the board subscribes to its own topic, as mentioned previously `/{name}/lora` where `{name}` is the board's Wi-Fi hostname as it is unique enough to serve as a reliable identifier. Afterwards a heartbeat message is sent to the visualization platform under the topic `/ISCTE_SoilIoT/hello` with the payload containing the hostname to indicate that the board is ready to receive its keys and new indentifier. Upon receiving the keys and the new identifier, these are saved to a file, using the JSON format for convenience, the visualization platform is notified that the setup was successful and the board prompts for a restart, as can be seen in Figure 3.14. Once this process is over, the screen is not used again.

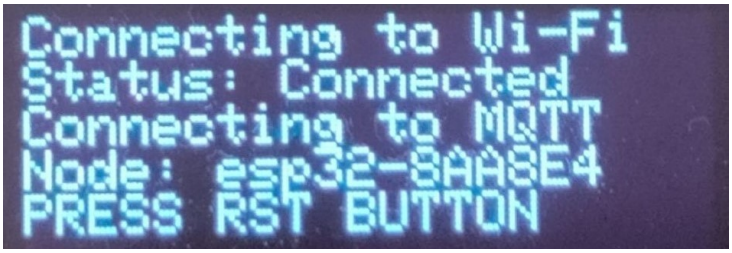


FIGURE 3.14. Board screen prompting for a restart, the various connection statuses can also be observed.

Once the setup process is over and the board is restarted, it checks for the presence of the previously mentioned file, namely `lora.json`, to determine whether it has been setup or not. If it has, just like before, a number of steps have to happen before a LoRaWAN payload can be sent, namely initilizating and polling the sensors, initializing the LoRa chip, implementing LoRaWAN in software and finally waiting for a transmission window before transmitting the payload. After initializing the NPK sensor and the GPS module, their polling functions are called, however in the case of the GPS module this only happens once as deep sleep lets us save variables between sleep cycles, meaning we only need to acquire the GPS coordinates once in the beginning, this is primarily done because the module requires a fair amount of time to warm up and acquire coordinates.

The NPK polling function will continuously poll the sensor, with an NPK query payload taken from the sensor's documentation, until a valid reading is achieved, this is necessary as sometimes the serial communications line may contain invalid responses due to external factors like a momentary bad contact. The sensor response payloads contain Cyclic redundancy check (CRC) bytes at the end, which let us verify whether the response has any errors. This can be achieved by manually calculating the CRC for the received payload (Figure 3.15) and then comparing the bytes.

```cpp
template<size_t N>
uint16_t crc(std::array<uint8_t, N> buf) {
  uint16_t crc = 0xFFFF;

  for (uint8_t pos = 0; pos < buf.size() - 2; pos++) { // last 2 are the crc bytes
    crc ^= buf[pos];                  // xor byte into least sig. byte of crc

    for (uint8_t i = 8; i != 0; i--) {// loop over each bit
      if ((crc & 0x0001) != 0) {      // if the LSB is set
        crc >>= 1;                    // shift right and xor 0xA001
        crc ^= 0xA001;
      } else {                        // else LSB is not set
        crc >>= 1;                    // just shift right
      }
    }
  }
  // this number has low and high bytes swapped, so use it accordingly (or swap bytes)
  return crc;
}
```

FIGURE 3.15. Modified CRC compute function courtesy of Continental Control Systems, LLC.

The NPK values are recalculated on each iteration and if they are all zero, the iteration is considered invalid even if the CRC bytes are valid. This is due to the fact that such a reading is generally an error which appears especially on the very first reading after the sensor is powered on. Due to the characterestics of the RS485 to TTL converter circuit, it is also important to set the data pins to 0 volts in order to prevent leakage onto the power line. The code for this process as a whole can be found in Figure 3.16, it also contains examples of what is a valid payload.

```
constexpr std::array<uint8_t, 8> NPK_PAYLOAD = {0x01,0x03,0x00,0x1E,0x00,0x03,0x65,0xCD };
uint16_t n = 0, p = 0, k = 0;
void handle_npk() {
  std::array<uint8_t, 11> answer = {};

  rs485.flush();
  uint16_t crc = 0xFFFF;
  // while crc doesn't match the calculated crc or the NPK values are invalid
  // {0x01,0x03,0x06,0x00,0x20,0x00,0x25,0x00,0x30,0xB1,0x6D} as {0xB1,0x6D} is valid
  // {0x01,0x03,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x21,0x75} is valid too, however the
  // NPK values are all 0, thus it is considered invalid
  while ((answer[9] != (crc & 0xFF) && answer[10] != (crc >> 8)) || (n == 0 && p == 0 && k ==
0)) {
    if (rs485.write(NPK_PAYLOAD.data(), NPK_PAYLOAD.size()) == NPK_PAYLOAD.size()) {
      delay(1000);
      rs485.flush();

      for (byte i = 0; i < 11; i++) { //the last 2 bytes contain the crc
        answer[i] = rs485.read();
        PRINT("%x ", answer[i]);
      }
      PRINT("\n");
      rs485.flush();
    }
    crc = crc(answer); // recalculate the crc

    // recalculate npk values
    n = (answer[3] << 8) | answer[4];
    p = (answer[5] << 8) | answer[6];
    k = (answer[7] << 8) | answer[8];
  }

  digitalWrite(NPK_TX, LOW);
  digitalWrite(NPK_RX, LOW);
}
```

FIGURE 3.16. NPK sensor handling code.

The GPS position polling function makes use of a library called `TinyGPS` to parse the NMEA data it receives, the format used by all GPS receivers, among other electronics. The data parsing continues until valid location data is acquired, this may take anywhere from 30 seconds to several minutes, which is why, as stated previously, it is only done once. Having acquired both NPK and location data, power is cut to the sensor via the power relay until these are necessary again in order to preserve battery.

There are two approaches to implementing LoRaWAN in firmware, using RadioLib [74] or using LMIC [75]. The initial implementation was done via `RadioLib` due to the amount of useful implementation abstractions, however connection and transmission problems began arising and so the choice to switch to LMIC was made. `LMICPP-Arduino` [76] was picked as the library of choice as it not only provides a port of LMIC from C to C++ but also band configurations for the european region out of the box.

A LoRAWAN transmitter implementation is generally divided into 3 parts: key acquisition, event handling and the transmission job itself, which also handles the initial Over-the-Air-Activation (OTAA) with the network server.

There are 3 necessary keys to establish a connect: the AppKey, the DevEUI and the AppEUI. At this point these keys are already saved on the board, from the setup process, in the `lora.json`

file, and only need to be passed to LMIC in the form of byte arrays, this is done by parsing the keys in string format into byte arrays.

In practice, the LMIC event handler only needs to handle the `EV_TXCOMPLETE` event which indicates that the payload has been sent after waiting for a transmission window. From a connection debugging perspective, it is more convenient to also implement the `EV_JOINING`, `EV_JOINED`, `EV_JOIN_FAILED`, `EV_RESET`, `EV_LINK_DEAD` and `EV_LINK_ALIVE` events which, respectively, signify that the node is joining the network, the node has joined the network successfully, the join request failed even after retrying, the connection is being reset due to a mismatch in the expected sequence counters (one of the problems encountered while using RadioLib), no confirmations have been received for an extended period time and thus transmissions might fail and the link is now considered alive as a confirmation was received (works in tandem with the previous event). Which results in the code in Figure 3.17.

```
void on_lmic_event(EventType ev) {
  switch (ev) {
    case EventType::JOINING:
      PRINT("EV_JOINING\n");
      break;
    case EventType::JOINED:
      PRINT("EV_JOINED\n");
      break;
    case EventType::JOIN_FAILED:
      PRINT("EV_JOIN_FAILED\n");
      break;
    case EventType::TXCOMPLETE: {
      PRINT("EV_TXCOMPLETE\n");
      if (lmic.getTxRxFlags().test(TxRxStatus::ACK)) {
        PRINT("Received ack\n");
      }

      // save before going to deep sleep.
      StoringBuffer store = StoringBuffer { save_state };
      lmic.saveState(store);
      save_state[300] = 51;

      // disable relay
      digitalWrite(RELAY_PIN, LOW);

      ESP.deepSleep(TX_INTERVAL.to_us());
      break;
    }
    case EventType::RESET:
      PRINT("EV_RESET\n");
      break;
    case EventType::LINK_DEAD:
      PRINT("EV_LINK_DEAD\n");
      break;
    case EventType::LINK_ALIVE:
      PRINT("EV_LINK_ALIVE\n");
      break;
    default:
      PRINT("Unknown event\n");
      break;
  }
}
```

FIGURE 3.17. LMIC event handling code.

Having setup the event handler, data payloads can now be sent. As mentioned previously LoRaWAN payloads are rather limited in size, but another important factor is that payload size

influences the airtime as bigger payloads take longer to transmit, meaning keeping the payload as small as possible is key to transmitting the most messages under TTN's fair usage policy. With this in mind, as mentioned previously, unlike MQTT payloads, LoRaWAN payloads do not use the JSON formatting instead being comma separated strings of values. For comparison, Figure 3.18 provides an example payload with the 2 possible formattings, by using the string of comma separated values 28 bytes can be saved.

```
json: {"n":"horta","n":255,"p":255,"k":255,"a":25.747852,"o":-11.1560066}
size: 67
comma: horta,255,255,255,25.747852,-11.1560066
size: 39
savings: 28
```

FIGURE 3.18. Payload size comparison.

