MESTRADO EM INFORMÁTICA E SISTEMAS

# The Things Network e infraestrutura de iluminação pública: gestão energética e monitorização ambiental


Autor

**Gonçalo Almeida da Silva**

Orientadores

**José Manuel Meireles Marinho**

**Frederico Miguel do Céu Marques dos Santos**

INSTITUTO POLITÉCNICO DE COIMBRA

INSTITUTO SUPERIOR DE ENGENHARIA DE COIMBRA

DEPARTAMENTO DE INFORMÁTICA E SISTEMAS

# The Things Network e infraestrutura de iluminação pública: gestão energética e monitorização ambiental

Relatório de Trabalho de Projeto para a obtenção do grau de Mestre em Informática e Sistemas

Especialização em Desenvolvimento de Software

Autor

**Gonçalo Almeida da Silva**

Orientadores

**José Manuel Meireles Marinho**

**Frederico Miguel do Céu Marques dos Santos**

Coimbra, julho de 2020

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

# Abstract


The continuous pursuit of more sustainable solutions in all areas of a city has reached the public illumination. The change involves the use of more efficient LED based lights in combination with what is designated as smart luminaires that, besides being more efficient, can be controlled with much more precision than what was previously possible.

This document is the report of the project work for the master's degree in Informatics and Systems by the Coimbra Institute of Engineering. It considers the opportunity of using those luminaires for more than just illumination: to manage the energy consumption, monitor the environment and even control other systems. This work focuses on the problem of noise pollution monitoring and how the Internet of Things (IoT) can help to update this information at a much faster rate than the methodologies currently in use: from an update rate measured in years to an update rate measured in minutes. It also goes a bit beyond by keeping the proposed infrastructure and communication costs down to a minimum. This project started with a survey about the concepts involved and available technologies. It also studied the sound pressure phenomenon and assessed the current solutions as well as their pros and cons. Next, the performance of several common low-cost microphones was evaluated, the connection to the The Things Network (TTN) was described and tested and, finally, a database model, a REST API (Application Programming Interface), and a web-based graphical user interface were proposed.

With the solution proposed in this work, is it possible to create a solution for noise pollution monitoring that allows a near-real-time update, which results in a public utility and a tool to optimise the management of the city. It should be noted that this approach is modular and, therefore, can integrate the public lighting management system, other environmental sensors, or even actuators, allowing both financial and environmental benefits. It also allows the information to be made available for the citizens.



Keywords: IoT, The Things Network, TTN, LoRa, LoRaWAN, Smart City, Sound Pressure Level, Noise Maps, Noise Pollution, Monitoring

# Resumo


A procura continua de soluções cada vez mais sustentáveis em todas as áreas de uma cidade chegou até à iluminação pública. A evolução envolve a mudança para tecnologia mais eficiente, como aquela baseada em LED, em combinação com luminárias inteligentes que, para além de serem mais eficientes, possibilitam o seu controlo com muito mais precisão do que era possível anteriormente.

Este documento é o relatório de trabalho de projeto para a obtenção do grau de Mestre em Informática e Sistemas pelo Instituto Superior de Engenharia de Coimbra e considera a oportunidade de usar estas luminárias para mais do que apenas a iluminação: gerir o consumo energético, monitorizar o ambiente e até controlar outros equipamentos. Este trabalho foca-se na problemática da monitorização da poluição sonora e como a internet das coisas (IoT) pode ajudar a atualizar informação de forma muito mais rápida do que as metodologias usadas atualmente: de uma atualização medida em anos para uma atualização medida em minutos. Para além disso vai um pouco mais além mantendo os custos da proposta infraestrutura e das comunicações o mais baixos possível. Este projeto começou por pesquisar sobre os conceitos envolvidos e tecnologias disponíveis, estudado o fenómeno da pressão sonora, avaliadas as soluções existentes bem como as suas vantagens e desvantagens. Em seguida avaliou-se a performance de vários microfones de baixo custo comummente encontrados no mercado. A ligação à The Things Network (TTN) foi descrita e testada e, finalmente, proposto um modelo de base de dados, uma REST API (Application Programming Interface) e uma interface de utilizador baseada em páginas web.

Com esta solução proposta neste trabalho, é possível criar uma solução de monitorização da poluição sonora que permite a atualização em quase tempo real, que resulta num serviço de utilidade pública, uma ferramenta para otimizar a gestão das cidades. Salienta-se que a solução assim proposta é modular e, portanto, pode integrar o sistema de gestão da iluminação pública, outros sensores ambientais, ou mesmo atuadores, possibilitando assim benefícios ambientais e financeiros. Permite ainda a disponibilização da informação recolhida aos seus cidadãos



**Palavras-chave:** IoT, The Things Network, TTN, LoRa, LoRaWAN, Cidade Inteligente, Nível de Pressão Sonora, Mapas de Ruído, Poluição Sonora, Monitorização

# Table of Contents

# List of Figures

# List of Tables

# Acronyms

**4G** – Fourth generation**.**

**5G** – Fifth generation**.**

**ABP** – Activation By Personalization

**ADR** – Adaptive Data Rate.

**Cat.** – Category**.**

**$CO_2$** – Carbon dioxide.

**CO** – Carbon monoxide.

**CoAP** – Constrained Application Protocol.

**dB** – Decibel**.**

**dB(A)** –A-weighting decibel.

**DEP** – Department of Environmental Protection.

**EWS** – Early Warning System.

**FFT** – Fast Fourier Transformation.

**HTTP –** Hypertext Transfer Protocol.

**$I^2C$** – Inter-Integrated Circuit (I-squared-C)

**IFTTT** – IF This Than That.

**$I^2S$** – Inter-Integrated Circuit Sound.

**JSON** - JavaScript Object Notation.

**IEEE –** Institute of Electrical and Electronics Engineers.

**ISM** – Industrial Scientific and Medical.

**ISO** – International Organization for Standardization.

**MIC** – Message Integrity Code.

**MQTT** – Message Queuing Telemetry Transport.

**$NO_2$** – Nitrogen dioxide.

**NP** – Norma Portuguesa.

**$O_3$** – Ozone.

**OLED** – Organic Light-Emitting Diode.

**OTAA** – Over The Air Activation.

**PCM** – Pulse Code Modulation.

**PDM** – Pulse Density Modulation.

**PM10** – Particulate Matter 10 µm.

**PM2.5** – Particulate Matter 2.5 µm.

**REST** – REpresentational State Transfer.

**RF** – Radio Frequency.

**SDK –** Software Development Kit.

**SLM –** Sound Level Meter.

**UI –** User Interface.

# Definitions

**6LoWPAN** – *Ipv6 over Low-Power Wireless Personal Area Networks* – network technology allowing IPv6 packets to be transmitted using small link layer frames.

**ADC** – *Analogue-to-Digital Converter* – is a system that converts an analogue signal, to a discreet digital signal.

**AES** – *Advanced Encryption Standard* – symmetric-key encryption algorithm widely used today to secure communication and data.

**API** – *Application Programming Interface* – software element that allow different entities to exchange data and/or functionality through a defined interface.

**DAC** – *Digital-to-Analogue Converter* – is a system that converts a digital signal, to a continuous analogue signal.

**CEPT** – *European Conference of Postal and Telecommunications Administrations* – conference gathering 45 members, to harmonize from the technical point of view the regulation and the coordination of European regional positions for the works of the International Organizations of the sector, the International Telecommunications Union (ITU) and the Universal Postal Union (UPU). It includes the ECC.

**ECC** – *Electronic Communications Committee* – has as objectives: to develop in an European context regulation policies at the electronic communications level, to create in Europe an harmonized plan for the effective use of the radio spectrum, of the numbering resources, taking always into account the needs of consumers and the industry, to promote European cooperation in the preparation of ITU fora, to encourage deregulation and liberalization as well as to encourage free circulation and use of radio equipment as a way to promote an open and more competitive market.

**EC-GSM** – *Extended Coverage GSM* – extension of the GSM technology for extended coverage to be used by IoT devices.

**eMTC** – enhanced Machine Type Communications – is an extension of LTE, focused on IoT and taking advantage of currently deployed LTE stations.

**GSM** – *Global System for Mobile communications* – standard developed for communication using second generation mobile devices.

**HID** – *High Intensity Discharge* – lamp technology in which light is produced by an electric arc between electrodes.

**ICT** – *Information and Communication Technologies* – is referred as the broader group of the information technologies and communication technologies where information can be collected, transformed, transmitted, accessed or stored.

**IDE** – *Integrated Development Environment* – application used to aid a programmer in coding.

**IoT** – *Internet of Things* – concept of extending internet connectivity to a range of other devices, or *things*, transmitting data between them without the need of human interaction.

**JFET** – *Junction gate Field-Effect Transistor* – a simple type of field-effect transistor.

**LED** – *Light Emitting Diode* – electronic device that gives off light when it receives an electrical current.

**$L_n$** – Average noise level in the night period, 4 meters above ground.

**$L_{den}$** – Weighted average noise level in a 24-hour period, 4 meters above ground.

**LoRa** – *Long Range* – communication technology focused on long-range communications with low power consumption based on chirp spread spectrum technology. Uses the unlicensed spectrum.

**LoRaWAN** – *LoRa Wide Area Network* – layer built on top of LoRa providing typical MAC layer functionality.

**LPWAN** – *Low-Power Wide-Area Network* – wireless network communication network designed for long-range communications using low-power.

**LTE** – *Long Term Evolution* – evolution over the 3G networks, offering greater speeds but inferior to those of 4G.

**M2M** – *Machine to Machine* – type of network for direct communication between devices, without user intervention.

**MAC** – *Medium Access Control* – part of the second layer of the Open System Interconnect model providing abstraction of the physical layer and controls device interaction.

**NB-Fi** – *Narrowband Fidelity* – technology that focuses on IoT and M2M communication, using the unlicensed spectrum and narrowband technology. Developed by WAVIoT, it supports duplex communication and encryption.

**NFC** – *Near Field Communication* – communication technology to exchange data between two electronic devices at close distance.

**PCB** – *Printed Circuit Board* – base board that provides support and electrical connection to components.

**RFID** – *Radio Frequency Identification* – communication technology to exchange data between two devices, from close distance (passive tags) to hundreds of meters (for active tags).

**RPMA** – *Random Phase Multiple Access* – is a wireless technology focused on IoT and M2M communications. Operates at the 2.4 GHz frequency and features encryption.

**SPL** – *Sound Pressure Level* – logarithmic representation of the local variation of pressure relative to a reference value caused by a sound wave.

**SSL** – *Secure Sockets Layer* – security protocol for authenticated and encrypted links between devices.

**TCP** – *Transmission Control Protocol* – connection-oriented communication protocol, part of the transport layer of the Open System Interconnect model, providing reliability.

**TTN** – *The Things Network* – Global, decentralized, open and free network targeted at Internet of Things, based on LoRaWAN technology.

**UN** – *United Nations* – International organization made up of 193 Member States and founded in 1945.

**UNB** – *Ultra-Narrow Band* – technology that takes advantage of the filtering capabilities of narrow-band signals to boost signal-to-noise ratio;

**USB** – *Universal Serial Bus* – industry standard of hardware (connectors) and protocols for serial communication and power delivery between devices;

# 1 Introduction

Environment is a general concern. It is not just from environmentalists or general individuals, but also from the State and Companies. A sustainable future is a more and more present objective. One of the ways to minimize the impact on the environment is to reduce or optimize electric energy consumption, which reduces the constant pressure on our natural resources and, as a consequence, on our environment. One of the key areas that help achieving this goal consists in replacing traditional public illumination infrastructures by solutions with better efficiency and less consumption.

In fact, most of the public lightning in the cities of Europe date back from the 1960's and the efficiency of the system was not a concern at that time. Nowadays most of the light sources used in public lighting are based in HID (High Intensity Discharge) technology, although the shift to more efficient technologies is in underway as many municipalities are changing to LED (Light Emitting Diode) technology. Additionally, just choosing the most efficient light source is not enough as the total efficiency of the system depends not only on the light source but also on the luminaire used. For instance, the possibility of dimming the lights in relation to changing environment conditions also leads to energy savings. To replace the old technology with a newer and environment friendly one we can simply replace the light bulbs, but the ideal is to replace the complete system [1].

Several cities have already started to change their public lightning systems from conventional to the more efficient LED technology. This is a step in the right direction as this new technology can provide significant savings in energy consumption [2]. This reduction in energy consumption will also have a positive effect in reducing the $CO_2$ emissions. Taking into account this shift in technology, it is the perfect time to develop a new LED based system that is not just more effective but also smarter, in the sense that it is possible to control each individual lamp and collect information about its performance (i.e., consumption and any failure occurrences). This creates the possibility of further saving as relying on a manual monitoring solution is expensive and slow. In fact, traditionally, when a streetlight malfunctions, this information can reach the management team in one of two ways: by a complaint of a citizen or by having a team that at predefined intervals, goes through all the streetlights and checks their status. This has additional cost and is not eco-friendly.

To make this system a real "thing", each lamp must have an electronic control component that can monitor and communicate with the system. As such, the idea of integrating additional sensors for environment monitoring emerged. These sensors enable gathering information about environmental characteristics and sending it to the system that will later analyse and/or make it available as is. The luminaires are in an ideal position to collect these city parameters as they are close to the roads, where collection information about the noise and air quality is important. On the other hand, they are also near parks and squares, which is also an important locus to collect this data.

This is a step forward in making the cities more intelligent, collecting data of several points and making them available for the benefit of all its habitants (i.e., the well-known paradigm of Open Data), the true concept of a Smart City. Additionally, and even more important, with clear advantages to the environment.

## 1.1 Motivation

The change from the traditional light sources to new and more efficient ones is already underway. This change has two main motivations already mentioned: economic and environmental. It has the added benefit of improving the lightning quality and providing greater flexibility. The use of different shaped lamps creates new possibilities for creative lightning. Following this change, these new lamps and luminaires are often smarter, with intelligent control systems that allow for further savings and the improvement of the service provided. The lights can be dimmed in response to environmental conditions and can report when a malfunction occurs. As these luminaires already contain an electronic control element for this monitoring and control functions, adding a set of sensors for environment data will not add much additional cost to the solution.

### 1.1.1 Limitations of usual solutions for environmental data collection

Environmental data collection is not new. There are meteorological stations collecting temperature, and precipitation for many years. Today, the environmental data of a city is collected by:

- Meteorological stations, collecting data such as temperature, humidity, atmospheric pressure, precipitation, wind direction, and speed and solar radiation [3];
- Fixed air quality monitoring stations, collecting air quality parameters such as Ozone ($O_3$), Nitrogen Dioxide ($NO_2$), Sulphur Dioxide ($SO_2$), Particulate Matter of 10 micrometre ($PM_{10}$), and 2.5 micrometre ($PM_{2.5}$) [4], [5];
- Graphical representation of noise levels in exterior environment in Noise Maps [6]. For this representation two indicators are used: $L_n$, representing the average noise level in the night (between 23:00 and 7:00), and $L_{den}$, representing a 24-hour period weighted average.

The data collected by the meteorological stations and from the fixed air quality monitoring stations, although collected automatically and several times a day, covers only specific spots of the city, where these instruments are located. Some of the parameters vary greatly depending on where they are collected. This is especially true when measuring air quality parameters and noise levels as they change with the distance to the source.

