Precision Irrigation with Cost-effective and Autonomic IoT Devices using Artificial Intelligence at the Edge

D2.2

Report on the development of the low cost AI-capable Smart Tensiometer

Responsible Editor: Innotec21

Contributors: Waziup e.V.

Document Reference: D2.2

Distribution: Public

Version: 1.0

Date: August, 2023

Table of Contents

1. Introduction **3**
2. Hardware architecture **4**
2.1. Overview 4
2.2. WaziSense V2 5
2.3. Microcontroller Optimization: Embracing Efficient Functionality Without AI 7
3. Software **8**
3.1. WaziGate 8
3.1.1. Introduction 8
3.1.2. WaziApp 8
3.2. Smart Tensiometer Software Architecture 10
3.3. Osirris WaziApp 11
3.4. User Interface 11
3.5. Implementation 11
3.6. Next steps and future prototypes 12
4. Casing **14**
4.1. Requirements 14
4.2. Design 15
5. Deployment **19**
5.1. Deployment of next iteration 19
5.2. Process of Deployment 19
5.3. Maintenance 19

1. Introduction

The Osirris irrigation platform introduces an innovative and cost-effective IoT and AI-based irrigation system, leveraging the Edge computing architecture. This report revolves around the development of an advanced irrigation system that adeptly operates at the edge level. Integral to this system is an inference engine that facilitates the precise calculation of optimal water quantities and irrigation schedules.

At the core of the Osirris platform is the "Smart Tensiometer," an embedded sensor system engineered to seamlessly determine local soil water content and plant water stress. These Smart Tensiometers showcase the ability for automatic calibration and corrective actions. With the integration of micro-controllers, the Smart Tensiometer becomes proficient in essential functions, including data collection and transmission. The collected data plays a dual role: it not only contributes to model refinement through further training but also facilitates adaptation to the local environment, thereby simplifying installation procedures and enhancing the approach's broader acceptance.

This report provides an extensive exploration of the Smart Tensiometer's architecture and design. The content is structured into five pivotal sections: Hardware architecture, Software, Casing and Deployment.

2. Hardware architecture

2.1. Overview

In the following we want to introduce the second iteration of the Smart tensiometer. We will elaborate about its components and give some insights about the functionality and its abilities. Below there is an overview of the used updated board:

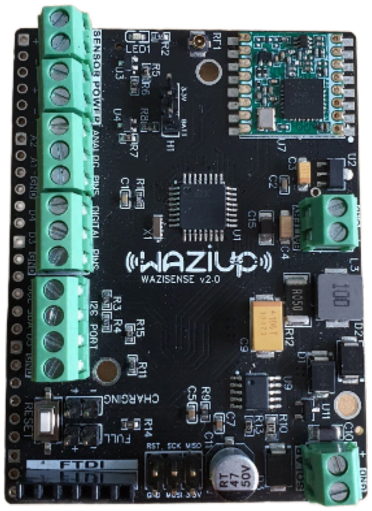

Figure 1: New micro controller board: WaziSenseV2

The Smart Tensiometer distinguishes itself within the domain of embedded systems and IoT solutions through its unique amalgamation of state-of-the-art features. Among its standout attributes are the integration of LoRa communication technology, an MPPT solar charge controller, and the utilization of the ATmega328P microcontroller. This convergence of advanced functionalities not only distinguishes the Smart Tensiometer from its counterparts but also unlocks a range of benefits that drive efficiency, sustainability, and adaptability.

1. LoRa Communication Technology: Paving the Way for Seamless Connectivity

The integration of LoRa (Long Range) communication technology is a hallmark of the Smart Tensiometer's uniqueness. This wireless communication protocol facilitates long-range and low-power data transmission, enabling the Smart Tensiometer to effortlessly connect with gateways and central hubs even in remote or challenging environments. The LoRa capability translates to extended communication ranges, reduced data transmission costs, and minimized power consumption—features that collectively enhance the system's effectiveness across diverse applications.

2. MPPT Solar Charge Controller: Optimizing Energy Utilization

A pivotal aspect that sets the Smart Tensiometer apart is its MPPT (Maximum Power Point Tracking) solar charge controller. This innovative inclusion eliminates the need for separate hardware components to manage solar panels and rechargeable batteries. The MPPT technology optimizes energy extraction from solar panels, ensuring that the system operates at its peak efficiency even in variable lighting conditions. By maximizing solar energy conversion, the Smart Tensiometer contributes to sustainable operations, extended deployment periods, and reduced reliance on external power sources.

