

FM4017 Project 2023

Exploration of LoRaWAN sensors and Infrastructure for IoT Applications



MP-13-23

Course: FM4017 Project, 2023

Title: Exploration of LoRaWAN Sensors and Infrastructure

This report forms part of the basis for assessing the students' performance on the course.

Project group: MP-13-23

Group participants: Majid Kadivartasooji
Iñigo Izaguirre
Gaveen Shamila Ranabahu

Supervisor: Hans-Petter Halvorsen

Project partner: Altibox

Summary:

The objective of this project is to develop a comprehensive open-source system that enables effortless monitoring and analysis of data collected by LoRaWAN sensors. The exploration of LoRaWAN devices and technology is discussed. A datalogging platform has been created to record sensor data onto the Dimension Four IoT platform. A web-based application that presents real-time sensor data in Tableau. A comprehensive compilation of historical data has been created.

Preface

This report is compiled for the FM4017 Project 2023 course by Master of Science students specializing in Industrial IT and Automation at the University of South-Eastern Norway in Porsgrunn. The assigned project involved the creation of an Internet of Things (IoT) solution utilizing LoRaWAN technology. System for storing and overseeing sensor data. Throughout the project, our project team diligently and enthusiastically pursued knowledge regarding. The user mentioned LoRaWAN technology, gadgets, Tableau, Dimension Four IoT platform, and Altibox ThingPark. Key components of the creation process include the implementation of a portal, data logging using Python, and data monitoring using Node.js, among other elements. A comprehensive open-source solution designed specifically for the University of South-Eastern Norway (USN). Mr. Daniel Wathne Warholm, an individual affiliated with Altibox AS, played a proactive role in our project and provided us with guidance. Throughout the entire implementation process. We express our heartfelt gratitude to him for his extensive knowledge.

We greatly appreciate the assistance and ongoing support provided to us in our endeavor. We express our gratitude to our project supervisor, Mr. HansPetter Halvorsen, for presenting us with an exceptional project assignment, granting us authority in decision-making, and demonstrating unwavering commitment.

We appreciate the involvement of three team members in our project and solution.

Porsgrunn, 09/22/2023

Majid Kadivartasooji
Gaveen Ranabahu
Iñigo Izaguirre

Contents

1	Introduction	9
1.1	Background	9
1.2	Project Objectives	9
1.3	System Architecture	10
2	IoT Network Technologies	11
2.1	LPWAN	11
2.1.1	LoRaWAN	11
2.1.2	Sigfox	19
2.1.3	A comparison between Sigfox and LoRaWAN	20
2.2	Zigbee	20
2.3	Cellular	21
3	Hardware Devices	23
3.1	Sensors	23
3.1.1	Adeunis Comfort Temperature/Humidity sensor	23
3.1.2	Adeunis Temperature sensor	24
3.1.3	Adeunis Contact Sensor	26
3.2	Field Test Device	27
3.3	Gateway	28
3.4	Hardware installation process	29
3.4.1	Comfort Sensor installation	29
3.4.2	Contact Sensor installation	30
3.4.3	Field Test Device installation	30
3.4.4	Temperature Sensor installation	30
3.5	Parameters and Identifiers for Sensors	31
3.6	LoRa/Sigfox IoT Configurator application	32
4	Altibox LoRaWAN Infrastructure	34
4.1	Altibox Thingpark portal	34
4.1.1	Network Survey	34
4.1.2	Wireless logger	35
4.1.3	Device Manager	35
4.1.4	Network Manager	39

5	Receiving data from Sensors using Thingpark X and Dimension4	40
5.1	ThingPark X	40
5.1.1	Drivers	42
5.1.2	Connections	42
5.1.3	Flows.....	44
5.2	Dimension4.....	45
5.2.1	Tenant.....	46
5.2.2	Spaces	47
5.2.3	Point.....	48
5.2.4	Signal	48
6	Datalogging Application.....	50
6.1	An overview of Pipedream	50
6.2	Reviewing Events.....	52
6.3	Configuring the devices communication path	52
6.3.1	Configuring the Application Server.....	53
6.3.2	Configuring the AS routing profiles	53
6.3.3	Assigning the routing path to Devices to communicate.	54
6.4	Building up the Workflow	55
6.4.1	Configure the Trigger	55
6.4.2	Adding the captured data.....	56
6.4.3	Adding received data to Google Drive	58
6.4.4	Data Stores.....	59
6.4.5	Getting the data from JSON file received	59
6.4.6	Aspects of the JSON file	61
6.4.7	Capturing the data	63
6.4.8	Decoding the data.....	64
6.4.9	Using TypeScript to decode the data.....	66
7	Monitoring Application	67
7.1	Tableau Overview.....	67
7.2	Connecting to the Tableau	68
7.3	Developed Dashboards	70
7.3.1	Outdoor temperature data Dashboard	70
7.3.2	Indoor temperature data Dashboard.....	70

7.3.3	Indoor humidity data Dashboard.....	71
7.3.4	Contact data Dashboard	71
7.3.5	Summary Dashboard.....	72
8	Discussion.....	73
9	Conclusion	74

Nomenclature

ABP: Activation By Personalization

ADR: Adaptive Data Rate

AES: Advanced Encryption Standard

API: Application Programming Interface

AS: Application Server

CSS: Chirp Spread Spectrum

DevEUI: Device Extended Unique Identifier

FTD: Field Test Device

GPS: Global Positioning System

GraphQL: Graph Query Language

HTTP: Hypertext Transfer Protocol

IP: Internet Protocol

IP68: Ingress Protection rating

IoT: Internet of Things

ISM: Industrial, Scientific and Medical frequency bands

JSON: Java Script Object Notation

JSLT: JavaScript for Linked Data

JoinEUI: Join Extended Unique Identifier

LoRaWAN: Long Range Wide Area Network

LPWAN: Long Power Wide Area Network

MAC: Media Access Control

MIC: Message Integrity Code

MQTT: Message Queuing Telemetry Transport

ODBC: Open Database Connectivity

OLE DB: Object Linking and Embedding Database

OTAA: Over The Air Activation

PER: Packet Error Rate

SF: Spreading Factor

SNR: Signal-to-Noise Ratio

SQL: Structured query language

UNB: Ultra-narrowband

URL: Uniform Resource Locator

USN: University of South-Eastern Norway

XML: Extensible Markup Language

1 Introduction

The world is continuously more connected thanks to Internet of Things (IoT). For the development of smart technology IoT can be defined as a way to interconnect devices and objects through a network. In this project LoRaWAN is a wireless communication method that offers an energy-efficient, long-range communication solution for many possible applications.

This project aims to make an exploration of LoRaWAN sensors and infrastructure in the context of Internet of Things applications. LoRaWAN infrastructure from Altibox will be used during the project and some LoRaWAN sensors will be used and installed on the USN Porsgrunn campus to get data from them.

1.1 Background

In the world of Internet of Things (IoT), Altibox is a leading digital service provider in Norway and has been an important part of the project. Collaborating with Altibox, that have a long background using LoRaWAN sensors is crucial to get a good understanding of how these sensors work. Altibox has built a LoRaWAN with big competence as the fiber network they offer. More than 100 municipalities and 1 million households are today covered by Altibox LoRaWAN network.

Altibox LoRaWAN infrastructure becomes very important as we strategically install LoRaWAN sensors on the USN Porsgrunn campus. This installation aims to capture real-world data and discover the practical applications of LoRaWAN mainly within Altibox ecosystem.

1.2 Project Objectives

Some objectives are defined with the idea of understanding the main objective and needs of the project. Objectives are defined from the project description given at the beginning of the project.

- Get an overview of LoRaWan and other relevant protocols used in IoT in general and in context of this work.
- Get an overview of the Altibox LoRaWan Infrastructure.
- Understand how to log and monitor Data using the available LoRaWAN sensors.
- The system should be Open-source and should be available at GitHub with proper documentation.
- Use Microsoft Teams and GitHub during project planning and development.
- Document the system in the form of a technical report, documentation on GitHub and e.g., on YouTube.

1.3 System Architecture

A system sketch has been developed in order to explain the overall overview of the project and give a general idea of the steps followed when installing the project sensors to achieve the defined objectives in the project. In Figure 1 can be seen the system sketch where the different sensors are displayed connected to a cloud using gateway and LoRaWAN technology. The data can be stored in a database or be presented in other devices. This process will be analyzed during the project.

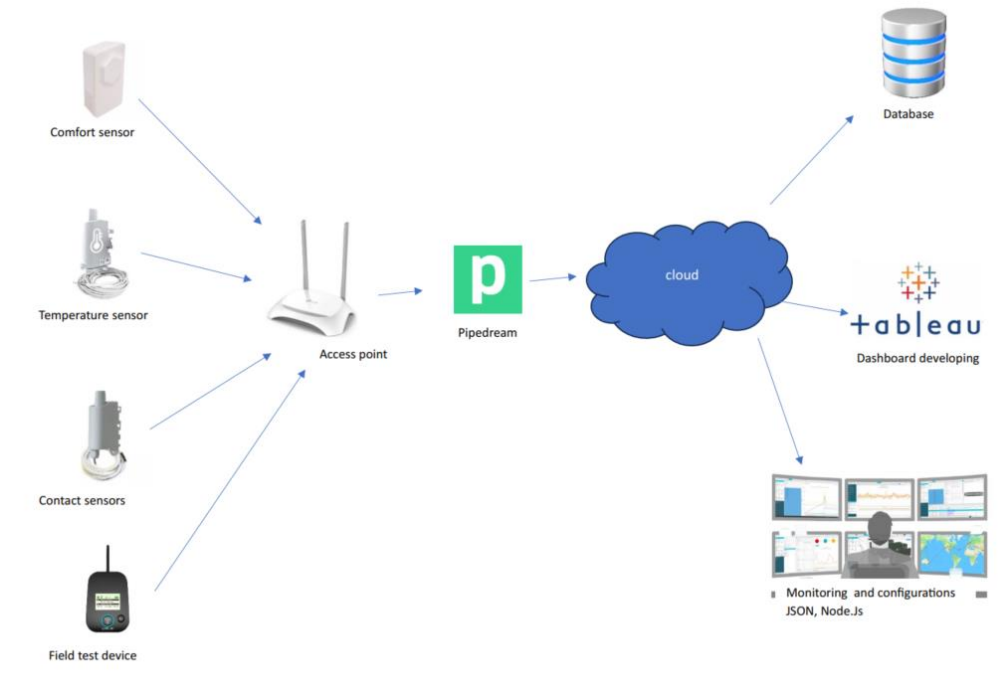


Figure 1: System sketch of the project.

2 IoT Network Technologies

In this chapter an overview of LoRaWAN and other relevant network technologies used in IoT will be given. As the main technology explored in this project is the LoRaWAN technology this will be the main focus of this chapter. However, a general overview is needed to get a good understanding of all the protocols so other protocols will be explained as well.

2.1 LPWAN

Low power wide area network (LPWAN) is a wireless technology that enables long-range communication between low-power devices. LPWANs are ideal for connecting Internet of Things (IoT) devices, such as sensors, smart meters, and asset trackers, that need to send small amounts of data over long distances.

Some common LPWAN technologies include:

2.1.1 LoRaWAN

LoRaWAN (Long Range Wide Area Network) is a low-power, wide-area network (LPWAN) technology designed to connect battery-powered devices over long distances. It is a popular choice for IoT applications, such as asset tracking, environmental monitoring, and smart city solutions.

LoRaWAN uses a spread spectrum modulation technique called Chirp Spread Spectrum (CSS) to achieve its long range and low power capabilities. CSS spreads the signal over a wider frequency band, which makes it more resistant to interference and allows it to travel further.

2.1.1.1 LoRaWAN Network Architecture and Components

LoRaWAN networks are typically made up of several components [1].

- **LoRaWAN Gateway:** A LoRaWAN gateway is a device that acts as a bridge between LoRaWAN devices and the LoRaWAN network server. It receives uplink messages from LoRaWAN devices and forwards them to the network server. It also receives downlink messages from the network server and forwards them to LoRaWAN devices.
- **LoRaWAN Join Server:** A join server in LoRaWAN is responsible for authenticating and activating LoRaWAN end devices, and generating and distributing session keys to them.
- **LoRaWAN Network Server:** A network server in LoRaWAN is a software component that is responsible for managing and coordinating the communication between LoRaWAN devices and applications

- **LoRaWAN Application Server:** LoRaWAN application servers are responsible for processing and managing application-specific data messages received from end devices. They also generate and send downlink messages to end devices.
- **End devices:** These are the devices that collect data and send it to the network. End devices can be battery-powered sensors, actuators, or other devices.

Figure 2 depicts the LoRaWAN network architecture.

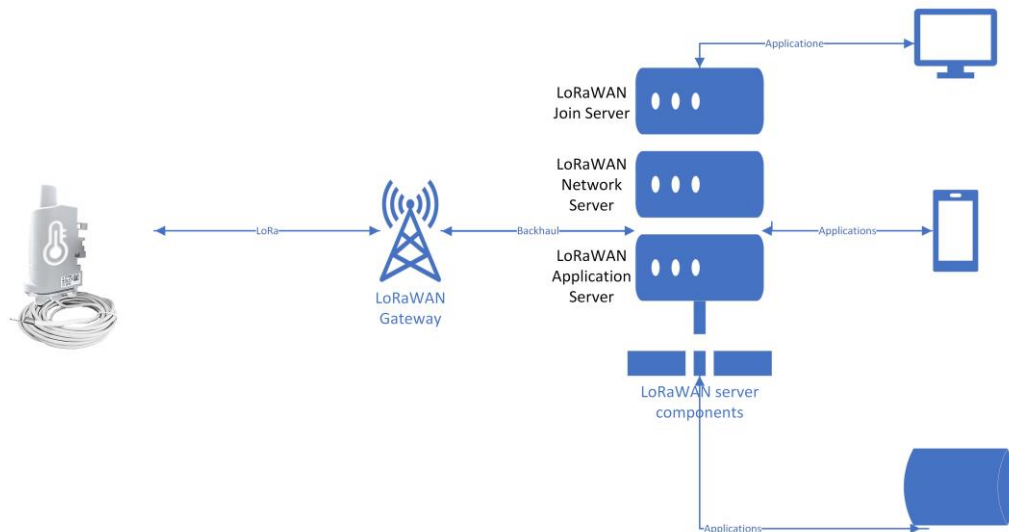


Figure 2: LoRaWAN network architecture.

2.1.1.2 LoRaWAN Technology Stack

The LoRaWAN technology stack is a layered architecture that consists of the several components[2]. Figure 3 depicts the LoRaWAN technology stack.

- **Physical layer:** The physical layer is responsible for the transmission and reception of LoRa signals. It defines the modulation scheme, coding rate, and other physical parameters of LoRa communication.
- **Media access control (MAC) layer:** The MAC layer is responsible for managing access to the LoRa channel and ensuring the reliable delivery of messages. It defines the framing of LoRa messages, the retransmission mechanism, and other MAC protocols.
- **Application layer:** The application layer is responsible for providing services to applications that use LoRaWAN. It defines the APIs that applications can use to send and receive messages, as well as the data formats that are used.

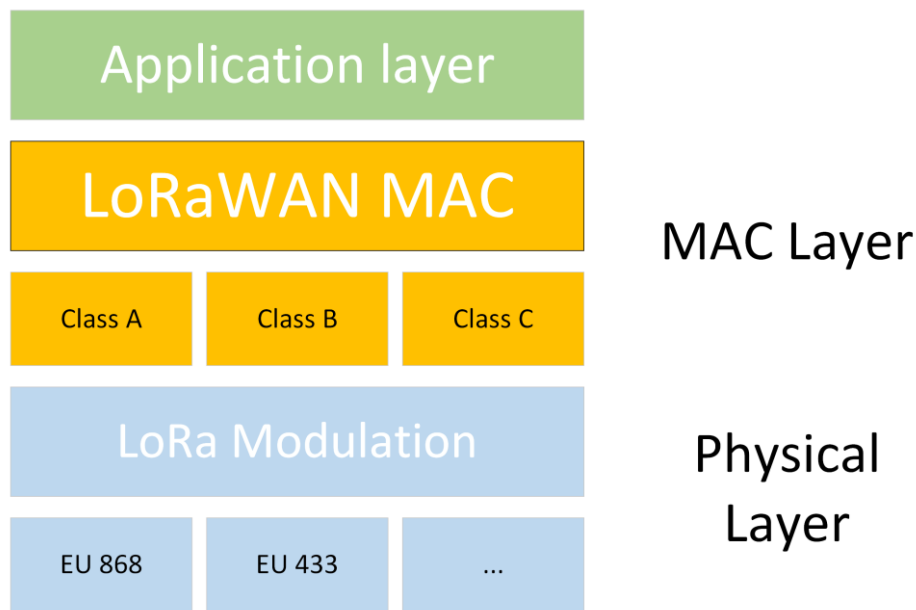


Figure 3: LoRaWAN technology stack

- **EU 868:** EU868 is a frequency band allocated for LoRaWAN usage within Europe. It encompasses the range of 863-870 MHz. This frequency band is a popular and well-balanced choice for LoRaWAN deployments in European regions, offering a combination of range, data rate, and power efficiency [3].

EU868 is divided into eight channels with specific frequencies:

- 868.1 MHz
- 868.3 MHz
- 868.5 MHz
- 867.1 MHz
- 867.3 MHz
- 867.5 MHz
- 867.7 MHz
- 867.9 MHz
- Of these, three channels are mandatory, operating at 868.1 MHz, 868.3 MHz, and 868.5 MHz. The remaining five channels are optional, providing flexibility for LoRaWAN network operators to extend capacity or coverage as needed.

2.1.1.3 LoRaWAN Security

LoRaWAN security relies on a combination of authentication, encryption, and integrity protection. Authentication ensures that only permitted devices can engage in network communication. Encryption safeguards the confidentiality of data transmitted across the network, while integrity protection guarantees that data remains unaltered during transmission [4].

For encryption and decryption, LoRaWAN employs the Advanced Encryption Standard (AES), a symmetric encryption algorithm using the same key for both processes. This key is shared between the end device and the application server. There are two forms of encryption used:

- Application payload encryption secures the confidentiality of data sent from end devices to the application server. This employs the Counter (CTR) mode of operation, which uses a counter to generate a unique nonce for each data block, and this nonce is utilized for data encryption.
- Frame header encryption safeguards the integrity of messages and prevents unauthorized devices from mimicking messages. It utilizes the Cipher Block Chaining (CBC) mode of operation, which employs the previous block of ciphertext to encrypt the current block of plaintext, making it harder for attackers to tamper with messages without detection.

The AES key is used for encrypting the application payload, while the CBC initialization vector (IV) is used for encrypting the frame header. The encrypted data is then transmitted over the LoRaWAN network.

The decryption process involves the application server decrypting the application payload using the AES key, and the network server decrypting the frame header using the CBC IV. Once decryption is complete, the application server can access and process the data accordingly.

Figure 4 shows the network architecture and security aspects.

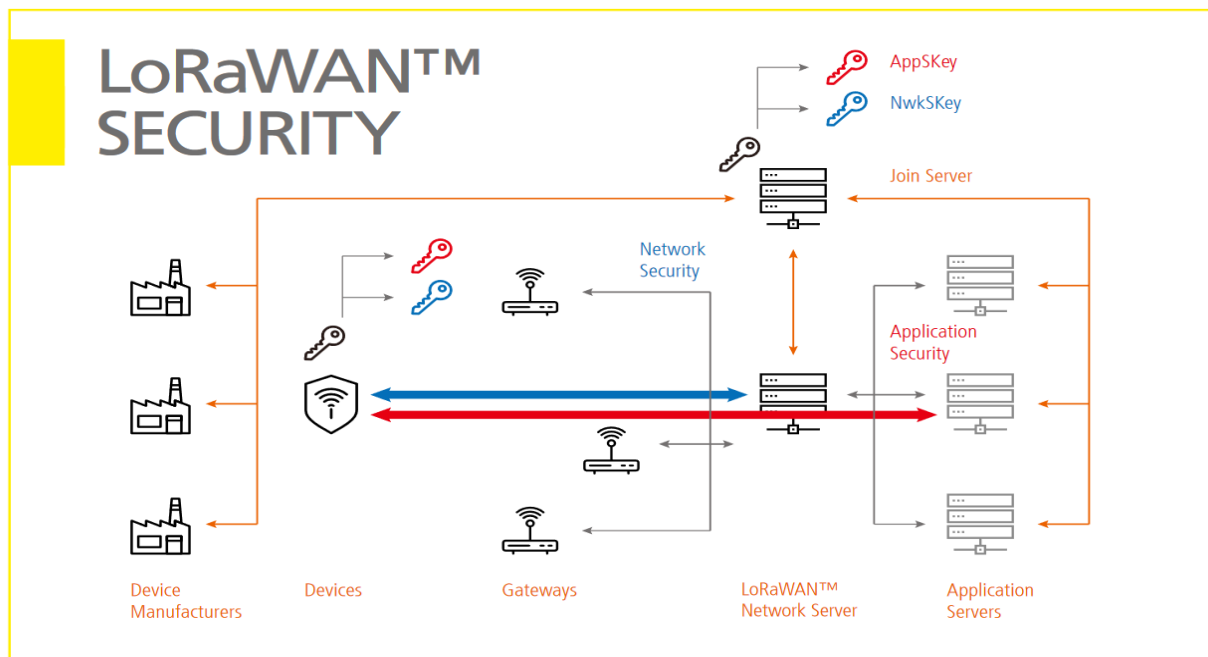


Figure 4: The security in LoRaWAN network.[5]

2.1.1.4 LoRaWAN Spreading Factor

The LoRaWAN spreading factor (SF) is a critical parameter that determines the balance between communication range, data speed, and power usage. SF quantifies how much a signal is stretched over time. A higher SF means a more stretched signal, resulting in a longer message

transmission time. However, it also extends the communication range and reduces power consumption.