The sound level monitoring method does not allow for real time monitoring. The data is presented in a static form, by using maps that are updated according to the regulation. An example of such a map is presented in Figure 1-1, showing the different colours corresponding to the different environmental noise levels. This example is an extract from the municipal master plan of Coimbra [7], dated from 2011, corresponding to $L_n$ noise descriptor (i.e., from 23:00 to 07:00 hours). The higher dB levels correspond to higher noise levels. These maps are created by means of a complex process based on [8]–[10]:

- Cartography;
- Location of buildings;
- Location of noise sources (transport infrastructures and fixed sources);
- Noise propagation barriers (walls, acoustic barriers, for example);
- Meteorological data;
- Several data collecting spots;
- Software based calculation.

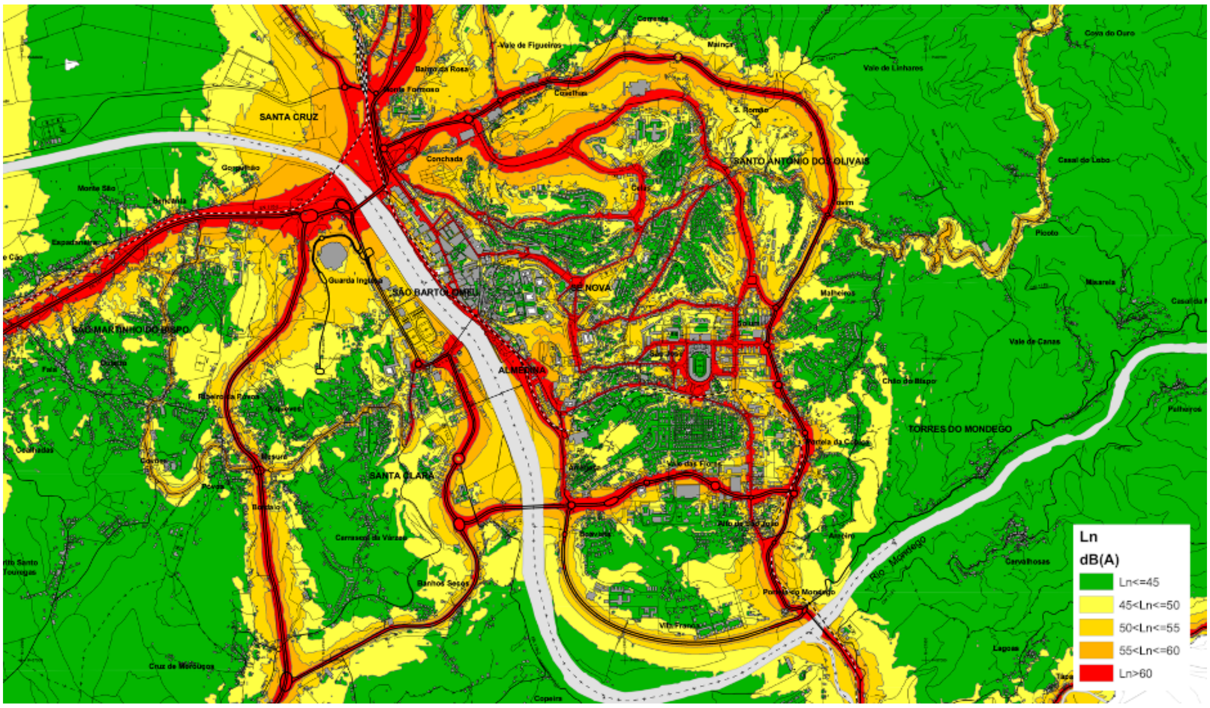


Figure 1-1 - Extract from the noise map of Coimbra. Source [7]

This process takes time and cannot be used for a real time sound monitoring solution. Furthermore, citizens often complain about noise pollution that is limited in time: construction works, bars, discos, and other sources that are not represented in these maps.

### 1.1.2 Usage of smart luminaires for environmental data collection

Leveraging the change in public lightning, from the traditional to the more efficient and smarter technology, it is the perfect opportunity to incorporate new features in this infrastructure. Each control module in this new system can incorporate not only the control and performance monitoring solution but also sensors that collect data regarding the surrounding environment that can be analysed or made available as is. As presented in Figure 1-2, a modular solution is considered in this work, allowing to further increase the efficiency of the system.

As such, the system incorporates a light control system allowing for not only turning the lights on and off but also dim them according to the environmental conditions. It collects data about its performance, with parameters like light emission, temperature of operation and power consumption. In addition to the performance parameters, each luminaire is fitted with a set of environment sensors that collect data about the environment in a much more detailed way than from a limited number of stations in a city. This information is collected, stored, and analysed in a centralized system and provided for the operators of the system. The city citizens are also allowed to access the data.

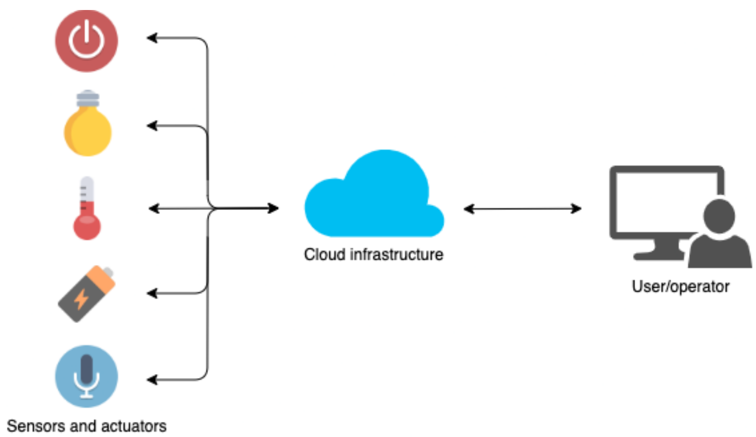


Figure 1-2 – System overview

By its real-time nature, the system proposed in Figure 1-2 allows for other uses, besides simply collecting environment data:

- Analysis of the environment in a city in a much more detailed way, both in time and space;
- Detection of environment changes in a city related to architectonic elements;
- Influence of changes in the city, such as traffic changes, in the city's environment;
- Detection of abnormal conditions, such as high noise levels, air pollutants, or others.

This environment data collection should be stored and used for further analysis but also be made available to the citizens creating a true Smart City.

## 1.2 Objectives

Based on the opportunities that the ongoing development of new and smart(er) luminaires enables, the aim of this work is to propose an IoT-based (Internet of Things) system that aims to capture the noise levels in a city and that is based on sensors incorporated in luminaires. As the control module in each luminaire can be connected to several sensors and/or actuators simultaneously, one luminaire can send and receive different types of information using the same communication channel. This helps to reduce costs and avoids component duplication.

The proposed solution was intended to be deployed massively in a city-wide area. To this end, it was designed to be modular, low-cost, able to integrate the control of the luminaires, capture the environment data reliably, and transmit this data to a central system. Although this approach can be applied to different types of data, the main focus of this work was on noise pollution. As such, microphones were the sensors of interest. The choice of noise pollution was based on the difficulty of collecting this type of data due its variability in both time and space.

For the study of the feasibility of the solution, several aspects were considered, which resulted in the following tasks:

- Study the concepts involved in creating such a solution;
- Study the technologies available;
- Study the sound pressure phenomenon;
- Survey the available solutions;
- Evaluate the suitability of the commonly available sensors (microphones);
- Test the connection to the The Things Network (TTN);
- Propose a database model;
- Propose a Web API;
- Propose a web-based user interface.

The results of these tasks are a roadmap for the development and implementation of the proposed solution.

## 1.3 Outline

The remainder of this report is structured as follows:

- Section 2 includes the concepts, technologies, state of the art as well as related work;
- Section 3 includes the definition of the proposed solution, architecture, components and the wireless communication structure;

- Section 4 includes the components tested, the experiments conducted and their results, the TTN connection tests and the definition of a database model, REST API and user interface.
- Section 5 presents the conclusions, limitations and a few questions for future work.

# 2 Related work and state of the art

In order to understand the kind of smart solutions that are in use today, this chapter presents an overview of the different solutions available and their applicability to the Smart Cities concept. It focuses not only on the quest for different communication infrastructure solutions, but also in different sensor systems located on the luminaires or outside them.

## 2.1 Smart City

Nowadays, more and more people are moving to the cities. This reality brings enormous challenges for urban growth and sustainability. As the overall world population increases and the number of people in rural areas tend to stop increasing, the population in cities is growing at an ever more accelerated pace. In an article by Ritchie and Roser [11], the urbanization data is presented showing how the population in urban and rural areas has evolved from 1960 to 2017. This evolution is presented in Figure 2-1 where it can be seen that the number of people living in cities overcame the number of people in rural areas in 2007. This chart was based on data from the UN (United Nations). This organization estimates that the percentage of the world population living in urban areas will reach 68% by 2050 [12].

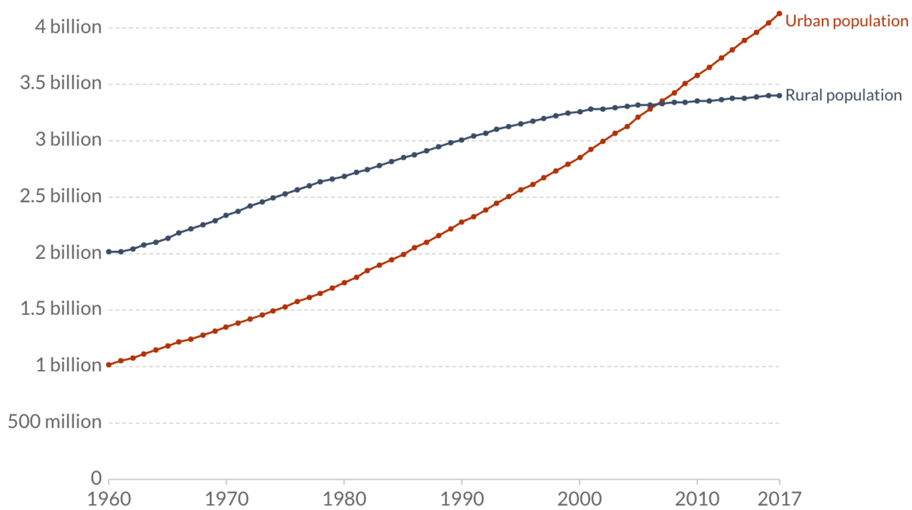


Figure 2-1 – Number of people living in urban and rural areas, World, 1960 to 2017

Mark Deakin [13] refers to some cities being called smart without defining what the concept means and/or failing to provide evidences of its affirmation. It means more than just making use of information and communication technologies (ICT). It means using these infrastructures to create social interactions as well as social and environmental development.

A profound study on what this concept of smart city means is further explained in other publications [14], [15].

A smart city uses data collected from a variety of electronic devices, transmitted over a communication network, in order to manage more efficiently all its assets and resources. It means also to share that information with all its citizens with the purpose of improving both the quality of life and governance. Many of these devices can be in areas where electrical power is not available and are required to use batteries. This requirement creates the need of low power consumption.

The data is collected with the use of, but not limited to, devices usually related to the concept of IoT [16]. These devices support the Smart City by providing added-value services for their residents. According to Bauer et al. [17], the Internet of Things can be defined as a network containing all "smart" devices with some sort of sensing mechanism that can communicate via the Internet with other smart devices or the cloud without human interaction. These devices, by their nature of autonomous acting, will populate our world in the next years. In fact, it is expected that the number of connected devices increases from the 7 to 10 billion installed in 2015 to 26 to 30 billion in 2020 [18].

Some examples of the application of the concept of smart city are listed below:

- Noise pollution monitoring;
- River overflow detection;
- Water leakage detection;
- Flood sensors;
- Bus schedule signs;
- Street lighting;
- Smart grid;
- Waste management;
- Air pollution monitoring;
- Parking;
- Traffic.

## 2.2 IoT

As previously mentioned, a smart city collects data from a multitude of sources, much of them IoT devices. These smart devices (electronic devices capable to connect, collect data, and/or process data) are becoming more ubiquitous and more numerous. Based on the growing number of traditional connected devices (mobile phones, tablets, computers, fixed line phones) and new kind of devices, soon the new devices will have the larger installed base in the world [18]. As the communication infrastructure has been developed with the traditional devices in mind, a new and more adapted communication infrastructure is needed [19].

### 2.2.1 Communication infrastructure

There are currently several communication infrastructures and technologies like telephone landlines, fibre optic cables, coaxial cables, Bluetooth, Near Field Communication (NFC), Radio Frequency Identification (RFID), Global System for Mobile communication (GSM), and Wi-Fi to name a few.

One characteristic of the new, small and portable smart devices is their low power consumption. They can collect and/or process significant amounts of data with very low power consumption [20]. Traditional networks have different approaches to the communications. For GSM (Global System for Mobile communications), the priority is data and voice transmission, at high data rates for medium distances, at the expense of power consumption and a contract with a network operator. For Bluetooth, the focus is on speed and low power consumption, at the expense of distance. IEEE 802.11, or the more commonly used Wi-Fi designation [21], has more coverage and consumes more power than Bluetooth but has less coverage than GSM. These technologies are not ideal for the new smart devices [22].

Another differentiating factor is the typical use case for smart devices: they transmit small amounts of data in predefined intervals. Humans, using the traditional communication channels, usually require more bandwidth during a non-determined interval such as a phone call, opening a web page or download data. These new devices, which have very low power consumption and long range requirements, can only transmit (or receive) data in small amounts, which is not the typical use case of the traditional networks [22].

Several technologies have emerged with the purpose of answering to these new requirements: Sigfox, Random Phase Multiple Access (RPMA) [23], NB-Fi, Weightless, DASH7, 6LoPWAN, and LoRa [24]. With this need in the Machine-to-Machine (M2M) networking business, the operators have developed their own alternatives such as NB-IoT, eMTC, EC-GSM, and communication standards such as those based on IEEE 802.15.4 (Zigbee), which make use of mesh network topology in order to increase their coverage. The alternatives in the Low Power Wide Area Network (LPWAN) use an infrastructure-based approach that has some benefits as well as drawbacks: it comes at a cost of reduced bandwidth but simplifies the management and reduces the required networking firmware for end devices [25]. In fact, when compared to a mesh network, a star topology allows all nodes to contact a central collector node, simplifying the network and the end devices as they do not need complex communication stacks. Nowadays, the focus on the future development of 5G promises to bring greater application in the area of IoT [26].

The comparison of a number of different technologies is presented in Figure 2-2 (range of the different technologies vs. their data rate). The energy consumption is related to the data rate, as higher data rate implies higher energy consumption. In the area identified as 1 in Figure 2-2, many competing standards as well as competing companies and consortiums make the use of these technologies in IoT difficult as they all compete for the same space. The lack of interoperability, incompatibilities in higher communication layers (6LoWPAN vs ZigBee), and competition between consortiums may cause additional conflicts. With many options in the same space, the developers can find difficult to develop for a standard that is incompatible with the others and without certainty about what standard will prevail. Following the same uncertainties, the users also may be reluctant to acquire a product that may not be compatible with other devices of the same type.

The result of such difficulties, combined with the need of a technology that can transmit data at long distances – allowing for a simpler infrastructure – and that allows for low power consumption – making possible to deploy IoT devices in places that only battery power can be used, creates a perfect area for communication technologies to emerge. The area, identified as 2 in Figure 2-2, is the area for low-data-rate, low-power, and high range applications where less technologies exist. It is the perfect area for IoT communications [17]: it covers long ranges with little bandwidth and allows for low power consumption.

