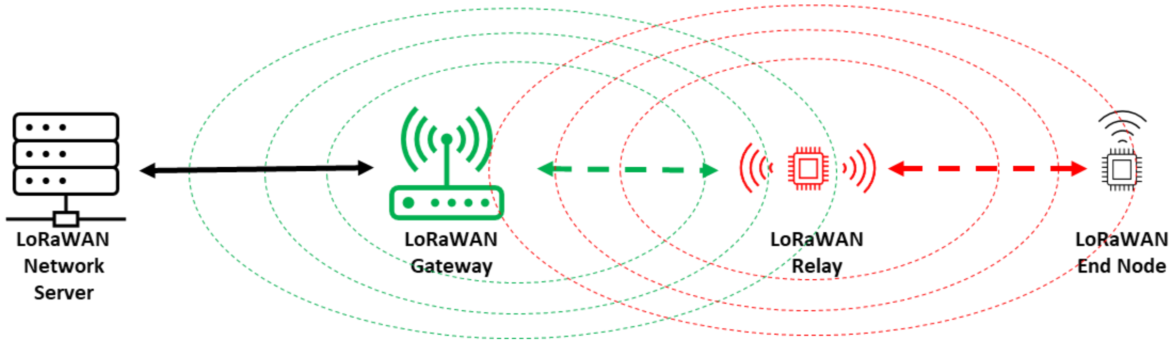
University of
South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2024

Industrial IT and Automation

# Using relays for utilizing and extending LoRaWAN networks with the use of Altibox LoRaWAN infrastructure for internet of things



Even Tviberg Hope

Faculty of Technology, Natural sciences and Maritime Sciences
Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

**Course**: FMH606 Master's Thesis, 2024

**Title**: Using relays for utilizing and extending LoRaWAN networks with the use of Altibox LoRaWAN infrastructure for internet of things

**Number of pages**: 90

**Keywords**: Internet of things, LoRaWAN, LoRaWAN relay, Altibox, Porsgrunn Kommune, LoRa Basics Modem

**Student:** Even Tviberg Hope

**Supervisor:** Hans-Petter Halvorsen, Saba Mylvaganam

**External partners:** Altibox, Porsgrunn Kommune

**Summary:**

LoRaWAN is an open-source protocol tailored for IoT applications, employing LoRa radio modulation in unlicensed frequency bands to achieve extensive range with minimal power consumption. Despite LoRa's remarkable penetration capabilities, there are scenarios where signal transmission between two transceivers becomes too unreliable, such as sensors situated deep indoors, in shadowed urban areas, or underground. To address such challenges, the LoRa Alliance recently introduced a new relay feature. The relay, as a hardware component like any LoRaWAN end node, acts as an intermediary message forwarder between LoRaWAN devices and a gateway.

The objective of this thesis was to investigate the application of the LoRaWAN relay feature in expanding LoRaWAN networks, leveraging the Altibox LoRaWAN infrastructure. Through a comprehensive review of the LoRaWAN protocol and an in-depth study of the LoRaWAN Relay Specification, a solid foundation was laid for conducting this evaluation. This involved utilizing a relay evaluation kit to test and assess the functionality of the relay feature.

As the results demonstrate, the relay significantly enhanced the radio coverage. However, findings regarding stability and occasional dropouts warrant additional investigation.

The final part of this thesis involves the development of a relay enabled end node, using a Nucleo development board combined with a LoRa radio shield and an analogue moisture sensor. The program was initiated using an example application sourced from Semtech's LBM software library.

*The University of South-Eastern Norway takes no responsibility for the results and conclusions in this student report.*

# Preface

My curiosity towards the internet of things was one of the main reasons why I chose to apply for the Master's program in Industrial IT and Automation at the University of South-Eastern Norway. After tinkering with smaller home automation projects for several years and as IoT started to emerge in the offshore industry with terms like API and OPC suddenly became common, I realised that it was time to return to the classroom. When I submitted the application, I had limited understanding of what it would entail to juggle part-time studies with work and family commitments, and I am grateful that I did not fully anticipate the challenges.

I had the opportunity to get familiar with LoRaWAN in connection with the project assignment last semester, involving development of a mobile application for monitoring of LoRaWAN sensors. It gave me a slight head start, yet I soon recognized the necessity for a more thorough review of the LoRaWAN standard than what had been explored in the autumn semester. The focus at that time primarily revolved around the data flow from the Altibox environment to the database and subsequently to the mobile application, which consumed the majority of our time. Eventually, I also realized that the relay functionality was not widely applied yet, and that Semtech's reference implementation was only an experimental released. It also turned out that the end node in the relay evaluation kit only transmitted an incremental counter. While this was irrelevant during relay functionality testing, it was significantly relevant when considering implementing one of the defined use cases.

The absence of commercially available end nodes compatible with the relay, coupled with the limitations of the relay evaluation kit, prompted me to reconsider the approach for implementing a use case. I decided to develop a relay-enabled end node using a combination of a development board and LoRa radio shield, complemented by a basic analogue moisture sensor for data input. This might not completely align with the task objective, but it was the best I could come up with given the circumstances.

I want to express my gratitude to my supervisors at USN, Hans-Petter Halvorsen and Saba Mylvaganam, for their continuous support and guidance. I also appreciate the presence and contributions of my external partners, Altibox and Porsgrunn Kommune, in our bi-weekly meetings, and a special thank you to Morten Hovland from Altibox for following up on all my questions, even during weekends.

To my employer Odfjell Technology, I am grateful for the time off provided to finalize this thesis, especially during a busy period for the company. I appreciate Hallvard Atle Bjånesøy for sharing ideas on MODU related IoT use cases, Christian Johnsen for providing detailed office construction information, and finally Pål from the IT department for his assistance in establishing the necessary network configuration for the gateway backhaul.

Throughout my journey of part-time studies, full-time work, and raising a family, I owe immense gratitude, and perhaps a bit of apology, to my wife. Not only for the past six months but over the last four years, she has steadfastly supported me, ensuring that I could dedicate countless evenings and weekends to my studies, often at the expense of our family time together.

Porsgrunn, 14/05/2024

Even Tviberg Hope

# Contents

# Nomenclature

| Abbreviation | Definition | Explanation |
|---|---|---|
| **ADR** | Adaptive Data Rate | A function for optimising the parameters of a LoRaWAN transmission. |
| **AES** | Advanced Encryption Standard | An algorithm for encrypting data. |
| **API** | Application Programming Interface | A set of definitions and protocols established for enabling interaction with a computer program. |
| **ATEX** | ATmosphères EXplosibles | Two EU directives which establish minimum safety requirements for workplaces and equipment used in potentially explosive atmospheres. |
| **CAD** | Channel Activity Detection | Low power mode of a LoRa radio IC, where the radio scans for activity. |
| **CLI** | Command Line Interface | A text-based interface with a defined set of commands used to communicate with a computer program. |
| **CSS** | Chirp Spread Spectrum | A RF modulation technique where linear frequency modulated chirp pulses are used to convey a specific symbol. |
| **CSV** | Comma Separated Vaues | A file containing information of which is separated by comma, semicolon or similar. |
| **DR** | Data Rate | Number of bits transmitted per time unit |
| **EIRP** | Equivalent Isotopically Radiated Power | The radiated output power referenced to an isotropic antenna radiating power equally in all directions. |
| **ESP** | Estimated Signal Power | An estimation of the real received signal strength of a desired signal considering the impact of background noise [1] |
| **FSK** | Frequency Shift Keying | A RF modulation technique where the frequency is used to convey a specific symbol. |
| **FTD** | Field Test Device | A device specifically designed for mapping radio coverage. |
| **GW** | Gateway | A device that connects the LoRaWAN and internet protocols, receiving and forwarding LoRaWAN end node data packets to the network server. |
| **HAL** | Hardware Abstraction Layer | A set of functions in a software program that facilitate the interaction with hardware resources through programming interfaces. |

| | | |
|---|---|---|
| **IC** | Integrated Circuit | A small electronic circuit consisting of many interconnected components. |
| **IEC** | International Electrotechnical Commission | An international standardization organization that designs and publishes standards for all types of electrical and electronic technologies |
| **IoT** | Internet of Things | Refers to a network of interconnected devices embedded with sensors and software. |
| **IP** | Internet Protocol / Ingress Protection | A set of requirements for addressing and routing data over the Internet / IEC standard for rating the grade of protection an enclosure can provide against ingress of dust and liquids. |
| **ISM** | Industrial Scientific and Medical | A radio spectrum reserved regionally for industrial, scientific, and medical devices. |
| **LBM** | LoRa Basic Modem | A full implementation of LoRaWAN stack based on TS001-v1.0.4 and RP002-v1.0.3, developed by Semtech. |
| **LNS** | LoRaWAN Network Server | A software that runs on a server managing the LoRaWAN network. |
| **LoRa** | Long Range | RF modulation technique for low-power, wide area networks. |
| **LoRaWAN** | Long Range Wide Area Network | A wireless communication protocol designed for long-range, low-power IoT devices. |
| **LOS** | Line Of Sight | Is the direct path from a transmitter to the receiver and the obstructions that may fall in that path [2]. |
| **LPWAN** | Low Power Wide Area Network | A grouping of networks featuring wireless low power and long-range communication at low bit rates. |
| **MAC** | Media Access Control | In the context of this thesis, it is the software layer that defines how devices use the LoRa radio hardware and specifies the format of the messages. |
| **MIC** | Message Integrity Code | A key used to validate the integrity of each LoRaWAN message [3] |
| **MODU** | Mobile Offshore Drilling Unit | A floating, often self-propelled, maritime vessel which is used to drill oil wells offshore. |
| **MCU** | Microcontroller Unit | A small computer on a single IC. |

| | | |
|---|---|---|
| **NC** | No Connection | Term used in tables to define no connection between the LoRaWAN node and the gateway |
| **OSI** | Open Systems Interconnection Model | A conceptual framework used to describe the functions of a networking system |
| **PCB** | Printed Circuit Board | A board made of non-conducting material with conductive lines printed on it to form circuits, used to connect electronic components. |
| **PER** | Packet Error Rate | A percentage ratio of missing uplink packets to total uplink packets that defines the quality of uplink communication between the device and the network [1] |
| **PoE** | Power over Ethernet | Means of powering a device from the suppling switch over the ethernet cable. |
| **RAM** | Random Access Memory | Hardware memory part used for temporary storing and quick access to stored values. |
| **RF** | Radio Frequency | The frequency spectrum of radio waves transmitted from a radio device |
| **RSSI** | Received Signal Strength Indicator | A number that determines the total received signal strength within a channel bandwidth summing up the useful signal, the interference, and the background noise [1] |
| **SF** | Spreading Factor | A number that determines the LoRa data transmission rate [1] |
| **SNR** | Signal-to-Noise Ratio | A number that determines the quality of the reception through the ratio between the received signal strength of the useful signal, and the signal strength of the interference added to  background noise [1] |
| **TPW** | ThingPark Wireless | The LoRaWAN Network Server used by Altibox. |
| **USB** | Universal Serial Bus | A standard connection socket for connecting peripherals to a computer. |
| **VPN** | Virtual Private Network | A method for creating a secure connection, known as a tunnel, between two points over an open network. |
| **WOR** | Wake On Radio | A specific frame sent between an end device and a relay with purpose of waking the relay from sleep and inform about coming LoRaWAN uplink. |

# 1 Introduction

This thesis explores the use of LoRaWAN relay feature for extending LoRaWAN networks by utilizing Altibox LoRaWAN infrastructure for the internet of things. This introductory chapter begins with a discussion on the project background, followed by an overview of the objectives. Thereafter follows an overview of the report outline, providing an overview of the topics covered in the subsequent chapters.

## 1.1 Background

Altibox AS, part of the Lyse group, is a Norwegian internet service provider, offering broadband, IPTV, and IP telephony through their network of fiber-optic cables. Since 2018, Altibox has branched into the IoT field, introducing a wireless IoT network which leverages the existing fiber grid as backhaul. This strategic move has enabled Altibox AS to rapidly extend the wireless coverage, now encompassing over 100 municipalities and reaching more than 1 million households [4].

The wireless network utilizes LoRaWAN technology, a purpose-built, open protocol designed specifically for IoT applications [1]. LoRa RF modulation uses the unlicensed ISM frequency bands, which combined with its long range and minimal power consumption makes this protocol ideal for IoT applications, particularly those relying on battery-powered sensors.

While both indoor and outdoor LoRaWAN networks typically offer impressive range and penetration through each installed gateway, there are instances where some remote or distorted sensors may face connectivity challenges. In such scenarios, establishing a new base station might be impractical or too costly. Recently, the LoRa Alliance introduced a new relay feature tailored to address such cases. The relay forwards messages from other LoRaWAN devices to gateways within its reach, effectively increasing the network coverage. This functionality can be implemented into the firmware of normal end nodes or applied to a dedicated node for this purpose, making this a much more cost-effective approach to extending the network coverage to challenging locations [5]. The relay can be battery operated and connects to the existing LoRaWAN network such as any other LoRaWAN device.

## 1.2 Objective

The objective for this thesis is to explore the new relay feature, to gain experience with the functionality and to evaluate and document its performance. For this purpose, an evaluation kit containing a relay and a relay enabled end node has been provided by Altibox. Additionally, Altibox has provided a LoRaWAN Gateway, a Field Test Device and access to their software for network management, all to be used for the testing and evaluation of the relay functionality.

As a basis for evaluating the relay's performance, it is necessary to study the LoRaWAN protocol and the LoRa Alliance's Relay Specification. Additionally, the Altibox infrastructure are to be reviewed, to gain insights into the backbone of the LoRaWAN network.

Potential use cases for the relay feature shall be explored and discussed, particularly those relevant to Altibox, Porsgrunn Municipality, and Odfjell Drilling's mobile offshore drilling units. Subsequently, one of the identified use cases will be selected, developed, and tested using the Altibox LoRaWAN infrastructure and relay feature.

## 1.3 Report outline

To gain an understanding of the LoRaWAN protocol, the report begins with an introduction to this technology, covering key aspects such as LoRa modulation, the physical layer of the communication protocol, network architecture and components, and the integration of security measures. Subsequently, the report provides an overview of available open-source software libraries, focusing on a detailed introduction of the LoRa Basics Modem. Additionally, a comparison of LoRaWAN with other IoT and LPWAN protocols is presented.

The following section delves into the newly introduced LoRaWAN relay feature, describing its functionality based on the LoRaWAN Relay Specification. Special attention is given to explaining the wake on radio protocol initiated by an end node to activate the relay prior to transmitting an uplink. This includes an explanation of the wake on radio packet structure and the supplementary security measures introduced by this standard. Additionally, the experimental release of the LoRa Basics Modem, incorporating the proposed implementation of the relay feature, will be detailed.

The upcoming chapter will introduce the Altibox LoRaWAN infrastructure, followed by an overview of ThingPark Wireless, the LoRaWAN network server employed by Altibox for network management. ThingPark Wireless provides various useful applications, where Device Manager and Wireless Logger have been utilized in this project and are thus described in greater detail.

An in-depth analysis of relay message sequences is presented using the Wireless Logger application, focusing on relayed join procedures, relayed uplink messages, and relayed LoRaWAN network server MAC commands. These sequences are then compared to the requirements outlined in the previous chapters.

A part of the project objective involved exploring use cases for the LoRaWAN relay feature, especially those relevant to Altibox, Porsgrunn Municipality, and mobile offshore drilling units. The proposed use cases are introduced in chapter 7.

Chapter 8 to 11 describes the test performed to assess the relay evaluation kit. Starting with an overview of the hardware components used in the test, followed by a description of the test environment and setup. Chapter 10 details the results from the mapping of coverage provided by the indoor gateway, while the succeeding chapter describes the outcomes of integrating a relay to extend the coverage.

Due to limited compatibility of end nodes with the relay TX feature, the decision was made to explore the development of a custom end node using a development board, incorporating a radio shield and an analogue sensor for demonstration purposes. Chapter 12 presents the outcomes of this effort, documenting the modification and adaptation of an example application from the relay enabled LoRaWAN Basics Modem library to serve as an end node for soil moisture measurement.

In the discussion chapter, the obtained results are examined, and potential areas for future work are proposed. Ultimately, the report concludes with a summary in the concluding chapter, assessing whether the task objectives have been met.

# 2 LoRaWAN

LoRaWAN is an open source network protocol built on top of the LoRa wireless modulation [6]. LoRa modulation composes the first layer of the OSI model, while LoRaWAN composes the second layer. This means that LoRaWAN handles the communication in the session layer, frames in the network layer and packets in the data link layer, while LoRa handles the packages in the physical layer transforming the data bits to radio signals.

In simple terms, LoRaWAN is a software standard that outlines how data packets are structured and handled by both the network and end devices. It is designed to facilitate long-range communication at a low bit rate and power consumption, which is particularly useful for transmitting small-size sensor data payloads [7].

The protocol is standardized and maintained by the LoRa Alliance, a non-profit global association which has been operational since 2015 [8]. The primary goal of the alliance is to enable large scale IoT deployments, by standardising and promoting the LoRaWAN technology. They believe that standardizing on a strong and growing ecosystem is the key for volume deployments of such networks. The associations members span from equipment manufacturers, system integrators and network operators to the academia and cities/Governments. All collaborating to develop and deploy IoT technology and specifically LoRaWAN worldwide.

## 2.1 Characteristics of LoRaWAN

The LoRaWAN protocol offers several key benefits, including long-range connectivity, extended sensor battery life, low data rates, and support for unlicensed ISM bands. The achievable range with LoRa radio frequency modulation depends heavily on the device's location. For instance, in urban areas, a range of up to 5km can be achieved, while in areas with a clearer LOS, distances of up to 15km are possible [6].

To illustrate the technology's potential, a remarkable test was conducted during the Things Virtual Conference in 2020 [9]. In this test, a LoRaWAN sensor was affixed to a high-altitude balloon filled with helium. At 38 km altitude, a gateway 832km away picked up the signal, despite the transmission power being as low as 25mW. This achievement set a world record at the time, highlighting the technology's capabilities. Subsequently, in 2023, the record was broken by a different sensor application. A tracking device mounted on a fishing boat and its buoys managed to achieve an impressive distance of 1336km at sea level [10]. These records underscore the significant impact of optimizing antenna design and device height on transmission range, suggesting that the estimated specifications provided earlier may be conservative.

What is particularly remarkable about these distances is that they are achieved using battery-powered sensors with remarkably low power consumption. In some cases, these sensors can operate for up to 10 years on a single battery triple A batch. Achieving such efficiency is made possible by transmitting small data packets infrequently and by employing power-saving mechanisms that allow most components of the device to enter sleep mode when not in use [6].

In the EU, where the 868MHz band is utilized for LoRaWAN, the data rate using LoRa modulation typically ranges from 0.3 to 5.5 kbps, with a bandwidth of 125kHz. This speed is contingent on the active spreading factor, which is elaborated upon in the subsequent section.

LoRaWAN operates within unlicensed Industrial, Scientific, and Medical (ISM) bands, reserved for these purposes. Frequencies allocated for ISM use vary across regions. In the EU, the 868MHz band, spanning from 863 to 870MHz, is used by LoRaWAN. Being unlicensed permits usage of the band without necessitating payment to a network operator.

Unlike devices paired with specific gateways, such as those employing Wi-Fi, LoRaWAN devices can freely roam between gateways within the same network. However, adherence to ISM band requirements is essential. Frequencies utilized vary between regions, implying that a sensor designed for the EU868 band will not function in regions like the US, where the US915 plan is employed.

As an alternative to GPS tracking, which consumes substantial power, LoRaWAN offers an alternative method for device geolocation based on triangulation principles. This approach requires the signal to be received by at least three gateways with known positions [6]. Using received signal strength data from these gateways, an algorithm calculates the device's approximate location. While not as precise as GPS, this method contributes to minimizing device power consumption.

### 2.1.1 Limitations of LoRaWAN

Before deciding to use LoRaWAN technology, it is important to consider some of the limitations. These include constraints on payload size and transmission speed. LoRaWAN is particularly well-suited for scenarios involving small sensor payloads sampled at infrequent intervals, such as a couple of samples per hour. However, it is not suitable for real-time applications that demand frequent data sampling or constant connections, such as phone calls.

Additionally, regional-specific regulations can pose limitations based on the application's requirements, especially if the end node needs to roam between regions governed by different ISM band regulations.

## 2.2 LoRa modulation

LoRa stands for Long Range and is a low-power wide-area network modulation technology developed and owned by Semtech [7], which also is one of the founding members of the LoRa Alliance. It is a wireless modulation technique that is derived from Chirp Spread Spectrum (CSS) technology, which has been used for military and space communication for decades due to its robustness towards interference and long range [11].

A chirp in this sense, is a sine wave where the frequency increases or decreases linearly with time [12]. If the frequency increases with time it is called an up-chirp, while if it decreases with time, it is called a down-chirp. The chirps are used as a carrier, and the signal data is encoded on to these where the resulting pattern is called a symbol. The symbol is representing one or more bits of data, where the spreading factor defines how many bits each symbol shall contain.

In his paper, L. Vangelista [13] describes LoRa modulation as a Frequency Shift Chirp Modulation. By this he means that it can be seen as a combination of FSK and CSS. Instead of just changing the frequency, as is done in FSK, a chirp is used as a basis for each modulated symbol. Like FSK, the start frequency of the chirp is used as a basis for defining each symbol. He also states that the number of symbols available to transmit is equal to $2^{SF}$, so depending on the active SF, which in the EU is a number between 7 and 12, this defines the number of symbols available for encoding the information.

The chirp principle can be plotted in time or frequency domain as shown in Figure 2.1. It is best depicted in the frequency versus time plot, where it will be represented by a straight line. A signal transmission is always started by a preamble to alert of the upcoming uplink message and for synchronization. The preamble consists of defined number of up-chirps followed by two down-chirps for time synchronization [14]. Thereafter, the signal payload is transmitted within a given period.

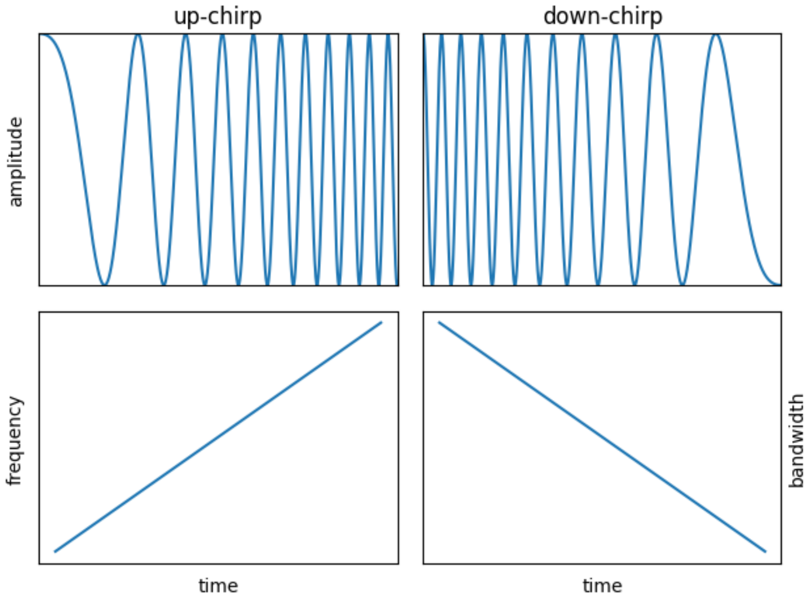


Figure 2.1 LoRa chirps represented in both time and frequency domain, up-chirp (increasing frequency) to the left and down-chirp (decreasing frequency) to the right

The benefit gained from this modulation technique is the ability to decode a symbol that is up to 20 dB below the noise floor (SNR below 0dB) when using SF12 (slow data rates) [15], [16]. Table 2.1 shows the acceptable SNR link between SF and SNR. This ability distinguishes LoRa from other modulation techniques, enabling lower transmitter power levels without compromising signal data rates or link budgets. All LoRa radios uses Semtech proprietary transceiver chips for handling the wireless communication. There are two main groups of these chips, the ones used for end nodes and ones used for gateways.

### 2.2.1 Frequency bands used in the EU

Even though long distances can be achieved, LoRa, like any other radio transmission technology, is susceptible to reduced signal strength when signal is blocked by physical barriers such as for instance a building made of concrete and steel. Utilizing sub-gigahertz frequency bands contributes to a longer range than e.g. technologies based on Wi-Fi.

LoRa operates on the ISM bands. These bands vary from region to region and in the EU the 868 MHz band is used. Which bands that can be used per region and which specific requirements that are placed upon these are detailed in RP002-1.0.4 Regional Parameters [17], updated by the LoRa Alliance, yet it is governed by the regional authorities.

As specified in RP002-1.0.4, to comply with the EU868MHz, there are three default channel frequencies that shall be implemented in every end device. The other network channels can be chosen by the network operators, within the 863-870 MHz range. The bandwidth for the default channels is 125kHz.

Since the ISM bands can be used by anyone, a set of rules have been established for the usage to avoid overflowing the frequency band. In Europe, it is the European Telecommunications Standards Institute (ETSI) that creates and maintains the standards which the regulatory authorities adhere to. Specific for the EU band is the requirement to duty cycle, meaning how much percent of the time during which a device can transmit data within a given time window. In the standard ETSI EN300.220 V3.2.1 (2018-06) Annex B – Table B.1 the duty cycle requirement is set to 1% per day for the channels used by LoRa. Table 2.1 shows how the SF influences the message limit given a 1% duty cycle for an uplink package transmitted with 125kHz bandwidth and a 10bit payload. The ETSI also regulates the maximum effective radiated power, which is 25mW for these bands.

### 2.2.2 Spreading factor

Radio communication can be compared to a conversation between two individuals [14]. When they are close, they can speak quickly and quietly. However, as the distance between them grows, along with the introduction of background noise, they must speak louder to ensure the message is received clearly. Eventually, if the distance becomes too great, comprehension becomes difficult. This analogy applies to radio communication, where increasing the distance between the transmitter and the receiver leads to signal attenuation, making it challenging for the receiver to distinguish the transmitted signal from the noise.

To address increased distance, LoRa adjusts the transmission rate by manipulating the spreading factor. The spreading factor in LoRa technology regulates the chirp rate [18], determining the rate at which the chirp moves from the lowest to highest frequency within the assigned bandwidth. It governs the data transmission rate. LoRa has six spreading factors, ranging from SF7 to SF12, with SF7 providing the highest transmission rate. By increasing the spreading factor as the signal weakens, LoRa decelerates the transmission rate, facilitating easier interpretation and differentiation of the signal from noise. However, a direct consequence of longer transmissions is the reduced battery life of the transmitting node [19].

There is a linear relationship between the spreading factors: increasing one step halves the transmission rate, thereby doubling the airtime for the same payload. Figure 2.2 illustrates this concept, plotting the carrier chirp pulse of a 125kHz bandwidth against time for each of the six spreading factors.

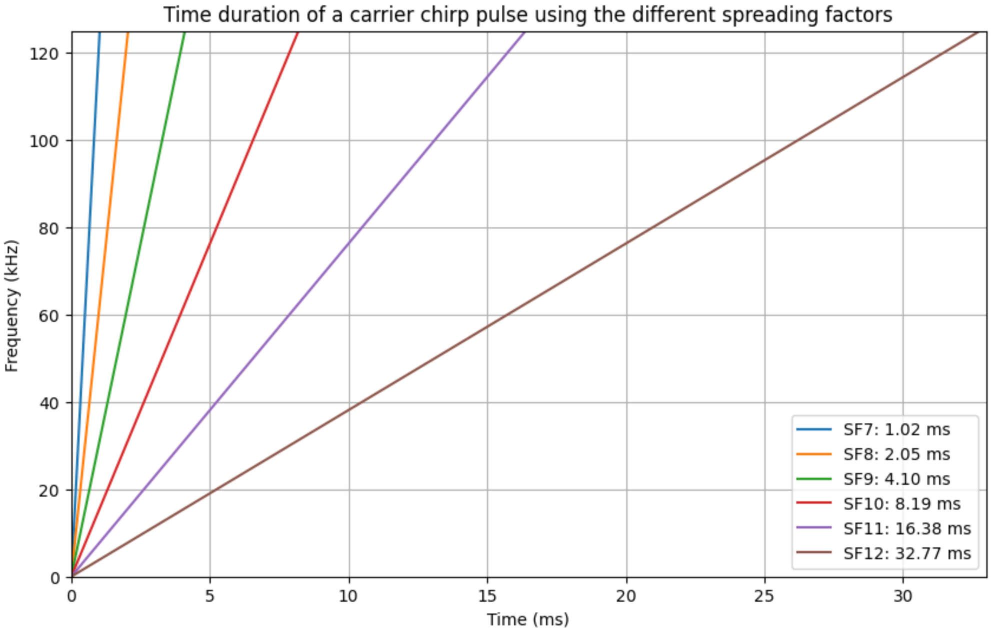


Figure 2.2 Frequency vs. time plot of a carrier chirp pulse using Spreading Factor 7 to 12

The plot visualizes the how the active spreading factor influences the speed of a transmission. The sweep time or symbol duration, $Ts$, at this SF can be calculated as shown in Eq. 2-1. Increasing the SF to 12 significantly extends the sweep time, as evident from Eq. 2-2.

$$Ts = \frac{2^{SF}}{BW} = \frac{2^7}{125000} = 1.024ms \tag{Eq. 2-1}$$

$$Ts = \frac{2^{SF}}{BW} = \frac{2^{12}}{125000} = 32.768ms \tag{Eq. 2-2}$$

The SF number corresponds to the number of bits represented by each symbol [20]. For instance, with SF 7, each symbol represents 7 bits, while with SF 8, each symbol represents 8 bits. The spreading factors are orthogonal. This ensures that signals modulated with different spreading factors and transmitted on the same frequency channel simultaneously will not interfere with each other [6]. They will simply be regarded as noise in the demodulation step.

In Table 2.1 the relation between DR, max payload size, SF, Bit Rate, time on air, duty cycle, message limit per hour, SNR limit and receiver sensitivity is shown for an uplink package transmitted with 125kHz bandwidth and a 23bit payload. The SNR is based on a Semtech SX1261 radio module, and the receiver sensitivity is calculated based on this.

Table 2.1 Relation between SF, bit rate, time on air, duty cycle, message limit, SNR limit and receiver sensitivity (UL @ BW 125 kHz, 23bit payload) given EU868 regional parameters and SX1261 radio module [6], [17, Tbls. 8, 12], [21], [22]

| DR | Max payload size [bytes] | SF | Bit Rate [bps] | Time on Air [ms] | Duty Cyle (1%) [sec] | Message limit per hour | SNR Limit [dB] | Receiver sensitivity [dBm] |
|---|---|---|---|---|---|---|---|---|
| **0** | 51 | 12 | 250 | 1 974,3 | 197,4 | 18 | -20 | -137 |
| **1** | 51 | 11 | 440 | 987,1 | 98,7 | 36 | -17,5 | -135 |
| **2** | 51 | 10 | 980 | 493,6 | 49,4 | 72 | -15 | -132 |
| **3** | 115 | 9 | 1760 | 267,3 | 26,7 | 134 | -12,5 | -130 |
| **4** | 222 | 8 | 3125 | 143,9 | 14,4 | 250 | -10 | -127 |
| **5** | 222 | 7 | 5470 | 77,1 | 7,7 | 467 | -7,5 | -125 |

## 2.3 LoRaWAN network architecture

The LoRaWAN network operates on the principle of a star of stars, with three types of networks: public, private, and open. Public networks, managed by telecommunication operators, utilize multiple base stations or gateways to ensure wide coverage and typically involve a fee structure based on the number of end devices and packets exchanged daily. Private networks, on the other hand, are closed systems accessible only to the owner, appealing to individuals seeking complete control over their network or companies prioritizing heightened security. Open networks, often known as community networks, are crowdsourced and publicly accessible at no charge, allowing anyone to connect their sensors or gateways.