3. ATmega328P Microcontroller: Empowering Efficient Processing

At the core of the Smart Tensiometer's operations lies the ATmega328P microcontroller—a choice that underscores the emphasis on efficiency, reliability, and versatility. Renowned for its balance between computational power and power consumption, the ATmega328P enables the Smart Tensiometer to execute tasks seamlessly without compromising energy efficiency. This microcontroller's incorporation aligns with the system's commitment to streamlined design, enabling the efficient collection and transmission of data from sensors to gateways.

Collectively, the Smart Tensiometer's unique selling point lies in the seamless integration of LoRa communication, MPPT solar charge control, and the ATmega328P microcontroller. This composition of advanced functionalities positions the Smart Tensiometer as a solution that optimizes data communication, energy utilization, and computational efficiency. In applications spanning agriculture, environmental monitoring, and beyond, the WaziSense emerges as a powerful tool that not only simplifies operations but also contributes to sustainable practices and the efficient management of valuable resources.

2.2. WaziSense V2

The Smart Tensiometer includes the new version of the WaziSense board, the WaziSense V2. This new version represents a significant advancement from its predecessor, the WaziSense V1. The first noteworthy improvement is the incorporation of an MPPT (Maximum Power Point Tracking) solar charge controller. This enhancement eliminates the need for extra hardware to connect a solar panel and rechargeable battery, simplifying the setup process while promoting energy efficiency.

The WaziSense V2 is 9.1% smaller, making it a more compact and versatile solution. During sleep mode with the Solar circuit active, it demonstrates an outstanding efficiency improvement, being approximately 23 times more power-efficient than its predecessor. Even with the Solar circuit inactive, the WaziSense V2 maintains a remarkable advancement, being approximately 105 times more power-efficient.

	WaziSense v1	WaziSense v2	
		with Solar Circuit On	with Solar Circuit Off
Regulator Max Output Current	150mA	400mA	
Regulator Max Input Voltage	20v	25v	
Sleep Current	680 µA	29 µA	6.5 µA
Idle Current	5.03 mA	5.8 mA	
Transmission Current	86.9 mA	80.3 mA	92.3.5 mA
Digital Pins	4	3	
Analog Pins	4	7	

PWM Pins	3	2
External Interrupt Pins	0	1

Table 1: Comparison of energy relevant attributes of different iterations of the WaziSense

This heightened efficiency not only extends operational periods but also contributes significantly to energy optimization, aligning with our commitment to sustainability and resource conservation.

Furthermore, the WaziSense V2 enhances its usability by providing users access to 3 digital pins, 2 PWM pins, and 7 analog pins. It also supports solar panels and lithium-ion/polymer batteries, enabling greater flexibility and functionality in various applications.

In a move toward enhanced efficiency, the inclusion of the relay for pump control has been reevaluated. Recognizing the primary usage of these microcontrollers for data gathering, the relay component has been omitted. This decision is grounded in the realization that, in most scenarios, a single device suffices for pump control. By minimizing redundant components, the design is further streamlined and optimized for its core functions.

While it's worth noting that the first iteration of the WaziSense (V1) can still accommodate pump control through the relay, out of the box. In cases where pump control is a requirement, the WaziSense V1 can still be used. But it is also possible to connect and actuate a relay with the WaziSense V2. This is particularly relevant as the Smart Tensiometer is envisioned for applications where the microcontroller often accompanies a readily available power source for the pump.

The developments outlined above encapsulate the evolution of the WaziSense (V2) and its deliberate optimization for efficiency, versatility, and improved functionality over its predecessor.

2.3. Microcontroller Optimization: Embracing Efficient Functionality Without AI

Microcontrollers play a pivotal role in the functionality of embedded systems, serving as the interface between sensors and higher-level processing. In recent developments, the decision to exclude AI models from microcontrollers and instead delegate these processing tasks to Microcontroller Optimization: Embracing Efficient Functionality Without AIo gateways has emerged as a strategic approach. This approach is underpinned by several compelling reasons, each contributing to the enhancement of system efficiency, resource allocation, and overall performance.

Resource Conservation: Microcontrollers, by design, possess limited computational resources compared to their gateway counterparts. Including AI capabilities within microcontrollers can lead to resource exhaustion and suboptimal performance. By redirecting AI tasks to gateways, microcontrollers can preserve their simplicity, operate within resource constraints, and remain cost-effective. This is especially crucial in applications where cost and energy efficiency are paramount concerns.