With the payload assembled and marked by the identifier acquired during setup, for example, "horta" in Figure 3.18, the next transmission is scheduled for when a transmission window is available. The code for this process can be found in Figure 3.19.

```
void send_lorawan_payload() {
  std::string payload = (const char*)lora["NODE_NAME"]; payload += ",";
  payload += std::to_string(n); payload += ",";
  payload += std::to_string(p); payload += ",";
  payload += std::to_string(k); payload += ",";
  payload += std::to_string(lat); payload += ",";
  payload += std::to_string(lon);

  // Prepare upstream data transmission at the next possible time.
  lmic.setTxData2(10, (uint8_t*)payload.c_str(), strlen(payload.c_str()), false);
  next_send_time = os_getTime() + TX_INTERVAL;
}
```

FIGURE 3.19. LoRaWAN payload transmission handling code.

Having sent the payload, the board will prepare for its next data collection and transmission cycle by saving the connection state so it can restored later in order to avoid having to rejoin the network and going into deep sleep for a configurable amount of time, as can be seen in the event handler shown previously in Figure 3.17.

3.4. Visualization

The visualization platform was created using Node-RED due to its cross-platform nature, modularity, ease-of-use and rich extension ecosystem allowing for quick development of new or expansion of old features.

The interface is divided into 2 screens:

- the node setup screen, where data can be sent to nodes (the keys and the identifiers); this screen is suitable for phones, tables and computers
- the data visualization screen, where node data can be seen and consulted; this screen is better suited for tables and computers due to the arrangement of the interface elements

The setup screen is divided into 2 sections: the data fields section and the interactive table of nodes, as can be seen in Figure 3.20.

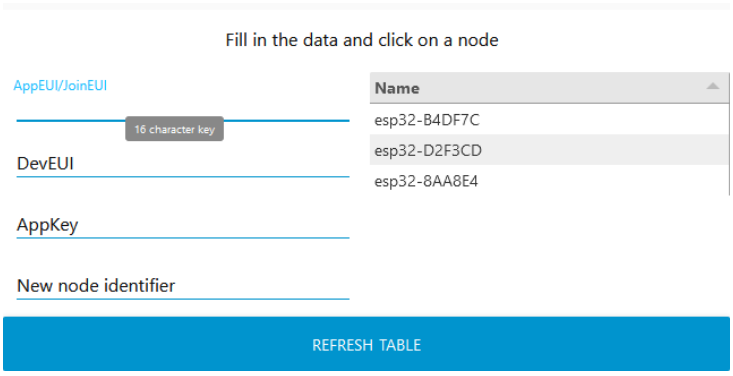
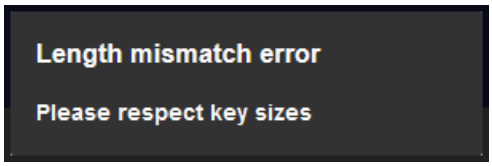
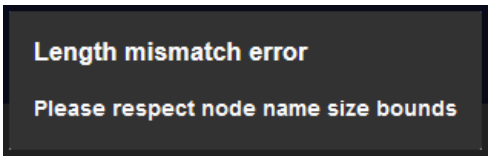


FIGURE 3.20. UI elements responsible for node setup.

The table of nodes consists of nodes that have sent a heartbeat message to the platform and thus are considered active with the intent to receive data. Nodes get added as they send their heartbeat messages, however the table can also be manually refreshed for cases where, for example, the platform was restarted while setting up nodes. Each input field has size requirements, the first 3, being LoRaWAN keys, have sizes dictated by the protocol, so, respectively, 16, 16 and 32 characters while the last field accepts inputs up to 16 characters. If any of these requires are not met, the user is notified, via contextually aware notifications about what is wrong, these can be seen in Figure 3.21.



(A) Key length error.



(B) Node name length error.

FIGURE 3.21. Possible input length errors.

As the table is interactive, the nodes are clickable. Upon a click and if there are no size mismatches in the input fields, the data gets sent to the respective node over its own MQTT topic, the fields get their values erased in preparation for the next node and the node in question gets removed from the table as soon as the visualization platform gets notified about the finalization of the setup process by the node. With this done, the platform will be able to receive readings from the node in question.

The data visualization screen is divided into 3 sections: the map, the graph and the gauges, as can be seen in Figure 3.22.

FIGURE 3.22. SoilIoT main UI.

The map displays all the nodes currently sending data as markers in their respective geographic positions, these markers are interactive and upon being clicked will show, in a box above the marker, the nodes' name, the latest readings and also notify the graph and the gauges.

The graph section is separated into 4 elements: the graph itself which plots a given node's NPK values over time, the elements responsible for querying historic data (this element is shared with the gauges), the number picker responsible for the amount of data points on the graph and the button to stop live updates (also shared with the gauges). As mentioned previously, clicking on a map marker notifies the graph and it in turn starts displaying received data in real time with updates as new data arrives, this can be stopped with the button mentioned previously. Querying past readings is done by choosing a date and time and then the node to query for data, just like with the live graph the amount of plotted points depends on the user's choice, this overrides the current data on the graph, also disabling live updates so the current data doesn't get overwritten by new data arriving, an example can be found in Figure 3.23.

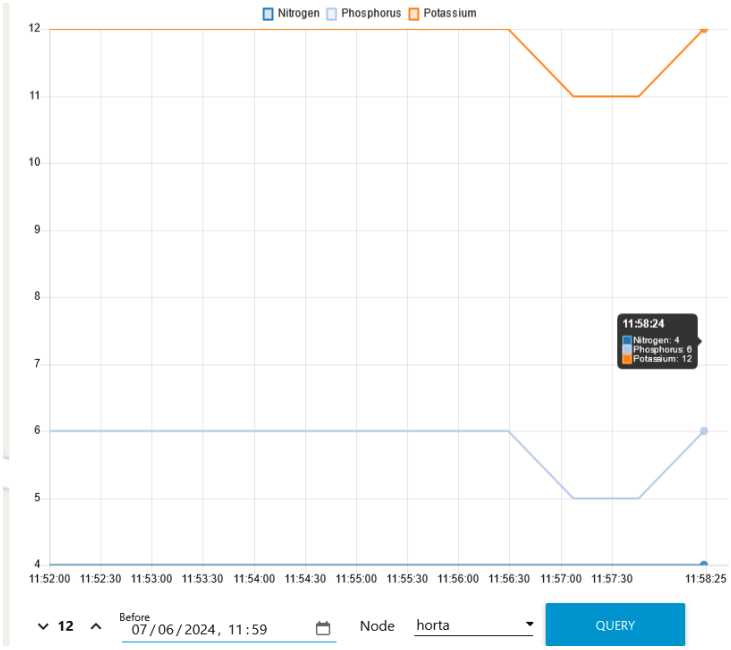


FIGURE 3.23. Historical data query.

The gauges connect to and work in tandem with the graph and can either display the latest live values or the latest values of the queried historic data, as can be seen in Figure 3.22. The scale chosen for these 3 elements was decided by dividing the maximum measurable value of the chosen NPK sensor, so 1999 mg/kg, by 10 and rounding it down, thus achieving 199 mg/kg. As mentioned previously, the button responsible for stopping live graph updates also affects the gauges.

When it comes to storing the acquired data, it is stored in a local SQLite database, with the tables found in Figure 3.24. The `Data` table stores, for each received reading, the node name, time of data acquisition, NPK values and the longitude and latitude of the reading.

```
CREATE TABLE "Data" (
  "ID"  INTEGER NOT NULL UNIQUE,
  "Node"  TEXT NOT NULL,
  "UnixTimestamp" INTEGER NOT NULL,
  "N" INTEGER NOT NULL,
  "P" INTEGER NOT NULL,
  "K" INTEGER NOT NULL,
  "A" REAL NOT NULL,
  "O" REAL NOT NULL,
  PRIMARY KEY("ID" AUTOINCREMENT)
);
```

(A) Data table creation query.

```
CREATE TABLE "Node" (
  "Id"  INTEGER NOT NULL UNIQUE,
  "Name"  TEXT NOT NULL UNIQUE,
  "Setup" INTEGER NOT NULL DEFAULT 0,
  PRIMARY KEY("Id" AUTOINCREMENT)
);
```

(B) Node table creation query.

FIGURE 3.24. SQLite table creation queries.

CHAPTER 4

Results and discussion

4.1. Experimental setup

Testing was done in an indoors environment due to several factors:

- running a continuous monitoring test indoors provides better and easier monitoring of the node itself
- faster reactions to issues that could arise
- hazard avoidance is easier, such as heat and rain (rain is particularly dicussed below)
- heat can have very detrimental effects on batteries

As for the target of the soil monitoring experiment, a random house plant was chosen. The only requirement was good soil that could absorb and keep water well. A picture of the plant can found in Figure 4.1.

FIGURE 4.1. The chose house plant.

