I.BA_IOT.H24: Final Report

December 21, 2024

IoT-Based decision support system for slurry and manure application in agriculture

Lukas Lottenbach, Jan Pollinger, Sandro Troxler and Flavio Waser
HSLU, Informatic Technology Department, Switzerland
{lukas.lottenbach, jandominic.pollinger, sandro.troxler, flavio.waser}@stud.hslu.ch

1 Abstract

The IoT-based Decision Support System leverages sensor devices to optimize slurry and manure application in agriculture. Key sensors monitor critical environmental factors such as temperature and air humidity. This sensor data is transmitted to the cloud, where it is stored and managed. The users can check optimal conditions are met for manure spreading on a website.

2 Introduction

2.1 Description

The aim of this project is to create an IoT-based decision support system that assists farmers in determining the optimal time to spread slurry or manure on their fields. The system leverages various environmental sensors to gather real-time data on atmospheric and soil conditions, such as air temperature, humidity, air pressure, and soil temperature. This data is processed and analyzed using Raspberry Pi and then transmitted over the internet to a centralized platform for evaluation.

2.2 Motivation

Our project team is motivated by the need to support farmers in making it more efficient and sustainable decisions when applying slurry and manure to their fields. By leveraging IoT technology, we aim to provide real-time, data-driven insights that help farmers optimize the timing of their spreading activities, ultimately saving resources and enhancing crop productivity. Additionally, our system promotes environmentally responsible farming by reducing the risk of nutrient runoff and soil degradation, ensuring that natural fertilizers are used effectively. Through this project, we strive to make agricultural practices more sustainable, cost-effective, and environmentally friendly.

3 Related Work

Research is advancing toward optimizing fertilizer application by tailoring the amount to specific areas of each field. This approach aims to enhance fertilization efficiency. Using satellite data, site-specific fertilization was applied to winter wheat, with the goal of reducing nitrogen surplus and improving plant nutrient uptake.

Central to this method is an application map generated from satellite data, which builds a dataset over the years to identify areas where fertilization is most effective and where fertilizer can be applied sparingly. However, an automated solution for accurately measuring soil nutrient reserves is not yet available.

(Latsch & Kramer, 2023)

The method was tested across seven fields, and the results are promising. Nitrogen application was reduced by 23% compared to conventional practices, with no negative impact on yield. While no direct effect on the wheat's protein content was observed, nitrogen surpluses—represented by nitrogen not absorbed by the plants—were reduced by 30% on the test plots.

Digital technologies are essential for capturing and managing the complex processes occurring in the field. The ultimate goal is to produce more efficiently, use available resources responsibly, and minimize the environmental impact of fertilization.

(Anken et al., 2022)

4 System Design and Implementation

The system is designed as an IoT solution for environmental monitoring, leveraging LoRaWAN for low-power, long-range communication. This enables sensors to be deployed in remote fields and transmit data over long distances efficiently. The architecture consists of four main components:

- Sensors for data acquisition
- Microcontroller for processing and communication
- LoRaWAN network for data transmission
- Webservice with MySQL database for storing the data and for providing a graphical interface for the user

4.1 System Overview

The system is specifically designed for long-range use, making it ideal for deployment in fields, rural areas, or other remote locations where traditional communication infrastructure may be limited. At the core of the system lies an Arduino microcontroller, which collects data from a variety of environmental sensors, such as temperature, humidity and soil moisture. This data is then transmitted using LoRaWAN (Long Range Wide Area Network), a communication protocol known for its energy efficiency and ability to transmit data over long distances with minimal power consumption.

The collected sensor data is sent to The Things Network (TTN), an open-source LoRaWAN network server that serves as the bridge between the device and the backend infrastructure. TTN ensures reliable data routing and provides a secure connection between the sensor nodes and the cloud environment. To enable seamless integration, a webhook is registered on TTN, which automatically forwards the transmitted data to a PHP-based Webservice. The webservice uses a MySQL database for storage. This webservice and database are securely hosted in a Swiss datacenter, ensuring that data storage complies with local privacy regulations and guarantees high levels of reliability and security.

The MySQL database serves as the central backbone for storing, organizing, and managing the incoming sensor readings. The structured data can then be accessed and analyzed for further insights. To provide users with a functional and intuitive interface, the webservice also hosts PHP-based frontend code. This frontend acts as the visual layer of the system, allowing users to interact with the data in real-time. Through the frontend, users can monitor live sensor readings, analyze trends, enabling them to make data-driven decisions efficiently.

By combining hardware components, LoRaWAN communication, cloud-based infrastructure, and web technologies, this system creates a robust and scalable solution for environmental monitoring. Its modular design allows for easy adaptation to different applications, from agricultural monitoring to smart city projects, while ensuring reliable performance even in challenging, remote environments.