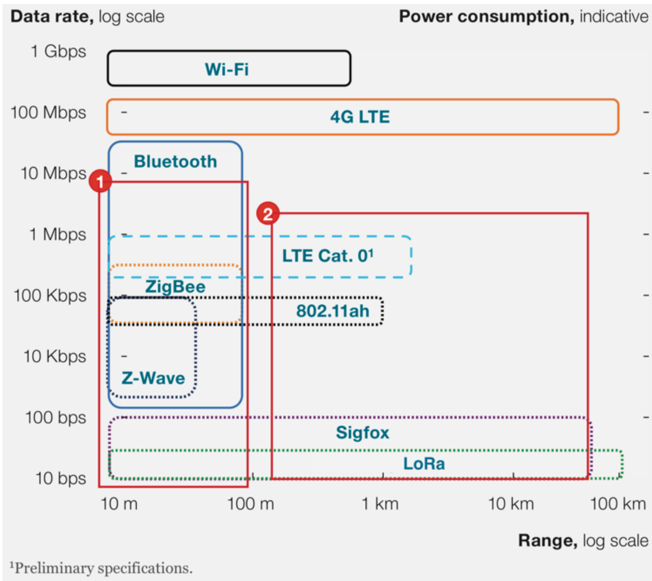


Figure 2-2 – Comparison of different communication technologies and their respective data rate, range, and indicative power consumption. Source: adapted from [17].

### 2.2.2 LoRa, LoRaWAN and TTN

As mentioned before, as more and more devices require data connection, the interest in LPWAN increased. Sigfox made significant advancements in the earlier days. It is a variation of the cellular system and allows for remote devices to connect to an access point using Ultra-Narrow Band (UNB). It is a proprietary technology, developed by Sigfox of France, operating in the 868 MHz band, and with a spectrum divided into 400 channels of 100 Hz. No public specification is available. Data rate can reach up to 100 bps. According to Sigfox, each access point can handle up to a million end devices, with ranges up to 30-50 km in rural areas and 3-

10 km in urban areas [27]. Sigfox started in 2009, in France, with the first LPWAN implementation.

LoRa was developed later. The physical layer of LoRaWAN (LoRa) was developed initially by Cycleo, France. This company was later acquired by Semtech which, currently, remains the sole LoRa integrated circuit producer [28]. This technology is promoted by the LoRa Alliance [29] and aims to be used in devices where the energy consumption is of paramount importance. Dividing it in two distinct layers, it is common to refer to the physical layer as LoRa and to the MAC layer as LoRaWAN (see

Figure 2-3). The LoRa physical layer uses the Chirp Spread Spectrum (CSS) radio modulation technology and operates in the EU863-870, US902-928, CN470-510, AU915-928, AS920-923, AS923-925, KR920-923 and IN865-867MHz bands depending on the region [30]. The modulation technique is proprietary of Semtech [20]. The typical range of LoRa is 2km to 5km in urban areas and 15km to 30km with free line of sight [31]. The maximum recorded distance of a LoRa packet transmission has been registered in April 2020 with 832km, with 25mW transmitting power, in an ideal situation with the transmitter located in a weather balloon [32] and breaking the previous record set in 2019 [33]. Another key benefit of LoRa is the use of the unlicensed radio spectrum allowing for free use by anyone. Of the several LPWAN solutions available, LoRaWAN is the most adopted [28].

TTN is built on top of LoRaWAN as detailed in

Figure 2-3. It focuses on providing an open-source decentralized network to be used by everyone [35]. It is composed of the software stack that allows to add gateways, devices, and applications by anyone, continuously growing the network. Every new added gateway can provide access to any device in a secure way as the payloads are encrypted from the device to the application. This is very different of the traditional networks were the operators build the infrastructure and charge for the data transferred within the network. The TTN architecture, from the devices to the application, is presented in Figure 2-4.

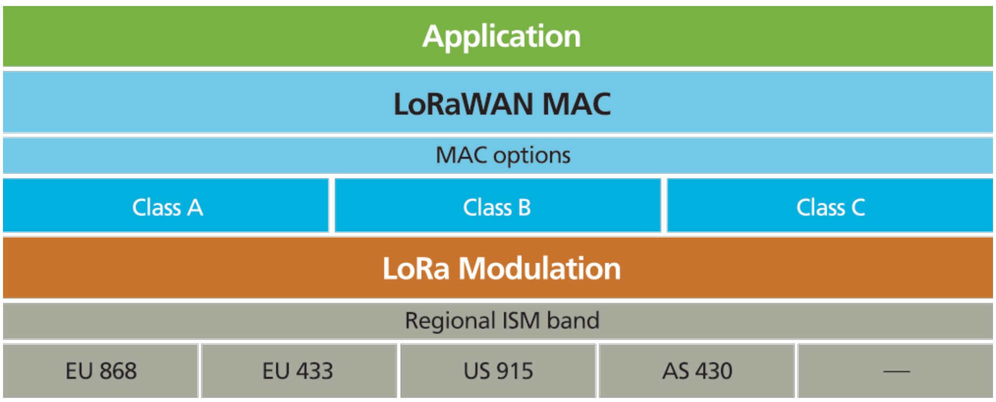


Figure 2-3 – LoRaWAN layers, from the regional Industrial, scientific and medical (ISM) bands to the application. Source: [34]

In simple terms, the TTN devices have the architecture presented in Figure 2-5. The devices contain a software LoRaWAN abstraction layer that interfaces with the LoRa hardware to send the information encoded in the LoRa protocol. Running on the Gateway, there is a program designated as packet forwarder, that receives the data from the LoRa hardware and forwards it by means of an IP stack to the Network Server.

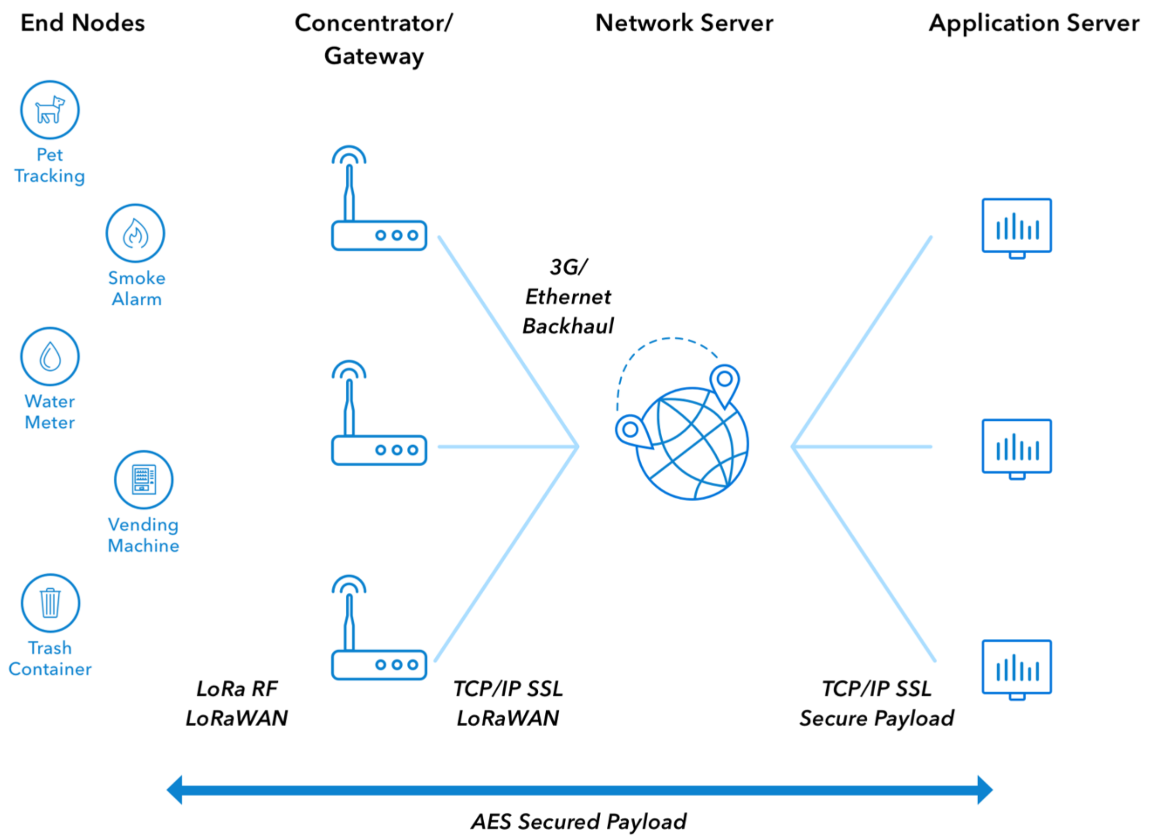


Figure 2-4 – TTN architecture. From the devices to the application featuring end-to-end security. Source: [36]

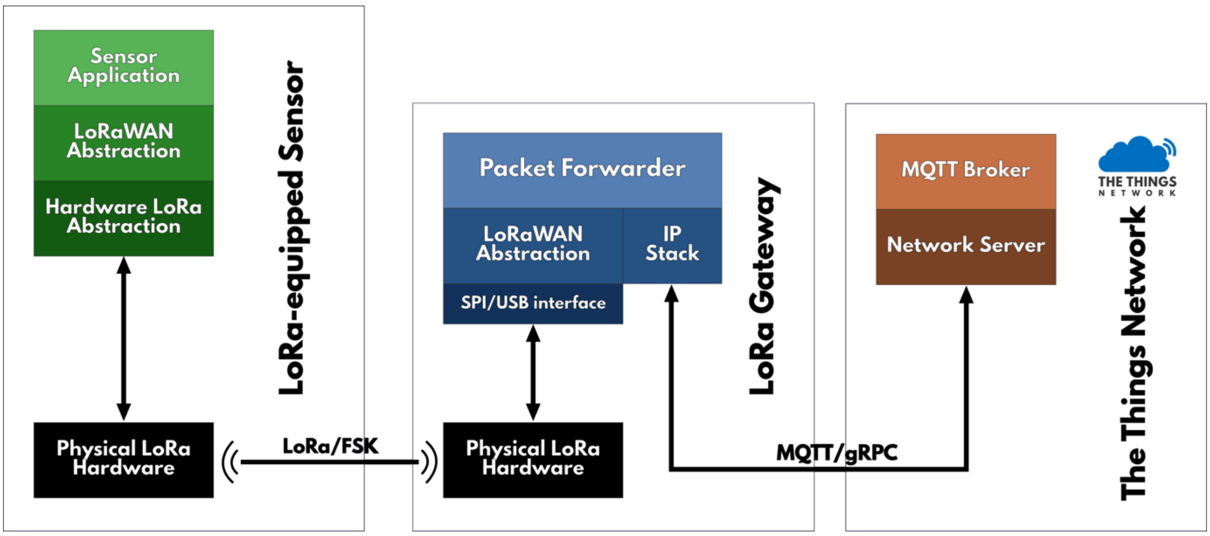


Figure 2-5 - TTN packet forwarder

## 2.3 Sound pressure level

As the main focus of this work is noise pollution, and noise pollution is directly linked to sound pressure level, a simple question emerges: how can sound pressure levels be determined by a simple IoT device? In order to answer this question, a few basic concepts must be clarified first.

Sound can be described as pressure waves that travel through a medium (for the purpose of this work, this medium is the air). A microphone converts the varying pressure levels into an electrical signal. The conversion of this signal into a digital value happens in the controller (in the case of an analogue interface microphone) or in the microphone module (in case of a digital interface microphone). Regardless of the ADC (Analogue to Digital Converter) where this conversion happens, it is always a conversion of an analogue signal to a digital representation.

The analogue representation of the sound is a continuously changing electrical signal that represents the continuously changing pressure exerted on the diaphragm. Through the use of an ADC, this continuously changing signal is represented by a discreet digital representation. It is important to emphasize that this value represents the sound wave and not the sound pressure level (SPL) at a given time. To further illustrate this concept, consider the graphs represented in Figure 2-6 and Figure 2-7. The first graph represents two sound waves with the same amplitude, but different frequencies (the sound wave corresponding to the solid line has twice the frequency of one represented by the dashed line). This will result in the same SPL. The second graph represents two sound waves with the same frequency but different amplitudes and this results in a higher SPL value corresponding to the sound wave represented by the dashed line (sound wave corresponding to the solid line has half the amplitude of one represented by the dashed line).

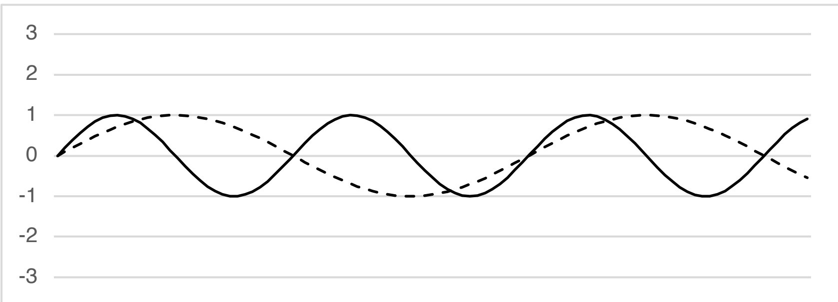


Figure 2-6 – Theoretical representation of two sound waves

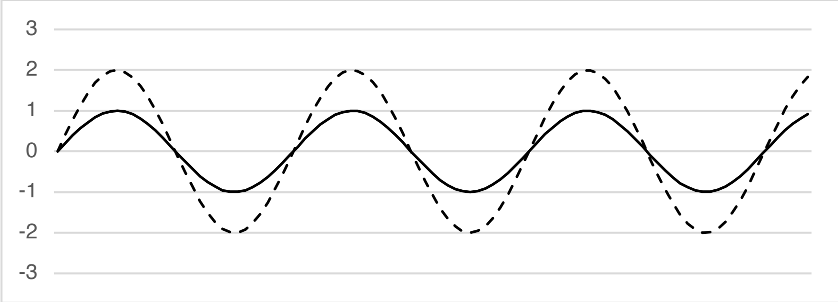

Figure 2-7 – Theoretical representation of two sound waves

Sound pressure can be defined as the difference between the local pressure caused by a sound wave and the ambient pressure in the same point. It is measured in Pa (Pascal) and the audible range varies between the threshold of hearing (20μPa) to the pain threshold (20Pa) [37]. However, the usual scale used for measuring sound pressure is dB, a logarithmic scale:

$$20\ log_{10}\frac{p_1}{p_2} = 10\ log_{10}\left(\frac{{p_1}^2}{{p_0}^2}\right) dB \quad (1)$$

In Equation (1), $p_1$ represents the actual measured sound pressure and $p_0$ is the reference for the threshold of hearing (20μPa). At the hearing threshold, this equation results in the value of 0dB and, at the pain threshold, it results in 120dB.

## 2.4 Related work

Previous work already exists concerning smart cities, IoT, and environment. A few examples are presented in the next sections, with a short explanation of the results and conclusions.

