Creating a digital smart flood warning system for surface water flooding in West Yorkshire: System Integration and Data Pipeline

July 2026

Document Title: Creating a digital smart flood warning system for surface water flooding in West Yorkshire: System Integration and Data Pipeline

Author(s): Ogundipe, F., and Bond, S.

Author affiliation(s):
School of Geography (Integrated Catchment Solutions Programme, iCASP), University of Leeds

Date of Issue: 14/07/2026

Version: V1.0

Please cite this document as: Ogundipe, F., and Bond, S. (2026) Creating a digital smart flood warning system for surface water flooding in West Yorkshire: System Integration and Data Pipeline. V1.0. A report developed in collaboration with Wakefield Council, Kirklees Council, Leeds City Council, Environment Agency and iCASP, under the West Yorkshire Flood Innovation Programme.

Contact details: iCASP@leeds.ac.uk | f.ogundipe@leeds.ac.uk

AI statement: Licenced artificial intelligence (Copilot) was used to assist with language editing and clarity of writing. The scientific content, data analysis, interpretation of results, and conclusions were developed by the authors.

Creating a digital smart flood warning system for surface water flooding in West Yorkshire project is funded by the Impact Acceleration Account 95551879 (NERC IAA HEIF)

Table of Contents

List of Figures

List of Tables

Introduction

Surface water flooding a growing challenge in the UK which the West Yorkshire Flood Innovation Programme (WYFLIP) are seeking to address via exploration of Digital Smart Flood Warning Systems. Long Range Wide Area Network (LoRaWAN) technology has been identified as a promising mechanism and pilot studies to test LoRaWAN-based water level monitoring are underway.

This document outlines the development of a LoRaWAN-based flood monitoring system using multiple water level sensors integrated through The Things Network (TTN) and integrated into TagoIO for storage, processing and visualisation through interactive dashboards. The system was developed in order to enable laboratory and field-based evaluation of the accuracy, precision and reliability of three sensors (Milesight, Decentlab and Maxbotix), with particular attention to their performance across varying static and dynamic flow conditions.

Detailed evaluation of sensor performance under laboratory and field conditions is presented in the associated laboratory and field reports for this project. This document therefore focuses on the technical integration and operational framework required to support digital flood monitoring systems.

The system architecture follows a structured data flow designed to enable real-time monitoring and analysis. Sensors transmit data via LoRaWAN to a nearby gateway, which forwards the data to TTN. From there, data is routed via a webhook integration into TagoIO, where it is stored, processed and visualised on dashboards for monitoring and decision-making. This architecture enables real-time data flow and supports scalable deployment for environmental monitoring.

Limitations of this approach include potential network variability and sensor-specific constraints (such as reporting interval restrictions). This guide is designed to enable repetition of the system creation process and facilitate troubleshooting. It should be understood in context of the WYFLIP Digital Smart Flood Warning Systems project.

Project Overview

Flooding poses a significant risk to communities, infrastructure and the environment, particularly in areas prone to surface water accumulation and extreme weather events. Effective flood monitoring is essential for early warning, risk mitigation, infrastructure maintenance and informed decision-making, enabling timely responses to potential flood events and reducing associated impacts.

The primary objective of this project is to develop and evaluate a real-time flood monitoring system using LoRaWAN-enabled sensors. This includes assessing the accuracy, precision and reliability of different water level sensors, validating their performance against reference measurements and establishing a robust data pipeline for continuous monitoring and analysis.

The project was implemented through three key stages. First, development of the LoRaWAN system architecture and physical sensor mounting required for laboratory and field testing. Second, a controlled laboratory test to evaluate sensor performance under varying conditions such as static water levels, drawdown and dynamic flow scenarios. And third, a pilot of sensors in real-world environments to test their performance as flood monitoring and early warning systems.

Key stakeholders involved in the project include the University of Leeds (UoL), Wakefield Metropolitan District Council (WMDC) and other members of the West Yorkshire Flood Innovation Programme (WYFLIP). WYFLIP comprises the five lead local flood authorities in West Yorkshire, Environment Agency, Yorkshire Water, West Yorkshire Combined Authority and University of Leeds. Their collaboration supports both the technical development and practical application of the system, ensuring that the solution aligns with real-world needs and policy objectives.

This document outlines stage one of the project: creation of the LoRaWAN system architecture for monitoring water levels for flood mitigation.

System Architecture

The system architecture is designed to enable real-time collection, transmission, processing and visualisation of water level data using a LoRaWAN-based framework. The end-to-end data flow follows a structured pipeline:

Sensor → Gateway → The Things Network (TTN) → Webhook → TagoIO → Dashboard

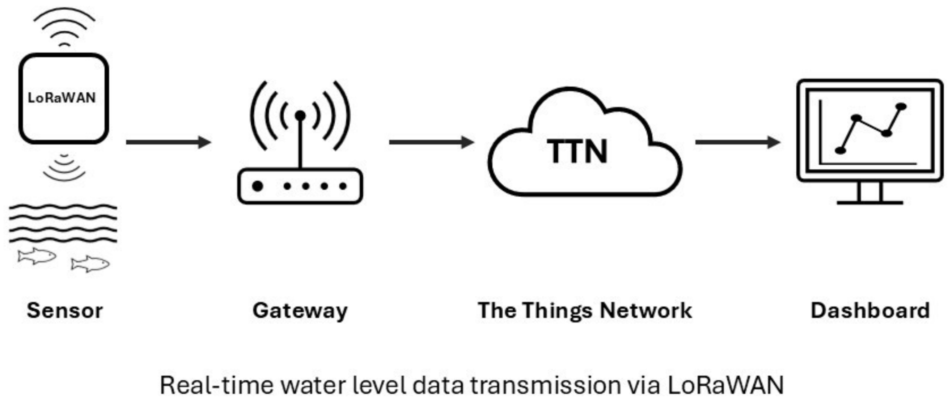

Figure 1: System architecture for real-time water level monitoring using LoRaWAN, TTN and TagoIO

System Components

Sensors:

The system integrates three LoRaWAN-enabled water level sensors:

- Milesight EM500-UDL
- Decentlab DL-MBX
- Maxbotix ELT2

These sensors measure distance to the water surface and transmit readings at configured intervals.

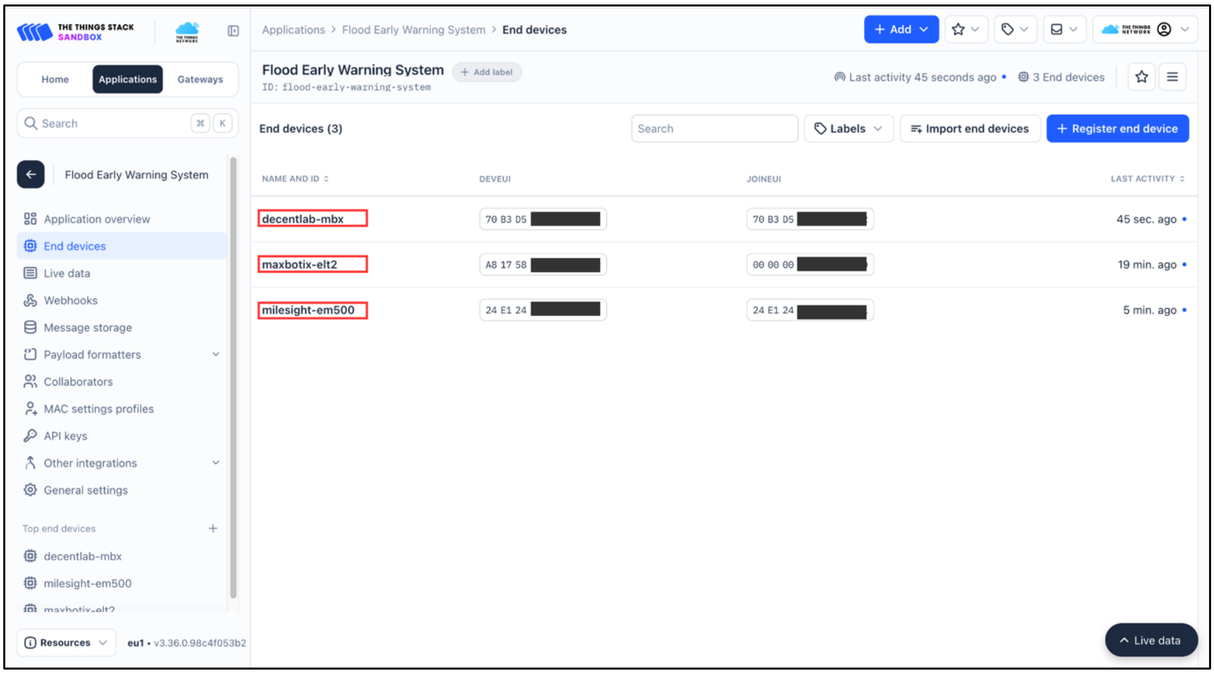

Figure 22: Registered LoRaWAN Sensors in TTN Application

The sensors evaluated in this study were selected by WYFLIP partners for Digital Smart Flood Warning Systems project. While the implementation presented here focuses on these specific devices, the integration approach is not sensor-specific and can be readily adapted to other LoRaWAN-enabled sensors and environmental monitoring applications using similar communication and data management protocols.

Table 1 provides a summary of the key technical characteristics and specifications of the three sensors used, based on information from their respective manufacturer datasheets.

Table 1: A summary of sensor properties as published in their respective data information manuals.

Sensor name	**Decentlab Ultrasonic Level Sensor** SKU: DL-MBX-002-EU868	**ELSYS ELT2 MaxBotix Distance Sensor** SKU: ELT2-MB7389	**Milesight EM500 Ultrasonic Distance/Level Sensor** SKU: EM500-UDL-868EU-W100
Sensor type	LoRaWAN-enabled non-contact ultrasonic distance level sensor	ELT-2 is a universal outdoor LoRaWAN device	Ultrasonic distance/level sensor within the EM500 outdoor LoRaWAN monitoring sensor series
Measurement principle	Measures distance using ultrasonic time-of-flight, temperature compensated.	Measures analogue/digital signals and connect to external sensors, including a MaxBotix distance sensor input.	Ultrasonic non-contact distance/level measurement
Measurement range (vertical water level)	0.3–5 m	Depends on the specific MaxBotix sensor model connected. Typical range is 0.3-5m.	0.5–10 m
Accuracy	1% or better, factory calibrated.	1% or better, factory calibrated.	±1% full scale.
Resolution	1 mm	1 mm	1 mm
Cone angle	A specific cone angle was not found. The ultrasonic beam is a ‘diffuse cone’ and the greatest sensitivity is at the centre of the measurement beam.	A specific cone angle was not found. Typically features a narrow, focused beam designed to minimize Interference from side objects, often referred to as having a ‘small cone’ or ‘narrow beam’	A detection cone angle of approximately 60^0 in front of the ultrasonic probe.
LoRaWAN connectivity range	Wireless range >10 km line-of-sight and approx. 2 km suburban.	Range stated as 8 km with actual range dependent on terrain/buildings.	up to 15 km communication range

Configuration method	Configurable via command line interface and LoRaWAN downlink command interface	Configurable via NFC and over the air. Sampling rate, spreading factor, encryption keys, port and modes can be changed.	Configurable via NFC using Milesight ToolBox App or via USB/ToolBox software
Battery power	2 x alkaline C batteries. Battery life estimate: 6.5 years at 10 min interval/SF7* 3.5 years at 10 min interval/SF12 12.5 years at 60 min interval/SF7 10.5 years at 60 min interval/SF12 *SF= Spreading factor	1x replaceable 3.6V AA 14505 Li-$SOCl_2$ battery Battery life estimate: up to 10 years, depending on the usage	1x 19000 mAh replaceable battery Battery life estimate: Up to 10 years, low power consumption.
Operating conditions	Temperature: -20 to 50°C Humidity: 0–100% RH	Temperature: -40 to 60°C, -40 to 85°C with external power supply Humidity: 0–100% RH Pressure: 260–1260 hPa	Temperature: -30°C to +65° Humidity: 0% to 80% RH (non-condensing).
Enclosure	Polycarbonate enclosure; weatherproof, impact- and UV-resistant. Enclosure is IP66/IP67, sensor is IP67.	Enclosed in IP67 box; designed for extreme outdoor conditions.	IP67 waterproof enclosure for harsh environment applications.
Dimensions & weight	170 × 81 × 70 mm, 450 g including Batteries.	94 × 59 × 35 mm, 100 g including batteries.	156.1 × 71 × 69.5 mm, 370.4 g
Mounting / installation notes	Mount with ultrasonic horn facing target. Keep target centred in measurement beam, avoid interfering objects/sharp corners, mount on a cantilever away from poles/walls/tubes, protect from rain/direct sun where possible and avoid metallic objects near the device for radio performance	Mount using two side holes. Ensure cable gland and antenna connection are tightened to prevent moisture ingress. External sensors should use as short a cable as possible for best performance	EM500 transceiver supports wall, pole and DIN rail mounting. For EM500-UDL, the device should sit vertically above the target/object, have a clear path, be more than 30 cm from side walls and avoid internal obstructions, arched/circular container centres and inlet orifices.