The selection of network type depends on the project’s requirements. Public networks can be cost-effective for managing fewer devices, despite the associated usage fees. However, it is important to verify that areas requiring coverage are adequately covered before subscribing to a public network, as additional gateways can only be added by the network operator. Conversely, deploying a private network may be more feasible for projects requiring many devices. In such a case it is important to be aware of the additional expenses involved in gateway installation and maintenance, and the necessity for managing the LNS.

### 2.3.1 End nodes

The components constituting the first star in the network, as illustrated in Figure 2.3 are the end nodes or sensors, which establish communication with one or more gateways via LoRa RF modulation [6]. End nodes come in distinct types, ranging from application-specific devices to more versatile ones designed for indoor or outdoor usage, featuring either PCB or external antennas.

These end nodes are categorized into three classes - A, B, and C. All devices must incorporate class A, while B and C are extensions to the A class. These classes define how the nodes communicate with the network [6].

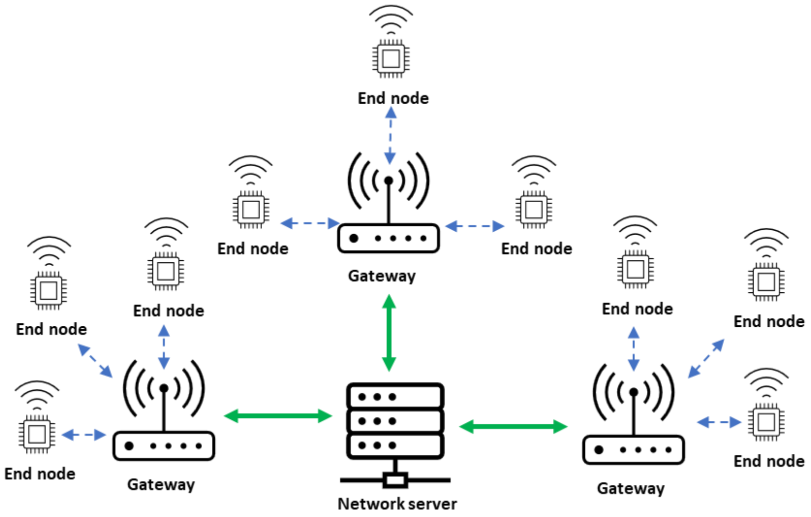


Figure 2.3 LoRaWAN utilizes a star network topology, where the network server serves as the central hub communicating with multiple gateways, which in turn connect with the end nodes. Blue signifies the LoRaWAN connection, while green denotes the IP connection.

Class A - All

The end device primarily remains in a sleeping state to conserve its battery, awakening either triggered by an event or periodically to transmit data to the network. Upon completing transmission followed by a short delay, the node awaits a response from the network within a configurable timeframe. If no response is received during the initial window, the device briefly returns to sleep before waking again for a second receive window. Should there be no response in the second window, the device goes back to sleep, preparing for the next uplink transmission. This process, depicted with colours in in Figure 2.4, where green indicating the device in sleep, blue representing the uplink period, and yellow denoting the receive windows. Since there is no means for the LNS to awaken Class A devices from sleep, this class is not suitable for actuators.

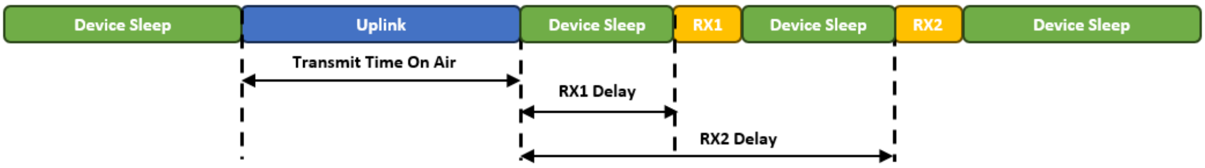


Figure 2.4 Illustration of Class A device operation, reproduced from [23, Fig. 2]

Class B - Beaconing

Class B end devices offer the same functionality as Class A devices and introduces regularly scheduled fixed time receive windows [24]. However, when using Class B, the network must periodically broadcast a time-synchronized beacon to synchronize the internal clocks of the end devices with the network. This synchronization enables end devices to open receive windows, also known as ping slots, at predetermined intervals, allowing the network to send downlink messages. In Figure 2.5, the beacon is shown in grey, while the ping slots are shown in orange, maintaining consistency with the other colours as with the Class A illustration. Compared to Class A, the devices can still be battery operated yet this operation mode have a large significant impact on the consumption due to the introduction of fixed intervals for powering up, receiving beacons, and opening ping slots. However, the introduction of Class B provides the benefit of significantly lower latency on downlinks, rendering the nodes more suitable for medium-level critical actuators.

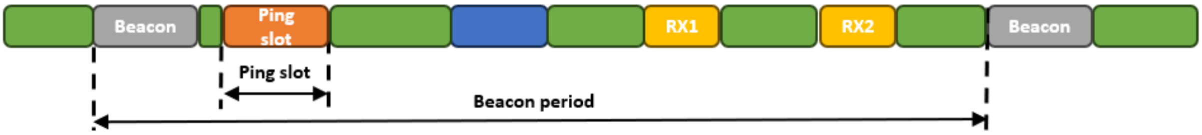


Figure 2.5 Illustration of Class B device operation, reproduced from [24]

Class C – Continuous

A Class C end device remains continuously active, consistently listening for downlink messages except during the transmission of an uplink [24]. This operational mode is only practical for sensors connected to mains power due to the increased energy consumption resulting from the device being always active. Like Class A, Class C devices open two receive windows, with the last window remaining open until a new uplink is transmitted. Figure 2.6 illustrates the transmission of an uplink, followed by a brief opening of the second receive window, and subsequently opening the first window. When the first window is closed the second is opened again and kept open until a new uplink is initiated.

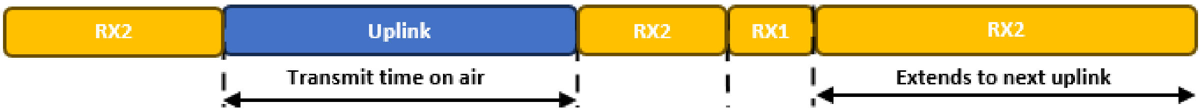


Figure 2.6 Illustration of Class C device operation, reproduced from [24]

## 2.3.2 Gateways

The second layer comprises gateways linked to the LNS via the IP protocol, establishing a backhaul connection through cabled Ethernet, Wi-Fi, or LTE. These devices pick-up messages from all end nodes within their coverage and transmit them to the LNS via a secure VPN connection. Operating solely on the physical layer, gateways function as LoRa radio message forwarders, conducting data integrity checks and discarding corrupted packets while forwarding intact ones to the LNS, accompanied by additional header data regarding received signal strength. Gateways are categorized into indoor and outdoor types, where outdoor variants provide larger coverage and higher sensitivity [25]. The gateways can monitor multiple channels simultaneously, typically supporting eight or sixteen channels [6]. The increased number of channels is closely correlated to the device cost.

### 2.3.3 Network server

The LNS is the central hub in the LoRaWAN star topology, operating as software on either a local or cloud server to manage the network. It ensures end-to-end bidirectional security through 128-bit AES encryption for all messages exchanged between end-devices and the application server, facilitating the transmission of application payloads to the appropriate application server and forwarding join-request and join-accept messages between end nodes and the join server [7].

Given that a single uplink can be received by multiple gateways, the LNS conducts de-duplication to eliminate duplicate transmissions, retaining only the one with the highest signal quality, while also authenticating each sensor on the network and verifying the integrity of every message [6].

For downlink messages transmitted from the LNS to end devices, the LNS optimally routes these messages, directing them to the gateway which has provided the best RSSI and SNR values based on previous transmissions between the respective end node. This ensures optimal link quality and minimizes airtime and packet loss. Additionally, the LNS manages the "Adaptive Data Rate" (ADR) function, which adjusts transmission parameters for each end node based on signal strength, directing nodes to use higher "Spreading Factors" (SF) for robust signals and lower SFs for weaker signals, thereby optimizing power consumption for each device.

### 2.3.4 Application server

The application server is responsible for securely processes the uplink sensor data [6]. There can be more than one application server in the LoRaWAN network depending on the types of services required by the users, an example with dashboard is shown in Figure 2.7. It also generates all the application-layer downlink payloads and sends them to the connected end devices through the LNS.

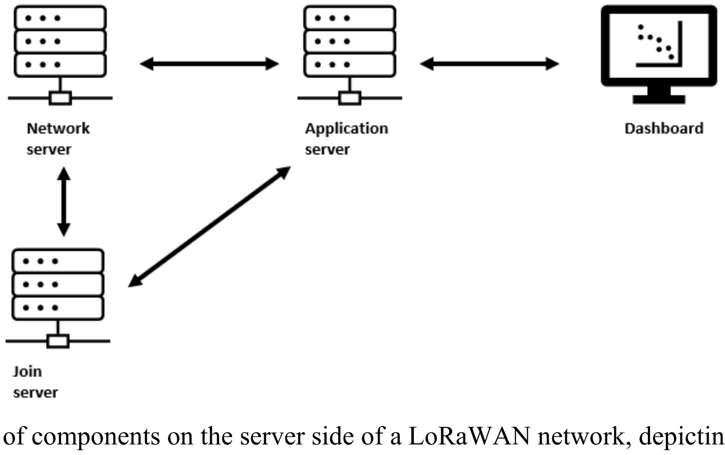


Figure 2.7 Illustration of components on the server side of a LoRaWAN network, depicting the network join and application servers. The application server connects to a dashboard application for displaying data.

### 2.3.5 Join server

The join server manages the over-the-air activation process for end devices to be added to the network [6]. It contains the necessary information to handle join-request frames from end devices requesting to join the network and generate corresponding join-accept frames if the request is accepted. It stores the root keys and generates session keys.

## 2.4 LoRaWAN security

Security is a critical aspect of any IoT implementation, and LoRaWAN is designed to uphold the confidentiality, integrity, and authenticity of transmitted data [6]. Confidentiality ensures that data is encrypted end-to-end, preserving its secrecy for the owner, while integrity involves verifying that the data remains untampered with, and authenticity confirms its origin.

In LoRaWAN, a two-layer security approach is mandatory [26]. The first layer, application encryption, guarantees end-to-end encryption between the end device and the application server, safeguarding application data from unauthorized access or tampering until the application server decrypts it. The second layer, network encryption, ensures the authenticity of the end device.

Figure 2.8 depicts these layers in a pipeline format, with the application layer shown in yellow and the network layer in blue. Additionally, a grey layer named backend security illustrates further protection methods applicable to the backend IP network, such as IPsec, TLS, and HTTPS.

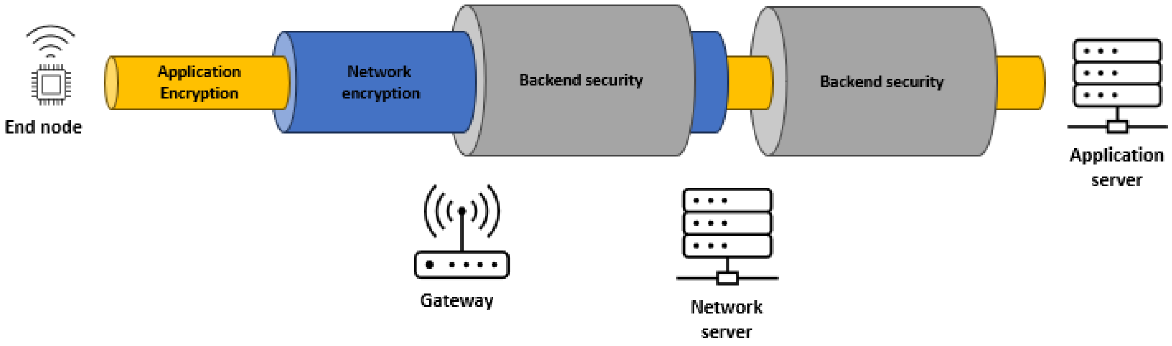


Figure 2.8 Pipeline illustration showing the different layers of security in LoRaWAN, reproduced and adapted from [27]

### 2.4.1 LoRaWAN join procedure and encryption keys

There are two methods for joining a LoRaWAN network: “Over The Air Activation" (OTAA) or “Activation By Personalization” (ABP).

When utilizing OTAA, the device initiates a dynamic join procedure to request access to the network. This join request includes the device’s JoinEUI, DevEUI, and DevNonce. The JoinEUI is a unique global application identifier identifying the join server, while the DevEUI serves as a global unique identifier for each end device, and the DevNonce is a counter that increments with each join request message. To prevent replay attacks, where a sensor is impersonated to disrupt the network, the LNS rejects join requests with a previously used DevNonce. A Message Integrity Code (MIC) is calculated based on these fields using the

AppKey, an AES-128-bit secret key stored in the end device and the Join Server [28]. Both the JoinEUI and DevEUI are publicly accessible information.

The LNS processes the join request, and the Join Server authenticates whether the sensor should be granted network access. If authorized, the join server generates two AES 128-bit session keys, Network Session Key (NwkSKey) and Application Session Key (AppSKey), followed by the network issuing a join accept message. The end device derives these session keys locally based on the AppNonce in the Join Accept message and the AppKey. Additionally, the device is assigned a specific device address (DevAddr) on the network. While the network obtains access to the NwkSKey, the AppSKey is only shared with the application server. Session keys are renewed each time a device joins the network, enhancing connection security.

Alternatively, with ABP, session keys are statically programmed into the device and herby bypass the join server. This accelerates the join process, as it does not require the handshake procedure of OTAA. However, ABP is less secure since it maintains the same security sessions throughout deployment, unless manually altered by accessing the device physically.

### 2.4.2 Security in deployment

The deployment of LoRaWAN networks is an essential part of overall security. Sensors should be strategically deployed to limit access by unauthorized individuals. In the event unauthorized persons gain access to a sensor node, encryption keys must be securely stored to prevent unauthorized retrieval. Hardware-level security can be enhanced by incorporating hardware secure elements, specialized chips designed to securely store keys, rendering physical access impossible. Similarly, securing LoRaWAN backend interfaces by segregating key storage in the join server is equally important [27].

A significant challenge with deploying battery-operated sensors intended for continuous operation over several years is maintaining their software and keeping pace with evolving security measures. Given the inevitability of requiring bug or security-related patches during this period, performing updates in the field across numerous devices proves both time-consuming and costly. A recent development, “Firmware Updates Over The Air” (FUOTA), offers a solution by enabling simultaneous updates of multiple nodes, ensuring field deployments remain up to date with essential security patches.

## 2.5 LoRaWAN software libraries

There are several open-source LoRaWAN software libraries available. Semtech’s LoRaMAC-node has served as a reference implementation for several years but has transitioned to maintenance status [29] following the release of its successor, the “LoRa Basics Modem” (LBM). LBM is the platform where Semtech implements new LoRaWAN features and improvements introduced by the LoRa Alliance. At the time of writing, the latest version of LBM’s master branch is 4.3.1, initially released in November 2021, with the software coded in C.

Another option is the LMIC (LoRaMAC in C) library, originally developed by IBM. Although IBM ceased development in 2016 [30], it was subsequently ported to the Arduino environment and renamed Arduino LMIC [31]. This library was deprecated at version 1.5.1 in October 2020. However, a maintained fork, the MCCI Arduino LMIC library, still exists, with the last release in December 2021. It supports LoRaWAN specification version 1.0.3 but lacks relay functionality.

Beyond these, several other libraries are available. Some LoRa development board manufacturers, such as STM or RAKWireless, develop their own libraries tailored to their hardware. Additionally, companies offer customized LoRaWAN stacks as commercial services, such as Stackforce. These options provide developers with flexibility to choose the library that best fits their project's requirements and development environment.

## 2.6 Lora Basics Modem

LBM, developed by Semtech, streamlines the creation of LoRaWAN end nodes by offering a comprehensive library coupled with extensive documentation, including user manuals, porting guides, and application examples. Its event-driven API provides a high-level abstraction, minimizing the prerequisite LoRaWAN knowledge needed for development.

The three main components of the software are the "Hardware Abstraction Layer" (HAL), the modem core and the "Application Programming Interface" (API). The HAL controls the interaction with the hardware. The modem core consists of the main LBM functionality, such as modem supervisor for task scheduling, modem crypto for software-implemented cryptographic engine, and radio abstraction layer which provides a generic interface for the radio interaction. The API serves as the interface for user application development and interaction with the event-driven software platform.

### 2.6.1 Building example application with GNU Make

LBM employs a Makefile-based approach, ensuring independence from specific Integrated Development Environments (IDEs). The Makefile servers as the recipe for the compiler, instructing which files to be built including their storage location. Utilizing GNU Make, which is part of the ARM GNU Toolchain, enables compilation and building of code targeting ARM architectures. Executing the Make command is straightforward within a Linux Bash shell. For Windows 11 users, it is possible to interact with the compiler through Bash on Ubuntu on Windows, allowing seamless integration of Bash commands within Visual Studio Code's terminal.

To compile one of the example applications, navigate to the Makefile location and input the command "make lr1110 MODEM_APP=PERIODICAL_UPLINK" in the terminal. Before building, ensure to update the "example_options.h" header file with the necessary DevEUI, JoinEUI, and AppKey, providing the end node with correct identification and keys for device activation on the LNS.

Multiple build options are available when using the make command. LBM supports various radio targets and two MCU variants, which must be specified in the make command. Additionally, options such as LoRaWAN region selection, choice between software or hardware cryptographic engines (with hardware support only available for LR11xx transceivers), and activation or deactivation of debugging options can be configured.

### 2.6.2 Portability

LBM is engineered for platform independence, with Semtech providing comprehensive documentation detailing the process of adapting the software for new MCUs or radios [32], a procedure known as porting. The library includes drivers for LR11xx, SX126x, SX127x, and SX128x radio transceivers, as well as support for two STMicroelectronics Nucleo development boards: L476RG and L073RZ, featuring MCU STM32L476RGT6 and STM32L073RZ, respectively.

To port the software to a different development board, several steps must be taken to ensure its functionality. Initially, MCU drivers must be obtained, typically supplied by the board manufacturer. ST, for instance, offers drivers for their boards which can be downloaded from GitHub. These drivers encompass both the MCU HAL and the ARM CMSIS library, which serves as the ARM standard microcontroller software interface. Additionally, a set of helper functions tailored to the development board's HAL library must be created. These functions are essential for facilitating communication between the modem HAL and the MCU, utilizing functionalities such as GPIO, SPI, and UART. If the porting process involves introducing a new radio chip, the radio HAL board support package must be updated accordingly. Subsequently, the modem HAL implementation must be adjusted to accommodate the new MCU. Finally, the Makefiles must be updated to properly link the corresponding files for the new build option.

## 2.7 Comparison of IoT protocols

IoT protocols can be categorized into two main groups: short range and long range. Short-range technologies include Bluetooth, NFC, Wi-Fi, Z-Wave, and Zigbee, with Z-Wave, Zigbee, and Wi-Fi commonly employed for home IoT appliances, NFC utilized for wireless payments, and Bluetooth favoured for handheld devices like headphones and speakers.

In contrast, the long-range group includes technologies such as LoRaWAN, Sigfox, NB-IoT, and LTE-M. These technologies are distinguished by differences in data rate, payload size, and spectrum usage. NB-IoT and LTE-M are cellular-based technologies offering higher data rates but require a SIM card and payment to a network operator as they operate in licensed spectrum. LoRaWAN and Sigfox operate in ISM bands, but Sigfox networks are managed by a service provider, preventing establishment of private networks. LoRaWAN relies on proprietary Semtech LoRa chips, making both LoRaWAN and Sigfox proprietary. While Sigfox may offer better coverage on paper, LoRaWAN compensates with increased uplink data rates and payload size.

Table 2.2 Comparison of LPWAN protocols [33], [34], [35], [36], [37]

| Technology | LoRaWAN | Sigfox | NB-IoT | LTE-M |
|---|---|---|---|---|
| Proprietary or open | LoRa – proprietary<br>LoRaWAN – open | Net – proprietary<br>Devices - open | Open | Open |
| Estimated coverage [km] | 5-20 | 10-40 | 1-10 | 1-10 |
| Licensed spectrum | No | No | Yes | Yes |
| Uplink data rate [kbit/s] | 0.3 (SF12)<br>5.47 (SF7) | 0.1 | < 66 | 1000 |
| Max. payload size [bytes] | 242 | 12 | 1280 | 1280 |
| Max. messages per day | Depends on duty cycle regional requirements | 140 | Unlimited | Unlimited |
| Battery life | Long | Long | Long | Long |

# 3 LoRaWAN relay

In cases where a few LoRaWAN end nodes are located at the edge or outside the existing network range, the traditional solution has been to deploy a new gateway closer to these nodes, hereby extending the network coverage. Improving the coverage will also reduce the node's battery consumption, as they will be able to transmit with a lower spreading factor and hereby reducing the time on air.

Installing additional gateways comes at a high installation and material cost, demanding mains power and a backhaul connection to the internet. For the cases described above, installing additional gateway quickly becomes non-cost-effective and in some cases non-feasible. Addressing this challenge, the LoRa Alliance recently introduced the new relay feature enabling LoRaWAN devices to forward messages from other nodes. This functionality can be implemented into the firmware of normal end nodes or applied to a dedicated device for this purpose [38, p. 2], making this a significantly more cost-effective approach to extending the network coverage to challenging locations [5]. Figure 3.1 illustrates the principle of the relay, where the relay is deployed within the original coverage provided by the gateway and extending this network to reach an end node previously out of network range.

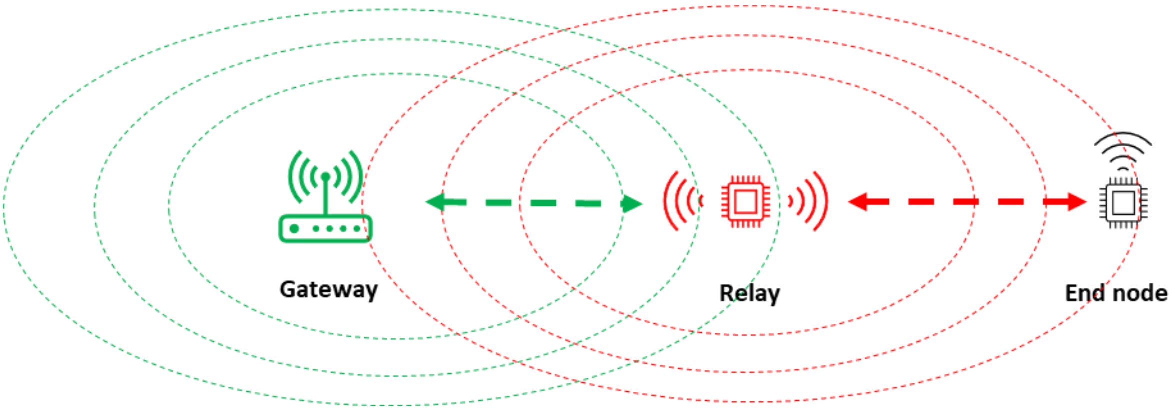


Figure 3.1 Operational principle of the relay, showing the relay operating within the coverage of the gateway extending the network to the stranded end node

The relay devices can be powered either by battery or mains and utilizes the existing LoRaWAN network for backhaul eliminating the necessity for internet connection. In addition to the hardware being more budget-friendly compared to a gateway, this makes relay a simpler and more budget friendly solution than deploying an additional gateway.

The relay specification requirements are described in the LoRaWAN Relay Specification TS011-1.0.0 document [39], and specific regional parameters is detailed in RP002-1.0.4 Regional Parameters [17, p. 36]. More details on how the relay feature has been designed will be described in the following sections.

## 3.1 The relay approach

The majority of LoRaWAN end devices operate as Class A devices, transmitting uplinks infrequently, randomly, or in response to events, asynchronously [38, p. 2]. To prevent the relay from continuously consuming power and rapidly depleting its battery, an energy-saving technique has been developed. This technique involves the utilization of relay listening windows and synchronization between the relay and connected end devices.

The listening windows are facilitated by a function named as Channel Activity Detect (CAD), which operates in a low-power mode where the LoRa radio periodically scans for channel activity while most of the onboard logic of the relay remains in sleep mode to conserve energy. The CAD periodicity, indicating the time between the beginning of each relay listening window, is configurable, with a default and maximum value of 1 s. In the event of detected activity, the CAD function will awaken the main control module for frame evaluation and further processing. Conversely, if no activity is detected, the radio module will return to sleep mode.

### 3.1.1 Wake on radio

Figure 3.2 depicts the message flow between the end device, relay, and gateway. To awaken the relay from the CAD mode, the end device employs a “Wake On Radio” (WOR) uplink frame. There are two types of WOR frames an end node can utilize to communicate through the relay: one for the LoRaWAN join procedure and the other for standard uplinks [39, p. 11].

The WOR uplink frame is comprised of a lengthy preamble followed by information regarding the frequency and spreading factor of the subsequent LoRaWAN uplink. This data enables the relay to adjust its reception channel to the specified frequency. The length of the WOR preamble is determined by the synchronization state of the end node, as detailed in the subsequent sub-chapter.

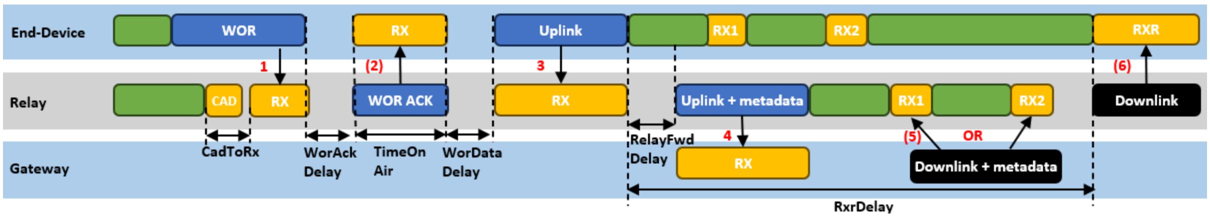


Figure 3.2 Typical message sequence between end device, relay and gateway, illustrated with the devices separated into rows. Drawing is reproduced and adapted from [39, p. 10], [40]

When the relay detects the WOR uplink, it attempts to demodulate the message and verify its integrity. Upon successful validation, the relay may respond with a WOR ACK message, signifying receipt of the WOR and readiness to receive the subsequent LoRaWAN uplink. This acknowledgment includes essential information for time synchronization, establishing a secure communication link, and detailing the relay forwarding transmission [39, p. 14].

Subsequently, regardless of whether the end device has received the WOR ACK, it transmits the uplink package in standard LoRaWAN format. As the relay protocol operates atop the existing LoRaWAN link layer specification, the standard uplink can be received by either the relay or a gateway if within range. Upon receiving the uplink, the relay encapsulates the entire message and adds metadata overhead from the received transmission [40]. Based on information provided in the WOR ACK message, the end device is aware of the payload limitations of the following relay uplink message. If the end device uplink message exceeds the payload limitation dictated by the relay data rate, the relay will drop it. The encapsulated uplink, along with metadata, is then forwarded in the relay's LoRaWAN uplink over dedicated frame port 226.

The network segregates the two uplink packets, associating the corresponding header data with each uplink and processing them as standard uplinks. If the LNS has a downlink packet for the end device, it encapsulates the message and sends it to the relay using port 226 and either RX window 1 or 2, as with a standard Class A device. Upon receiving this packet, the relay extracts the message for the end device and forwards it, using an additional dedicated reception window for the end device, called as RXR.

### 3.1.2 End device synchronization

Synchronization between the relay and the relayed end devices is a critical aspect of the relay protocol, with three defined synchronization states: initialized, unsynchronized, and synchronized. During initialization, the device lacks key relay parameters such as the relay's listening time on the default channel, CAD periodicity, CAD to RX mode delay, crystal oscillator frequency accuracy, default channel index, and second channel usage [39, p. 18]. The CAD to RX mode delay, depicted as CadToRx in Figure 3.2, indicates the time the relay transitions from CAD mode to reception mode. Crystal oscillator accuracy determines the relay clock's precision, aiding the end device in estimating the relay's listening window.

In this state, the end device must make assumptions to initialize communication, beginning with default values such as no second channel, a CAD periodicity of 1ms, crystal oscillator accuracy of 40ppm, and 8 symbols for the CAD to RX transition [39, p. 18]. The WOR frame preamble, during initialization, is set to a maximum size of 1000ms to align with the relay's maximum CAD periodicity. Transitioning from initialized to synchronized occurs upon receipt of a WOR ACK from the relay, enabling the end device to estimate the relay's CAD windows and adjust future WOR frame preambles, accordingly, thus conserving energy.

If the end device estimates a listening window longer than the CAD periodicity, it shifts from synchronized to unsynchronized state, changing the preamble length to align with the relay's CAD periodicity received in the WOR ACK. Furthermore, the end device transitions to unsynchronized state after transmitting eight consecutive uplinks without receiving a WOR ACK frame, and to initialized state following eight more consecutive uplinks without the receipt of an WOR ACK frame.

### 3.1.3 WOR RF Channels

The LoRaWAN Regional Parameters specification (RP002-1.0.4) outlines the default channels utilized for the WOR frames, with the EU employing four default channels as listed in Table 3.1. The relay's configuration permits it to scan for activity on either one or both channels, where if both channels are utilized, the relay alternates between the channel scans, effectively doubling the total CAD periods to maintain consistent periodicity on each channel.

Table 3.1 EU868 default WOR channels [17, Tbl. 15]

| Channel Index | 0 | 1 |
|---|---|---|
| Frequency WOR | 865.1 | 865.5 |
| Frequency WOR ACK | 865.3 | 865.9 |
| SF | 9 | |
| BW | 125 | |

### 3.1.4 Relay default parameters

Table 3.2 outlines the default relay parameters outlined in RP002-1.0.4 Regional Parameters [17, Ch. 4.4], applicable to all regions covered by the document. The listed delays are depicted in Figure 3.2. WorAckDelay represents the delay between the conclusion of a WOR frame and the commencement of an ACK frame, while WorDataDelay signifies the delay between the conclusion of an ACK frame and the initiation of a LoRaWAN uplink. The TimeOnAir for the WOR ACK can be computed by the end device based on the known data rate. If the end device does not receive an ACK frame, it should utilize the calculated TimeOnAir along with the two specified delays to schedule its LoRaWAN uplink.

Additionally, RelayFwdDelay denotes the delay between the completion of receiving a LoRaWAN end device uplink and the relay commencing the forward transmission to the LNS. The RxrDelay defines the delay between the conclusion of the end device LoRaWAN uplink and the commencement of the end device relay-specific reception window.

Table 3.2 Default relay settings [17, Ch. 4.4.5]

| | |
|---|---|
| TRUSTED_ED_NUMBER | 16 |
| WOR_ATTEMPTS_WO_ACK | 8 |
| WOR_DATA_DELAY | 50ms |
| WOR_ACK_DELAY | 50ms |
| RELAY_FWD_DELAY | 50ms |
| RXR_DELAY | 18 s |

## 3.2 Relay package structure

This chapter provides an overview and visualisation of the formatting and size of the WOR, WOR ACK and forward uplink packages.

### 3.2.1 WOR format

Figure 3.3 illustrates the structure of a WOR uplink, comprising a WOR header field and a WOR payload field. The header specifies the WOR message type, utilizing the four least significant bits of the assigned byte, while the remaining bits are reserved for potential future use. The WOR payload includes information relevant to the WOR type, with the field size varying depending on the specific WOR type. Table 3.3 lists the two types of WOR messages and their corresponding allocated bits as defined in the LoRaWAN Relay Specification version 1.0.0 [39].