### 2.4.1 San Diego streetlights get smart

In the United States of America, California's San Diego East Village neighbourhood, some of the streetlights were modified to include sensors to monitor sound, movement, temperature, humidity and other air characteristics [38]. This test area is expected to be expanded from the 50 smart sensor-driven streetlights to about 3,200. Each of the lamps monitors an area of 36 meters by 54 meters. These smart devices will first start to monitor the vacant parking spaces and alert the authorities for illegally parked cars. This application is expected to be just the beginning as the data will be available for the city and software developers to create new services for residents and visitors. This is a costly solution both in construction and in communication transmission, despite its high capacity and multifunctionality. Its characteristics include an Intel Atom processor, 512 GB of storage capability, Bluetooth, Wi-Fi, two 1080p cameras, acoustic, temperature, pressure, humidity, vibration and magnetic field sensors, while its communication is performed by GSM/LTE.

### 2.4.2 New York City SONYC project

Known as "The City that Never Sleeps", New York City is known for being always active as well as having high levels of noise. In 2016, the cities' information/complaints line received an average of 48 noise complaints per hour [39]. It is estimated that 90% of the residents are exposed to noise levels exceeding the Environmental Protection Agency's (EPA) guidelines on levels considered harmful to people [40]. This problem originated the regulation of sources of noise since the 1930s, and the revision, in 2005 [41], went to effect in 2007. The problem now was that the city lacked the tools to effectively monitor and understand the sources of noise in order to be able to enforce the new regulation. An automated system capable of monitoring the sound levels in the city was needed.

Before this project, after registering a complaint, an officer from the Department of Environmental Protection (DEP) would carry out an inspection using a sound level meter (SLM) such as a Bruel & Kjaer's 2238A. This meter has a high level of precision in acoustic measurements, higher than even required. For the measurements that are less accurate, a Type/Class 2 standard device is required [42]. The city's regulation requires that the offending noise does not exceed the ambient noise by more that 10dB at 15 ft. This requires a before and after evaluation of the noise in the area that can only be achieved with a constant measure of the sound levels. Previous attempts have used the Bruel & Kjaer's Noise Sentinel system, but its cost limits their scalability and deployability, with each sensor costing more than $15,000 USD. This led to the development of the Sounds of New York City (SONYC) project [43].

The solution is based on a network of 35 low-cost standalone modules with acoustic sensors, capable of collecting, analysing, and transmitting the data by Wi-Fi [44]. In this use case, not only the intensity of sound was monitored but also the sounds were annotated or labelled with their primary cause. This means that the sensors must identify the cause of the sound and label it without human interaction. One important conclusion that was possible to reach is that using this technique, it was verified that although it was not possible to observe a violation by the DEP officers, 94% of the complaints were authentic. This work allowed to conclude that even if a DEP official does not observe a violation, it does not mean that the complaint is not genuine. Often a DEP official is recognized, and the noise activities are reduced invalidating the measurements. This system requires Wi-Fi coverage of the entire targeted area as well as energy supply for those modules. Due to the energy requirements of the chosen technology, these modules cannot be supplied by batteries and do require external power supply.

### 2.4.3 Calgary acoustic monitoring

In the case of Calgary, the first application of the LoRaWAN technology was the real-time monitoring of the noise pollution in the municipality [45]. In September 2017 the network was completed and with 3 gateways covers a large area of the Calgary region. Supported by existing radio towers and fibre networks the LoRaWAN network took only a minimal additional cost. They started with the example from New York city where the use of Wi-Fi and live streaming originated a much greater cost and required electrical power supply to the devices. This had higher maintenance costs as well. In the case of Calgary's approach, the devices were powered by batteries. This system clearly managed to have a higher flexibility as the modules can be installed away from a power supply network. Another key advantage was that a big part of the city was covered with only 3 gateways (antennas). This implementation of this solution required a minimum amount of investment when compared to the solution it was based upon.

### 2.4.4 Tabasco province Early Warning System (EWS) for river overflows

Near the Yucatan's Peninsula, in the southern Mexican province of Tabasco, a study was conducted with the objective of developing an Early Warning System (EWS) for river overflows. The communication technology chosen for the implementation was LoRa, mainly for its considerable range but also for the low power consumption. This allows for the sensors to be placed away from the power grid and powered by batteries. This decision did also take into account the work done previously [46], where the potential of the LoRa communication protocol is tested and the maximum range for the application is determined.

They also leverage the knowledge from previous works with different kinds of sensors and communication protocols. The hardware for receiving, analysing and sending the data is the following:

- 1 Raspberry Pi 3 (based, again, in previous work [47]);
- 1 Ultrasonic sensor
- 2 TTGO programmable cards (consisting of an OLED screen, a LoRa radio module and a battery);
- 1 Wemos D1 card (containing an ESP8866 chip for the processing tasks).

The Raspberry Pi is responsible for saving the data in the database and publish the result in the Twitter account. The implementation of the system included a sensor and a receiver node. The sensor node was built using the ultrasonic sensor and one of the TTGO boards, connected using serial communication. The receiver node was built using the other TTGO board and USB communication to the computer. The communication between the node and the receiver is done wirelessly using LoRa. It can be easily used in different points of the river, particularly upstream, where it can collect the water level data and relay that information to a central system where the data can be analysed, and a warning can be sent.

### 2.4.5 Padova sensor testbed

Based on previously developed work [16], the university of Padova, Italy, deployed a sensor testbed in its premises containing more than three hundred sensor nodes in order to demonstrate the smart grid and health care services. For the smart grid, devices are to be deployed to monitor the energy consumption of the city. This energy consumption takes place in several services such as:

- Public lighting;
- Transportation;
- Traffic lights;
- Control cameras;
- Heating and cooling of public buildings.

The energy consumption of public buildings is of great importance as the buildings in general are accountable for approximately forty percent of the total energy consumption. Studies in this area aim to reduce the total energy consumption as well as reduce the related $CO_2$ emissions [48]. The goal is to determine which of services consumes the most and to set priorities in order to improve their performance/power ratio. This approach can also be applied to local power producing structures such as solar panels. In addition to the energy monitoring devices, this study also makes use of wireless nodes capable of collecting environmental data such as, for example:

- CO level;
- Air temperature;
- Humidity;
- Vibration;
- Noise.

These devices will be placed in the streetlight poles and transmit data through the use of a central gateway unit. This approach will allow to monitor not only the energy consumption of the public lighting and the environmental parameters but also the light intensity of each post. The focus was the 6LoWPAN technology, the system architecture, and general data collection of a city not only capturing data regarding the street area but also the buildings. System architecture components such as HTTP-CoAP Proxy, network layer, link layer, and data base server were studied, as well as data analysis. This study allows a better view of the entire system, as well as pros and cons of the chosen solutions.

### 2.4.6 IoT-enabled solutions in Smart Waste Management

In the previous examples, the focus was in the smart city solutions, with different communication systems. In a different work [49], the IoT paradigm is applied to the smart waste management problem. The main goal is to evaluate different solutions for communicating the captured data. Several approaches are studied, and their primary strengths and weaknesses are compared. The included projects are: SmartBin [50], Smart City Waste

Management using GSM [51], IoT based smart garbage and waste collection [52], smart garbage monitoring and clearance system using IoT [53], smart garbage monitoring system for waste management [54], smart garbage collection system in residential areas [55], intelligent bin management system for smart cities using mobile application [56], SWACHH [57], Smart and wireless waste management [58], waste bin monitoring system using integrated technologies [59], a versatile scalable smart waste-bin system based on resource limited embedded devices [60], IoT-enabled citizen attractive waste management system [61], smart bin: an intelligent waste alert and prediction system using machine learning [62], and cloud-based smart waste management system [63].

This study work demonstrates how IoT technology can be applied throughout the city and in particular, to waste management. The typical smart waste management architecture is shown in Figure 2-8, with the representation of the bin with a sensor, alerting automatically the waste management authority, or manually through the use of a phone. All of the cases covered in this study have a method for sensing how full the waste bin is and a method for transmitting that data to a central point. The different methods of transmitting this data are compared. Other aspects are also compared such as the effectiveness of the sensing device.

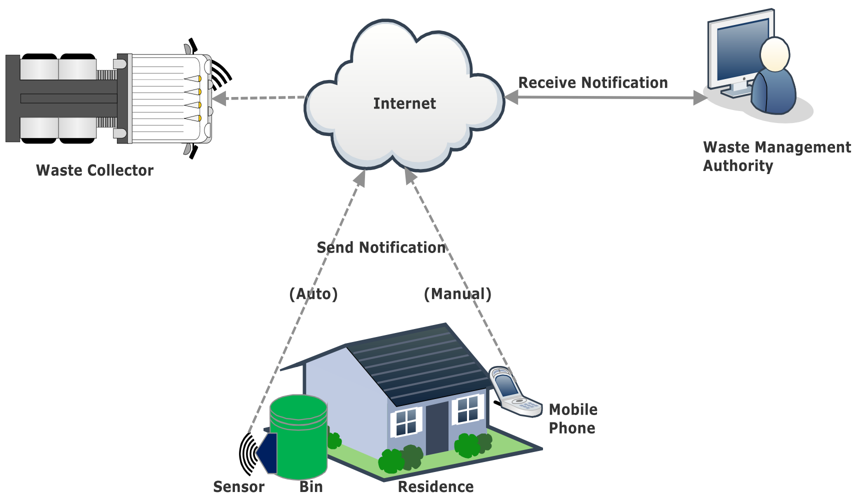


Figure 2-8 – General smart-waste collection system architecture. Source: adapted from [49]

This allows for a clear view of the strengths and weaknesses of the different technologies used in the projects. GSM and the bandwidth lag, Wi-Fi and its short range, ZigBee and its vulnerability to attacks and RFID with its short range and its vulnerability to unauthorized access are the comparisons made between the different technologies. More, it identifies LoRa as a technology that can benefit these projects' performance and longevity. The key strengths are coverage, less power consumption, and better reliability.

## 2.5 Lessons learned

A few conclusions can be taken from the previous sections. Based on the mentioned examples, noise pollution is a major concern and can be linked to several health problems. It is also the least measured parameter in our cities today. The highly dynamic nature causes it to vary in time and in space. This creates the follwing requirement: for a system to capture noise in a city, several of these sound pressure meters must be deployed. With the information from this array of sensors a noise map of the city in near real-time can be achieved. It creates the perfect opportunity to develop a system that can leverage the savings in public light monitoring with noise pollution monitoring. As it is modular, this system will allow to monitor other parameters and even to control other systems, such as public space irrigation for instance. This will lead to further savings and keep the return of investment short.

It is possible to determine that the communication technology that better suites the proposed system is LoRa. Using this technology with TTN allows a fast and cooperative IoT city coverage deployment. Using more than one gateway for the same area allows redundancy and consequently, improves the reliability of the network. The bandwidth limitations do not allow for audio or video samples to be sent. The same with photos. This allows privacy by design: only the values of the SPL can be sent. The citizens can be assured that no conversations can be recorded and transmitted.

On the side of cost concerns, LoRa is also the most interesting option, considering the amount and the dynamic of the transmitted data. It also allows to upgrade the nodes' firmware, which assures the longevity and adaptability of the solution. The use of encrypted communication, securing the information exchanged between the node and the application is yet another benefit of the LoRa technology.

# 3 Proposed architectural solution

Based on the discussion of the previous chapters, experiments were conducted in order to assess how a low cost IoT noise metering device could be created. The basic value that is important to evaluate is the Sound Pressure Level (SPL). As mentioned before, for capturing the sound pressure level an acoustic sensor or microphone is required. Also, for controlling and managing the information (input and output) a controller is needed. Finally, a radio communication cannot exist without some radio module. This defines the basic elements of the system node as represented in Figure 3-1. The controller board receives data from the microphone and sends it to the LoRaWAN radio module. This radio module can also receive data and send it to the controller. This bi-directional communication allows for configuration instructions to be sent to the device and allows it to function as an actuator and further enhance the node's functionality. This node can be equipped with a switch (actuator) allowing it to control the lamps or/and with a temperature sensor allowing it to detect a possible malfunction for example. This device was designed taking into account this modularity.


Figure 3-1 - System architecture, including the sensor node

By using a modular architecture, different devices can be deployed with different configurations depending on the purpose. For example, some nodes can have the sound sensor and a switch, and some can have a switch and a light sensor. By using this approach, a set of different environmental parameters can be collected thus contributing to a broader sense of the city's condition.

As detailed in Figure 3-1, the proposed modular device contains a controller, a communication module and one or more sensors and/or actuators. The connection between the controller and the sensors and/or actuators is accomplished by means of standard communication protocols further enhancing its adaptability. A node can be deployed today, with a set of features that can be upgraded in the future without requiring the total replacement of the node. As such, a node can have a variety of functions resulting in a highly modular and customizable module.

As for the roles of each part of the node it can be divided as follows. The controller is responsible for running the code, collecting data from sensors and send commands to the actuators. This controller functionality can be easily changed by software, either by changing the parameters or the code itself.

The sensors can collect data from the system, to monitor its health, such as temperature, power consumption and light intensity, amongst others. This data allows for preventive or corrective actions without requiring a technician to be present and troubleshoot a fault or waiting for a citizen to complain about a problem with the public illumination.

The actuators can produce actions within the luminaire or even outside of it. Typical examples include the obvious power on and off of the light as well as dimming, warnings or even an outside device, such as a gate for instance.

Finally, the communication module allows for the exchange of information between the nodes and the central system. The main focus of this work is the sound monitoring but is also applicable to other sensors. In fact, monitoring other parameters follows exactly the same principle, allowing for a system that is highly flexible.

## 3.1 Ambient noise monitoring principles

Based in the aforementioned works, and on the lessons learned, a proposal for a system for monitoring ambient noise can be designed with less uncertainty. In the aforementioned works, many solutions for sensors implementation are mentioned. These IoT devices are located in the luminaires, as implemented in San Diego, or deployed individually as in SONYC project implemented in New York, or in Calgary.

Fot the noise monitoring use case the location of the devices in the luminaires is the best option due to the following reasons:

- Provides constant power supply;
- Located in areas where noise pollution is most significant (due to traffic noise);
- High position where reception is more effective;
- This high position also allows for the capture of environment noise and not a localized sound.

This last one is important as the sound capture devices (microphones) are very sensitive to the distance from the sound source. This is due to the sound propagation properties. Figure 3-2 gives a graphical representation of this phenomenon. As the distance between the source (man) and sensor (microphone) increases, the sound waves intensity decreases. To take this phenomenon into account, the data from different sound sensors in an area will be combined in order increase precision.