Simplified Design: An essential advantage of microcontrollers is their straightforward architecture. By avoiding the integration of AI models, their design can remain uncomplicated. This not only reduces manufacturing complexity but also streamlines troubleshooting and maintenance procedures. As a result, microcontrollers become more reliable and accessible components in embedded systems.

Reduced Complexity: The absence of AI processing on microcontrollers minimizes their computational load. This, in turn, facilitates the swift and accurate forwarding of raw sensor data to gateways. Microcontrollers can thus focus on their primary role of data collection and communication without being encumbered by additional processing tasks, optimizing their performance in data transmission.

Gateway Optimization: Gateways, equipped with more substantial computational resources, can efficiently handle complex AI algorithms. Centralizing AI processing at the gateway level enables more sophisticated data analysis, trend identification, and predictive modeling. This empowers the system to make informed decisions based on aggregated data, resulting in higher-level optimization that microcontrollers alone cannot achieve.

Enhanced Data Management: Gateways serve as centralized data hubs, allowing for comprehensive data aggregation, management, and optimization. By consolidating data from multiple microcontrollers, gateways can apply advanced analytics, pattern recognition, and optimization algorithms to yield insights that microcontrollers operating independently cannot provide.

Scalability and Flexibility: The decentralized approach allows for seamless scalability of the system. Additional microcontrollers can be integrated without the need for complex modifications, as the gateway retains the primary AI processing responsibilities. This scalability facilitates the expansion of the system's capabilities without compromising its efficiency.

In conclusion, the decision to exclude AI processing from microcontrollers in favor of gateway-based computation represents a strategic approach that enhances overall system efficiency and performance. By delegating AI tasks to gateways, microcontrollers maintain their simplicity, conserve resources, and operate optimally within their intended scope. This optimization strategy not only reduces complexity but also ensures cost-effective and reliable operation in embedded systems.

3. Software

3.1. WaziGate

3.1.1. Introduction

WaziGate is a versatile IoT LoRa Gateway designed to meet diverse remote IoT needs. It can connect up to 100 IoT sensors and actuator nodes using a robust LoRa radio network, making it suitable for various applications like weather monitoring, soil analysis, GPS tracking and much more.

What sets WaziGate apart is its edge capacity, allowing you to host applications directly within the gateway for a responsive and tailored environment. It not only gathers data but also enables control of actuators such as electro-valves.

Flexibility is a key feature, with the ability to host applications within the gateway and enhanced connectivity through WiFi. It boasts a long LoRa communication range of up to 10-12 kilometers and establishes a WiFi hotspot for device connectivity.

WaziGate adapts to different connectivity options, supporting WiFi, 3G, and Ethernet connections, ensuring continuous connectivity regardless of infrastructure availability. It's energy-efficient, optimizing performance while conserving power.

Automation capabilities streamline IoT network operations, enabling devices to communicate intelligently. Remote management allows real-time adjustments, even without internet access, thanks to its embedded database and web-based visualization module.

In summary, WaziGate is a powerful IoT LoRa Gateway that combines connectivity, control, data access, and advanced features like edge hosting, extended communication range, low power consumption, and remote management. Its versatility and resilience in limited connectivity situations make it an essential asset in the IoT landscape.

3.1.2. WaziApp

WaziApps are applications developed specifically for the Wazigate platform, which is an IoT LoRa Gateway designed for remote IoT applications. These apps are designed to enhance the capabilities of the Wazigate by extending its functionality to cater to various use cases. The core functionality of WaziApps revolves around leveraging the microservice architecture of Wazigate to create independent, isolated applications that can run on the gateway.

Key aspects of WaziApps' core functionality include:

- **Microservice Architecture:** WaziApps operate using a microservice architecture. Each app functions as an independent microservice, running in its own isolated container. This architecture enhances development, maintenance, and scalability by allowing apps to operate autonomously.
- **Docker Containers:** WaziApps are encapsulated within Docker containers. This encapsulation ensures that each app runs independently of other apps, preventing potential conflicts or interference. These containers are conveniently accessible through the Wazigate's user interface, allowing users to directly retrieve them from Docker Hub. This streamlined process eliminates the need for manual configuration and enables users to easily deploy the desired containers for

their WaziApps. By offering this direct integration with Docker Hub, Wazigate simplifies the deployment of applications and enhances the user experience.