Additional features/ parts required	N/A	Requires ANT868 OR ANT915C whip antenna & 2XER14500 batteries.	N/A
Price (August 2025)	£606.76	£384.45 including the whip antenna and batteries	£315.08

Gateway (Public/Private):

A LoRaWAN gateway receives radio signals from the sensors and forwards them to TTN. Gateways may be public (open to devices from multiple users) or private (restricted to a specific organisation or application). Without a gateway, sensor data cannot reach TTN for processing and onward transmission to downstream platforms such as TagoIO.

For this project, publicly accessible gateways were used during sensor setup and initial connectivity testing, including the Leeds City Council LoRaWAN gateway network. A dedicated MultiTech LoRaWAN gateway was subsequently deployed for laboratory and field testing to ensure consistent network availability and greater control over data transmission. When selecting a gateway, factors such as coverage, line-of-sight, distance to the deployment site and environmental obstructions should be considered, as these can affect communication performance and data reliability.

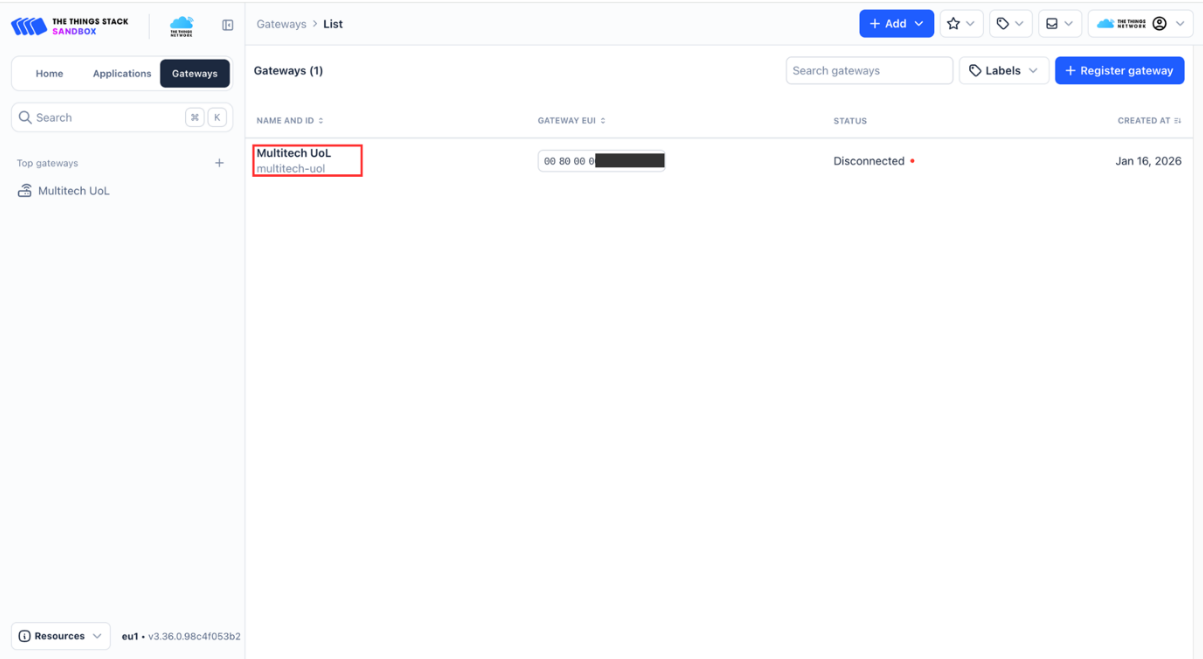

Figure 33: Registered LoRaWAN Gateway (MultiTech UoL) in TTN application

The Things Network (TTN):

TTN acts as the LoRaWAN network server, managing device registration, authentication and communication between sensors and connected applications. It handles uplink and downlink messages, processes incoming data and routes sensor information to external platforms via integrations and monitoring. In this system, TTN provides the link between the sensors, gateways and TagoIO, enabling reliable real-time data transmission and monitoring.

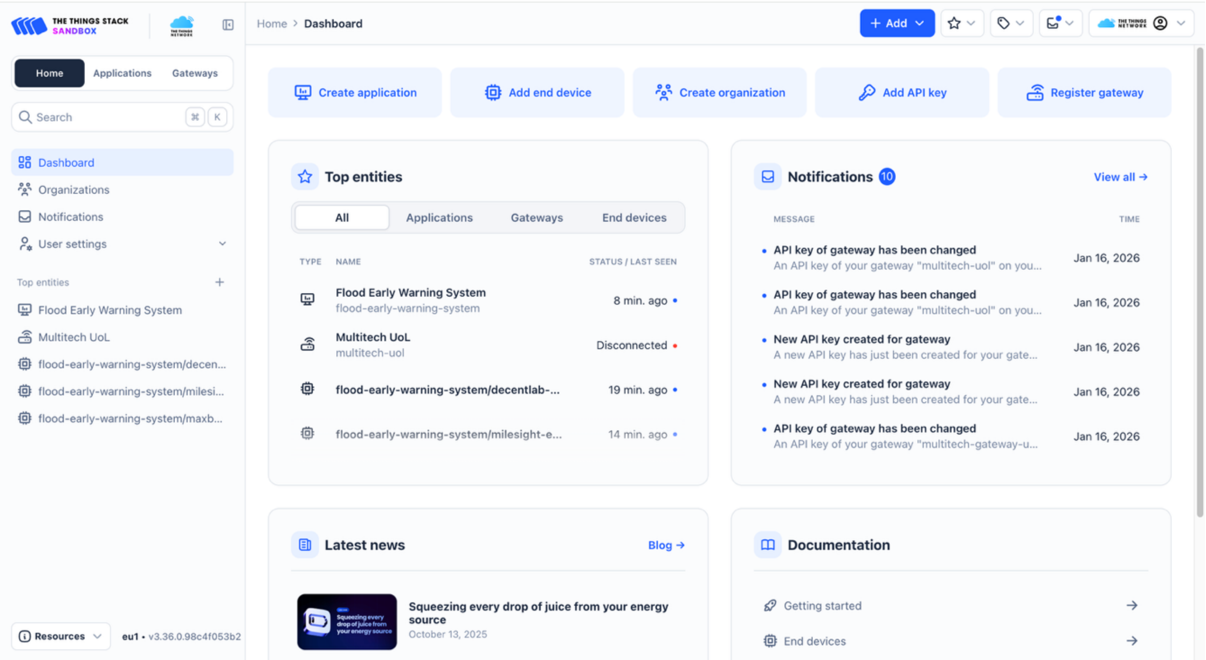

Figure 44: TTN Dashboard Overview Displaying Application, Gateway and End Device Status

TagoIO:

TagoIO was used for data ingestion, storage, processing and visualisation within this project. The platform supports device management, dashboard creation, real-time monitoring and historical data analysis, making it suitable for rapid development of IoT applications.

The Things Network (TTN) supports integration with a range of third-party platforms through built-in webhook templates, including Akenza, AllThingsTalk Maker, Datacake and ThingSpeak. TTN also supports custom webhook implementations, allowing sensor data to be routed to bespoke applications, databases, dashboards, or cloud services. For this project, TagoIO was selected as the preferred platform due to its integrated device management, visualisation capabilities and ease of deployment.

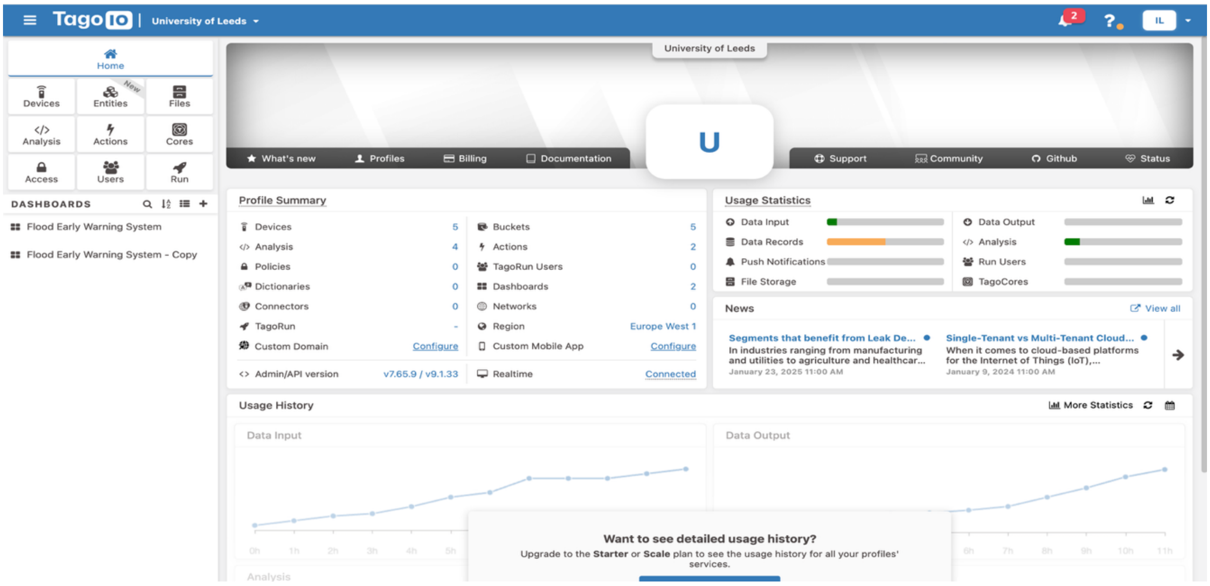

Figure 55: TagoIO Platform Overview Showing Device Usage and Data Management

Data Flow

Uplink Communication

Sensors periodically transmit water level measurements and device telemetry via LoRaWAN to a gateway. The gateway forwards these uplink messages to The Things Network (TTN), where device authentication and network management functions are performed. The uplink data is then made available within the TTN application and routed to downstream platforms for further processing.