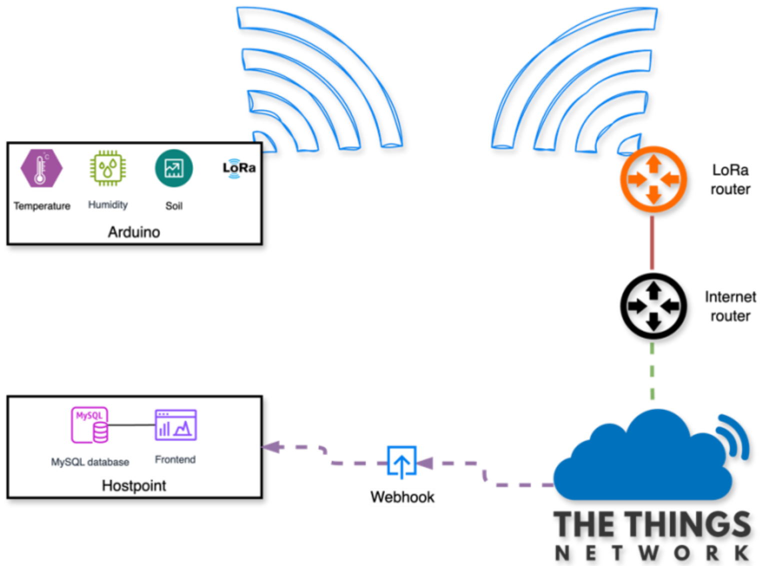


Figure 1: LoRa Overview

4.2 Innovation

The innovation behind the code lies in its practical and efficient implementation of IoT principles for environmental monitoring. Designed for long-term deployment in remote areas, the system focuses on energy efficiency, reliability, and scalability.

One standout feature is its energy-efficient design. The soil moisture sensor, for example, is powered only when readings are needed, significantly conserving battery life. Coupled with the use of the LoRaWAN protocol, the system achieves long-range communication with minimal power consumption, making it ideal for use in rural or agricultural settings.

Another innovative aspect is the way data is processed and transmitted. The system averages multiple sensor readings to smooth out noise and outliers, ensuring more reliable data. It then encodes the data into a compact 8-byte payload, optimized for LoRaWAN's bandwidth limitations. By sending this data at regular intervals the system strikes a balance between timely updates and communication efficiency.

The code also demonstrates scalability and security in IoT communication. By utilizing the LMIC library, it seamlessly integrates with existing infrastructure like "The Things Network," making it easy to expand or adapt for different use cases. Security is a priority as well, with device and application keys protecting data integrity and ensuring secure transmissions.

Tailored for agricultural applications, the system combines soil moisture, temperature, and humidity monitoring into a single, low-cost solution. Its ability to operate autonomously for extended periods with minimal maintenance makes it a practical tool for addressing real-world challenges in remote and resource-constrained environments.

4.3 System Architecture

The system architecture is a layered design combining IoT hardware, communication protocols, cloud-based processing, and a user-facing interface. The architecture can be divided into four main components:

4.3.1 Sensor Layer (Data Acquisition)

The system integrates multiple sensors:

- Si7021 temperature and humidity sensor: Measures ambient conditions like temperature and humidity.
- SEN-13322 Soil moisture sensor: Evaluates soil wetness and is powered intermittently to conserve energy.

These sensors are connected to an Arduino-compatible microcontroller via I²C and analog interfaces. The microcontroller collects sensor readings and processes the data. To ensure efficient communication, the data is encoded into an 8-byte payload. Additionally, the microcontroller manages peripheral power control, such as powering the soil moisture sensor only during readings to optimize energy consumption.

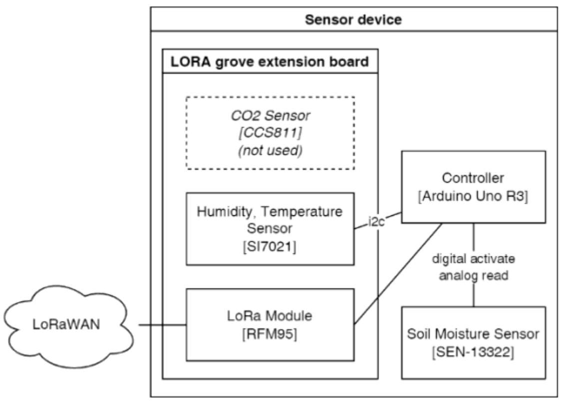


Figure 2: Sensor modules overview

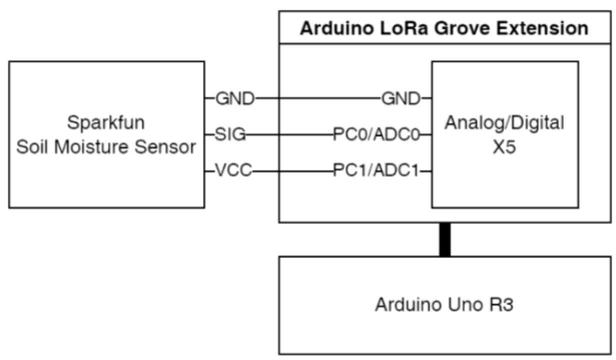


Figure 3: Wiring of soil moisture sensor

4.3.2 Processing Layer (Microcontroller)

Hardware: Arduino microcontroller

Function:

- Reads and processes sensor data
- Averages and encodes data into an 8-byte payload to ensure efficient transmission
- Manages peripherals and sensor power for energy optimization

4.3.3 Communication and Data Transmission

The LoRaWAN protocol is used for data transmission, implemented using the LMIC library. The Arduino prepares the processed data and transmits it to a LoRa gateway provided by Group 1. This gateway forwards the data packets to "The Things Network" (TTN), a cloud-based IoT platform. Key security measures, such as device and application keys, ensure secure data transmission.