- **Custom Development:** WaziApps offer developers the flexibility to create custom applications to suit their specific needs. Whether it's a weather station, soil monitoring, GPS application, or any other IoT-related functionality, WaziApps can be tailored accordingly.
- **Multi-Language Support:** WaziApps can be developed using various programming languages, including Python, GoLang, and Javascript. This enables developers to work with the language they are most comfortable with, promoting productivity and creativity.
- **API Creation:** WaziApps allow developers to create APIs to interact with the app's functionality. Also the WaziGates API enables communication between the app and other components of the system, facilitating data exchange and control.
- **Local Hosting:** WaziApps can host their applications directly on the Wazigate. This means that the gateway can operate as a server for the developed applications, providing a self-contained environment for running and managing apps.
- **Remote Management:** Apps running on the Wazigate can be managed remotely. This enables developers to update, maintain, and monitor their applications from a distance.
- **Data Visualization:** WaziApps can provide data visualization capabilities, allowing users to visualize collected data through web-based interfaces. This visualization aids in data analysis and decision-making.
- **Extensibility:** As the possibilities are endless, WaziApps can be extended to encompass various functionalities and use cases. Whether it's gathering environmental data, controlling actuators, or other IoT-related tasks, WaziApps can adapt to different scenarios.

In summary, WaziApps uses Wazigate microservice architecture to enable developers to create customized, independent applications for a wide range of IoT applications. This versatility empowers users to develop solutions that cater to their specific requirements, enhancing the overall capabilities of the Wazigate platform.

3.2. Smart Tensiometer Software Architecture

At the core of the Smart Tensiometer's functionality stands its embedded software, a pivotal element that orchestrates a series of essential tasks. The embedded software undertakes the following key responsibilities:

1. Measurement Execution:

The software efficiently manages the process of measuring soil tension through the Smart Tensiometer's internal sensors. It controls sensor activation, oversees data acquisition, and oversees the conversion of raw data into meaningful measurements.

2. Calibration for Accuracy:

Following the acquisition of raw data, the software applies calibration algorithms to transform the measurements into calibrated centibar values. This calibration process ensures the accuracy of soil tension readings, providing dependable insights for agricultural analysis.

3. LoRa Data Transmission:

Once calibrated centibar values are determined, the software leverages LoRa communication technology for wireless data transmission to a LoRa Gateway. It formats the data into LoRa-compatible packets and manages the transmission process to ensure efficient and reliable data transfer.

4. Efficient Energy Management:

The software incorporates energy management strategies to optimize the Smart Tensiometer's battery life. It monitors the battery voltage level and implements energy-saving measures to extend the operational duration.

5. Low Battery Adaptation:

When the battery level reaches a low threshold, the software adapts by adjusting the transmission interval. It reduces the frequency of data transmission to conserve energy and extend the device's operational capability, ensuring functionality even under limited power conditions.

By seamlessly handling these tasks, the embedded software within the Smart Tensiometer plays a pivotal role in enabling accurate measurements, dependable data transmission, and effective energy management. It ensures that the device operates optimally, providing valuable soil tension data that contributes to informed agricultural monitoring and decision-making processes.

The microcontroller code is written in the C programming language. For easy access and collaboration, we have created a GitHub repository where you can find the code using the following link:

GitHub repository

3.3. Osirris WaziApp

The initial version of the smart irrigation application for the WaziGate is focused on providing farmers with actionable guidance and predictions for optimal watering practices within a reasonable forecast horizon. This predictive functionality revolves around the **farmer's pre-set soil tension threshold.**

The application integrates various crucial factors into its decision-making process. It begins by collecting data from the WaziSense-attached sensor probes, encompassing soil moisture and temperature readings. From these readings, **the volumetric water content is derived**. This calculation can be customized based on the specific soil type or through the use of a tailored soil water retention curve. Additional features are also being engineered, including the status of the pump and various moving averages derived from the sensor data.

Supplementary data is fetched through an API, providing **historical weather information** for the farm's GPS location. This data encompasses temperature, humidity, rainfall, cloud cover, shortwave radiation, wind speed, wind direction, soil temperature, soil moisture, and evapotranspiration (Et0). **Forecasts for this meteorological data are also accessible**, contributing to the application's predictive capabilities.

Following an initial warm-up period, during which the farmer has the option to train the model with pump state data (later automated detection can be implemented), the system becomes proficient in making predictions when specific soil moisture thresholds are reached. This prediction process takes into account various environmental variables.

The model undergoes periodic training cycles, during which multiple models are compared, tuned, and evaluated using ensemble techniques. The best-performing model is selected based on its performance on an evaluation set that was not used during training. This iterative process ensures that the predictions stay updated and incorporate the latest changes in the environment.