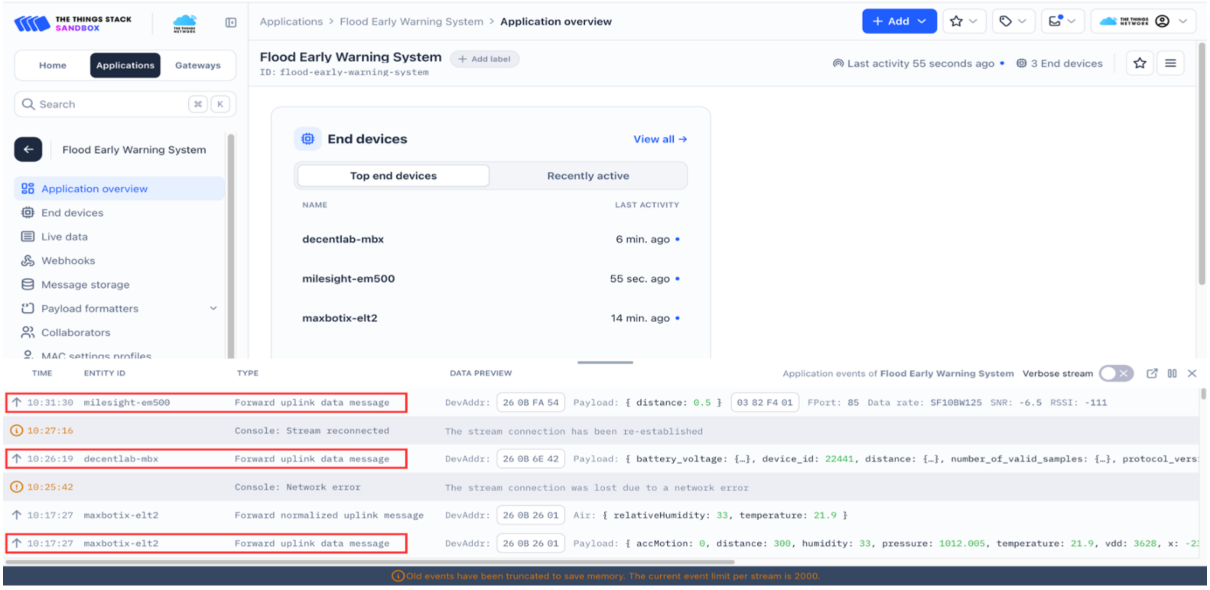

Figure 66: TTN Application Live Data Showing Uplink Messages from Registered Sensors

Webhook Integration

TTN forwards uplink data to TagoIO through a webhook configured with a dedicated endpoint (/uplink) and an authorization token generated within TagoIO. This integration enables automated, real-time transfer of sensor data from the LoRaWAN network server to the visualisation platform. In this project, payload formatters were not configured within TTN and data was forwarded for downstream processing within TagoIO.

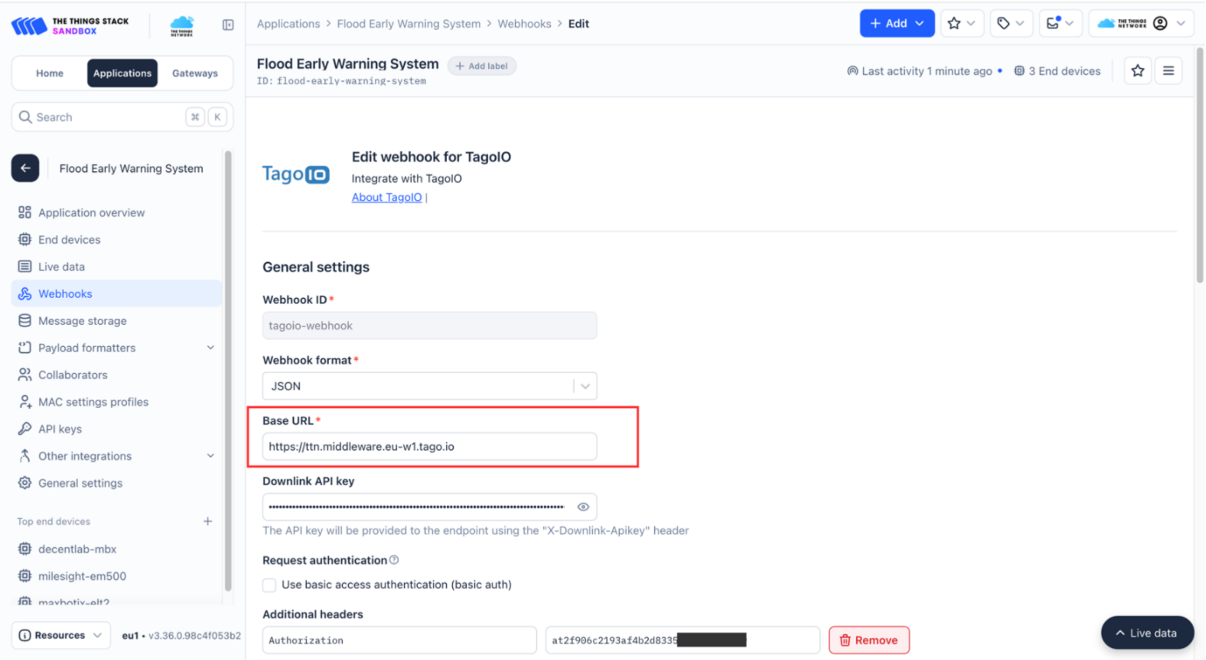

Figure 77: TTN Webhook Configuration for TagoIO Integration (a)

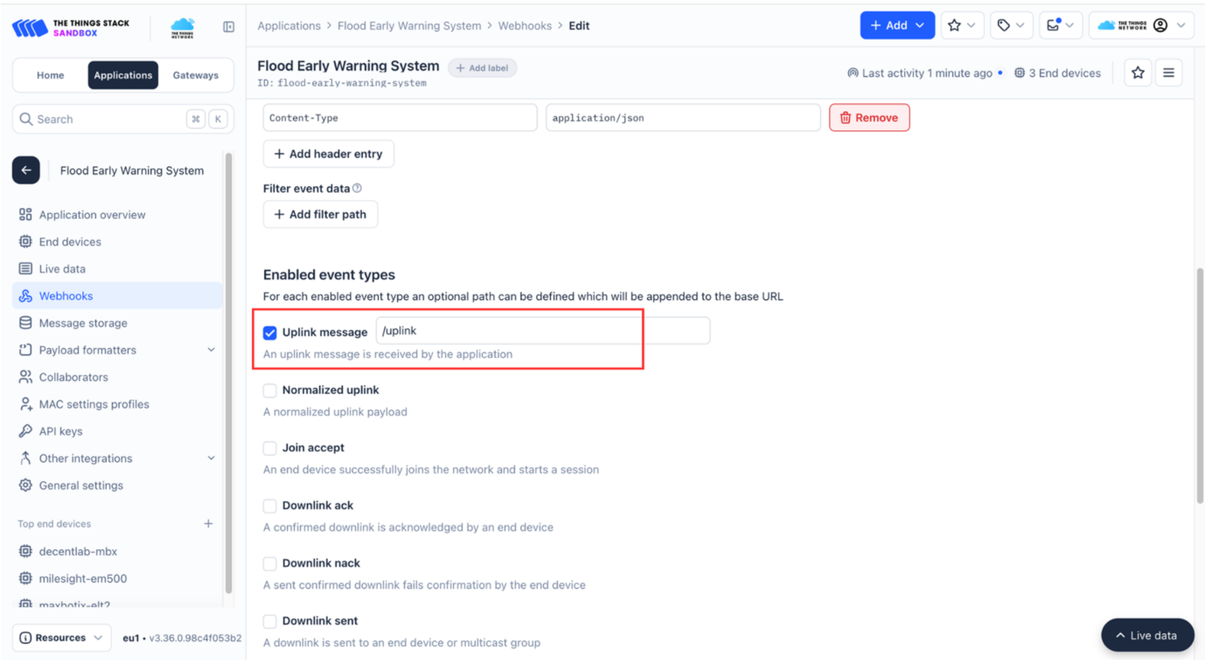

Figure 88: TTN Webhook Configuration for TagoIO Integration (b)

Downlink Communication

Downlink messages enable remote configuration of sensor parameters, such as reporting intervals and operational settings. These messages are transmitted from TTN to the sensor via the gateway and are primarily used during sensor setup and testing. Downlink usage should be minimised during routine operation due to LoRaWAN duty cycle restrictions and network limitations.

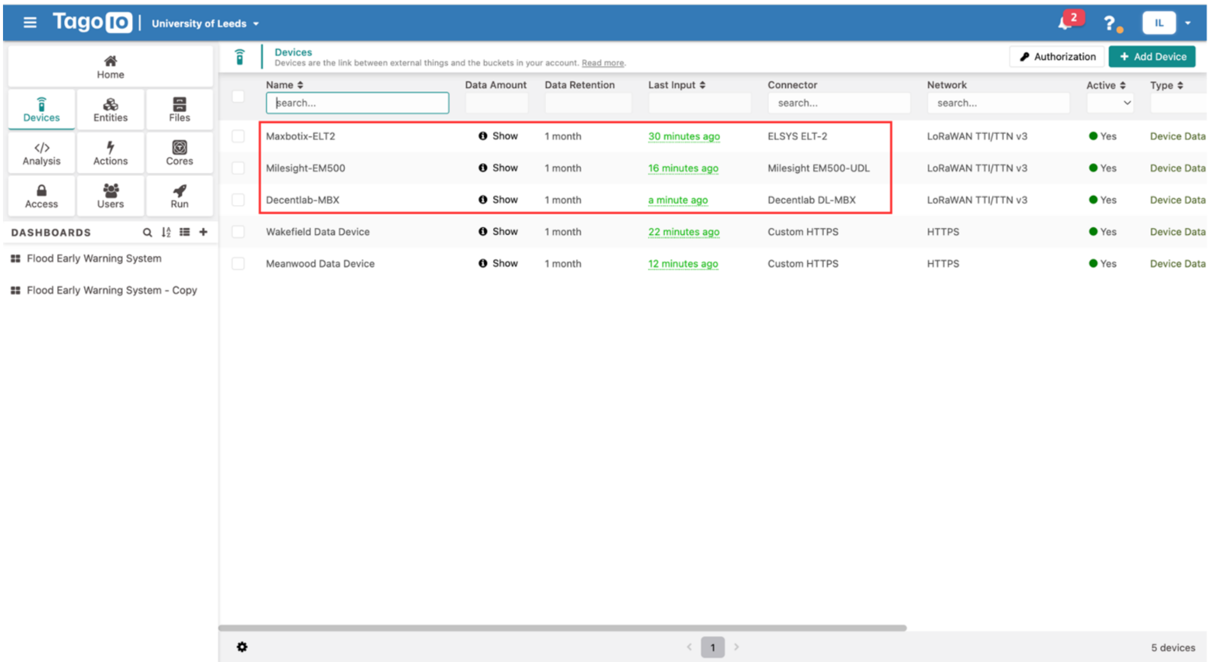

Figure 99: TagoIO Device List Showing Multi-Sensor Integration

Data Storage and Visualisation

Once received by TagoIO, sensor data is stored within device-specific data buckets and made available for real-time monitoring and historical analysis. Interactive dashboards provide visualisation of water level measurements, battery status, signal quality and other sensor parameters. This enables users to monitor system performance, compare sensor behaviour and analyse trends over time.