The program initializes sensors and the LoRaWAN connection in the setup() function. During operation, the system runs in a continuous loop() where sensor readings are updated at regular intervals (every 60 seconds). The data is smoothed by averaging multiple readings before being sent. The LMIC library manages LoRa communication events such as joining the network, transmitting data, and handling acknowledgments. Each transmission includes temperature, humidity, and soil moisture data, encoded efficiently for LoRaWAN communication.

This design is energy-efficient, scalable, and ideal for deployment in remote locations like agricultural fields. The combination of efficient data encoding, reliable communication protocols, and secure transmission ensures long battery life and robust system performance.

The integration of the LMIC library for LoRaWAN communication ensures compatibility with existing infrastructure like "The Things Network." This modular approach allows the system to scale easily and integrate into broader IoT ecosystems.

4.3.4 Webservice with Backend and Frontend Layer

Backend:

- Cloud Platform: "The Things Network" manages device registration and forwards data via webhooks. Which are received by a PHP interface
- Database: A MySQL database hosted in a Swiss datacenter stores the sensor data securely

Frontend:

- PHP-based Web Application: Provides an interface for users to view and analyze data in real-time
- User Access: Displays environmental conditions, trends, and insights for field monitoring

4.3.5 Data Flow Overview

Sensors	Arduino reads raw environmental data
Processing	Arduino processes, encodes, and sends data via LoRaWAN
Transmission	Data is transmitted to TTN via a LoRa gateway
Storage	TTN webhook forwards data to the MySQL database
Visualization	PHP frontend accesses the database and presents data to users

This architecture ensures the system is scalable, energy-efficient, and suitable for remote field deployments while providing a secure and accessible platform for data monitoring.

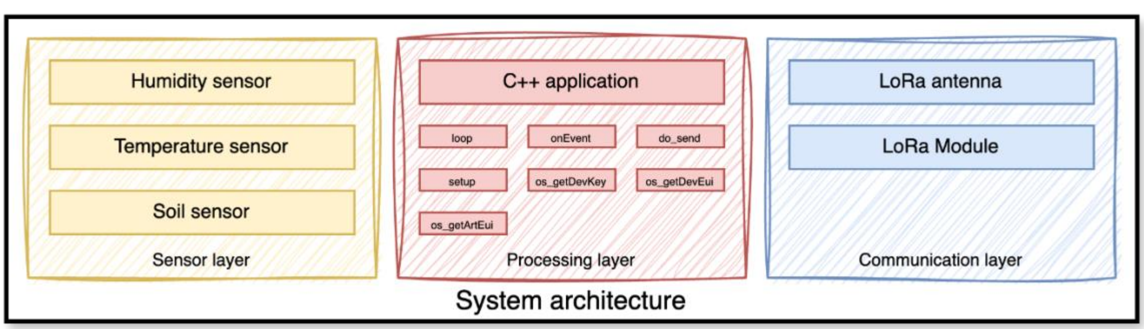


Figure 4: System architecture overview

4.4 Software Architecture Layers & Modules

4.4.1 Sensor Interface Layer

Handles the integration and communication with the environmental sensors.

Modules:

- Si7021 Module: Reads temperature and humidity data via I^2C.
- SEN-13322 Soil Moisture Module: Reads analog values from the soil moisture sensor.

Sensor Device – Public Interfaces	
Interface	**Description**
Provides	
LoRaWAN out	Sends every minute the current measurements.
Uses	
Adafruit_Si7021.h	Library for accessing temperature and humidity sensor data.
hal.h	Library for accessing pin level input and output.
SPI.h	Library for accessing pin level input and output
lmic.h	Library for LoRaWAN access.

Table 1: Sensor device interfaces

4.4.2 Processing and Data Handling Layer

Processes raw sensor data, performs calculations, and prepares it for transmission.

- Data Encoding: Converts sensor readings into a compact 8-byte payload for efficient transmission.

Implementation: Runs on the Arduino microcontroller, using C++ to integrate logic.

4.4.3 Communication Layer

Manages the transmission of data from the Arduino to the cloud.

Modules:

- LoRaWAN Protocol: We used the LMIC library to implement LoRaWAN communication. This module also manages events like network joins, data transmission, and acknowledgments.
- Data Queueing: Ensures reliable data handling, including retries if transmission fails.

Implementation: Encapsulates LoRaWAN functionality with device-specific settings for secure and efficient data transfer.

4.4.4 Backend Layer

Processes, stores, and organizes transmitted data in the cloud.