3.4. User Interface

The application's user interface comprises two primary pages:

Configuration Page: This page enables users to set up essential parameters:

1. Soil type or a custom soil water retention curve.
2. A crop coefficient curve with four values, aiding in crop evapotranspiration calculation.
3. A threshold indicating when the system would typically apply watering.

Overview Page: This page offers farmers comprehensive visualizations, including:

1. Diagrams illustrating future weather predictions.
2. Soil moisture predictions generated by the system.

By considering a range of variables and incorporating predictive modeling, the smart irrigation application empowers farmers with insights and recommendations, ultimately contributing to more informed and efficient watering practices.

3.5. Implementation

Regarding the WaziApp for smart irrigation, the backend of this application is developed using Python, harnessing a suite of powerful packages to achieve its functionality. Here's a brief overview of the key packages utilized:

- **PyCaret:** PyCaret is a versatile machine learning library that streamlines the process of model training and deployment. It offers automation for various machine learning tasks, making it an efficient choice for building predictive models in a user-friendly manner.
- **Scikit-learn (sklearn):** Scikit-learn is a renowned machine learning library that provides a comprehensive range of tools for data analysis, modeling, and predictive analysis. It offers an array of algorithms for classification, regression, clustering, and more.
- **XGBoost:** XGBoost is a popular gradient boosting library that excels in boosting and ensemble learning. It's particularly effective for improving the performance of machine learning models, making it a valuable addition to the application's toolkit.
- **Pandas:** Pandas is a fundamental library for data manipulation and analysis. It offers data structures and functions that simplify working with structured data, enabling efficient data preprocessing and transformation.
- **NumPy:** NumPy is a fundamental package for numerical computing in Python. It provides support for arrays, matrices, and mathematical functions, making it essential for efficient handling of large datasets.
- **Matplotlib:** Matplotlib is a widely used plotting library that facilitates the creation of various types of visualizations, aiding in data exploration and presentation.
- **Subprocess:** The Subprocess module allows the application to spawn new processes, connect to their input/output/error pipes, and obtain return codes. It's useful for executing external commands and processes from within the Python code.
- **JSON:** JSON (JavaScript Object Notation) is a lightweight data interchange format. It's employed for encoding and decoding structured data, enabling seamless communication between different components of the application.
- **Datetime:** The Datetime module provides classes for manipulating dates and times. It's crucial for managing timestamps, durations, and other temporal data within the application.
- **Missingno:** Missingno is a visualization library tailored for identifying and visualizing missing data in datasets. This aids in understanding data completeness and quality.

On the frontend side, the smart irrigation application is likely to be implemented as a **Flask web application**. This micro web framework enables the creation of interactive and dynamic web interfaces. The application's frontend will utilize **Apex Charts**, a visualization library, to render a variety of informative charts and graphs, enhancing the user experience. Configuration settings will be stored in a **JSON file**, allowing users to easily customize and tailor the application's behavior to their specific requirements.

This integration of Python packages and frontend technologies results in a robust and user-centric smart irrigation application, designed to provide accurate predictions and guidance for optimal irrigation practices.

3.6. Next steps and future prototypes

In the roadmap for future prototypes of the smart irrigation WaziApp, the aim is to achieve a comprehensive level of automation for the irrigation system, streamlining the process and reducing manual intervention. This progression involves incorporating key components to enhance system functionality:

WaziAct Integration: The addition of a WaziAct module serves as a pivotal component to automate the irrigation process. The WaziAct module controls the pump, ensuring efficient and timely water delivery to the plants. This integration marks a significant step toward achieving a fully automated irrigation system.

Flow Meter Integration: The inclusion of a flow meter introduces the capability to accurately measure the amount of water delivered to the plants. This data provides crucial insights into water consumption and usage, aiding in optimizing irrigation strategies and resource management.

Complete Automation: With the combination of the WaziAct module and the flow meter, the irrigation system can be fully automated. The system will autonomously monitor soil moisture levels, weather forecasts, and other relevant factors to determine the optimal irrigation schedule. This minimizes the need for constant manual oversight and intervention by farmers.

Visual Detection via Camera: To further enhance the automation, the integration of a camera system could be considered. This camera could be strategically positioned to oversee specific plants. By leveraging object detection techniques on a real-time video stream, the system can identify and assess plant conditions. This enables remote monitoring and eliminates the need for physical presence for visual inspections.