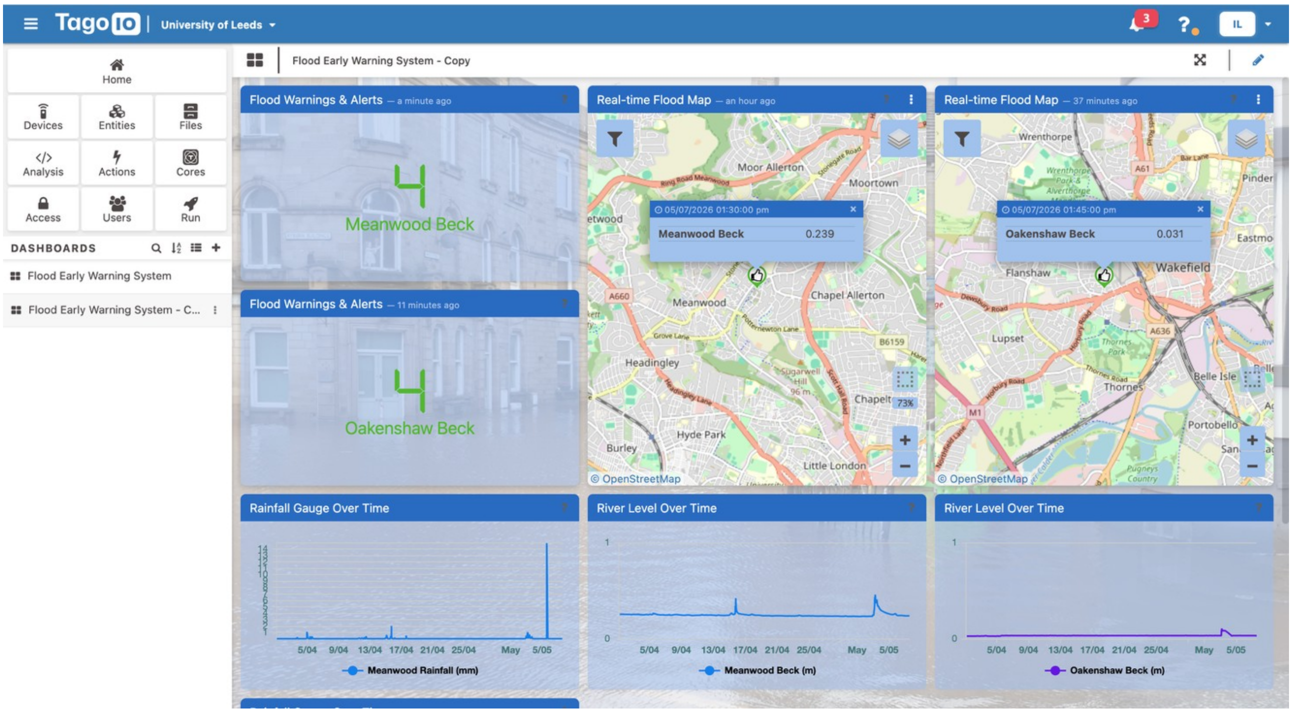

Figure 1010: TagoIO Device List Showing Multi-Sensor Integration and Data Management

Network Setup (TTN)

The LoRaWAN network for this project was configured using The Things Network (TTN), which serves as the network server responsible for device authentication, connectivity management, message routing and integration with external platforms.

Account and Application Setup

A TTN account was created, followed by the creation of an application titled "Flood Early Warning System". The application acts as the central management environment for all registered LoRaWAN devices, enabling monitoring of device activity, message traffic, integrations and network performance.

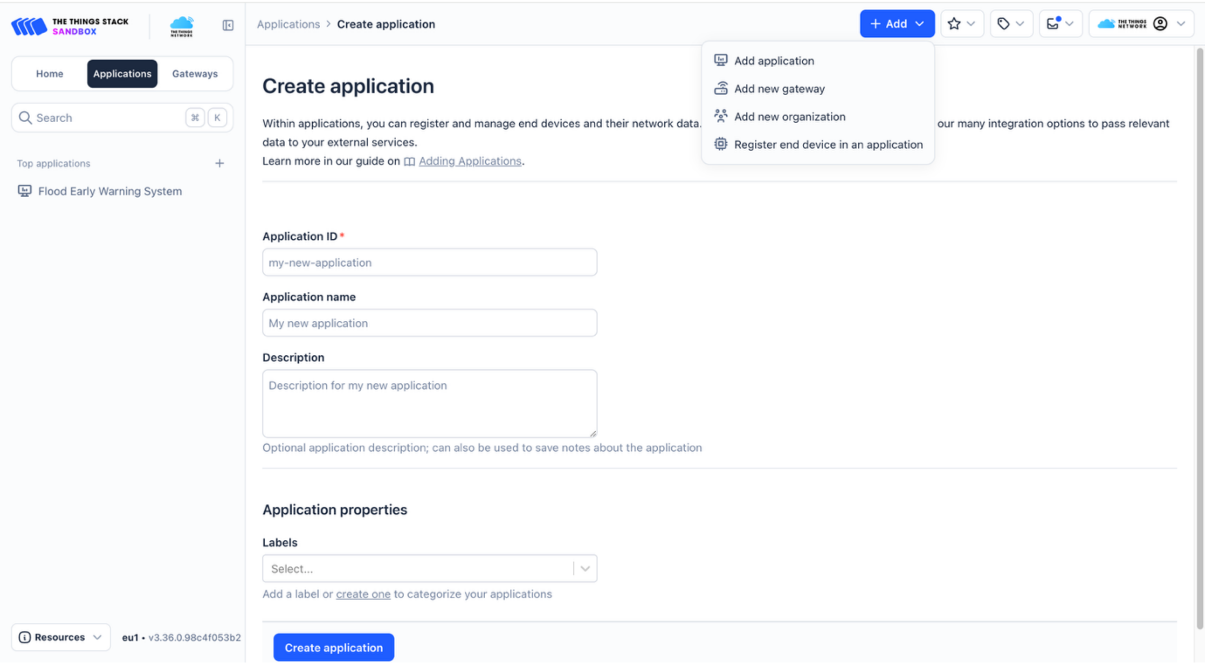

Figure 1111: TTN Application Setup

Region Configuration

The application was configured using the EU863–870 MHz (SF9 for RX2) frequency plan, which is the standard LoRaWAN frequency band for Europe. Selecting the correct frequency plan is essential to ensure compatibility between sensors, gateways and regional radio regulations.

Device Registration and Activation

Each sensor was registered as an end device within the TTN application using manufacturer-provided credentials.

The following parameters were required during device registration:

- DevEUI - Unique identifier assigned to each device.
- JoinEUI (AppEUI) - Identifier linking the device to the application.
- AppKey - Security key used for device authentication and session establishment.

These credentials are typically provided by the sensor manufacturer and can be found on the device label, packaging, accompanying documentation, or within the manufacturer's configuration tools. The values were used during TTN device registration to enable secure network access using Over-The-Air Activation (OTAA).

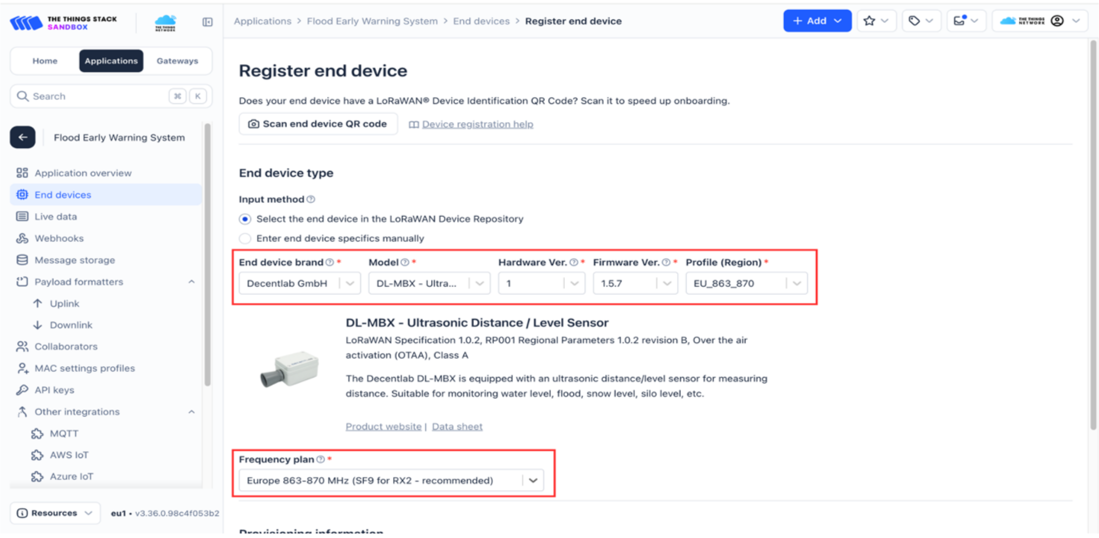

Figure 1212: TTN Device Registration Configuration for Decentlab, Maxbotix and Milesight Sensors Showing Model, Firmware and EU868 Frequency Plan (a)

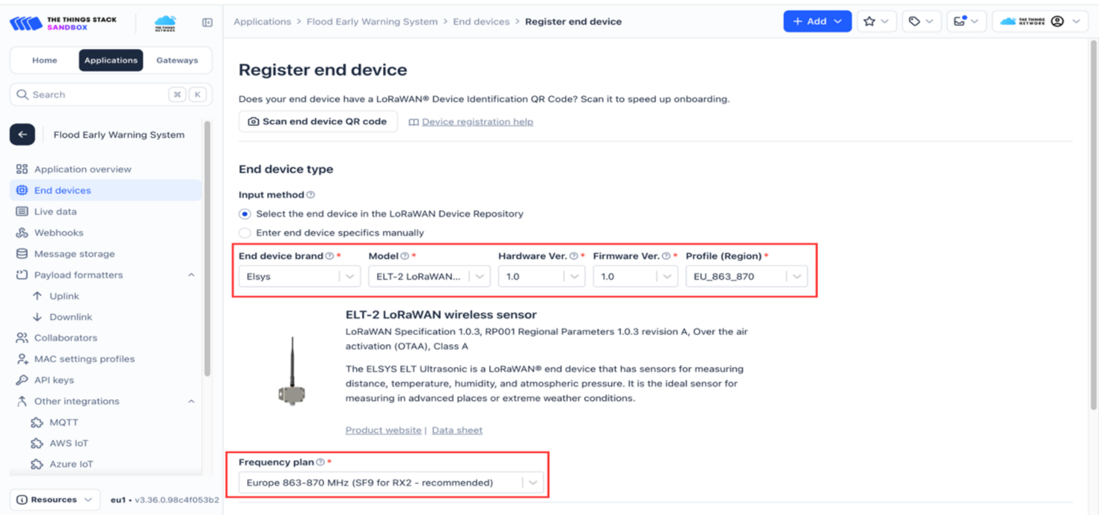

Figure 1313: TTN Device Registration Configuration for Decentlab, Maxbotix and Milesight Sensors Showing Model, Firmware and EU868 Frequency Plan (b)

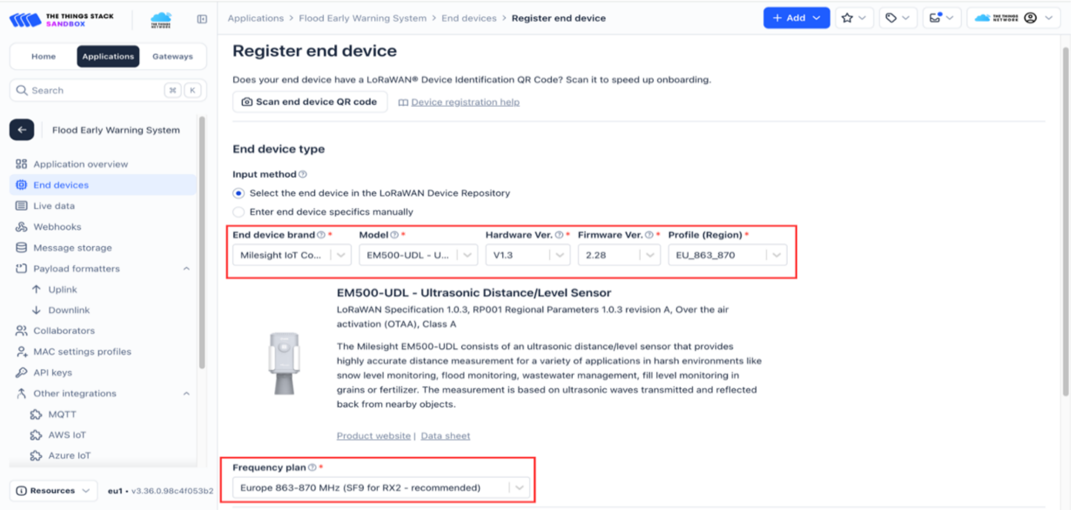

Figure 1414: TTN Device Registration Configuration for Decentlab, Maxbotix and Milesight Sensors Showing Model, Firmware and EU868 Frequency Plan (c)

Payload Decoding

Payload formatters were not configured within TTN for both uplink and downlink messages and were left as None. In LoRaWAN communications, a payload refers to the actual data transmitted by a device, typically encoded as a sequence of bytes. For the sensors used in this project, the payload contained measurements such as water level, battery voltage and other device-specific information.