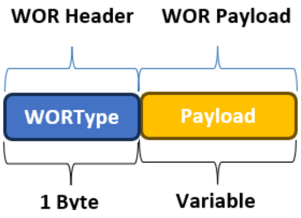


Figure 3.3 Format of WOR uplink

Table 3.3 WORType bit decoding [39, Tbl. 4]

| WORType | Description |
|---|---|
| 0 | Relay Join-Request |
| 1 | Relay Class A Uplink |

The structure and size of the WOR payload vary between the two WOR types. Figure 3.4 illustrates the WOR join-request, comprising the WorDrPL and Frequency fields. The WorDrPL, occupying one byte, designates the data rate of the subsequent join-request uplink frame, with the four least significant bits dedicated to this purpose, while the remaining four bits are reserved for future use. The Frequency field, spanning three bytes, denotes the frequency of the next join-request uplink frame.

The payload format of the WOR class A uplink frame is depicted in Figure 3.5. It includes the device address of the end device, the encrypted WOR uplink (WorUplinkEnc), the WOR frame counter, and the MIC. WorUplinkEnc represents the WOR Uplink encrypted with the WOR Encryption Session Key (WorSEncKey), as detailed in chapter 3.3. Once decrypted, the WOR uplink format mirrors the payload field of a WOR join-request, encompassing the data rate and frequency of the subsequent uplink frame.

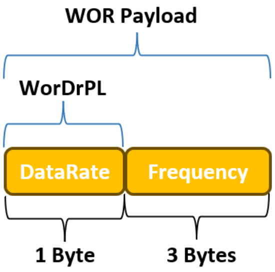


Figure 3.4 Format of a WOR join-request payload

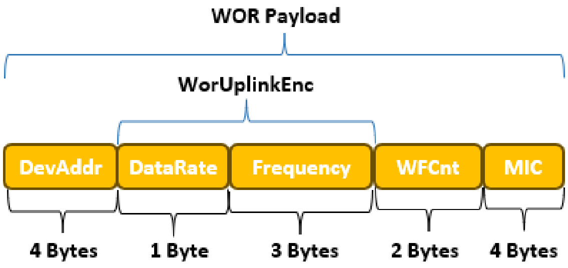


Figure 3.5 Format of a WOR Class A Uplink payload

### 3.2.2 WOR ACK format

As previously explained, in response to a WOR frame, the relay may transmit a WOR ACK message. Figure 3.6 illustrates the format of an ACK for a WOR class A uplink, comprising two fields: AckUplinkEnc (encrypted WOR ACK uplink) and MIC. AckUplinkEnc represents the ACK uplink encrypted with the WorEncSKey. Upon decryption, the ACK uplink format corresponds to the StateSync frame, which includes the elements detailed in Table 3.4, providing insight into the internal state of the relay. This information enables the end device to optimize the length of the WOR preamble and constrain the size of the uplink package based on the constraints imposed by the relay forwarding data rate.

Table 3.4 Description of relay internal state elements [39, pp. 25–26]

| Bits | Abbreviation | Description |
|---|---|---|
| 2 | CadToRx | The delay, measured in number of symbols, between the start of CAD and opening of the relay reception window |
| 2 | Forward | The current state of the relay forward limitations |
| 4 | RelayDR | The data rate (bandwidth and SF) used by the relay to forward the end device payload |
| 2 | XTALAcc | The relay crystal oscillator frequency accuracy in ppm |
| 3 | CADPer | The periodicity between two consecutive CAD scans on a single channel |
| 11 | TOffset | The time, in ms, between the start of the scan and the end of the reception of the WOR frame preamble. |

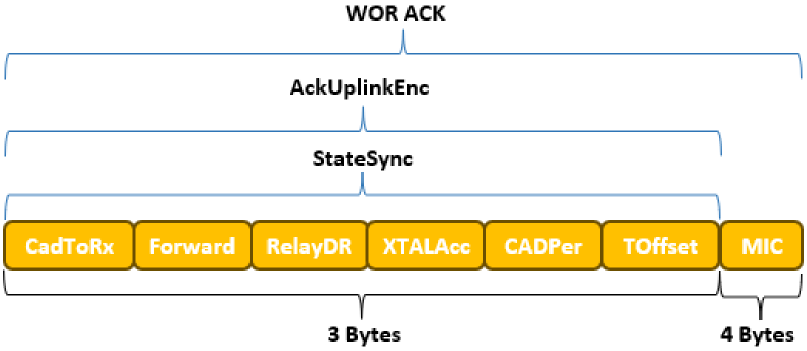


Figure 3.6 Format of WOR ACK relay class A uplink

### 3.2.3 Forward uplink format

Upon receipt of the uplink, the relay encapsulates the entire message and supplements it with metadata obtained from the received transmission. This metadata encompasses details such as RSSI, SNR, RX frequency, RX data rate, and the WOR channel employed by the end node. Subsequently, the encapsulated data functions as the payload in the message relayed to the LNS. Figure 3.7 portrays the complete forwarded package, encompassing the relay header data, relay payload, and the relay MIC.

In this example, when transmitting a small payload from the end device, a substantial portion of the transmitted size constitutes overhead data, with only 4 out of the 36 bytes transmitted originating from the actual payload.

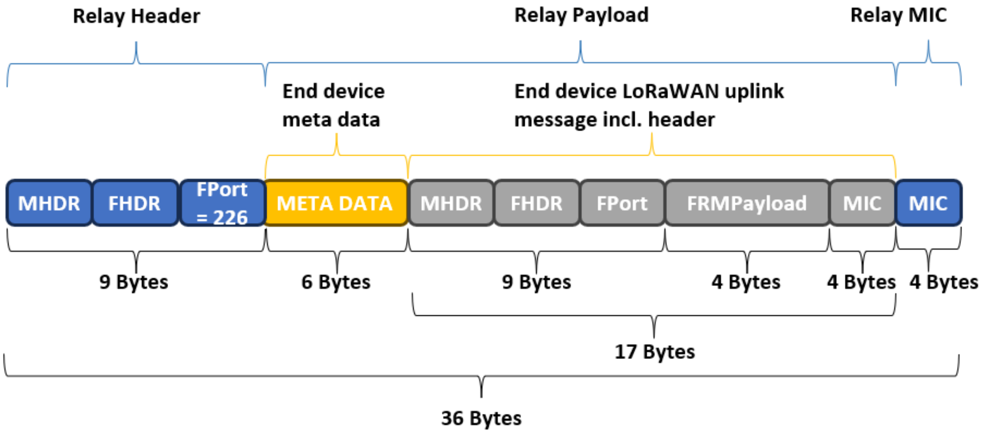


Figure 3.7 Format and size of relay forwarded uplink packet, based on the 4 bytes counter used when testing the relay functionality in the upcoming chapters, reproduced and adapted from [38, p. 6], [39, Ch. 9]

## 3.3 Relay security

Being a Class A LoRaWAN device, the relay utilizes the same security elements as described in the LoRaWAN chapter with its own individual identifier and AppKey. In addition, a new relay session key (RootWorSKey) is introduced with the relay protocol to secure the WOR communication. This is a unique key for each end device which is used by both the relay and end device to derive a WOR Integrity Session Key (WorSIntKey) and a WOR Encryption Session Key (WorSEncKey). The end device derives the RootWorSKey from its NwkSKey after receiving a Join Accept while the relay will receive a copy of this key from the LNS through the UpdateUplinkListReq MAC command [38, p. 5]. WorSIntKey is used to ensure the data integrity of the WOR and WOR ACK frames while the WorSEncKey is used for encryption and decryption of the payloads in these messages [39, p. 20]. Since the end device encrypts its payload in the same manner as a standard LoRaWAN uplink message, this means that the relay does not possess the keys to decrypt the payload, maintaining the LoRaWAN end-to-end encryption.

## 3.4 Relay limitations & restrictions

Before determining to utilize a relay as a network extender, some limitations should be underlined. Firstly, the relay should not be viewed as a substitute for a gateway. Gateways incorporate a more robust and costly radio chip, whereas the relay is specifically engineered to serve as a simple and cost-effective range extender.

In terms of functionality, the relay protocol exclusively supports single-hop communication, meaning it cannot forward messages from other relays. Additionally, there is a maximum limit of sixteen connected end devices per relay unit.

Operating within the ISM bands, the relay must comply with the same regulatory requirements as other devices. As outlined in Table 2.1, there are regional constraints on the maximum payload size permitted for each LoRa transmission. These listed maximum payloads are repeater compatible. Given that the relay appends additional header data to the original uplink message, it is essential for end devices to ensure their payloads remain within the specified limits for each data rate. Failure to do so will result in the relay dropping packages.

Additionally, regional regulations impose constraints on the permissible duty cycle, which limits the frequency of uplinks the relay can transmit. As a result, end devices need to adapt their uplink frequency accordingly. This aspect should be considered when evaluating relay use cases, particularly and especially when positioning the relay. The goal should be to minimize the spreading factor while maintaining transmission parameters that ensure effective communication with the gateway.

Another factor restricting uplinks is the forwarding limitations imposed by the relay [39, Ch. 8.8]. This function operates on a token bucket principle, where the relay maintains a bucket with a maximum number of tokens. Each uplink depletes one token from the bucket, and the algorithm can be configured to separate tokens for different message types. If the bucket becomes empty, the relay will cease listening to end device uplinks. A fixed number of tokens regenerate hourly, and end devices receive notifications regarding the forwarding limit status through the WOR ACK message.

The default parameters for each message type are outlined in TS011-1.0.0 Table 23. For instance, the default bucket size for the shared bucket for all Class A end device uplinks is sixteen tokens, with a reload rate of eight tokens per hour. Alternatively, it is possible to bypass this function by setting the reload rate to no limit. However, it is important to note that the duty cycle requirements always take precedence over the token bucket mechanism.

Currently, the most notable limitation lies in the absence of commercially available devices featuring the experimental release of the LBM stack. Two device manufacturers have been identified as providers of relays off the shelf: Abeeway and Deviceroy. Abeeway, a subsidiary of Actility, has developed a relay based on their existing compact tracker hardware, which forms part of the evaluation kit used to test the relay functionality in this thesis. However, it should be noted that this product has not been found to be commercially available and was supplied to Altibox for testing and familiarization purposes. The other supplier, Deviceroy, a start-up based in Salt Lake City, offers a relay product on the market running the official fourth version of Semtech's LBM.

## 3.5 LBM relay implementation

A branch of LBM, named feature-relay, proposes an implementation of the LoRaWAN Relay Specification TS011-1.0.0. This branch serves as an experimental release and has not yet been merged into the master branch. The latest software version for this branch is at the time of writing 4.4.0, which can be found on GitHub [41]. This implementation introduces options to specify relay RX or TX during the build process. Enabling RX configures the device to act as a relay, listening to WOR messages. Conversely, enabling TX configures the device as a relay-enabled end node, triggering WOR messages before each consecutive uplink.

To accommodate the relay modes, Semtech has modified the periodic uplink example application as previously mentioned in chapter 2.6.1, now incorporating additional modem events.

### 3.5.1 Relay hardware requirements

The relay utilizes the same single-channel Semtech LoRa transceiver chip as any other LoRaWAN end device [38, p. 7]. However, certain chip families have limitations regarding compatibility, notably the SX128x and SX127x, due to their restricted CAD functionality [42]. According to the information provided with the experimental release of the relay enabled LBM, this applies to both the relay unit and the relayed end devices, whereas the LR11xx and SX126x chips are fully supported.

To enable the relay functionality in the Semtech LBM stack, additional onboard storage is required. The amount of additional memory needed varies depending on whether building a relay or an end node. Specifically, the relay necessitates an extra 2.5 KB of RAM and 10 KB of flash storage, while an end node requires 500 bytes of RAM and 5.5 KB of flash [42].

Semtech also recommends incorporating a temperature-compensated crystal oscillator (TCXO) into the relay hardware. This type of crystal oscillator offers high clock precision, minimizing the impact of temperature variations. This ensures better synchronization between the relay and the end nodes, thereby maintaining the synchronized state of the end nodes for longer periods.

# 4 Altibox LoRaWAN infrastructure

Reverting to the description of the LoRaWAN architecture in chapter 2.3, Altibox provides a public network where base stations are strategically placed to provide coverage over large areas. After several years of expanding their network, it now covers over 100 municipalities and reaching more than 1 million households [4]. Trondheim, Molde, Stavanger and Porsgrunn is among the larger cities covered. Having the network infrastructure in place makes it easier for municipalities or businesses to harvest from the unique opportunities provided by the internet of things. It reduces the initial capital expenditure, eliminating the need to purchase and install gateways and establishing the LNS, enabling the end users to focus on the sensor end the application ends. Replacing or simply adding new sensors connected to the internet can provide data which can be used to increase production efficiency, reduce waste, and cut costs.

Altibox provides three main services:

- IoT Access is the basis service offering. Through this option, end user can integrate their sensors into the network and access the data via one of the available APIs. This choice is ideal for users who have their own application server for data management and visualization purposes.

- IoT Total represents the complete service package, offering clients the simplest entry point into IoT. With this option, end users integrate their own sensors, while Altibox manages the entire connectivity process, from end node setup to data presentation through a user-friendly graphical user interface. Through this package, Altibox delivers both the application server and the data visualization capabilities, in addition to the networking.

- IoT Water Insights is a specialized service tailored for municipalities seeking to monitor their water resources. By installing sensors in the water network, it enables monitoring of water flow, facilitating the detection of water leaks. This service assists clients in conserving water, time, and money by promptly identifying leaks thereby minimizing time spent on leak detection and reducing maintenance costs. This is a collaborative service offered in partnership with InfoTiles. InfoTiles provides the application server and user interface, while Altibox delivers the end-to-end network, including the LNS.

# 5 ThingPark Wireless

Altibox uses the ThingPark Wireless (TPW) network server for managing their LoRaWAN network. ThingPark is a software supplied and developed by Actility, one of the major players in the worldwide LPWAN environment. TPW is accessed through a web interface, and is used to connect, manage, and monitor the end devices and gateways. The solution comes with the following integrated applications:

- Network Manager, used to manage the gateways.
- Device Manager, used to manage the end devices in the network.
- Wireless Logger, used to monitor the network traffic and inspect packages.
- Network Survey, used to assess and visualize the LoRaWAN network coverage.

The Network Manager application encompasses a dashboard presenting a graphical status overview of the gateways, a page listing alarms, and another page dedicated to provisioning new base stations through an intuitive step-by-step guide. Since the gateways provided by Altibox for this project were already provisioned, the Network Manager application was not utilized.

The Network Survey application is a useful tool to analyse the LoRaWAN coverage in outdoor areas. It generates a graphical representation of the network, color-coding samples based on signal quality and plotting them on a map according to the GPS coordinates reported by the test device. However, since the tests conducted in this project were performed indoors, this application did not offer any additional value in this particular use case.

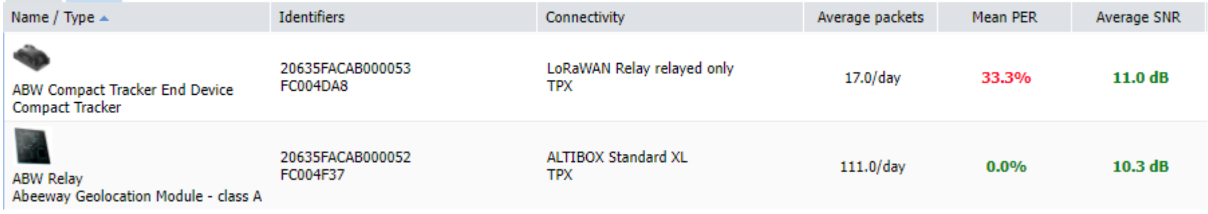
## 5.1 Device Manager

The Device Manager application is the back-end user interface for adding and managing the sensors. Adding new devices is done through a pop-up dialogue box, where first the activation method must be chosen. This will then define which device identification fields are required. For OTAA, the DevEUI, JoinEUI and AppKey are required and for ABP DevEUI, DevAddr and NwkSKey must be entered. Common for the two methods is that the device needs a name, a manufacturer and model type. Finally, the device needs to be assigned a connectivity plan and application server routing profile.

As the Altibox network type is public, there are recurring fees related to the use of the network. These are managed through connectivity plans, which are created by the network supplier and required for any end device to connect to the network. The plan defines the type of connectivity and related parameters and the number of permitted end devices.

The application server routing profile defines which application servers should receive uplink from the specific end node. The destination is defined in the application server profile, by specifying the server address.

Figure 5.1 shows a snippet from the device list in Device Manager, where the relay and the relayed end device have been onboarded by use of OTAA. As indicated, the two devices utilize different connectivity plans. This is done for the relay mapping, so that the gateway would not accept the traffic from the end node directly, only when it is forwarded from the relay. In addition to the connectivity, the list includes information about the device identifiers, which is the DevEUI and the DevAddr, daily average packet rate, mean PER and, average SNR for each device.



| Name / Type | Identifiers | Connectivity | Average packets | Mean PER | Average SNR |
|---|---|---|---|---|---|
| ABW Compact Tracker End Device<br>Compact Tracker | 20635FACAB000053<br>FC004DA8 | LoRaWAN Relay relayed only<br>TPX | 17.0/day | 33.3% | 11.0 dB |
| ABW Relay<br>Abeeway Geolocation Module - class A | 20635FACAB000052<br>FC004F37 | ALTIBOX Standard XL<br>TPX | 111.0/day | 0.0% | 10.3 dB |

Figure 5.1 Relay and end device provisioned in ThingPark Wireless, listing identifiers, connection details and radio parameters

## 5.2 Wireless Logger

The Wireless Logger application enables the monitoring and analysis of all traffic captured by the LNS, encompassing both uplinks and downlinks, excluding WOR messages exchanged solely between the end device and the relay. This tool allows for package inspection and even payload decoding using a built-in decoder. It provides insights to network metadata, including the “Frame Count” (FCnt), “Received Signal Strength Indicator” (RSSI), “Signal-to-Noise Ratio” (SNR), “Estimated Signal Power” (ESP), and “Spreading Factor” (SF). While SF and FCnt have been previously discussed, the remaining parameters represent radio metrics that characterize the quality of received uplinks:

- RSSI quantifies the total received signal strength across the channel bandwidth, calculated by summing the signal (S), interference (I), and background noise (N) as indicated in Eq. 5-1. It is expressed in dBm and represented as a negative value, with values closer to zero indicating stronger signals.

$$RSSI = S + I + N \qquad \text{Eq. 5-1}$$

- SNR denotes the ratio between the useful signal and interference plus background noise, calculated by the gateway as shown in Eq. 5-2. It is expressed in dB, with positive values indicating a stronger signal.

$$SNR = \frac{S}{I + N} \qquad \text{Eq. 5-2}$$

- ESP represents the estimated received signal strength of the useful signal, calculated by the gateway based on the RSSI and SNR using the formula in Eq. 5-3. ESP is expressed in dBm and consistently holds a negative value. Like RSSI, values closer to zero signify a stronger signal.

$$ESP = S = RSSI - 10\log\left(1 + 10^{-\frac{SNR}{10}}\right) \qquad \text{Eq. 5-3}$$

Wireless Logger uses a colour code system for visualizing the SNR, ESP, and SF, which are shown in Table 9.1. All the above parameters are important for the LoRa transmission, and as can be seen from the equations they are intricately linked. As mentioned, LoRa can operate with SNR ratio as low as -20dB by using spreading factor 12. Thereby the SNR can give an indication of the margins within the given SF.

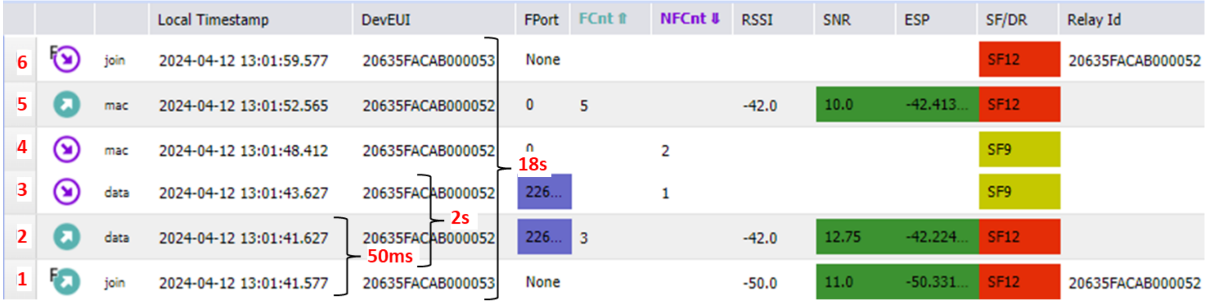
The packets can be filtered based on several values, most relevant when using only one base station is the DevEUI or DevAddr, from/to timestamps and packet type. There is also an option to automatically update the list of packets on a set time interval, useful when testing to quickly identify the receipt of an uplink. The search result can be exported to a CSV file.

# 6 Study of relay message sequences

The Wireless Logger facilitates the visualization of the relay function by allowing inspection of message sequences and their contents. The subsequent sections will examine the relay join procedure, uplink forwarding, and LNS MAC commands facilitated by the relay function.

## 6.1 Relayed join procedure

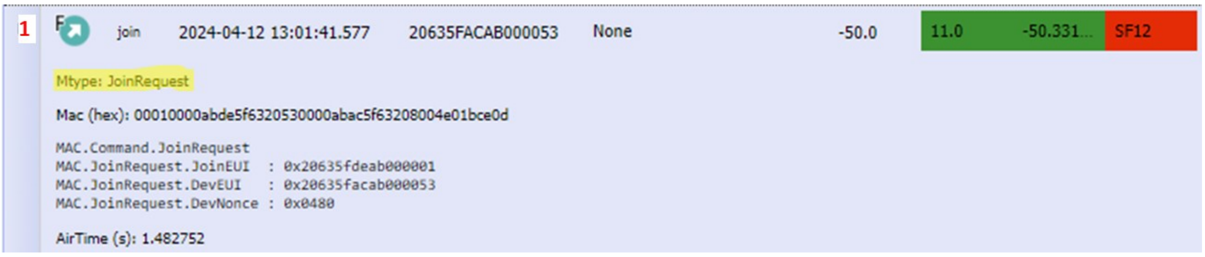
Figure 6.1 outlines the message flow following a join request message initiated by the end node, marked as step number one. The uplink symbol is labeled with the letter F, indicating that this message was forwarded through a relay. The "Relay Id" column displays the identifier of the relay responsible for forwarding the message. Step two involves the relay forwarding the request to the LNS. By calculating the RelayFwdDelay, depicted in Figure 3.2, as the difference between the timestamps of the first and second messages, arriving at 50ms, consistent with the default value specified in Table 3.2. Additionally, the relay employs frame port 226 for message forwarding. In step three, the LNS transmits the encapsulated join accept, initially as a downlink forward message via frame port 226 to the relay. Observing the timing of the third message, it is transmitted two seconds after the second message, utilizing the relay RX2 window, as illustrated in Figure 3.2. Step four involves the LNS sending a MAC command to the relay to update its internal list of trusted end devices, a topic elaborated in chapter 6.3. The relay responds to this command with an uplink in step five. The final message is the relay-forwarded join accept message. The timing indicates that the message is sent 18 seconds after the initial join request message, aligning with the RxrDelay illustrated in Figure 3.2. Consequently, the relay utilizes the RXR window of the end device, matching the default value specified in Table 3.2.



| | | | Local Timestamp | DevEUI | FPort | FCnt ⇑ | NFCnt ⇓ | RSSI | SNR | ESP | SF/DR | Relay Id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | F↘ | join | 2024-04-12 13:01:59.577 | 20635FACAB000053 | None | | | | | | SF12 | 20635FACAB000052 |
| 5 | ↗ | mac | 2024-04-12 13:01:52.565 | 20635FACAB000052 | 0 | 5 | | -42.0 | 10.0 | -42.413... | SF12 | |
| 4 | ↘ | mac | 2024-04-12 13:01:48.412 | 20635FACAB000052 | 0 | | 2 | | | | SF9 | |
| 3 | ↘ | data | 2024-04-12 13:01:43.627 | 20635FACAB000052 | 226... | | 1 | | | | SF9 | |
| 2 | ↗ | data | 2024-04-12 13:01:41.627 | 20635FACAB000052 | 226... | 3 | | -42.0 | 12.75 | -42.224... | SF12 | |
| 1 | F↗ | join | 2024-04-12 13:01:41.577 | 20635FACAB000053 | None | | | -50.0 | 11.0 | -50.331... | SF12 | 20635FACAB000052 |

Figure 6.1 Message sequence following a LoRaWAN join procedure through a relay. Timing added for illustrating values detailed in Table 3.2 and Figure 3.2.

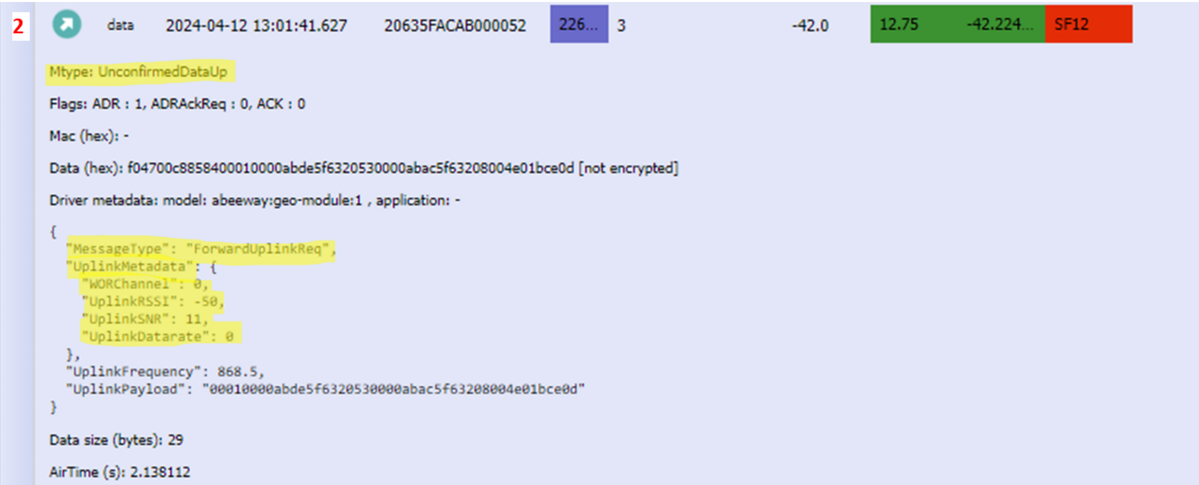
Figure 6.2 shows the package content of the join request, where message type is highlighted. This is a standard LoRaWAN join request message, containing the JoinEUI, DevEUI and DevNonce. The message was picked up and forwarded by the relay and followingly unpacked by the LNS and assigned the metadata, which was appended to it by the relay, as indicated in Figure 3.7.



```
1  F  join   2024-04-12 13:01:41.577   20635FACAB000053   None             -50.0   11.0   -50.331...  SF12

Mtype: JoinRequest

Mac (hex): 00010000abde5f6320530000abac5f63208004e01bce0d

MAC.Command.JoinRequest
MAC.JoinRequest.JoinEUI   : 0x20635fdeab000001
MAC.JoinRequest.DevEUI    : 0x20635facab000053
MAC.JoinRequest.DevNonce : 0x0480

AirTime (s): 1.482752
```

Figure 6.2 Content of join request uplink message from the end device

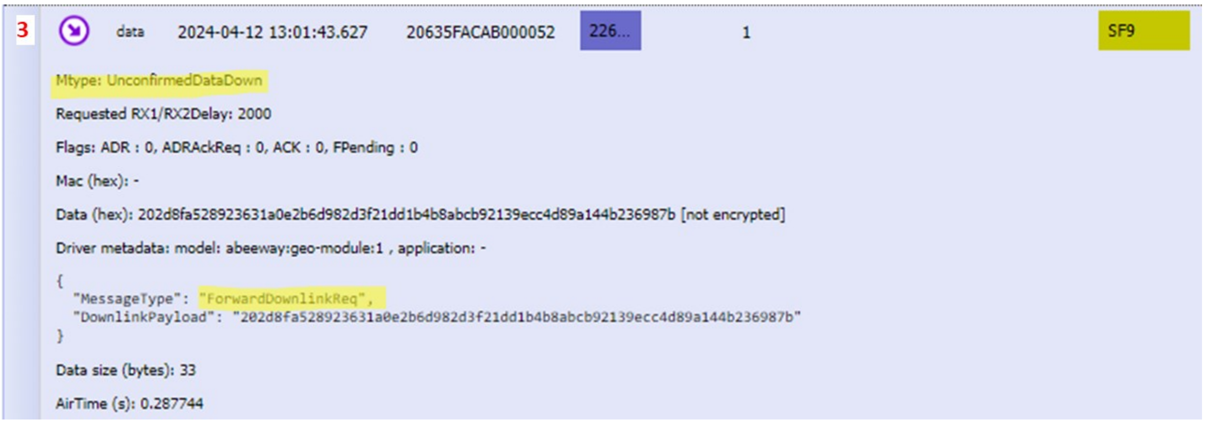
Figure 6.3 shows the package content of the uplink message forwarded by the relay. The relay treats the uplink as any other, as indicated by the message type. The label "Unconfirmed" indicates that no receipt confirmation is required. The highlighted uplink metadata is the data appended to the payload by the relay, while the metadata displayed in the list header originates from the subsequent relay transmission. The appended data shows the end node utilizing WOR channel 0 to wake up the relay. This corresponds to 865.1 MHz, 125kHz bandwidth and SF9 according to Table 3.1. Unfortunately, it is not possible to see the WOR messages in the Wireless Logger application. The remaining appended parameters shows the RSSI, SNR and the DR from the end node uplink message.



```
2     data   2024-04-12 13:01:41.627   20635FACAB000052   226...  3         -42.0   12.75   -42.224...  SF12

Mtype: UnconfirmedDataUp

Flags: ADR : 1, ADRAckReq : 0, ACK : 0

Mac (hex): -

Data (hex): f04700c8858400010000abde5f6320530000abac5f63208004e01bce0d [not encrypted]

Driver metadata: model: abeeway:geo-module:1 , application: -

{
  "MessageType": "ForwardUplinkReq",
  "UplinkMetadata": {
    "WORChannel": 0,
    "UplinkRSSI": -50,
    "UplinkSNR": 11,
    "UplinkDatarate": 0
  },
  "UplinkFrequency": 868.5,
  "UplinkPayload": "00010000abde5f6320530000abac5f63208004e01bce0d"
}

Data size (bytes): 29

AirTime (s): 2.138112
```

Figure 6.3 Content of join request uplink message, forwarded by the relay

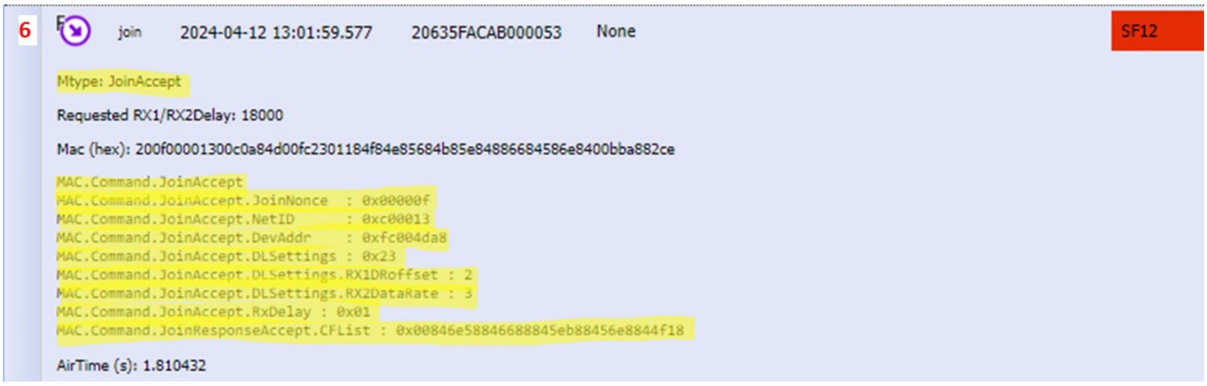
Figure 6.4 shows the package content of the downlink forward request message sent by the LNS to the relay. The message type is ForwardDownlinkReq is used to indicate that this message shall be forwarded to the end node.



```
3    data    2024-04-12 13:01:43.627    20635FACAB000052    226...    1    SF9

Mtype: UnconfirmedDataDown

Requested RX1/RX2Delay: 2000

Flags: ADR : 0, ADRAckReq : 0, ACK : 0, FPending : 0

Mac (hex): -

Data (hex): 202d8fa528923631a0e2b6d982d3f21dd1b4b8abcb92139ecc4d89a144b236987b [not encrypted]

Driver metadata: model: abeeway:geo-module:1 , application: -

{
  "MessageType": "ForwardDownlinkReq",
  "DownlinkPayload": "202d8fa528923631a0e2b6d982d3f21dd1b4b8abcb92139ecc4d89a144b236987b"
}

Data size (bytes): 33

AirTime (s): 0.287744
```

Figure 6.4 Content of LNS downlink reply message to the relay

Figure 6.5 shows the package content of the join accept downlink forwarded by the relay to the end node. The message contains the JoinNonce issued by the join server, the NetID, the DevAddr assigned to the end node, downlink configurations DLSettings, the RxDelay and CFList which is a list of network parameters. The NetID is a specific network identifier of the LoRaWAN network which is assigned by the LoRa Alliance. The DLSettings specifies the offset between the uplink and downlink DR of the RX1 window, where the offset is given by the mapping in Table 14 in RP002. The RX2 window uses a fixed frequency and DR which by default per RP002 in the EU868 band is 869.525 MHz and DR0. The RX2DataRate field changes the default data rate of the RX2 window. The RxDelay specifies the delay before the end node is to open the RX1 receive window, in seconds. The relay uses spreading factor 12 to transmit the message, which is the same as was used by the relay for the join request uplink.