Modules:

- The Things Network (TTN): Acts as an intermediary, handling incoming data packets from the LoRa gateway
- Webhook Integration: Forwards data from TTN to a MySQL database using RESTful APIs
- Database Management: MySQL database stores sensor data securely for later retrieval and analysis

Implementation: Configured on TTN and a Swiss-hosted PHP, MySQL server.

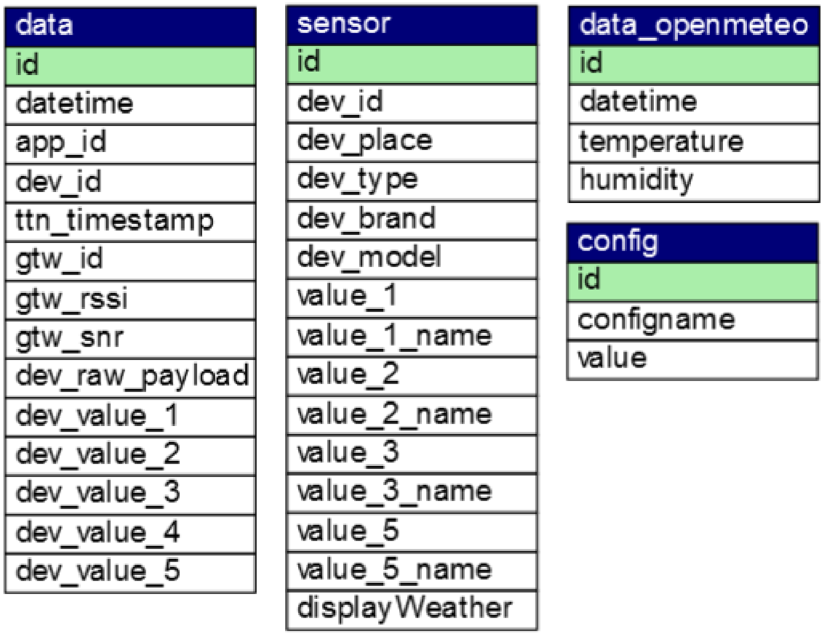


Figure 5: MySQL schema

Webservice Backend – Public Interfaces	
Interface	**Description**
Provides	
iot-database-input.php	Adds a new measurements entry into the database.
load-openmeteo.php	Reads forecast values from open-meteo and writes them into the database.
notification-slack.php	Sends values to slack for notifications.
anyboard-data.php	Returns last 500 measurements in JSON format.
anyboard-devices.php	Returns all sensor devices in JSON format.
Uses	
MySQL Database	Used for storage.
Slack	Used for notifications.

Table 2: Webservice backend interfaces

4.4.5 Frontend Layer

Provides a user interface for data visualization and interaction.

Modules:

- PHP Web Application: Retrieves data from the database and renders it into a user-friendly format.
- Visualization Tools: Displays environmental metrics like temperature, humidity, and soil moisture trends.
- User Interaction: Allows users to filter, analyze, and download data if needed.

Implementation: PHP scripts run on a server in the Swiss datacenter (Hostpoint), connected to the MySQL database.

Webservice Frontend – Public Interfaces	
Interface	**Description**
Provides	
index.php	Shows the current measurements of all sensors
chartview.php, chartview-weekly.php	Shows the history of measurements and weather forecasts of the past week in a graph for a selected sensor device
chartview-daily.php	Shows the history of measurements and weather forecasts of the current day week in a graph for a selected sensor device
device-admin.php	Shows infos of all sensor devices
device-add.php	Register a new sensor device in webservice
device-logs.php	Shows sensor devices and sensor measurements in tables
configuration.php	Configure logs and notifications
Uses	
MySQL Database	Used for storage
Bootstrap	Used for website styling
ChartJS	Used for creating graphs with JavaScript

Table 3: Webservice frontend interfaces

4.4.6 Layered Interaction Flow

Sensor Layer: Captures environmental data.

Processing Layer: Processes and prepares the data.

Communication Layer: Transmits the data to TTN.

Backend Layer: Stores and organizes data in a database.

Frontend Layer: Displays data to end-users.

Divided are the responsibilities, into layers that are distinct and modular in software architecture. Establishing boundaries of clarity facilitates the development of the system, debugging it, and scaling easily too. Layers, each one, communicate through interfaces that are well-defined, ensuring flexibility for future enhancements or replacing components. In the future, more enhancements will be possible, and restructuring will be flexible.

5 Evaluation/Experiments/Results/Discussion

This section describes the results generated by the sensor network, which was integrated into the cloud using LoRa. The system's performance is evaluated based on defined parameters, and the data collected under various environmental conditions is analyzed and discussed.