As no payload formatter was applied, raw payload data was forwarded directly without network-level decoding. Decoding and transformation of sensor measurements into human-readable values were instead performed downstream during data processing and analysis. This approach provided greater flexibility and control over data cleaning, transformation, validation and comparison across multiple sensor types.

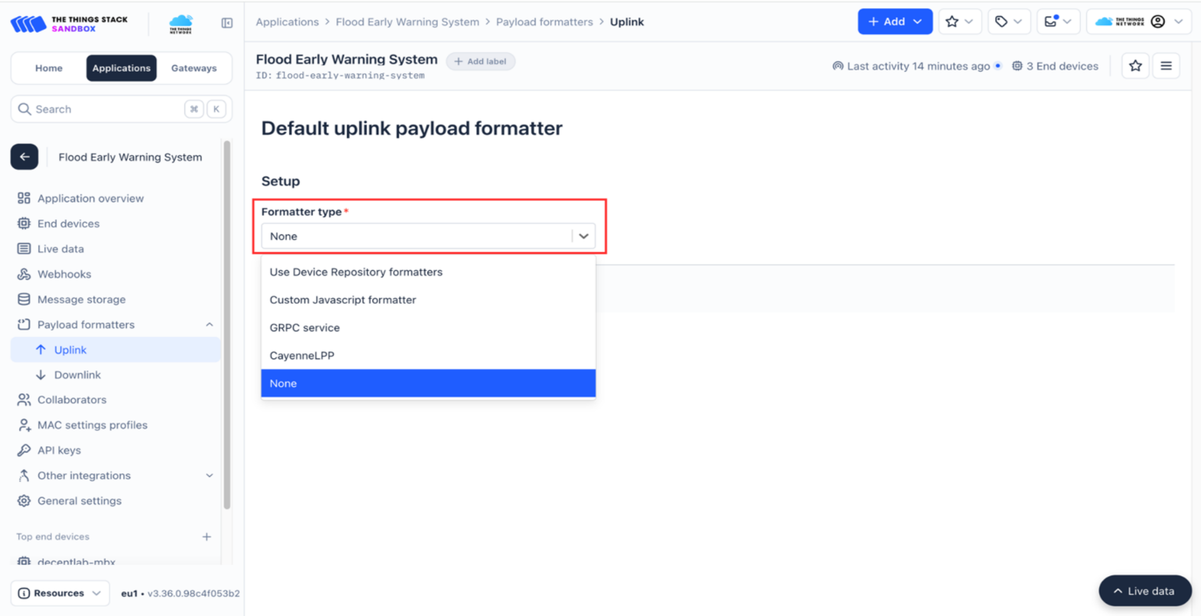

Figure 1515: TTN Uplink Payload Formatter Configuration Set to None (No Decoding Applied at Network Level)

Gateway Connectivity

A LoRaWAN gateway provides the communication bridge between sensors and TTN by receiving radio transmissions from sensors and forwarding them to the network server via an internet connection.

For this project, publicly accessible gateways were used during sensor setup and initial connectivity testing. A dedicated MultiTech LoRaWAN gateway was subsequently deployed for both laboratory and field testing to ensure consistent network availability and greater control over data transmission.

When selecting a gateway, factors such as coverage, line-of-sight, distance to the deployment site, antenna placement and environmental obstructions should be considered, as these can influence communication reliability and signal quality.

Live Data Verification

Successful network configuration was verified using the TTN Live Data console.

Verification included:

- Successful device joins requests confirming OTAA activation
- Receipt of uplink messages confirming ongoing sensor communication
- Monitoring of signal quality metrics including:
  - RSSI (Received Signal Strength Indicator)
  - SNR (Signal-to-Noise Ratio)

These checks confirmed that the sensors, gateways and TTN infrastructure were functioning correctly prior to integration with downstream platforms.

Sensor Configuration

The sensors were configured to support both standard operation and high-frequency testing to evaluate performance under different conditions, particularly during controlled laboratory experiments. The procedures used to modify sensor reporting intervals and operational settings are described in this section and may be followed to reconfigure the sensors for alternative monitoring requirements.

Reporting Intervals

Default Interval: All sensors tested in this project operate at a default reporting interval of 10 minutes, which supports long-term deployment by balancing data availability with battery efficiency.

Testing Intervals:

For laboratory testing, the reporting interval was reduced to 2 minutes to enable higher-resolution data collection and improve assessment of sensor responsiveness. Attempts were made to achieve a 1-minute interval; however, this was not consistently supported due to device-specific and LoRaWAN network constraints, including duty cycle restrictions, TTN fair use considerations and limitations on the frequency of downlink communication.

Configuration Methods

Sensor configuration varied by manufacturer, reflecting differences in device design and management approaches.

Decentlab Sensor

The Decentlab sensor was configured using a downlink-based approach. A downlink is a message sent from the network server to a device to modify its settings or behaviour. Configuration payloads were generated using the Decentlab Downlink Command Encoder and transmitted via The Things Network (TTN) to adjust parameters such as reporting interval and transmission schedule.

In the configuration shown in Figure 17, the reporting interval parameter was set to 120 seconds (2 minutes). This value determines how frequently the sensor transmits measurements to the network during operation. The generated hexadecimal payload was then used in a TTN downlink message to apply the updated reporting schedule to the device.

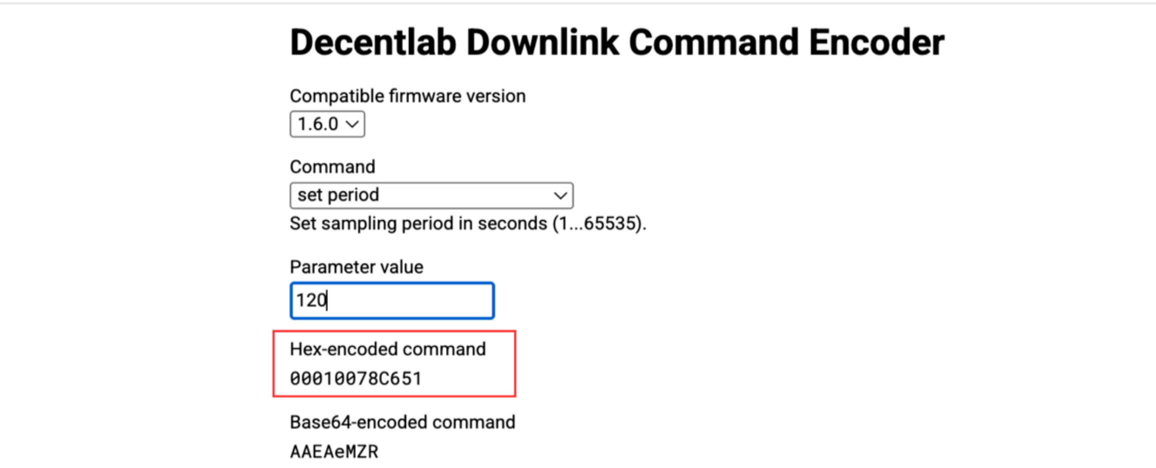

Figure 1616: Decentlab Downlink Command Encoder Generating Hex Payload for Sensor Reporting Interval Configuration

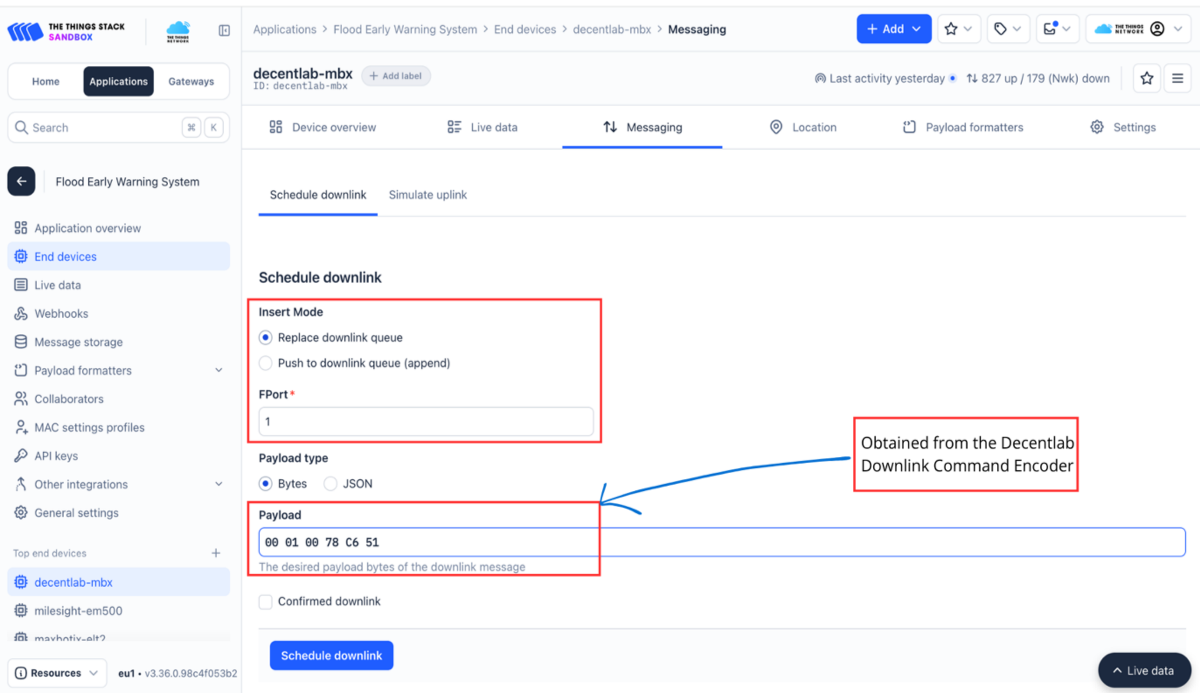

Figure 1717: Decentlab Downlink Configuration in TTN Showing FPort and Payload Scheduling for Interval Control

Milesight and ELSYS Sensors

The Milesight EM500-UDL and ELSYS ELT-2 sensors were configured using Near Field Communication (NFC) through their respective mobile applications:

Milesight: [Milesight Toolbox] (available on Apple App Store and Google Play)

ELSYS: [Sensor Settings] (available on Apple App Store and Google Play)

Configuration was performed by scanning the devices with a smartphone using NFC, allowing direct adjustment of parameters such as reporting interval. Settings were written directly to the sensors through their respective mobile applications and subsequently verified using the applications' read-back functionality to confirm successful configuration. Unlike the Decentlab sensor, these configuration changes were applied locally via NFC rather than through TTN downlink messages.