LoRaWAN offers support for six SF options: SF7, SF8, SF9, SF10, SF11, and SF12. Among them:

- SF7 provides the quickest data transfer but has the shortest range and consumes more power.
- SF12 is the slowest SF, offering the longest communication range and the lowest power consumption.

The choice of SF depends on the specific requirements of the application. For instance, when prioritizing long-range communication and energy efficiency, SF12 is a suitable choice. Conversely, if a high data rate is needed, SF7 is the preferred option. The selection of SF should align with the particular needs and constraints of the application to achieve optimal performance.

Table 1 shows the different spreading factor and their properties.

Spreading factor	Data rate (kbps)	Range (km)	Power consumption
SF7	5.4	1-2	High
SF8	3.75	2-4	Medium
SF9	2.7	4-8	Low
SF10	1.77	8-16	Very low
SF11	1.1	16-32	Extremely low
SF12	0.72	32-64	Ultra low

Table 1: Different spreading factor in LoRaWAN network

2.1.1.5 LoRaWAN Messages Format

LoRaWAN messages are structured in a specific way to ensure effective and dependable communication between LoRaWAN devices and the network. The format of LoRaWAN messages is defined in the LoRaWAN 1.1 specification and comprises the following components:

1. Preamble: The preamble is a sequence of bits that serves to synchronize the receiver with the transmitter.
2. PHYPayload: The PHYPayload encompasses the MAC header, MAC payload, and MIC (Message Integrity Code).
3. Frame Header (PHDR): The FHDR contains message details such as source and destination addresses, message type, and spreading factor.
4. PHYPayload: The PHYpayload holds either application data or MAC commands.
5. CRC: The CRC is employed to validate the integrity of the message.

LoRaWAN messages can be categorized into uplink messages and downlink messages. Uplink messages originate from LoRaWAN devices and are sent to the network server. Downlink messages, conversely, are dispatched from the network server to LoRaWAN devices. Figure 5 shows the format of messages in LoRaWAN networks.

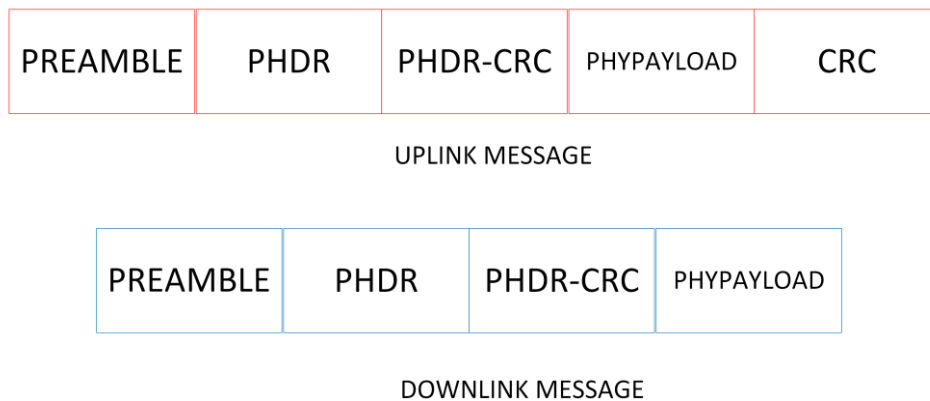


Figure 5: Message formats in LoRaWAN network

To facilitate efficient transmission and easy storage and processing, LoRaWAN messages are encoded using the Base64 encoding scheme. This encoding method optimizes the transfer of LoRaWAN messages through the air and enhances their manageability.

2.1.1.6 Classes of the Sensors in LoRaWAN Architecture

LoRaWAN devices are classified into three classes: Class A, Class B, and Class C. The main difference between the classes is the way they receive downlink messages from the gateway.

Downlink messages are messages sent from the network server to the end device, while uplink messages are messages sent from the end device to the network server.

2.1.1.6.1 Class A

Class A devices are the simplest and most power-saving class of LoRaWAN devices. Class A devices only listen for downlink messages during two brief time windows after they have sent an uplink message. These time windows are called RX1 and RX2. The RX1 window opens immediately after the uplink message is sent, and the RX2 window opens a short time later (typically 1-2 seconds). The duration of the RX1 and RX2 windows is configurable, but it must be long enough for the device to receive a downlink message. If the device does not receive a downlink message during either of the receive windows, it will not receive any other downlink messages until it sends the next uplink message. Class A devices are ideal for applications where battery life is the most important consideration. Figure 6 shows the transmission packages in class A.

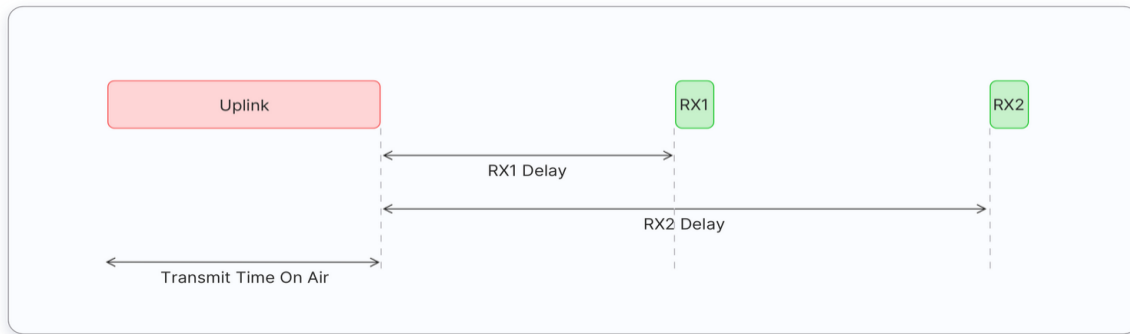


Figure 6 Transmission in class A [6]

2.1.1.6.2 Class B

Class B devices are more power-hungry than Class A devices, but they offer lower latency for downlink communication. Class B devices listen for downlink messages in the RX1 and RX2 windows, just like Class A devices, but they also open additional receive windows at regular intervals. These additional receive windows are called ping slots. The frequency of the ping slots is configurable, but it is typically set to once every few seconds. Class B devices listen for downlink messages in the ping slots for a short period of time (typically 1-2 seconds). If the device receives a downlink message during a ping slot, it will not listen for any more downlink messages until the next ping slot. Class B devices are ideal for applications where latency is important, but battery life is still a consideration. Class B end devices are suitable for both monitoring sensors as well as actuators. Figure 7 shows the transmission procedure and timing in class B.

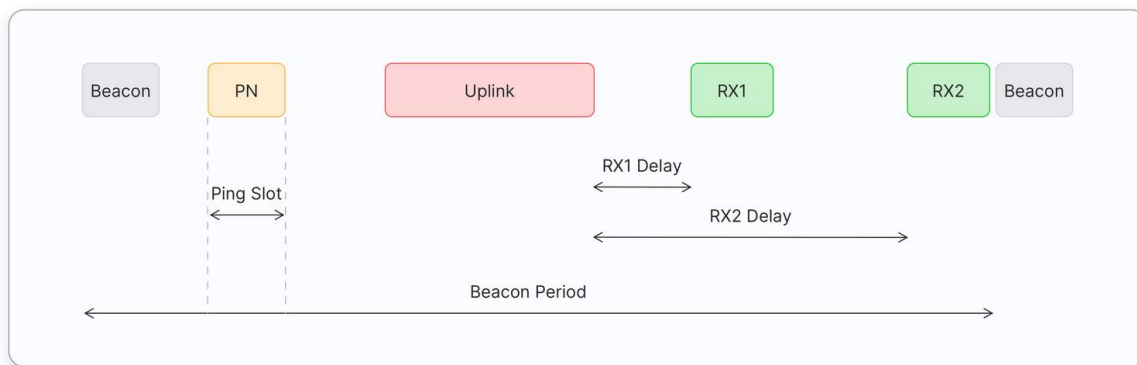


Figure 7: Transmission in class B [6]

2.1.1.6.3 Class C

Class C devices are the most power-hungry class of LoRaWAN devices, but they offer the lowest latency for downlink communication. Class C devices have their receive windows open continuously, except when they are transmitting an uplink message. This means that Class C devices can receive downlink messages at any time. However, the high-power consumption of

Class C devices makes them unsuitable for battery-powered applications. Class C devices are ideal for applications where latency is critical and battery life is not a major concern. Figure 8 depicts the transmission procedure and timing in class C.

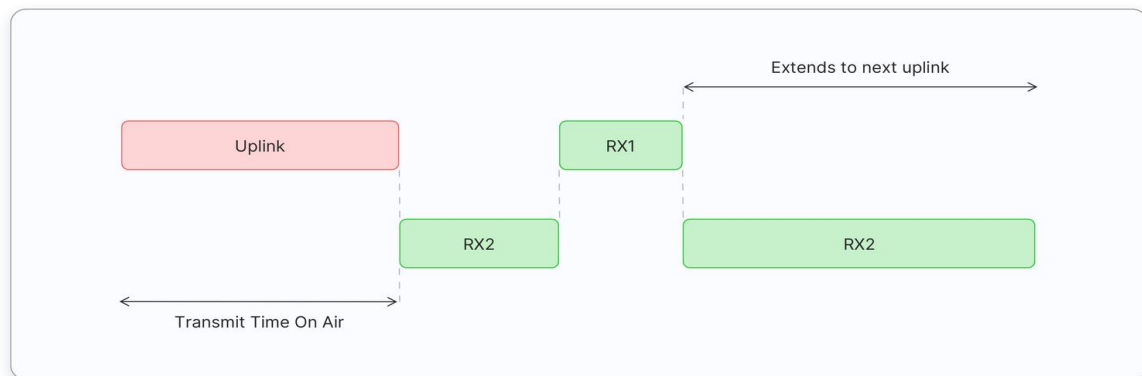


Figure 8:Transmission in class C [6]

In overall, LoRaWAN has a number of advantages over the other IoT technologies, including:

- Long range: LoRaWAN can reach distances of up to 15 kilometers in rural areas and up to 5 kilometers in urban areas.
- Low power: LoRaWAN devices can operate for years on a single battery.
- Wide coverage: LoRaWAN networks can be deployed in a variety of environments, including urban, rural, and underground.
- Secure: LoRaWAN uses encryption to protect data from unauthorized access.

LoRaWAN is a mature technology that is supported by a wide range of vendors. It is a good choice for a variety of IoT applications where long range, low power, and wide coverage are important.

Here are some of the applications of LoRaWAN:

- Asset tracking: LoRaWAN can be used to track the location of assets, such as vehicles, equipment, and livestock.
- Environmental monitoring: LoRaWAN can be used to monitor environmental conditions, such as temperature, humidity, and air quality.
- Smart city solutions: LoRaWAN can be used to create smart city solutions, such as smart parking, smart lighting, and smart waste management.

- Industrial IoT: LoRaWAN can be used in industrial IoT applications, such as predictive maintenance and asset monitoring.

2.1.2 Sigfox

Sigfox is a proprietary low-power wide-area network (LPWAN) technology that uses ultra-narrowband (UNB) modulation to transmit data over long distances at low power consumption. It has global coverage and is well-suited for applications that require long-range communication with small devices that have limited battery life. Sigfox uses a UNB modulation technique that transmits data at a very low data rate (100 bps). This makes Sigfox very efficient in terms of power consumption, allowing devices to operate for several years on a single battery. Sigfox has global coverage with over 70 countries covered. This makes Sigfox a good choice for applications that need to operate across a large geographical area [7].

Like LoRaWAN, Sigfox is also a well-suited for a wide range of applications, like Asset tracking, Environmental monitoring, Smart city infrastructure, Industrial automation and Healthcare monitoring. Figure 9 shows the network structure of Sigfox.

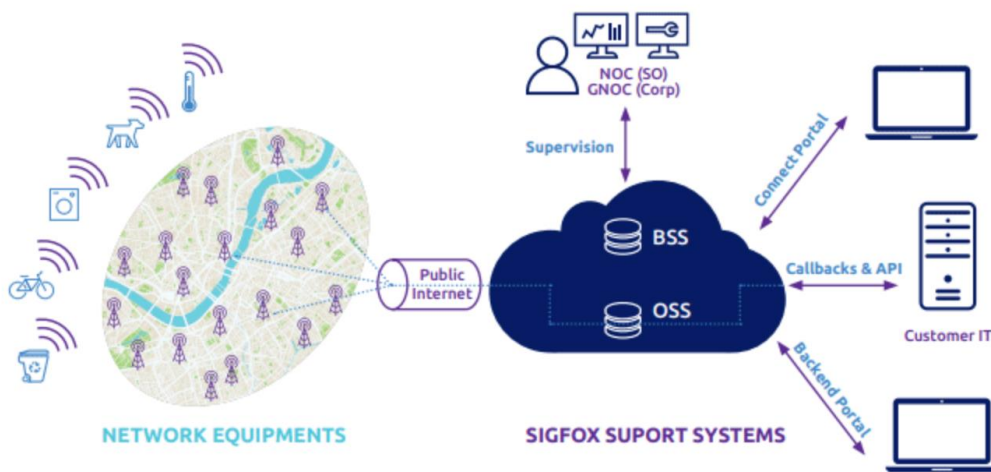


Figure 9: Sigfox network architecture [8].

2.1.3 A comparison between Sigfox and LoRaWAN

Sigfox is a proprietary low-power wide-area network (LPWAN) technology that uses ultra-narrowband (UNB) modulation to transmit data over long distances at low power consumption. It has global coverage and is well-suited for applications that require long-range communication with small devices that have limited battery life.

LoRaWAN is an open-standard LPWAN technology that uses LoRa, a spread spectrum modulation technique. It is designed to be scalable and flexible, and it can be deployed on either public or private networks. LoRaWAN also has global coverage and is well-suited for applications that require long-range communication with small devices that have limited battery life. However, it offers higher data rates and more flexibility than Sigfox.

Here is a Table 2 that summarizes the key differences between Sigfox and LoRaWAN:

<i>Feature</i>	Sigfox	LoRaWAN
Technology	Proprietary	Open standard
Modulation	Ultra-narrowband (UNB)	Spread spectrum (LoRa)
Data rate	100 bps	Up to 50 kbps
Range	Up to 50 km in rural areas	Up to 10 km in urban areas, up to 40 km in rural areas
Battery life	Several years	Several years
Coverage	Global (over 70 countries)	Global (over 170 countries)
Cost	Low	Low

Table 2:Differences between Sigfox and LoRaWAN.

2.2 Zigbee

Zigbee is a low-power, short-range wireless mesh network standard targeted at battery-powered devices in wireless control and monitoring applications. Zigbee delivers low-latency communication. Zigbee chips are typically integrated with radios and with microcontrollers. Zigbee operates in the industrial, scientific and medical (ISM) radio bands, including 2.4 GHz in most jurisdictions worldwide.

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low bandwidth needs, designed for small scale projects which need wireless connection. Hence, Zigbee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network [9].

Zigbee networks are self-organizing and self-healing, meaning that they can automatically configure themselves and recover from lost or failed nodes. Zigbee networks are also scalable to support a large number of devices.

Zigbee offers a number of advantages, such as low power consumption, long range(Zigbee devices can communicate over distances of up to 100 meters), scalability, reliability and security(Zigbee networks use a variety of security features to protect data from unauthorized access).

Some of the disadvantages of Zigbee are Lower data rates than other IoT technologies and it is more complex to set up and configure than other IoT technologies.

Figure 10 shows a sample diagram of a Zigbee network.

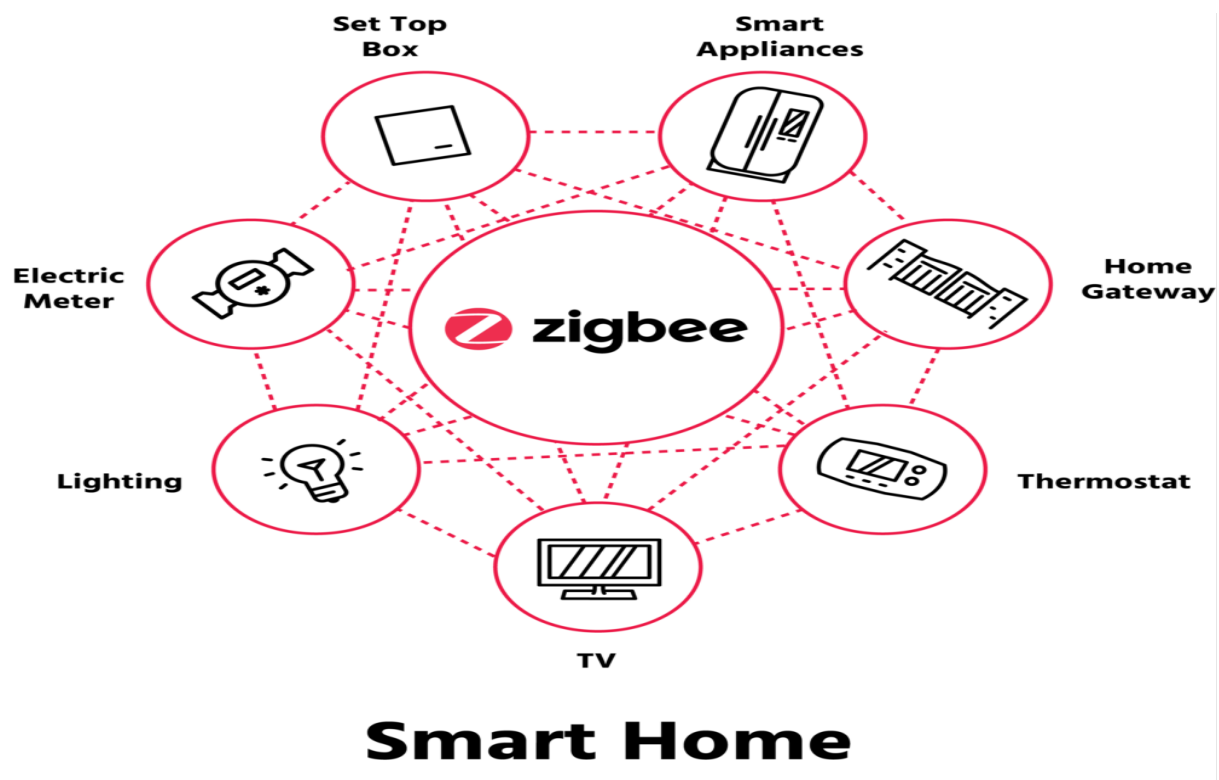


Figure 10: A sample diagram of a smart home using ZigBee [10].

2.3 Cellular

Cellular networks are a type of wireless network that divides a large geographic area into smaller cells. Each cell is served by a base station, which is a fixed-location transceiver. Cellular networks use radio waves to transmit data between the base stations and mobile devices [11].

Cellular networks are designed to provide coverage over a large area, including rural areas and remote locations. This is achieved by using multiple base stations, each of which covers a

relatively small area. As a mobile device moves from one cell to another, the call is handed off seamlessly to the next base station.

Cellular networks are a popular choice for IoT applications because they offer a number of advantages, including[12]:

- **Wide coverage:** Cellular networks provide coverage over a large area, including rural areas and remote locations. This makes them ideal for connecting IoT devices that are deployed in remote areas or that need to be able to communicate with a central server from anywhere in the world.
- **High capacity:** Cellular networks can support a large number of users simultaneously. This is important for IoT applications because there are billions of IoT devices connected to the internet and that number is expected to grow significantly in the coming years.
- **Reliability:** Cellular networks are very reliable, even in areas with high traffic. This is important for IoT applications because they often need to be able to transmit data reliably and in real time.
- **Security:** Cellular networks use a variety of security features to protect data from unauthorized access. This is important for IoT applications because they often collect and transmit sensitive data.