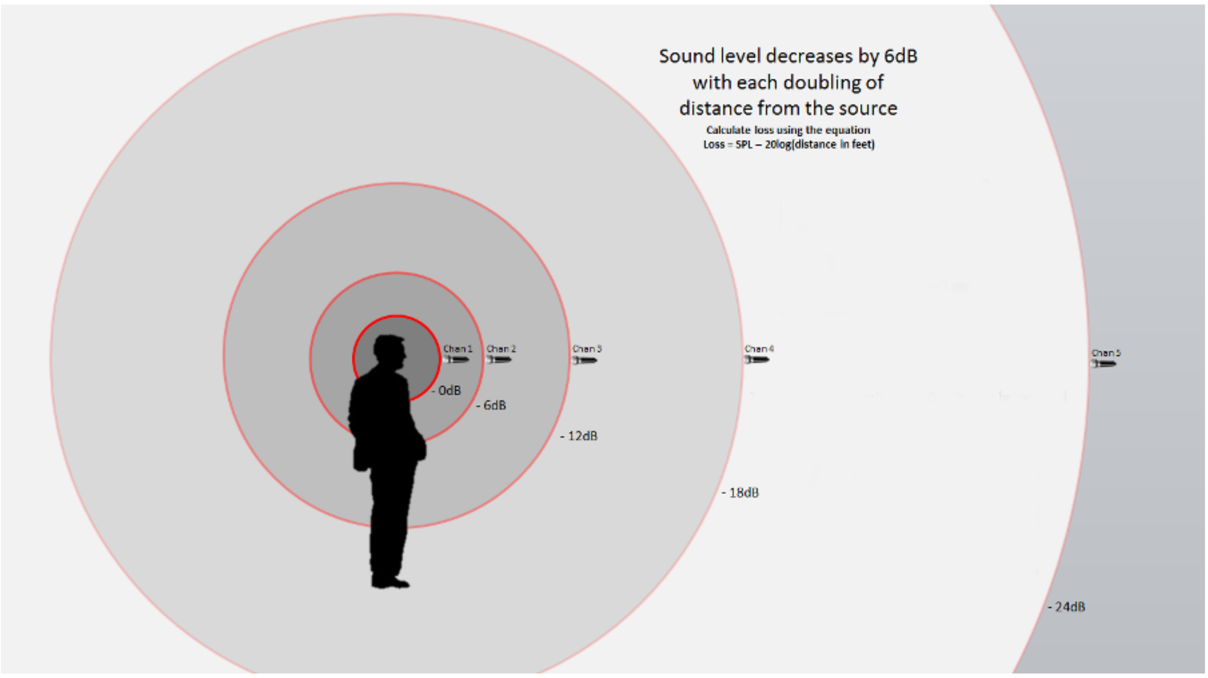


Figure 3-2 - Sound level variation from distance to source. Source: adapted from [64]

Typical implementation of noise monitoring in cities uses the NP ISO 1996:2019 as a guideline [65][10]. This document describes the methods for characterizing the environmental noise, by direct measurement and by means of extrapolation of results using calculus. This system is of particular relevance regarding the long-term monitoring and the need to consider the environmental conditions at the time of recording the noise values. As the system can record several environmental data sources at the same time, the values collected can be correlated to the environmental conditions with no additional work. This data collection also allows for future proofing the system, as the data is available to be processed by different methods any time in the future. This additional data can be collected by taking advantage of the modularity of the proposed system using one or more modules.

## 3.2 Sensors for ambient sound measurement

As already discussed, various sensors can be added to each module. The focus is now in one key aspect of environmental pollution: noise pollution. This is one of the more challenging environmental parameters to measure due to its large variation in time and space. In fact, a regulation compliant SPL can only be present in a limited area, and for a limited amount of time while keeping the cost at a reasonable level. Using this traditional methodology, it is improbable to have such a device in any place when a violation occurs. As such, in the case of a complaint, an operator with a certified SPL must be deployed in the area and start monitoring. As it is easy to understand, this is not always possible. Furthermore, the transgressor may stop its action by sighting the operator adding another difficulty in this methodology.

In order to develop a system that can monitor the ambient sound, a basic requirement is a sensor that can be used to measure the sound pressure level. As many of these devices will be needed in the city in order to create a visual map of the real time noise levels, the price should be kept as a low as possible.

The work development presented in this document started with the search for the appropriate sensor. For this purpose, different types of microphones were considered as well as their technology.

### 3.2.1 Microphone transducer type

Microphones are simple devices that convert sound to some type of signal, usually electric. The transducer used can be one of several types. The most common types are condenser, more specifically, electret condenser, dynamic and MEMS (MicroElectrical-Mechanical System). Figure 3-3 illustrates a typical electret microphone capsule section, with its several components.

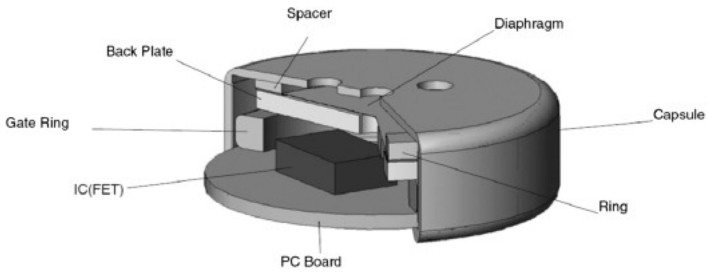


Figure 3-3 – Section of a typical electret condenser microphone. Source: [66]

According to J. Kuehn [67], a condenser microphone is a pressure microphone, which means that the electrical output is directly proportional to the sound pressure acting on the diaphragm. This feature is of particular interest for the purpose of measuring the sound pressure level. Also, the electret condenser type is one of the most common due to the ease of manufacture, low cost and good performance. Prior to the proliferation of MEMS microphones, it was the most common type of microphone [68]. These microphones contain a condenser capsule whose capacitance changes with the vibrations of the diaphragm. Unlike the conventional condenser microphone, where a power source is required for the capacitator plate voltage, a permanently charged electret material is used. They do require power for a preamplifier that is usually contained in the module.

The dynamic microphone, according to Boré, P. and Peus, S. [69], the dynamic type of microphones are velocity transducers and the mode of operation is based on the laws of induction. This means that, contrary to the condenser microphones, the energy generated is proportional to velocity of the moving conductor. Two subtypes can be identified: moving coil and ribbon microphones. The dynamic moving coil type is the most common dynamic microphone by far and the easiest to understand as it is basically an inverter loudspeaker. As represented in Figure 3-4, they contain a membrane to which a coil is glued. This coil then surrounds or/and is surrounded by a strong magnet. As the sound waves hit the membrane, the coil moves relative to the static magnet and a current is induced, thus converting the sound waves into electricity. In the case of the ribbon microphones, the diaphragm is the electrical conductor (usually a very thin aluminium foil). As they produce very small changes in voltage, they require a step-up transformer. In Figure 3-5 a schematic of this kind of microphone is shown. The thin aluminium foil indicated by (A) is connected directly to two conductors which, in turn, are connected to the transformer.

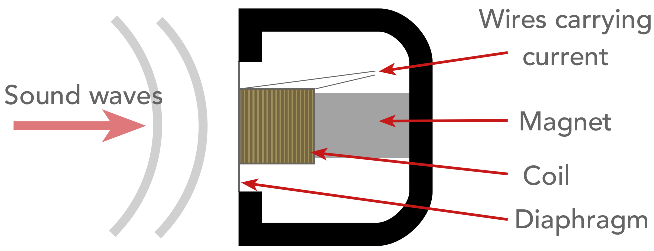


Figure 3-4 – Section of a typical moving coil condenser microphone. Source: [70]

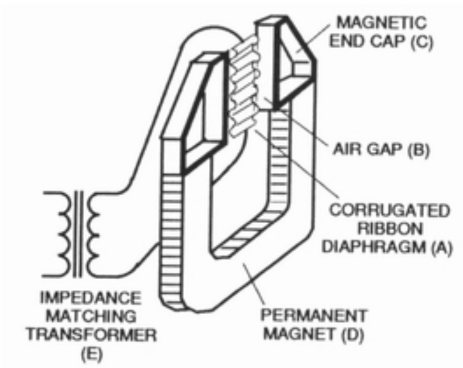


Figure 3-5 – Schematic of a ribbon dynamic microphone. Source: [71]

The MEMS type is a very small device, constructed using MEMS technology. By using this technology, the diaphragm can be directly etched into the silicon wafer. The very small package (usually smaller than 4 by 3 mm) contains not only the microphone but also the application-specific integrated circuit (ASIC) for analogue or digital output [72]. Figure 3-6 represents a cross-section of two variants of this kind of device: on top, a top port MEMS microphone and on the bottom, a back-port MEMS microphone.

### 3.2.2 Microphone interface types

There are several types of microphones and interfaces that can be used. From the older analogue to the new digital interface [72].

The analogue interface consists of a variation in alternating current directly produced by a moving element inside the microphone that moves in sync with the sound waves. This current is measured in millivolts (mV) or decibels relative to voltage (dBV, dBv or dBu[1]). They can have a built-in amplifier order to produce higher values of current.

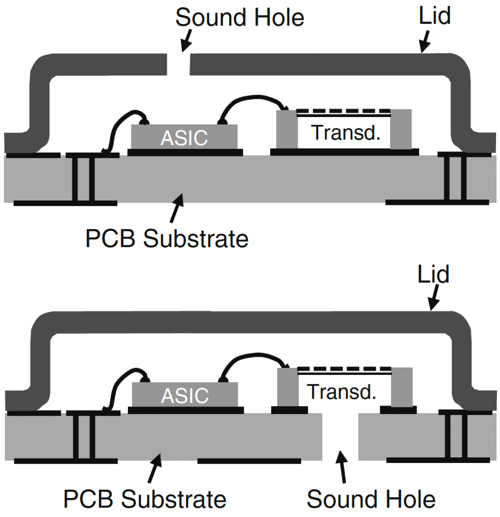


Figure 3-6 – Section of a typical MEMS microphone device. Source: [74]

The digital interface consists of a stream of bits, a digital representation of the analogue wave. This conversion can be achieved using different standards. A short list of these standards should contain mentions to Inter-IC Sound ($I^2S$), Pulse Code Modulation (PCM) and Pulse Density Modulation (PDM) to name three of the most common.

A conversion from the analogue signal to a digital stream of bits is always required. It can take place wither in the microphone, the controller, or somewhere in between.

### 3.2.3 Microphone amplifier

In the case of a microphone with an analogue interface, an amplifier is required in order to produce values of current change that can be detected by the controller ADC. The value of the amplification must be carefully chosen in order to detect the low energy sound waves (for quiet scenarios) and not saturate or "clip" the value for high energy sound waves (very loud conditions).

Some microphone modules come with an automatic gain [75], allowing to amplify more in quiet scenarios and less in very loud conditions. These microphones are not suitable for this project since the value of the amplification varies, changing the result. The amplification must be predictable and, better yet, if it is possible to adjust it to an optimal value for the application.

[1] dBV – decibels or dB relative to 1 volt, dBv or dBu – decibels relative to 1mW with 600 Ω (0.775 volt) [73]

## 3.3 Communication

As mentioned previously in Section 2.2.2, the LoRaWAN architecture is composed of nodes (also known as end devices), such as sensors and/or actuators, gateways, network servers and applications. The nodes are usually low-power devices, and can be classified as one of three types, depending on the required functionality:

- Class A devices – default and required functionality for all devices. It supports bi-directional communication with a gateway. It can send (uplink) information to the network at any time and can receive information (downlink) from the network at predefined times, specifically 1 and 2 seconds after an uplink transmission. If no information is received in one of these two windows the next opportunity will be after the next uplink transmission;
- Class B devices – these devices extend the class A devices' functionality by defining receive windows not linked to the uplink communications. The synchronization between devices and the network is accomplished through time-synchronized beacons transmitted by the gateways;
- Class C devices – these devices extend class A devices' functionality by keeping the receive window always open except when they are transmitting. This makes the device always available to receive information but requires much more energy than class A devices as the radio must be always powered on.

All devices, namely nodes and gateways, must be registered in the network in order to send and receive data. In order to accomplish this, the TTN uses a set of identifiers in order to identify the device and fort the activation :

- *DevEUI* – a unique 64-bit end-device identifier, assigned to the device by the radio chip manufacturer – the device identification;
- *DevAddr* – a non-unique 32-bit device address, assigned by the TTN to the device during the activation process. Seven bits are locked by the LoRa Alliance to the TTN. As this address is not unique, the device is found using the cryptographic signature (Message Integrity Code – MIC);
- *AppEUI* – a unique 64-bit application identifier, assigned by the TTN's account server. The address block assigned to the TTN starts with 70B3D57ED, meaning that all the applications registered in the network start with this identifier;
- *GatewayEUI* – a unique 64-bit gateway identifier, similar to the one used for end devices. The gateways can also be set with an MQTT-based forwarder requiring only a user set *GatewayID*.

The activation process required to define the DevAddr can be accomplished by using one of two methods:

- OTAA – Over-the-Air Activation – as the default, preferred and most secured method. This requires a negotiation between the device and the network to dynamically assign a device address and session keys;
- ABP – Activation By Personalization – this method requires the information regarding the keys and device address to be hardcoded in the device.

The first method is more secure and the second simpler as it skips the negotiation phase with the network it is also faster. However, as it requires the keys to be hardcoded in the device, it means that the code for each device must be different. As such, this method brings serious problems in production. It also creates security problems, as the information that secures the communication is available from the device. Another consequence is related to the frame counters, which will be discussed further next.

The data rate available for communication in TTN is derived from LoRaWAN. It depends on two elements: spreading factor and bandwidth. As the main focus of this type of communications is the low power transmission at long distances, the payload should be kept to a minimum. The longer the radio stays powered, the higher will be the power consumption. A graphical representation of this relation is presented in Figure 3-7. The concentric rings represent the range, the yellow line the power consumption and the green line represents the bitrate. The higher the bit rate, the shorter the time that is needed to send the message.

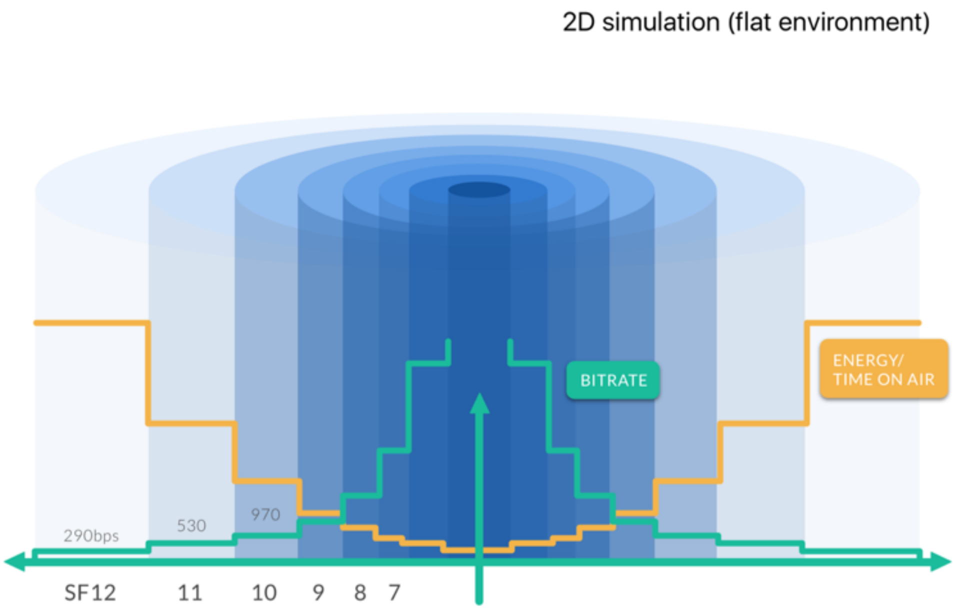


Figure 3-7 - Relationship between spreading factor and airtime for LoRa modulation. Source [77]

In order to keep the balance between power consumption and range, there are several options. The bandwidth can be adjusted to 125 kHz, 250 kHz or 500 kHz depending on the region and frequency plan. The distribution of the available frequencies is shown graphically in Figure 3-8. For most of Europe, the frequency used is 868 MHz. This is a consequence of the legislation followed by the different countries. The TTN supplies a simple way to calculate the airtime for the messages, based on four parameters: payload in bytes, spreading factor, region and bandwidth [76]. Table 3-1 presents the different combinations available in Europe. The configuration represents spreading factor / bandwidth.