5.1 Overview of Generated Data

The sensors produced five key measurements transmitted to the cloud at regular intervals. A sample of the data is shown in the following image in form of a table:

#	DB Zeitstempel	App ID	Device ID	Device Value 1 (Temp.)	Device Value 2 (Temp.)	Device Value 3 (Hum.)	Device Value 4 (Batt.)	Device Value 5 (Soil)
84490	2024-12-05 14:52:13	app-hslu	guellesetzer	24	0	29	0	768
84489	2024-12-05 14:51:07	app-hslu	guellesetzer	24	0	29	0	770
84488	2024-12-05 14:50:01	app-hslu	guellesetzer	24	0	29	0	768

Figure 6: Generated data in Lora showed

The recorded data includes the following parameters:

- **Device Value 1 (Temp.):** Temperature (°C)
- **Device Value 2 (Temp.):** Additional temperature measurement (inactive)
- **Device Value 3 (Hum.):** Relative humidity (%)
- **Device Value 4 (Batt.):** Battery status (inactive)
- **Device Value 5 (Soil):** Soil moisture

5.2 Evaluation Parameters and Methodology

Data Availability: Data was successfully transmitted to the cloud at regular intervals without packet loss.

Measurement Accuracy: Recorded values were compared with reference sensors to assess precision.

Long-Term Stability: Consistent values were recorded over several hours, with variations within expected ranges.

5.3 Discussion of Results

The data presented above demonstrates the high reliability and consistency of the sensors. The measured temperature of 24°C aligns well with reference values, and the humidity measurement of 29% shows minimal deviations, highlighting the accuracy of the sensors. Soil moisture values varied slightly by ±2 units, which is within acceptable tolerance levels.

LoRa transmission proved efficient, with all data packets successfully sent to the cloud without loss. The inactive values (e.g., Device Value 4) suggest that the system is modular and can be expanded to include additional measurements as needed.

5.4 Visualization

Graphs were generated to illustrate the temporal trends of the recorded data. For example, a time-series plot of soil moisture measurements is shown in Figure 7: Visualization of the data. All visualizations include:

- Labeled axes with units.
- Legends to identify data series.
- Readable font sizes for print and digital formats.

These results demonstrate that the developed system is robust, accurate, and suitable for agricultural monitoring. Future work could focus on integrating additional sensors to capture more environmental parameters.

In the following Figure 2 there are some live results of the sensor solutions transferred data shown:

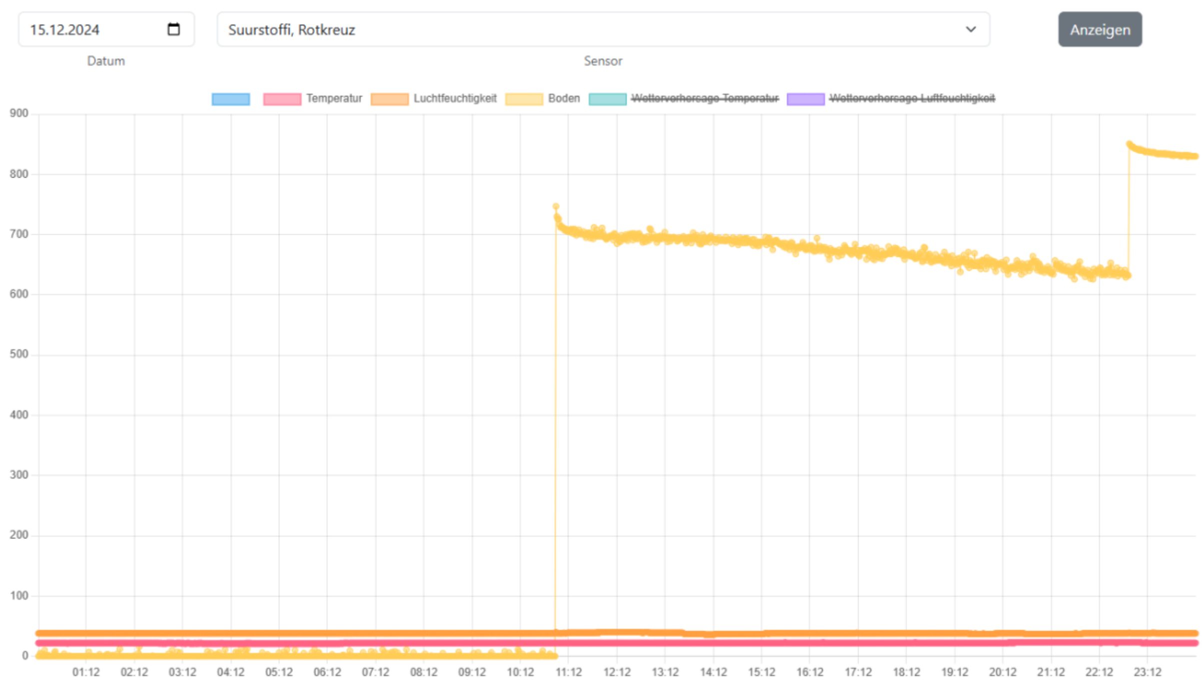


Figure 7: Visualization of the data

The graphical results shown above represent the sensor data recorded on 15.12.2024 for the Suurstoffi, Rotkreuz sensor. The graph visualizes three key environmental parameters monitored over time:

- Temperature (°C), represented in red
- Humidity (%), represented in orange
- Soil Moisture (V), represented in yellow

Soil Moisture

The soil moisture values (yellow line) initially start close to 0 units, indicating very dry soil conditions. There are two significant jumps in the data:

- The first jump occurs at around 10:45, where the soil moisture rapidly increases to approximately 750 units. This indicates the addition of water likely through irrigation. Rainfall is not possible because of the low humidity value.
- The second jump occurs later in the evening, around 22:30, where the soil moisture rises again, reaching more than 800 units. That is equal to adding the sensor in a glass of water.