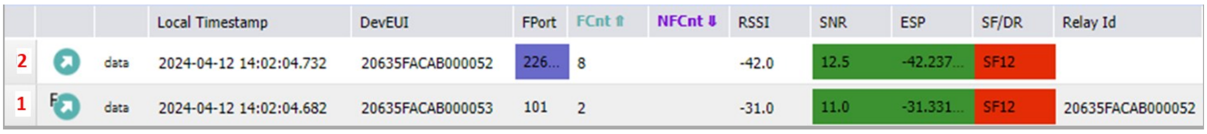


```
6    join    2024-04-12 13:01:59.577    20635FACAB000053    None    SF12

Mtype: JoinAccept

Requested RX1/RX2Delay: 18000

Mac (hex): 200f00001300c0a84d00fc2301184f84e85684b85e84886684586e8400bba882ce

MAC.Command.JoinAccept
MAC.Command.JoinAccept.JoinNonce  : 0x00000f
MAC.Command.JoinAccept.NetID      : 0xc00013
MAC.Command.JoinAccept.DevAddr    : 0xfc004da8
MAC.Command.JoinAccept.DLSettings : 0x23
MAC.Command.JoinAccept.DLSettings.RX1DRoffset : 2
MAC.Command.JoinAccept.DLSettings.RX2DataRate : 3
MAC.Command.JoinAccept.RxDelay : 0x01
MAC.Command.JoinResponseAccept.CFList : 0x00846e58846688845eb88456e8844f18

AirTime (s): 1.810432
```

Figure 6.5 Content of join accept downlink message to the end device, forwarded by the relay

## 6.2 Relayed uplink message

When the relay is used to forward a standard uplink, the procedure is like the one for handling join requests. Figure 6.6 shows the sequence of messages, starting with the uplink issued from the end node. Secondly the message is picked up by the relay, encapsulated and forwarded to the LNS through frame port 226 after the 50ms delay.



| | | | Local Timestamp | DevEUI | FPort | FCnt ⇑ | NFCnt ⇓ | RSSI | SNR | ESP | SF/DR | Relay Id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | ↗ | data | 2024-04-12 14:02:04.732 | 20635FACAB000052 | 226... | 8 | | -42.0 | 12.5 | -42.237... | SF12 | |
| 1 | F↗ | data | 2024-04-12 14:02:04.682 | 20635FACAB000053 | 101 | 2 | | -31.0 | 11.0 | -31.331... | SF12 | 20635FACAB000052 |

Figure 6.6 Message sequence following a LoRaWAN uplink through a relay

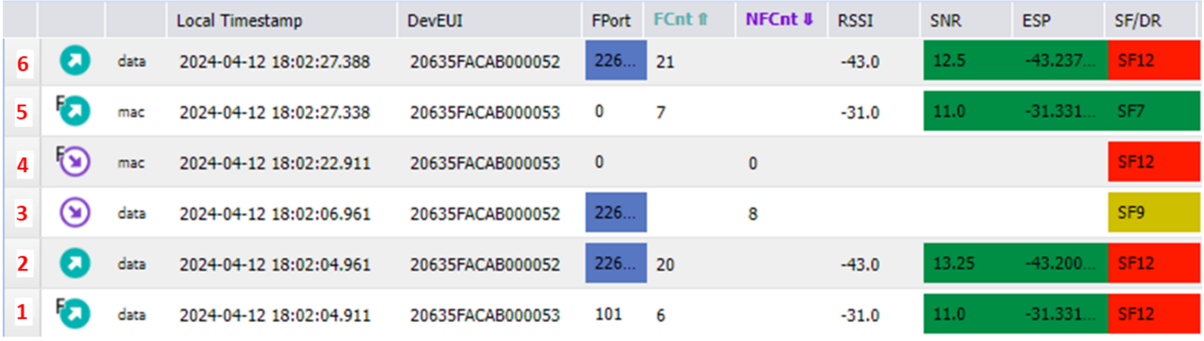
## 6.3 Relayed LNS MAC commands

The downlink MAC commands from the LNS provides useful information to analyse the network performance. As both the end node and the relay are class A LoRaWAN devices, the LNS must await an uplink before being able to utilize the receive windows of the end devices.

### 6.3.1 Link ADR command

Figure 6.7 shows the message sequence of a LinkADRReq MAC command issued by the LNS to the end node. This MAC command is available when Adaptive Data Rate (ADR) is active and is used when the LNS wants the end node to update its transmission parameters. The connectivity plan used for this project is configured to use ADR version 3. This is a PER based ADR optimization, where the target is set to 10%.

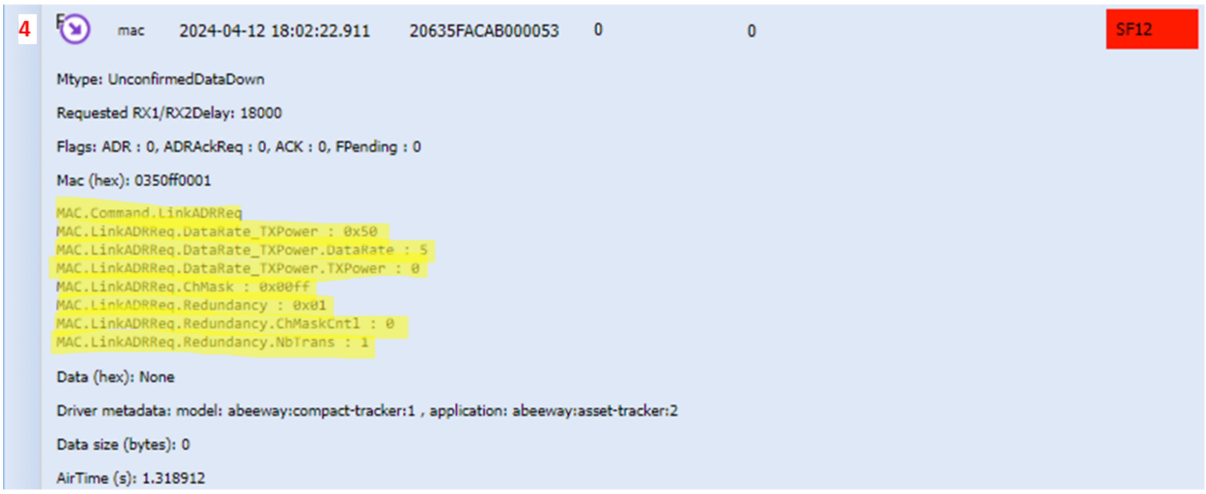
The message sequence is started by the end node issuing a standard uplink which is forwarded to the LNS by the relay through port 226. The LNS replies by issuing a ForwardDownlinkReq message through the same port, utilizing the RX2 window of the relay. Next the relay forwards the downlink to the end node using the same SF as the original uplink. In the fifth step, the end node answers the LNS with a LinkADRAns message. The spreading factor has now been changed to SF7. The last step in the sequence is the relay forwarding the reply to the LNS.



| | | | Local Timestamp | DevEUI | FPort | FCnt ⇑ | NFCnt ⇓ | RSSI | SNR | ESP | SF/DR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | ↗ | data | 2024-04-12 18:02:27.388 | 20635FACAB000052 | 226... | 21 | | -43.0 | 12.5 | -43.237... | SF12 |
| 5 | F↗ | mac | 2024-04-12 18:02:27.338 | 20635FACAB000053 | 0 | 7 | | -31.0 | 11.0 | -31.331... | SF7 |
| 4 | F↘ | mac | 2024-04-12 18:02:22.911 | 20635FACAB000053 | 0 | | 0 | | | | SF12 |
| 3 | ↘ | data | 2024-04-12 18:02:06.961 | 20635FACAB000052 | 226... | | 8 | | | | SF9 |
| 2 | ↗ | data | 2024-04-12 18:02:04.961 | 20635FACAB000052 | 226... | 20 | | -43.0 | 13.25 | -43.200... | SF12 |
| 1 | F↗ | data | 2024-04-12 18:02:04.911 | 20635FACAB000053 | 101 | 6 | | -31.0 | 11.0 | -31.331... | SF12 |

Figure 6.7 Message sequence of a LinkADRReq MAC command from the LNS to the end node through the relay

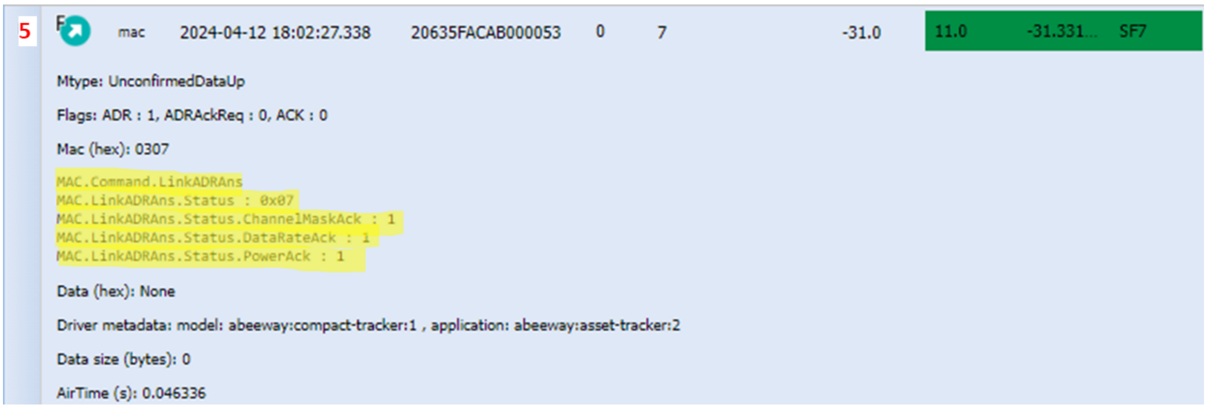
Figure 6.8 shows the package content of the LNS LinkADRReq MAC command. It specifies the data rate and maximum allowed transmission power to be used by the end node. It also specifies the channel mask and redundancy parameters. The channel mask defines the channels available to the end node for future uplinks. The redundancy parameters covers the ChMaskCntl and NbTrans fields, where ChMaskCntl is a region specific interpretation of the previously defined channel mask parameter defined in RP002 while NbTrans defines the number of redundant uplink message transmissions for each uplink.



```
4   F   mac   2024-04-12 18:02:22.911   20635FACAB000053   0          0          SF12

Mtype: UnconfirmedDataDown

Requested RX1/RX2Delay: 18000

Flags: ADR : 0, ADRAckReq : 0, ACK : 0, FPending : 0

Mac (hex): 0350ff0001

MAC.Command.LinkADRReq
MAC.LinkADRReq.DataRate_TXPower : 0x50
MAC.LinkADRReq.DataRate_TXPower.DataRate : 5
MAC.LinkADRReq.DataRate_TXPower.TXPower : 0
MAC.LinkADRReq.ChMask : 0x00ff
MAC.LinkADRReq.Redundancy : 0x01
MAC.LinkADRReq.Redundancy.ChMaskCntl : 0
MAC.LinkADRReq.Redundancy.NbTrans : 1

Data (hex): None

Driver metadata: model: abeeway:compact-tracker:1 , application: abeeway:asset-tracker:2

Data size (bytes): 0

AirTime (s): 1.318912
```

Figure 6.8 Content of LinkADRReq MAC command from LNS to end node

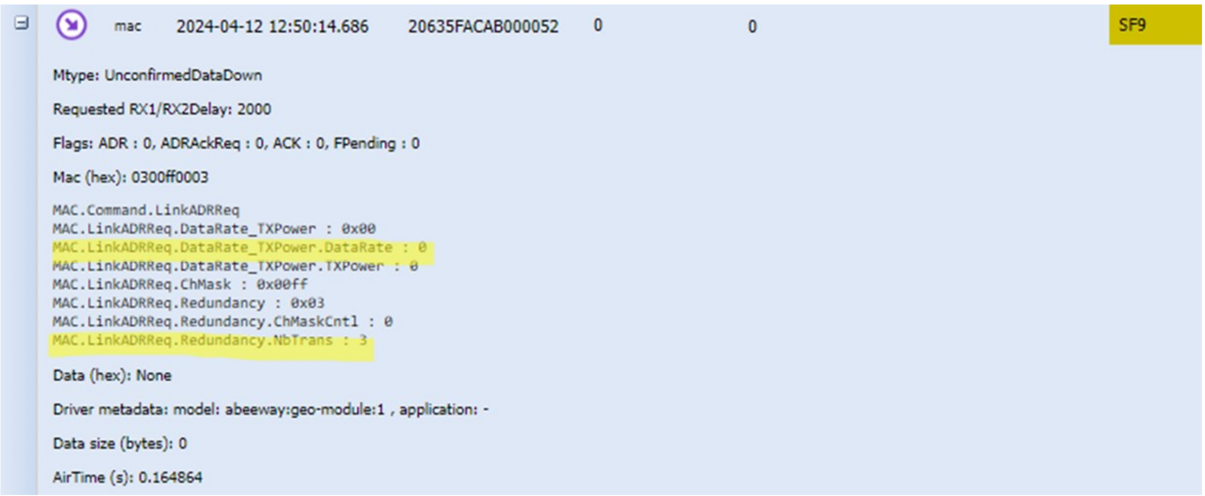
Figure 6.9 shows the package content of the LinkADRAns message from the end node. This covers the status from the end node following the request for changing the ADR parameters. The status covers acknowledge of the channel mask, data rate and power settings. As the LNS requested the data rate changed to DR5, which as per Table 2.1 corresponds to SF7, the SF of the reply message is changed accordingly.



```
5   F   mac   2024-04-12 18:02:27.338   20635FACAB000053   0   7          -31.0   11.0   -31.331...   SF7

Mtype: UnconfirmedDataUp

Flags: ADR : 1, ADRAckReq : 0, ACK : 0

Mac (hex): 0307

MAC.Command.LinkADRAns
MAC.LinkADRAns.Status : 0x07
MAC.LinkADRAns.Status.ChannelMaskAck : 1
MAC.LinkADRAns.Status.DataRateAck : 1
MAC.LinkADRAns.Status.PowerAck : 1

Data (hex): None

Driver metadata: model: abeeway:compact-tracker:1 , application: abeeway:asset-tracker:2

Data size (bytes): 0

AirTime (s): 0.046336
```

Figure 6.9 Content of LinkADRAns MAC reply from end node

The LinkADRReq MAC command is also used to control the relays uplink transmission parameters. The message is shown in Figure 6.10. Two parameters worth noticing, which differs from the end node, is the data rate and the number of redundant transmissions. The data rate is specified to be kept at DR0, even though the transmission parameters indicates that a higher SF should be used. The NbTrans is set to three, meaning each uplink will be transmitted three times. This improves the quality of service of the relay, leading to a lower PER at the cost of an increase in both battery consumption and airtime.

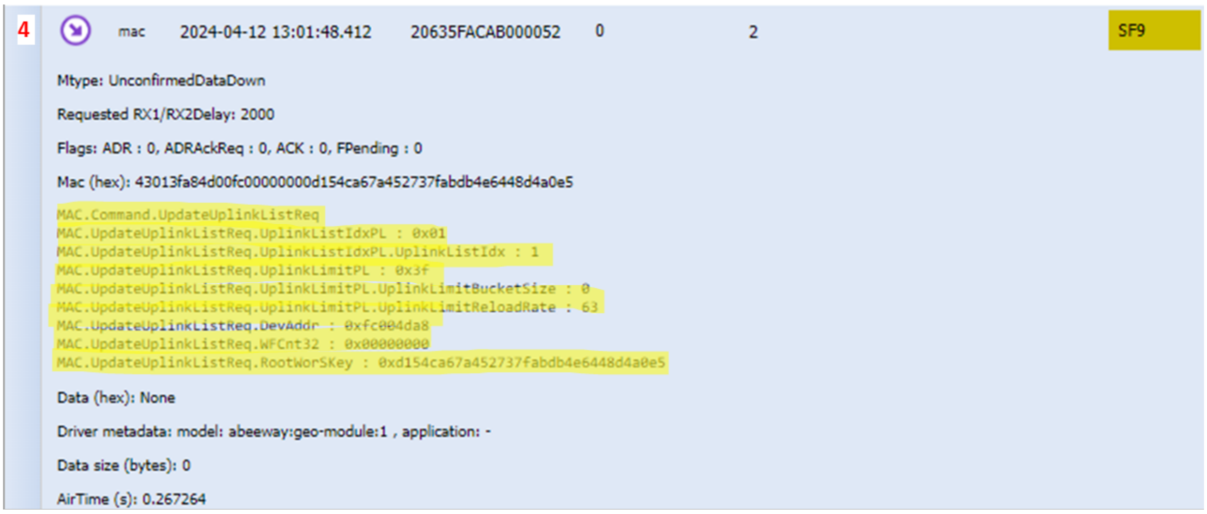


```
mac     2024-04-12 12:50:14.686     20635FACAB000052     0          0          SF9

Mtype: UnconfirmedDataDown

Requested RX1/RX2Delay: 2000

Flags: ADR : 0, ADRAckReq : 0, ACK : 0, FPending : 0

Mac (hex): 0300ff0003

MAC.Command.LinkADRReq
MAC.LinkADRReq.DataRate_TXPower : 0x00
MAC.LinkADRReq.DataRate_TXPower.DataRate : 0
MAC.LinkADRReq.DataRate_TXPower.TXPower : 0
MAC.LinkADRReq.ChMask : 0x00ff
MAC.LinkADRReq.Redundancy : 0x03
MAC.LinkADRReq.Redundancy.ChMaskCntl : 0
MAC.LinkADRReq.Redundancy.NbTrans : 3

Data (hex): None

Driver metadata: model: abeeway:geo-module:1 , application: -

Data size (bytes): 0

AirTime (s): 0.164864
```

Figure 6.10 Content of LinkADRReq MAC command from LNS to relay

### 6.3.2 Uplink list command

To inform the relay of the new trusted end device following the relayed join accept procedure, the LNS issues an UpdateUplinkListReq MAC command. This updates the relays internal list of trusted end devices, adding the recently approved new node. The message content of this MAC command is shown in Figure 6.11. It specifies the specific list to be updated with the UplinkListIdx field, the uplink payload limitations with the UplinkLimitPL field, the address of the new node, the WOR frame counter and RootWorSKey. The UplinkLimitPL field consists of the bucket size and reload rate parameters following the forward limitation function which was introduced in chapter 3.4. As UplinkLimitReloadRate is set to 63, this function is bypassed, and all valid uplinks will be forwarded. The RootWorSKey is the relay session key as described in chapter 3.3, forwarded to the relay by the LNS.



```
4   mac     2024-04-12 13:01:48.412     20635FACAB000052     0          2          SF9

Mtype: UnconfirmedDataDown

Requested RX1/RX2Delay: 2000

Flags: ADR : 0, ADRAckReq : 0, ACK : 0, FPending : 0

Mac (hex): 43013fa84d00fc00000000d154ca67a452737fabdb4e6448d4a0e5

MAC.Command.UpdateUplinkListReq
MAC.UpdateUplinkListReq.UplinkListIdxPL : 0x01
MAC.UpdateUplinkListReq.UplinkListIdxPL.UplinkListIdx : 1
MAC.UpdateUplinkListReq.UplinkLimitPL : 0x3f
MAC.UpdateUplinkListReq.UplinkLimitPL.UplinkLimitBucketSize : 0
MAC.UpdateUplinkListReq.UplinkLimitPL.UplinkLimitReloadRate : 63
MAC.UpdateUplinkListReq.DevAddr : 0xfc004da8
MAC.UpdateUplinkListReq.WFCnt32 : 0x00000000
MAC.UpdateUplinkListReq.RootWorSKey : 0xd154ca67a452737fabdb4e6448d4a0e5

Data (hex): None

Driver metadata: model: abeeway:geo-module:1 , application: -

Data size (bytes): 0

AirTime (s): 0.267264
```

Figure 6.11 Content of UpdateUplinkListReq MAC command from LNS to relay

# 7 LoRaWAN relay use cases

The relay feature was developed for two primary use cases, network expansion to cover nodes on the edges of an existing network, and for extending the network to places that are hard to reach within an existing network, such as deep indoor or subterrain areas [43]. Focusing on being a simpler and more cost-efficient method of increasing the reach of an existing LoRaWAN network, the relay adds a new tool to network operator's toolbox compared to the traditional method of installing new gateways. Installing a new gateway is not always feasible, as it will require both a power supply and a solution for data backhaul. Facilitating for these requirements can come at a considerable capital expenditure for getting the gateway operational. Thereafter it will lead to an increase in the networks overall operational expenditures, as the gateway will require management and fees related to backhaul and power consumption. Installing a relay on the other hand, will only require that its placement be within the coverage of the existing network and that the connecting end nodes and managing LNS is compatible with the TS011 specification.

The first commercial implementation featuring the use of relays involve the four companies Semtech, Actility, Deviceroy and Arad [44]. In this collaboration, Semtech provided their relay enabled LBM stack, Actility provided their relay enabled LNS and Deviceroy their Aria relay to provide an end-to-end interoperability tested solution. Arad, which is a company specializing in water metering, is ready to implement this function as part of their water metering network.

## 7.1 Altibox

As a LoRaWAN network operator, Altibox stands to benefit from the potential of relays across various use cases. By continually expanding their network, relays offer a cost-effective solution for both new and existing network deployments. In any wireless network rollout, certain areas may experience coverage gaps due to distance from the nearest gateway or obstacles obstructing RF line of sight between the gateway and end node. It falls upon network facilitators like Altibox to ensure that their coverage aligns with the needs of end clients. These requirements can vary based on application, but shared challenges include providing reception in deep-indoor locations such as basements or within buildings constructed by use of signal-attenuating materials. Furthermore, subterranean locations such as utility holes or natural obstacles like hills, which obstruct coverage, can present additional challenges. For instance, ensuring connectivity to a lake situated on the opposite side of a hill can be particularly problematic. Urban environments, with their dense structures, present their own set of challenges, making it difficult to achieve adequate signal coverage despite strategically placed gateways.

Utilizing relays to attain extensive network coverage customized to meet the requirements of end clients can effectively manage costs and enable opportunities previously deemed unfeasible. In the instances mentioned above, relays can be strategically positioned closer to end nodes while remaining within the range of the existing gateway. This approach expands the reach of the LoRaWAN network simply by deploying small, battery-operated remote nodes.

## 7.2 Porsgrunn municipality

Porsgrunn municipality has initiated a smart city initiative, aiming to integrate IoT solutions to enhance the city's efficiency and sustainability, improving the well-being of its residents. Several projects have already been initiated, ranging from remotely monitoring waste bins to automatically measuring bathing temperatures. Remote monitoring of waste bins involves LoRaWAN end nodes that measure the fill level of each container. The collected data is utilized by the municipality's waste management service, in this case managed by "Skjærgårdstjenesten", enabling them to make informed decisions regarding garbage collection schedules [45]. This initiative conserves fuel, leading to a reduction in the service's environmental impact and operational costs, especially given that these bins are in remote areas. Additionally, it improves workforce usage, freeing up time that can be allocated to other tasks.

The municipality is actively seeking additional initiatives, primarily within the smart city and buildings sectors. As an existing user of LoRaWAN networks, introducing relays in their network can help to complement current network deployments and extend coverage to hard-to-reach areas. For instance, Porsgrunn municipality is using LoRaWAN sensors to monitor stormwater levels in subsurface sumps. However, due to insufficient coverage from the existing LoRaWAN network, a relay can be deployed near the utility hole, such as being affixed to a nearby light pole, effectively expanding the network range to this challenging location.

There are numerous other potential applications where the relay can prove valuable. In the city of Porsgrunn, for example, municipal gardeners are responsible for planting and maintaining flower beds and urns throughout various locations. To ensure optimal plant growth, especially during dry summer periods, moisture sensors could be strategically inserted into the soil. These sensors could then alert gardeners when watering is needed. Particularly in the case of urns, which may be situated in areas with limited LoRaWAN coverage, such as small squares surrounded by tall buildings, relays can be deployed. Placed at elevated positions within existing coverage, these relays effectively extend coverage to previously inaccessible areas. Expanding on this application, a LoRaWAN actuator could remotely activate the water hose, automating the entire process.

## 7.3 Mobile offshore drilling unit

A "Mobile Offshore Drilling Unit" (MODU) is a large floating vessel constructed, as other marine vessels, by use of thick steel plates and structures. The main eye-catching element, which makes the MODU stand out in comparison to other marine vessels, is the derrick. A tall metal structure, usually centralized on the vessel and cladded with metal plates to shield against weather. This is the centre of operations on the rig, where the drilling machine is hoisted and lowered as new wells are being produced. Given its structural design, which is not favourable to radio communication, most onboard systems are hardwired, with some exceptions. One exception to the hardwired systems is the use of UHF radios for internal communication. Whip antennas are deployed on the open deck to provide UHF coverage for outdoor areas. Inside the vessel, long leaky feeder cable networks are installed to ensure adequate reception in challenging areas.

A more recent addition to the onboard systems is the implementation of Wi-Fi coverage in machinery and open deck areas. With the increasing digitization of the world, providing Wi-Fi coverage in working areas has become a key step towards streamlining maintenance tasks and operations. Live dashboards can now offer a comprehensive overview of all ongoing work onboard, enhancing both the permit-to-work and computerized maintenance systems. This makes it easier for operators to report any potential faults and update the status of performed work directly in the field.

With Wi-Fi already established onboard, integrating a LoRaWAN network becomes more feasible. LoRaWAN gateways can utilize the existing Wi-Fi network as a backbone or connect to the established Ethernet infrastructure and thereby be able to utilize power over Ethernet which will reduce the installation cost. Depending on the specific use case, relays can contribute to ensuring adequate onboard network coverage.

There are several applications in which LoRaWAN can be utilized within this sector, some of which will be explored in the following sections. A central common requirement across these applications is the need for explosion-protected field equipment, which can significantly influence the total costs.

### 7.3.1 Utilizing LoRaWAN for condition based maintenance

Equipment maintenance is a critical aspect related to safety and cost. It typically falls into two main categories: preventive and corrective maintenance. Preventive maintenance is carried out proactively before a component or equipment experiences a failure, whereas corrective maintenance occurs in response to a failure.

Upon closer examination of preventive maintenance, the schedule typically follows an interval-based approach, which can be either time-based (weekly, monthly, etc.) or usage-based (e.g., running hours or cycles). This type of maintenance adheres to manufacturer recommendations or industry best practices, involving a set of tasks to be performed at specified intervals. These intervals are generalized and does not necessarily represent the best practice for each installation. Several factors are important, such as environmental conditions and use pattern.

An alternative approach, known as condition-based maintenance, addresses this challenge. By assessing the actual state of a machine, operators can determine whether it is operating optimally or in need of maintenance. Moreover, by analysing multiple measurements, operators can predict when maintenance will be required. This approach is known as predictive maintenance. The principle was initially applied for rotating machinery but can be extended to other equipment, including monitoring leaks on valves and flanges, which are abundant on a MODU. LoRaWAN sensors offer a viable solution for such monitoring tasks, measuring ultrasound and vibration alongside environmental factors like ambient temperature, pressure, and humidity, or equipment-specific parameters such as power consumption. Through the application of classification machine learning techniques, real-time measurements can be compared with historical time and frequency domain data to classify equipment states as either in good condition, requiring no immediate action, in a warning state, indicating the need for additional attention, or critical, signifying that a failure has occurred or is imminent.

Condition based maintenance often relies on sensors that measure frequency spectrums, resulting in large data samples. When working with LoRaWAN, this can be challenging, as discussed in a whitepaper by STMicroelectronics [46]. The primary issue is the required payload size for frequency spectrum data, which quickly exceeds the maximum allowable regional limits. One workaround is segmenting the data into several packets, but this introduces new challenges in recovering and retransmitting lost packets. The whitepaper proposes a solution that involves adding only two additional fields per uplink to track individual and total amount of fragments. Once the transmission is complete, the end node sends a downlink request message to the LNS, which can acknowledge packet receipt and request retransmission of any missing fragments.

### 7.3.2 Utilizing LoRaWAN for personnel tracking

In an alternative application, LoRaWAN could be employed for personnel tracking onboard, although this topic requires sensitivity. Nonetheless, it offers safety-related benefits. Small tracking devices could implement geofencing methods to alert personnel when entering restricted areas or remaining in noise hazard zones for extended periods. Moreover, these devices could improve safety by enabling more effective tracking of personnel during emergencies. For instance, individuals could promptly notify the control room of an incident by activating a button on the tracker, providing their location simultaneously. Likewise, in search and rescue or muster situations, the tracker system could provide a swift overview of the whereabouts of all onboard personnel.

### 7.3.3 Utilizing LoRaWAN to streamline routine inspections

Routine inspections constitute a considerable portion of the daily tasks for the technical personnel onboard a MODU. Leveraging LoRaWAN sensors, many of these checks can be automated.

Consider the case of battery-powered emergency light fixtures. Regulations stipulate that these lights must be self-sustaining for a minimum of 30 minutes after a power loss. Testing these fixtures involves isolating the circuit breaker for one loop, conducting a verification round to ensure proper function, then waiting for 30 minutes before conducting a second verification round. However, this process becomes cumbersome with hundreds of circuits to test, requiring personnel to possess knowledge of circuit routing. Additionally, identifying the root cause of failures poses a challenge. Without having details about the timing, it is difficult to identify whether it is the battery, driver, or control circuit that is faulty.

LoRaWAN sensors measuring lux or temperature within each fixture could pose a solution. This data allows for the evaluation of each fixture’s performance, reducing labour efforts and aiding in the identification of faults. By analysing this information, personnel can pinpoint the root cause of failures more efficiently. Simple dashboard solutions can display failure rates and quickly highlight fixtures that do not pass the test.

Another valuable application is the integration of LoRaWAN temperature sensors in all battery rooms and compartments. These batteries typically supply uninterruptible power supplies, which is essential for powering safety-critical equipment during main power outages. The lifetime of lead-acid batteries is closely tied to ambient temperature, with higher temperatures leading to a reduction in expected lifetime. Thus, monitoring ambient temperature closely is crucial for ensuring battery function.