To keep the payload to a minimum, the TTN proposes in the documentation some simple methods of encoding the information. Instead of sending data in a JSON format, it should be encoded in the most efficient method possible. For example, to send the status of a battery, it is often enough to know if it is in different intervals, like less then 100%, 75%, 50% or 25%. To encode this information only two bits are required, as 2 bits can represent 4 different status.

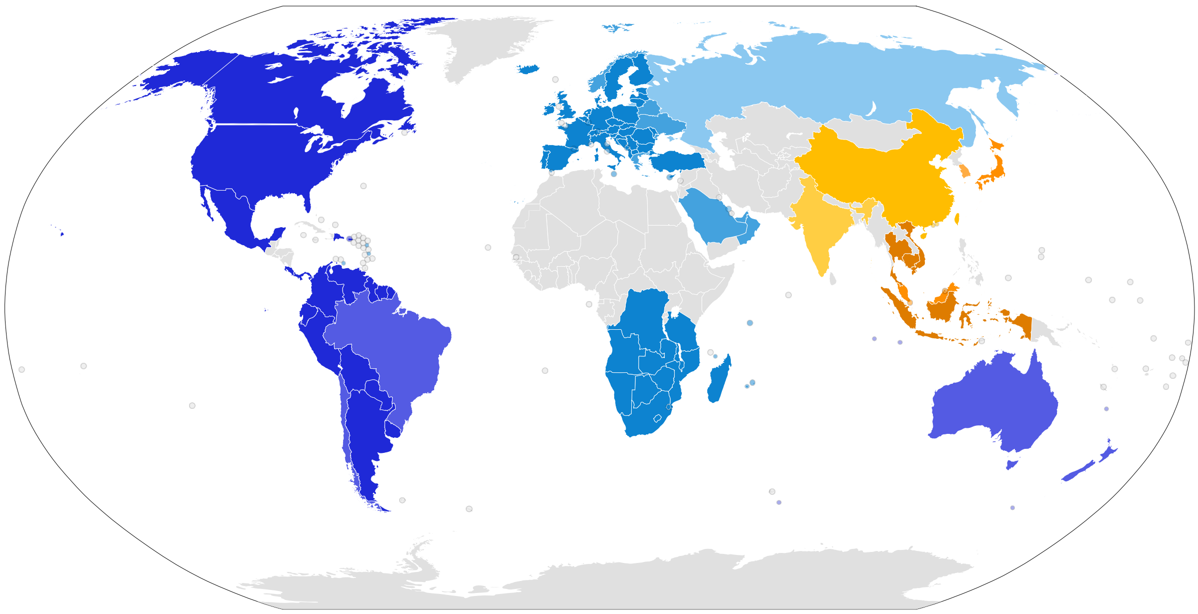

Figure 3-8 - Frequency plans by Country. Source [78]

Table 3-1 - Configurations for the spreading factor in Europe. Source [35]

| Configuration | Bitrate (bits/sec) | Max payload size (bytes) |
|---|---|---|
| SF12 / 125 kHz | 250 | 51 |
| SF11 / 125 kHz | 440 | 51 |
| SF10 / 125 kHz | 980 | 51 |
| SF9 / 125 kHz | 1760 | 115 |
| SF8 / 125 kHz | 3125 | 222 |
| SF7 / 125 kHz | 5470 | 222 |
| SF7 / 250 kHz | 11000 | 222 |

In order to optimize the power consumption, a mechanism known as LoRaWAN Adaptive Data Rate (ADR) should be used whenever an end device is static with suitably radio conditions. This is an option of the end devices and not the network or application. Since the messages contain a frame counter, signal-to-noise ratio (SNR) and the number of gateways that received the message, this information is used to calculate if there is enough margin to increase the data rate or to lower the transmission power.

Another important factor is duty cycle. This corresponds to the percentage of time that a device transmits data. The regulations impose a maximum duty cycle for fair use. This value is often set at 1% but it depends on local regulation. For Europe, following the ESTI EN300.220 standard [79], defines sub-bands and maximum duty cycles:

- g (863.0 – 868.0 MHz): 1%
- g1 (868.0 – 868.6 MHz): 1%
- g2 (868.7 – 869.2 MHz): 0.1%
- g3 (869.4 – 869.65 MHz): 10%
- g4 (869.7 – 870.0 MHz): 1%

For join frequencies the LoRaWAN specification also dictates duty cycles. The most common value for maximum duty cycle is 1%. The TTN, in the public community network there is even another limit, named Fair Access Policy. This limits the uplink airtime to 30 seconds and the downlink messages to 10 in a 24-hour period. This limitation is not imposed in a private network. It is important to note that the duty cycle limits are imposed for both end devices and gateways. Some radio modules implement these limits, sending an error message indicating when the limits are exceeded. This is one reason why the payload must be kept to a minimum and the spreading factor as fast as possible.

Finally, it is important to mention security, a key element in LoRaWAN. The 1.0 specification states 3 security keys: NwkSKey, AppSKey and AppKey. All implement AES-128 algorithm with 128-bit length:

- *NwkSKey* – Network Session Key – is generated as part of a join request. This key is shared with the network and is used to communicate from the end device to the network server. It is also used as part of the DevAddr verification to match a device to a message by using its MIC check. Furthermore, this key allows to validate the integrity of a message;
- *AppSKey* – Application Session Key – is also generated as part of a join request. It is kept private and used to encrypt and decrypt the payload. This encryption takes place at the end node and the decryption takes place at the application server of the TTN. This guaranties that only the owner can read the contents of the message;

- *AppKey* – Application Key – is shared between the end device and the application. It is used in the OTAA process to create the NwkSKey and AppSkey (that vary in each activation). It can be the same for a number of devices or customised per device.

If the ABP method is used, the first two keys are set in the device, and will remain the same until changed.

Another protection is implemented by the use of frame counters. These protect against the re-transmission of a message by numbering each uplink and downlink. Every time a message is received with a frame counter with a number inferior to one already received, the message is simply discarded. This might cause problems with devices using the ABP method as the frame counters are reset at each activation of OTAA but are kept in ABP. The network has no way of knowing if a device was reset so a manual reset on the network side is required.

# 4 Development

For the purpose of building the module, several microphones breakouts where considered and tested. The goal was to determine the best solution for sound monitoring considering performance and price.

## 4.1 Tested microphones

Table 4-1 details the compared microphones and their technical specifications (with n/a representing not available data). Further details are presented in the next sections.

Table 4-1 - Technical details of the different microphones tested

| Designation | Frequency | Sensitivity | Amplifier | Interface | Documentation | Price (inc.VAT) |
|---|---|---|---|---|---|---|
| Electret microphone with LM393 | n/a | n/a | no | analogue | n/a | 0,44€ |
| Electret microphone with LM386 | n/a | n/a | no | analogue | n/a | 0,79€ |
| Makeblock Me Sound Sensor | 16Hz-20KHz | 50-54dB | yes | analogue | yes | 5,99€ |
| Adafruit electret microphone with adjustable gain | 20Hz-20KHz | 42-48dB | yes | analogue | yes | 7,76€ |
| Adafruit $I^2S$ MEMS Microphone Breakout | 50Hz-15KHz | 94dB | yes | digital $I^2S$ | yes | 7,76€ |
| Adafruit Silicon MEMS Microphone Breakout | 100Hz-10KHz | 94dB | yes | analogue | yes | 5,54€ |

### 4.1.1 Electret microphone with LM393

Also known as “Keyes”, “KY-038” or “HXJ-17”, this microphone can be easily found in many online stores and electronics kits. Albeit its popularity, no rigorous documentation about it could be located. The module contains an electret microphone capsule, a potentiometer and an LM393 chip. Several different capsules, with different sizes, can be found in this reference of microphone module, all without any reference to the performance characteristics. The capsule sizes of these example vary from 6mm for the smaller capsule to 9.6mm for the larger ones.

The documentation of the chip identifies it as a dual comparator [80]. Some documentation can be found indicating that this module has an amplifier [81]. Analysis of this module shows that no amplifier is present. The digital output sensitivity can be adjusted with the potentiometer. In some versions of this module, like the one with a black PCB in Figure 4-1, there is no analogue output present. Figure 4-1 presents three of the tested microphones that fall into this designation. As shown, all of these devices include a potentiometer, a LM393 chip, an electret microphone capsule, and some passive components. The red PCB versions of the module include an analogue output absent in the black version. A key advantage of these devices is that they can be found for as low as 0.44€ [82].

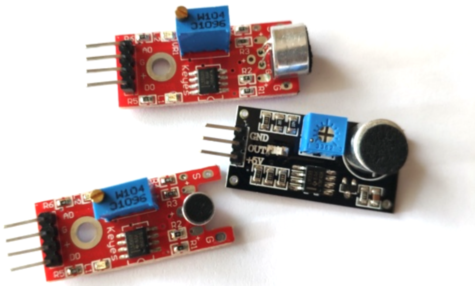

Figure 4-1 – Several versions of the electret microphone with LM393 module

### 4.1.2 Electret microphone with LM386

Sometimes referred as “Keyes brick”, this module is similar to the previously mentioned, using an LM386 chip instead of the LM393. In addition to this chip, it also contains an electret microphone capsule and a potentiometer. The documentation of LM386 chip identifies it as a low voltage audio power amplifier [83]. The amplification can be adjusted by using the potentiometer. As presented in Figure 4-2, the output pins of the module are not labelled but were found to be: ground at the potentiometer side, followed by VCC in the centre and output on the opposite side. These devices can be found for 0.79€ [84].

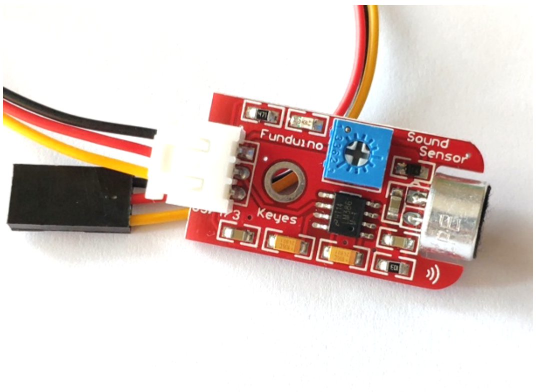


Figure 4-2 – Electret microphone with LM386

### 4.1.3 Makeblock Me Sound Sensor

This breakout board is designed from scratch to be used to detect the ambient sound intensity. This device has a good documentation [85] and a schematic is provided as shown in Figure 4-3. It contains an electret microphone capsule and a LM2904 low-power amplifier. The microphone has a sensitivity of 50-54dB at 1KHz, with an impedance of 2.2KΩ. It has an operating frequency of 16Hz to 20KHz and a SNR of 54dB. It features a single analogue output port. Figure 4-4 presents this device, clearly showing the microphone capsule, the LM2904 chip and several passive components. Also, in the RJ11 socket is present a single analogue output. This board is available for 5.99€ [86].

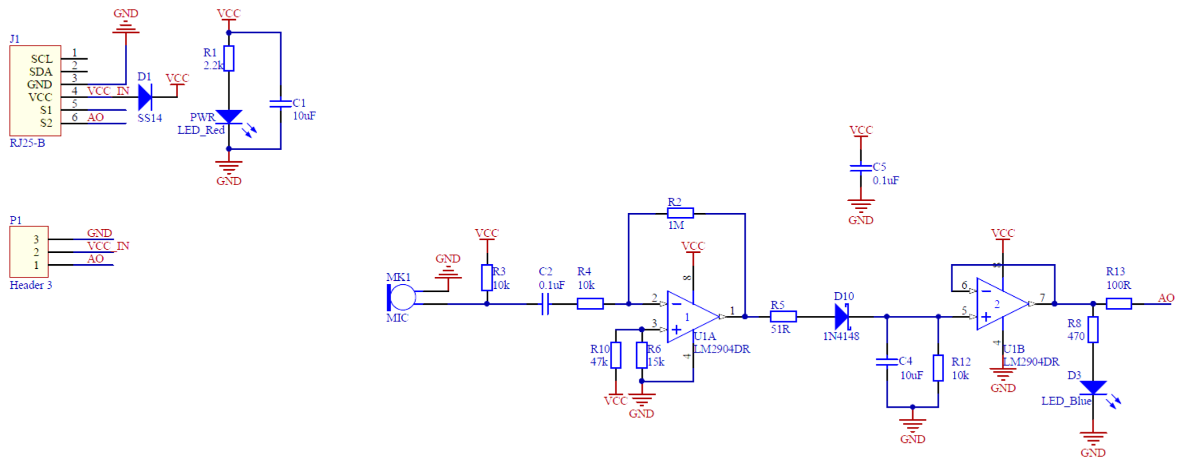


Figure 4-3 - Makerblock Me Sound Sensor schematic. Source: [85]

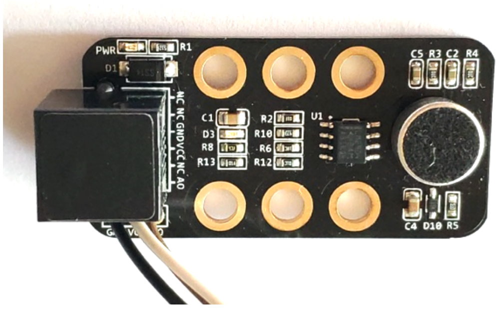


Figure 4-4 – Makeblock Me Sound Sensor

## 4.1.4 Adafruit electret microphone with adjustable gain

As the first two modules, this also contains an electret microphone capsule and a potentiometer [87]. However, it has a far better documentation supporting it, including schematics, documentation of the main parts used and some examples [87]. The amplification is achieved using a Maxim MAX4466 [88] op-amp. The capsule has an operating frequency between 20 Hz and 20 KHz, with a sensitivity of -44dB (±2dB) at a frequency of 1KHz, 1Pa [89]. The amplifier features power supply noise rejection and with the use of the trimmer pot the gain can be set from 25x to 125x. As it can be seen in Figure 4-5, the output pins are clearly labelled. This module is available for 7.76€ [90].

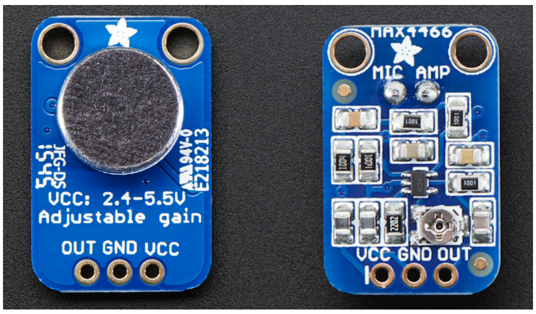


Figure 4-5 – Adafruit electret microphone with adjustable gain. Source [87]

### 4.1.5 Adafruit I²S MEMS Microphone Breakout

This is the first module to use a digital interface. It is supported by a helpful set of documentation from the manufacturer. It contains a single active component, a SPH0645LM4H MEMS type microphone that is sensible to frequencies between 50 Hz and 15 KHz and a sensitivity of 94dB at a frequency of 1KHz. Inside the capsule there is a sound sensor, a serial ADC and an interface to convert the signal to 24-bit $I^2S$ format [91]. As this module outputs a digital signal according to the $I^2S$ specification [92] it connects to the digital side of the controller board. Instead of an analogue output, this module has 3 digital pins: clock, signal and left/right clock. As shown in Figure 4-6, the MEMS microphone is a back-port variant as the sound is captured through a hole in the board. It can also be observed that no active components are present in the board, since all sound capture and conversion takes place inside the microphone capsule. It is available for 7.78€ [93].