Plant Stress Detection: Another potential advancement is the training of models to detect plant stress indicators through image analysis. By incorporating machine learning algorithms, the system can identify signs of plant stress, such as discoloration or wilting. These findings can serve as additional inputs for the predictive model, allowing it to refine its irrigation recommendations based on real-time plant health data.

Continuous Improvement: The iterative nature of the system enables continuous improvement. As the system gathers more data over time, it can refine its predictions and recommendations. By incorporating feedback loops and machine learning techniques, the overall system becomes more accurate and tailored to the specific needs of the plants.

By integrating these components and capabilities, the future iterations of the smart irrigation WaziApp aim to create a highly automated and intelligent irrigation system. This evolution not only optimizes water usage but also empowers farmers with valuable insights and control over their agricultural practices, paving the way for increased efficiency and sustainability in irrigation management.

4. Casing and fixation system

The further development of the Smart Tensiometer goes beyond its core functions and includes, among other things, a comprehensive design of its housing and holder system. This pivotal upgrade marks a departure from off-the-shelf enclosures to a meticulously crafted custom casing that addresses specific needs while advancing the system's usability and reliability. While acknowledging the cost-effective nature of pre-existing casings, the decision to embark on this redesign journey stemmed from the critical evaluation of the Smart Tensiometer's operational efficiency and overall user experience.

4.1. Requirements

The following requirements list include requirements for the electronic component casing and fixation system (e.g. stick and sensor fixation).

#	Title	Requirement
1	Waterproof	The ST casing must be waterproof in accordance with the IP55 rating.
2	Cables protection	The ST casing must ensure that internal cables are shielded from direct exposure to sunlight and potential interference from animals.
3	Solar panels	The ST casing must incorporate space and support for the integration of a solar panel.
4	Battery holder	The ST casing must have a designated compartment with a battery holder specifically designed to accommodate a 18650 battery.
5	UFL connector	The ST casing must include a provision for a UFL connector.
6	Antenna	The ST casing should allow for the internal placement of an antenna.
7	Microcontroller fixation	The ST casing must include a provision for securing the mainboard.
8	Microcontroller removability	The ST casing design should enable convenient removal of the microcontroller for maintenance or replacement purposes.
9	Durability	The ST casing must exhibit resistance to the potentially damaging effects of UV radiation emitted by the sun, ensuring long-term durability.
10	Installation	The ST casing should be easy to install by planting it in the ground
11	Sensor fixation	The ST casing should have provisions to fix and protect the sensors in the ground.
12	Sensor depth	The ST casing should have indications to help deploying the sensors at given depths.

4.2. Design

The newly designed case for the microcontroller, slated for field deployment, is engineered to fulfill all previously delineated prerequisites. Driven by the requirements mandated, the development of a customized case design became compulsory. This enclosure was modeled from scratch within Blender and subsequently printed, using a 3D printer. In the following there are some renderings of the case illustrated:

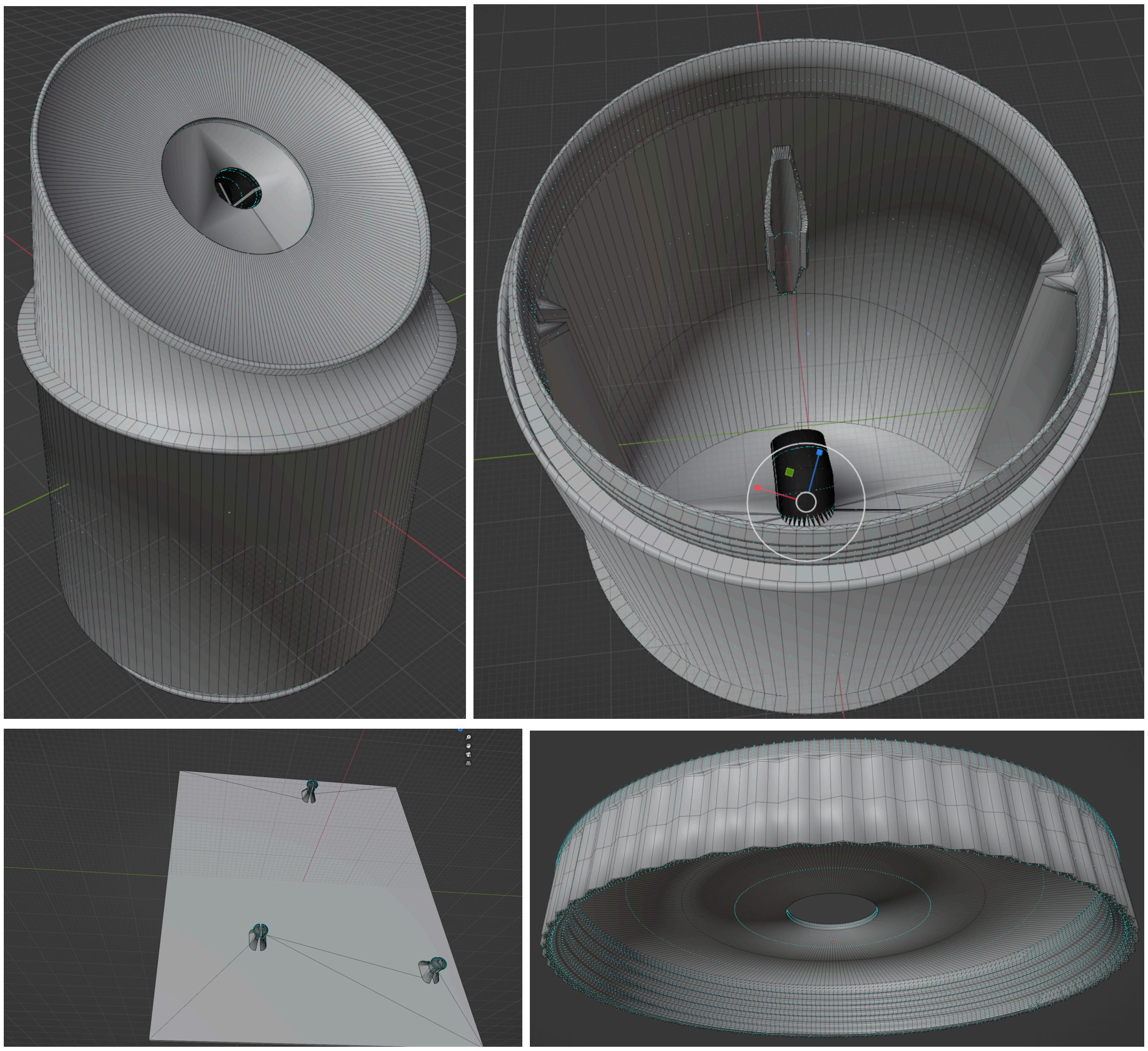

Figure 2: Renderings of the 3D model in Blender showing the design of the case.

Below, you will find an illustration showcasing the initial printed prototypes:

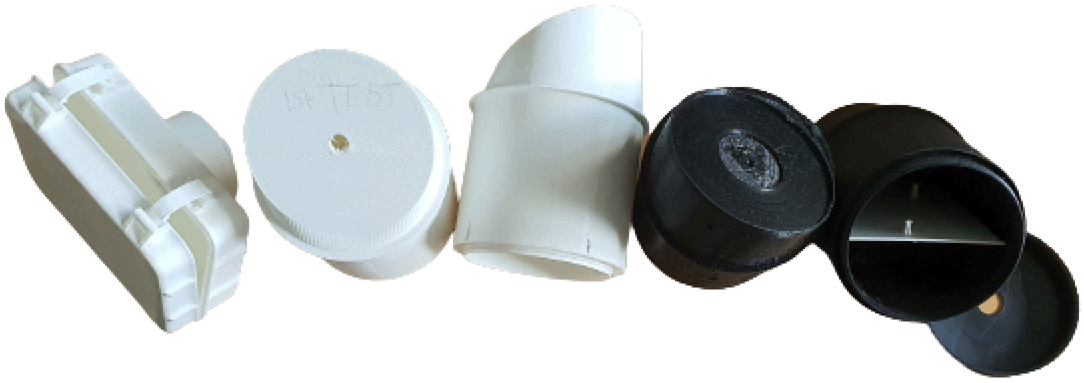

Figure 3: Selected iterations of the designed case for WaziSenseV2.

After many design iterations and various tests, we came to the conclusion that the design below is best fitted to address the various requirements.