In Figure 11, a sample of IoT network using cellular technology can be seen.



Figure 11:Cellular IoT [13].

3 Hardware Devices

In this chapter hardware devices used during the project will be described. The used sensors, test and network devices will be exposed and their installation and activation in Altibox IoT platform will be explained. Before moving forward, it is important to discuss some important points.

3.1 Sensors

The sensors used during the project are provided by Adeunis company and have been installed by the participants of this project. Each sensor is explained in this chapter.

3.1.1 Adeunis Comfort Temperature/Humidity sensor

The Adeunis Comfort is a LoRaWAN-enabled sensor that measures temperature and humidity. It is designed for indoor use and has an IP20 rating, making it to be used indoors [14].

The sensor has a temperature range of -20°C to 60°C and a humidity range of 0 to 100%. It has a battery life of up to 10 years, depending on the frequency of data transmission. It can be configured to send data periodically or when thresholds are exceeded.

The sensor also has a number of additional features, including:

- **Autonomy optimization:** The sensor can be configured to store data for a period of time before sending it to the network, which can help to extend the battery life.
- **Accessibility of the data:** The sensor can be configured to send data to multiple gateways, which can help to ensure that the data is always available.
- **Alarm repetition in case of a persistent event:** The sensor can be configured to send repeated alarms if the temperature or humidity exceeds a threshold.
- **Error or Default management:** The sensor can be configured to send alerts if there is a hardware error or if the configuration is inconsistent.
- **Timestamp of the frame (LoRaWAN):** The sensor can be configured to include a timestamp in the data frame, which can be used to track the time and date of the measurements.
- **Network Quality Test at start-up (LoRaWAN):** The sensor can be configured to perform a network quality test at startup, which can help to ensure that the sensor is connected to a reliable network.

The Adeunis Comfort is a well-designed and reliable sensor that is a good choice for a variety of applications where temperature and humidity monitoring is required. This sensor can be

used in different applications such as building automation, home automation, agriculture and industrial IoT.

Here are some of the pros and cons of the Adeunis Contact sensor:

Pros:

- **Versatility:** Suitable for a variety of applications, from building automation to agriculture and industrial IoT.
- **Long Battery Life:** Offers up to 10 years of monitoring without frequent battery changes.
- **Additional Features:** Includes data logging, timestamping, and network quality tests for enhanced functionality.

Cons:

- **Indoor Use Only:** Designed for indoor use, limiting outdoor or harsh environment applications.
- **Limited Temperature Range:** Covers -20°C to 60°C, may not suit extreme conditions.

A Comfort temperature sensor can be seen in figure 12.



Figure 12: Adeunis Comfort sensor [14].

3.1.2 Adeunis Temperature sensor

The Adeunis Temp is a LoRaWAN-enabled sensor that measures temperature. It is available in two versions [15]:

- **TEMP:** 1 ambient + 1 remote probe (this one is used in this project)
- **TEMP2S:** 2 remote probes

The sensor can be used to monitor the temperature of a variety of objects and environments, such as:

- The temperature of the domestic hot water at the start and end of the circuit.
- The temperature of a storage area for sensitive products.
- The temperature of the road surface optimizes the triggering of winter services.

The sensor has a temperature range of -25°C to +70°C and a precision of +/-0.2°C for ambient temperatures between 0°C and 60°C. The remote temperature probe has a precision of +/-0.2°C for temperatures between 0°C and 60°C.

The sensor has a battery life of up to 10 years, depending on the frequency of data transmission. It can be configured to send data periodically or when thresholds are exceeded.

The sensor also has several additional features, including:

- **Autonomy optimization:** The sensor can be configured to store data for a period before sending it to the network, which can help to extend the battery life.
- **Accessibility of the data:** The sensor can be configured to send data to multiple gateways, which can help to ensure that the data is always available.
- **Alarm repetition in case of a persistent event:** The sensor can be configured to send repeated alarms if the temperature exceeds a threshold.
- **Error or Default management:** The sensor can be configured to send alerts if there is a hardware error or if the configuration is inconsistent.
- **Timestamp of the frame (LoRaWAN):** The sensor can be configured to include a timestamp in the data frame, which can be used to track the time and date of the measurements.
- **Network Quality Test at start-up (LoRaWAN):** The sensor can be configured to perform a network quality test at startup, which can help to ensure that the sensor is connected to a reliable network.

Pros:

- **Versatility:** Suitable for monitoring temperature in various environments, including outdoors.
- **Precision:** +/-0.2°C accuracy for ambient temperatures between 0°C and 60°C.
- **Long Battery Life:** Up to 10 years.
- **Additional Features:** Autonomy optimization, data accessibility, redundancy, and more.
- **IP68 Rating:** Suitable for outdoor use in harsh environments.
- **Configuration Flexibility:** Local and remote configuration options.

Cons:

- **Limited Temperature Range:** Covers -30°C to 150°C.

Figure 13 shows an Adeunis temperature sensor.



Figure 13: Adeunis Temperature sensor [15].

3.1.3 Adeunis Contact Sensor

The Adeunis Contact sensor is a LoRaWAN-enabled sensor that detects openings and closures. It is designed for indoor use and has an IP67 rating, making it dustproof and water resistant [16].

The sensor can be used to monitor a variety of objects and environments, such as:

- The doors and windows of a building to detect intrusions.
- The refrigerator door to know how long it has been open.
- The hatch of a smoke extraction system to prevent it from being opened too often.

The sensor has a battery life of up to 10 years, depending on the frequency of data transmission. It can be configured to send data periodically or when thresholds are exceeded.

Like the other Adeunis sensors, this sensor has some capabilities like Date time stamping, network quality test and error/fault management.

The disadvantage of this sensor could be that there is no built-in display, The sensor does not have a built-in display, so you will need to use a gateway or other device to view the sensor data. A contact sensor is shown in Figure 14.



Figure 14: Adeunis Contact sensor [16].

3.2 Field Test Device

The Adeunis FTD is a LoRaWAN-enabled field test device that can be used to test and troubleshoot LoRaWAN networks [17].

The FTD can be used to perform a variety of tests, such as:

- Signal strength and quality measurements
- Network coverage analysis
- Channel quality analysis
- ADR (Adaptive Data Rate) optimization
- Packet loss analysis
- Network latency analysis

The FTD also has a number of additional features, including:

- Easy configuration and firmware updates
- Support for multiple LoRaWAN regions
- Support for OTAA and ABP

In figure 15 a field test device can be observed.



Figure 15: Adeunis Field Test device [17].

3.3 Gateway

A gateway in the context of networking and the Internet of Things (IoT) is a hardware device or software program that serves as an intermediary connecting two different networks. It acts as a bridge, facilitating the flow of data between devices or sensors in one network to another. In the case of LoRaWAN and IoT deployments, a gateway plays a crucial role by receiving data from sensors and transmitting it to a central server or the broader internet, enabling communication between low-power, wide-area networks (LPWANs) and the larger network infrastructure. Essentially, a gateway facilitates the exchange of information between devices in distinct networks, ensuring seamless connectivity and data transfer.

The installation of a gateway has been considered because the measured LoRaWAN network signal in the room where the dry contact sensor and indoor temperature sensor are located is low as the Field test device indicates. With the objective to improve the LoRaWAN network signal in the project group room in the C building of the USN Porsgrunn campus. The used gateway has been located near the sensors. It supports class A and C devices. In Figure 16 the used gateway for this project can be seen. Upon conducting a field test using a network coverage assessment device, it was observed that the signal strength at the installation location was insufficient. Consequently, a decision was made to implement the use of a gateway.



Figure 16: Installed gateway in the university.

3.4 Hardware installation process

In this section the installation process of the used hardware will be explained.

3.4.1 Comfort Sensor installation

To install the Adeunis Comfort Sensor, a suitable location should first be chosen. The sensor should be placed in a central location within the room or space that is supposed to be monitored. It is important to place the sensor away from direct sunlight, drafts, and heat sources. Once a suitable location has been chosen, the sensor can be mounted on a wall or ceiling using the included screws or other types. A double-sided tape is used to install the sensor in room C229-A. Figure 17 shows the installed sensor in mentioned location.

To use the sensor in LoRaWAN network, an account with a network provider should be created and the sensor should be registered. Once the sensor is registered, it can be configured using the network provider's online portal. Here Altibox ThingPark portal has been used.



Figure 17: Comfort sensor installed on the project room.

3.4.2 Contact Sensor installation

Before installing the Adeunis Door Contact Sensor, it is important to identify a suitable location for the sensor. The sensor should be placed on the door frame, with the magnet attached to the door.

The Adeunis Door Contact Sensor is installed by attaching the magnet to the door in such a way that it is aligned with the sensor when the door is closed. The sensor is then mounted on the door frame using the double-sided tape. Figure 18 shows the installed sensor.



Figure 18: Contact sensor installed on the project room.

3.4.3 Field Test Device installation

The field test device does not need any installation as the main function of the device is to check the network signal to see if the place where the device is being used is a good place to install the LoRaWAN sensors. During the project a network analysis has been done to see conditions of different places in Campus Porsgrunn.

3.4.4 Temperature Sensor installation

Prior to installation, the sensor's optimal location was identified through a site survey. Factors such as exposure to direct sunlight, protection from extreme weather conditions, and accessibility for maintenance were considered.

The Adeunis Outdoor Temperature Sensor was installed by mounting it on the wall at a height of 2 meters above ground level. The sensor was oriented in a way that the temperature sensor

was exposed to the open air without any obstructions, and it was securely fastened to the pole using appropriate brackets and clamps. Figure 19 shows the installed sensor.



Figure 19: Adeunis Temperature sensor installed on USN campus.

3.5 Parameters and Identifiers for Sensors

For making a good connection and configuration between the sensor and the IoT platform some parameters are needed. In this section these parameters are described.

- DevEUI

DevEUI, short for "Device EUI" or "Device Extended Unique Identifier," is a globally unique 64-bit identifier assigned to an IoT device, such as a sensor or a node, in a LoRaWAN network. It serves as a fundamental identifier for the device within the LoRaWAN ecosystem, ensuring its uniqueness across the network. DevEUI is essential for secure device registration, authentication, and communication with the LoRaWAN network server, enabling proper data routing and management in IoT applications.

- AppEUI

AppEUI, or "Application EUI," is a 64-bit globally unique identifier used in LoRaWAN IoT networks. Unlike the DevEUI, which uniquely identifies the IoT device itself, the AppEUI is associated with the application or service that communicates with the device. It helps in the identification and secure communication between a specific device and its corresponding application server in the LoRaWAN network. The AppEUI is essential for proper data routing, ensuring that data from the device reaches the correct application or service. This separation of identifiers enhances security and scalability in LoRaWAN IoT deployments.

- AppKey

An AppKey, short for "Application Key," is a cryptographic key used in LoRaWAN IoT networks to ensure secure communication between an IoT device and its associated application server. AppKeys are crucial for encrypting and decrypting data transmitted between the device and the server, ensuring the confidentiality and integrity of the information exchanged. Each device typically has a unique AppKey that is known only to the device and the application server it communicates with. AppKeys play a pivotal role in IoT security by preventing

unauthorized access to device data and ensuring that messages are only accessible to the intended recipient, enhancing the overall security of the LoRaWAN IoT ecosystem.

- PER

"Packet Error Rate," is a crucial metric in wireless communication systems, including IoT networks. It measures the percentage of transmitted data packets that are received with errors or not received at all. A low PER indicates a high level of data packet reliability, while a high PER suggests that a significant portion of data packets is lost or corrupted during transmission. Monitoring and minimizing PER is essential in IoT applications to ensure the accuracy and integrity of data being sent from sensors or devices to the network. Achieving a low PER is particularly important in critical IoT deployments, such as smart cities, industrial automation, and healthcare, where data accuracy and reliability are paramount for decision-making and safety.

- SNR

"Signal-to-Noise Ratio," is a fundamental parameter in wireless communication systems, including IoT networks. It quantifies the ratio of the strength of the desired signal to the level of background noise and interference in the communication channel. A higher SNR indicates a stronger, more reliable signal with less interference, while a lower SNR signifies a weaker, potentially less reliable signal submerged in noise. In IoT applications, maintaining a favorable SNR is crucial for ensuring accurate and error-free data transmission. A high SNR leads to better communication reliability and lower packet loss rates, resulting in more robust and dependable IoT networks. Achieving a good SNR is vital in various IoT scenarios, including remote monitoring, asset tracking, and smart sensor deployments, where data integrity is essential for informed decision-making and efficient operations.

3.6 LoRa/Sigfox IoT Configurator application

The IoT Configurator app by Adeunis is a mobile app that allows customers to configure and manage Adeunis IoT devices. It is a free app that is available for both Android and iOS devices.

The IoT Configurator app supports the following Adeunis IoT devices:

- Sensors, such as the LoRaWAN temperature and humidity sensor, LoRaWAN CO2 sensor, and LoRaWAN air quality sensor
- Actuators, such as the LoRaWAN relay and LoRaWAN dimmer
- Gateways, such as the LoRaWAN gateway

The IoT Configurator app can be used to configure the following settings for Adeunis IoT devices:

- Device name and description
- Network settings
- Sensor settings
- Actuator settings

- Gateway settings

Figure 20 shows some parts of graphic user interface in this application.

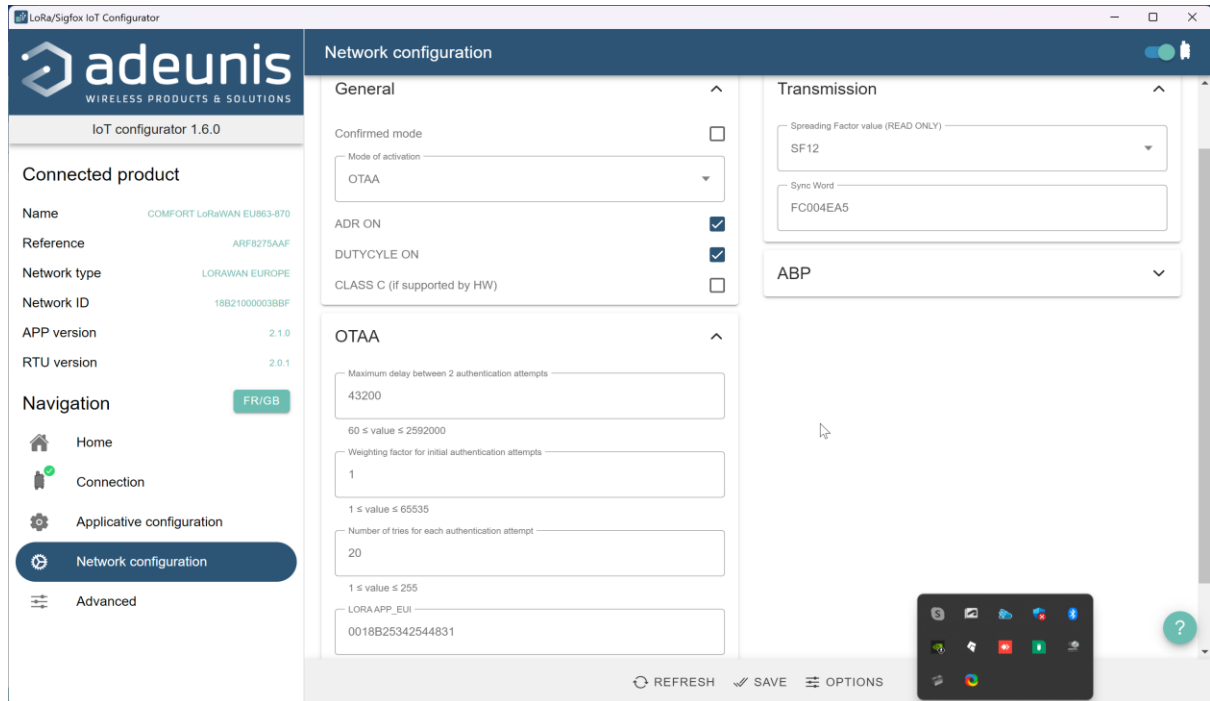


Figure 20: A sample of GUI of LoRa/Sigfox IoT configurator.

4 Altibox LoRaWAN Infrastructure

As explained in previous chapters for the LoRaWAN sensor installation, Altibox company infrastructure has been used in order to configure and install the Adeunis sensors at the Porsgrunn campus. In this chapter the used infrastructure will be explained. ThingPark portal and ThingPark X platforms will be described.

4.1 Altibox Thingpark portal

Altibox ThingPark portal is the main portal for the project to access and configure the Adeunis sensors to the LoRaWAN network. This portal has been used in order to connect the sensors and see that they are working in a correct way. In Figure 21 the main page of the portal can be seen, where it is possible to choose different platforms.

The Altibox ThingsPark platform provides a wide range of functionalities for efficient device management, allowing users to oversee packages transmitted by devices, access network-related information, and gain insights into various device-related metrics and details.

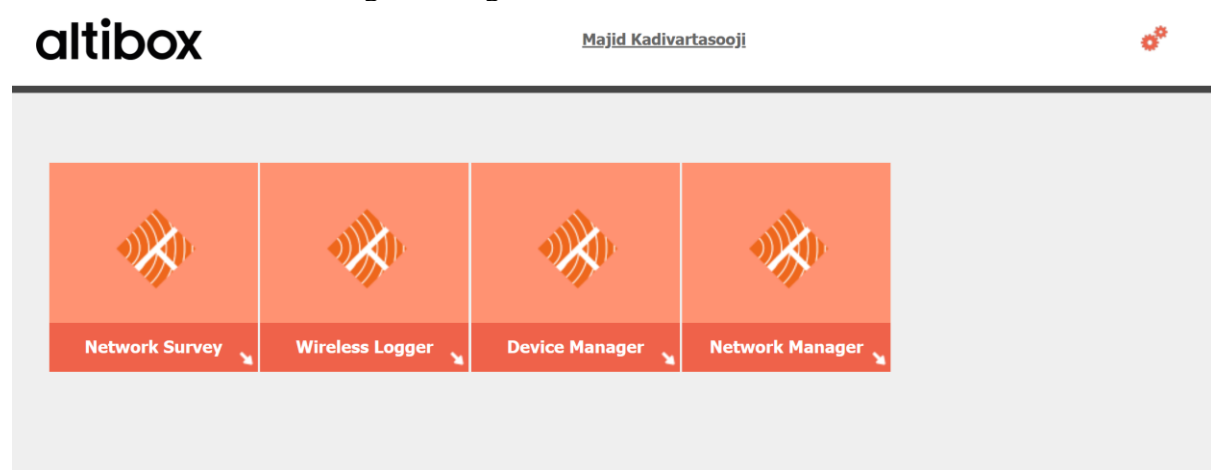


Figure 21: Altibox ThingsPark platform.

4.1.1 Network Survey

The Network Survey feature offers a comprehensive view of active devices within a specified area, including valuable information about the properties of their signals. Additionally, when an IoT device provides its own location, the survey can accurately depict the geographical distribution and signal characteristics of the devices operating in that area, allowing for precise location-based insights. Figure 22 shows the network survey page.

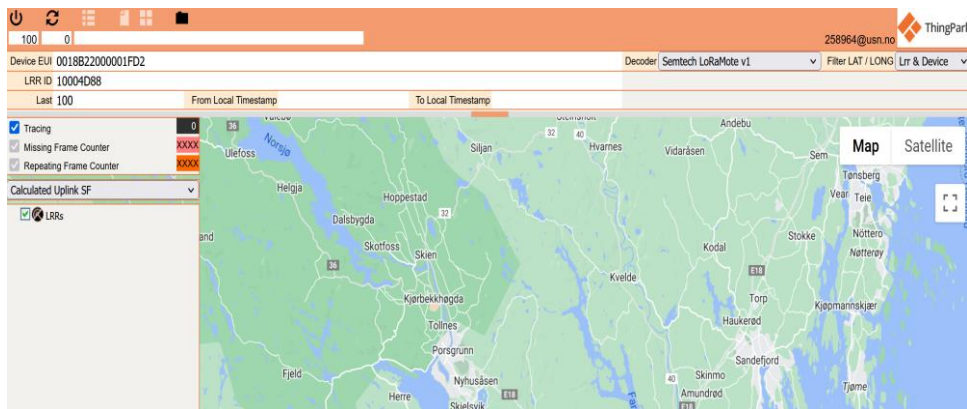


Figure 22: Network survey.

4.1.2 Wireless logger

The Wireless logger is a valuable tool that provides a comprehensive overview of the network by capturing and analyzing the content of packets transmitted by all available devices. It offers detailed insights into the data traffic and communication patterns within the network, enabling a deeper understanding of device interactions and network performance. Figure 23 depicts the packages and raw data sent by already installed sensor for this project.