Currently, manual temperature readings are taken during routine inspections, but this method lacks insight into temperature trends between consecutive measurements. By leveraging LoRaWAN for this application, trends can be continuously monitored and compared against battery specifications. This not only enhances confidence in the integrity of the safety barrier but also has the potential to save costs by enabling proactive measures before battery damage occurs.

### 7.3.4 Utilizing LoRaWAN for earth fault prevention

Earth faults pose a common challenge on marine vessels, occurring when an unintended connection is established between a live conductor and the ship's hull. If left unaddressed, these faults can result in electrical shock hazards, equipment damage, and even fires. Consequently, significant effort is dedicated to detecting and rectifying such faults.

Various factors can contribute to earth faults, with water ingress in junction boxes being a frequent cause. In this context, LoRaWAN moisture and humidity sensors can be utilized to detect water ingress before it escalates into a problem. Certain onboard systems are more susceptible to such issues, often due to their location, which may expose them to rough weather or frequent cleaning with high-pressure washers. Targeting these systems could be a low-hanging fruit for improving safety as well as reducing time spent on fault-finding onboard a MODU.

# 8 Hardware used for testing relay functionality

This chapter describes the equipment used for performing the subsequent tests.

## 8.1 Base station

The gateway used is a UfiSpace Pico Cell Enterprise, model number GEE810E-868U [47]. This is an indoor graded gateway, powered by either 12V adapter from mains or 48V PoE. For the tests conducted, the 12V mains option has been used. The gateway offers three different options for backhaul: ethernet, Wi-Fi or LTE. It was not delivered with LTE as an option, as this requires a separate USB dongle with a SIM card for accessing the licensed LTE bands. The chosen method for backhaul was ethernet, as patch ports are abundant in the office spaces. Due to the IT security policy, a separate VLAN was configured for selected port locations within the building. This to segregate the traffic from internal to the company and simplify firewall implementation. As this was time consuming to establish, due to high workload and absence of required personnel in the IT department, a solution with Wi-Fi as backhaul were also investigated. It turned out that the connection between the LNS and the gateway was not configured for this method, hereby requiring some work by the software service provider to enable this option. It did take a couple of weeks before addressing this, yet after enabling, the gateway does seamlessly change-over between ethernet and Wi-Fi if the main channel drops out. The gateway uses the StrongSwan VPN implementation for securing the backhaul communication, which is an is an OpenSource IPsec-based VPN solution [48].

As stated by the model's name, the gateway is configured for the EU868 ISM band. It has an internal LoRaWAN antenna of 1,4dBi gain, supporting up to 8 LoRa channels. External antenna can be installed but is not supplied for this test. The maximum EIRP set to 14dBm, which is according to the EU rules for the 868 ISM band [17, Tbl. 10].

## 8.2 Field test device

The field test device is a LoRaWAN node specifically designed to map out the radio coverage of a LoRaWAN network. It is manufactured by Adeunis and the model number is ARF8123AA. This is a handheld device which is used for analysing the network, to aid in planning for both base station and relay field deployment [49]. The unit is a Class A LoRaWAN device and is battery operated. It can transmit uplink messages intermediately or by use of a push button. A handy display shows live information about the network status, including an overview of the join process, last uplink/downlink (RSSI, SNR etc.) and packet counts. It also incorporates a GPS antenna for positional coordinates, yet this function requires clear LOS to a GPS satellite hence it only works outdoors. As all tests related to the relay performance are executed indoors, this function has not been used.

The model is configured for the EU868 band. It is compliant with LoRaWAN specification v1.0.2, hence not supporting the relay functionality. The device is built with an external antenna and comes with a software function called "integrated antenna" mode. This function can be used to emulate a PCB antenna, indicating how such a device would perform in each circumstance. This function could have been utilized for the mapping but was not as the end node was available for this testing. The RF transmitted power is maximum 14dBm with a receiver sensitivity of down to -140dBm at SF12.

## 8.3 Relay evaluation kit

The relay evaluation kit consists of two devices, a LoRaWAN relay and a relay enabled end node. They are both manufactured by Abeeway under the model number is DEABE700EU. Both devices are delivered in a compact and robust waterproof IP68 casing and contains hardware which is like the Compact Tracker model. The devices are also available in IECEx or ATEX approved versions, for installation in hazardous areas. The units are both defined as Class A LoRaWAN device and are powered by 3xAA Lithium batteries.

The firmware's are based version 4.4.0 of Semtech's LBM stack and implements the LoRa Alliance specification v1.0.4 [50]. As this is an evaluation kit, the functionality of these devices has been stripped down to the core in the current firmware version. The end node application periodically transmits a demo uplink and a WOR message which can be picked up by either the relay or a gateway. The uplink payload is only containing an incremental counter, which increases by one for each transmission. The uplink can also be triggered by a Hall-effect sensor. The relay application is also configured to send an uplink every hour, or by manually triggering by a magnet.

The Hall-effect sensor is also used to activate the USB mode or to enter bootloader mode. The USB-mini socket can be uncovered by unscrewing the four small bolts holding the device housing together. The devices support a serial port command line interface, where the device parameters can be modified and LoRa commands can be issued. The LoRa relay parameters are configured according to the defaults as listed in Table 3.1 and Table 3.2. Only WOR channel 0 is active and CAD periodicity is set to 1 second.

The model is configured for the EU868 band, featuring a muRata Type 1WL (LBEU5ZZ1WL) IC, which consist of a Semtech's LR1110 radio chip which supports the relay enabled LBM [51]. The devices feature an omnidirectional PCB antenna, and the RF output power is set to 14dBm with a receiver sensitivity -139dBm at SF12.

# 9 Test site and setup

The selected site for evaluating the relay functionality is a spacious concrete office building characterized by a distinctive layout. The building comprises six wings, horizontally separated by a 12-meter open space, as illustrated in Figure 9.1. The outer walls are 48 cm thick, consisting of three layers: a 20 cm inner layer of concrete, a 20 cm middle layer of Styrofoam plates, and an 8 cm outer layer of concrete. Each wing spans 12 meters in width. The entire building stretches 138 meters, from wing one to wing six.

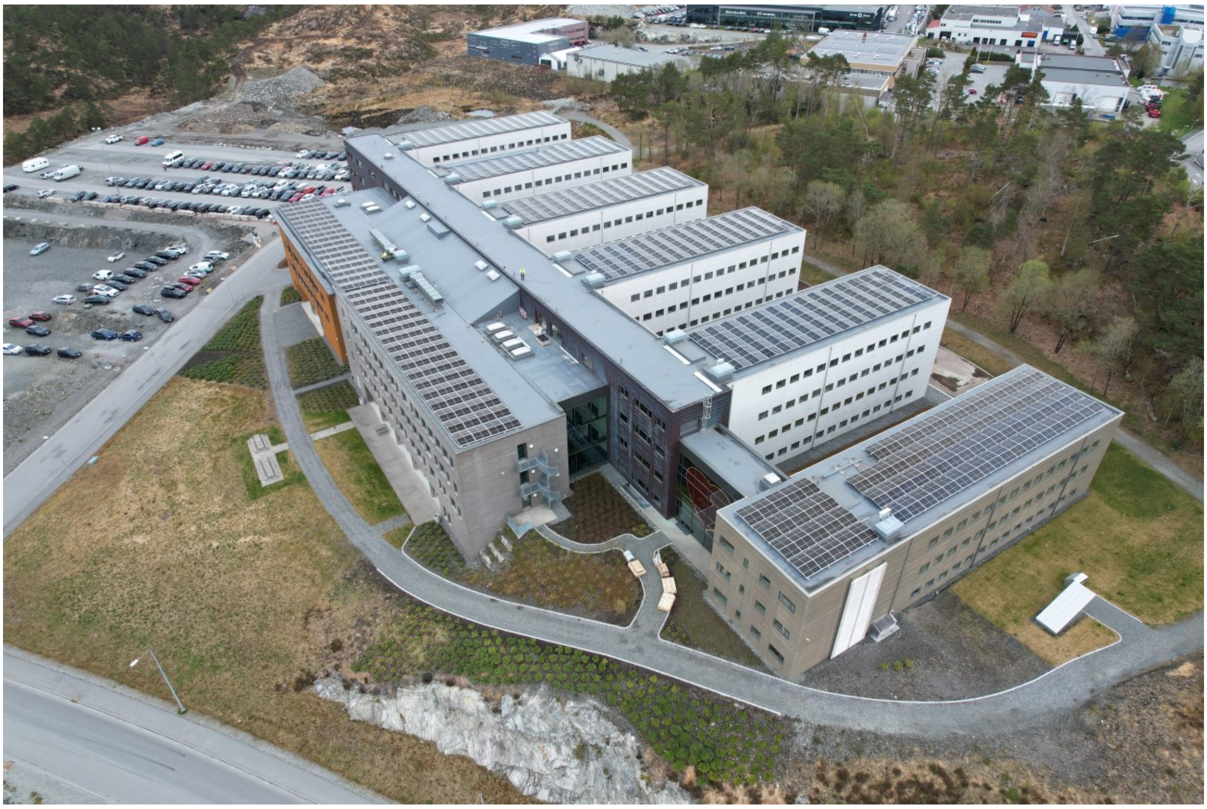


Figure 9.1 Aereal photo of the test building, picture credit to Eigil Schinnes / Solipro AS

Underneath the wings, a sizable parking garage occupies the main portion of the first floor. The building comprises five floors, apart from the sixth wing, which spans four floors. The inter-floor structures consist of 32 cm thick concrete, supplemented by a 1 cm levelling compound. The second floor, situated above the garage, incorporates an additional 20 cm of insulation and an 8 cm levelling compound.

The main load-bearing inner walls measure 20 cm thick concrete, while light walls are a composition of 70/95 mm + 4 layers of 13 mm gypsum plates. The layout and construction of this building provides a test landscape with several barriers which will attenuate the LoRa radio coverage. This will be exploited when testing the relay feature for extending the reach of deep indoor coverage.

## 9.1 Test setup

When placing the gateway, three separate locations within the building were evaluated. The first location was at the second floor where the LoRaWAN gateway is placed at the first wing as shown in Figure 9.2. The second was at the third floor where the gateway is placed in the third wing as shown in Figure 9.3. The last was at the fifth floor where the gateway is placed in the fifth wing as shown in Figure 9.4.

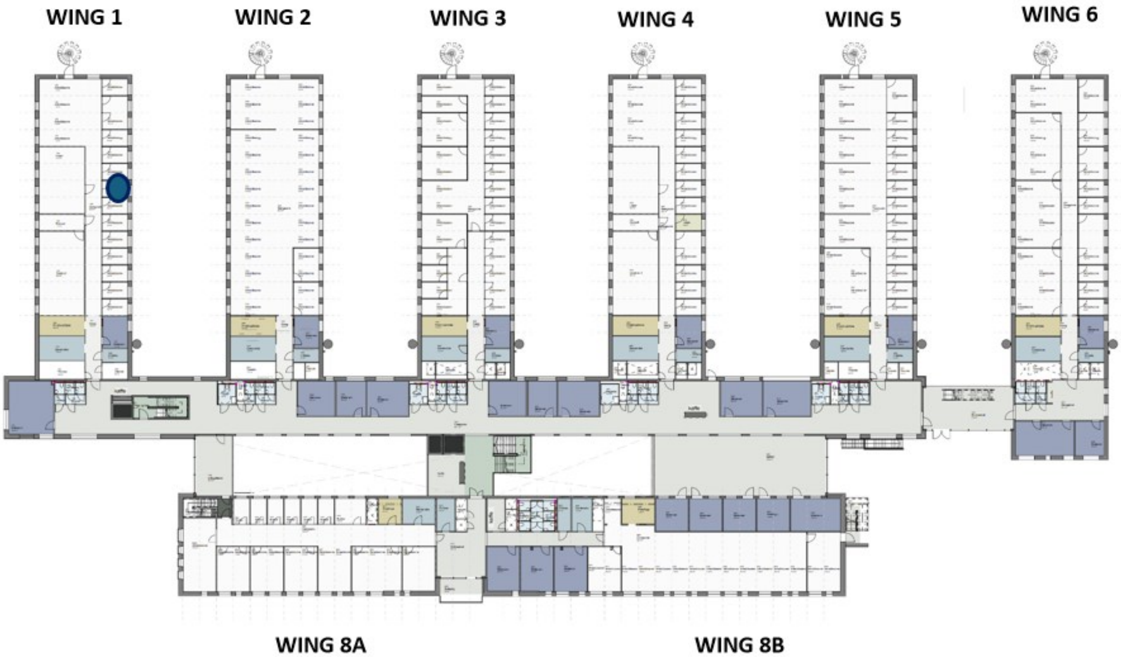


Figure 9.2 Test option one. Blue dot indicates the GW positioned at the first wing of the second floor.

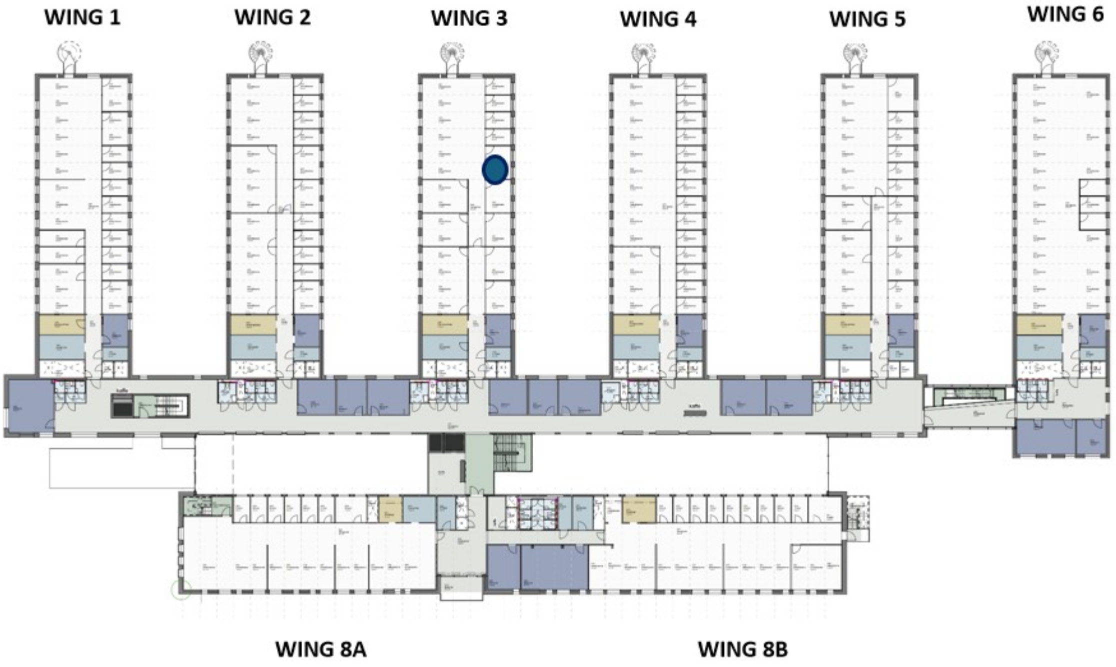


Figure 9.3 Test option two. Blue dot indicates the GW positioned at the third wing of the third floor.

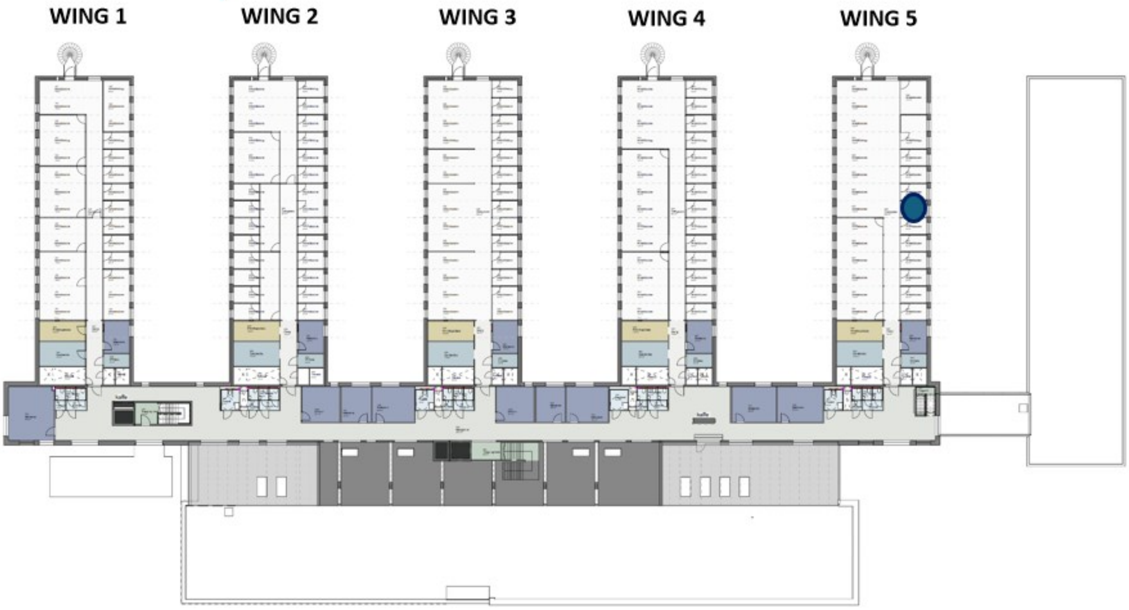


Figure 9.4 Test option three. Blue dot indicates the GW positioned at the fifth wing of the fifth floor.

In each of these scenarios, the idea was to first map out the coverage provided by the gateway by use of the field test device and the end device. These two devices are quite different, and it is therefore expected that the samples from the two devices will differ. The field test device which is designed for this specific purpose, has an external antenna and a display which gives immediate indication of the result, while the end device only holds an antenna printed on the internal circuit board.

One of the main use cases from Porsgrunn municipality were to deploy relays closer to subterranean water wells, to provide for better coverage for the LoRaWAN sensors located within the utility holes. When considering the first location, where the gateway is paced on the second floor, this provides a good comparison to such a use case if looking into the coverage in the floor below. This is the garage, and the two floors is separated by 40 cm of concrete and 20 cm isolation.

When comparing the first location to the other two, looking at the indoor coverage, these will be quite comparable when it comes to the relay functionality. Therefore, it was decided only to move further with the gateway located at the first wing of the second floor.

When mapping the coverage for this location, samples were issued at various places to give an indication of the GW range. Figure 9.5 illustrates the signal path for the packages when performing the coverage testing. A sample in this context is an uplink package from a device, where for this purpose the information of interest is the RSSI, SNR, ESP, and SF. For the field test device an uplink is issued by simply pressing a button on the device. The display will indicate the SF, transmitted effect, and SNR for the transmitted signal and similarly SF, RSSI and SNR for the received package. For the end device, a magnet is required for activating a hall-effect sensor by holding the magnet over the indicated position on the enclosure for 1 to 5 seconds. As this proved highly unreliable, the method was changed to opening the container and connecting the node to a laptop with a USB cable. This way the uplink could be issued through a command in the CLI interface. For confirmation of packet receipt, ThingPark Wireless Logger was used, and a note was taken linking the frame count with the respective sample location.

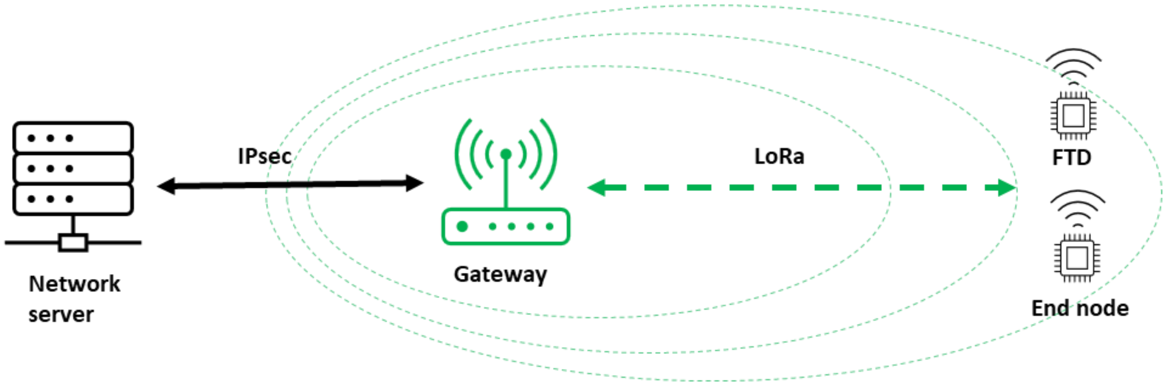


Figure 9.5 Illustration of the signal path during the mapping of gateway coverage, from end nodes through the gateway to the network server

After mapping, the measurements were exported from ThingPark as a CSV file and imported into Excel for processing and evaluation. Each sample was manually labelled to its respective location based on the noted frame counts. Before summarizing in a table, all single samples from the same device and location were averaged. To enhance clarity, the table was color-coded to highlight the most significant measurements. The colour coding is presented in Table 9.1, and is based on the ranges defined in ThingPark [16].

Table 9.1 Radio colour codes used to indicate signal quality in the tables presenting the test results [16]

| | | | | | | |
|---|---|---|---|---|---|---|
| **SNR [dB]** | ≥ -7 | -7 < x ≤ -9 | -9 < x ≤ -11.5 | -11.5 < x ≤ -14 | -14 < x ≤ -16.5 | < -16.5 |
| **ESP [dBm]** | ≥ -90 | -90 < x ≤ -100 | -100 < x ≤ -110 | -110 < x ≤ -120 | -120 < x ≤ -130 | < -130 |
| **SF** | SF7 | SF8 | SF9 | SF10 | SF11 | SF12 |

The overview of averaged samples and their respective locations will guide the identification of a suitable placement for the relay. Upon deployment of the relay in a new location, a new mapping will be conducted. Figure 9.6 depicts the altered signal path for the packets during the relay testing phase. Given that the end device is the sole node supporting relay functionality, this mapping exercise will exclusively involve this device. The objective is to determine how the relay can expand network coverage and assess its effectiveness.

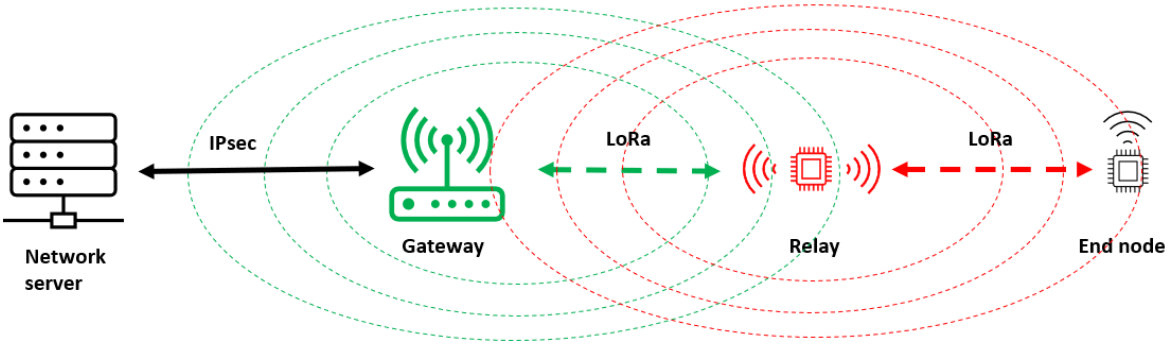


Figure 9.6 Illustration of the signal path while testing the operation of the relay, from end nodes through the relay to the gateway to the network server

# 10 Mapping of gateway coverage

A systematic method was employed to chart the LoRa coverage facilitated by the gateway. The list of sampled locations, along with their respective averaged RSSI, SNR, and ESP for each of the two nodes, is presented in Table 14.1 of Appendix C. To visually represent these locations within the building layout, Figure 10.1 and Figure 10.2 illustrate the sample points as red dots, with numbering corresponding to Table 14.1.

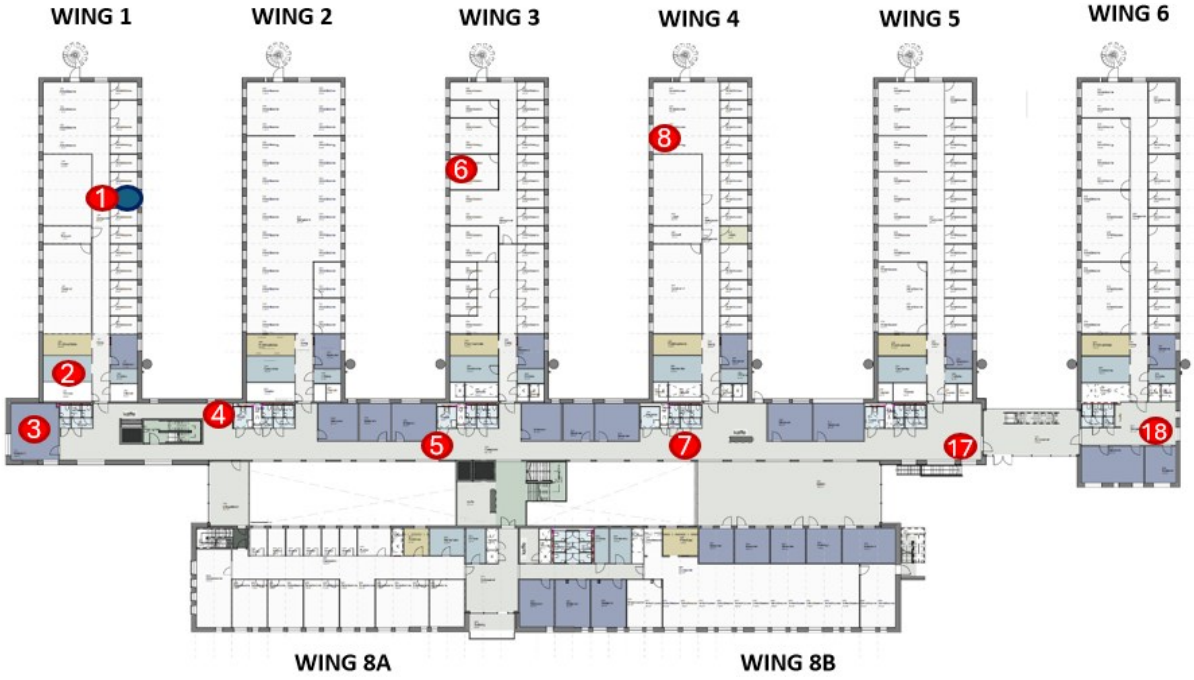


Figure 10.1 Second floor: Layout drawing indicating the locations where network samples have been taken

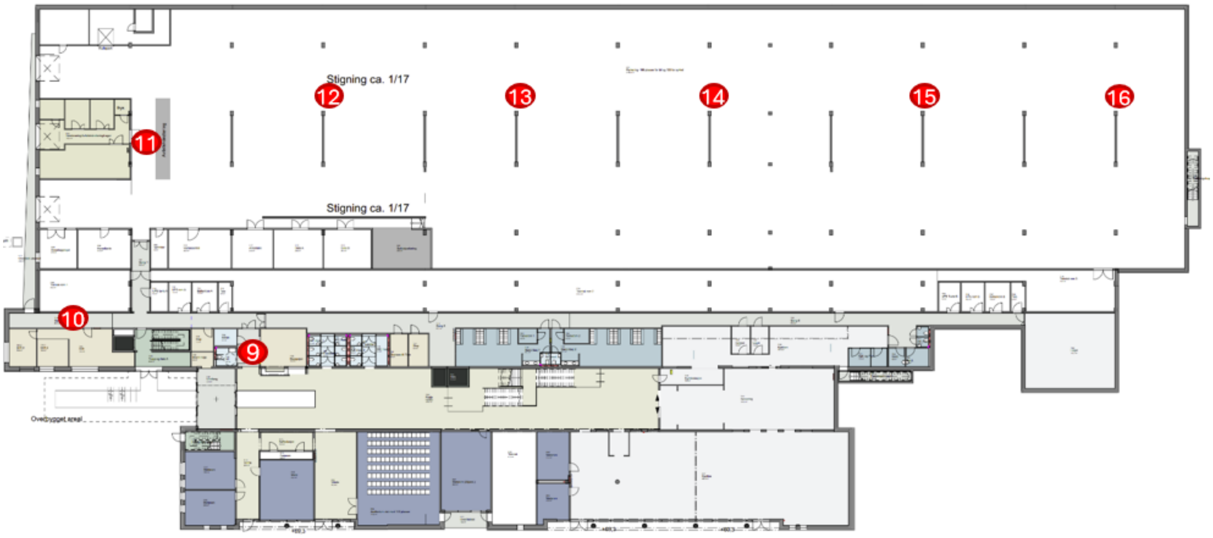


Figure 10.2 First floor: Layout drawing indicating the locations where network samples have been taken

## 10.1Identifying optimal relay positions

Examining the building layout alongside the sampled data presented in Table 10.1 reveals a trend: as one moves farther away from the initial wing, the signal strength diminishes rapidly with each successive wing. Analysis of the table indicates that the end node experiences severely degraded signal quality at location five, adjacent to the third wing, and completely loses reception at location seven, near the fourth wing. This deterioration in signal strength stems from the increase in the number of concrete walls encountered as each wing is traversed, as depicted in Figure 10.3. Naturally, this scenario poses significant challenges for effective radio communication.

Table 10.1 Average radio transmission parameters for each designated sample location during the of mapping gateway coverage. The table includes only locations relevant for the first relay position.

| Loc. | Location description | Device | RSSI [dBm] | SNR [dB] | ESP [dBm] | SF |
|---|---|---|---|---|---|---|
| 4 | Corridor by second wing second floor | FTD | -97,50 | 7,31 | -98,24 | 12 |
| | | End Node | -101,25 | 7,38 | -102,01 | 12 |
| 5 | Corridor by third wing second floor | FTD | -113,75 | -13,25 | -127,25 | 12 |
| | | End Node | -114,50 | -14,13 | -128,79 | 12 |
| 7 | Corridor by fourth wing second floor | FTD | -113,50 | -13,81 | -127,58 | 12 |
| | | End Node | NC | NC | NC | NC |
| 9 | By reception first floor | FTD | -113,25 | -17,25 | -130,58 | 12 |
| | | End Node | NC | NC | NC | NC |
| 17 | Corridor by fifth wing second floor | FTD | -111,00 | -12,25 | -123,50 | 12 |
| | | End Node | NC | NC | NC | NC |
| 18 | Corridor by sixth wing second floor | FTD | -113,00 | -5,00 | -119,19 | 12 |
| | | End Node | NC | NC | NC | NC |

While a straightforward solution would involve relocating the gateway to a more strategic position, covering a larger number of nodes, it is worth considering situations where such adjustments may not be feasible. In such cases, deploying a relay in a strategic location could be a viable solution. Figure 10.4 illustrates how the relay can be strategically deployed to cover multiple nodes, facilitating a more optimal LOS between the nodes and the relay.