As for the measuring process, the NPK sensor was simply inserted into the soil at an angle in order to spread the measuring rods better in the soil. It's also important to note that the sensor had to be inserted fully as it is based on TDR and the rods can't have air gaps to get proper measurements.

A 10000mAh 5V 2A powerbank was chosen as the source of power for convenience.

The test ran for about 3 days and collected data every 15 minutes. The whole setup can be found in Figure 4.2.

FIGURE 4.2. Data collection setup.

4.2. Data Analysis

One interesting result that came early on from testing was that soil dryness, as a factor, affects the sensor readings in an extreme manner. In Figure 4.3 we can see 3 different states of the sensor. In the first state, before 17:45 the sensor wasn't inserted into soil, which we can see by the readings being 0. Then the sensor was inserted into the soil that hasn't been watered in a while, and managed to read very low values as there is not enough humidity, indicated by the values rising slighty. Finally, the soil received water and the transmitted readings rose in an extreme manner and then began to stabilize as the soil absorbed the water.

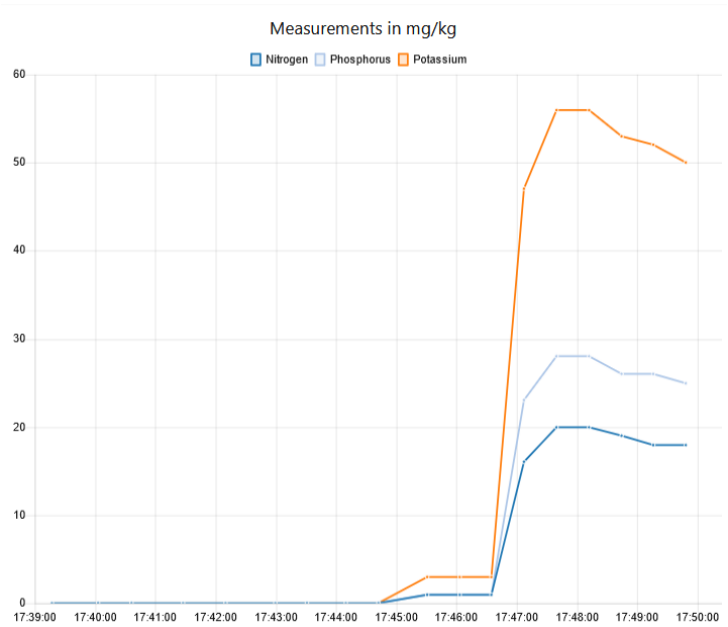


FIGURE 4.3. Soil dryness affecting readings.

As mentioned previously, during data acquisition, NPK values were acquired every 15 minutes, meaning 4 readings per hour, continuously for a number of days. As the purpose of the test was to demonstrate the ability of the system to collect data, the collection ran for 3 days in order to receive a fair amount of readings. The plant received water sparsingly after the initial watering in order to prevent overwatering. Data regarding this methodology with times and relative amounts can be found in Table 4.1.

Time	Date and Time	Relative amount
1st	14/09 11:39	Large
2nd	14/09 23:32	Small
3rd	16/09 12:25	Medium

TABLE 4.1. Watering date, time and relative amount

The results of the data collection, aggregated in 1 graph for easier analysis, can be seen in Figure 4.4.

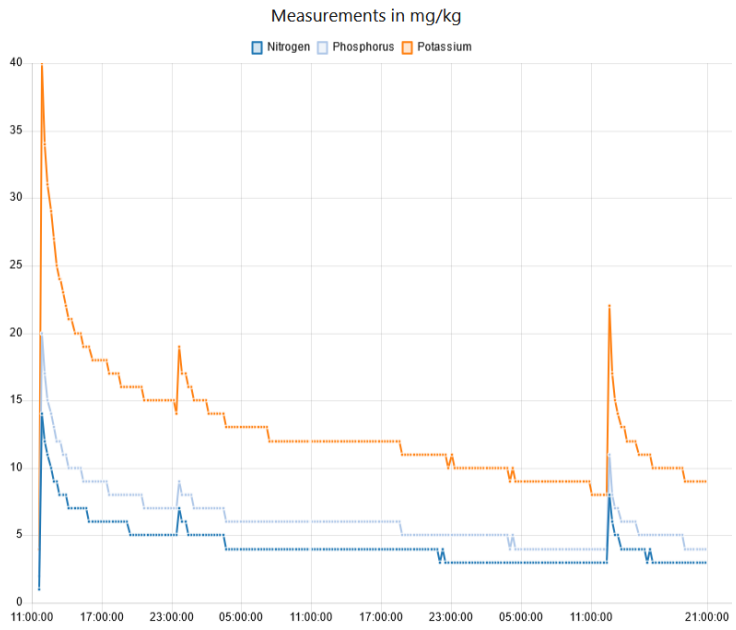


FIGURE 4.4. Graph of data collected over around 3 days.

The aforementioned graph aggregates over 200 transmitted readings. Out of these readings, only 1 is from before the soil was watered and for convenience it is shown in Table 4.2. The soil was still dry, and, as expected, the acquired values are extremely low.

	11:37
N (mg/kg)	1
P (mg/kg)	2
K (mg/kg)	4

TABLE 4.2. Readings from the first received transmission.

After a fair amount of water was added, the concentration reading had a noticeable jump reaching the maximum shown in Table 4.3. This is also the all time peak reading of the whole data collection process.

	11:52
N (mg/kg)	14
P (mg/kg)	20
K (mg/kg)	40

TABLE 4.3. The highest registered measurement.

Afterwards the readings began to decrease relatively quickly as the soil absorbed the water, a small amount of water was then added around 12 hours later and the readings spiked again, however this time the change in values was not much as can be seen in Table 4.4, the amount of water reflects on the increase.

	22:36	22:41
N (mg/kg)	5	7
P (mg/kg)	7	9
K (mg/kg)	14	19

TABLE 4.4. Difference in measurements after a small amount of water.

The sensor readings then reached relative stability for around 18 hours, between around 3:40 and 21:50 as can be seen in Figure 4.5. This likely means that the soil has absorbed the water and its humidity has stabilized.

	3:40	21:48
N (mg/kg)	4	4
P (mg/kg)	6	5
K (mg/kg)	13	11

TABLE 4.5. Span of time with stable measurements.

After 21:50 readings began to decrease faster before water was added again at around 12:25. This time a medium amount of water was added, less than the first time and more than the second. As with the large amount of water, afterwards the readings began to stabilize relatively quickly.

	12:21	12:36
N (mg/kg)	3	8
P (mg/kg)	4	11
K (mg/kg)	8	22

TABLE 4.6. Difference for a medium amount of water.

From this data we can reach several conclusions:

- soil humdity and especially water have a direct influence on the sensor readings
- related to the previous point, we shouldn't consider the readings right away after watering, instead, although the process takes a fair amount of time, we should wait for the readings to stabilize
- considering the reaction to water being introduced, the graph can also be used to detect rain and its relative amount, as the reaction is bigger the more water is introduced
- the graph could be used to figure out an optimal watering pattern (time between and amount of water) taking into account the amount of time it takes to reach stability and how the graph reacts to different amounts of water

CHAPTER 5

Conclusions and future work

5.1. Conclusions

This dissertation proposed a prototype of an IoT ecosystem of soil characteristics measuring nodes with data visualization capabilities via a web based user interface. Bringing the following benefits:

- cost effectiveness, unlike other ways of testing for NPK concentrations in soil, the hardware required for a node only needs to be purchased once, especially when the batteries are rechargable;
- time effectiveness, having distributed measuring nodes that continuously measure soil parameters eliminates the need for manual soil sample collection and testing;
- easy continuous monitoring, the nodes continuously measure NPK concentrations and transmit their readings to a visualization platform with configurable time intervals;
- easy access to the data, everything is shown on a convenient user interface which also allows querying for past data;
- full control, everything can be hosted locally, from the visualization platform to the database, with everything being fully customizable;
- better fertilizer usage, with the NPK concentrations readily available, fertilizer can be applied precisely when it is needed;
- range, LoRaWAN's range allows the nodes to be deployed in places where other protocols can't be used, as long as there is a gateway in range;

However the developed nodes currently have some limitations, the solutions to which may hinder the maintenance of the nodes themselves.

One of main problems is, of course, humidity which can quickly cause corrosion and it in turn could, for example, cause a short circuit, damaging the hardware in the process. More specifically, the RS485 to TTL converter and the step-up converter as they do not have any protection. A possible solution would be to isolate the node from the outside environment using, for example, silicone to fill the gaps, this however hinders maintenance work like battery replacement.

Rain is another problematic factor as it not only causes humidity, but also adds water into the equation. The sealing solution has the same problems as before, elevating the node above ground would solve the problem created by water, however the humidity variable remains without sealing.

Another issue is the sound produced by the mechanical power relay module, its intensity depending on the number of nodes and the frequency of readings. This could be solved by solid

state relays, however an issue arises with the way the current flow is cut using these. While mechanical relays are normally open and require a high signal to connect both terminals, solid state relays operate in the opposite way, letting current flow when they receive a low signal. It is possible to pin a pin to high to keep the switch open in deep sleep, but this also means extra power consumption by the development board which leads to a degraded autonomy. Another way to achieve a power switch would be by using transistors, or more specifically, Metal-oxide-semiconductor field-effect transistors (MOSFETs). While the principle behind and the assembly of a MOSFET switch are rather simple the issue arises in the fact that not all MOSFETs are equal and making a switch that works for the current nodes would require a careful iterative process with many different MOSFETs so as to not damage the hardware.