WIRELESS-LOGGER Last Update: 2023-10-28 12:55:23

Dashboard [100050938]

DevAddr Filtering: [] Clear DevEUI Filtering: [] Clear LRR Id Filtering: [] Clear LRR Filtering: [] Clear AS ID Filtering: [] Clear

From: [] To: [] Packet Type: [] Clear

Decoder: No decoding

Auto Reload: no Expand All: [] Refresh Export size: 100 Export Map

		UTC Timestamp	Local Timestamp	DevAddr	DevEUI	FPort	FCnt	NFCnt	AFcnt	RSSI	SNR	ESP	SF/DR	SubBand	Channel
mac data	2023-10-28 10:49:01.167	2023-10-28 12:49:01.167	34C194C1	0018B21000004540	1	5910				-101.0	5.25	-102.13...	SF7	G1	LC3
mac	2023-10-28 10:42:21.933	2023-10-28 12:42:21.933	FC004EA5	0018B21000003BBF	0		233						SF9	G2	LC7
mac data	2023-10-28 10:42:20.933	2023-10-28 12:42:20.933	FC004EA5	0018B21000003BBF	1	239				-110.0	6.75	-110.83...	SF9	G2	LC7
mac	2023-10-28 10:39:02.076	2023-10-28 12:39:02.076	34C194C1	0018B21000004540	0		1008						SF9	G2	LC7
data	2023-10-28 10:39:01.076	2023-10-28 12:39:01.076	34C194C1	0018B21000004540	1	5909				-99.0	6.0	-99.973...	SF7	G2	LC7
data	2023-10-28 10:29:01.140	2023-10-28 12:29:01.140	34C194C1	0018B21000004540	1	5908				-98.0	4.5	-99.318...	SF7	G2	LC6
mac data	2023-10-28 10:26:07.100	2023-10-28 12:26:07.100	FC004E1F	0018B22000001FD2	1	1745				-39.0	10.0	-39.413...	SF7	G1	LC3
data	2023-10-28 10:19:01.092	2023-10-28 12:19:01.092	34C194C1	0018B21000004540	1	5907				-99.0	6.0	-99.973...	SF7	G1	LC1
data	2023-10-28 10:09:00.776	2023-10-28 12:09:00.776	34C194C1	0018B21000004540	1	5906				-100.0	6.5	-100.87...	SF7	G2	LC8

Figure 23: Wireless logger shows the package and data sent by sensors.

4.1.3 Device Manager

The Device Manager offers a range of functionalities designed to streamline device management within the network. These capabilities include the ability to register new devices, edit existing device profiles, and remove devices as needed. It also allows for the detailed specification of each device, ensuring that the network is efficiently configured and organized. Figure 23 shows the device manager and the installed sensor in this project.

The device manager is in charge of monitoring and checking the correct functionality of the connected devices to the LoRaWAN network. In the figure 24 connected devices can be observed as shown in the device manager. The portal gives information about the states of the connected devices like the average packets is the device sending per day, the mean Packet Error Rate, the average Signal-to-Noise Ratio, the battery of the device, alarm information and is able to give the location where the device is installed.

Name / Type	Identifiers	Connectivity	Average packets	Mean PER	Average SNR	Battery	Alarm	Locate
Comfort Sensor	0018B210000038BF FC004EAS	ALTIBOX Standard XL Pipedream	46.0/day	0.0%	1.0 dB		2	
Contact Sensor Dry Contact Sensor	0018B22000001FD2 FC004E1F	ALTIBOX Standard XL Pipedream	44.0/day	0.0%	10.0 dB		4	
Field Test Device	0018B2000000263A8 FC004FBA	ALTIBOX Standard XL Pipedream	0.0/day	0.0%	6.8 dB		1	
Outdoor Temperature Sensor	0018B21000004540 34C194C1	ALTIBOX Standard XL Pipedream	160.0/day	2.0%	5.7 dB		3	

Figure 24: Device Manager is a perfect tool to manage IoT devices.

4.1.3.1 Adding Devices to Device Manager

To add a device to the device manager ThingsPark portal has been used. First device name and small description needs to be added. In addition, device location can be added. LoRaWAN is the used connectivity for connecting the device as it can be seen in the Figure 25.

Connectivity: LoRaWAN

Administrative data

Device name:

Marker: * Change marker

Administrative info:

Administrative location: * Network location Change location

Motion indicator: Device profile settings

Figure 25: Description to add a device in ThingsPark portal.

After setting the administrative data of the device, the device has to be identified and for that device identification details are added. During this project Adeunis sensors are used and the device identification settings has been given from Altibox company for adding the devices in a correct way to their ThingPark portal.

In this section many identification settings are asked like device manufacture and model, the activation method. During the project Over The Air Activation OTAA has been used. Join server also is required as well as devEUI, joinEUI and the Appkey. Is really important to put the correct identification information in order to connect the devices in the Figure 26.

Device identification	
Manufacturer: *	Adeunis
Model: *	<Empty>
Device activation:	Over The Air Activation (OTAA)
Join server: *	Local Join server with software encryption
DevEUI: *	AC-DE-48-23-45-67-AB-CD
JoinEUI (AppEUI):	AC-DE-48-23-45-67-AB-CD
Key format:	Clear text
AppKey: *	BE-C4-99-C6-9E-9C-93-9E-41-3B-66-39-61-63-6C-61

Figure 26:Device identification in ThingsPark portal.

To end adding the device to the ThingPark portal Network connectivity plan is required. Altibox Standard XL connectivity plan provided from Altibox has been used. Application server routing profile also is required and can be chosed when adding the device, Figure 27.

Network parameters	
Connectivity plan:	ALTIBOX Connectivity Supplier / ALTIBOX Standard XL (6)
DevAddr: *	Allocated by the network server

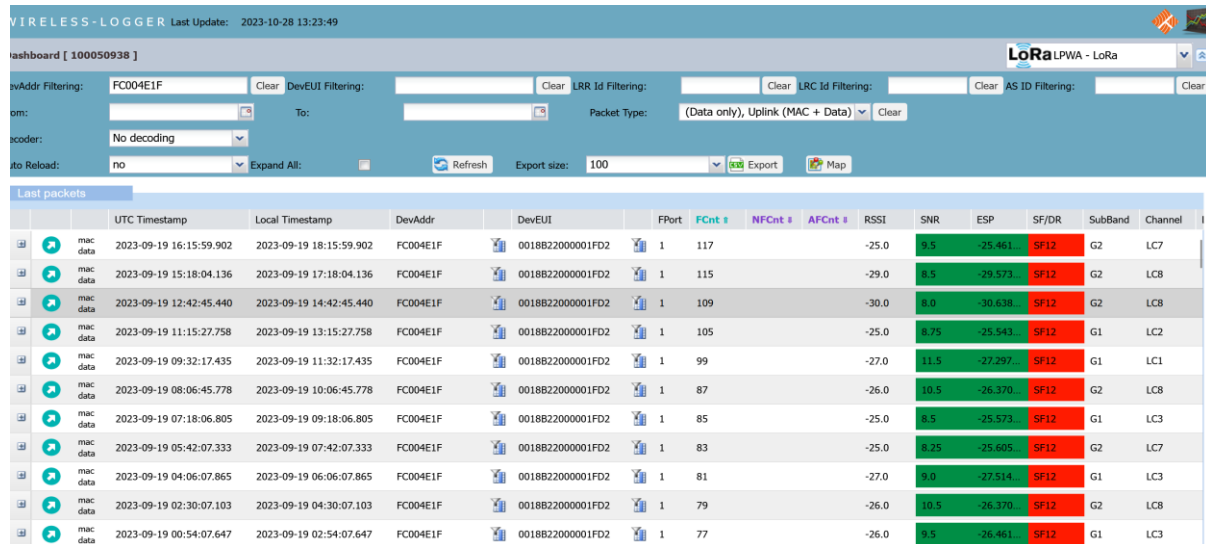
Application layer handling	
Application server routing profile:	TPX

Figure 27:Network parameters and Application layer in ThingsPark.

4.1.3.2 Sensor Troubleshooting

While installing this sensor, it was observed that the device encountered difficulties in establishing a connection with the gateway. The device sent join requests to join the network,

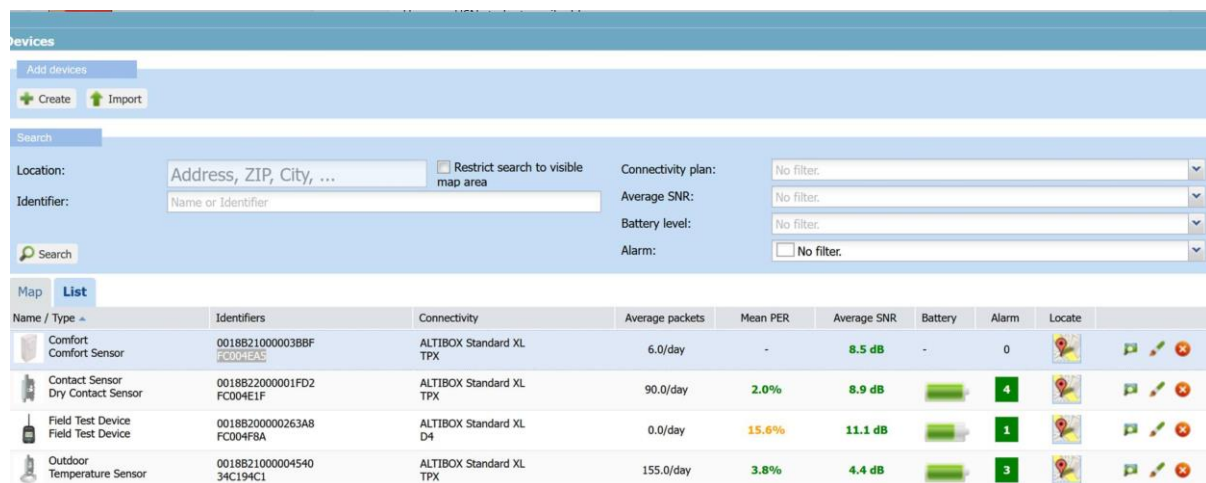
but unfortunately, it was unable to successfully connect to the network. Figure 28 shows the requests sent by the sensor.



		UTC Timestamp	Local Timestamp	DevAddr	DevEUI	FPort	FCnt	NFCnt	AFCnt	RSSI	SNR	ESP	SF/DR	SubBand	Channel
mac data	2023-09-19 16:15:59.902	2023-09-19 18:15:59.902	FC004E1F	0018B22000001FD2	1	117				-25.0	8.5	-25.461	SF12	G2	LC7
mac data	2023-09-19 15:18:04.136	2023-09-19 17:18:04.136	FC004E1F	0018B22000001FD2	1	115				-29.0	8.5	-29.573	SF12	G2	LC8
mac data	2023-09-19 12:42:45.440	2023-09-19 14:42:45.440	FC004E1F	0018B22000001FD2	1	109				-30.0	8.0	-30.638	SF12	G2	LC8
mac data	2023-09-19 11:15:27.758	2023-09-19 13:15:27.758	FC004E1F	0018B22000001FD2	1	105				-25.0	8.75	-25.543	SF12	G1	LC2
mac data	2023-09-19 09:32:17.435	2023-09-19 11:32:17.435	FC004E1F	0018B22000001FD2	1	99				-27.0	11.5	-27.297	SF12	G1	LC1
mac data	2023-09-19 08:06:45.778	2023-09-19 10:06:45.778	FC004E1F	0018B22000001FD2	1	87				-26.0	10.5	-26.370	SF12	G2	LC8
mac data	2023-09-19 07:18:06.805	2023-09-19 09:18:06.805	FC004E1F	0018B22000001FD2	1	85				-25.0	8.5	-25.573	SF12	G1	LC3
mac data	2023-09-19 05:42:07.333	2023-09-19 07:42:07.333	FC004E1F	0018B22000001FD2	1	83				-25.0	8.25	-25.605	SF12	G2	LC7
mac data	2023-09-19 04:06:07.865	2023-09-19 06:06:07.865	FC004E1F	0018B22000001FD2	1	81				-27.0	9.0	-27.514	SF12	G1	LC3
mac data	2023-09-19 02:30:07.103	2023-09-19 04:30:07.103	FC004E1F	0018B22000001FD2	1	79				-26.0	10.5	-26.370	SF12	G2	LC8
mac data	2023-09-19 00:54:07.647	2023-09-19 02:54:07.647	FC004E1F	0018B22000001FD2	1	77				-26.0	8.5	-26.461	SF12	G1	LC3

Figure 28: Wireless manager shows the requests sent by sensor.

In Figure 29, it is evident that despite having an average Signal-to-Noise Ratio (SNR) for the sensor, the Packet Error Rate (PER) is unknown. This indicates that the sensor is not connected to the network, suggesting potential issues with installation and configuration.



Name / Type	Identifiers	Connectivity	Average packets	Mean PER	Average SNR	Battery	Alarm	Locate	
Comfort Sensor	0018B21000038BF FC004E1F	ALTIBOX Standard XL TPX	6.0/day	-	8.5 dB	-	0		
Contact Sensor	0018B22000001FD2 FC004E1F	ALTIBOX Standard XL TPX	90.0/day	2.0%	8.9 dB		4		
Field Test Device	0018B200000263A8 FC004F8A	ALTIBOX Standard XL D4	0.0/day	15.6%	11.1 dB		1		
Outdoor Temperature Sensor	0018B21000004540 34C194C1	ALTIBOX Standard XL TPX	155.0/day	3.8%	4.4 dB		3		

Figure 29: Device manager shows that the sensor is installed but its not a part of the network.

After relocating the sensor, it has been observed that the sensor is functioning correctly without the need for a gateway. the reason has been investigated and the most probable reason could be the Incorrect configuration. Both the IoT device and the gateway need to be configured correctly in order to work together. If either device is not configured correctly, they will not be able to communicate. The Figure 30 displays the devices that are connected to the network via the gateway.

SERVED DEVICES 1

▼

↺

↻

📄

List

Map

1-2 of 2

Show: 100

Name	DevEUI	DevAddr	Lat	Long	Last Uplink <div>⬇</div>	SNR	ESP	Best gateway
Outdoor	00-18-B2-10-00-00-45-40	34-C1-94-C1			Today 13:39:01	7.25 dB	-101.75 dBm	This base station
Contact Sensor	00-18-B2-20-00-00-1F-D2	FC-00-4E-1F			Today 13:14:06	10.25 dB	-37.39 dBm	This base station

⏮

<

1

>

⏭

Figure 30: devices are connected to LoRaWAN using gateway.

4.1.4 Network Manager

The Network Manager provides a comprehensive overview of all the registered Gateways within the network. This feature allows users to easily access information and insights about these Gateways, facilitating efficient monitoring and management of these critical network components. Figure shows a gateway used for this project in Network manager tool. Figure 31 shows the gateway used for this project in the network manager portal.

altibox

Dashboard

Base Stations

Alarms

Administration

Base stations

+ ADD BASE STATION


List

Map

1-1 of 1

Add filter

Show: 100

		Name	LRR-ID	Last Uplink	Packets (1h)	Alarms	Tags	
		aib-entgw5-ufi-test	10-00-4D-88	Today 13:09:01	4			...

<

1

>

Figure 31: the gateway is used for the project.

5 Receiving data from Sensors using Thingpark X and Dimension4

5.1 ThingPark X

ThingPark X is a cloud-based platform for managing and automating the flow of data between IoT devices, applications, and services. It provides a visual editor for creating and managing flows, as well as a set of APIs for automating tasks, as it can be seen in Figure 32.

Flows can be used to perform a wide range of tasks, such as:

- Collecting and storing data from IoT devices
- Transforming data into a usable format
- Routing data to different applications and services
- Performing data analysis
- Sending alerts and notifications
- Automating tasks

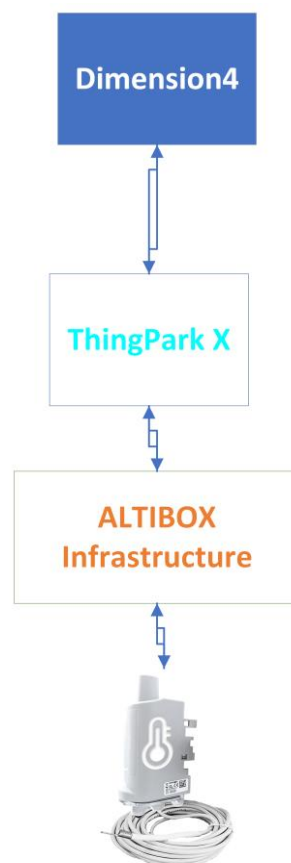


Figure 32: Dataflow structure from LoRaWAN sensor to Dimension 4.

Altibox provided connectivity plan has been used in order to connect the four devices as shown previously. ALTIBOX Connectivity Supplier / ALTIBOX Standard XL (6) is the used connectivity plan. This connectivity plan is LoRoWAN connectivity plan with unicast communication type.

In the development of the LoRaWAN sensor project, the integration of ThingPark X IoT-Flow within the Altibox infrastructure serves to learn more about how this kind of sensors work. This section will explain the inner workings of ThingPark X IoT-Flow, shedding light on its role in getting data from sensors, bidirectional communication between ThingPark-powered networks like ThingPark portal and an array of external application servers or cloud-based IoT services.

ThingPark X can be defined as a mediation layer between ThingPark-powered networks and diverse application servers or cloud-based IoT services . In the figure Figure 33 the overview of the data flow in ThingPark X can be observed.

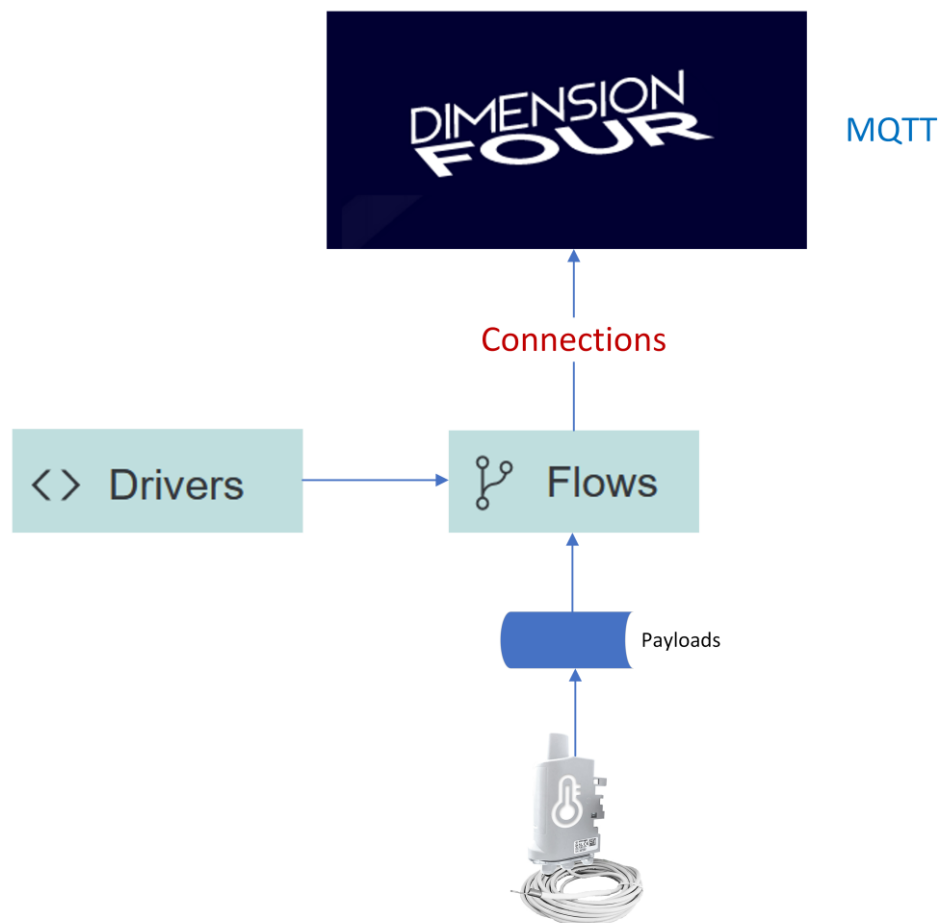


Figure 33: ThingPark X, overview of the data flow.

The critical capabilities that ThingPark X brings include:

5.1.1 Drivers

In the ThingPark X platform, drivers play a pivotal role in facilitating the exchange of messages between devices and the cloud. Their primary function is to transform the raw data transmitted by devices into a format that can be comprehended by the cloud and vice versa. These drivers are the linchpin for seamless communication within the platform, as they ensure that data sent by devices is intelligible to the cloud and that commands from the cloud can be interpreted by the devices. Essentially, they act as intermediaries, bridging the gap between the diverse data formats used by devices and the cloud infrastructure.