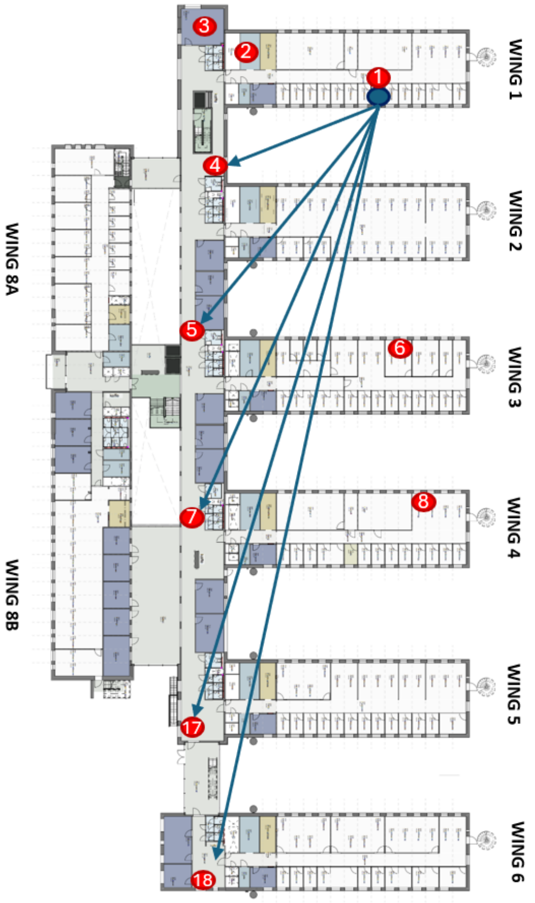


Figure 10.3 Layout drawing of second floor, where arrows indicate the LOS from gateway to node

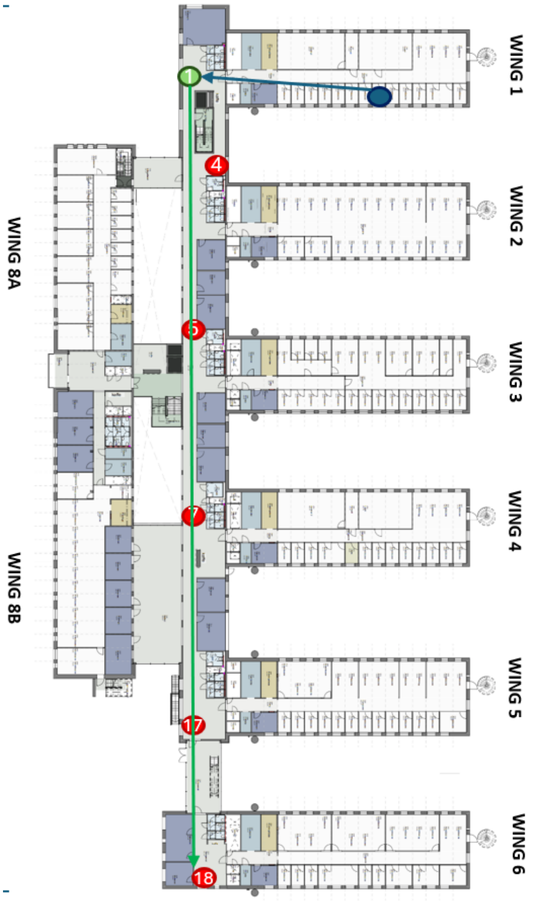


Figure 10.4 Layout drawing of second floor, where arrows indicate the LOS from relay to node

Similarly, upon analysing Table 10.2 in conjunction with the building layout, a comparable pattern emerges in the garage area. As one progresses deeper into the garage, thereby increasing the distance from the gateway, the signal weakens rapidly with each successive wing traversed. At location twelve, situated beneath the second wing, the signal is severely attenuated, becoming completely lost by location thirteen, beneath the third wing. As in the previous scenario, the rapid decline in signal strength is primarily attributed to the presence of concrete structures. Furthermore, the thick concrete floor separating the tow two storeys significantly contributes to this signal degradation, as illustrated in Figure 10.5.

Table 10.2 Average radio transmission parameters for each designated sample location during the of mapping gateway coverage. The table includes only locations relevant for the second relay position.

| Loc. | Location description | Device | RSSI [dBm] | SNR [dB] | ESP [dBm] | SF |
|---|---|---|---|---|---|---|
| 11 | Waste management area in garage, beneath first wing | FTD | -80,75 | 10,44 | -81,13 | 12 |
| | | End Node | -96,00 | 5,50 | -97,52 | 12 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **12** | Garage by second load bearing beam, beneath second wing | FTD | -90,80 | 8,25 | -91,46 | 12 |
| | | End Node | -107,00 | -0,50 | -111,01 | 12 |
| **13** | Garage by fourth load bearing beam, beneath third wing | FTD | -97,00 | 7,25 | -97,83 | 12 |
| | | End Node | NC | NC | NC | NC |
| **14** | Garage by sixth load bearing beam, beneath fourth wing | FTD | -103,75 | 4,00 | -105,27 | 12 |
| | | End Node | NC | NC | NC | NC |
| **15** | Garage by eighth load bearing beam, beneath fifth wing | FTD | -104,25 | 2,94 | -106,14 | 12 |
| | | End Node | NC | NC | NC | NC |
| **16** | Garage by tenth load bearing beam, beneath sixth wing | FTD | -108,75 | 0,06 | -111,76 | 12 |
| | | End Node | NC | NC | NC | NC |

Figure 10.6 depicts how the deployment of a relay could potentially enhance signal quality by providing a more optimal LOS between the nodes and the relay.

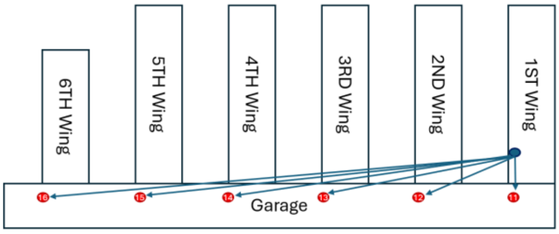


Figure 10.5 Illustration of LOS from gateway to node, building side view

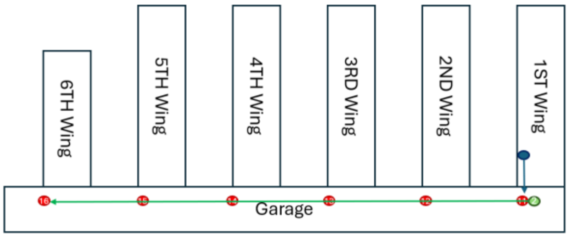


Figure 10.6 Illustration of LOS from relay to node, building side view

# 11 Extending coverage with a relay

The two defined relay test cases were in turn mapped out, taking new samples on the locations where an improvement of signal quality was expected. Chapter 11.1 describes the results from the test with a relay placed in the long corridor of the second floor. Chapter 11.2 describes the test with a relay placed close to the waste management are in the garage.

For the following tests, the end node was reconfigured in ThingPark Wireless, only allowing traffic to be relayed and not picked up by the gateway directly.

## 11.1Relay positioned on the second floor

As depicted in Figure 10.4 from the previous section, the relay was strategically positioned in the corridor just outside the first wing. Placing the relay on top of a chair elevated its position approximately 60 cm from the floor. This arrangement aimed to prevent accidental tripping, as the relay blended into the grey carpet, and the tests were conducted during business hours.

The updated list of samples concerning this relay scenario is presented in Table 14.2 of Appendix D. This table incorporates results from the gateway mapping, retaining the old measurements for comparison and visualization purposes. Notably, the relay is listed as a separate row, while the relayed end node is added as a new column for each measurement.

Table 11.1 highlights the most significant findings from the mapping exercise. A notable enhancement in signal quality can be observed across all the listed locations. As anticipated, improved coverage is evident at locations five, seven, seventeen, and eighteen, benefiting from the clear LOS provided by the relay. Notably, the latter three locations are now within coverage range. While location four remains shadowed behind the staircase, there is a noticeable improvement in signal strength compared to direct communication with the gateway. Furthermore, location nine, adjacent to the reception on the first floor, regains coverage thanks to the relay. This improvement is attributed to fewer obstructive barriers within the LOS between the relay and the end node.

Table 11.1 Average radio transmission parameters for each designated sample location during the relay coverage testing. The table includes only locations relevant when the relay was positioned on the second floor.

| Loc. | Device | RSSI [dBm] | | SNR [dB] | | ESP [dBm] | | SF | |
|---|---|---|---|---|---|---|---|---|---|
| | | Directly | Relayed | Directly | Relayed | Directly | Relayed | Directly | Relayed |
| 4 | End Node | -101,25 | -90,00 | 7,38 | 11,00 | -102,01 | -90,33 | 12 | 12 |
| 5 | End Node | -114,50 | -97,00 | -14,13 | 7,00 | -128,79 | -97,79 | 12 | 12 |
| 7 | End Node | NC | -100,67 | NC | 7,33 | NC | -101,41 | NC | 12 |
| 9 | End Node | NC | -105,00 | NC | 5,00 | NC | -106,19 | NC | 12 |
| 17 | End Node | NC | -104,50 | NC | 6,50 | NC | -105,38 | NC | 12 |
| 18 | End Node | NC | -117,50 | NC | -6,50 | NC | -124,99 | NC | 12 |

## 11.2 Relay positioned on the first floor

As depicted in Figure 11.1, the relay was relocated to the garage beneath the first wing. Positioned atop a stack of wooden pallets, as shown in Figure 11.2, the relay's elevation was approximately 1,6 meters from the floor. This adjustment was made due to the garage's initial 20-meter inclining slope of approximately 1:17, aiming to ensure a clearer LOS into the garage.

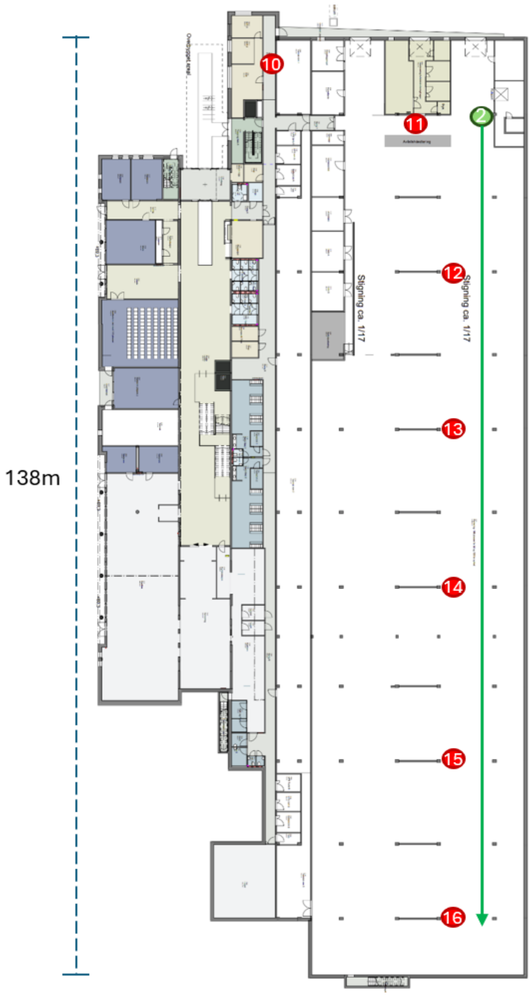


Figure 11.1 Layout drawing of first floor, where arrows indicate the LOS from relay to node

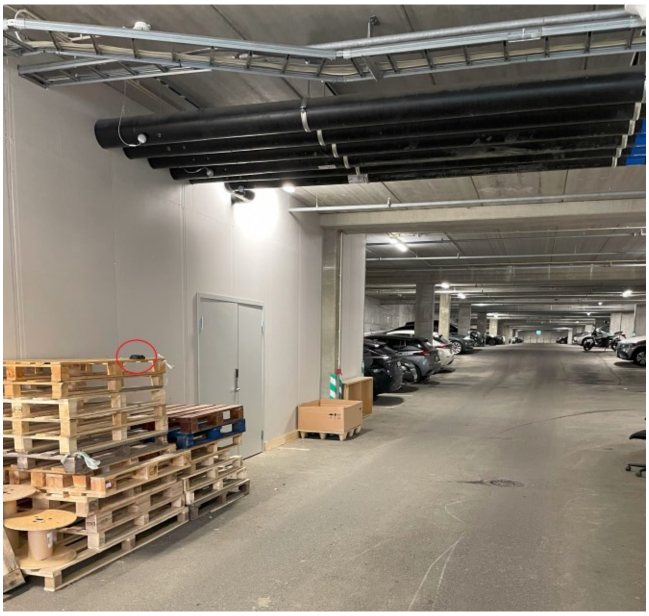


Figure 11.2 Picture of relay mounted in garage. Relay location highlighted by red circle

The updated list of samples corresponding to this relay scenario is detailed in Table 14.3 of Appendix E, maintaining the same format as the previous relay test scenario. Table 11.2 highlights the most notable outcomes from the mapping exercise. As anticipated, there is a substantial improvement in coverage for node locations twelve to sixteen, benefiting from the relay's unobstructed LOS. Particularly noteworthy is that the latter four locations are now within the coverage range.

Table 11.2 Average radio transmission parameters for each designated sample location during the relay coverage testing. The table includes only locations relevant when the relay was positioned on the first floor.

| Loc. | Device | RSSI [dBm] | | SNR [dB] | | ESP [dBm] | | SF | |
|---|---|---|---|---|---|---|---|---|---|
| | | Directly | Relayed | Directly | Relayed | Directly | Relayed | Directly | Relayed |
| **12** | End Node | -107,00 | -91,00 | -0,50 | 9,50 | -111,01 | -91,49 | 12 | 12 |
| **13** | End Node | NC | -90,50 | NC | 10,00 | NC | -90,91 | NC | 12 |
| **14** | End Node | NC | -103,00 | NC | 5,50 | NC | -104,08 | NC | 12 |
| **15** | End Node | NC | -103,50 | NC | 5,50 | NC | -104,58 | NC | 12 |
| **16** | End Node | NC | -112,00 | NC | 2,00 | NC | -114,12 | NC | 12 |

# 12 LoRaWAN relay-enabled end node

To develop a LoRaWAN relay application based on any of the suggested use cases, a relay-enabled end node is required. Since the end node following the relay evaluation kit lacked usable sensor output, a new end node was required. However, due to the novelty of the relay feature and the experimental status of Semtech's LBM stack, no off-the-shelf sensors were found to be compatible.

To address this, options for sensor development using development boards were explored. Initially, both Raspberry Pi and Arduino platforms were considered, leveraging two HAT/Shield radio boards designed by Pi Supply. These boards are tailored to directly interface with the PCB headers of the respective development boards. However, as these radio modules utilize the RAKWireless RAK811 module, based on Semtech's SX1276 LoRa radio, and given that the SX1276 is not supported, both solutions were deemed unsuitable. Consequently, a search commenced for radio boards equipped with either a SX126x or LR11xx RF module. Another Arduino-based option explored was the "MKR WAN 1310". However, it also employs the SX1276 LoRa module, rendering it incompatible for the intended purpose.

The STMicroelectronics Nucleo-WL55JC development board seemed promising, utilizing a STM32WL55JC MCU which incorporates a SX1261 radio transceiver module on the same chip. This board is based on the same technology as the STM32L4, where the L476RG version is already supported in the LBM. As the process of porting the proved time-consuming, a decision was made to focus on an excitingly supported board and RF module combination. That narrowed the choice down to the two supported MCU boards in combination with a LR11xx or SX126x radio board. The choice fell on a SX1261MB2BAS radio board in grouped with the Nucleo-L476RG MCU. On the chosen radio shield the SX1261 module is assembled on a PCB, designed to fit on top of the Arduino headers on top of the Nucleo module. The following chapters will detail how the relay enabled version of the periodical uplink application were modified to suit input from a soil moisture sensor.

## 12.1 Soil moisture sensor

One of the discussed use cases involved moisture sensors installed in flower beds and urns around across Porsgrunn city. To measure the moisture, a sensor input was required for the development board. The Grove Capacitive Moisture Sensor by Seed Technology was selected for this purpose. It operates on either 3.3 or 5V power and produces an analogue voltage output based on the chosen power source. Figure 12.1 illustrates the connection diagram between the sensor and the development board, with 5V power selected for this setup. To read the analogue value on the Nucleo, an analogue input channel connected to an ADC was required. The Nucleo board features three 12-bit ADCs, each with 16 channels. Eq. 12-1 demonstrates the calculation of ADC resolution for a 12-bit ADC with a 5V reference voltage.

$$ADC\ output = \frac{2^N \times Analogue\ Input\ Voltage}{Ref\ Voltage} = \frac{2^{12} \times 5}{5} = 4096 \qquad \text{Eq. 12-1}$$

This indicates that the ADC output will range from 0 to 4096. The sensor's datasheet provides values for a 10-bit ADC. When converted to 12-bit, an ADC output of 0-1200 corresponds to dry soil, 1200-2800 indicates humid soil, and 2800-4096 represents wet soil.

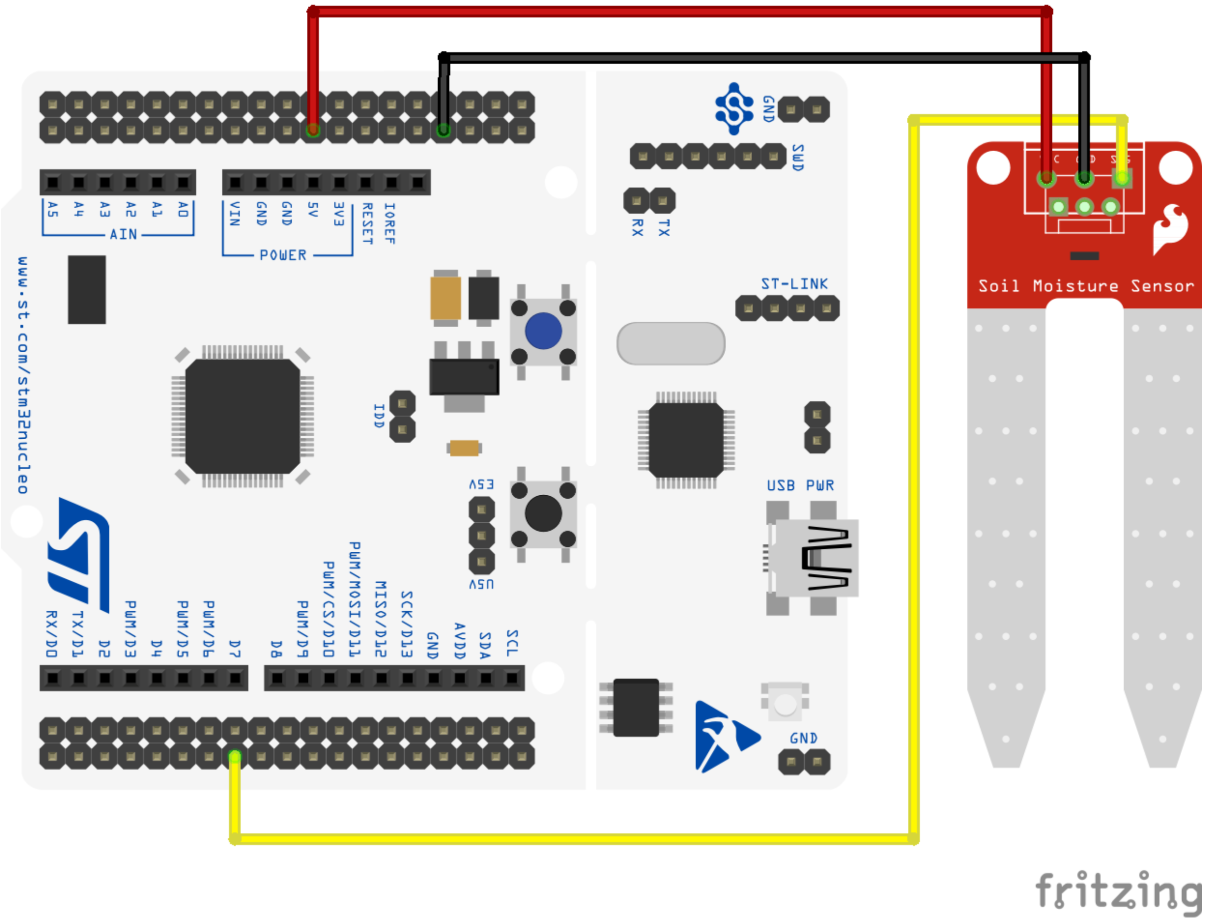


Figure 12.1 Connection diagram of the Nucleo development board with an analogue moisture sensor. The LoRa RF shield is not shown but is connected to the Arduino header.

### 12.1.1Software configuration

In STM32CubeIDE, developed by STMicroelectronics, peripheral selection and configuration are streamlined through a graphical interface. The IDE automatically generates initialization code based on these selections. However, when using the relay-enabled LBM, this process requires manual configuration.

Within the LBM, the "modem_pinout.h" header file defines the pins used by the radio modem. As pin PB_1 was available, it was chosen as the ADC input channel. The ADC peripheral was not utilized in the example application. Enabling it involved modifying the "stm32l4xx_hal_conf.h" file to include the ADC HAL driver in the build by including the commented-out code line defining the ADC driver.

To initialize the ADC peripheral, the MX_ADC1_Init function is invoked at the start of the main application. Within this function, parameters such as resolution (12 bits), clock pre-scaler (none), and continuous conversion mode (disabled) are specified in a struct, which is then passed to the HAL_ADC_Init function. This function further calls HAL_ADC_MspInit to initialize the ADC peripherals. In this process, the ADC clock is enabled using the __HAL_RCC_ADC_CLK_ENABLE function, and GPIO is initialized using the __HAL_RCC_GPIOB_CLK_ENABLE function. Since the ADC clock is already initialized to

80MHz through Semtech code's main HAL_Init function, re-initialization is unnecessary. However, it still needs to be enabled. Similarly, GPIO initialization is performed through HAL_GPIO_Init, defining the pin, setting its mode to analogue ADC control, and configuring pull settings via the GPIO struct. As depicted in the connection diagram (Figure 12.1), PB_1 serves as input to channel 16 of the ADC.

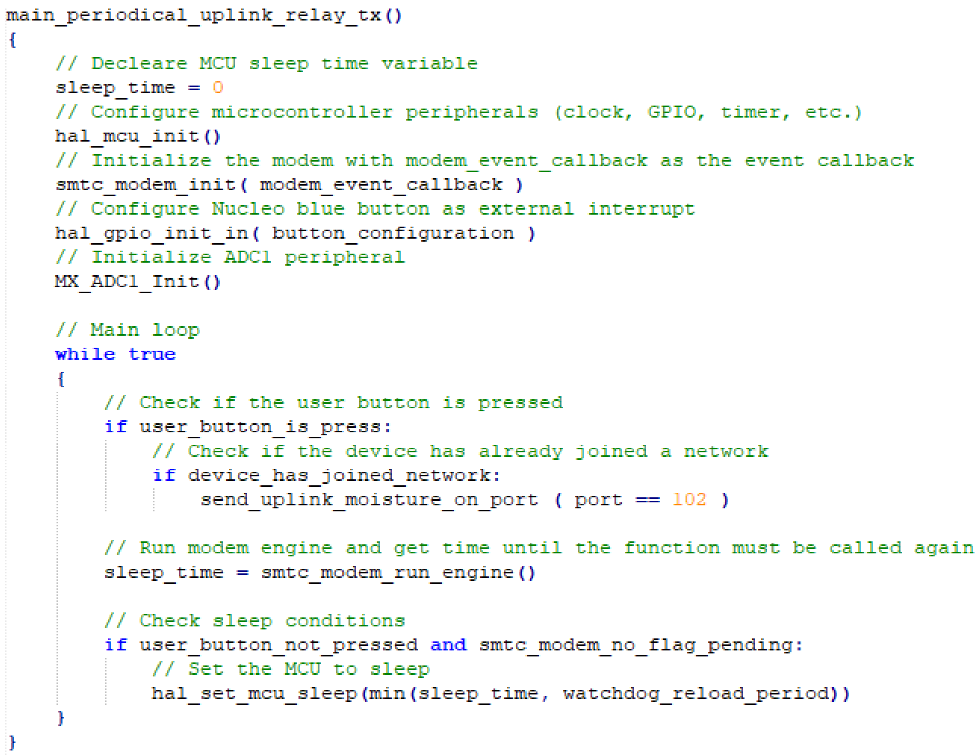
In MX_ADC1_Init, the final configurations include setting up the ADC channel, specifying the sampling time, and determining whether it is a single or differential measurement. Since the datasheet for the moisture sensor did not provide a recommended sampling time, a long sampling time of 247.5 clock cycles was selected. Eq. 12-2 calculates the ADC conversion time, as outlined in the Nucleo reference manual.

$$Tconv = \frac{Sampling\ time + 12.5\ ADC\ Clock\ Cycles}{ADC\ Clock} = \frac{247.5 + 12.5}{80 \times 10^{6}} = 3.25\mu s \qquad \text{Eq. 12-2}$$

## 12.1.2Program description

To describe the main program code, pseudocode will be utilized, presenting a simplified and more human-readable version of the original code, focusing solely on the main functionality. The main program code is based on the periodic uplink example provided by Semtech. The link to the GitHub repository is available in Appendix F.

In Figure 12.2, the pseudocode outlines the main program loop. The function begins by initializing a variable to store the MCU sleep duration. It then proceeds to run initialization functions for the MCU, the modem, external interrupt button, and the ADC.

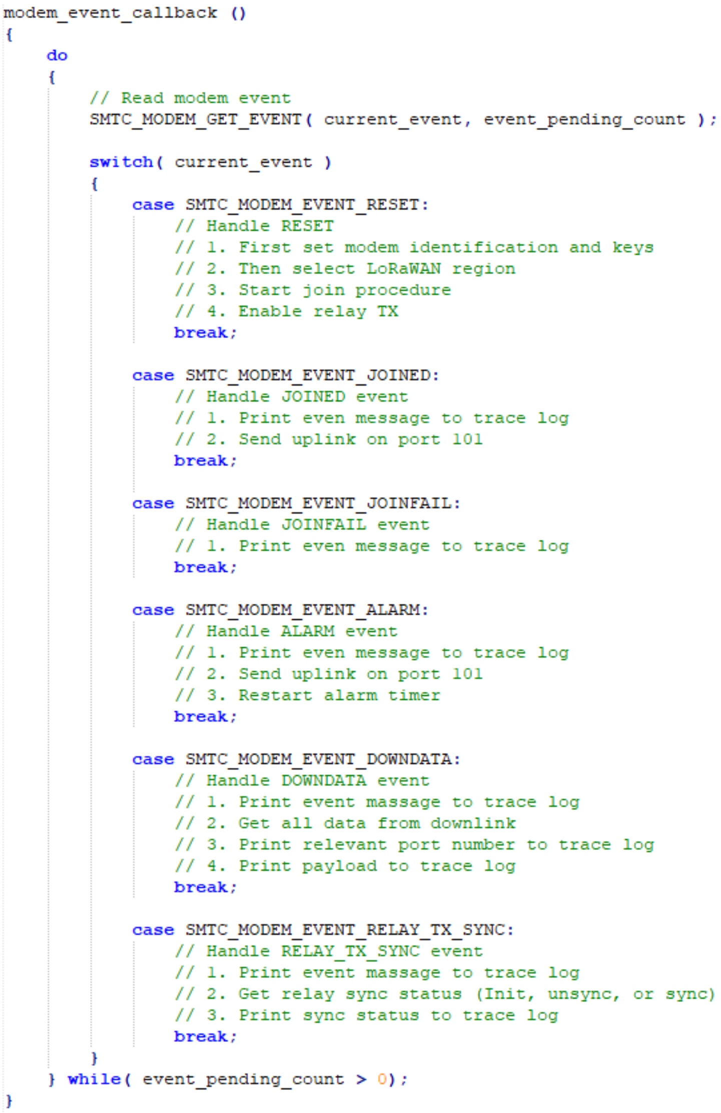


```
main_periodical_uplink_relay_tx()
{
    // Decleare MCU sleep time variable
    sleep_time = 0
    // Configure microcontroller peripherals (clock, GPIO, timer, etc.)
    hal_mcu_init()
    // Initialize the modem with modem_event_callback as the event callback
    smtc_modem_init( modem_event_callback )
    // Configure Nucleo blue button as external interrupt
    hal_gpio_init_in( button_configuration )
    // Initialize ADC1 peripheral
    MX_ADC1_Init()

    // Main loop
    while true
    {
        // Check if the user button is pressed
        if user_button_is_press:
            // Check if the device has already joined a network
            if device_has_joined_network:
                send_uplink_moisture_on_port ( port == 102 )

        // Run modem engine and get time until the function must be called again
        sleep_time = smtc_modem_run_engine()

        // Check sleep conditions
        if user_button_not_pressed and smtc_modem_no_flag_pending:
            // Set the MCU to sleep
            hal_set_mcu_sleep(min(sleep_time, watchdog_reload_period))
    }
}
```

Figure 12.2 Pseudocode describing the main loop of the program

After initialization, the program enters the main while loop. Here, the button status is monitored, and upon detecting a button press, an uplink with the latest moisture value is sent over port 102. If no button press is detected, the modem engine is executed, returning the time until a new run must take place. If neither a button press nor pending modem flags are detected, the MCU is set to sleep for the duration specified by the modem.