Another important limitation comes from the possible positioning error associated with the geographic positioning module, meaning that the node may not be displayed on its exact position in the visualization platform. This could cause issues when two nodes are in relatively close proximity.

5.2. Future work

The system was experimentally validated, however it can be improved in the future. Thus, several future developments can be considered:

- Using solar power to help with node autonomy, as the chosen battery booster comes with a charging circuit.
- Using the cloud to host both the UI and the database. Although this likely comes with extra monhtly costs which have to be evaluated against a local deployment.
- Building several nodes and optimizing their distribution in the field in order to map out the NPK concentration in area.
- Increase the realiability of the prototype.

References

[1] A. Snir, D. Nadel, I. Groman-Yaroslavski, *et al.*, "The Origin of Cultivation and Proto-Weeds, Long Before Neolithic Farming," *PLoS One*, vol. 10, no. 7, e0131422, 2015.

[2] L. Tabah and S. Kono, "World population trends in 1960-70," *Int Labour Rev*, vol. 109, no. 5-6, pp. 401–412, 1974.

[3] H. Ritchie, L. Rodés-Guirao, E. Mathieu, *et al.*, "Population growth," *Our World in Data*, 2023, https://ourworldindata.org/population-growth.

[4] H. Ritchie, P. Rosado, and M. Roser, "Agricultural production," *Our World in Data*, 2023, https://ourworldindata.org/agricultural-production.

[5] L. Medlicott, "Floods, droughts and panic attacks: Climate change is taking its toll on europe's farmers," *euronews.green*, Nov. 18, 2023. [Online]. Available: https://www.euronews.com/green/2023/11/18/floods-droughts-and-panic-attacks-climate-change-is-taking-its-toll-on-europes-farmers.

[6] C. Sorvino, "U.s. farmers struggle through drought to bring food to the table but face more challenges ahead," *Forbes*, Sep. 2, 2022. [Online]. Available: https://www.forbes.com/sites/chloesorvino/2022/09/02/us-farmers-struggle-through-drought-to-bring-food-to-the-table-but-face-more-challenges-ahead/?sh=77aac17fc084.

[7] T. N. Liliane and M. S. Charles, "Factors affecting yield of crops," in *Agronomy*, Amanullah, Ed., Rijeka: IntechOpen, 2020, ch. 2. DOI: 10.5772/intechopen.90672. [Online]. Available: https://doi.org/10.5772/intechopen.90672.

[8] *Nitrogen, Phosphorus, Potassium Fertilizer Promoting Plant Growth*. [Online]. Available: https://www.fertilizer-machines.com/solution/fertilizer-technology/how-N-P-K-fertilizer-promote-plant-growth.html.

[9] *Beginning the Internet of Things - Kevin Ashton*. [Online]. Available: https://medium.com/@kevin_ashton/beginning-the-internet-of-things-6d5ab6178801.

[10] A. S. Yeole and D. R. Kalbande, "Use of internet of things (iot) in healthcare: A survey," *ACM International Conference Proceeding Series*, vol. 21-22-March-2016, pp. 71–76, Mar. 2016. DOI: 10.1145/2909067.2909079. [Online]. Available: https://dl.acm.org/doi/10.1145/2909067.2909079.

[11] H. Arasteh, V. Hosseinnezhad, V. Loia, *et al.*, "Iot-based smart cities: A survey," *EEEIC 2016 - International Conference on Environment and Electrical Engineering*, Aug. 2016. DOI: 10.1109/EEEIC.2016.7555867.

[12] *Smart farming ag-tech | ChrysaLabs*. [Online]. Available: https://www.chrysalabs.com/.

[13] *Smart Farming system and Precision Agriculture software by GeoPard*. [Online]. Available: https://geopard.tech/.

[14] *NPK Soil Test Kit*. [Online]. Available: https://www.pacificsensortech.com.au/mt6003-npk-soil-test-kit-australia.

[15] F. Guan, M. Xia, X. Tang, and S. Fan, "Spatial variability of soil nitrogen, phosphorus and potassium contents in moso bamboo forests in yong'an city, china," *CATENA*, vol. 150, pp. 161–172, Mar. 2017, ISSN: 0341-8162. DOI: 10.1016/J.CATENA.2016.11.017.

[16] Y. Zhao, X. Xu, J. L. Darilek, B. Huang, W. Sun, and X. Shi, "Spatial variability assessment of soil nutrients in an intense agricultural area, a case study of rugao county in yangtze river delta region, china," *Environmental Geology*, vol. 57, pp. 1089–1102, 5 May 2009, ISSN: 09430105. DOI: 10.1007/S00254-008-1399-5/TABLES/6. [Online]. Available: https://link.springer.com/article/10.1007/s00254-008-1399-5.

[17] *Soil Master™ Chemistry - How Do the Tests Work?* [Online]. Available: https://www.labtechtests.com/Page/Soil_Master_How_Do_the_Tests_Work.aspx.

[18] *AMOLA® AGRAR MOBILE LAB, base unit | Nutrient analysis | Agricultural Measuring | Products | Pronova*. [Online]. Available: https://pronova.de/en/products/agricultural-measuring/nutrient-analysis/851/amola-agrar-mobile-lab-base-unit.

[19] *Sensores de nitrogênio, fósforo e potássio no solo (saída 4-20mA)*. [Online]. Available: https://eiccontrols.com/pt/inicio/480-sensores-de-nitrogeno-fosforo-y-potasio-del-suelo-salida-4-20ma.html.

[20] *Max485 module*. [Online]. Available: https://www.botnroll.com/pt/rs485/1037-max485-module.html.

[21] W. Fan, K. A. Kam, H. Zhao, P. J. Culligan, and I. Kymissis, "An optical soil sensor for npk nutrient detection in smart cities," *2022 18th International Conference on Intelligent Environments, IE 2022 - Proceedings*, 2022. DOI: 10.1109/IE54923.2022.9826759.

[22] *Learn about Types and Applications of Microcontrollers - EIT: Engineering Institute of Technology*. [Online]. Available: https://www.eit.edu.au/resources/types-and-applications-of-microcontrollers/.

[23] *Hesp32-wroom-32e esp32-wroom-32ue datasheet*, Version 1.6, Espressif Systems, 2023.

[24] *Esp32-s2-saola-1*. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32s2/hw-reference/esp32s2/user-guide-saola-1-v1.2.html.

[25] *Arduino Nano Every*. [Online]. Available: https://store.arduino.cc/collections/boards-modules/products/arduino-nano-every.

[26] *Raspberry Pi Pico W*. [Online]. Available: https://mauser.pt/catalog/product_info.php?products_id=095-0597.

[27] *Arduino® nano every product reference manual*, https://docs.arduino.cc/resources/datasheets/ABX00028-datasheet.pdf, Arduino.

[28] *ESP SoCs | Espressif Systems*. [Online]. Available: https://www.espressif.com/en/products/socs.

[29] *ESP Product Selector*. [Online]. Available: https://products.espressif.com/#/product-selector?names=.

[30] *ESP32 Series Datasheet*, Version 4.4, Espressif Systems, 2023.

[31] S. Farahani, *ZigBee Wireless Networks and Transceivers*. USA: Newnes, 2008, ISBN: 0750683937.

[32] *zigbee Specification*, Revision 22 1.0, ZigBee Alliance, Apr. 2017.

[33] *ZigBee and Wi-Fi Coexistence | MetaGeek*. [Online]. Available: https://www.metageek.com/training/resources/zigbee-wifi-coexistence/.

[34] *Learn About Z-Wave Protocol*. [Online]. Available: https://www.silabs.com/wireless/z-wave?tab=learn.

[35] *Z-Wave 500 vs. 700 vs. 800 - Why Use the New 800 Series for Smart Home Devices*. [Online]. Available: https://community.silabs.com/s/share/a5U8Y000000bwgaUAA/zwave-500-vs-700-vs-800-why-use-the-new-800-series-for-smart-home-devices?language=en_US.

[36] *Z-Wave Long Range (LR) Overview - Silicon Labs*. [Online]. Available: https://www.silabs.com/wireless/z-wave/z-wave-long-range-overview.

[37] *Introduction of Z-Wave - Silicon Labs*. [Online]. Available: https://community.silabs.com/s/article/Introduction-of-Z-Wave.

[38] *Application Work Group Z-Wave Specifications*, Release 2023A, Z-Wave Alliance, Jan. 2023.

[39] *Z-Wave Global Regions - Silicon Labs*. [Online]. Available: https://www.silabs.com/wireless/z-wave/global-regions.

[40] *Bluetooth Technology Overview | Bluetooth® Technology Website*. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/.

[41] *Understanding Bluetooth Range | Bluetooth® Technology Website*. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/.