The versatility of drivers is a key asset for the ThingPark X platform, enabling it to connect with a diverse array of devices. While the platform comes equipped with a library of pre-built drivers for many commonly used devices, it also offers the flexibility to develop custom drivers. These drivers serve a multitude of purposes, including decoding messages from devices into standardized formats like JSON or XML, encoding cloud commands into device-compatible formats, processing device data by filtering, aggregating, or transforming it, and even generating commands for devices based on cloud-received data.

Figure 34 shows some of the drivers used in this project.

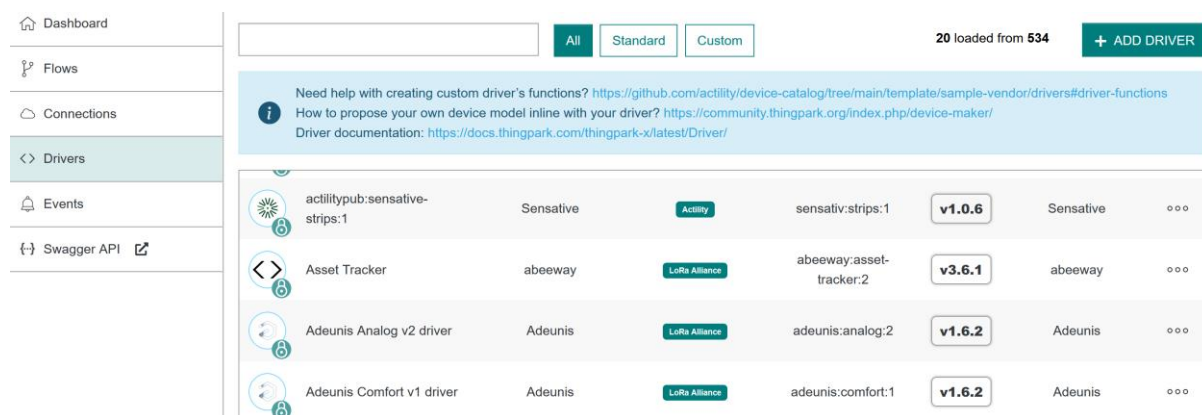


Figure 34: some of the drivers used for this project.

5.1.2 Connections

Within the ThingPark X, connections are the conduits that define the channels of communication between devices and the cloud infrastructure. They serve as the bedrock, specifying essential elements like the communication protocol, authentication method, and security parameters, ensuring a seamless and secure exchange of data.

The ThingPark X platform relies on connections to establish robust links with a diverse array of devices. It offers a diverse selection of preconfigured connections tailored to widely used protocols, yet it also allows for the creation of custom connections to meet specific needs. Connections fulfill a multitude of roles, beginning with the definition of the communication protocol used to engage with devices, which can span from standardized protocols like MQTT and CoAP to proprietary options. Additionally, they detail the chosen authentication method. Moreover, connections configure the security settings, incorporating encryption and certificate settings, providing a safeguard for the data exchange.

For this project, connections utilizing the MQTT protocol have been established to facilitate the transfer of data from LoRaWAN sensors to the DimensionFour platform.

The figure displays some of the connections that have been established for this project. Within the figure, you can also view the settings of a specific connection.

A custom JSLT transformation is employed for the aforementioned connection to convert data into the format expected by DimensionFour. In the figure, you can observe the JSLT code used for the temperature sensor.

JSLT (JavaScript for Linked Data) is a query and transformation language for JSON data. JSLT is a powerful tool that can be used to extract, filter, transform, and generate JSON data.

JSLT is a declarative language, which means that you describe what you want to do with the JSON data, rather than how to do it.

The screenshot shows a web interface for managing connections. On the left is a sidebar with navigation links: Dashboard, Flows, Connections (highlighted), Drivers, Events, and Swagger API. The main area features a table of connections with a search filter and an 'ADD CONNECTION' button. The table columns are: Connection Name, Last Restart, Active Devices (last 1h/24h), Uplinks (last 1h/24h), Downlinks (last 1h/24h), State, and an actions column with a dropdown and a delete icon. All connections are of type MQTT and have a state of 'OPENED'.

	Connection Name	Last Restart	Active Devices (last 1h/24h)	Uplinks (last 1h/24h)	Downlinks (last 1h/24h)	State	
MQTT	Project2023-USN	2 days ago	1 / 1	5 / 142	0 / 0	OPENED	
MQTT	MQTT_D4_MobileApp_TempSensor	2 days ago	1 / 1	5 / 142	0 / 0	OPENED	
MQTT	MQTT_D4_MobileApp_DoorSensor	a day ago	1 / 1	1 / 52	0 / 0	OPENED	
MQTT	MQTT_D4_MobileApp_ComfortSensor	2 days ago	1 / 1	1 / 24	0 / 0	OPENED	
MQTT	MQTT_D4_Comfort_USN000	2 days ago	1 / 1	1 / 24	0 / 0	OPENED	
MQTT	MQTT_D4_Temperature_USN000	2 days ago	1 / 1	5 / 142	0 / 0	OPENED	

Figure 35: TPX flow overview picture from documentation.

1
active devices

5
uplinks
Success

0
downlinks

BASIC SETTINGS (Connection id: 4963)

Hostname ⓘ <input type="text" value="mqtt.dimensionfour.io:1883"/>	Protocol ⓘ <input type="text" value="TCP"/>
MQTT Username ⓘ <input type="text" value="d4-mqtt-majid"/>	MQTT Password ⓘ <input type="password" value="....."/>
Published topic pattern ⓘ <input type="text" value="POINT/PUSH"/>	Subscribed topic pattern ⓘ <input type="text" value="mqtt/things/{DevEUI}/downlink"/>

Figure 36: The setting of a connection to mqtt.

```

1 {
2   "pointId": "652a6f06e5daa3768b133b5c",
3   "tenantId": "majid",
4   "tenantKey": "80a15aec5de5fdbcd3e8ff09",
5   "signals": [
6     {
7       "value": string(.DevEUI_uplink.payload),
8       "unit": "CELSIUS_DEGREES",
9       "type": "Temperature",
10      "timestamp": .DevEUI_uplink.Time,
11      "metadata": {
12        "DevEUI" : .DevEUI_uplink.DevEUI,
13        "DevAddr" : .DevEUI_uplink.DevAddr,
14        "payload_hex" : .DevEUI_uplink.payload_hex,
15        "frequency" : .DevEUI_uplink.Frequency,
16        "RSSI" : .DevEUI_uplink.LrrRSSI.

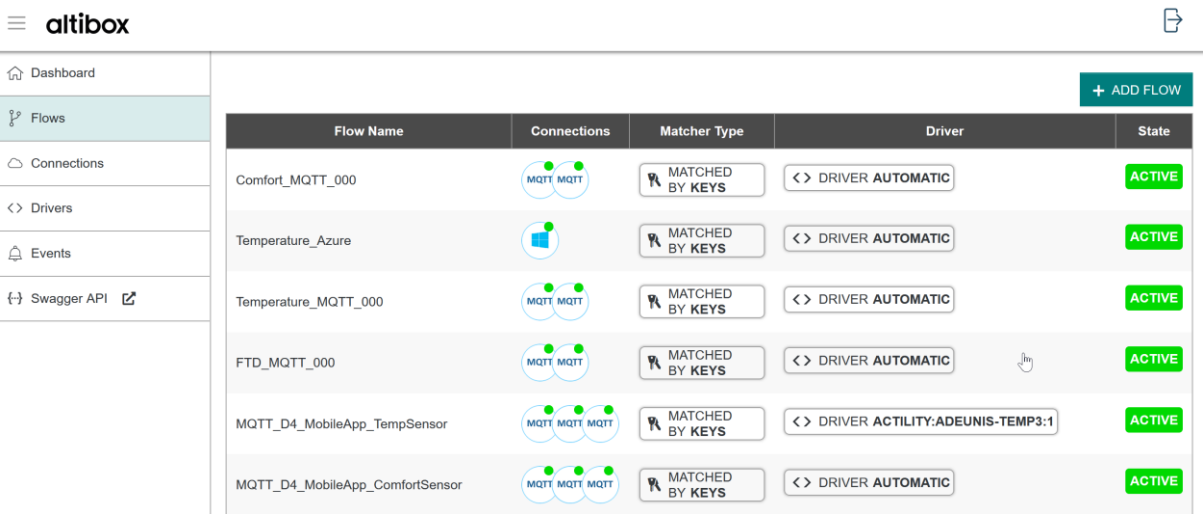
```

Figure 37: LST code to convert json object.

5.1.3 Flows

Within the framework of ThingPark X, Flows constitute a versatile tool that simplifies the process of creating and managing data pipelines connecting devices, the cloud, and various

systems. Notably, it ensures accessibility to users regardless of their programming skills, thanks to its user-friendly visual editor. Flows offer a wide range of essential functionalities, including data aggregation, transformation, forwarding, integration, and task automation. Data aggregation involves efficiently collecting information from connected devices for applications such as real-time monitoring, asset tracking, and operational optimization. As proficient data transformers, Flows refine and enhance data before transmitting it, encompassing actions like filtering and enrichment. The ability to forward data to the cloud is pivotal for data analysis and visualization. Figure 38 depicts some of the flows in this project.



Flow Name	Connections	Matcher Type	Driver	State
Comfort_MQTT_000		MATCHED BY KEYS	<> DRIVER AUTOMATIC	ACTIVE
Temperature_Azure		MATCHED BY KEYS	<> DRIVER AUTOMATIC	ACTIVE
Temperature_MQTT_000		MATCHED BY KEYS	<> DRIVER AUTOMATIC	ACTIVE
FTD_MQTT_000		MATCHED BY KEYS	<> DRIVER AUTOMATIC	ACTIVE
MQTT_D4_MobileApp_TempSensor		MATCHED BY KEYS	<> DRIVER ACTIVITY:ADEUNIS-TEMP3:1	ACTIVE
MQTT_D4_MobileApp_ComfortSensor		MATCHED BY KEYS	<> DRIVER AUTOMATIC	ACTIVE

Figure 38: Flows have been defined in the project.

5.2 Dimension4

Dimension4 is a Norwegian technology company that provides a cloud-based IoT platform. The platform enables users to connect, manage, and analyze data from IoT devices. It is designed to be scalable and flexible and can be used by businesses of all sizes to develop and deploy IoT applications.

Dimension Four's IoT platform is built on top of GraphQL, a query language that allows users to request specific data from an API. This makes the platform easy to use and integrate with other systems.

Dimension Four's IoT platform includes a number of features, such as:

- **Device management:** Dimension Four allows users to connect and manage a wide range of IoT devices, including sensors, actuators, and controllers.
- **Data management:** Dimension Four collects and stores data from IoT devices, and provides a variety of tools for analyzing and visualizing the data.
- **Application development:** Dimension Four provides a set of APIs and tools for developing IoT applications, such as mobile apps, web apps, and enterprise applications.

Figure 39 shows the role of dimension4 in a schematic view.

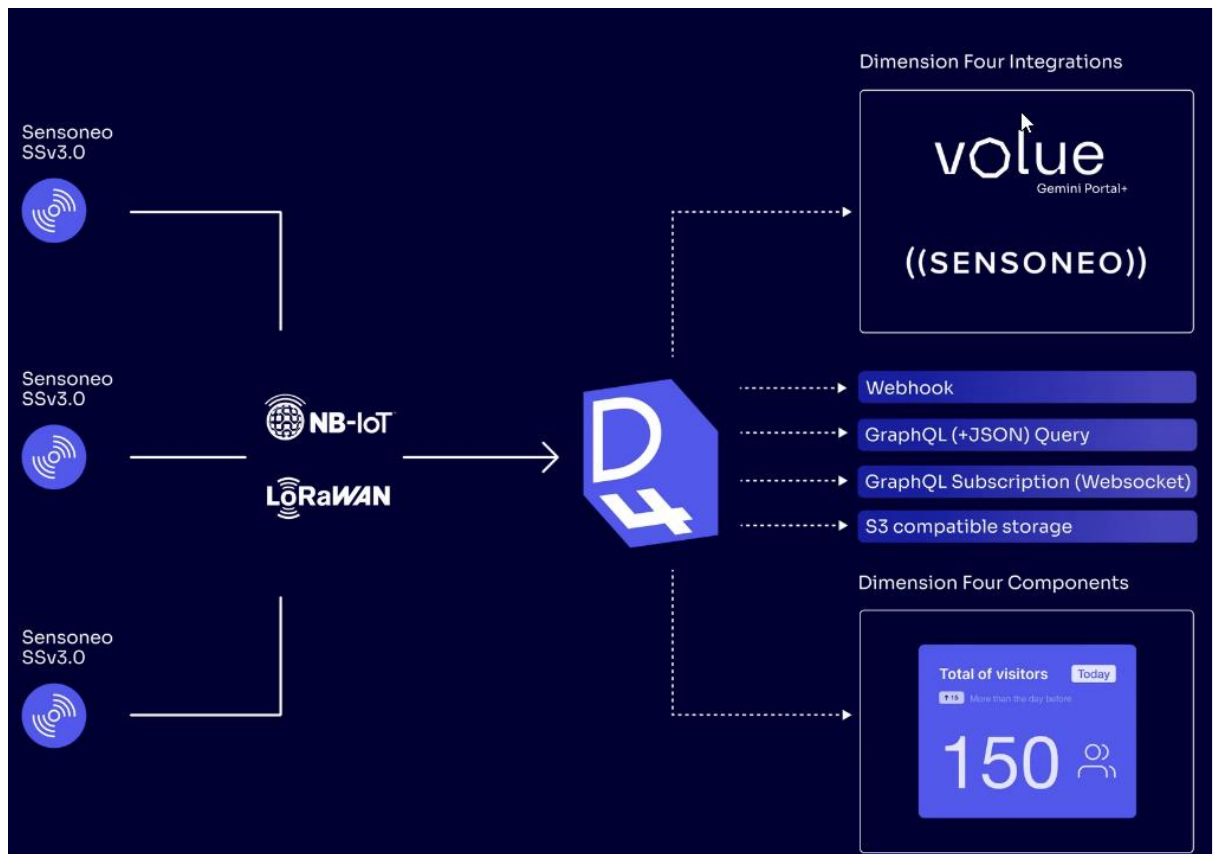


Figure 39: Dimesion 4 schematic view [18].

5.2.1 Tenant

In Dimension4, a tenant is a logical grouping of users, devices, and applications. Tenants are isolated from each other, which means that users in one tenant cannot access the resources of another tenant. This makes tenants a good way to organize and manage IoT solutions for multiple customers or departments.

Each tenant has its own unique identifier and its own set of permissions. Permissions control what users in a tenant can do, such as which devices they can access and which applications they can use.

Tenants, Figure 40, can be created and managed using the DimensionFour Console or the DimensionFour API. Figure shows the tenant which is created to be used in this project.

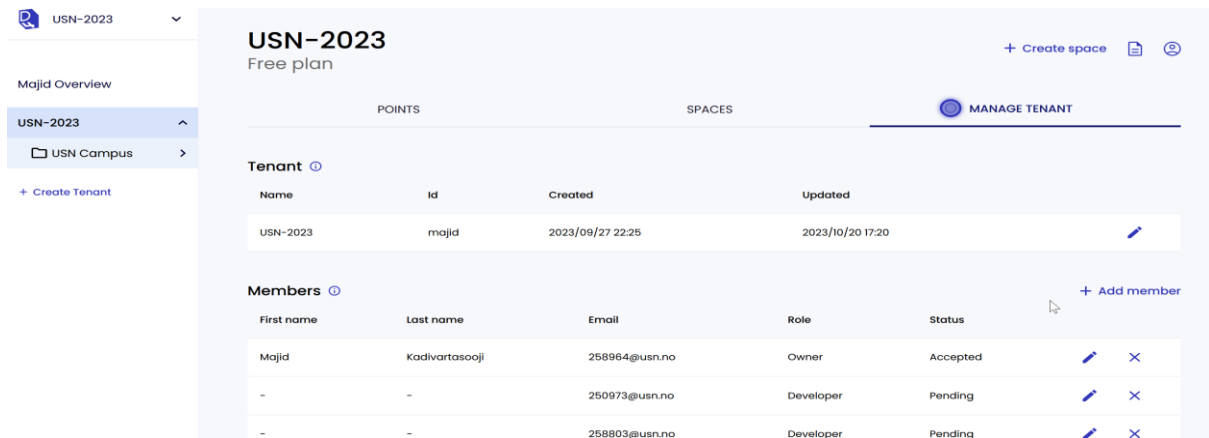


Figure 40: Tenant

5.2.2 Spaces

Spaces in Dimension4 are logical containers that can be used to organize and manage IoT devices, applications, and data. Spaces are isolated from each other, which means that users in one space cannot access the resources of another space. This makes spaces a good way to group related devices, applications, and data together.

Each space has its own unique identifier and its own set of permissions. Permissions control what users in a space can do, such as which devices they can access and which applications they can use.

Spaces, Figure 41, can be created and managed using the DimensionFour Console or the DimensionFour API. figure depicts the spaces that have been created to manage points and signals of this project.

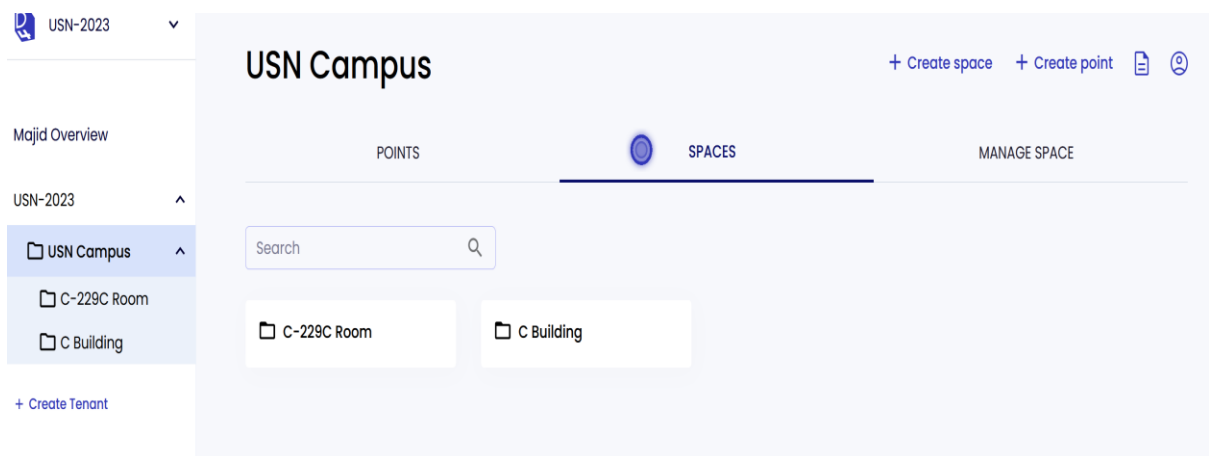


Figure 41: Spaces in the project

5.2.3 Point

Points in Dimension4 are the fundamental building blocks of data in the platform. A point represents a single data record, and it can contain any type of data, such as sensor readings, GPS coordinates, or product information.

Points are organized into spaces, which are logical containers for grouping related data together. Spaces can be nested, so you can create hierarchies of data to organize your data in a way that makes sense for your application.

Points can also be tagged with metadata, which is additional information about the point. Metadata can be used to describe the point, such as the device that generated it or the location where it was collected. Metadata can also be used to filter and sort points, and to create custom visualizations. Figure 42 shows a point which has been created to be connected to outdoor temperature sensor.

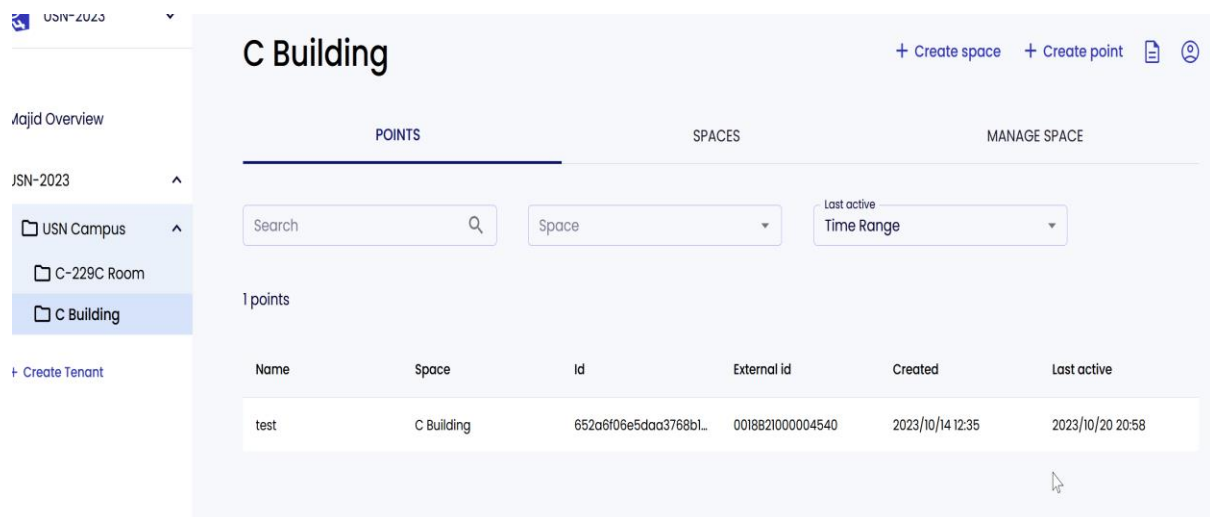


Figure 42: Point created in C building.