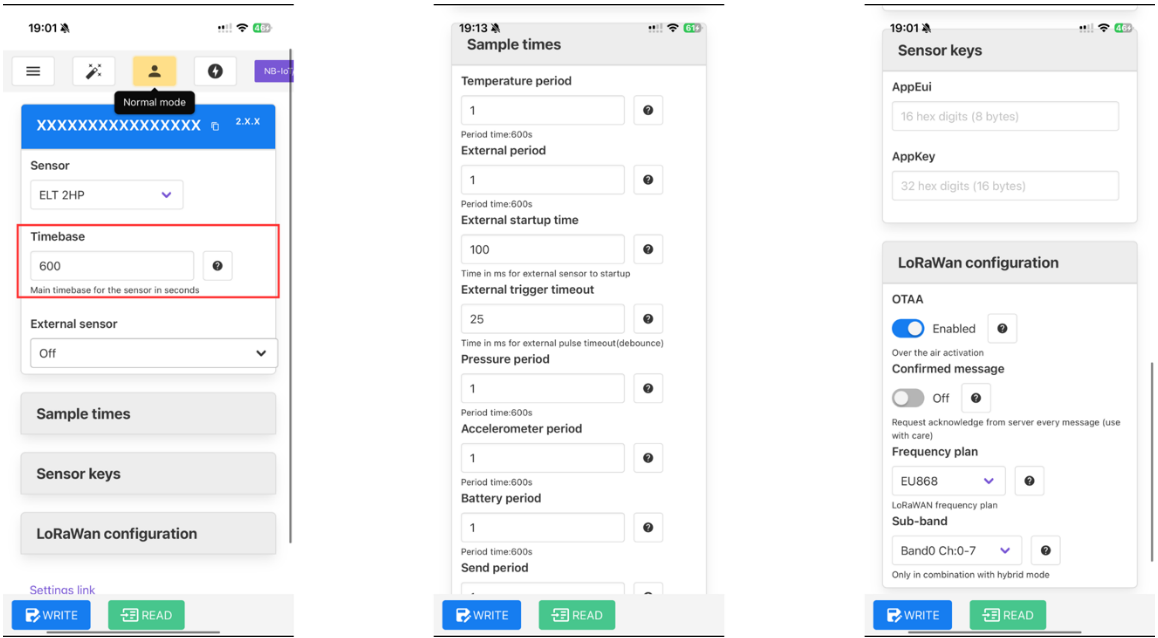

Figure 1818: Screenshot of ELSYS Sensor Configuration via NFC Mobile Application

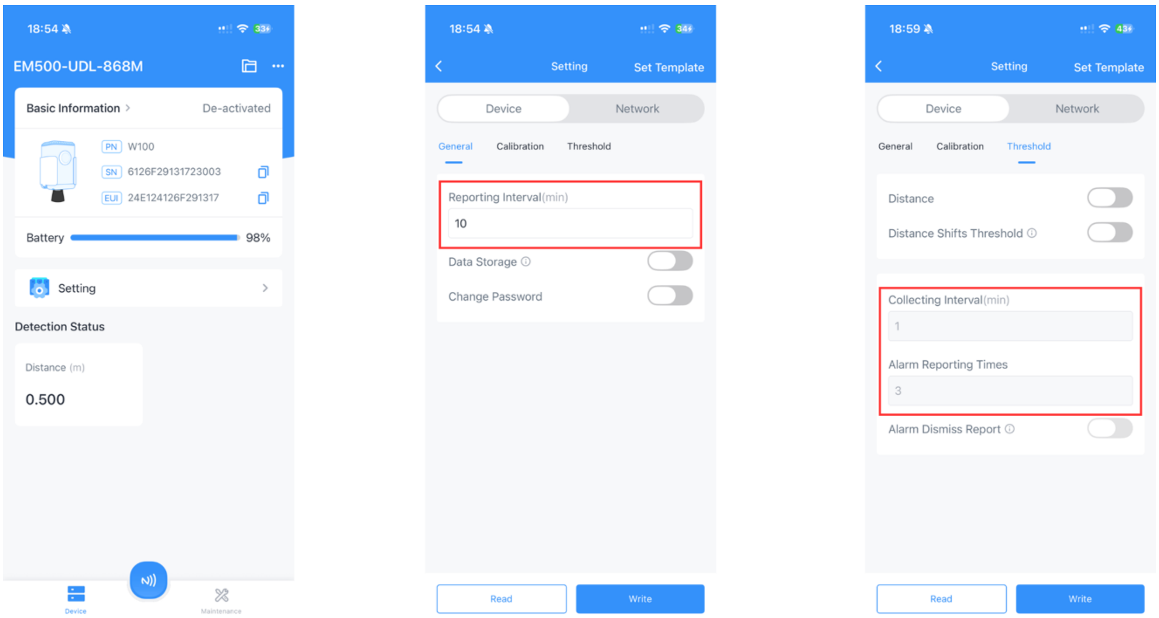

Figure 1919: Screenshot of Milesight Sensor Configuration via NFC Mobile Application

Limitations

Device Constraints (Decentlab)

The Decentlab sensor requires structured downlink configuration for setting sampling and transmission intervals. Due to these constraints, achieving very high-frequency reporting was limited, with a 1-minute interval not reliably attainable and 2 minutes observed as the minimum stable interval during testing.

Network Constraints (Duty Cycle)

LoRaWAN duty cycle restrictions and TTN fair use policies limit the frequency of uplink and downlink messages, affecting how often sensors can transmit data and how frequently configuration updates can be applied.

Observed Behaviours

Reset After Connectivity Loss

It was observed that some sensors reverted to default settings (e.g., 10-minute reporting intervals) following prolonged loss of gateway or network connectivity.

Join/Rejoin Behaviour

Devices occasionally rejoined the network after interruptions, temporarily affecting data transmission continuity and reporting schedules.

This configuration approach enabled effective testing of sensor responsiveness while highlighting practical limitations associated with LoRaWAN communication and device-specific behaviour.

Webhook Integration

Data integration between The Things Network (TTN) and TagoIO was achieved using a webhook-based architecture, enabling real-time transmission of sensor data for storage, processing and visualisation.

Webhook Integration Setup

The webhook integration was configured within TTN by selecting the TagoIO integration template. This establishes a direct pipeline for forwarding uplink messages from the TTN application to TagoIO.

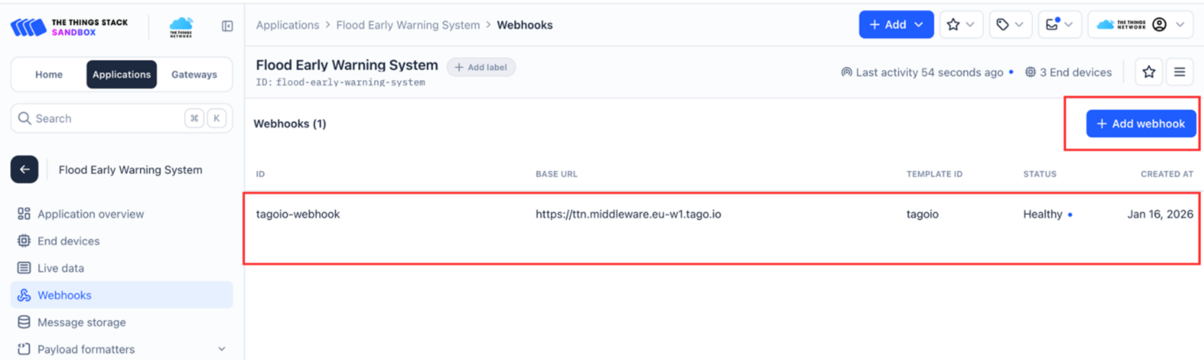

Figure 2020: TTN Webhook Integration Interface Showing Registered TagoIO Webhook, the Add Webhook Button and the Selection of TagoIO Integration Template (a)

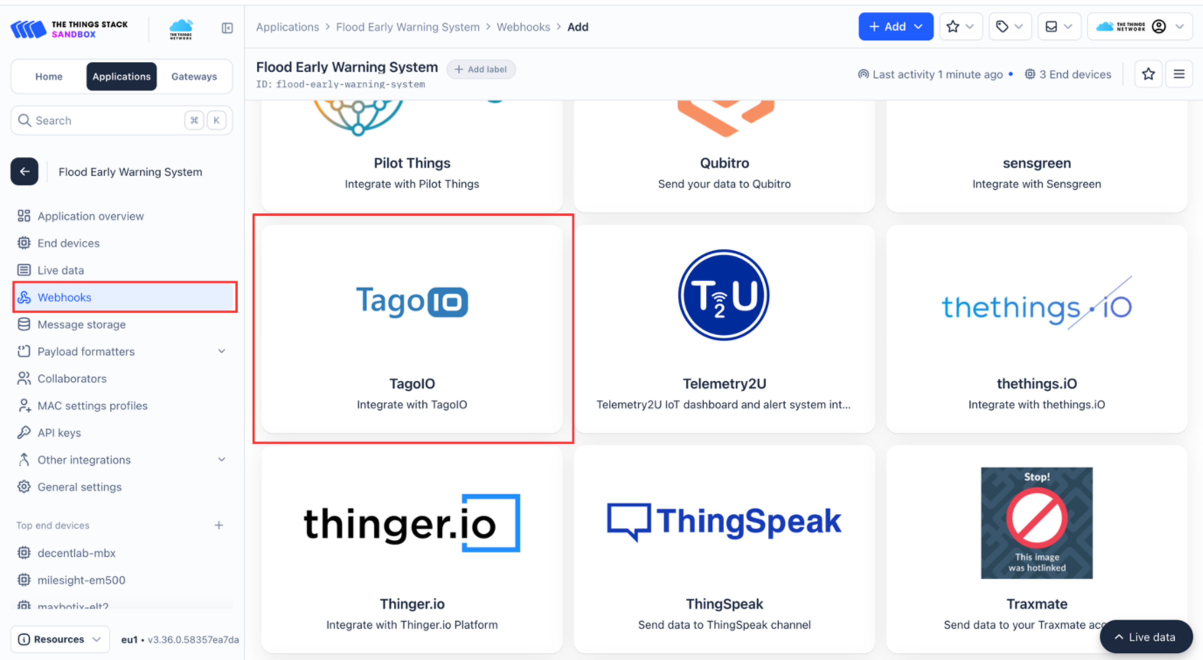

Figure 2121: TTN Webhook Integration Interface Showing Registered TagoIO Webhook, the Add Webhook Button and the Selection of TagoIO Integration Template (b)

Webhook Creation and Configuration

A webhook instance was created with a defined ID and a base URL, https://ttn.middleware.eu-w1.tago.io/, pointing to TagoIO, with the /uplink endpoint enabled to receive uplink messages only. This ensures that only relevant sensor data is transmitted downstream. The webhook status was verified as healthy, confirming successful configuration.