[42] J. Marcel, *How bluetooth technology uses adaptive frequency hopping to overcome packet interference*, Nov. 2020. [Online]. Available: https://www.bluetooth.com/blog/how-bluetooth-technology-uses-adaptive-frequency-hopping-to-overcome-packet-interference/.

[43] *Bluetooth core specification*, Rev. 5.3, Core Specification Working Group, Jul. 2021.

[44] *What is Sigfox? - Sigfox 0G Technology*. [Online]. Available: https://www.sigfox.com/what-is-sigfox/.

[45] *Qualification | Sigfox build*. [Online]. Available: https://build.sigfox.com/study.

[46] *Sigfox Technical Overview*, May 2017, sigfox, May 2017.

[47] *Radio Configurations | Sigfox build*. [Online]. Available: https://build.sigfox.com/sigfox-radio-configurations-rc.

[48] *Select country | Sigfox Buy*. [Online]. Available: https://buy.sigfox.com/buy.

[49] *Uplink / Downlink messages*. [Online]. Available: https://build.sigfox.com/technical-quickstart#uplink-downlink-messages.

[50] *What is LoRaWAN® Specification - LoRa Alliance®*. [Online]. Available: https://lora-alliance.org/about-lorawan/.

[51] *What are LoRa and LoRaWAN?* [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/.

[52] *Frequency Plans by Country*. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country/.

[53] *Fair use policy*. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/duty-cycle/#fair-use-policy.

[54] *MQTT - The Standard for IoT Messaging*. [Online]. Available: https://mqtt.org/.

[55] *MQTT Server | The Things Stack for LoRaWAN*. [Online]. Available: https://www.thethingsindustries.com/docs/integrations/mqtt/.

[56] *openHAB BasicUI Demo*. [Online]. Available: https://demo.openhab.org/basicui/app?sitemap=demo.

[57] *Rules | openHAB*. [Online]. Available: https://www.openhab.org/docs/configuration/rules-dsl.html.

[58] *Node-RED*. [Online]. Available: https://nodered.org/.

[59] O. Kobelyuk, O. Postolache, and B. Mataloto, "Smart Sensing and IoT for Precision Agriculture - Soil Characteristics Monitoring," in *Proceedings of the 4th International Symposium on Sensing and Instrumentation in IoT Era*, (Azores, Portugal), IEEE, Ed., IEEE, Aug. 2024.

[60] *node-red-dashboard (node) - Node-RED*. [Online]. Available: https://flows.nodered.org/node/node-red-dashboard.

[61] *Grafana: The open observability platform | Grafana Labs*. [Online]. Available: https://grafana.com/.

[62] *IoT Analytics - ThingSpeak Internet of Things*. [Online]. Available: https://thingspeak.com/.

[63] *How to Buy - ThingSpeak IoT*. [Online]. Available: https://thingspeak.com/prices.

[64] *Air quality with SDS011 - ThingSpeak IoT*. [Online]. Available: https://thingspeak.com/channels/357142.

[65] *NPK Mapper - Aero Vines*. [Online]. Available: https://aerovines.com.au/pages/npk-mapper.

[66] F. Ahmad and L. Goparaju, "Geospatial approach for agroforestry suitability mapping: To enhance livelihood and reduce poverty, fao based documented procedure (case study of dumka district, jharkhand, india)," *Biosciences Biotechnology Research Asia*, vol. 14, p. 15, Jun. 2017. DOI: 10.13005/bbra/2491.

[67] S. Postolache, P. Sebastião, V. Viegas, O. Postolache, and F. Cercas, "Iot-based systems for soil nutrients assessment in horticulture," *Sensors 2023, Vol. 23, Page 403*, vol. 23,

p. 403, 1 Dec. 2022, ISSN: 1424-8220. DOI: 10.3390/S23010403. [Online]. Available: https://www.mdpi.com/1424-8220/23/1/403.

[68] B. Cheruvu, S. B. Latha, M. Nikhil, H. Mahajan, and K. Prashanth, "Smart farming system using npk sensor," *2023 9th International Conference on Advanced Computing and Communication Systems, ICACCS 2023*, pp. 957–963, 2023. DOI: 10.1109/ICACCS57279.2023.10112795.

[69] J. Aliparo, C. D. Santos, C. M. Sese, *et al.*, "Iot-based assessment and monitoring of npk content and fertility condition of soil," *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2022-November, 2022, ISSN: 21593450. DOI: 10.1109/TENCON55691.2022.9978040.

[70] M. Pyingkodi, K. Thenmozhi, M. Karthikeyan, T. Kalpana, S. Palarimath, and G. B. A. Kumar, "Iot based soil nutrients analysis and monitoring system for smart agriculture," *3rd International Conference on Electronics and Sustainable Communication Systems, ICESC 2022 - Proceedings*, pp. 489–494, 2022. DOI: 10.1109/ICESC54411.2022.9885371.

[71] L. K. Sundari, M. Rana, S. T. Ahmed, and K. Anitha, "Real-time iot based temperature and npk monitoring system sugarcane-crop yield for increasing," *3rd IEEE International Virtual Conference on Innovations in Power and Advanced Computing Technologies, i-PACT 2021*, 2021. DOI: 10.1109/I-PACT52855.2021.9696564.

[72] R. Madhumathi, T. Arumuganathan, and R. Shruthi, "Soil npk and moisture analysis using wireless sensor networks," *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*, Jul. 2020. DOI: 10.1109/ICCCNT49239.2020.9225547.

[73] *LPS8N Indoor LoRaWAN Gateway*. [Online]. Available: https://www.dragino.com/products/lora-lorawan-gateway/item/200-lps8n.html.

[74] *jgromes/RadioLib: Universal wireless communication library for embedded devices*. [Online]. Available: https://github.com/jgromes/RadioLib.

[75] *mcci-catena/arduino-lmic: LoraWAN-MAC-in-C library, adapted to run under the Arduino environment*. [Online]. Available: https://github.com/mcci-catena/arduino-lmic.

[76] *ngraziano/LMICPP-Arduino: Lmic (LoraWAN-in-C) modified to C++*. [Online]. Available: https://github.com/ngraziano/LMICPP-Arduino.

Annexes

The article presented during the "4th International Symposium on Sensing and Instrumentation in IoT Era" conference comes annexed next alongside related information about the conference.

ISSI 2024

AUGUST, 29-30

AZORES, PORTUGAL

4th International Symposium on Sensing and Instrumentation in IoT Era

ISSI 2024 Technical and Financial Sponsors

Platinum Sponsor

ANACOM AUTORIDADE NACIONAL DE COMUNICAÇÕES

Sponsoring Organizations

iscte UNIVERSITY INSTITUTE OF LISBON

SHANGHAI MARITIME UNIVERSITY

IEEE

IEEE INSTRUMENTATION & MEASUREMENT SOCIETY®

instituto de telecomunicações

IEEE IMS TC-13
Wireless & Telecommunications in Measurements

11:40 - 12:20

Keynote 1.3: Precision Nutrition in the Digital Age: Wearable Sensors, AI, and Personalized Health
Edward Sazonov

Room: Nonagon Auditorium
Chair: Mariana Jacob Rodrigues and Octavian Postolache

12:20 - 14:00

Lunch – Q'énosso restaurant

S1.1. Session: IoT Applications I (14:00 – 15:45)

Oral Presentation Session

Room: Nonagon Auditorium
Chair: Carlos Coutinho and José Miguel Dias Pereira

14:00
Time-Constraint Multiple Mobile Chargers Scheduling Method for Large-Scale Wireless Rechargeable Sensor Networks
Xiaoqiang Lu; Qian Zhang; Haiqing Yao

14:15
Smart Sensing and IoT for Precision Agriculture - Soil Characteristics Monitoring
Oleksandr Kobelyuk; Octavian Adrian Postolache; Bruno Mataloto

14:30
Smart Sensors and IoT Applied in Digital Twin for Industry 4.0
Assiya Boltaboyeva; Zhanel Baigarayeva; Octavian Postolache; Madina Mansurova; Nurgul Karymssakova; Baurzhan Belgibayev

14:45
Integrating IoT and Machine Learning for Advanced Chemoresistive Sensor Development in Smart Lab Environments
Bibars Amangeldy; Nurdaulet Tasmurzayev; Gaukhar Smagulova; Bayan Kaidar; Madina Mansurova; Nurdaulet Izmailov

Smart Sensing and IoT for Precision Agriculture - Soil Characteristics Monitoring

Oleksandr Kobelyuk
Iscte - Instituto Universitário de Lisboa
Lisbon, Portugal
okkrl@iscte-iul.pt

Octavian Postolache
Iscte - Instituto Universitário de Lisboa
and
Instituto de Telecomunicações
Lisbon, Portugal
opostolache@lx.it.pt

Bruno Mataloto
Iscte - Instituto Universitário de Lisboa
Lisbon, Portugal
bruno_mataloto@iscte-iul.pt

***Abstract*—This paper proposes and implements an energy-efficient battery-powered IoT system to monitor soil for macronutrients. Each measurement node is comprised of a development board, a sensor to measure the N, P and K concentrations in soil, a GPS module to acquire the geolocation and a power relay to cut power in order to conserve battery. The acquired data is displayed on a Node-RED dashboard with geographic and concentration data tracking capabilities for each node, with the possibility to consult historic data. The system was validated locally by monitoring soil conditions for plants.**

I. Introduction

The world population has been steadily increasing since the dawn of humanity until the 1950s, when an unprecedented population boom occurred due to the wide availability of public health measures in less developed regions [1], and in less than a century the population has grown from 2.5 billion to 8 billion [2]. As a consequence, humanity had to drastically increase its agricultural production, from 205 million tons to 1.2 billion tons [3].