5.2.4 Signal

After a point is installed, a signal can be created to show the data. By creating a token, this data can be consumed to build the application. Figure 43 shows the signals from the point that connected to outdoor temperature sensor [18].

USN-2023

Majid Overview

USN-2023

USN Campus

C-229C Room

C Building

+ Create Tenant

test

SIGNALS

MANAGE POINT

Last 10 signals with real time data

Type	Unit	Data	Created at
Temperature	CELSIUS_DEGREES	4.2	2023-10-20T18:58:01.912Z
Temperature	CELSIUS_DEGREES	4.2	2023-10-20T18:48:01.886Z
Temperature	CELSIUS_DEGREES	4.2	2023-10-20T18:36:01.892Z
Temperature	CELSIUS_DEGREES	4.2	2023-10-20T18:28:01.848Z
Temperature	CELSIUS_DEGREES	4.3	2023-10-20T18:18:02.040Z
Temperature	CELSIUS_DEGREES	4.3	2023-10-20T18:08:09.945Z

Figure 43: Test section in Dimension 4.

6 Datalogging Application

6.1 An overview of Pipedream

Pipedream offers the swiftest means to streamline the automation of processes that involve API connections. You can create and execute workflows with granular code-level control when necessary, and effortlessly implement solutions without writing any code when it's not required. In Figure 44 is shown the Pipedream interface.

The Pipedream platform encompasses:

- A serverless runtime and workflow service
- Access to source code for triggers and actions compatible with a wide array of integrated applications
- Convenient one-click OAuth and key-based authentication for over 1000 APIs, allowing you to utilize tokens directly within your code or with pre-built actions.

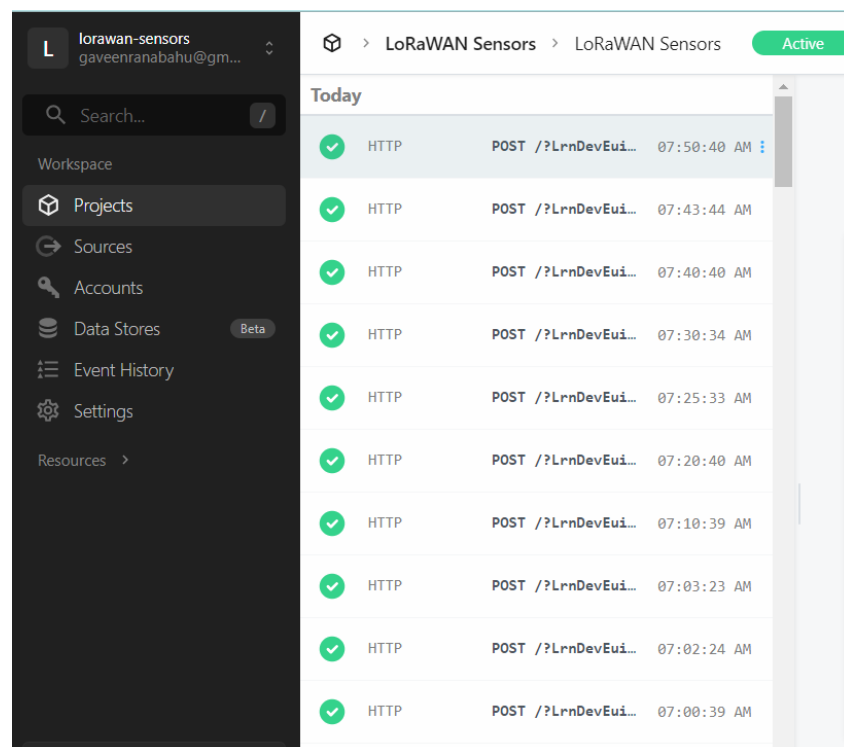


Figure 44 - Pipedream interface

Workflows, Figure 45, simplify the process of integrating the applications, data, and APIs, all without the need to handle servers or infrastructure, [19].

- These workflows are built from code organized and executed as a series of linear steps.
- You can initiate your workflow in response to any event, such as HTTP requests or scheduled tasks.
- Incorporate steps to execute code written in Node.js, Python, Go, or Bash, utilizing a wide range of npm, pip, or Go packages, along with prebuilt actions.
- The steps are executed in the order in which they are arranged in your workflow.
- You can export values from each step and access them through the steps object.

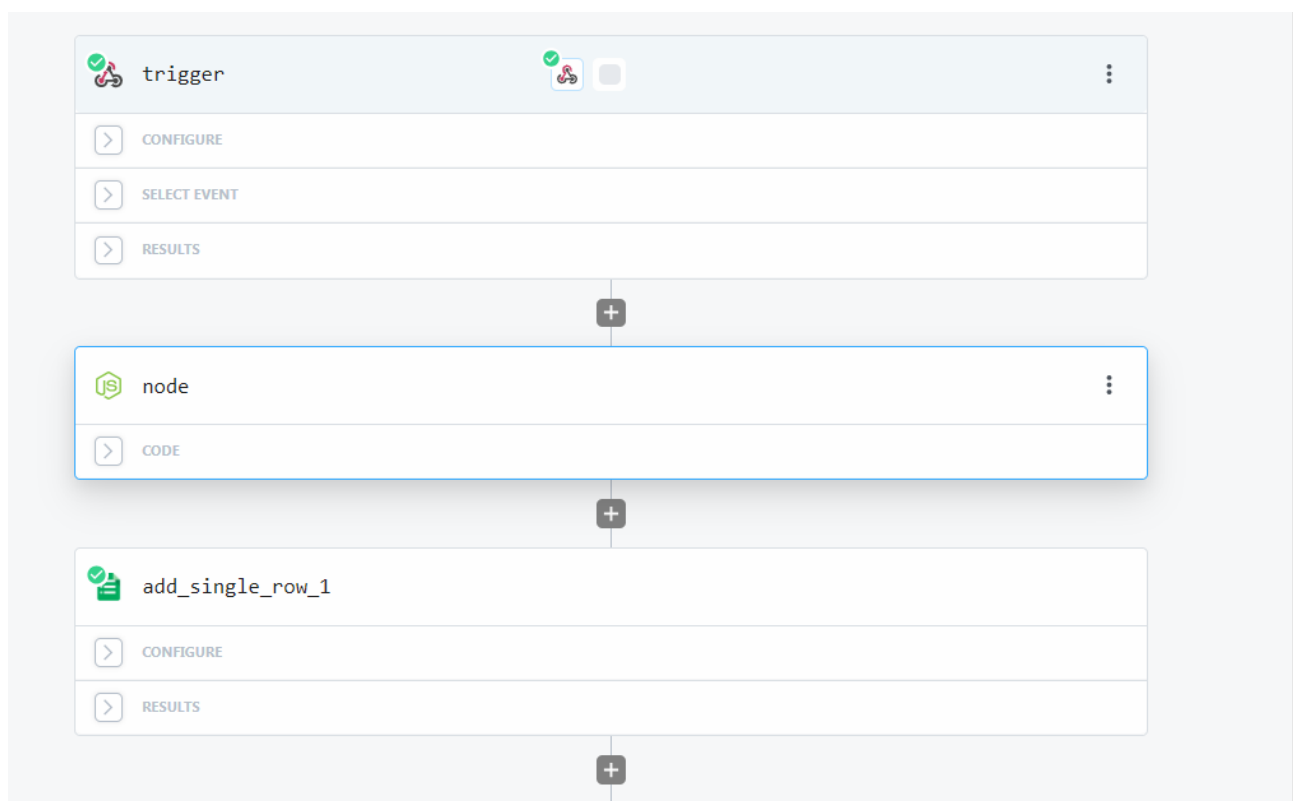


Figure 45 - Workflow interface

6.2 Reviewing Events

When you examine events, selecting a specific event will trigger the opening of a panel that showcases the executed steps, their respective configurations, and the comprehensive performance and outcomes of the entire workflow.

Please note that currently, it is not feasible to modify the workflow using the event history you've selected. You can exclusively choose events visible in the event inspector for the workflow builder.

The upper section of the event history details will provide information, including the overall status of that particular event execution and any errors encountered.

In the case of an error message, Figure 46, the hyperlink at the bottom of the error message will direct to the corresponding workflow step responsible for the error.

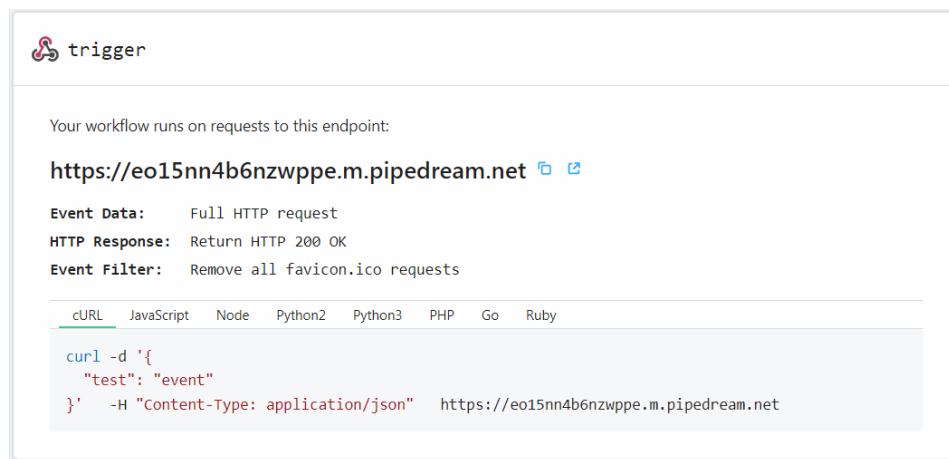


Figure 46 - Endpoint Configuration

6.3 Configuring the devices communication path

Following the establishment of the request endpoint, the next crucial step is to associate this endpoint with the devices in the network. This association is essential as it enables these devices to transmit data packets to the central server effectively. These data packets are specifically formatted in JSON (JavaScript Object Notation), a widely adopted and human-readable data interchange format. JSON format provides a standardized and structured way to encapsulate and convey data, ensuring that the data transmission between devices and the server is not only efficient but also easily interpretable. This process plays a pivotal role in facilitating seamless and organized data transfer, which is a fundamental aspect of many modern data-driven applications and systems.

6.3.1 Configuring the Application Server

The configuration interface provided below is part of the Application Server, serving as a crucial component of the system. It is imperative that the content type for this configuration is set to JSON, ensuring the appropriate data format for processing. In this context, the "destination" refers to the vital specification of the precise path to be utilized in conjunction with the Pipedream HTTP location, emphasizing the need for precision and accuracy in determining where the data is to be directed and processed. This configuration thus plays a pivotal role in facilitating seamless communication and data transmission within the system. Figure 47 shows the explained configuration.

The screenshot displays the 'Application server' configuration window. At the top, there are buttons for Delete, Save, Cancel, and Close. The 'Application server' tab is active, showing fields for Name (Pipedream), ID (TWA_100050938.66184.AS), Content Type (JSON), Type (HTTP Application Server (LoRaWAN)), and Status (Active). Below these is a table for 'HTTP custom headers' with columns 'Name' and 'Value', currently showing 'No result found.' and buttons for 'Add' and 'Delete'. The 'Uplink/downlink security' section shows 'Status: Inactive' and 'Max timestamp deviation: -' with an 'Activate' button. The 'Route' section shows 'Source ports: *' and 'Routing strategy: Sequential'. A 'Destinations' table is also visible with one entry: 'Destination https://eo15nn4b6nzwppm.pipedream.net'. The bottom right corner indicates the version 'v17.18.4-441cb1c67' and copyright '©2023 Actility'.

Figure 47 - Application Server Configuration

6.3.2 Configuring the AS routing profiles

When you're in the process of choosing the appropriate AS routing profiles, it becomes essential to meticulously assign the designated destination that we previously established within the Application Server infrastructure. This assignment is of paramount importance, as it serves as the key mechanism through which data packets are transmitted to the web. It is only through this meticulous assignment that the system can effectively capture and process incoming requests, ensuring a seamless and responsive interaction between the Application

Server and the web, thereby facilitating the reliable and efficient exchange of data as below Figure 48.

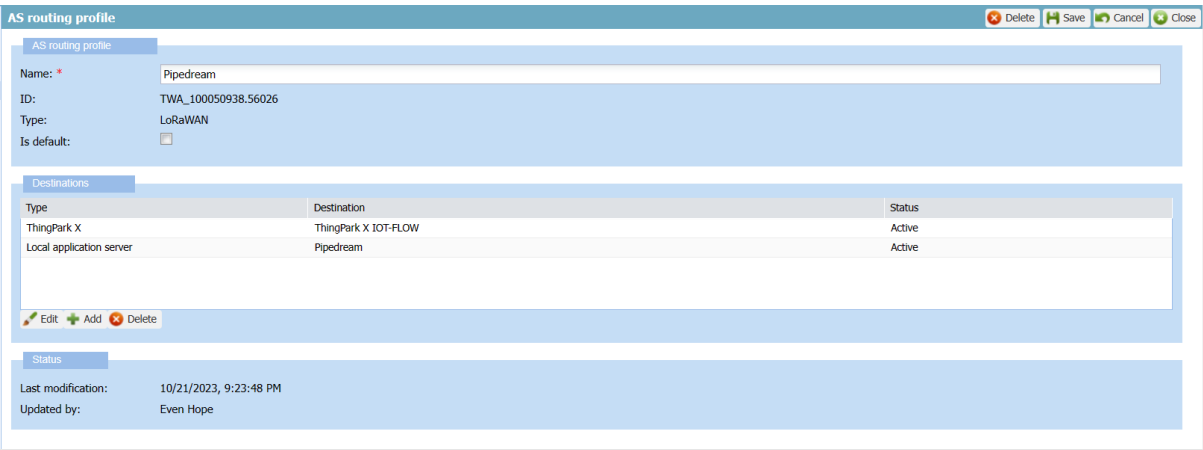


Figure 48 - AS Routing Profile Configuration

6.3.3 Assigning the routing path to Devices to communicate.

Presented below is the Network configuration interface tailored specifically for the sensor system. Within this interface, one of the critical elements that demand your attention is the network routing option. In this context, it is imperative to meticulously choose the network routing configuration corresponding to the Pipedream service, which has been explicitly named and defined for this purpose. By making this well-considered selection, you ensure that the sensor system is precisely connected to the intended network pathway, thereby establishing a seamless and efficient data communication channel that aligns with the specified Pipedream service, ensuring the successful transmission of data between the sensor and its intended destination.



Figure 49 - Network Routing for Device Configuration

6.4 Building up the Workflow

Below steps are taken to configure the workflow and build the data pipeline.

6.4.1 Configure the Trigger

The settings page for a trigger allows users to configure the trigger to run when certain criteria are met. For example, you could configure a trigger to run when a new email is received with a certain subject line or when a new file is uploaded to a certain folder.

The settings page also allows users to select the data that they want to export from the trigger and the data that they want to reference in future steps of the workflow. This data can then be used to automate tasks in other apps or services.

Here is a more specific detail about the below interface.

- The Trigger section shows the event that will start the workflow. In this case, the trigger is set to run when a new HTTP request is made to a certain endpoint URL.
- The Domains section allows you to select one or more domains to use for your trigger URL, Figure 50.
- The Event Data section allows you to select the data that you want to export from the trigger and the data that you want to reference in future steps of the workflow.
- The HTTP Response section allows you to customize what happens when an HTTP request is made to the trigger URL.

The screenshot shows a web interface titled 'trigger' with a green checkmark icon. The interface is divided into several sections:

- CONFIGURE**: A tab with a dropdown arrow and a mute icon.
- Event Data**: A dropdown menu showing 'Full HTTP request'. Below it, text says 'Select the data to export and reference in future steps via the `steps` object'.
- HTTP Response**: A dropdown menu showing 'Return HTTP 200 OK'. Below it, text says 'Customize what happens when an HTTP request is made to this endpoint URL'.
- Domains**: A dropdown menu showing 'pipedream.net'. Below it, text says 'Select one or more domains to use for your trigger URL'.
- Optional Fields**: A section with a button '+ Filter favicon.ico' and the text 'Remove all favicon.ico requests'.
- SELECT EVENT**: A button with a right arrow.
- RESULTS**: A button with a right arrow.
- Buttons**: At the bottom, there are two buttons: 'Test workflow' and 'Continue'.

Figure 50 - Data Capturing Interface

6.4.2 Adding the captured data

To send the data to the app, you would need to select the app from the list of available apps. You can then configure the workflow to send the data to the app in a variety of ways, such as by sending an HTTP request or by adding the data to a database. Here are the ways we can send data to other locations, Figure 51. For our easiness and considering the cost factors we decided to store the values in google sheets.

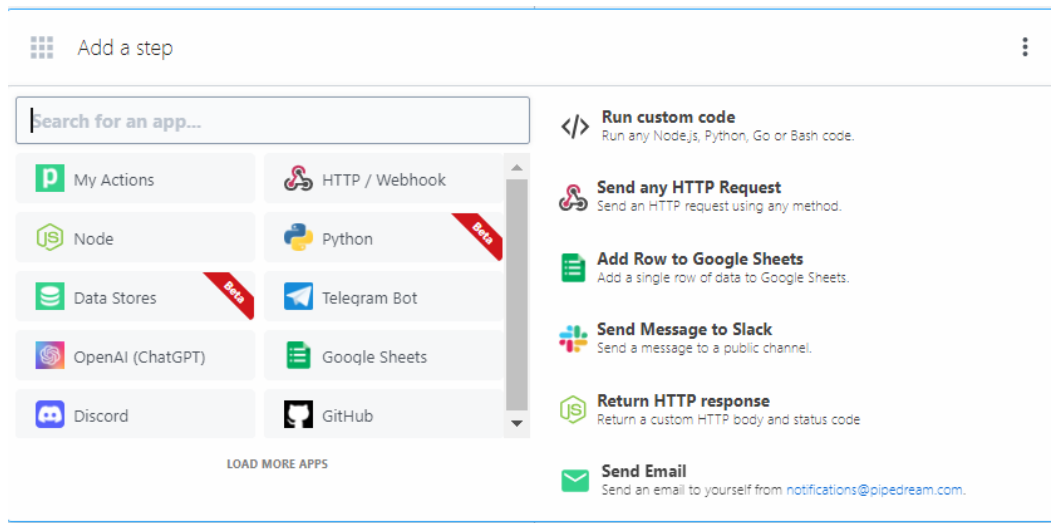


Figure 51 - Channels to Storing data.

6.4.3 Adding received data to Google Drive

The following configuration pertains to the process of appending rows to the data store. To accomplish this, it is essential you must furnish the JSON path that designates the destination location. Once this is done, the system will diligently store specific data within each data packet at the designated location. to specify the column headings. Subsequently, in the workflow as below Figure 52.

The screenshot displays the configuration interface for a workflow step named 'add_single_row_1'. The interface is organized into several sections:

- Google Sheets Account:** A dropdown menu showing 'gaveenranabahu@gmail.com' with a note that 'Credentials are encrypted. Revoke anytime.'
- Spreadsheet:** A dropdown menu showing 'Lorawan Sensors Outdoor Temprature data' with a unique ID '1AObx2gA7gJoFnftwD0pPhpRQokkdm9byAGXYHjz3Wo0'.
- Sheet Name:** A dropdown menu showing 'Sheet1'.
- Does the first row of the sheet have headers?:** A dropdown menu set to 'Yes', with a sub-note: 'If the first row of your document has headers we'll retrieve them to make it easy to enter the value for each column.'
- DevEUI:** A text field containing the JSON path '{{steps.trigger.event.body.DevEUI_uplink.DevEUI}}' with a 'Hide' link.
- DevAddr:** A text field containing the JSON path '{{steps.trigger.event.body.DevEUI_uplink.DevAddr}}' with a 'Hide' link.
- CustomerID:** A text field containing the JSON path '{{steps.trigger.event.body.DevEUI_uplink.CustomerID}}' with a 'Hide' link.
- BaseStationData:** A text field containing the JSON path '{{steps.trigger.event.body.DevEUI_uplink.payload_hex}}' with a 'Hide' link.