Figure 4-6 – Adafruit $I^2S$ MEMS Microphone Breakout. Source [94]

### 4.1.6 Adafruit Silicon MEMS Microphone Breakout

Using a similar microphone technology as the previous module, it features an analogue interface. It is also backed up by a comprehensive set of documentation. A SPW2430 MEMS type microphone is the main element of this module [95]. This microphone package includes the MEMS microphone capsule, a low noise input buffer and an output amplifier. This microphone is sensitive to frequencies between 100Hz and 10KHz, has a sensitivity of 94dB at 1KHz and an output impedance of 450Ω [96]. Inside the capsule there is an acoustic sensor, a low noise input buffer, and an output amplifier. As shown in Figure 4-7, this is a top port microphone. Aside from this package, also present in this module, is a MIC5225 low dropout regulator, configured for 3.3V, which allows for the module to be powered by voltages between 3.3V and 5V [97]. This module can produce a peak-to-peak value as high as 1Vpp. The output signal is obtained from the DC pin. The AC pin has a capacitor in series for equipment that may require AC coupled audio. This module is available for 5.54€ [98].

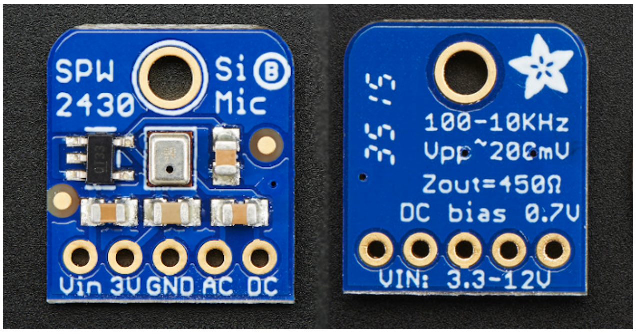


Figure 4-7 – Adafruit Silicon MEMS Microphone Breakout. Source [95]

## 4.2 Testbed setup

Since the considered microphones in Section 4.1 are very low-cost devices, some of the technical data is not present or is inaccurate. Consequently, they must be tested in order to access their real performance. For this purpose, a testbed was prepared, using an off-the-shelf microcontroller.

The prepared testbed was based on a Seeeduino LoRaWAN w/GPS [99]. It is an Arduino compatible development board with built-in LoRaWAN and GPS capabilities. This development board uses a 32 bit, low-power, ARM Cortex-M0+ ATSAMD21G18 microcontroller, running at 48MHz. It contains Arduino compatible pinouts, 4 grove connectors, external antenna connector, battery management circuitry, a GPS module and, for wireless communications, it uses a RHF76-052AM module, that contains a Semtech SX1276 LoRa transceiver. The microcontroller has support for $I^2S$ communications as well as $I^2C$, a requirement for testing the proposed microphone modules. It consumes very little power, just 9µA/MHz.

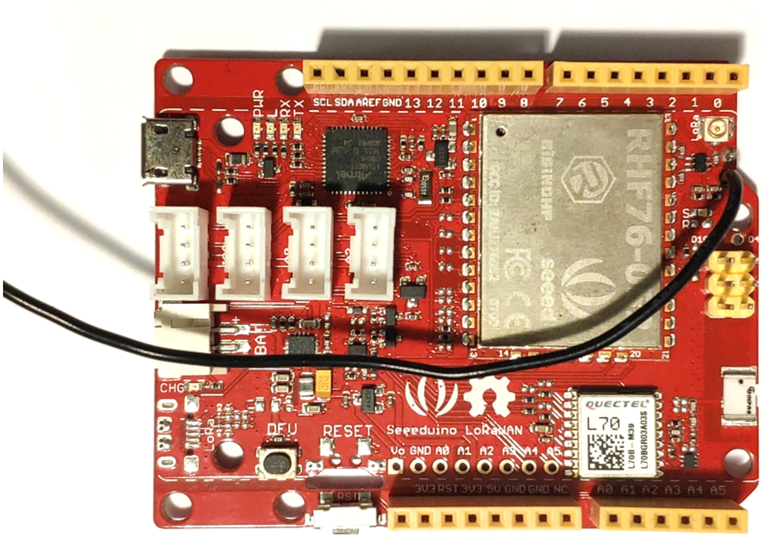


Figure 4-8 - Seeeduino LoRaWAN w/GPS

The Semtech SX1276 LoRa transceiver focuses on ultra-long range spread spectrum communication and high interference immunity with very low current consumption [100]. The manufacturer lists the following features:

- 137 MHz to 1020 MHz Long Range Low Power;
- 168dB maximum link budget;
- +20dBm - 100 mW constant RF output vs. V supply;
- +14dBm high efficiency PA;
- Programmable bit rate up to 300kbps;
- High sensitivity: down to -148dBm;
- Bullet-proof front end: IIP3 = -11dBm;
- Low RX current of 9.9mA, 200nA register retention;
- Fully integrated synthesizer with a resolution of 61Hz;
- FSK, GFSK, MSK, GMSK, LoRa and OOK modulation;
- Preamble detection;
- 127dB Dynamic Range RSSI;
- Automatic RF Sense and CAD with ultra-fast AFC;
- Packet engine up to 256 bytes with CRC.

The transceiver is capable of frequencies between 137 MHz and 1020 MHz, but it was set as 868 MHz, the frequency used in most countries in Europe, including Portugal (TTN does not use the EU433 frequency [30]).

The inclusion of a battery is also welcome in the final node design. Combining the low power consumption of the solution with a battery allows for an alarm to be sent to the system in the case of a power outage.

The tests were made with the Seeeduino LoRaWAN but they can be done with any other microcontroller board. One of the advantages of this approach is that it is compatible with a wide range of solutions. With minimal adaptations, it can be done in a The Things Uno [101], that already contains a LoRa module, any other compatible Arduino or clone by adding a LoRaWAN module, or even other low-cost microcontrollers such as an ESP32 [102] with a LoRa module, for instance. For use with $I^2S$ devices, it requires the controller to have an $I^2S$ interface. These boards are well suited to develop prototypes but for a final production device a customized board is required. From Figure 4-9 to Figure 4-12, alternatives options are presented.

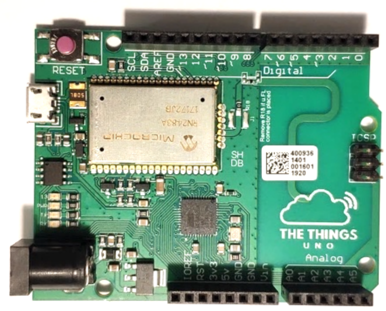


Figure 4-9 - The Things Uno board

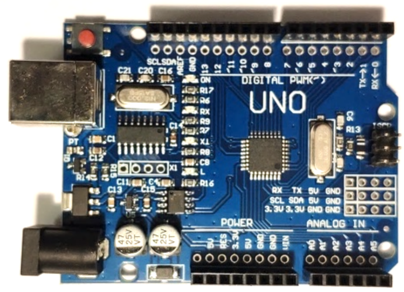


Figure 4-10 - Arduino UNO clone board

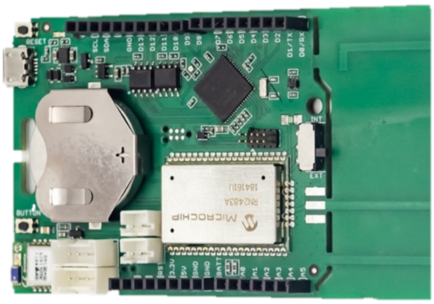

Figure 4-11 - SODAQ Explorer. Source [103]

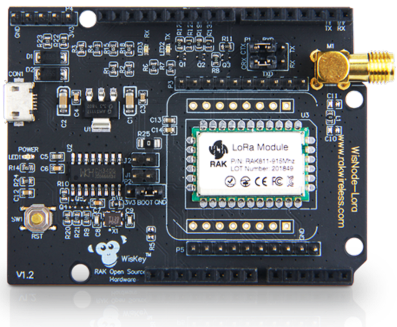


Figure 4-12 - RAK811. Source [104]

## 4.3 Microphone evaluation and comparison

Several microphones were tested by using the same controller board: the Seeed Studio Seeeduino LoRaWAN w/GPS. The purpose was to find what devices are better suited for detecting sound intensity while keeping costs as low as possible. The test setup consists of the previously mentioned board, a breadboard and the microphones. The microcontroller board was connected to the computer via a powered USB hub for consistent power supply.

A simple program was implemented to enable each analogue input to be read, stored, transmitted through the serial output, and evaluated. The algorithm is presented in Figure 4-13. The implemented code for the program is presented in the Annex C. The Arduino IDE [105] was used for programming the controller and also to provide a visual representation of the results based on its Serial Plotter option.

The tests were conducted in a quiet environment, with a low noise level. To evaluate the ambient noise level where this experiment took place, a simple Android app was used: Sound Meter by Info Stack [106]. Though not a true sound level meter, it allows to determine a reference value of ambient sound level. Using this tool, the value of the ambient noise floor was measured as 30dB. This value was found to be repeatable throughout the experiment.

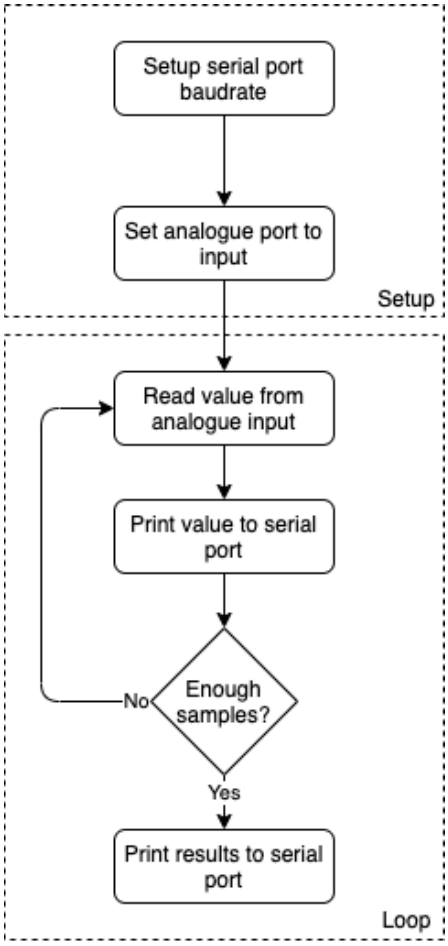


Figure 4-13 – Program diagram

For the sound source, a website was used: White Noise & Co. [107]. This is a simple way of creating white noise samples. White noise samples were chosen as they include all frequencies in the audible range and with equal intensity. By using white noise samples, it is possible to test the microphone's sensitivity to the complete audible range, instead of a single frequency. The sound was produced by a speaker system at the same distance from the microphones or microphone arrays under test and the Android device. Using this setup, repeatable conditions can be achieved.

The first devices tested were the least expensive ones: the electret microphone with LM393. Just as presented in Figure 4-1, this device is available with three variants and all were tested. As this module contains a potentiometer that changes the DC component of the output, it was set so that all devices output the same. For this purpose, the serial plotter of the Arduino IDE was used. It is a simple and real time way to see the graphical representation of the outputs an adjust the potentiometer so that all outputs are similar. An alternative is to use an oscilloscope and measure the output but as all comparisons used the same method it was determined that no further detail or precision is required.

Using this methodology, 10 000 samples were recorded for analysis, for each device, and for an environment set to 30dB and 60dB. The values captured represent the voltage levels as captured from the ADC. With this controller board, the values from the ADC vary from 0 (for 0V) to 1023 (for 3.3V).

The values captured are a digital representation of the sound wave. As mentioned before, for the purpose of determining the sound pressure level, only the variation from the average is of interest. The higher the variation, the louder the sound. The average from the measurements was taken and the values were normalized by using the following formula:

$$absolute\ value = |captured\ value - average| \quad (2)$$

These results are compared between devices in order to compare their performance.

### 4.3.1 Electret microphone with LM393

The three microphone modules presented in Figure 4-14 were connected to the Seeeduino LoRaWAN w/GPS development board using a standard breadboard. The connections were as presented in Figure 4-15 (Seeeduino to microphone module):

- GND to G ;
- 3V3 to + ;
- A2 to A0.

The analogue port A2 was used in order to avoid issues related to other circuitry inside the development board. Further details are available in Annex B.

Using the same methodology, the other two variants were connected. In the case of the module with the black PCB, the A2 input from the board was connected to the OUT port on the module.

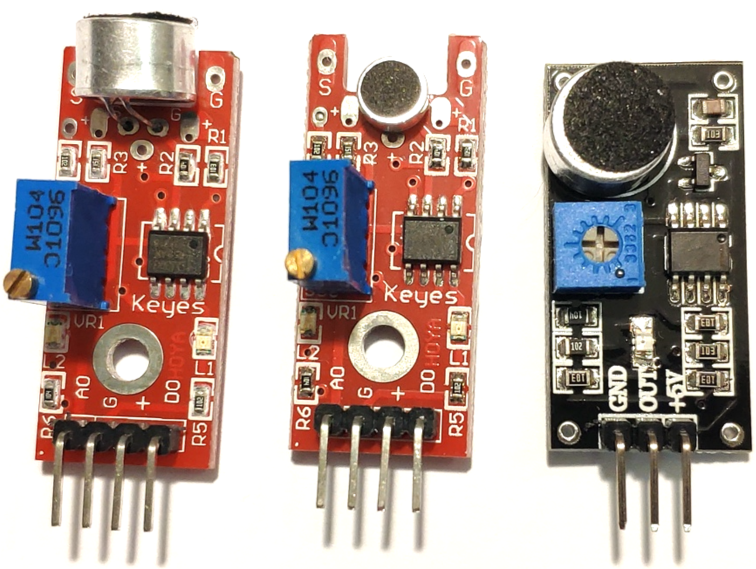

Figure 4-14 - Three variants of the Electret microphone with LM393

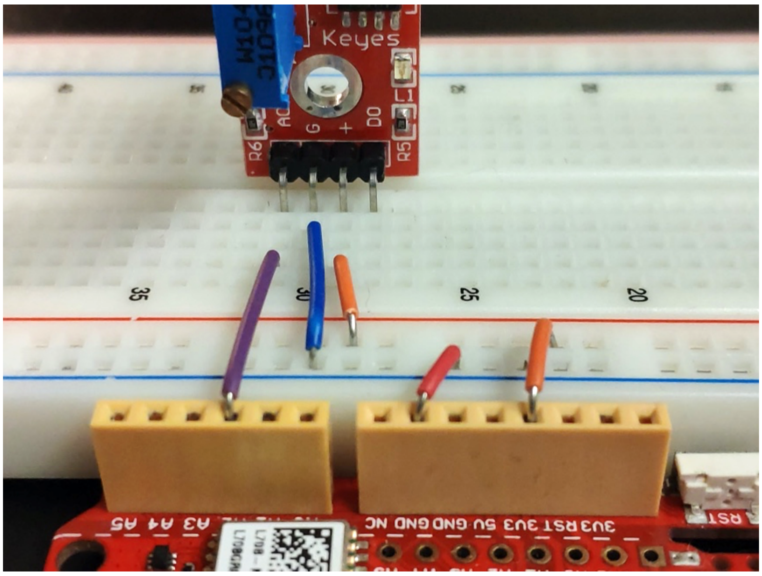


Figure 4-15 – Electret microphone with LM393 module connected to the Seeeduino LoRaWAN w/GPS on A2 port

Comparing these devices has shown that the device with the bigger capsule is capable of showing a higher change (peak-to-peak) in the values captured by the ADC. However, during experimentation, it was found that the potentiometer also changes the value of the analogue output, shifting the values without changing peak-to-peak representation of the sound (no amplification). This was determined by running the program, with an environment at 30dB and 60dB and using the following formula:

$$\Delta Input = Max\ Input - Min\ Input \quad (3)$$

$\Delta Input$ is the peak-to-peak value, $Max\ Input$ and $Min\ Input$ are, respectably, the maximum and minimum value reported by the ADC, during the 10 000 samples time. For confirmation, the developed program was run and the values for both devices were captured. In order to capture the values from the serial port, the following terminal command was used (MacOS):

```
screen -L /dev/[device] [baudrate]
```

`[device]` is the path of the USB port where the device is connected and `[baudrate]` is the baud rate set in the serial communication (optional). Each device was tested with two ambient sound levels (30dB and 60dB) and three potentiometer settings. The obtained results are presented in Table 4-2. This procedure allowed to confirm that no amplification is applied to the input as the difference in values is marginal. However, it changes the average value.