Such an increase wouldn't be possible without advancements in the agicultural field, be it machinery or the means to monitor the soil and plants for various factors, in particular, humidity, temperature and macronutrients such as NPK (nitrogen (N), phosphorus (P), and potassium (K)) for maximum efficiency and yield, especially nowadays with climate change and prolonged droughts being an ever increasing issue for farmers [4], [5].

One of the strategies employed to achieve improved production sustainability is called Precision Agriculture. Monitoring and reacting upon the previously mentioned factors to increase yield and reduce waste is one way to apply this practice to fields.

NPK macronutrients are of high importance for crop yield [6]: Nitrogen (N) is crucial for the production of proteins, nucleic acids and chlorophyll which drives photosynthesis, Phosphorus (P) is necessary for root growth, formation of fruits and seeds and blooming of flowers, and Potassium (K) helps with moving water and nutrients in the plant. A deficiency of these elements can lead to stunted growth, reduced yields and necrosis [7].

The measurement of the soil characteristics is an expensive and/or time consuming endeavor: easy-to-use solution-based soil tests have limited uses and thus require continuous repurchasing and sample collection in addition to only providing relative results, laboratory tests are both time consuming due to soil sample gathering and expensive due to the cost of reliable portable laboratories and finally capacitive soil sensors require extra electronics in order to interface with them over the RS485 standard.

With the previously mentioned factors in mind, this paper will present an easy-to-use innovative IoT ecosystem of nodes for measuring NPK concentrations for precision agriculture applications.

II. Related work

Specialized companies have emerged to offer these IoT-based services to farmers, facilitating and optimizing the agriculture industry in general.

ChrysaLabs [8] provides a proprietary probe which can measure over 35 soil parameters and transmit them to their cloud service geotagged. However this probe requires the user to walk and test each soil spot manually, and as the field grows, time efficiency decreases unless more probes are bought and workers hired, which in turn diminishes economic efficiency.

Geopard [9] provides precision agriculture solutions, including but not limited to: crop monitoring, soil and yield data analytics. The incurred cost and the saturation of features may be unjustified for small or medium scale farms. Aditionally the end user doesn't have full control of their own measuring nodes.

Drones, being a non invasive technology, have also seen an increased use in farming, namely for crop and soil condition monitoring and pest control. Some companies that offer these services include Mapware [10] with an expansive suite of soil monitoring solutions and DJI Agriculture [11] offering pest control solutions via spraying.

IoT and sensor based precision agriculture is a relatively new field in academia, the following paragraphs will focus on some implementations of NPK monitoring.

Postolache et al. [12] reported an implementation of an ecosystem for monitoring NPK and moisture values in a public garden, however a system like this cannot be scaled for

farms as communications occur over Wi-Fi, which does not have a high range, additionally the visualization application is mobile-only and does not allow consulting historic data or charting said data on a graph.

Cheruvu et al. [13] unlike the previous implementation, allows the consulting and plotting of historic data, however the implemeneted nodes use Wi-Fi for data transmission which is not an ideal solution for farms, the platform used to plot data is proprietary, meaning the user doesn't have full control over the system, and there is no mention of running multiple nodes.

Aliparo et al. [14], similarly to previous implementations, uses Wi-Fi for data transmission and doesn't mention running multiple nodes, however, unlike other implementations, tests were performed on multiple soil types albeit in a laboratory environment.

Pyingkodi et al. [15] while oriented for smart farming, the power is provided by AC power grid and uses GSM for communications, neither ideal for farming fields.

Sundari et al. [16], unlike previous implementations, does not come with a user facing interface, opting instead to display data on a node-mounted LCD screen meaning each node would need to be checked manually and individually.

Madhumath et al. [17] reported another implementation using Wi-Fi for communications and a mobile application that may not offer all the features needed by a user, additionally, in this case, Amazon Web Services (AWS) are used which means extra costs for the end user as the platform is not free.

It is also imporant to mention that none of the previously mentioned works refer their approach regarding battery longevity/node autonomy optimization.

III. System Description

The proposed IoT ecosystem includes a set of energy-efficient sensing nodes that can measure NPK concentrations in soil and can be easily augmented with new sensors if required.

A. Hardware and Communication Protocols

The proposed sensor node comprises a few mandatory components to properly work, this list includes a development board, a GPS module, a NPK sensor, a step-up module, a UART TTL to RS485 module and a relay module, thus the following choices were made:

- LilyGO TTGO T3 LoRa32 was chosen as the development board due to its integrated SX1276 LoRa chip, LoRa antenna ports, OLED screen and easy battery integration with a power switch and cables for direct connection. However any other development board with similar capabilities could also be used, such as a Heltec NodeMCU. Additionally, the SiP used on the chosen board, ESP32-PICO-D4, provides excellent current consumption in deep sleep [18, Table 15], a factor necessary for node autonomy on battery power. The aforementioned development board comes with a small 1.6dBi LoRa antenna, which was upgraded to increase the range.
- Air530 was chosen as the GPS module due to its relatively low positioning error of 10 meters, good temperature and humidity operating ranges and support for all the major satellite navigation systems: GPS, GALIEO, GLONASS and BEIDOU. However any other GPS module with the same connections could also work, such as a NEO-6M.
- JXBS-3001-TR-RS was chosen as the NPK sensor as it is a 7-in-1 sensor providing not only NPK values but also soil pH, conductivity, temperature and humidity. Meaning the nodes can be easily expanded to read other parameters without integrating other sensors.
- A generic XL6009E1 step-up module was chosen in order to convert 5V into 12V, the recommended voltage for the NPK sensor.
- A UART TTL to RS485 Two-way Converter (with the CD4069BM chip) is responsible for enabling communications between the NPK sensor and the development board.
- A single channel RobotDyn DC 5V relay module, running SLA-05VDC-SL-A, was chosen as the switch is open by default and only closes, letting current pass, when it is enabled with a high signal.

Figure 1 represents an assembled node prototype with all the components included: the development board, the upgraded antenna, the GPS module, the NPK sensor, the voltage regulator, the UART TTL to RS485 module and the relay module.

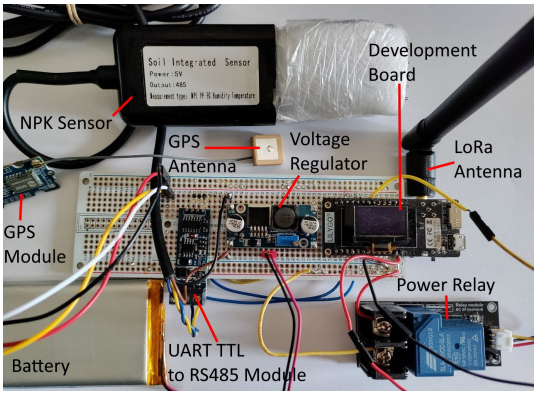


Fig. 1. Node prototype

As for the LoRaWAN gateway, the Dragino LPS8N, powered by the SX1302 and the SX1250, was chosen due to its support for most of the frequency bands, 3 methods for internet connection (Wi-Fi, ethernet and the built-in 4G modem) and smart agriculture application as suggested by the manufacturer.

The setup process needs a fast short range protocol while soil data transmission needs a long-range-low-power protocol, thus the choice falls on MQTT-over-Wi-Fi and LoRaWAN:

- MQTT is a topic-based publish-subscribe messaging protocol. Deployments are made out of two types of devices: clients that may both publish data and sub-

scribe to topics, and the broker that receives and routes data to subscribers. Being topic-based, it is adequate for devices with different names as data can be published directly on the topic of the device, for example `/SoilIoT/esp32-XYZ123/lora`.

- LoRaWAN is a Low-power Wide-area Network (LPWAN) with ranges from 3 kilometers in urban environments and 10 kilometers in rural ones and can last up to 10 years on a single coil cell battery [19], however it can go beyond that in terms of range as the current record stands at 1336 km [20]. Communications occur in sub-1GHz frequencies meaning there is no interference with, for example, Wi-Fi and ZigBee (another popular IoT communications protocol, runs over 2.4GHz thus, realistically, it can interfere with Wi-Fi), with data rates between 0.3 and 50 kbit/s. Due to the potential locations of the nodes, the available power sources, and the nature of the data being transmitted, these characteristics make LoRa an ideal solution for our proposed IoT System. The most popular network server belongs to TTN (The Things Network) due to its free plan with good fair usage policies, which translate to: per day, the sum of all uplink (node to server) airtimes, which means the time it takes for a message to reach the server, can only sum up to 30 seconds and downlinks (server to node) are limited to 10. TTN, among many other protocols, allows for the usage of MQTT to distribute the data received over uplinks.

Figure 2 represents a data flowchart of the system with the equipment and data transmission included.