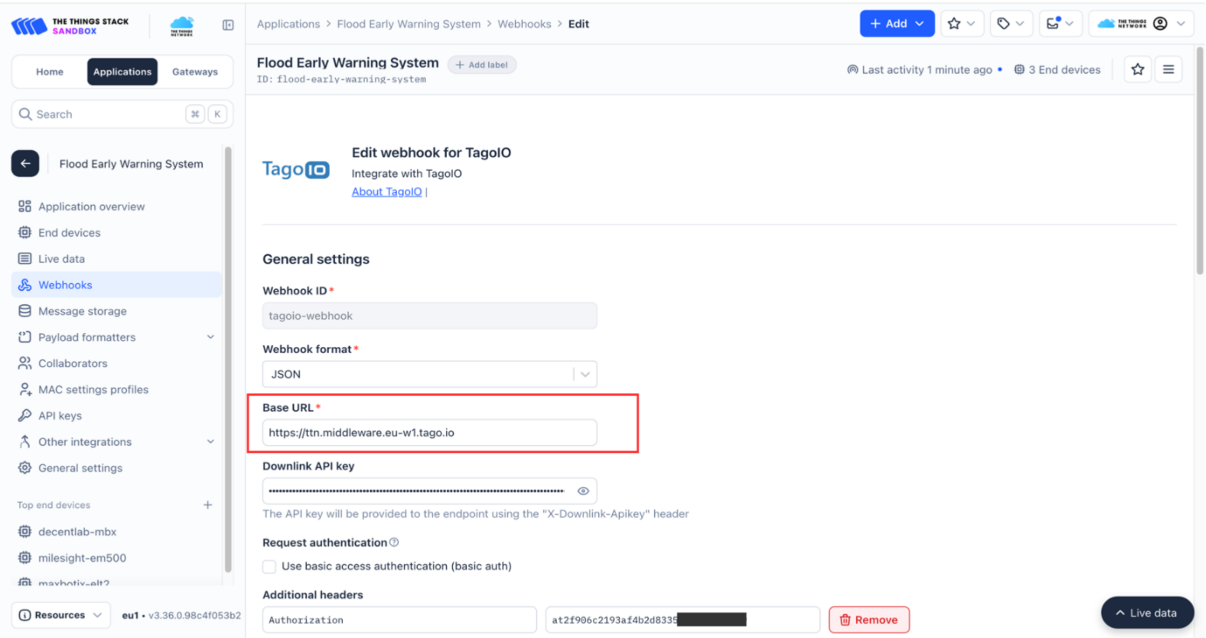

Figure 2222: Webhook Configuration showing the ID, base URL and Authorization Token and /uplink Endpoint (a)

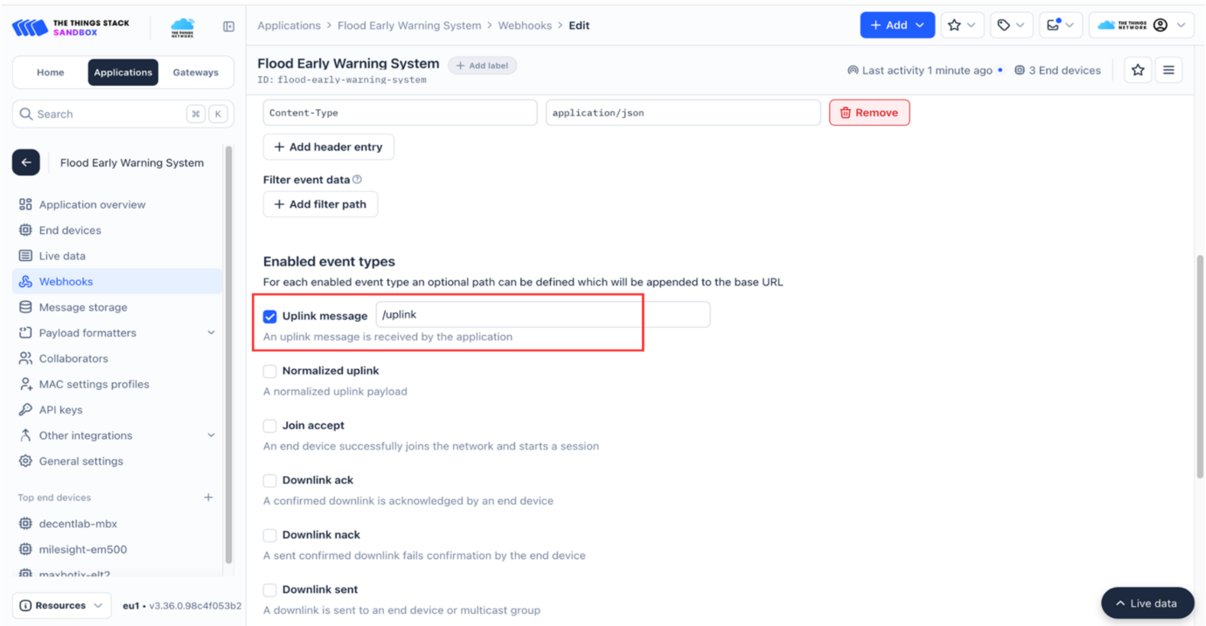

Figure 2323: Webhook Configuration showing the ID, base URL and Authorization Token and /uplink Endpoint (b)

Authentication Mechanism

Authentication between TTN and TagoIO was implemented using an authorization token generated in TagoIO. The token was created using the standard format within the TagoIO authorization service, it was then configured in TTN as part of the webhook setup. This ensures secure data transmission and proper device identification within TagoIO.

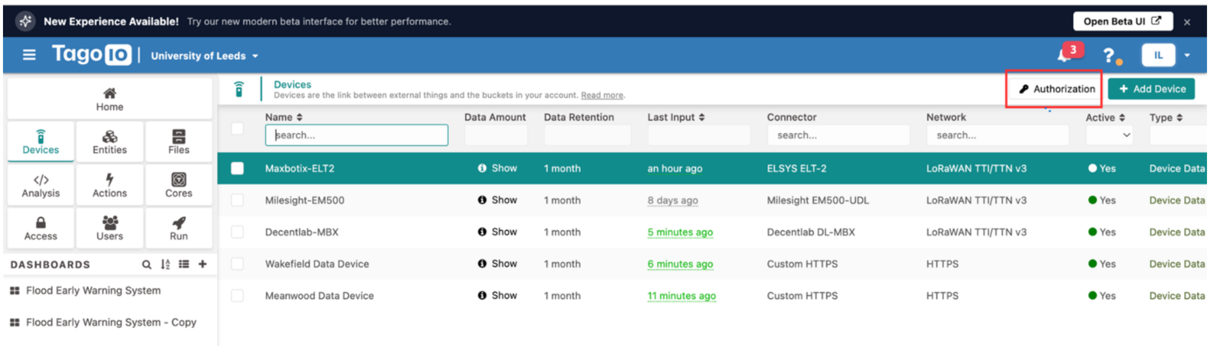

Figure 2424: TagoIO Authorization Token Generation and Configuration for Secure Webhook Authentication (a)

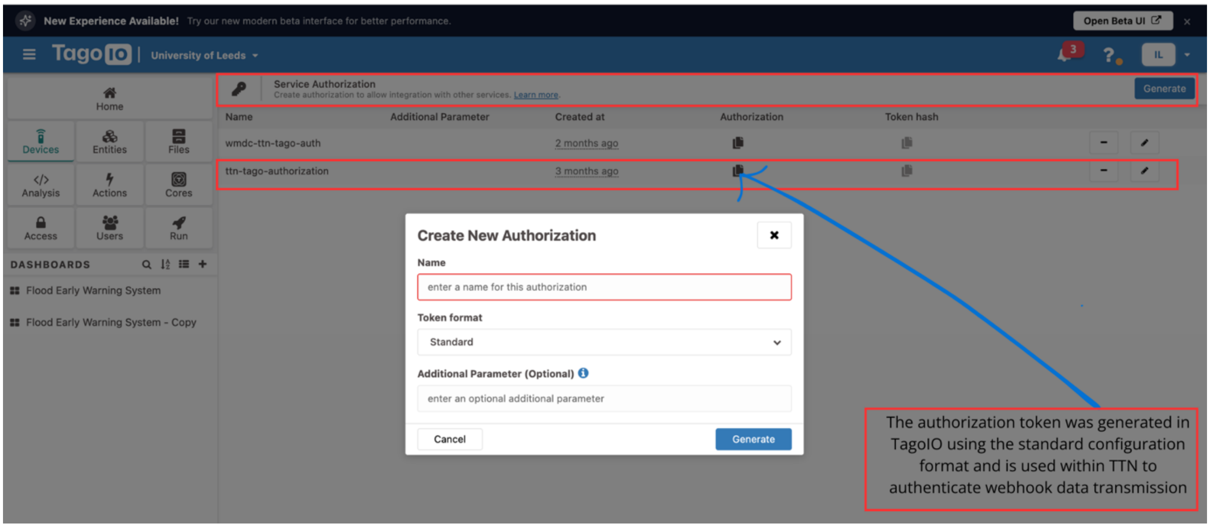

Figure 2525: TagoIO Authorization Token Generation and Configuration for Secure Webhook Authentication (b)

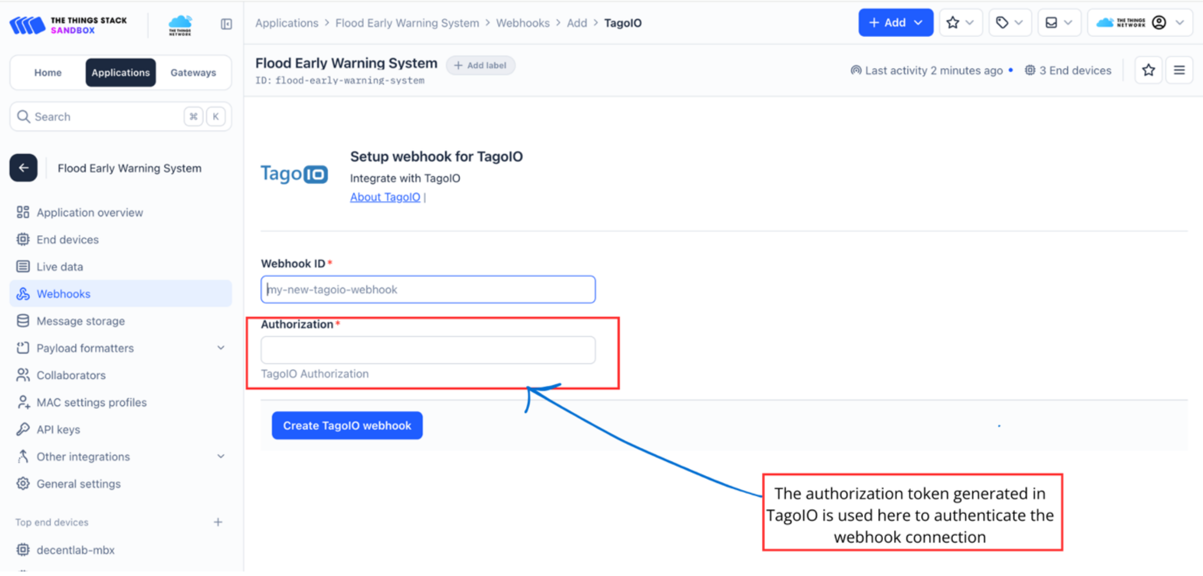

Figure 2626: TagoIO Authorization Token Generation and Configuration for Secure Webhook Authentication (c)

Multi-Sensor Data Ingestion Strategy

The data integration system supports ingestion from multiple sensors within a single TTN application. Each sensor transmits data that is forwarded through the same webhook pipeline into TagoIO. Sensor identity is preserved using metadata (e.g., device ID). Data is later separated and processed downstream

To simplify integration and optimise resource usage, the following design decisions were adopted:

- Single TTN Application: All sensors were registered under one application for easier management
- Centralised Webhook Integration: A single webhook was used to route all incoming data
- TagoIO-Based Data Handling: Data processing, separation and visualisation were handled within TagoIO

This architecture reduces complexity, avoids duplication and ensures scalability for future deployment.

TagoIO Platform Setup

The TagoIO platform was used for device management, data ingestion, storage, visualisation and dashboard development within the Flood Early Warning System. The platform enabled integration of multiple LoRaWAN sensors through The Things Network (TTN) and supported both real-time monitoring and historical data analysis.

Account and Workspace Setup

A dedicated TagoIO account and workspace were created for the Digital Smart Flood Warning Systems project and managed by the University of Leeds. The workspace served as the central environment for managing sensor devices, storing incoming data, developing dashboards and configuring integrations with The Things Network (TTN).

During account setup, the Europe West 1 (EU-W1) region was selected to align with the European deployment environment and minimise latency between TTN and TagoIO services.