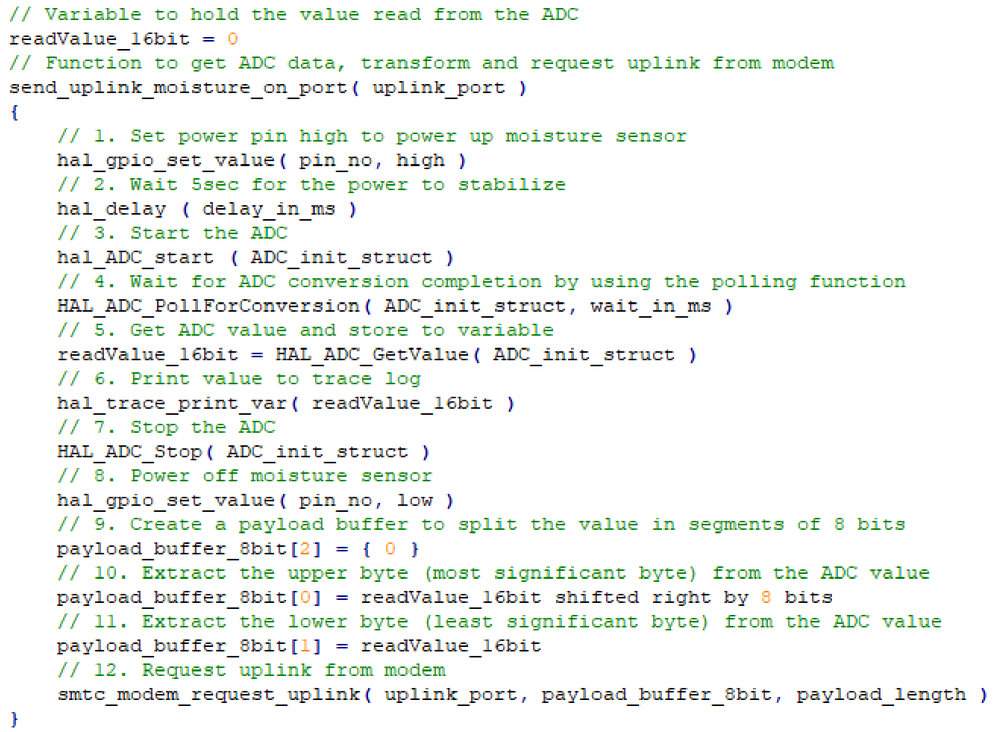
The modem callback structure is defined by the pseudo-like code structure shown in Figure 12.3. This code operates on an event-based mechanism, where a switch statement is used to navigate between different cases based on modem events. Upon a modem reset event, the user region is set, a join procedure is initiated, and the relay TX functionality is enabled. This is followed up by either a join accept or failure event. In the case of acceptance, an uplink is sent on port 101, while a failure results in a timeout and reattempt to join the network. The modem alarm event manages the periodic uplink timing, set to occur every 30 minutes. Upon receiving a downlink, the data is collected and printed to the trace log. Similarly, any changes in the relay TX synchronization status are logged. The modem callback function continues to execute until no further modem events are pending.



```
modem_event_callback ()
{
    do
    {
        // Read modem event
        SMTC_MODEM_GET_EVENT( current_event, event_pending_count );

        switch( current_event )
        {
            case SMTC_MODEM_EVENT_RESET:
                // Handle RESET
                // 1. First set modem identification and keys
                // 2. Then select LoRaWAN region
                // 3. Start join procedure
                // 4. Enable relay TX
                break;

            case SMTC_MODEM_EVENT_JOINED:
                // Handle JOINED event
                // 1. Print even message to trace log
                // 2. Send uplink on port 101
                break;

            case SMTC_MODEM_EVENT_JOINFAIL:
                // Handle JOINFAIL event
                // 1. Print even message to trace log
                break;

            case SMTC_MODEM_EVENT_ALARM:
                // Handle ALARM event
                // 1. Print even message to trace log
                // 2. Send uplink on port 101
                // 3. Restart alarm timer
                break;

            case SMTC_MODEM_EVENT_DOWNDATA:
                // Handle DOWNDATA event
                // 1. Print event massage to trace log
                // 2. Get all data from downlink
                // 3. Print relevant port number to trace log
                // 4. Print payload to trace log
                break;

            case SMTC_MODEM_EVENT_RELAY_TX_SYNC:
                // Handle RELAY_TX_SYNC event
                // 1. Print event massage to trace log
                // 2. Get relay sync status (Init, unsync, or sync)
                // 3. Print sync status to trace log
                break;
        }
    } while( event_pending_count > 0);
}
```

Figure 12.3 Pseudo-like code describing a selection of the events within the modem event callback function

Figure 12.4 displays the pseudocode for the function responsible for obtaining data from the moisture sensor, dividing it into 8-bit segments, and initiating a modem uplink transmission. Initially, a 16-bit variable is allocated to store the ADC reading. The function begins by configuring the GPIO pin used to power the moisture sensor. Following a five second delay to allow the sensor to stabilize after power-up, the ADC peripheral is activated, and conversions commence. The PollForConversion function is utilized to wait for conversion completion, after which the conversion results are retrieved and stored in the 16-bit variable using the GetValue function. The stored value is then printed to the trace log. Subsequently, conversion is halted, the ADC peripheral is disabled, and power to the sensor is cut by switching off the GPIO pin. To accommodate the 16-bit variable in the 8-bit modem payload, a buffer array with two 8-bit elements is employed to temporarily store the most and least significant bytes of the 16-bit variable. Finally, a request is made to the modem to transmit an uplink package at the specified port.



```
// Variable to hold the value read from the ADC
readValue_16bit = 0
// Function to get ADC data, transform and request uplink from modem
send_uplink_moisture_on_port( uplink_port )
{
    // 1. Set power pin high to power up moisture sensor
    hal_gpio_set_value( pin_no, high )
    // 2. Wait 5sec for the power to stabilize
    hal_delay ( delay_in_ms )
    // 3. Start the ADC
    hal_ADC_start ( ADC_init_struct )
    // 4. Wait for ADC conversion completion by using the polling function
    HAL_ADC_PollForConversion( ADC_init_struct, wait_in_ms )
    // 5. Get ADC value and store to variable
    readValue_16bit = HAL_ADC_GetValue( ADC_init_struct )
    // 6. Print value to trace log
    hal_trace_print_var( readValue_16bit )
    // 7. Stop the ADC
    HAL_ADC_Stop( ADC_init_struct )
    // 8. Power off moisture sensor
    hal_gpio_set_value( pin_no, low )
    // 9. Create a payload buffer to split the value in segments of 8 bits
    payload_buffer_8bit[2] = { 0 }
    // 10. Extract the upper byte (most significant byte) from the ADC value
    payload_buffer_8bit[0] = readValue_16bit shifted right by 8 bits
    // 11. Extract the lower byte (least significant byte) from the ADC value
    payload_buffer_8bit[1] = readValue_16bit
    // 12. Request uplink from modem
    smtc_modem_request_uplink( uplink_port, payload_buffer_8bit, payload_length )
}
```

Figure 12.4 Pseudocode describing the function for retrieveing ADC data, splitting it into 8-bit segments, and initiating an uplink request to the modem

### 12.1.3 Demo use case

To verify that the sensor generates uplinks as anticipated, it was placed in a pot with soil and a red pepper plant, as illustrated in Figure 12.5. Upon connecting the USB cable, the board powers up automatically and initiates a join procedure, subsequently transmitting an uplink containing the moisture data.

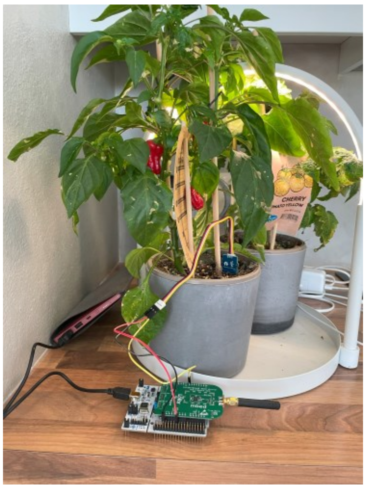
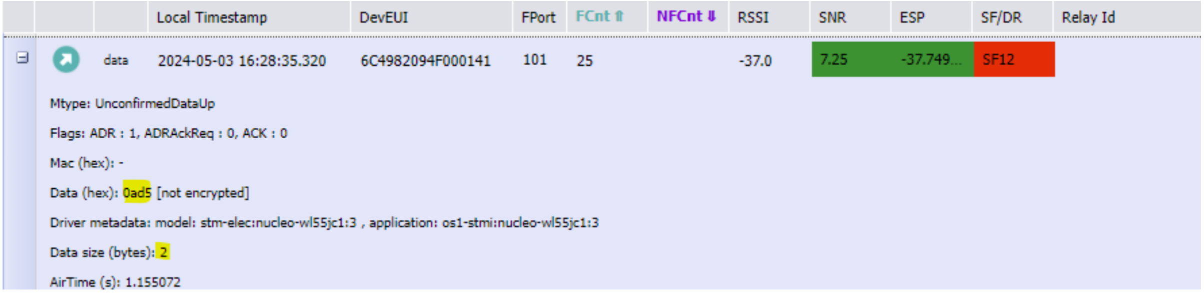


Figure 12.5 Test scenario: The moisture sensor placed in a pot with soil and a red pepper plant, with the Nucleo board powered by a power bank

Figure 12.6 displays the contents of an uplink message as observed in Wireless Logger. In this instance, the node is directly linked to the gateway, and the message is not relayed. The data field highlighted in yellow presents the hexadecimal representation of the moisture level. Upon conversion to a decimal number, this corresponds to 2773. Referring to Eq. 12-1, which states that the ADC output should range between 0 and 4096, a better representation of the soils moisture level is in percentage, as demonstrated in Eq. 12-3.

$$Plant\ moisture = \frac{Measured\ moisture}{Maximum\ moisture} = \frac{2773}{4096} \times 100\% = 67.7\% \qquad \text{Eq. 12-3}$$



| | | | Local Timestamp | DevEUI | FPort | FCnt ⇑ | NFCnt ⇓ | RSSI | SNR | ESP | SF/DR | Relay Id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ⊟ | | data | 2024-05-03 16:28:35.320 | 6C4982094F000141 | 101 | 25 | | -37.0 | 7.25 | -37.749... | SF12 | |

Mtype: UnconfirmedDataUp

Flags: ADR : 1, ADRAckReq : 0, ACK : 0

Mac (hex): -

Data (hex): 0ad5 [not encrypted]

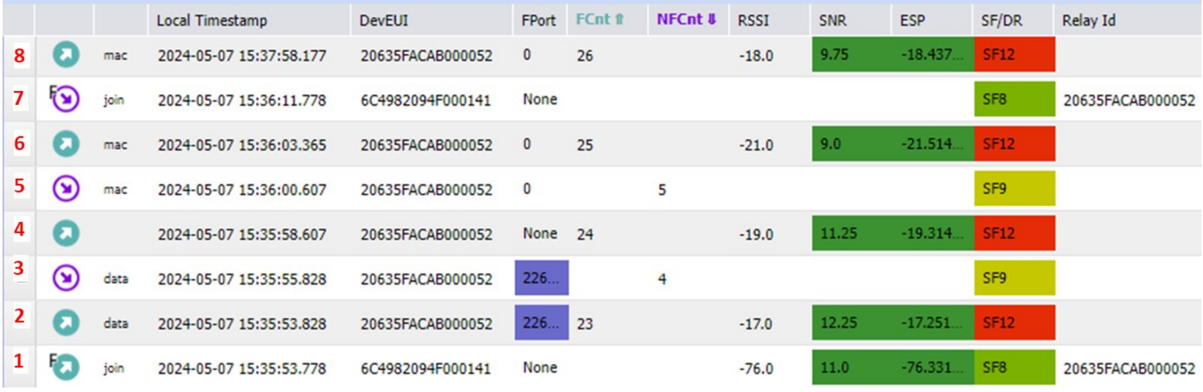
Driver metadata: model: stm-elec:nucleo-wl55jc1:3 , application: os1-stmi:nucleo-wl55jc1:3

Data size (bytes): 2

AirTime (s): 1.155072

Figure 12.6 Content of the uplink message from the end node directly to the gateway, inspected in ThingPark Wireless-Logger

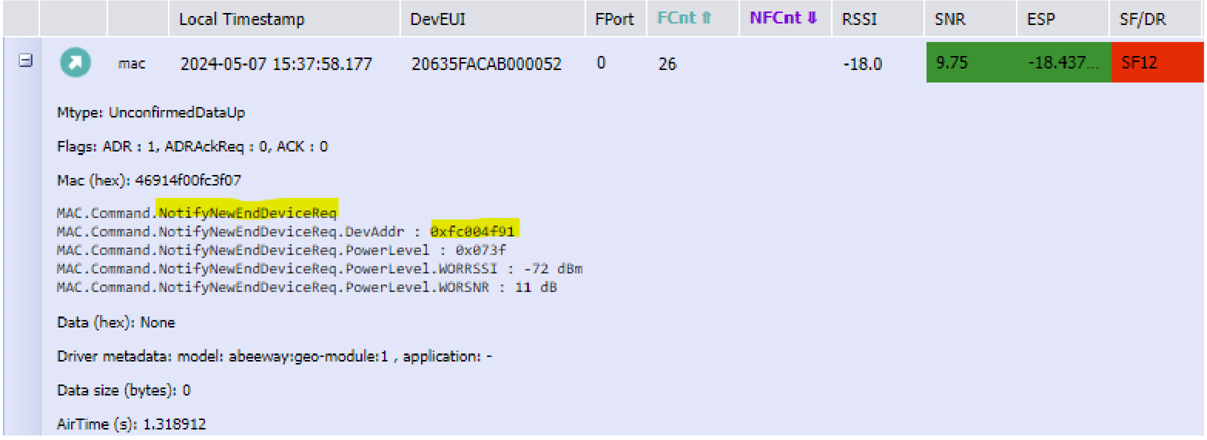
When the end node is forced to connect to the relay using the same procedure as demonstrated with the evaluation kit, everything functions correctly up until the issuance of the join accept and the transmission of the initial uplink. As depicted in Figure 12.7, steps one through seven mirror the example illustrated in chapter 6.1, albeit with the omission of step number four in the previous example. This step involves the relay sending an empty uplink to open a new receive window for use by the LNS.



| | | | Local Timestamp | DevEUI | FPort | FCnt ⇑ | NFCnt ⇓ | RSSI | SNR | ESP | SF/DR | Relay Id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | ↗ | mac | 2024-05-07 15:37:58.177 | 20635FACAB000052 | 0 | 26 | | -18.0 | 9.75 | -18.437... | SF12 | |
| 7 | F↘ | join | 2024-05-07 15:36:11.778 | 6C4982094F000141 | None | | | | | | SF8 | 20635FACAB000052 |
| 6 | ↗ | mac | 2024-05-07 15:36:03.365 | 20635FACAB000052 | 0 | 25 | | -21.0 | 9.0 | -21.514... | SF12 | |
| 5 | ↘ | mac | 2024-05-07 15:36:00.607 | 20635FACAB000052 | 0 | | 5 | | | | SF9 | |
| 4 | ↗ | | 2024-05-07 15:35:58.607 | 20635FACAB000052 | None | 24 | | -19.0 | 11.25 | -19.314... | SF12 | |
| 3 | ↘ | data | 2024-05-07 15:35:55.828 | 20635FACAB000052 | 226... | | 4 | | | | SF9 | |
| 2 | ↗ | data | 2024-05-07 15:35:53.828 | 20635FACAB000052 | 226... | 23 | | -17.0 | 12.25 | -17.251... | SF12 | |
| 1 | F↗ | join | 2024-05-07 15:35:53.778 | 6C4982094F000141 | None | | | -76.0 | 11.0 | -76.331... | SF8 | 20635FACAB000052 |

Figure 12.7 Message sequence showing the LoRaWAN join procedure through the relay, followed by an uplink transmission initiated from the end node

The peculiar behaviour occurs after the end node transmits the initial data uplink, which is not depicted in the figure. Rather than forwarding this message, the relay responds to the uplink by sending a NotifyNewEndDeviceReq MAC command to the LNS, the content of which is displayed in Figure 12.8.



| | | | Local Timestamp | DevEUI | FPort | FCnt ⇑ | NFCnt ⇓ | RSSI | SNR | ESP | SF/DR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ⊟ | ↗ | mac | 2024-05-07 15:37:58.177 | 20635FACAB000052 | 0 | 26 | | -18.0 | 9.75 | -18.437... | SF12 |

Mtype: UnconfirmedDataUp

Flags: ADR : 1, ADRAckReq : 0, ACK : 0

Mac (hex): 46914f00fc3f07

```
MAC.Command.NotifyNewEndDeviceReq
MAC.Command.NotifyNewEndDeviceReq.DevAddr : 0xfc004f91
MAC.Command.NotifyNewEndDeviceReq.PowerLevel : 0x073f
MAC.Command.NotifyNewEndDeviceReq.PowerLevel.WORRSSI : -72 dBm
MAC.Command.NotifyNewEndDeviceReq.PowerLevel.WORSNR : 11 dB
```

Data (hex): None

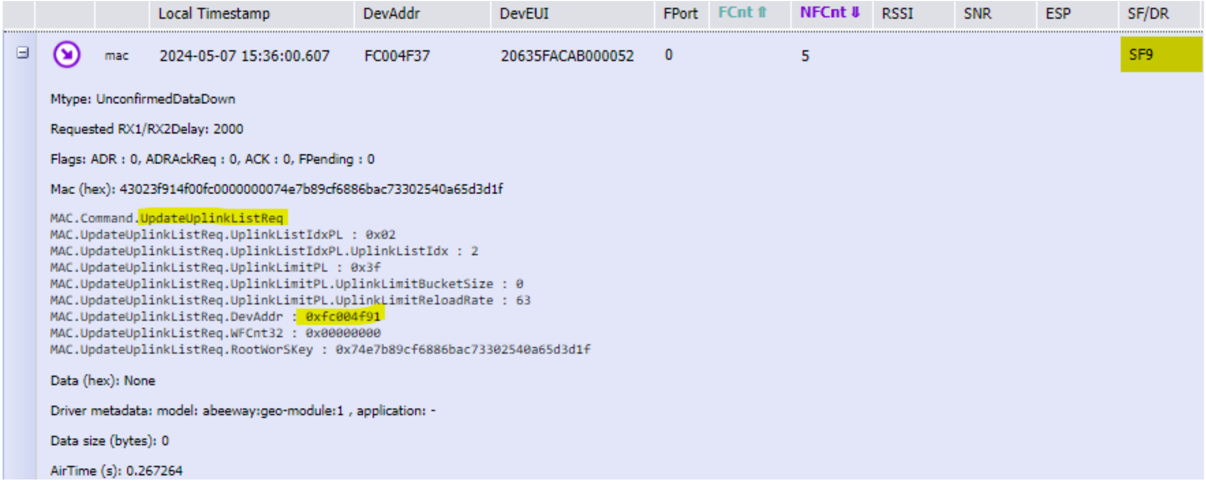
Driver metadata: model: abeeway:geo-module:1 , application: -

Data size (bytes): 0

AirTime (s): 1.318912

Figure 12.8 Content of the NotifyNewEndDeviceReq MAC command sent from the relay to the LNS

The LoRaWAN Relay Specification specifies that the relay should behave in this manner if it is not able to verify the MIC of the WOR Relay Class A uplink message. Despite receiving the UpdateUplinkListReq MAC command from the LNS just two minutes prior, as shown in Figure 12.9, the relay should have possessed all the necessary tools to validate the MIC. The DevAddr matches, and the relay has received the RootWorSKey, both of which are necessary for evaluating the WOR message integrity.



|  |  | Local Timestamp | DevAddr | DevEUI | FPort | FCnt ⇑ | NFCnt ⇓ | RSSI | SNR | ESP | SF/DR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ⊟ | mac | 2024-05-07 15:36:00.607 | FC004F37 | 20635FACAB000052 | 0 |  | 5 |  |  |  | SF9 |

Mtype: UnconfirmedDataDown

Requested RX1/RX2Delay: 2000

Flags: ADR : 0, ADRAckReq : 0, ACK : 0, FPending : 0

Mac (hex): 43023f914f00fc0000000074e7b89cf6886bac73302540a65d3d1f

```
MAC.Command.UpdateUplinkListReq
MAC.UpdateUplinkListReq.UplinkListIdxPL : 0x02
MAC.UpdateUplinkListReq.UplinkListIdxPL.UplinkListIdx : 2
MAC.UpdateUplinkListReq.UplinkLimitPL : 0x3f
MAC.UpdateUplinkListReq.UplinkLimitPL.UplinkLimitBucketSize : 0
MAC.UpdateUplinkListReq.UplinkLimitPL.UplinkLimitReloadRate : 63
MAC.UpdateUplinkListReq.DevAddr : 0xfc004f91
MAC.UpdateUplinkListReq.WFCnt32 : 0x00000000
MAC.UpdateUplinkListReq.RootWorSKey : 0x74e7b89cf6886bac73302540a65d3d1f
```

Data (hex): None

Driver metadata: model: abeeway:geo-module:1 , application: -

Data size (bytes): 0

AirTime (s): 0.267264

Figure 12.9 Content of the UpdateUplinkListReq MAC command sent to relay by the LNS

# 13 Discussion

Several topics have been addressed in this work, ranging from evaluating the performance of an existing relay and end node combination to developing a relay-enabled end node based on the LBM example application. In this chapter, the results will be evaluated, and areas for further work will be suggested.

## 13.1Relay performance in office tests

During the tests conducted in the office building, there were some notable findings which should be discussed. Firstly, the field test device was included in the initial tests of the gateway coverage to compare its radiometric parameters to the ones of the end node. Table 10.1 and Table 10.2 reveal a significant difference in the registered radio parameters between the two devices. This is attributed to the external antenna, as both devices have approximately the same receiver sensitivity. The evaluation kit is based on a re-purpose of an existing design, hereby it is not purposely built for the relay functionality such as with e.g. Deviceroy's Aria relay. As the main function of the relay is to extend the coverage of an existing network, adding an external antenna to the device should be considered.

An aspect not covered in the previous sections is the link budget. In chapter 9, several details related to the materials composition and construction of the test building were described. These details were included to offer a more comprehensive understanding of the test location, shedding light on the barriers that attenuate LoRa signals. While evaluating the relay's performance, the focus was not on the link budget. Instead, the methodology involved mapping coverage with and without the relay to demonstrate the technology's capabilities.

Another observation was related to the stability of the relay. Initially, both uplink and downlink messages were received without issue. However, after an hour of testing, the relay ceased forwarding messages. No alarms were active in the Device Manager and packets were showing up in the Wireless Logger. Several factors could account for this occurrence, with duty cycle limitations being the most probable. Although the relay employs ADR, the network server appears not to adjust the SF of the relay even when it remains within close proximity to the gateway for extended periods. Consequently, the relay unnecessarily prolongs the airtime for each uplink, including both data and MAC transmissions. This exacerbates duty cycle concerns, as per EU regulations limiting it to 1% of the time. As indicated in Eq. 13-1, with a packet transmitted during office testing consuming an airtime of 1.974272 seconds, adherence to duty cycle requirements dictates that a packet can only be sent every 3.29 minutes, averaging approximately 18 packets per hour. While this limit was not considered during testing, it remains unverified whether this was the actual root cause.

$$Wait\ time\ between\ packets = 1.974272s \times 100\% = 197.4272s = 3.29min \qquad \text{Eq. 13-1}$$

Besides the ADR failing to limit the SF, the relay is configured with three packet redundancy, evident from the NbTrans field in Figure 6.10. This setting mandates the relay to transmit three consecutive uplinks for every uplink packet, intended to enhance the quality of service. However, this feature substantially consumes the total duty cycle budget.

The last two factors potentially influencing the uplinks are the token bucket principle and the assigned connectivity plan. Since the token bucket was configured to bypass, it is not the limiting factor. Regarding the assigned connectivity plan, there is an uplink limiting factor known as the rate regulator. This is configured to allow 144 packets per day, with 5 packets allowed in a burst. However, as the uplink regulator policy is set to mark, packets exceeding these limits are marked but not dropped. Consequently, this also cannot be the limiting factor.

One thing to note is that when the relay is reset in bootloader mode, previously connected end nodes do not reconnect automatically without manually triggering a new join request. This can quickly become an issue if deployed in field and the end node is not possible to reach directly from the gateway. Then the only option to get the device back on the network is to manually reset the device in the field.

## 13.2Availability of relay enabled end nodes

Since the version of LBM containing the relay feature remains experimental, few relay compatible end nodes has been identified. This statement is based on several hours spent looking through online catalogues and datasheets. Technical support inquiries were also made to two end node manufacturers, Seed Technology and Adeunis. Both companies responded similarly, stating that their sensors currently do not support the relay feature, with no information available regarding plans for its implementation. Additionally, they confirmed that their LoRaWAN software implementations were not based on the LBM library.

## 13.3Porting LBM to Nucleo WL board

As a part of this project, an initial port for the STM32 Nucleo-WL55JC development board was initiated but not completed. The relevant board HAL and CMSIS libraries were obtained from the STM32 GitHub pages as part of the STM32CubeWL software package. Core file templates were relocated to the core folder within the template structure, and the HAL configuration file was adjusted to commenting out unused drivers. However, further integration work was required to connect these drivers with the Semtech modem library, which was deemed labour-intensive and was not pursued. The GitHub repository link is available in Appendix F for reference.

## 13.4Relayed end node implementation

As detailed in Chapter 12.1.3, an unusual issue arises when the relayed end node attempts to transmit uplinks for the relay to forward. This problem persists even when compiling and uploading the example application without any code modifications apart from entering the device identification and keys in the specified header file. Pinpointing the exact cause is challenging, particularly as it is not feasible to inspect the WOR communication between the devices. One potential explanation is that the RootWorSKey is incorrectly derived, either at the end node or the network server responsible for sending this session key to the relay. If these keys do not match, the Message Integrity Code (MIC) cannot be verified. In LoRaWAN L2 specification version 1.0.4, the AppKey is used to derive the NwkSKey, which in turn is used to derive the RootWorSKey. Since there is a slight difference in how these keys are derived between L2 versions 1.0.4 and 1.1, a test was conducted by switching the specification from

1.0.4 to 1.1 in ThingPark Device Manager. However, this only resulted in a join loop after resetting the end node, exacerbating the situation. Another potential cause could be that the firmware running on the relay is unable to accommodate new end devices. If the UpdateUplinkListReq MAC command is not handled correctly, the list of trusted end nodes might not be properly updated. This issue has been reported to Actility, but no response has been received. Unfortunately, the relay CLI interface does not facilitate this type of fault finding.

## 13.5Soil moisture application

A corrosion challenge emerged when the moisture sensor was left in moist soil for a few days. The selected sensor, being inexpensive, had exposed copper anodes, and continuous 5V power caused it to function like an electrolysis experiment. To address this, the power source was switched from the 5V output to a GPIO pin, which was powered up a few seconds before sampling began and turned off afterward. However, as the GPIO pin on the Nucleo board only provided 3.3V output, this also affected the ADC resolution, which expected a 5V reference.

## 13.6Further work

To address the issue causing the relay to issue the NotifyNewEndDeviceReq MAC command, building a relay node based on the same software as the end node could help eliminate potential misalignments with the relay from the evaluation kit.

As for the soil moisture sensor, several areas for further work exist. Since the data is transmitted in hexadecimal format, a driver in ThingPark or conversion at the application server end would be necessary to convert it back to a decimal number. Additionally, developing an application for visualizing sensor data could be beneficial, although integrating it into existing environments might be more convenient. To achieve the full ADC resolution, solutions involving power through a transistor, operational amplifier or upgrading to a better sensor model could be explored.

# 14 Conclusion

The LoRaWAN relay feature was developed by the LoRa Alliance to cover two primary use cases; network expansion and extending network places that are difficult to reach. The primary focus of their development was to present a simple and cost-efficient method for increasing the reach of an existing network. Through the comprehensive review of the LoRaWAN protocol together with the study of the LoRaWAN Relay Specification performed as part of the work with this thesis, a solid foundation was laid for evaluation the relay feature.

A relay evaluation kit supplied by Altibox, was used as a test case within a large office building, constructed of dense concrete walls forming natural barriers for the LoRa radio. These barriers were exploited to effectively attenuate the radio signals, minimizing the coverage provided by an indoor LoRaWAN gateway. The gateway coverage was first mapped, followed by strategically placing a relay to extend the coverage to areas previously out of reach. For both relay test cases provided to a significant improvement in the signal quality. Not only does this extend coverage to the nodes, but it also enables utilization of ADR, thereby effectively reducing battery consumption.

Even though the presented results displayed impressive improvements, there were findings in need of discussion. As debated in the discussion chapter, the challenges related to the relay's stability must be solved. If it is indeed the duty cycle limits that leads to these registered dropouts, the first action should be to investigate why the ADR does not permit the relay to switch to a higher data rate. Additionally, the number of relayed end nodes and their uplink frequency must be evaluated towards the duty cycle requirements before deciding to deploy a relay to increase the network reach.

Relay use cases were explored and discussed, where one specific use case founded the basis for developing a relay enabled end node. The need for developing a node was evident as no relay enable end nodes were found to be available on the market. Semtech's experimental release of the relay enabled LoRa Basics Modem software was used as the basis for the application. The hardware utilized for the end node was a STM32 Nucleo development board together with a Semtech RF shield and an analogue moisture sensor. As presented, the relay was unable to verify the MIC of the uplinks issued by the end node, leading to the packets being dropped. The possible cause is evaluated in the discussion chapter, where a reply from the relay manufacturer is required to conclude.

As a request and a call to action for all readers of this thesis, especially those with the influence to encourage suppliers and manufacturers to develop relay-enabled end devices, please do so. The relay feature presents exciting potential, particularly for extending LoRaWAN network coverage to challenging areas. To advance this innovative solution, it is essential to raise awareness in the market about its capabilities and the demand for such products.

The LBM software, designed with a focus on portability, streamlines the development process for the LoRaWAN aspect of any end device. Manufacturers investments in further development and testing will improve the software and its functionalities, thereby making this feature more widely accessible.

# References

[1] 'Checking the device settings and activity | ThingPark Wireless documentation', ThingPark. Accessed: Mar. 05, 2024. [Online]. Available: https://docs.thingpark.com/thingpark-wireless/7.3/docs/user-guide-tpw/device-manager/manage-a-device/check-device-settings-activity

[2] 'What is line of sight (LOS) and why is it important?', WhatIs. Accessed: Mar. 13, 2024. [Online]. Available: https://www.techtarget.com/whatis/definition/line-of-sight-LOS

[3] 'Security', The Things Network. Accessed: Mar. 29, 2024. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/security/

[4] 'IoT-teknologi: LoRaWAN: Lang rekkevidde, lavt forbruk, lisensfri, lav pris', Altibox. Accessed: Feb. 11, 2024. [Online]. Available: https://www.altibox.no/bedrift/iot/lorawan/

[5] M. Stroeve, 'New LoRaWAN Relay Feature', Inside Out - Semtech's Corporate Blog. Accessed: Feb. 08, 2024. [Online]. Available: https://blog.semtech.com/the-new-lorawan-relay-feature

[6] 'LoRa and LoRaWAN: Technical overview'. Semtech Corporation, Dec. 2019. Accessed: Jan. 22, 2024. [Online]. Available: https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf

[7] 'Introduction to LoRaWAN®', Wiki by ST. Accessed: Feb. 17, 2024. [Online]. Available: https://wiki.st.com/stm32mcu/wiki/Connectivity:Introduction_to_LoRaWAN

[8] 'About LoRa Alliance®', LoRa Alliance®. Accessed: Mar. 15, 2024. [Online]. Available: https://lora-alliance.org/about-lora-alliance/

[9] T. Network, 'LoRa World Record Broken: 832km/517mi using 25mW', The Things Network. Accessed: Mar. 18, 2024. [Online]. Available: https://www.thethingsnetwork.org/article/lorawan-world-record-broken-twice-in-single-experiment-1

[10] T. Network, 'New LoRa world record: 1336 km / 830 mi', The Things Network. Accessed: Mar. 18, 2024. [Online]. Available: https://www.thethingsnetwork.org/article/new-lora-world-record-1336-km-830-mi

[11] 'What is LoRa', Arridae Infosec - Blog. Accessed: Mar. 18, 2024. [Online]. Available: https://www.arridae.com/blogs/LoRa.php

[12] 'Chirp spread spectrum', *Wikipedia*. Jan. 10, 2024. Accessed: Mar. 18, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Chirp_spread_spectrum&oldid=1194731725

[13] L. Vangelista, 'Frequency Shift Chirp Modulation: The LoRa Modulation', *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1818–1821, Dec. 2017, doi: 10.1109/LSP.2017.2762960.