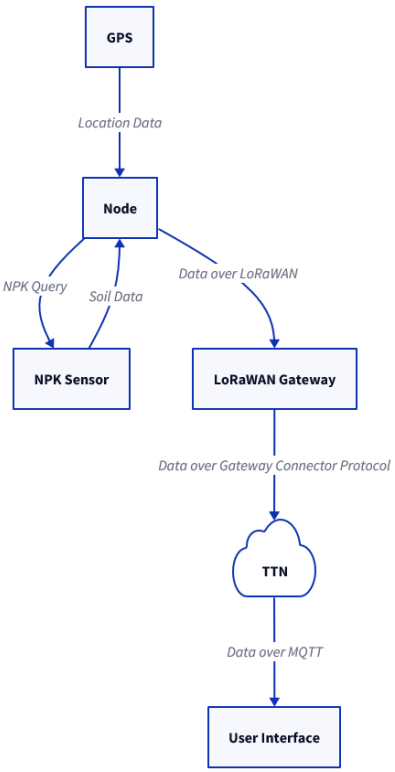


Fig. 2. System data flow flowchart

B. Firmware

The node firmware was written with C++ using the Arduino framework on PlatformIO, an extensible integrated environment for embedded development.

Nodes can only have 2 states: setup mode and data-sending mode. During the setup process the node looks for a given Wi-Fi network, which can be modified, to then connect to the user interface via MQTT. Upon successful connection, the OLED screen displays the node name so the end user knows which node in the list to pick (Fig. 3).

Fig. 3. Setup information on the node's OLED screen

The end user can then set the LoRaWAN keys and is encouraged to change the node's name. This data is then sent to the node by clicking on it in the list (Fig. 4).

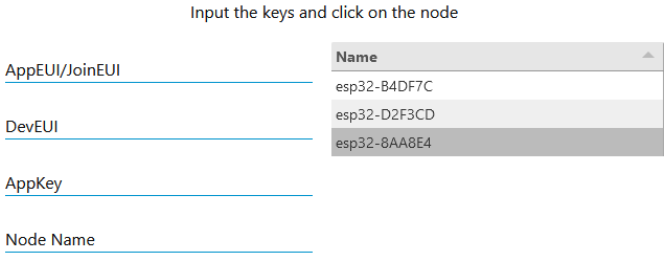


Fig. 4. Node setup user interface

Upon a succesful setup, the node sends a success message over MQTT to the backend and its name is removed from the "to be configured" list. At the same time the end user is prompted to restart the node (Fig. 5) to put it into the data-sending mode, at this stage the node can be switched to battery power, with the switch turned off, and placed at the final location.

It is also important to mention that the OLED screen will remain turned off from now on.

Fig. 5. Node restart prompt

At this point each node will attempt to acquire and send data once every defined period of time which can be altered. The node will wait for an open transmission window and upon a successful transmission will cut power to every sensor using the previously mentioned power relay, save the state of the LoRaWAN connection, so as not to reconnect every time, and then enter deep sleep to conserve even more energy. After

waking up, power gets restored to the sensors, the LoRaWAN connection is restored and the process repeats.

It is also important to mention that the GPS location is acquired only once, during the first power cycle, and is then saved between power cycles by using RTC memory, a low power memory which runs independently of the main processor. This is due to the fact that the GPS module takes time to warm up and acquire valid location data.

C. Data visualization

As alluded before, readings are sent to the user facing interface for easy access. The interface is implemented in Node-RED which describes itself as "Low-code programming for event-driven applications" meaning it is suitable for application in this system as data is sent via MQTT topics which can be considered an event. Node-RED flow programming is done by wiring blocks together, meaning it is easy to use and learn, additionally the platform provides a rich ecosystem of extensions, one of which allows the development of interactive dashboards. Figure 6 represents a simple example of Node-RED programming where data is received via MQTT, formatted into a SQL query, inserted into a database and finally a notification about this is sent back via MQTT.

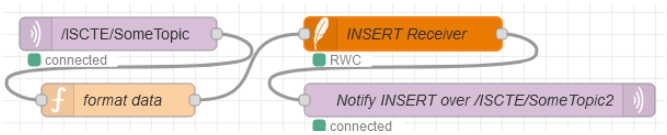


Fig. 6. Simple Node-RED flow.

This interface is divided in 3 parts: node map, data graph and data gauges. The map displays all the nodes currently sending data. Upon clicking on a node 3 things happen: the latest readings are shown in a text box above the node (Fig. 7), the graph gets set to track the selected node (Fig. 8) and the gauges also show the latest values on a scale (Fig. 9).

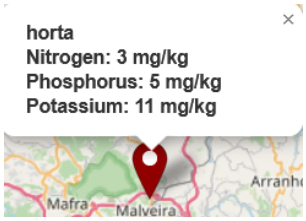


Fig. 7. Node on the map

It is important to mention that the graph shows the latest N values, where N is a value chosen by the user in the visualization platform.

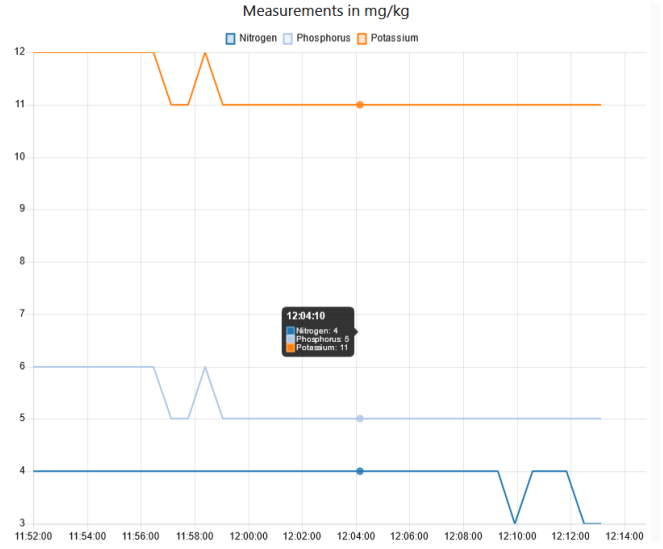


Fig. 8. Evolution graph of the concentration readings.

The gauges exist to complement the text box and their scale can be set to any value, 199 mg/kg being chosen in this case for demostration purposes.

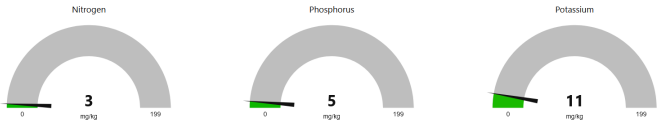


Fig. 9. Latest values on the gauges

Additionally the end user can consult historic data of any node by querying the database with a date and time as can be seen on Fig. 10. This feature may allow, for example, tracking how NPK concentrations change per season.

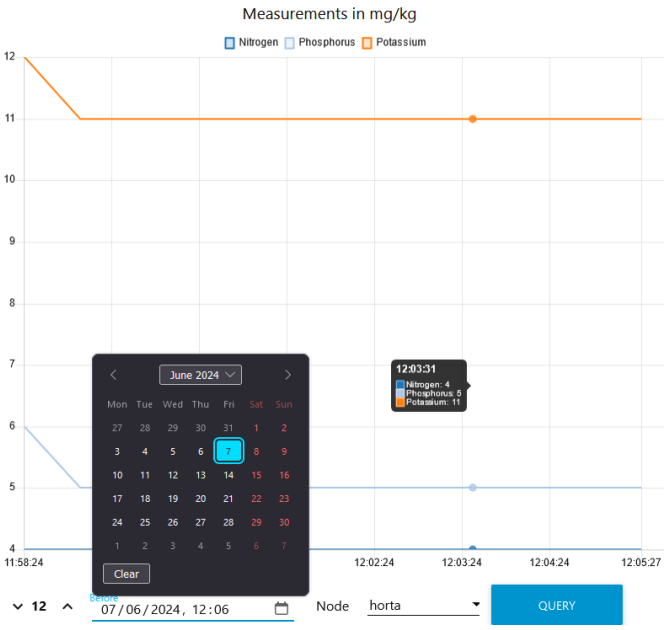


Fig. 10. Evolution of NPK concentration data of a node.

All data is stored in a local SQLite allowing future queries and data analitics over collected data.

IV. Results and Discussion

In order to test the system while respecting TTN's fair-use policy, the data transmission interval was set to 30 seconds and ran for about 20 minutes, this process netted 34 data points. Some of the results can be seen in the previous figures, however Figure 11 shows what the user would see while using the user interface in its default state.

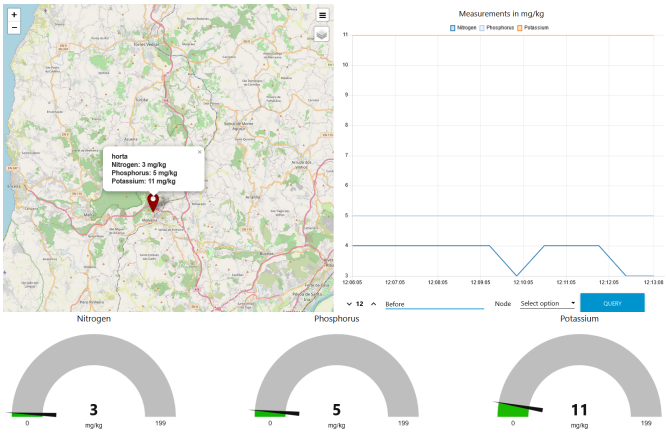


Fig. 11. The user interface

Table I shows the latest 10 values with reference to Figure 11, data is sent every 30 seconds plus the time spent waiting for an open transmission window. Additionally, the latitude and longitude are always the same because, as referred previously, the location is acquired only once so as not to wait for the GPS module to warm up every time.