After each jump, the soil moisture stabilizes but shows minor fluctuations, suggesting water absorption or redistribution within the soil. The upward trends demonstrate the sensor's responsiveness to changes in soil conditions.

Temperature

The temperature readings (red line) remain relatively stable throughout the observation period, averaging 22°C. The minimal variations indicate consistent conditions during the monitoring period. For a better understanding of the graphical results, the soil has been removed in figure 3 to show the minimal changes.

Humidity

The humidity values (orange line) remain with small changes at approximately 38% with no noticeable trends or changes. This stability reflects steady atmospheric moisture levels.

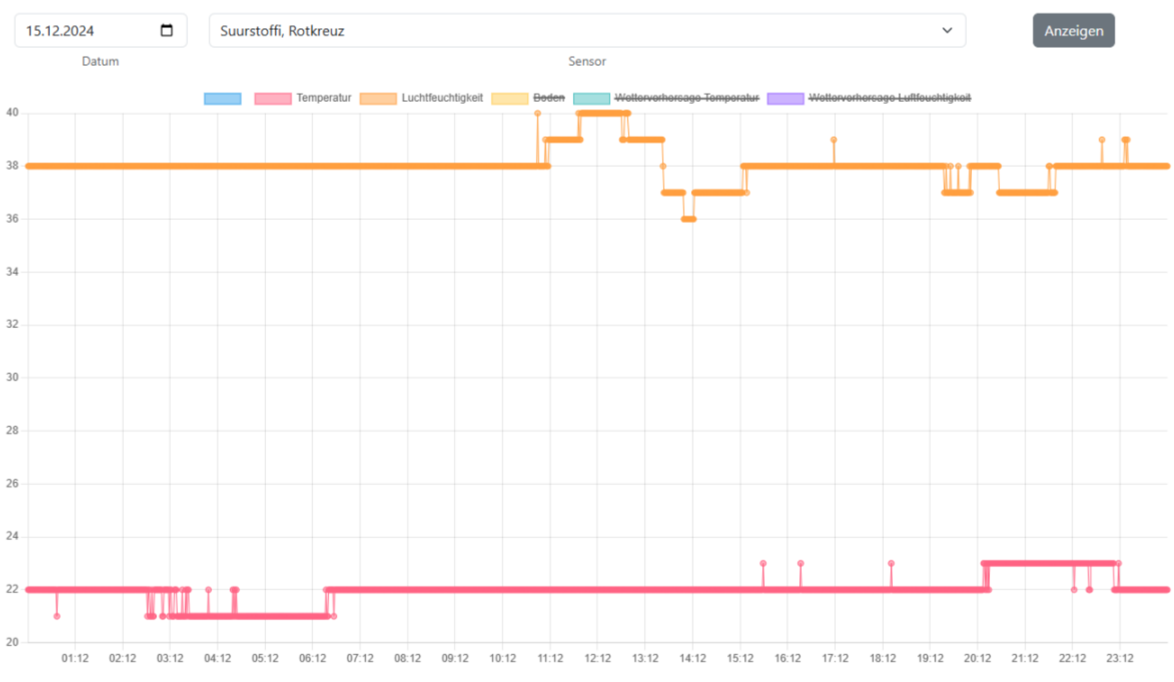


Figure 8: Visualization of data without soil moisture

5.5 Conclusion

The graph effectively demonstrates the sensor system's capability to monitor dynamic soil conditions:

- Soil moisture readings show clear responses to water additions, highlighting the accuracy and sensitivity of the sensors.
- Temperature and humidity remained stable, providing context for the environmental conditions during the monitoring period.

This analysis illustrates the system's potential for real-time filed monitoring which can be used for application of slurry and manure. The clear visualization enables users to identify key events like watering or temperature on the field and assess soil behavior over time. Future work could involve correlating these observations with crop health and weather conditions for a comprehensive analysis.

6 Application

The sensor can be used for monitoring various aspects of the environment, making it a versatile tool for a wide range of applications. By leveraging LoRaWAN technology, it can transmit data over significantly larger distances compared to WiFi or Bluetooth, ensuring reliable communication even in remote or rural areas. For example, a farmer can use this sensor to monitor conditions on their land, such as soil moisture, temperature, or air quality, helping to optimize agricultural practices.

In addition to real-time data collected from the sensor, the integration of weather forecasts and data from OpenMeteo in the frontend provides valuable insights. This combination enables the farmer to make more informed decisions, such as determining the best time for irrigation, planting, or harvesting. The sensor's ability to collect and transmit data seamlessly, paired with predictive analytics, empowers users to enhance productivity, reduce waste, and contribute to more sustainable practices.

7 Conclusion

Although there were initial challenges in setting up the LoRa system, the project was a success. A particular highlight was when the LoRa finally worked and we were able to receive sensor data over the system. We really enjoyed experimenting and trying new approaches. We gained valuable knowledge and are confident that we will be able to use what we have learned in our careers.