At the bottom, there is a 'Test' button with a dropdown arrow and a blue 'Continue' button.

Figure 52 - Configuration Interface

6.4.4 Data Stores

Data stores are key-value databases to easily set and get any JSON-serializable data and maintain state across workflow executions.

It's useful for counting values, summing up data between run, or tracking unique data points like email addresses.

Data stores are:

- Persisted between workflow runs
- Shareable between workflows
- Quick to get started, since they don't require any setup or external connections.

You can also use native pre-built actions to store, update, and clear data without code.

6.4.5 Getting the data from JSON file received

The below configuration in Figure 53 is to show the structure of the data packet.

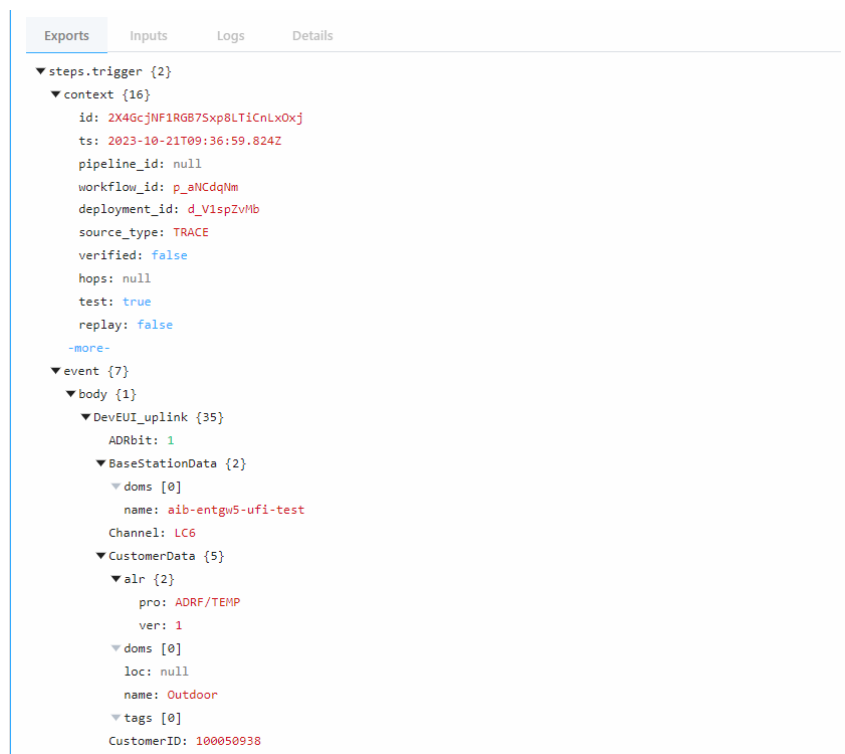


Figure 53 – Structure of the data packet.

Below, Figure 54 the full JSON file can be seen.

```
Lrrid: 10004D88
▼ Lrrs {1}
  ▼ Lrr [1]
    ▼ 0 {5}
      Chain: 0
      LrrESP: -90.832695
      LrrRSSI: -90
      LrrSNR: 6.75
      Lrrid: 10004D88
    MType: 2
    MeanPER: 0
    ModelCfg: 0
    NbTrans: 1
    SpFact: 7
    SubBand: G2
    Time: 2023-10-21T09:18:05.874+00:00
    TxPower: 2
    mic_hex: 5e64c856
    payload_hex: 57b000120016

DevAddr: 34C194C1
DevEUI: 0018821000004540
DevLrrCnt: 1
▼ DriverCfg {2}
  ▼ app {3}
    mId: temp
    pId: adeunis
    ver: 2
  ▼ mod {3}
    mId: temp
    pId: adeunis
    ver: 2
  DynamicClass: A
  -more-
client_ip: 52.16.83.187
► headers {5}
method: POST
path: /
▼ query {3}
  LrnDevEui: 0018821000004540
  LrnFPort: 1
  LrnInfos: TWA_100050938.66184.AS-1-803612983
▼ url
  https://eo15nn4b6nzwpe.m.pipedream.net/?
  LrnDevEui=0018821000004540&LrnFPort=1&LrnInfos=TWA_100050938.66184.AS-1-803612983
```

Figure 54 - Captures JSON file Overview

6.4.6 Aspects of the JSON file

The provided JSON file contains structured data that appears to represent information related to the trigger event or communication event. Here's a detailed description of its key components:

1. **steps.trigger**: This section seems to provide information about the trigger event, possibly indicating its position or order within a sequence of steps.
2. **context**: This part contains contextual information related to the trigger event. It includes various properties such as deployment ID, emitter ID, hops, event ID, owner ID, platform version, replay status, resume status, and the source type, which might describe the origin of the event.
3. **event**: This section provides additional details about the event itself and its associated data.
 - **body**: The "body" object appears to contain the primary data related to the event, specifically under the key "DevEUI_uplink."
 - **ADRbit**: A property indicating the ADR (Adaptive Data Rate) bit with a value of 1, possibly denoting an ADR status.
 - **BaseStationData**: This object includes information related to a base station, such as "doms," "name," "BatteryLevel," and "BatteryTime."
 - **CustomerData**: Contains customer-specific data, including "alr" data with "pro," "ver," and "doms" properties, as well as "name" and "tags."
 - **DriverCfg**: Provides driver configuration data, including "app," "mId," "pId," "ver," and "mod."
 - **DynamicClass**: A property that seems to represent a classification or type, indicated as "A."
 - **FCntDn**: An integer representing FCntDn (Downlink Frame Counter).
 - **FCntUp**: An integer representing FCntUp (Uplink Frame Counter).
 - **FPort**: An integer specifying the FPort, which may be related to the port used for data transmission.
 - **Frequency**: A floating-point value indicating the frequency.
 - **InstantPER**: An integer that may denote Instant Packet Error Rate.
 - **Late**: An integer indicating whether the event is late, with a value of 0 (not late).
 - **LostUplinksAS**: An integer indicating the count of lost uplinks, with a value of 0.
 - **Lrcid**: A string representing Lrcid.
 - **LrrESP**: A floating-point value indicating LrrESP (Energy to Sensitivity Performance).
 - **LrrLAT**: A floating-point value representing LrrLAT (Latitude of the base station).
 - **LrrLON**: A floating-point value representing LrrLON (Longitude of the base station).
 - **LrrRSSI**: An integer indicating LrrRSSI (Received Signal Strength Indicator).

- **LrrSNR**: A floating-point value indicating LrrSNR (Signal-to-Noise Ratio).
- **Lrrid**: A string representing Lrrid, which may be related to the base station ID.
- **Lrrs**: This object contains information about multiple Lrr (base station) instances.
 - **Lrr**: An array with properties for individual Lrr instances, including "Chain," "LrrESP," "LrrRSSI," "LrrSNR," and "Lrrid."
- **MType**: An integer indicating MType (Message Type).
- **Margin**: An integer specifying the margin.
- **MeanPER**: An integer indicating Mean Packet Error Rate.
- **ModelCfg**: An integer representing Model Configuration.
- **NbTrans**: An integer indicating the number of transmissions.
- **SpFact**: An integer specifying Spreading Factor.
- **SubBand**: A string that might indicate the sub-band used.
- **Time**: A timestamp indicating the event's time and date.
- **TxPower**: An integer specifying the transmission power.
- **mic_hex**: A hexadecimal string.
- **payload_hex**: A hexadecimal string representing payload data.
- **client_ip**: A string representing the client's IP address.
- **headers**: This object includes HTTP request headers with properties like "accept," "content-length," "content-type," "host," and "user-agent."
- **method**: A string specifying the HTTP request method (e.g., "POST").
- **path**: A string indicating the request path ("/").
- **query**: This object contains query parameters, including "LrnDevEui," "LrnFPort," and "LrnInfos."
- **url**: A string representing the full URL of the request, including query parameters.

In summary, the JSON file provides detailed information about a specific event, including data related to a communication or data transmission process, HTTP request details, and context information.

6.4.7 Capturing the data

The "payload_hex" field in the provided JSON file is a hexadecimal string that represents the payload data associated with a specific event. Hexadecimal (or hex) is a base-16 numbering system commonly used to represent binary data in a more human-readable format. Each hexadecimal digit represents four binary bits.

In this context:

- payload_hex is the key or label for this field.
- "57b000120016" is the actual hexadecimal string.

To interpret this payload data, you would need to convert it to a format that makes sense in the context of the specific application or protocol that generated it. The meaning and structure of the payload depend on the system and the data being transmitted.

If you have information about the protocol or application used to generate this payload, you can decode it to obtain the actual data or information it represents. Typically, payload data contains information, measurements, or commands relevant to the communication event or device involved in the process as in Figure 55.

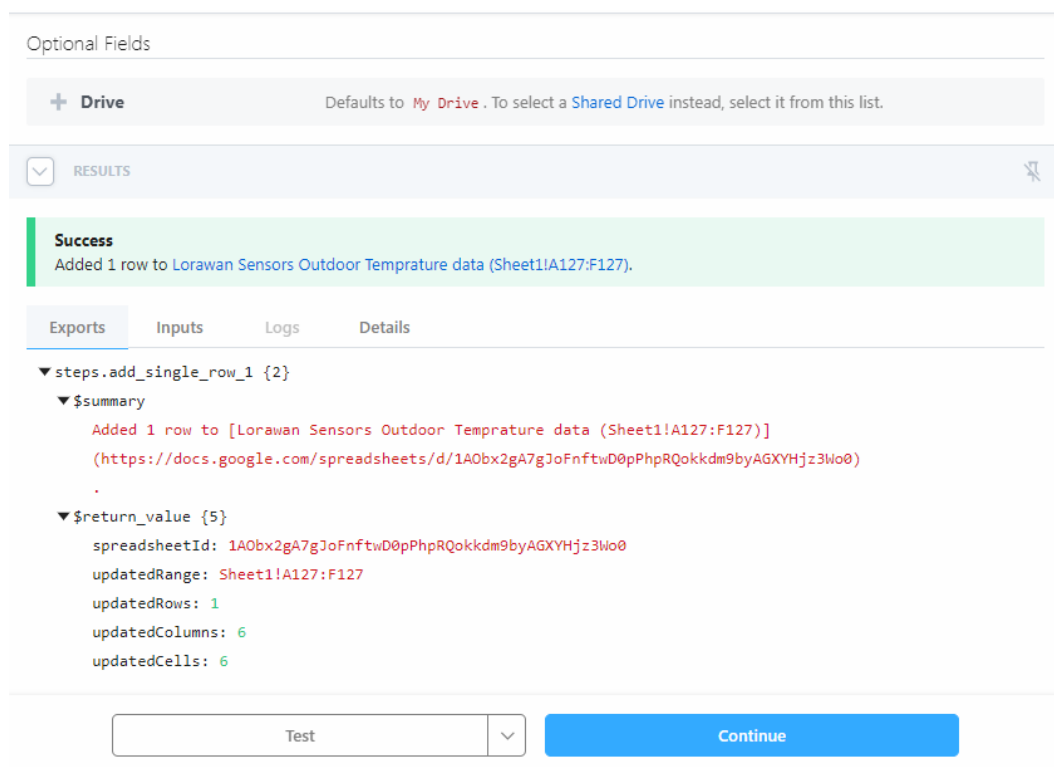


Figure 55 - Data Adding Configuration

6.4.8 Decoding the data

LoRaWAN (Long Range Wide Area Network) sensors use a specific codec to encode and decode the data transmitted between LoRaWAN devices and network servers [20]. The codec plays a crucial role in ensuring that the data sent by LoRaWAN sensors is properly formatted and can be interpreted by the network server and application server.

The Figure 56 shows the interface of the Adeunis decoder.

Decoder

Select the product to decode

Supported frames

Enter below the payload to be decoded

Decode

Information
This tool is made available by Adeunis for illustrative purposes to facilitate the understanding and implementation of data of adeunis@ products.
Adeunis is not responsible for the results and their use.

adeunis
IoT PRODUCTS & SOLUTIONS

Version 1.6.1

www.helpdesk-adeunis.com

Figure 56 - ADEUNIS Decode Interface


Here's a general description of LoRaWAN sensors codecs:


- **Data Compression and Packaging:** LoRaWAN sensors codecs are responsible for compressing and packaging the data generated by the sensors. They optimize data transmission to minimize bandwidth usage and power consumption, which is essential for LoRaWAN's long-range and low-power characteristics.
- **Data Format:** The codec defines the format in which the sensor data is encoded before transmission. This includes specifying data types, units of measurement, and how the data is structured within the payload.
- **Decoding at the Server:** On the server side, the network server and application server use the codec to decode the received data. This process involves unpacking the payload, converting binary data into human-readable formats, and interpreting sensor values.
- **Interoperability:** LoRaWAN sensors codecs are standardized to ensure interoperability among different vendors' LoRaWAN devices and network servers. This standardization


allows devices from various manufacturers to work together seamlessly within a LoRaWAN network.


- Customization: In some cases, LoRaWAN sensors codecs can be customized to suit specific applications or industries. Customization might involve defining additional data types or incorporating proprietary data formats.
- Security: Data encryption and security measures are often part of the codec's functionality to protect sensitive information transmitted over the network.
- Efficiency: Codecs are designed to be efficient in terms of data transmission, making the most of the limited bandwidth available in LoRaWAN networks. They also play a role in ensuring that devices can operate on battery power for extended periods.


LoRaWAN sensors codecs are a critical component of the LoRaWAN ecosystem, ensuring that data is transmitted, received, and interpreted accurately and efficiently. They are essential for a wide range of applications, including smart agriculture, smart cities, industrial IoT, and more.

 ADEUNIS CODEC

 Decoder

 Encoder

 Download

 Changelog

Adeunis Codecs library

JavaScript/Node.js library of **Adeunis codecs v1.6.1**
Library able of decoding all the frames coming from our products.

Supported products

Analog	Breath	Comfort	Comfort CO2
Delta P	Dry Contacts	Motion	Pulse
Pulse 4 NB-IoT	Repeater	Temp	
TIC CBE Linky	TIC PME-PMI		

You can find all the details of each product (reference, app version, compatibility) [here](#) .





Getting started

Go to project folder and install dependencies

```
npm install
```

Create an empty directory `npm_demo`, go into it, and execute

Download v1.6.1

-  Library
-  Libraries by products
-  Multitech library
-  Source

External links



-  NPM
-  Node-RED

Figure 57 - ADEUNIS Codecs Library

6.4.9 Using TypeScript to decode the data

The provided JavaScript code below, Figure 58, is responsible for decoding a payload originating from a specific device type identified as 'analog'. It utilizes the '@adeunis/codecs' module to facilitate the decoding process. The payload data, represented as the string '42500110000002100000', is to be decoded, and the code logs a descriptive message to the console before beginning the decoding operation. It instantiates a Decoder object and configures it for the 'analog' device type. The payload is then decoded using this decoder, and the result is stored in the parserResult variable. The code subsequently checks for decoding errors, and if none are found, it converts the parserResult into a nicely formatted JSON string and assigns it to the payloadResult variable. Finally, the code logs the value of payloadResult to the console, displaying either the JSON representation of the successfully decoded data or, in the case of an error, the message 'decoding issue'.

```
const codec = require('@adeunis/codecs');

// Products types are defined in DecoderProducts enum (src/shared/product.enum.ts)
const productType = 'analog';
const payloadValue = '42500110000002100000';
let payloadResult;

console.log(`Decoding ${productType} frame => ${payloadValue}`);

const decoder = new codec.Decoder();

// Configure the decoder for the appropriate device
decoder.setDeviceType(productType);

// Decode the given payload
let parserResult = decoder.decode(payloadValue);

// Incompatible frame and product
if (parserResult.error) {
    payloadResult = 'decoding issue';
} else {
    // Display result
    payloadResult = JSON.stringify(parserResult, null, 2);
}

console.log(payloadResult);
```

Figure 58 - TypeScript code to decode the Payload.

7 Monitoring Application

In this section the monitoring application will be taken into discussion.

7.1 Tableau Overview

Tableau is a robust data visualization and business intelligence tool that empowers organizations to convert unprocessed data into valuable insights [21]. It facilitates the exploration, analysis, and dissemination of data in an interactive and easily understandable manner. Tableau allows users to generate dynamic dashboards, reports, and graphs for visualizing intricate datasets, simplifying the process of discovering patterns, trends, and actionable knowledge. Whether you're a data analyst, business expert, or anyone dealing with data, Tableau offers a versatile solution for data-informed decision-making and the efficient communication of findings.

Here are some key features and aspects of Tableau:

1. **Data Connectivity:** Tableau can connect to various data sources, including databases, spreadsheets, cloud services, and web data connectors. This wide range of connectivity options allows users to access and analyze data from diverse sources.
2. **Data Preparation:** The software includes data preparation features that help users clean, shape, and transform their data, ensuring it's in the right format for analysis.
3. **Visualization:** Tableau is known for its exceptional data visualization capabilities. Users can create interactive and customizable charts, graphs, maps, and dashboards, making it easy to communicate insights effectively.
4. **Drag-and-Drop Interface:** Tableau's intuitive drag-and-drop interface simplifies the process of creating visualizations, even for those without extensive technical skills.
5. **Real-Time Analytics:** It supports real-time data analysis, allowing businesses to make informed decisions based on up-to-the-minute information.
6. **Mobile-Friendly:** Tableau offers mobile apps for both iOS and Android, ensuring that users can access and interact with their data on the go.
7. **Collaboration:** Users can share their Tableau dashboards and reports with team members and stakeholders. Collaboration features enable real-time updates and comments.
8. **Security:** Tableau offers robust security features to protect sensitive data and comply with privacy regulations.
9. **Integration:** It can be integrated with various other tools and platforms, including cloud services like AWS and Azure, as well as big data technologies.
10. **Scalability:** Tableau is suitable for small teams and large enterprises alike. It can scale with the needs of your data.

7.2 Connecting to the Tableau

Tableau is a powerful data visualization and business intelligence tool that allows you to connect to various data sources, including databases. Here are some different types of databases that you can connect to using Tableau:

1. Relational Databases:
 - Microsoft SQL Server: You can connect to SQL Server databases to visualize and analyze your structured data.
 - MySQL: Tableau supports MySQL, an open-source relational database management system.
 - PostgreSQL: Connect to PostgreSQL, a powerful open-source database system.
 - Oracle Database: Tableau can connect to Oracle databases for enterprise-level data analysis.
 - IBM Db2: You can use Tableau to connect to IBM Db2 databases.
2. NoSQL Databases:
 - MongoDB: Tableau has connectors for MongoDB, a popular NoSQL database, to analyze unstructured or semi-structured data.
 - Cassandra: Connect to Apache Cassandra, a distributed NoSQL database, for large-scale data analysis.
3. Cloud-Based Databases:
 - Amazon Redshift: Tableau supports Amazon Redshift, a cloud data warehousing service.
 - Google BigQuery: Connect to Google BigQuery, a fully-managed data warehouse, for big data analytics.
 - Microsoft Azure SQL Database: Tableau can connect to Microsoft's cloud-based SQL Database service on Azure.
4. Data Warehouses:
 - Snowflake: Connect to Snowflake, a cloud-based data warehousing platform.
 - Teradata: Tableau supports Teradata data warehouses for advanced analytics.
5. OLAP Cubes:
 - Microsoft Analysis Services: Connect to OLAP cubes created using Microsoft Analysis Services for multidimensional data analysis.
6. Web Data Connectors:
 - Web data connectors in Tableau allow you to connect to web-based data sources and APIs. You can use them to access data from various web services, such as social media platforms or online data repositories.
7. Excel Spreadsheets:
 - You can import data from Excel spreadsheets into Tableau for analysis and visualization.
8. ODBC and OLE DB Data Sources:
 - Tableau provides support for generic ODBC (Open Database Connectivity) and OLE DB (Object Linking and Embedding Database) connections, allowing you to connect to a wide range of data sources.
9. Custom Data Connectors:

- Tableau provides APIs and SDKs that allow you to develop custom data connectors to connect to proprietary or specialized data sources.

10. Spatial Databases:

- You can connect to spatial databases like PostGIS for mapping and geospatial analysis.

Tableau's versatility in connecting to various data sources makes it a valuable tool for working with different types of databases and data storage systems, enabling users to create insightful and interactive data visualizations and dashboards.

In Figure 59 the connection for the tableau is taken from google sheets. It is created as below connection.