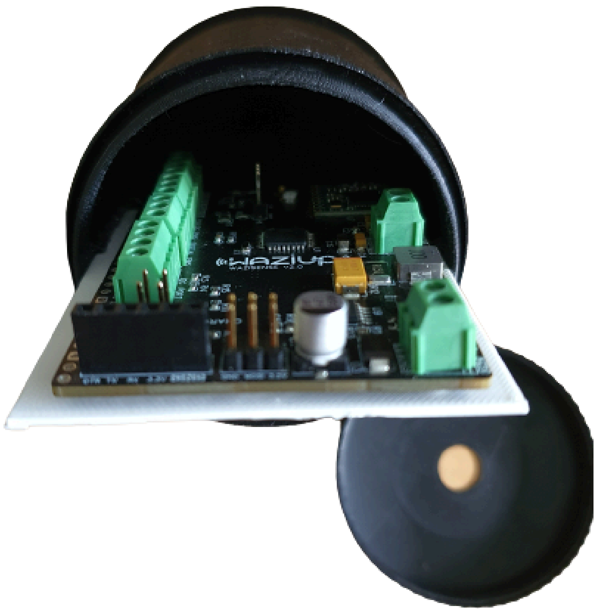

Figure 4: Last iteration from below with opened service cover, for easy access.

An objective in the case design was to guarantee IP55-level dustproofness and watertightness. This led to the decision not to expose electrical wiring in areas exposed to the environment. To fix the solar panel in place, silicone glue is applied to ensure watertightness. A significant portion of the case's profile is concealed beneath a 75mm PVC sewage pipe, a globally accessible component at an affordable price point. This pipe serves a dual purpose, functioning both as a protective cover and a stable platform. Notably, this design element acts as a conduit for routing cables connected to sensors, safeguarding them against sunlight exposure and potential disruption from animals.

Addressing sustainability, the case incorporates a compact solar panel, a vital component for battery recharging and the attainment of self-sufficiency. Another notable evolution pertains to the antenna placement, now internalized within the case structure. This design enhancement involves a clipping mechanism that firmly anchors the antenna in a fixed position, facilitating easier water resistance implementation.

The prototype significantly enhances installation and maintenance convenience. The case's innovative bottom-opening design allows straightforward access to all internal components. The cables are routed through a waterproof cable connector, this ensures that no moisture from the ground is getting into the case. Further streamlining this process is the integration of a sliding sledge, hosting the WaziSense microcontroller and the 18650 battery holder, allowing them to be effortlessly extracted from the case.

Resilience against environmental factors, particularly UV rays, emerged as a pivotal concern. To fortify the case against these external influences, we use acrylonitrile styrene acrylate (ASA) material, renowned for its exceptional resistance to environmental stressors.

This comprehensive approach underscores our commitment to developing a solution that satisfies in performance, durability and adaptability, ensuring seamless integration of the microcontroller under field conditions.

5. Deployment

This section gives details about future deployments and the process involved, of setting a system up.

5.1. Deployment of next iteration

We are targeting the deployment of the new sensor devices by December of this year, at a site in Tunisia as suggested by our partners. In this iteration of the prototype, our goal is to establish a system that can predict soil water tension or soil water content in advance, while also factoring in environmental (meteorological) conditions.

5.2. Process of Deployment

The deployment process can be broken down into several straightforward subtasks:

1. **Device Installation:** Place a sufficient number of devices (2-4) in the field, this is dependent on the homogeneity of the soil in the field and the size of the field.
2. **Device Placement:** Dig a hole for each device, insert the stand and sensor, mix the soil with water, and then refill the hole. This setup ensures proper contact between the sensor and the soil. The depth of the sensor placement should be aligned with the main root zone of the plant that is being monitored. More information on that is contained in the installation guide by irrometer.
3. **Data Collection:** Allow 2-5 days for data collection, during which the devices gather necessary information to train the machine learning models.
4. **Gateway Setup:** Install and set up the Gateway near the field. This requires access to electricity and an intern
5. **Remote Access:** Establish a connection to WaziCloud for remote access and monitoring capabilities.
6. **LoRa Connection:** Establish connections to all sensor devices via LoRa communication.
7. **App Activation:** Start the WaziApp and follow the provided instructions for configuration and operation.

By following these steps, the deployment of the sensor devices and the associated system can be completed efficiently and effectively.

5.3. Maintenance

Incorporating a rechargeable battery and solar panel in this version significantly reduces the need for maintenance efforts. The main focus is to avoid any hindrance to the solar panels caused by factors like sunlight blockage, accumulation of dust, dirt, or nearby vegetation. Periodic inspections are recommended, such as checking the stability of the pole, which is typically done once per season, to ensure it remains securely positioned.

The battery within this device boasts an average lifespan of approximately 8-10 years, while the other components are projected to have even longer lifespans. This design choice ensures the longevity and reliability of the system, contributing to its overall efficiency and minimal maintenance requirements.