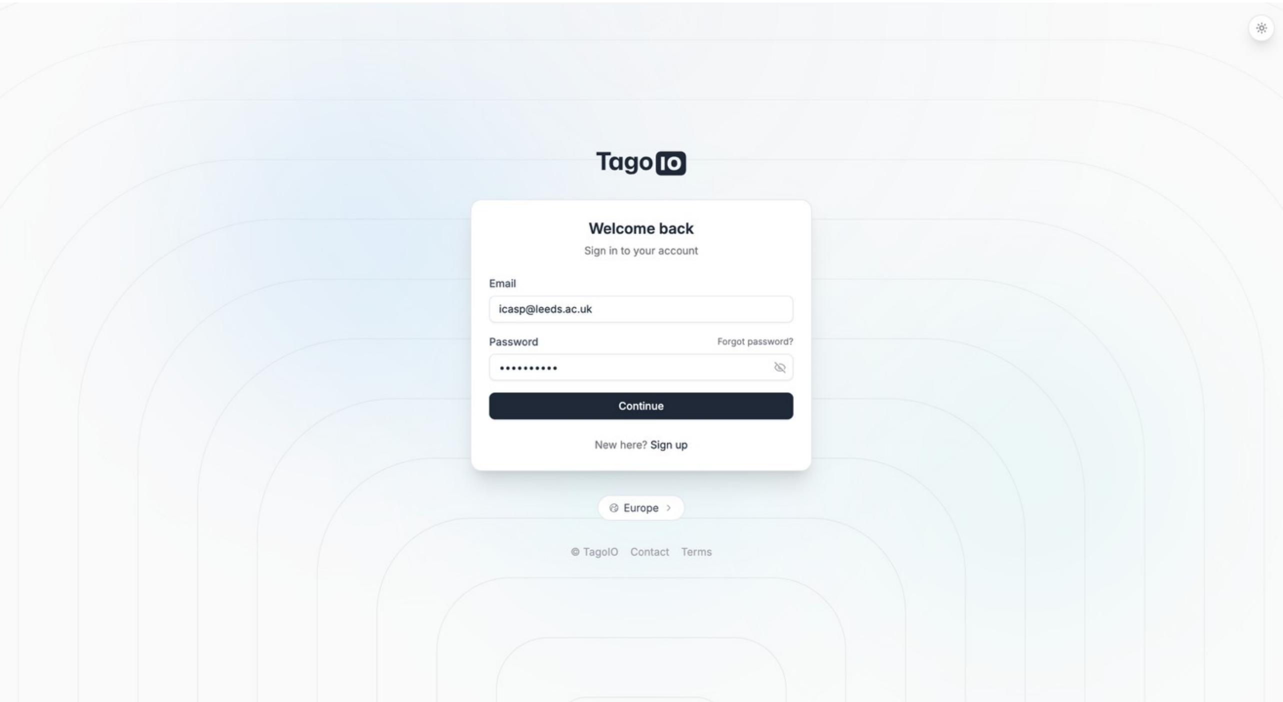

Figure 2727: TagoIO Account Login Page with Europe (EU-W1) Region Selected for Workspace Access

The workspace was configured to support device management, data storage, dashboard development, data analysis, authorisation and integration services.

Device Registration and Configuration

Within TagoIO, individual devices were created to represent each sensor deployed within the flood monitoring system. In TagoIO, a device is a digital entity that represents a data source. While a device may correspond to a physical sensor, it can also represent other sources of information, such as gateways, APIs, or virtual devices. In this project, each TagoIO device represented a single LoRaWAN water level sensor, enabling TagoIO to receive, store and manage data transmitted from the sensors via The Things Network (TTN).

To create a device, the user navigates to the Devices section within TagoIO and selects Add Device. A unique device name is then assigned to the sensor, after which the appropriate connector is selected. Network integration is configured using LoRaWAN TTI/TTN v3 and the data storage model is set to Device Data Optimised (Immutable) to support efficient storage and retrieval of time-series sensor data. The device is then linked to the corresponding sensor by entering its Device EUI before being saved and activated.

This configuration ensures that incoming sensor data is correctly associated with the corresponding device within TagoIO and is available for storage, analysis and visualisation.

Metadata Management

Metadata was used throughout the project to preserve sensor identity and provide additional context for incoming measurements. Information such as the device ID, Device EUI, sensor model, deployment location and gateway details can be stored alongside sensor measurements without affecting the measurement data itself.

The use of metadata allowed multiple sensors to share a common data ingestion pipeline while maintaining clear separation of data during storage, processing, analysis and visualisation. This approach also simplified dashboard development, troubleshooting and future expansion of the monitoring network.

Dashboard Development and Visualisation

Interactive dashboards were developed within TagoIO to support real-time monitoring, historical analysis and system diagnostics. The dashboards provide a central interface through which users can view sensor measurements, assess device health and monitor communication performance across the flood monitoring network (See figure 11).

Dashboard Creation

Dashboards were created using the TagoIO dashboard builder. Individual widgets were added and configured to display sensor data and system information in a format that is easy to interpret. The dashboard layout was designed to provide both an overview of current conditions and access to historical trends for more detailed analysis.

Widget Configuration

A range of widget types was used to visualise different aspects of system performance. Value widgets were used to display current sensor readings, while line charts provided historical trend analysis of water level measurements and other variables. Gauge widgets were configured to monitor signal quality metrics such as RSSI and SNR and tables were used to display device status information. Where appropriate, map-based visualisations were also used to show sensor locations and deployment sites.

Dashboard Benefits

By bringing data from multiple sensors into a single platform, the dashboard enables users to monitor current conditions, review historical performance, compare sensor behaviour and identify communication issues. This supports operational decision-making and provides a foundation for future flood warning and environmental monitoring applications.

The dashboard architecture has been designed to be scalable and can be extended to support additional sensors, automated alert generation and enhanced flood warning functionality as the project develops.

Flood Monitoring Dashboard

The dashboard was designed to provide users with a clear overview of both water level conditions and system performance. Real-time monitoring of water levels was combined with historical trend visualisation to support ongoing analysis. Additional information, including battery voltage, device activity status, RSSI, SNR and gateway communication metrics, was incorporated to assist with system diagnostics and maintenance.

Data Export and Processing Workflow

Sensor data was exported from TagoIO and processed using Python to support quality assurance, laboratory testing and performance evaluation. The workflow was designed to transform raw sensor measurements into structured datasets suitable for analysis and comparison against reference measurements collected during testing.

Data Export

Raw sensor data was exported from TagoIO in CSV format. The exported datasets contained water level measurements, battery voltage, communication metrics such as RSSI and SNR, timestamp information and device-specific metadata. These exports served as the primary data source for subsequent processing and analysis.

Data Processing Pipeline

Data processing was undertaken using Python, primarily using the Pandas library for data manipulation and transformation. The workflow began by importing the exported CSV files for each sensor and filtering records to the specific laboratory testing periods of interest. Variables not required for analysis were removed and data types were reviewed to ensure consistency across datasets.

Timestamp fields were standardised to a consistent format and timezone information was removed where necessary to simplify comparison with manually recorded reference measurements. The data was then transformed into an analysis-ready structure suitable for statistical evaluation and visualisation.

Data Cleaning

Several data cleaning procedures were applied to improve data quality and consistency. These included the removal of non-essential variables, validation of timestamp consistency, handling of missing values, removal of duplicate records and standardisation of variable names across sensors. These steps ensured that data generated by different sensor manufacturers could be analysed using a common workflow.

Output Datasets

Separate processed datasets were produced for the Decentlab, Maxbotix and Milesight sensors. These cleaned datasets formed the basis for the laboratory and field performance evaluations presented in the associated reports. The processed outputs also provided a consistent format for calculating summary statistics, assessing sensor accuracy and comparing sensor performance under different testing conditions.

Lessons Learned

Several important lessons were identified during the development and testing of the flood monitoring system. Public LoRaWAN gateways proved useful for sensor setup and initial connectivity testing; however, the dedicated MultiTech gateway provided more reliable and consistent communication during both laboratory and field deployments.

Different sensor configuration methods were also evaluated. NFC-based configuration, used by the Milesight and ELSYS sensors, was generally quicker and easier for routine parameter changes than the downlink-based configuration required by the Decentlab sensor.

The use of a centralised webhook integration simplified the overall system architecture by allowing data from multiple sensors to be managed through a single data pipeline. Forwarding raw payload data from TTN also provided greater flexibility for data cleaning, transformation and analysis within downstream applications.

Finally, while all sensors were successfully configured to transmit data at two-minute intervals for testing purposes, achieving a stable one-minute reporting interval proved challenging due to device-specific limitations and LoRaWAN network constraints. These findings highlight the importance of considering both technical and operational factors when designing and deploying LoRaWAN-based flood monitoring systems.

Future Development

The system developed in this project provides a strong foundation for future flood monitoring and early warning applications. Future improvements could include installing additional water level sensors to increase monitoring coverage and integrating other environmental sensors, such as rainfall and weather sensors, to provide more information about flood conditions.

The system could also be enhanced by introducing automatic alerts that notify users when water levels exceed predefined thresholds. These alerts could be delivered through email, SMS, or other communication channels. In the future, the platform could be linked with local authority flood management systems to support decision-making during flood events.

As more sensors are deployed and more data is collected, the system could be expanded to cover multiple catchments across West Yorkshire. Historical sensor data could also be used to develop flood forecasting models, helping to predict potential flooding events before they occur.

References

1. Bond, S., Ogundipe, F., Naz, F., Brown, H., and Keevil, G.(2026) Creating a digital smart flood warning system for surface water flooding in West Yorkshire: Laboratory Report. V1.0. A report developed in collaboration with Wakefield Council, Kirklees Council, Leeds City Council, Environment Agency and iCASP, under the West Yorkshire Flood Innovation Programme
2. Decentlab AG. (n.d.). DL-MBX Ultrasonic Level Sensor Datasheet. Available at: https://cdn.decentlab.com/download/datasheets/Decentlab-DL-MBX-datasheet.pdf (Accessed: June 2026).
3. ELSYS AB. (n.d.). ELT-2 Operating Manual. Available at: https://elsys.se/public/manuals/Operating%20Manual%20ELT-2.pdf (Accessed: June 2026).
4. LoRa Alliance. (2020). LoRaWAN® Link Layer Specification v1.0.4. Fremont, CA: LoRa Alliance. Available at: https://lora-alliance.org/wp-content/uploads/2021/11/LoRaWAN-Link-Layer-Specification-v1.0.4.pdf (Accessed: June 2026).
5. Milesight IoT Co., Ltd. (n.d.). EM500-UDL Ultrasonic Distance/Level Sensor User Guide. Available at: https://resource.milesight.com/milesight/iot/document/em500-udl-user-guide-en.pdf (Accessed: June 2026).
6. TagoIO. (n.d.). TagoIO Documentation. Available at: https://docs.tago.io/ (Accessed: June 2026).
7. The Things Industries (TTI). (n.d.). The Things Stack Documentation. Available at: https://www.thethingsindustries.com/docs/ (Accessed: June 2026).

Appendix A: Abbreviations and Acronyms

Abbreviation	Full Term
AWS	Amazon Web Services
DevEUI	Device Extended Unique Identifier
EU868	European 863–870 MHz LoRaWAN Frequency Band
GPS	Global Positioning System
IoT	Internet of Things
LoRaWAN	Long Range Wide Area Network
OTAA	Over-The-Air Activation
RSSI	Received Signal Strength Indicator
SNR	Signal-to-Noise Ratio
TTN	The Things Network
UoL	University of Leeds
WMDC	Wakefield Metropolitan District Council
WYFLIP	West Yorkshire Flood Innovation Programme
NFC	Near Field Communication
API	Application Programming Interface
MQTT	Message Queuing Telemetry Transport
AppKey	Application Key
JoinEUI	Join Extended Unique Identifier

Abbreviation	Full Term
RX2	Receive Window 2
SF	Spreading Factor
CSV	Comma-Separated Values
JSON	JavaScript Object Notation
TTI	The Things Industries
KPI	Key Performance Indicator