TABLE I
THE 10 LATEST DATA READINGS FOR NODE "HORTA".

Time	N	P	K	Latitude	Longitude
HH:MM:SS	mg/kg			Degrees	Degrees
12:07:22	4	5	11	38.9389153	-9.2531672
12:08:00	4	5	11	38.9389153	-9.2531672
12:08:39	4	5	11	38.9389153	-9.2531672
12:09:17	4	5	11	38.9389153	-9.2531672
12:09:55	3	5	11	38.9389153	-9.2531672
12:10:34	4	5	11	38.9389153	-9.2531672
12:11:12	4	5	11	38.9389153	-9.2531672
12:11:51	4	5	11	38.9389153	-9.2531672
12:12:29	3	5	11	38.9389153	-9.2531672
12:13:07	3	5	11	38.9389153	-9.2531672

As can be seen, the system is able to capture NPK concentrations and the GPS location for each node and transmit them to the user interface using LoRaWAN and MQTT, meaning it is a working implementation which allows for future development and augmentation, the chosen NPK sensor in particular allows for the easy augmentation of collected data as mentioned previously.

The main benefits of this node ecosystem over previous approaches include:

- cost effectiveness - other easily accessible methods of measuring NPK concentrations include chemical tests, either via kits that have limited uses and provide relative results (low, medium, high) or portable laboratories that provide numeric results in exchange for a high cost (including the purchase of new reagents); sensors, on the other hand, are a one-time-purchase and provide numeric results
- time effectiveness - continuous autonomous monitoring removes the need for manual soil testing, which can be especially time consuming on big fields as NPK macronutrients are not distributed equally [21], [22]
- continuous monitoring - having reliable timed data allows for more precise fertilizer usage, which, in turn, may save costs and increase yields
- rural area support - LoRaWAN's long range allows the nodes to be deployed in more remote fields as long as there is a gateway in range
- full control - everything is fully customizable and augmentable, from the hardware to the software and user interface

V. Conclusion

This paper presents the design and implementation of a long-range IoT-based continuous soil monitoring system.

The system was tested by performing soil measurements every 30 seconds for a period of time, this test yielded the expected results of being able to measure concentrations periodically without interruption while also reporting the geographical location of the node. The measured data was accessible via the user interface in 2 ways: the live feed which shows the latest data (node on map with tooltip, graph with the latest 10 values and the gauges with the latest values on a scale) and historic data accessible by querying the database and later displayed on a graph.

The following is brought to the end user: cost effectiveness, multi-node support, full control and customization, node autonomy on battery power, easy setup process, easily expandable array of collected data, long-range transmissions powered by a standardized low-power protocol and an easily augmentable user inteface with node and data tracking capabilities.

Future work could involve several enhancements to improve the system's functionality and user experience. One improvement could be the development of a heatmap-style concentration overlay for each macronutrient, providing a visual representation of nutrient distribution across the field. Additionally, designing a custom PCB tailored to the specific hardware used would streamline the system, potentially reducing size and increasing efficiency. Another valuable feature would be the integration of cloud storage options for data, allowing for easy access, analysis, and sharing of information from anywhere. Finally, solar power support could be implemented in order to charge the power source during the day, improving autonomy.

Acknowledgment

This work was funded by the Instituto de Telecomunicações and by FCT/MCTES through national funds and, when applicable, from EU funds co-financed under the UIDB/50008/2020 project. In addition, this work was also funded by ISCTE-Instituto Universitário de Lisboa.

References

[1] L. Tabah and S. Kono, "World population trends in 1960-70," *Int Labour Rev*, vol. 109, no. 5-6, pp. 401–412, 1974.

[2] H. Ritchie, L. Rodés-Guirao, E. Mathieu, *et al.*, "Population Growth," *Our World in Data*, 2023, https://ourworldindata.org/population-growth.

[3] H. Ritchie, P. Rosado, and M. Roser, "Agricultural Production," *Our World in Data*, 2023, https://ourworldindata.org/agricultural-production.

[4] L. Medlicott, "Floods, droughts and panic attacks: Climate change is taking its toll on Europe's farmers," *euronews.green*, Nov. 18, 2023. [Online]. Available: https://www.euronews.com/green/2023/11/18/floods-droughts-and-panic-attacks-climate-change-is-taking-its-toll-on-europes-farmers.

[5] C. Sorvino, "U.S. Farmers Struggle Through Drought To Bring Food To The Table But Face More Challenges Ahead," *Forbes*, Sep. 2, 2022. [Online]. Available: https://www.forbes.com/sites/chloesorvino/2022/09/02/us-farmers-struggle-through-drought-to-bring-food-to-the-table-but-face-more-challenges-ahead/?sh=77aac17fc084.

[6] T. N. Liliane and M. S. Charles, "Factors Affecting Yield of Crops," in *Agronomy*, Amanullah, Ed., Rijeka: IntechOpen, 2020, ch. 2. DOI: 10.5772/intechopen.90672.

[7] *Nitrogen, Phosphorus, Potassium Fertilizer Promoting Plant Growth*. [Online]. Available: https://www.fertilizer-machines.com/solution/fertilizer-technology/how-N-P-K-fertilizer-promote-plant-growth.html.

[8] *Smart farming ag-tech — ChrysaLabs*. [Online]. Available: https://www.chrysalabs.com/.

[9] *Smart Farming system and Precision Agriculture software by GeoPard*. [Online]. Available: https://geopard.tech/.

[10] *Mapware for Agriculture*. [Online]. Available: https://mapware.com/mapware-for-agriculture/.

[11] *DJI Agriculture - Drones Better Growth, Better Life*. [Online]. Available: https://ag.dji.com/.

[12] S. Postolache, P. Sebastião, V. Viegas, O. Postolache, and F. Cercas, "IoT-Based Systems for Soil Nutrients Assessment in Horticulture," *Sensors 2023, Vol. 23, Page 403*, vol. 23, p. 403, 1 Dec. 2022. DOI: 10.3390/S23010403.

[13] B. Cheruvu, S. B. Latha, M. Nikhil, H. Mahajan, and K. Prashanth, "Smart Farming System using NPK Sensor," *2023 9th International Conference on Advanced Computing and Communication Systems, ICACCS 2023*, pp. 957–963, 2023. DOI: 10.1109/ICACCS57279.2023.10112795.

[14] J. Aliparo, C. D. Santos, C. M. Sese, *et al.*, "IoT-based Assessment and Monitoring of NPK Content and Fertility Condition of Soil," *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2022-November, 2022. DOI: 10.1109/TENCON55691.2022.9978040.

[15] M. Pyingkodi, K. Thenmozhi, M. Karthikeyan, T. Kalpana, S. Palarimath, and G. B. A. Kumar, "Iot based soil nutrients analysis and monitoring system for smart agriculture," *3rd International Conference on Electronics and Sustainable Communication Systems, ICESC 2022 - Proceedings*, pp. 489–494, 2022. DOI: 10.1109/ICESC54411.2022.9885371.

[16] L. K. Sundari, M. Rana, S. T. Ahmed, and K. Anitha, "Real-time iot based temperature and npk monitoring system sugarcane-crop yield for increasing," *3rd IEEE International Virtual Conference on Innovations in Power and Advanced Computing Technologies, i-PACT 2021*, 2021. DOI: 10.1109/I-PACT52855.2021.9696564.

[17] R. Madhumathi, T. Arumuganathan, and R. Shruthi, "Soil NPK and Moisture analysis using Wireless Sensor Networks," *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*, Jul. 2020. DOI: 10.1109/ICCCNT49239.2020.9225547.

[18] *ESP32 PICO Series Datasheet*, Version 4.4, Espressif Systems, 2023. [Online]. Available: www.espressif.com.

[19] *What are LoRa and LoRaWAN?* [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/.

[20] *New LoRa world record: 1336 km / 830 mi*. [Online]. Available: https://www.thethingsnetwork.org/article/new-lora-world-record-1336-km-830-mi.

[21] F. Guan, M. Xia, X. Tang, and S. Fan, "Spatial variability of soil nitrogen, phosphorus and potassium contents in Moso bamboo forests in Yong'an City, China," *CATENA*, vol. 150, pp. 161–172, Mar. 2017. DOI: 10.1016/J.CATENA.2016.11.017.

[22] Y. Zhao, X. Xu, J. L. Darilek, B. Huang, W. Sun, and X. Shi, "Spatial variability assessment of soil nutrients in an intense agricultural area, a case study of Rugao County in Yangtze River Delta Region, China," *Environmental Geology*, vol. 57, pp. 1089–1102, 5 May 2009. DOI: 10.1007/S00254-008-1399-5.