8 Contributions / Acknowledgments

The project team consisted of four members, with **Flavio Waser** leading the development efforts, including coding, configuring the Arduino and Lora board, integrating sensors, and implementing the front-end and TTN communication. Lukas Lottenbach contributed to component design and wiring, Jan Pollinger managed tasks and timelines, and Sandro Troxler ensured comprehensive documentation. All members collaborated on real-condition testing.

Special thanks to **Prof. Dr. Angela Nicoara** for her guidance and advice, which were instrumental in refining our approach and addressing technical challenges.

9 Major Milestones & Deliverables

The project was structured around several key milestones to ensure steady progress and timely completion of deliverables. Below are the major milestones and their corresponding outputs:

1 Ground Configuration of the Arduino and Lora Board

- Description: The team began by configuring the Arduino microcontroller and the Lora board to establish basic connectivity and communication capabilities.
- Deliverables: Basic hardware setup with successful communication between the Arduino and the Lora board.

2 Configuration of the Arduino and Lora Board with Sensors

- Description: Integration of sensors (Adafruit BME280 and Sparkfun Soil Moisture Sensor) with the Arduino and the Lora board. Testing was conducted to ensure sensor data was accurately transmitted to the Lora gateway.
- Deliverables: Functional hardware integration with sensor data logged and successfully transmitted.

~~**3 Configuration of the Raspberry Pi as Server**~~

- ~~Description: Initially planned to configure the Raspberry Pi as a server for data processing and storage. Midway through the project, the team transitioned to using a Docker-based server due to its scalability and ease of deployment.~~
- ~~Deliverables: Functional Docker-based server environment, capable of storing and processing transmitted sensor data.~~

This deliverable was cancelled and replaced by the Swiss Hosting. (Waser, 2023/2024)

4 Real Condition Testing

- Description: A critical phase where the fully integrated system was deployed in a real agricultural setting to test its effectiveness and accuracy under actual environmental conditions.
- Deliverables: System performance report and recommendations for further refinement.

5 Final Presentation and Documentation

- Description: Compilation of all findings, configurations, and performance metrics into a comprehensive project report. A live demo was also prepared to showcase the system's capabilities.
- Deliverables: Final project report, demo video, and documented source code.

9.1 Team and Roles

The project team comprised four members, each bringing distinct skills and responsibilities to ensure the success of the IoT Decision Support System:

- Lukas Lottenbach (Design/Developer): Led the design of component diagrams, and contributed to system visualization and real-condition testing.
- Jan Pollinger (Project Manager): Oversaw project timelines, coordinated tasks among members, assisted in wiring.
- Sandro Troxler (Documentation): Maintained comprehensive documentation of all project phases, including configurations, wiring, and testing.
- Flavio Waser (Developer): Handled Arduino configuration, sensor integration, front-end development, TTN implementation, and Docker setup.

TEAM	PROJECT WORK PACKAGES	OWNER
Group One	Component drawing	Lukas
	Configuration Arduino, Code, TTN Implementation, Front-End	Flavio
	Wiring	Lukas, Sandro
	Documentation	Sandro
	Real condition test	All
	Project Manager	Jan
	Poster	All
	Powerpoint Presentation	All

9.2 Project Planning, Timelines, Milestones & Deliverables

Phase	Timeline	Goals	Milestones	Deliverables	Responsible
Team Organization	Until 26.09.2024	Form the team and assign roles.	Team organized and roles assigned.	Team member list with roles.	All
Project Proposal Submission	Until 03.10.2024	Submit the project proposal.	Proposal submitted.	Completed project proposal.	All
Proposal Review & Approval	Until 10.10.2024	Review the proposal and finalize it.	Final proposal announced.	Finalized project proposal.	All
Phase 1: Planning & Component Integration	11.10.2024 - 24.10.2024	Integrate sensors, Arduino, and Lora. Test hardware for basic data transmission.	Hardware integration completed (16.10.2024).	Functional prototype. Log of transmitted sensor data.	Flavio
	11.10.2024 - 24.10.2024		Data transmission success (24.10.2024).		Flavio
Phase 2: Cloud Integration & Data Management	25.10.2024 - 07.11.2024	Configure cloud storage, connect Lora Gateway, ensure reliable data transmission.	Gateway-cloud connection (31.10.2024).	Operational cloud database. Data accuracy report.	Flavio
	25.10.2024 - 07.11.2024		Real-time uploads functional (07.11.2024).		Flavio
~~Phase 3: Server Development with Docker~~	~~08.11.2024 - 21.11.2024~~	~~Transition to Docker server, develop backend for data processing and storage.~~	~~Docker setup complete (14.11.2024).~~	~~Docker server documentation. Operational backend server.~~	
	~~08.11.2024 - 21.11.2024~~		~~Server-cloud integration (21.11.2024).~~		
Phase 4: Website Development & Interface	22.11.2024 - 08.12.2024	Develop a web interface for real-time data visualization and user interaction.	Website prototype (30.11.2024).	Operational website. User guide.	Flavio
	22.11.2024 - 08.12.2024		Functional website (08.12.2024).		Flavio