[14] *LoRa CHIRP*, (Nov. 17, 2017). Accessed: Mar. 19, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=dxYY097QNs0

[15] F. Kuan, ‘What is the Technology Behind LoRa Frequency’, MOKOSmart #1 Smart Device Solution in China. Accessed: Mar. 19, 2024. [Online]. Available: https://www.mokosmart.com/lora-frequency/

[16] ‘Analyzing LoRaWAN® traffic | ThingPark Enterprise documentation’, ThingPark. Accessed: Mar. 02, 2024. [Online]. Available: https://docs.thingpark.com/thingpark-enterprise/7.3/docs/user-guide/network-tools/wireless-logger/lorawan-traffic/

[17] ‘LoRaWAN Regional Parameters’. Sep. 2022. Accessed: Mar. 13, 2024. [Online]. Available: https://resources.lora-alliance.org/technical-specifications/rp002-1-0-4-regional-parameters

[18] ‘Spreading Factors’, The Things Network. Accessed: Mar. 20, 2024. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/spreading-factors/

[19] T. Tulka, ‘LoRa Spreading Factor Explained by Tomas Tulka’, Tomas Tulka’s Blog. Accessed: Mar. 19, 2024. [Online]. Available: https://blog.ttulka.com/lora-spreading-factor-explained

[20] *LoRa/LoRaWAN tutorial 13: Symbol, Spreading Factor and Chip*, (Oct. 04, 2018). Accessed: Mar. 21, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=0FCrN-u-Vpw

[21] avbentem, ‘Airtime calculator for LoRaWAN’. Sep. 05, 2020. Accessed: Mar. 19, 2024. [Online]. Available: https://avbentem.github.io/airtime-calculator/ttn/eu868/4

[22] ‘Product details - SX1261’, Semtech. Accessed: Mar. 21, 2024. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-connect/sx1261

[23] ‘LoRaWAN® L2 1.0.4 Specification’. Oct. 2020. Accessed: Mar. 13, 2024. [Online]. Available: https://resources.lora-alliance.org/technical-specifications/ts001-1-0-4-lorawan-l2-1-0-4-specification

[24] ‘Device Classes’, The Things Network. Accessed: Mar. 28, 2024. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/classes/

[25] ‘LoRaWAN Architecture’, The Things Network. Accessed: Mar. 22, 2024. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/architecture/

[26] *Destination LoRaWAN: LoRaWAN® Security - from the Network to the Device*, (Dec. 02, 2021). Accessed: Mar. 29, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=Z48FKH7RLMQ

[27] ‘LoRaWAN® Is Secure (but Implementation Matters)’, LoRa Alliance®. Accessed: Mar. 29, 2024. [Online]. Available: https://lora-alliance.org/resource_hub/lorawan-is-secure-but-implementation-matters/

[28] ‘End Device Activation’, The Things Network. Accessed: Mar. 30, 2024. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/

[29] ‘Lora-net/LoRaMac-node’. Semtech, Apr. 29, 2024. Accessed: May 03, 2024. [Online]. Available: https://github.com/Lora-net/LoRaMac-node

[30] ‘matthijskooijman/arduino-lmic - This repository is retired’, GitHub. Accessed: May 05, 2024. [Online]. Available: https://github.com/matthijskooijman/arduino-lmic/issues/297

[31] 'matthijskooijman/arduino-lmic -Arduino-LMIC library', GitHub. Accessed: May 05, 2024. [Online]. Available: https://github.com/matthijskooijman/arduino-lmic

[32] 'LoRa Basic Modem Porting Guide', GitHub. Accessed: May 06, 2024. [Online]. Available: https://github.com/Lora-net/SWL2001/blob/feature/relay/lbm_lib/PORTING_GUIDE.md

[33] 'LoRa', *Wikipedia*. Mar. 22, 2024. Accessed: Mar. 31, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=LoRa&oldid=1214945587

[34] 'LTE-M', *Wikipedia*. Dec. 21, 2023. Accessed: Mar. 31, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=LTE-M&oldid=1191136479

[35] 'Narrowband IoT', *Wikipedia*. Aug. 16, 2023. Accessed: Mar. 31, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Narrowband_IoT&oldid=1170617264

[36] 'Sigfox', *Wikipedia*. Mar. 23, 2024. Accessed: Mar. 31, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Sigfox&oldid=1215172795

[37] 'Comparing LPWAN connectivity technologies - Implementing Low-Power Wide-Area Network (LPWAN) Solutions with AWS IoT', Amazon. Accessed: Mar. 31, 2024. [Online]. Available: https://docs.aws.amazon.com/whitepapers/latest/implementing-lpwan-solutions-with-aws/comparing-lpwan-connectivity-technologies.html

[38] P. van Eijk, 'LoRa Basics$^{TM}$ Modem Relay: A Low-Cost Battery Powered Network Extender'. Semtech, Aug. 03, 2023. Accessed: Mar. 13, 2024. [Online]. Available: https://info.semtech.com/lorawan-relay-white-paper

[39] 'LoRaWAN Relay Specification'. Sep. 2022. Accessed: Mar. 13, 2024. [Online]. Available: https://resources.lora-alliance.org/technical-specifications/ts011-1-0-0-relay

[40] 'Using LoRaWAN® relays | ThingPark Enterprise documentation', ThingPark. Accessed: Jan. 10, 2024. [Online]. Available: https://docs.thingpark.com/thingpark-enterprise/7.3/docs/user-guide/lorawan-relays

[41] 'LoRa Basic Modem - Relay branch'. Semtech, Dec. 21, 2023. Accessed: Mar. 14, 2024. [Online]. Available: https://github.com/Lora-net/SWL2001/tree/feature/relay/lbm_lib

[42] 'LoRa Basic Modem - Readme file', GitHub. Accessed: Apr. 09, 2024. [Online]. Available: https://github.com/Lora-net/SWL2001/blob/feature/relay/lbm_lib/README.md

[43] *LoRaWAN Relay for Extended Gateway Coverage*, (Dec. 20, 2023). Accessed: Apr. 29, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=2bEdnFqW5CI

[44] F. Lavrut, 'Leading IoT Industry Players Bring Implementation of Relay Utilizing LoRaWAN® to Market', Actility. Accessed: Apr. 23, 2024. [Online]. Available: https://www.actility.com/fr/lorawan-relay-pr/

[45] 'Skjærgårdstjenesten Porsgrunn Driftsrapport 2023', Skjærgårdstjenesten, Porsgrunn, Driftsrapport, Dec. 2023. Accessed: Apr. 30, 2024. [Online]. Available: https://www.porsgrunn.kommune.no/media/sfnlppqw/2023-driftsrapport.pdf

[46] 'LoRa connectivity for Condition-Based and Predictive Maintenance in Smart Industry'. ST Microelectronics, Nov. 2021. Accessed: May 01, 2024. [Online].

Available: https://www.st.com/resource/en/brochure/lora-connectivity-for-condition-based-and-predictive-maintenance.pdf

[47] ‘UfiSpace Enteprise Pico Cell - ThingPark Ready V1.5 Indoor LoRaWAN Gateway’, ThingPark Market. Accessed: Feb. 10, 2024. [Online]. Available: https://market.thingpark.com/ufispace-enteprise-pico-cell.html

[48] ‘strongSwan - About’, strongSwan. Accessed: May 13, 2024. [Online]. Available: https://www.strongswan.org/about.html

[49] ‘FTD: network tester’, Adeunis. Accessed: Feb. 10, 2024. [Online]. Available: https://www.adeunis.com/en/produit/ftd-network-tester/

[50] S. Boudaud and O. Hersent, ‘LoRaWAN Relay Application note. Revision 1.1-0.’ Abeeway, Jan. 23, 2024.

[51] ‘LPWA Products - Type 1WL(LBEU5ZZ1WL)’, Murata Manufacturing Co., Ltd. Accessed: Mar. 13, 2024. [Online]. Available: https://www.murata.com/en-eu/products/connectivitymodule/lpwa/overview/lineup/type-1wl

# Appendices

Appendix A – Project Description

Appendix B – Project plan

Appendix C – Averaged values from mapping of gateway coverage

Appendix D – Averaged samples from mapping with relay located at second floor

Appendix E – Averaged samples from mapping with relay located at first floor

Appendix F – Github repository

## Appendix A – Project Description

# FMH606 Master's Thesis

**Title**: Using Relays for Utilizing and Extending LoRaWAN Networks with the use of Altibox LoRaWAN Infrastructure for Internet of Things

**USN supervisor**: Hans-Petter Halvorsen

**External partner**: Altibox, Porsgrunn kommune, Odfjell Drilling

**Task background**:
Since the autumn of 2018, the Altibox and Altibox partnership has expanded the LoRaWAN Sensor Network in Norway and currently has coverage for more than 1,000,000 households in 100 municipalities. The LoRaWAN Sensor Network is a natural extension of the fiber network with major synergies of established infrastructure and the development continues.

Altibox and Partners now offer IoT Access as a commercial service, so that more people can use the Sensor Network for their own sensors. https://www.altibox.no/iot/

LoRa Alliance has announced the new Relay feature that extends LoRaWAN coverage for metering, utilities, Smart Cities, and Industrial Applications. LoRaWAN networks offer great range and penetration, but there may be cases where devices fall outside of the network's coverage or optimal signal strength. For these instances, relays utilizing LoRaWAN can be used to extend the coverage of the network.

References:

- https://lora-alliance.org/lora-alliance-press-release/lora-alliance-announces-new-relay-feature-that-extends-lorawan-coverage-for-metering-utilities-smart-cities-and-industrial-applications/
- https://blog.semtech.com/the-new-lorawan-relay-feature

**Task description**:

Examples of activities that should be performed:

- Get an overview of LoRaWan and the Altibox LoRaWan Infrastructure in general and in context of this work.
- Explore the new Relay feature that LoRa Alliance has announced that extends LoRaWAN coverage for metering, utilities, Smart Cities, and Industrial Applications.
- Explore the “Relay Demo Kit” that will be provided by Altibox. The “Relay Demo Kit” consists of different sensors and equipment that can be used to explore the Relay feature.
- Explore and find proper User cases for LoRaWan and the new Relay feature within these areas. Examples: Explore use of the Relay features as a use case within Porsgrunn kommune. Does Altibox has other relevant suggestions for use cases? Explore use cases within Odfjell Drilling (the company where the student is working).

- Plan, Design, Implement and Test one or more selected User Case(s) for utilizing and extending LoRaWAN networks in the Altibox LoRaWan infrastructure using the new Relay feature. Use cases within Porsgrunn kommune or use cases within Odfjell Drilling.
- Collaboration with other ongoing LoRaWAN project.
- Microsoft Teams and GitHub should be used during project planning and development.
- The system should be properly documented in form of a technical report.

More project details and activities will be discussed when the project starts. You will have great opportunities to influence the content of the project in collaboration with Altibox.

**Student category**: IIA

**Is the task suitable for online students (not present at the campus)?** Yes

**Practical arrangements**: None

**Supervision:**

As a general rule, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

**Signatures**:

Supervisor (date and signature):

Hans-Petter Halvorsen

2024.02.02

Student (date and signature):

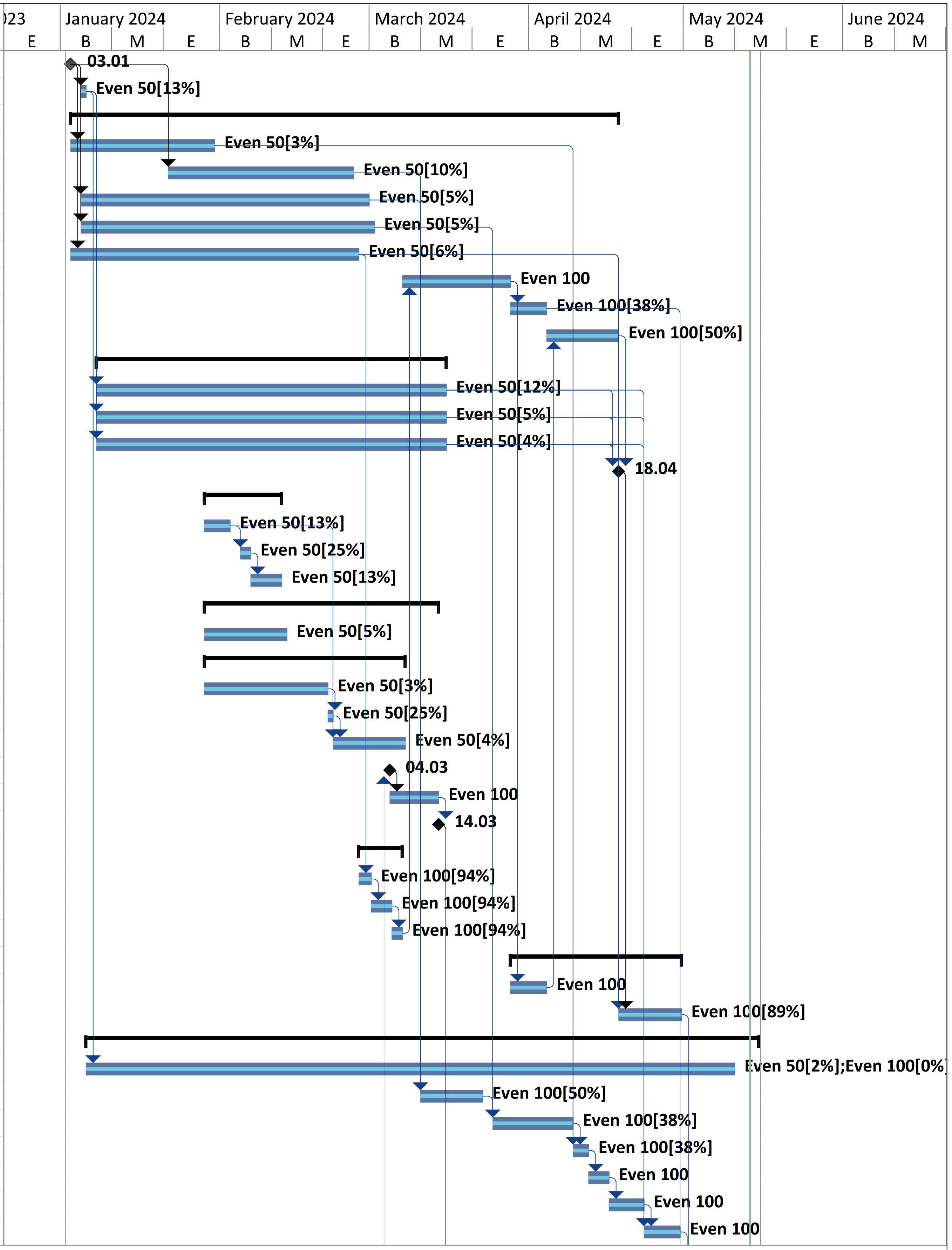
Even Hope

2024.02.02

# Appendix B – Project plan

The project milestone plan follows on the next two pages. This was used to provide an overview of the planned work, while the planner tool in Teams were used for day-to-day planning accompanied by a weekly status report in the Teams channel.

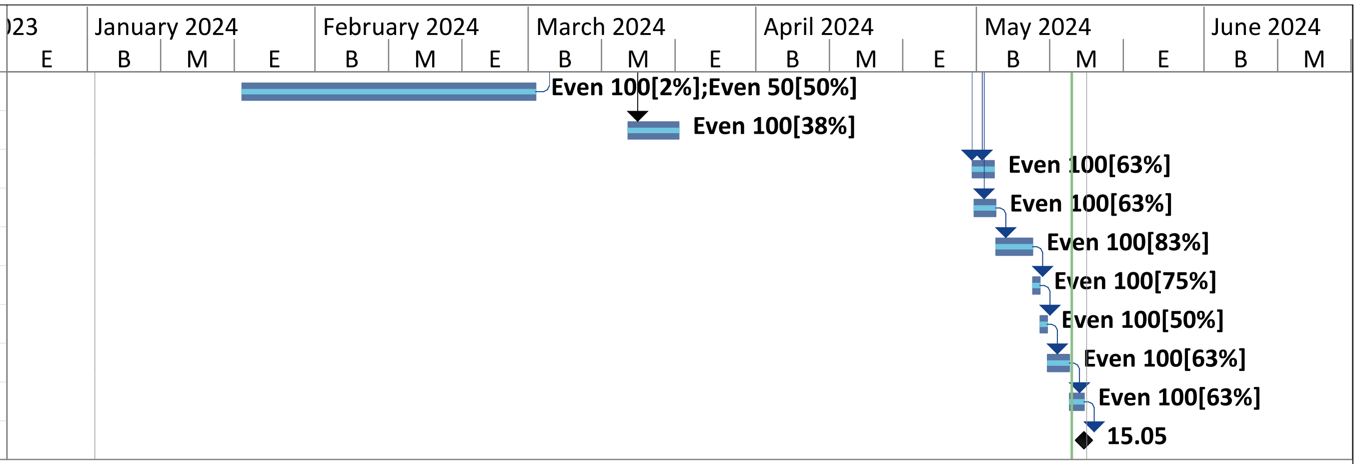
| ID | Task Name | Duration | Work | Start | Finish | Predecessors |
|---|---|---|---|---|---|---|
| 1 | **MS - Start Project** | 0 days | 0 hrs | Wed 03.01.24 | Wed 03.01.24 | |
| 2 | Project start-up meeting | 1 day | 1 hr | Fri 05.01.24 | Fri 05.01.24 | 1 |
| 3 | **Technical research** | **76,4 days** | **170 hrs** | **Wed 03.01.24** | **Thu 18.04.24** | |
| 4 | Altibox infastructure | 20 days | 5 hrs | Wed 03.01.24 | Tue 30.01.24 | 1 |
| 5 | Semtech - Academy for LoRaWAN course | 26 days | 20 hrs | Mon 22.01.24 | Mon 26.02.24 | 1 |
| 6 | LoRa and LoRaWAN | 40 days | 15 hrs | Fri 05.01.24 | Thu 29.02.24 | 1 |
| 7 | LoRaWAN Relays | 41 days | 15 hrs | Fri 05.01.24 | Fri 01.03.24 | 1 |
| 8 | Development boards compatible with LoRaWAN Relay | 40 days | 20 hrs | Wed 03.01.24 | Tue 27.02.24 | 1 |
| 9 | Porting LBM to WL55JC | 15 days | 40 hrs | Thu 07.03.24 | Thu 28.03.24 | 33 |
| 10 | Semtech LBM software library | 5 days | 15 hrs | Thu 28.03.24 | Thu 04.04.24 | 9 |
| 11 | STM32 development | 10 days | 40 hrs | Thu 04.04.24 | Thu 18.04.24 | 35 |
| 12 | **Explore LoRaWAN use cases** | **50 days** | **35 hrs** | **Mon 08.01.24** | **Fri 15.03.24** | |
| 13 | Explore Use Cases for Odfjell | 50 days | 15 hrs | Mon 08.01.24 | Fri 15.03.24 | 2 |
| 14 | Explore use cases for Altibox | 50 days | 10 hrs | Mon 08.01.24 | Fri 15.03.24 | 2 |
| 15 | Explore use cases for Porsgrunn Kommune | 50 days | 10 hrs | Mon 08.01.24 | Fri 15.03.24 | 2 |
| 16 | **MS - Decide which Use Case(s) to implement** | 0 days | 0 hrs | Thu 18.04.24 | Thu 18.04.24 | 13;14;15;11 |
| 17 | **Component initialization** | **11 days** | **13 hrs** | **Mon 29.01.24** | **Mon 12.02.24** | |
| 18 | Explore LoRaWAN Gateway and Field test device | 5 days | 5 hrs | Mon 29.01.24 | Fri 02.02.24 | |
| 19 | Explore onboarding of sensors into ThingPark | 2 days | 4 hrs | Mon 05.02.24 | Tue 06.02.24 | 18 |
| 20 | Explore LoRaWAN Relay demo kit | 4 days | 4 hrs | Wed 07.02.24 | Mon 12.02.24 | 19 |
| 21 | **Test of relay feature at KF35** | **33,5 days** | **43 hrs** | **Mon 29.01.24** | **Thu 14.03.24** | |
| 22 | Coordinate with landlord | 12 days | 5 hrs | Mon 29.01.24 | Tue 13.02.24 | |
| 23 | **Coordinate with IT department** | **29 days** | **9 hrs** | **Mon 29.01.24** | **Thu 07.03.24** | |
| 24 | Get new VLAN configured for test purpose | 18 days | 4 hrs | Mon 29.01.24 | Wed 21.02.24 | |
| 25 | Get switch ports assigned in three locations | 1 day | 2 hrs | Thu 22.02.24 | Thu 22.02.24 | 24 |
| 26 | Connect gateway and verify connection | 10 days | 3 hrs | Fri 23.02.24 | Thu 07.03.24 | 18;25 |
| 27 | Map out and document GW coverage with field tester | 1 day | 9 hrs | Mon 04.03.24 | Mon 04.03.24 | 45 |
| 28 | Deploy relay at edge points and map out range for end device | 7,5 days | 20 hrs | Tue 05.03.24 | Thu 14.03.24 | 27 |
| 29 | **MS - Done testing at KF35** | 0 days | 0 hrs | Thu 14.03.24 | Thu 14.03.24 | 28 |
| 30 | **Development board experimentation** | **6,4 days** | **48 hrs** | **Wed 28.02.24** | **Thu 07.03.24** | |
| 31 | RPI | 2,13 days | 16 hrs | Wed 28.02.24 | Fri 01.03.24 | 8 |
| 32 | Arduino | 2,13 days | 16 hrs | Fri 01.03.24 | Tue 05.03.24 | 31 |
| 33 | ST Nucleo WL55JC | 2,13 days | 16 hrs | Tue 05.03.24 | Thu 07.03.24 | 32 |
| 34 | **Development** | **23,4 days** | **100 hrs** | **Thu 28.03.24** | **Tue 30.04.24** | |
| 35 | Porting Nucleo-WL55JC | 5 days | 40 hrs | Thu 28.03.24 | Thu 04.04.24 | 9 |
| 36 | Design, implement & test moisture appliaction on Nucleo-L476 | 8,4 days | 60 hrs | Thu 18.04.24 | Tue 30.04.24 | 8;16 |
| 37 | **Report** | **92,8 days** | **253 hrs** | **Fri 05.01.24** | **Wed 15.05.24** | |
| 38 | Introduction | 90 days | 15 hrs | Fri 05.01.24 | Fri 10.05.24 | 2 |
| 39 | LoRaWAN | 10 days | 40 hrs | Mon 11.03.24 | Fri 22.03.24 | 5;6 |
| 40 | LoRaWAN relay | 11,67 days | 35 hrs | Mon 25.03.24 | Tue 09.04.24 | 39;7 |
| 41 | Altibox infastructure | 3 days | 10 hrs | Tue 09.04.24 | Fri 12.04.24 | 4;40 |
| 42 | ThingPark | 2 days | 8 hrs | Fri 12.04.24 | Tue 16.04.24 | 41 |
| 43 | Study of relay message sequences | 4,5 days | 20 hrs | Tue 16.04.24 | Tue 23.04.24 | 42 |
| 44 | Relay use cases | 5 days | 30 hrs | Tue 23.04.24 | Tue 30.04.24 | 43;13;14;15 |



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Task | ▬ | Project Summary | ▬ | Inactive Milestone | ◇ | Manual Summary Rollup | ▬ | Deadline | ⇩ |
| Split | ┈ | External Tasks | ▬ | Inactive Summary | ▬ | Manual Summary | ▬ | Progress | ▬ |
| Milestone | ◆ | External Milestone | ◇ | Manual Task | ▬ | Start-only | [ | Manual Progress | ▬ |
| Summary | ▬ | Inactive Task | ▭ | Duration-only | ▬ | Finish-only | ] | | |

Project: msproj11
Date: Tue 14.05.24

| ID | Task Name | Duration | Work | Start | Finish | Predecessors |
|---|---|---|---|---|---|---|
| 45 | Relay test setup and hardware | 30 days | 5 hrs | Mon 22.01.24 | Fri 01.03.24 | |
| 46 | Relay test results | 5 days | 15 hrs | Thu 14.03.24 | Thu 21.03.24 | 29 |
| 47 | LoRaWAN libraries | 3 days | 15 hrs | Tue 30.04.24 | Fri 03.05.24 | 44;10 |
| 48 | Relay-enabled end node | 3 days | 15 hrs | Tue 30.04.24 | Fri 03.05.24 | 36 |
| 49 | Discussion | 3 days | 20 hrs | Fri 03.05.24 | Wed 08.05.24 | 48 |
| 50 | Conclusion | 1 day | 6 hrs | Wed 08.05.24 | Thu 09.05.24 | 49 |
| 51 | Summary | 1 day | 4 hrs | Thu 09.05.24 | Fri 10.05.24 | 50 |
| 52 | Preface | 1 day | 5 hrs | Fri 10.05.24 | Mon 13.05.24 | 51 |
| 53 | Last touch up | 2 days | 10 hrs | Mon 13.05.24 | Wed 15.05.24 | 52 |
| 54 | **MS - Deliver project report** | 0 days | 0 hrs | Wed 15.05.24 | Wed 15.05.24 | 53 |



Project: msproj11
Date: Tue 14.05.24

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Task | ▬ | Project Summary | ┌─┐ | Inactive Milestone | ◇ | Manual Summary Rollup | ▬ | Deadline | ⇩ |
| Split | ┈ | External Tasks | ▬ | Inactive Summary | ┌─┐ | Manual Summary | ┌─┐ | Progress | ▬ |
| Milestone | ◆ | External Milestone | ◆ | Manual Task | ▬ | Start-only | [ | Manual Progress | ▬ |
| Summary | ┌─┐ | Inactive Task | ▭ | Duration-only | ▬ | Finish-only | ] | | |

# Appendix C – Average radio transmission parameters for each location during gateway coverage mapping

Table 14.1 Average radio transmission parameters for each location during gateway coverage mapping

| Loc. | Location description | Device | RSSI [dBm] | SNR [dB] | ESP [dBm] | SF |
|---|---|---|---|---|---|---|
| **1** | Same room as gateway in first wing second floor | FTD | -27,25 | 7,88 | -27,93 | 12 |
| | | End Node | -39,80 | 9,75 | -40,25 | 12 |
| **2** | Wardrobe in the first wing second floor | FTD | -70,50 | 8,81 | -71,05 | 12 |
| | | End Node | -40,75 | 11,50 | -41,06 | 12 |
| **3** | Meeting room just outside first wing second floor | FTD | -101,00 | 5,06 | -102,21 | 12 |
| | | End Node | -70,40 | 10,80 | -70,78 | 12 |
| **4** | Corridor by second wing second floor | FTD | -97,50 | 7,31 | -98,24 | 12 |
| | | End Node | -101,25 | 7,38 | -102,01 | 12 |
| **5** | Corridor by third wing second floor | FTD | -113,75 | -13,25 | -127,25 | 12 |
| | | End Node | -114,50 | -14,13 | -128,79 | 12 |
| **6** | Office space in third wing second floor | FTD | -112,50 | -3,94 | -117,99 | 12 |
| | | End Node | -113,67 | -7,25 | -121,68 | 12 |
| **7** | Corridor by fourth wing second floor | FTD | -113,50 | -13,81 | -127,58 | 12 |
| | | End Node | NC | NC | NC | NC |
| **8** | Office space in fourth wing second floor | FTD | -113,50 | -10,25 | -124,33 | 12 |
| | | End Node | -113,50 | -10,25 | -124,14 | 12 |
| **9** | By reception first floor | FTD | -113,25 | -17,25 | -130,58 | 12 |
| | | End Node | NC | NC | NC | NC |
| **10** | By admin offices first floor | FTD | -94,75 | 7,13 | -95,53 | 12 |
| | | End Node | -87,50 | 7,75 | -88,23 | 12 |
| **11** | Waste management area in garage, beneath first wing | FTD | -80,75 | 10,44 | -81,13 | 12 |
| | | End Node | -96,00 | 5,50 | -97,52 | 12 |
| **12** | Garage by second load bearing beam, beneath second wing | FTD | -90,80 | 8,25 | -91,46 | 12 |
| | | End Node | -107,00 | -0,50 | -111,01 | 12 |
| **13** | Garage by fourth load bearing beam, beneath third wing | FTD | -97,00 | 7,25 | -97,83 | 12 |
| | | End Node | NC | NC | NC | NC |
| **14** | Garage by sixth load bearing beam, beneath fourth wing | FTD | -103,75 | 4,00 | -105,27 | 12 |
| | | End Node | NC | NC | NC | NC |

| | | | | | | |
|---|---|---|---|---|---|---|
| **15** | Garage by eighth load bearing beam, beneath fifth wing | FTD | -104,25 | 2,94 | -106,14 | 12 |
| | | End Node | NC | NC | NC | NC |
| **16** | Garage by tenth load bearing beam, beneath sixth wing | FTD | -108,75 | 0,06 | -111,76 | 12 |
| | | End Node | NC | NC | NC | NC |
| **17** | Corridor by fifth wing second floor | FTD | -111,00 | -12,25 | -123,50 | 12 |
| | | End Node | NC | NC | NC | NC |
| **18** | Corridor by sixth wing second floor | FTD | -113,00 | -5,00 | -119,19 | 12 |
| | | End Node | NC | NC | NC | NC |

# Appendix D – Average radio transmission parameters for each location during second-floor relay coverage testing

Table 14.2 Average radio transmission parameters for each location during second-floor relay coverage testing

| Loc. | Device | RSSI [dBm] | | SNR [dB] | | ESP [dBm] | | SF | |
|---|---|---|---|---|---|---|---|---|---|
| | | No relay | Relayed | No relay | Relayed | No relay | Relayed | No relay | Relayed |
| **4** | FTD | -97,50 | | 7,31 | | -98,24 | | 12 | |
| | End Node | -101,25 | -90,00 | 7,38 | 11,00 | -102,01 | -90,33 | 12 | 12 |
| | Relay | -77,00 | | 12,50 | | -77,25 | | 12 | |
| **5** | FTD | -113,75 | | -13,25 | | -127,25 | | 12 | |
| | End Node | -114,50 | -97,00 | -14,13 | 7,00 | -128,79 | -97,79 | 12 | 12 |
| | Relay | -75,50 | | 12,00 | | -75,77 | | 12 | |
| **7** | FTD | -113,50 | | -13,81 | | -127,58 | | 12 | |
| | End Node | NC | -100,67 | NC | 7,33 | NC | -101,41 | NC | 12 |
| | Relay | -78,33 | | 11,75 | | -78,63 | | 12 | |
| **9** | FTD | -113,25 | | -17,25 | | -130,58 | | 12 | |
| | End Node | NC | -105,00 | NC | 5,00 | NC | -106,19 | NC | 12 |
| | Relay | -71,00 | | 10,00 | | -71,50 | | 12 | |
| **10** | FTD | -94,75 | | 7,13 | | -95,53 | | 12 | |
| | End Node | -87,50 | -91,50 | 7,75 | 9,50 | -88,23 | -91,96 | 12 | 12 |
| | Relay | -69,50 | | 10,75 | | -69,89 | | 12 | |
| **17** | FTD | -111,00 | | -12,25 | | -123,50 | | 12 | |
| | End Node | NC | -104,50 | NC | 6,50 | NC | -105,38 | NC | 12 |
| | Relay | -89,50 | | 8,38 | | -90,12 | | 12 | |
| **18** | FTD | -113,00 | | -5,00 | | -119,19 | | 12 | |
| | End Node | NC | -117,50 | NC | -6,50 | NC | -124,99 | NC | 12 |
| | Relay | -91,33 | | 10,17 | | -91,74 | | 12 | |

# Appendix E – Average radio transmission parameters for each location during first-floor relay coverage testing

Table 14.3 Average radio transmission parameters for each location during first-floor relay coverage testing

| Loc. | Device | RSSI [dBm] | | SNR [dB] | | ESP [dBm] | | SF | |
|---|---|---|---|---|---|---|---|---|---|
| | | No relay | Relayed | No relay | Relayed | No relay | Relayed | No relay | Relayed |
| **10** | FTD | -94,75 | | 7,13 | | -95,53 | | 12 | |
| | End Node | -87,50 | -118,50 | 7,75 | -9,50 | -88,23 | -128,59 | 12 | 12 |
| | Relay | -67,50 | | 11,50 | | -67,82 | | 12 | |
| **11** | FTD | -80,75 | | 10,44 | | -81,13 | | 12 | |
| | End Node | -96,00 | -77,00 | 5,50 | 11,00 | -97,52 | -77,33 | 12 | 12 |
| | Relay | -67,50 | | 13,00 | | -67,71 | | 12 | |
| **12** | FTD | -90,80 | | 8,25 | | -91,46 | | 12 | |
| | End Node | -107,00 | -91,00 | -0,50 | 9,50 | -111,01 | -91,49 | 12 | 12 |
| | Relay | -68,00 | | 10,50 | | -68,38 | | 12 | |
| **13** | FTD | -97,00 | | 7,25 | | -97,83 | | 12 | |
| | End Node | NC | -90,50 | NC | 10,00 | NC | -90,91 | NC | 12 |
| | Relay | -69,50 | | 11,75 | | -69,78 | | 12 | |
| **14** | FTD | -103,75 | | 4,00 | | -105,27 | | 12 | |
| | End Node | NC | -103,00 | NC | 5,50 | NC | -104,08 | NC | 12 |
| | Relay | -70,00 | | 10,88 | | -70,34 | | 12 | |
| **15** | FTD | -104,25 | | 2,94 | | -106,14 | | 12 | |
| | End Node | NC | -103,50 | NC | 5,50 | NC | -104,58 | NC | 12 |
| | Relay | -69,00 | | 13,25 | | -69,20 | | 12 | |
| **16** | FTD | -108,75 | | 0,06 | | -111,76 | | 12 | |
| | End Node | NC | -112,00 | NC | 2,00 | NC | -114,12 | NC | 12 |
| | Relay | -70,00 | | 14,75 | | -70,14 | | 12 | |

# Appendix F – GitHub repository

Link to the repository containing the code used for the end node application in Nucleo-L476RG:

https://github.com/EvenHopless/LoRaWAN_Relay_TX_L476RG

Link to the repository containing the current progress in porting of LBM to Nucleo-WL55JC:

https://github.com/EvenHopless/LoRaWAN_Relay_TX_WL55JC