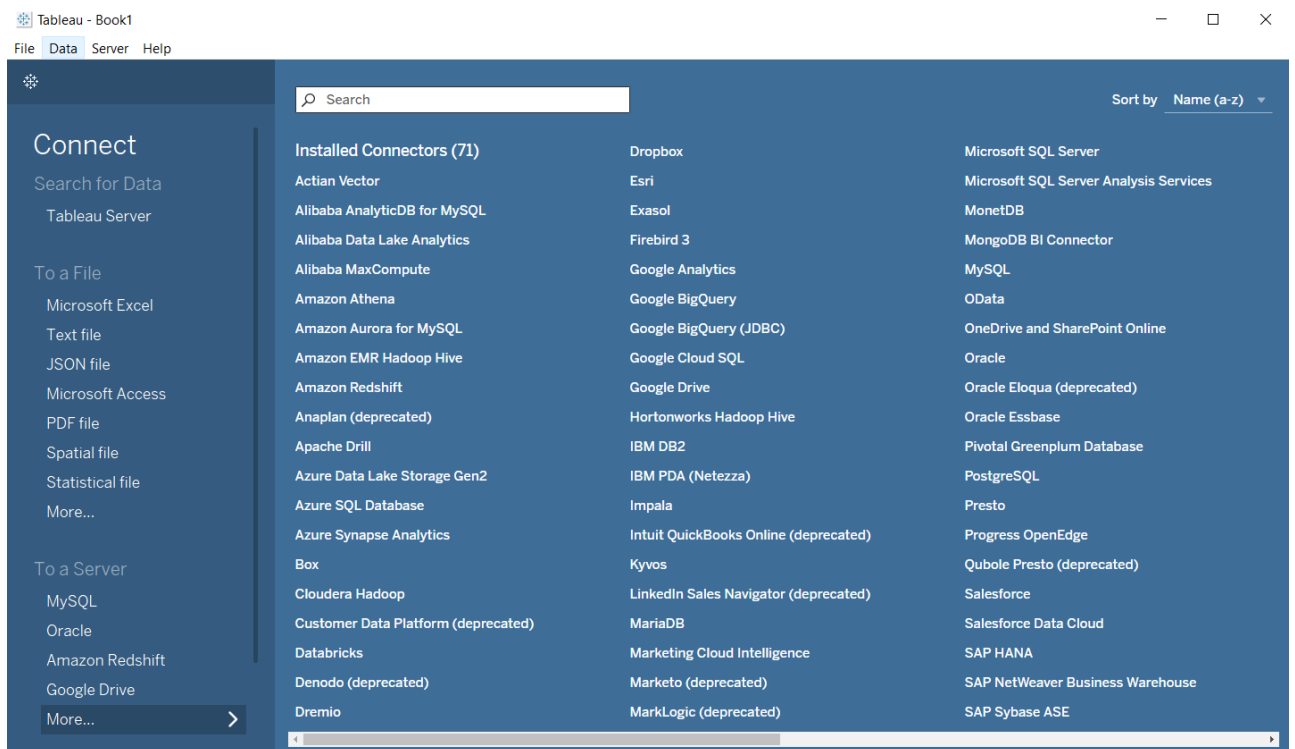


Figure 59:- Tableau Data Connectors

7.3 Developed Dashboards

In this section developed dashboards of the monitoring application are shown.

7.3.1 Outdoor temperature data Dashboard

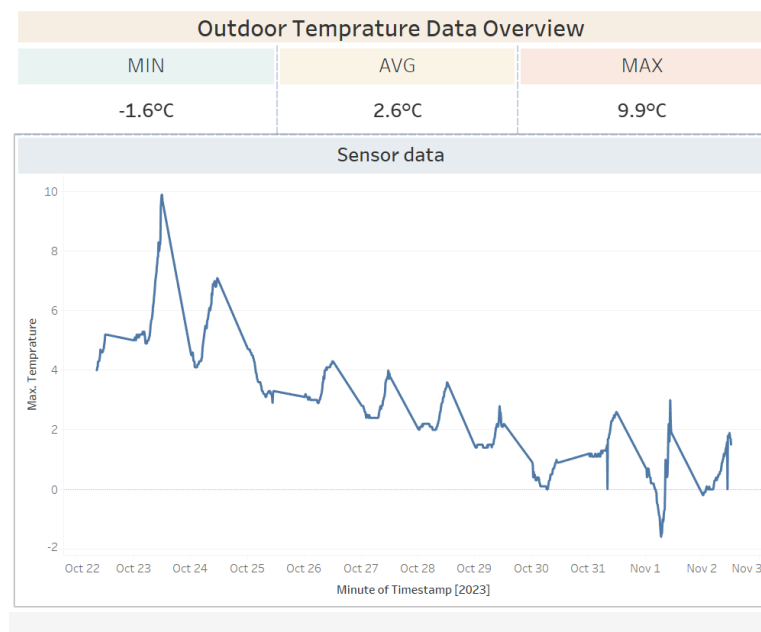


Figure 60 - Outdoor temperature data Dashboard

7.3.2 Indoor temperature data Dashboard

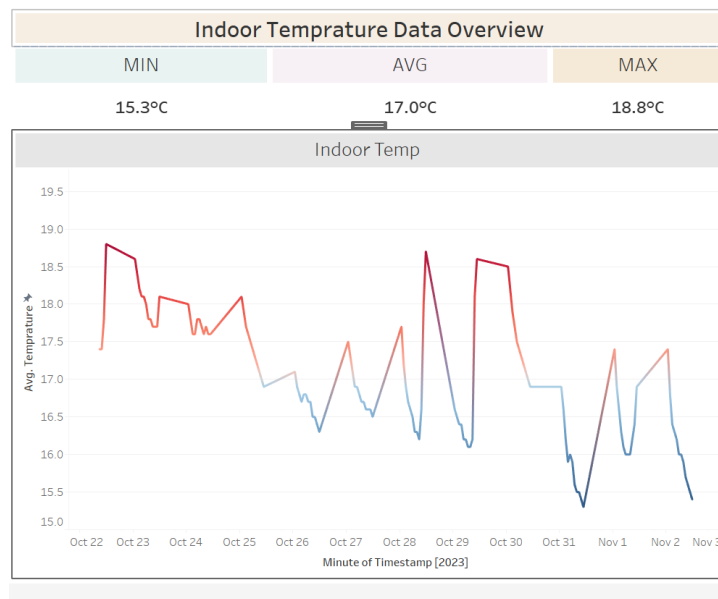


Figure 61 - Indoor temperature data Dashboard

7.3.3 Indoor humidity data Dashboard

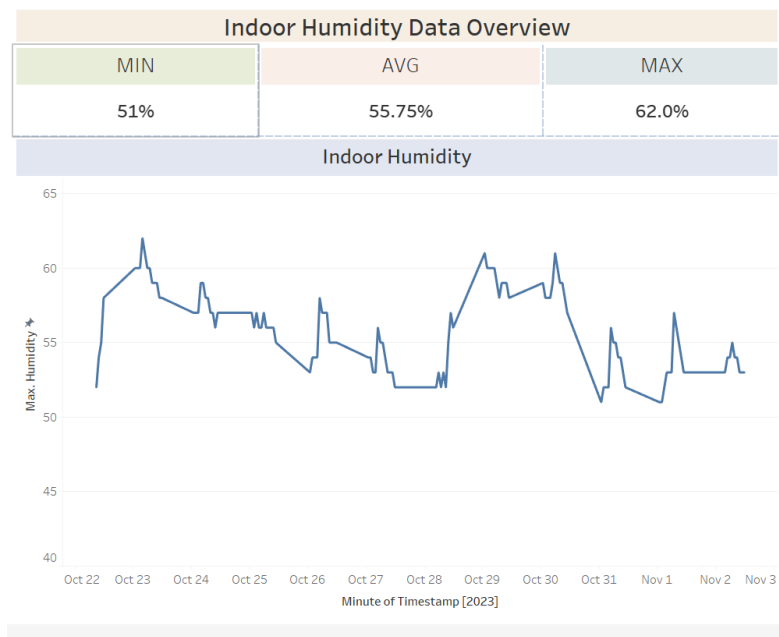


Figure 62 - Indoor humidity data Dashboard

7.3.4 Contact data Dashboard

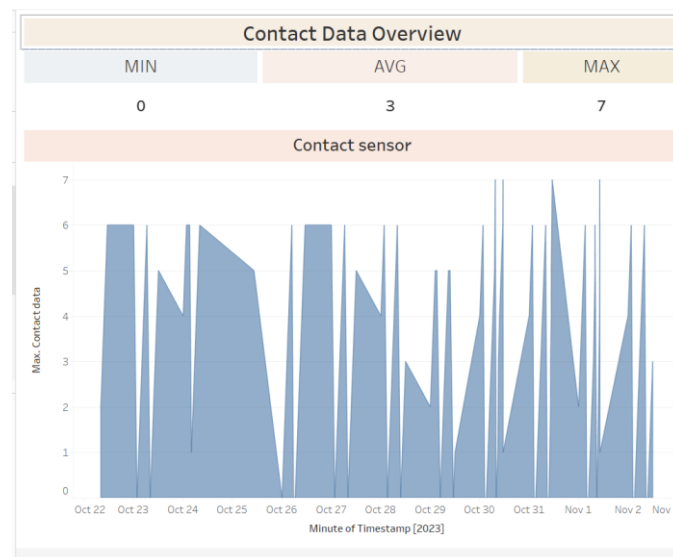


Figure 63 - Contact data Dashboard

7.3.5 Summary Dashboard

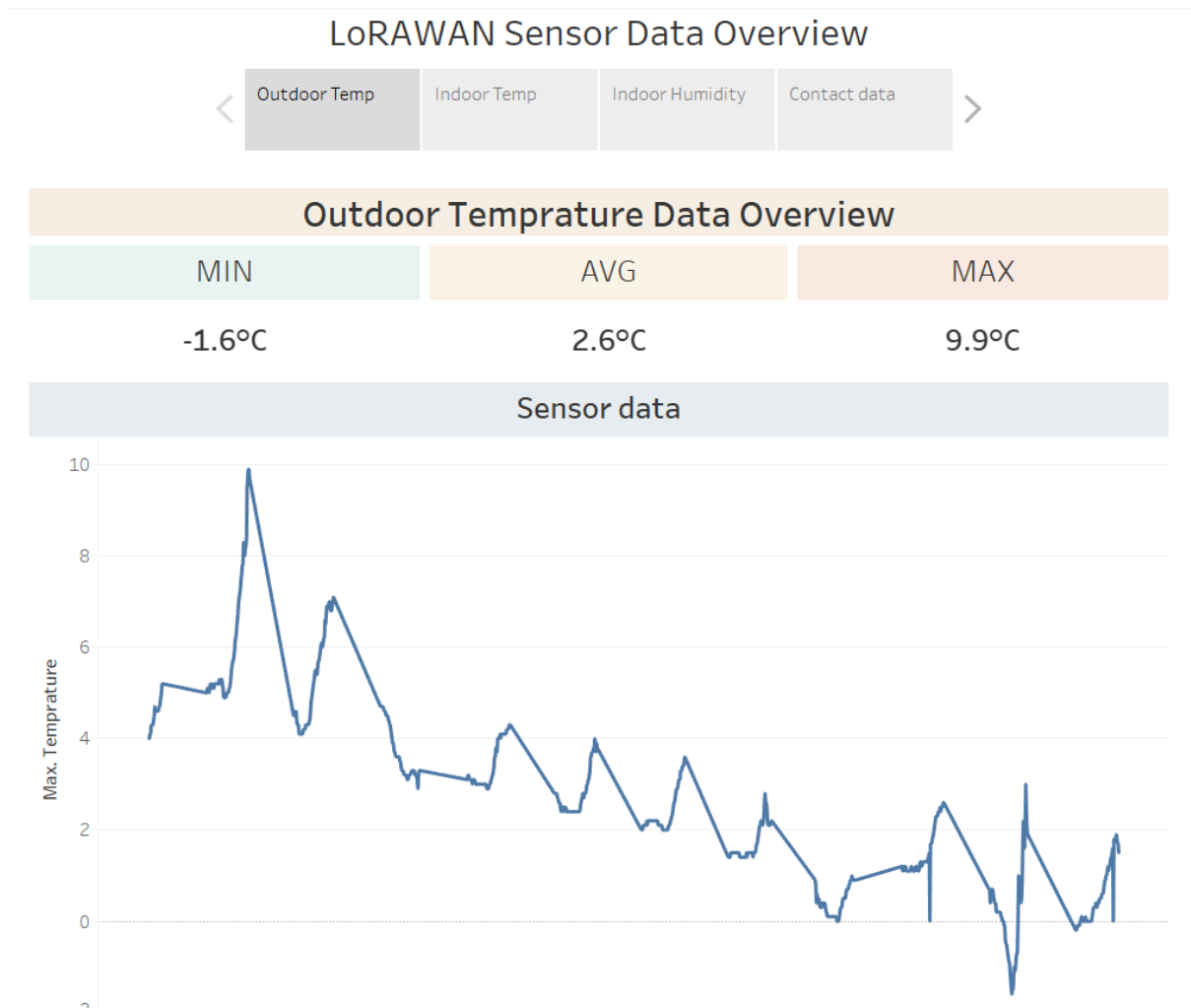


Figure 64 - Summary Dashboard

8 Discussion

For the purpose of investigating and analyzing LoRaWAN sensors, the placement of the sensors must be done correctly. The use of a gateway is crucial in certain locations where the LoRaWAN network coverage is inadequate since it ensures that the sensors will connect reliably.

An easy and secure connection to the LoRaWAN sensors can be established by utilizing the capabilities of various platforms. Furthermore, the platform's features can be helpful for troubleshooting.

The nature of the HTTP request trigger is characterized by its distinctiveness and complexity, making it challenging to predict it. Nevertheless, the release of the URL could potentially result in substantial consequences in terms of a data breach. To guarantee the preservation of data integrity and confidentiality in commercial and production settings, it is crucial to integrate a resilient and secure HTTP trigger solution throughout the developmental stage of the datalogging platform.

The initial development of the datalogging platform took place within the Pipedream framework, with the primary objective being investigation rather than programming. The potential for implementation as a production application lies in the ability to store data in a cloud database. The utilization of cloud database services such as AWS or Azure can be considered for this purpose. Currently, the datalogging application is functioning on a JSON integrated with a Google spreadsheet, acting as a demonstration of its feasibility. Nevertheless, in order to achieve further advancements, it is imperative to migrate the program to a cloud platform. By doing so, the application will be able to leverage the capabilities of cloud databases, hence guaranteeing consistent and uninterrupted stability.

The restriction imposed by Pipedream, which limits the number of records per pull request to a certain limit, poses challenges when seeking to access large amounts of historical data. The existence of this limitation necessitates the incorporation of iteration and looping methods, leading to significant delays within the monitoring program. One potential method for improving the response time of the monitoring application involves raising the previously stated threshold.

There are several frameworks, like ASP.NET, Laravel and PHP, that are accessible for the construction of web applications. But more than programming it manually in web applications using an inbuilt data analysis software as PowerBI or Tableau is easier for the monitoring the Data. The selection of the Tableau Server for data analysis is more modern and powerful way of representing the data for clients and customers. The Dashboards and the widgets in the Tableau is more efficient and user friendly for the users

The monitoring application can enhance its functionalities by integrating machine learning algorithms with past data in order to make predictions regarding temperature and relative humidity. Furthermore, the user interface can be improved through the utilization of pictorial representations to visually represent meteorological conditions.

9 Conclusion

In conclusion, we have achieved the objectives outlined at the beginning of the project, each contributing to understanding and successful installation of LoRaWAN sensors and implementation of the LoRaWAN-based datalogging and monitoring system. Our efforts encompassed gaining a profound overview of LoRaWAN and other pertinent protocols in the IoT domain, specifically within the context of this project. Moreover, it has taken the time to thoroughly grasp the complexities of the Altibox LoRaWAN Infrastructure, laying a robust groundwork for our subsequent steps.

The core functionality of logging and monitoring data using LoRaWAN sensors has been realized, aligning with the project's fundamental needs. The collaborative tools of Microsoft Teams and WhatsApp were effectively utilized throughout the project's planning and development stages, fostering a cohesive and efficient working environment.

In adherence to the project requirements, the system has been documented in the form of a detailed technical report.

With these accomplishments, not only the specific objectives are met but also a robust foundation for future developments and implementations in the realm of IoT and LoRaWAN-based systems has been established.

References

- [1] “Implementing Low-Power Wide-Area Network (LPWAN) Solutions with AWS IoT - Implementing Low-Power Wide-Area Network (LPWAN) Solutions with AWS IoT.” Accessed: Nov. 16, 2023. [Online]. Available: <https://docs.aws.amazon.com/whitepapers/latest/implementing-lpwan-solutions-with-aws/implementing-lpwan-solutions-with-aws.html>
- [2] “LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL.” Accessed: Nov. 13, 2023. [Online]. Available: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>
- [3] “LoRaWAN® Specification v1.1.” Accessed: Nov. 13, 2023. [Online]. Available: <https://resources.lora-alliance.org/technical-specifications/lorawan-specification-v1-1>
- [4] “LoRaWAN Security Whitepaper,” Feb. 02, 2017. Accessed: Nov. 13, 2023. [Online]. Available: <https://resources.lora-alliance.org/whitepapers/lorawan-security-whitepaper>
- [5] “Introduction - ChirpStack open-source LoRaWAN® Network Server documentation.” Accessed: Nov. 13, 2023. [Online]. Available: <https://www.chirpstack.io/docs/>
- [6] “Device Classes,” The Things Network. Accessed: Nov. 03, 2023. [Online]. Available: <https://www.thethingsnetwork.org/docs/lorawan/classes/>
- [7] “Sigfox oG Technology,” Sigfox 0G Technology. Accessed: Nov. 02, 2023. [Online]. Available: <https://www.sigfox.com/>
- [8] “An Introduction to Sigfox Technology – Basics, Architecture and Security Features.” Accessed: Nov. 03, 2023. [Online]. Available: <https://circuitdigest.com/article/what-is-sigfox-basics-architecture-and-security-features>
- [9] Shahin Farahani, [PDF] *ZigBee Wireless Networks and Transceivers*. Accessed: Nov. 03, 2023. [Online]. Available: https://www.perlego.com/book/1813705/zigbee-wireless-networks-and-transceivers-pdf?utm_source=google&utm_medium=cpc&campaignid=17490270403&adgroupid=140283297000&gclid=CjwKCAjwkY2qBhBDEiwAoQXK5TyDLx_Up4SaB1t7YbssiwdFujr6UEuQC9PnyazhSp-BSxYgMW4OixoCBawQAvD_BwE
- [10] “CSA-IOT,” CSA-IOT. Accessed: Nov. 03, 2023. [Online]. Available: <https://www.zigbee.org/>
- [11] “GSMA | Internet of Things.” Accessed: Nov. 03, 2023. [Online]. Available: <https://www.gsma.com/iot/>
- [12] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, and R. Schober, “Massive Access for 5G and Beyond,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 615–637, Mar. 2021, doi: 10.1109/JSAC.2020.3019724.
- [13] “Whitepaper on cellular IoT for industry digitization.” Accessed: Nov. 03, 2023. [Online]. Available: <https://www.ericsson.com/en/reports-and-papers/white-papers/cellular-iot-evolution-for-industry-digitalization>
- [14] “COMFORT: IoT temperature and humidity sensor | Adeunis.” Accessed: Nov. 16, 2023. [Online]. Available: <https://www.adeunis.com/en/produit/comfort-temperature-humidity-2/>

- [15] “TEMP: IoT temperature reading | Sigfox, LoRaWAN | Adeunis.” Accessed: Nov. 16, 2023. [Online]. Available: <https://www.adeunis.com/en/produit/temp-temperature/>
- [16] “Detect an opening/closing | CONTACT SENSOR | Adeunis.” Accessed: Nov. 16, 2023. [Online]. Available: <https://www.adeunis.com/en/produit/contact-sensor-detect-an-opening/>
- [17] “Test IoT network coverage: Sigfox, LoRaWAN | Adeunis.” Accessed: Nov. 16, 2023. [Online]. Available: <https://www.adeunis.com/en/produit/ftd-network-tester/>
- [18] “Unleash thePower of IoT Data.” Accessed: Nov. 13, 2023. [Online]. Available: <https://dimensionfour.io/index>
- [19] “Explore Popular APIs, Triggers, Actions and More - Pipedream.” Accessed: Nov. 16, 2023. [Online]. Available: <https://pipedream.com/explore>
- [20] “Adeunis Codec - Decoder.” Accessed: Nov. 16, 2023. [Online]. Available: <https://codec-adeunis.com/decoder>
- [21] “A new home page for Tableau Server and Tableau Online makes browsing personal (now in beta).” Accessed: Nov. 16, 2023. [Online]. Available: <https://www.tableau.com/blog/new-homepage-tableau-server-and-tableau-online-makes-browsing-personal-105237>