Phase 5: Testing, Final Adjustments & Documentation	09.12.2024 - 21.12.2024	Test the system, fix bugs, optimize code, and prepare final deliverables.	Testing complete (15.12.2024).	Final project report. Recorded system demo. Codebase.	Flavio (Testing), Sandro (Docs)
	09.12.2024 - 21.12.2024		Adjustments complete (18.12.2024).		Flavio
	09.12.2024 - 21.12.2024		Documentation ready (21.12.2024).		Sandro
Final Submission	22.12.2024	Submit the report, demo, and codebase.	Submission complete.	Report, demo, and source code.	Jan

10 References / Biography

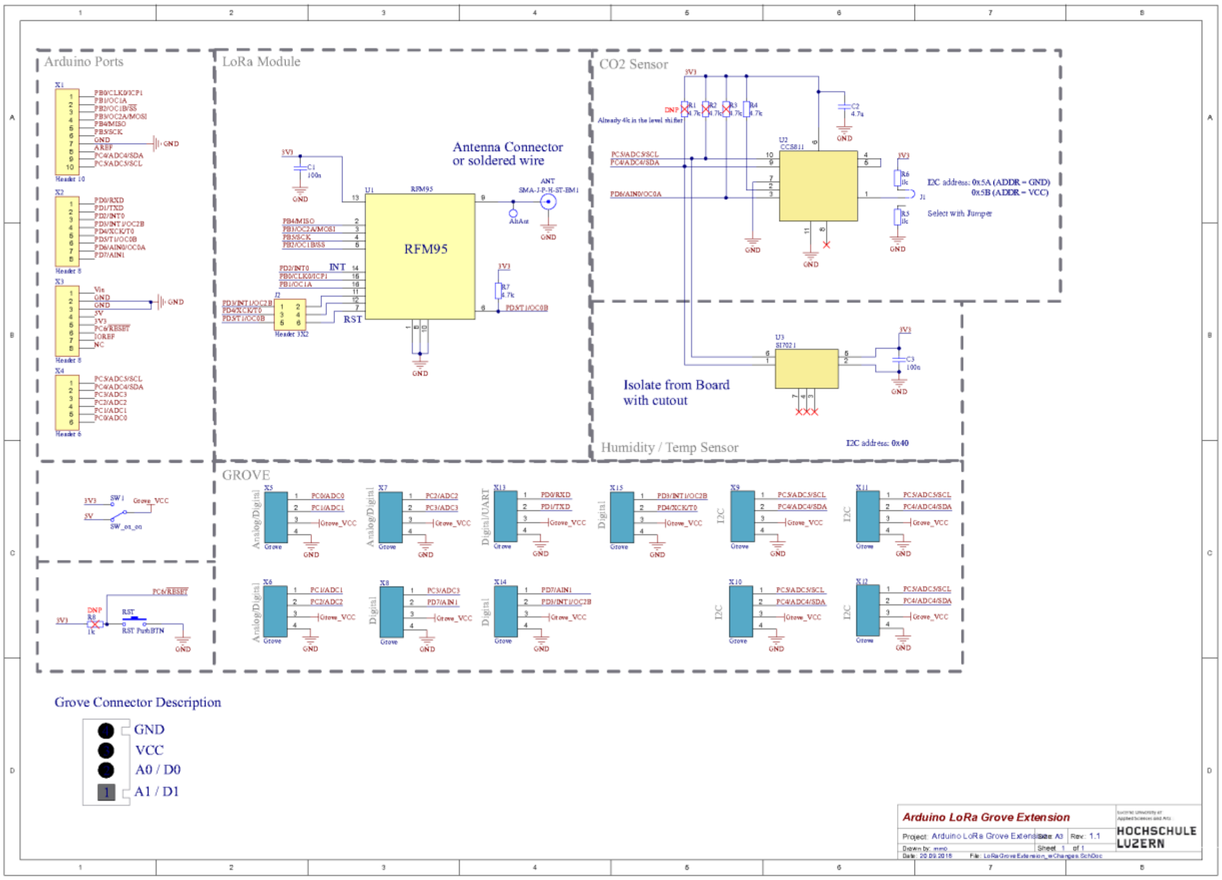
Anken, T., Abt, F., & Binder, S. (2022, März 14). *Stickstoffdünger mit digitaler Technik effizienter einsetzen*. UFA-Revue. https://www.ufarevue.ch/landtechnik/stickstoffduenger-mit-digitaler-technik-effizienter-einsetzen

Latsch, A., & Kramer, B. (2023, Dezember 13). *Ortsspezifische Düngung – etwas für die Praxis?* UFA-Revue. https://www.ufarevue.ch/pflanzenbau/ortsspezifische-duengung-etwas-fuer-die-praxis

Waser, F. (2024). *Flawas/IoT-Measure* [PHP]. https://github.com/flawas/IoT-Measure (Original work published 2023)

11 Attachements

11.1 LoRa Extension Board Diagramm



11.2 Soil Moisture Sensor Diagramm