Another interesting value is obtained by using the following formula:

$$\sum |\Delta Input| = \sum_0^n \left| Input - \frac{\sum_0^n Input}{n} \right| \quad (4)$$

Where $n$ is the number of samples and $Input$ is the value reported by the ADC for the sample. This allows a deeper view into de sum of differences in the measurements.

Table 4-2 - Comparison of the different variants of the electret microphone with LM393

| 9.6mm capsule | Mininum setting | | Medium setting | | Max setting | |
|---|---|---|---|---|---|---|
| | 30 dB | 60dB | 30 dB | 60dB | 30 dB | 60dB |
| Max Input | 23 | 23 | 596 | 598 | 1023 | 1023 |
| Min Input | 19 | 19 | 592 | 591 | 1023 | 1023 |
| Δinput | 4 | 4 | 4 | 7 | 0 | 0 |
| Average Input | 20 | 20 | 593 | 594 | 1023 | 1023 |

| 6mm capsule | Mininum setting | | Medium setting | | Max setting | |
|---|---|---|---|---|---|---|
| | 30 dB | 60dB | 30 dB | 60dB | 30 dB | 60dB |
| Max Input | 20 | 20 | 602 | 608 | 1023 | 1023 |
| Min Input | 17 | 17 | 598 | 603 | 1023 | 1023 |
| Δinput | 3 | 3 | 4 | 5 | 0 | 0 |
| Average Input | 18 | 18 | 600 | 605 | 1023 | 1023 |

| Black PCB | Mininum setting | | Medium setting | | Max setting | |
|---|---|---|---|---|---|---|
| | 30 dB | 60dB | 30 dB | 60dB | 30 dB | 60dB |
| Max Input | 36 | 36 | 36 | 36 | 1023 | 1023 |
| Min Input | 33 | 33 | 33 | 33 | 1023 | 1023 |
| Δinput | 3 | 3 | 3 | 3 | 0 | 0 |
| Average Input | 34 | 34 | 34 | 34 | 1023 | 1023 |

Using only the data corresponding the medium setting, graphs were plotted. These are frequency graphs with the number of samples in the yy axis and interval considered in the xx axis. These were calculated by using the formula (2). Although two different microphones were tested and with two different environmental conditions, all graphs represent the same: all 10 000 samples are within 5 ADC values from the average, as such, only the graph in Figure 4-16 is presented.

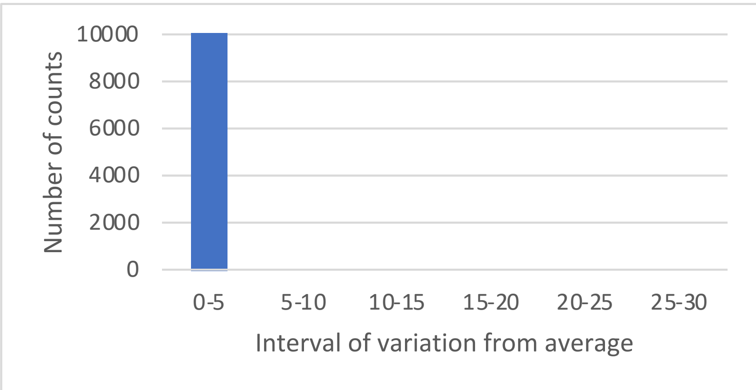


Figure 4-16 - Output from the electret microphones

Some variation is visible in the log files but is too small to show in theses graphs. However, by using the formula (4), the difference is more obvious:

Table 4-3 - ∑|ΔInput| comparison of the two capsules for 30dB and 60dB

| ∑\|ΔInput\| | Medium setting | | Variation |
|---|---|---|---|
| | 30 dB | 60dB | (%) |
| 9.6mm capsule | 5308 | 7086 | 33% |
| 6mm capsule | 3454 | 6506 | 88% |

The module with the 9.6mm capsule and black PCB has only a digital output. When tested the output is always low (33-36) until an ambient noise level causes it to change to high (1023). The level of the ambient noise that causes this change is adjustable with the potentiometer. As the variation of the output of this module is very low for the two tested environments, this module is considered unusable for the purpose.

### 4.3.2 Electret microphone with LM386

As the LM386 based sensor module has no documentation and no labels on the board, the chip was analysed. This is a low voltage audio power amplifier and from the specifications, it can be supplied with 4-12V or 5V-18V. The connection is presented in Figure 4-17. Connecting this module to the 5V rail in the controller board and checking its output shows no change in the output with any change in the environment noise.

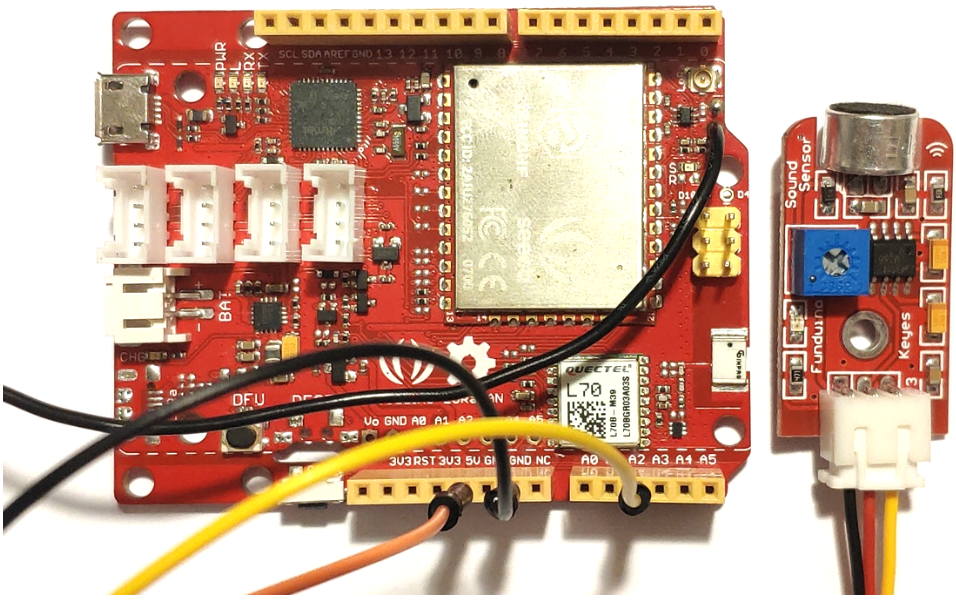

Figure 4-17 – Electret microphone with LM386 module connected to the Seeeduino LoRaWAN w/GPS on A2 port

This module presents a similar behaviour to the electret microphone with LM393 and black PCB. However, the output changes with the potentiometer setting without influence from the environment noise. The results are presented in Table 4-4. As the previous black variant of the electret microphone with LM393, since changes in the environment conditions do not change its output, this module is also unusable for the purpose.

Table 4-4 - Performance of the electret microphone with LM386

| 9.6mm capsule | Mininum setting | | Medium setting | | Max setting | |
|---|---|---|---|---|---|---|
| | 30 dB | 60dB | 30 dB | 60dB | 30 dB | 60dB |
| Max Input | 10 | 10 | 973 | 970 | 1023 | 1023 |
| Min Input | 0 | 0 | 937 | 936 | 993 | 993 |
| Δinput | 10 | 10 | 36 | 34 | 30 | 30 |
| Average Input | 5 | 5 | 952 | 952 | 1006 | 1005 |

### 4.3.3 Makeblock Me Sound Sensor

This breakout board has a much better documentation than the previous modules. This documentation states a 5V input, so the module was connected to the 5V rail of the Seeeduino as presented in Figure 4-18. The analogue A0 port of the sensor was connected to the A2 port on the controller, as the previous modules. This module has no adjustments, only the factory set configuration. After a run of measurements, it became obvious that this module has much more resolution as changing from 30dB to 60dB is very apparent in the results as shown in Table 4-5.

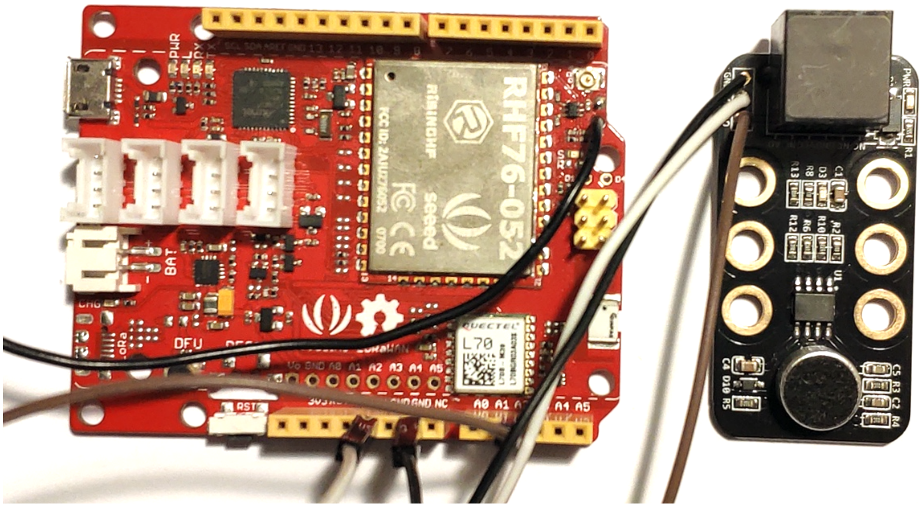


Figure 4-18 – Makeblock Me Sound Sensor breakout connected to the Seeeduino LoRaWAN w/GPS on A2 port

Table 4-5 - Performance of the Makeblock Me Sound Sensor

| Makeblock | 30 dB | 60dB |
|---|---|---|
| Max Input | 222 | 576 |
| Min Input | 214 | 308 |
| Δinput | 8 | 268 |
| Average Input | 216 | 407 |

As before, the following graphs were plotted with the bars representing the number of samples in the same interval from the average value, calculated from (2). Theses graphs are presented in Figure 4-19 and Figure 4-20 for 30dB and 60dB, respectively. As before, these are frequency graphs with the number of samples in the yy axis and interval considered in the xx axis. Contrary to the first modules, this one presents a clear variation when changing from 30dB to 60dB environment noise.

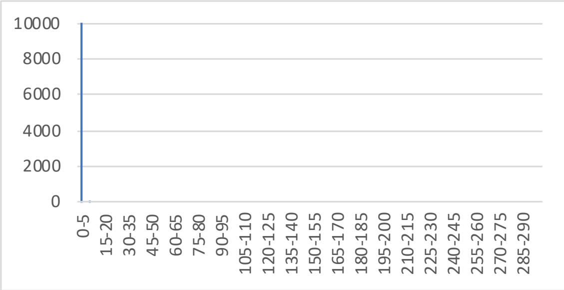


Figure 4-19 - Output from Makeblock module at 30dB

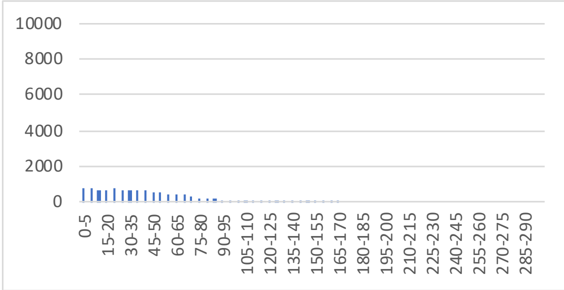


Figure 4-20 – Output from Makeblock module at 60dB

These results are much more useful than the previous modules. Furthermore, by the results of applying the formula (4) the output. This result is presented in Table 4-6.

Table 4-6 - ∑|ΔInput| comparison for 30dB and 60dB for Makeblock Me Sound Sensor

| ∑\|ΔInput\| | 30 dB | 60dB | Variation (%) |
|---|---|---|---|
| Makeblock | 7843 | 389046 | 4861% |

However, this module presents a behaviour that is not as intended: the output does not match to the sound pressure at the moment. This can be observed in the graph from the input at 60dB presented in Figure 4-21. The yy axis is the value reported by the ADC and in the xx axis the number of the sample.

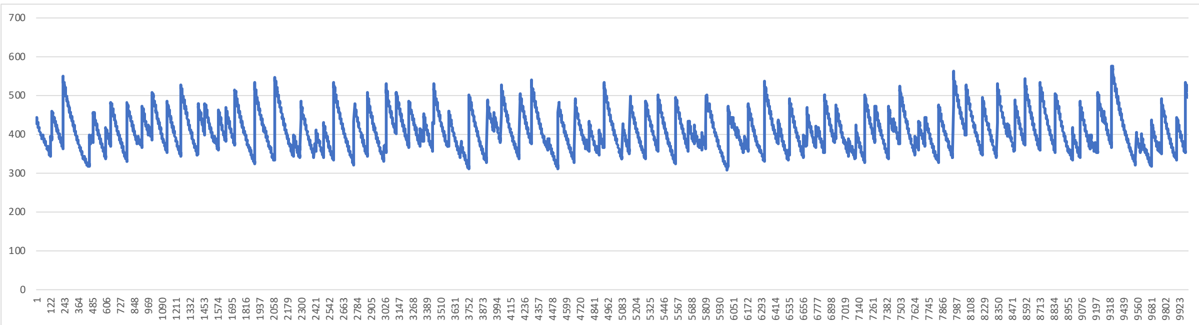


Figure 4-21 - Graph of the results obtained from the Makeblock Me Sound Sensor at 60dB

This behaviour is not intended as it introduces artifacts in the readings. These artifacts will cause deviation in the calculation of the value of SPL whatever the sample window considered. As the sample window should be able to be configured by the client of the system, this microphone is also not suitable for the purpose.

### 4.3.4 Adafruit electret microphone with adjustable gain (1063)

Such as the previous one, this module also has good documentation. The power source is indicated in the module: 2.4V to 5.5V so, this module was connected to the 3.3V rail from the Seeeduino. As before, the OUT pin from the microphone module was connected to the analogue A2 port of the controller board. The setup is presented in Figure 4-22. As this module is adjustable, it was tested for the same three scenarios: minimum, maximum and medium potentiometer settings.

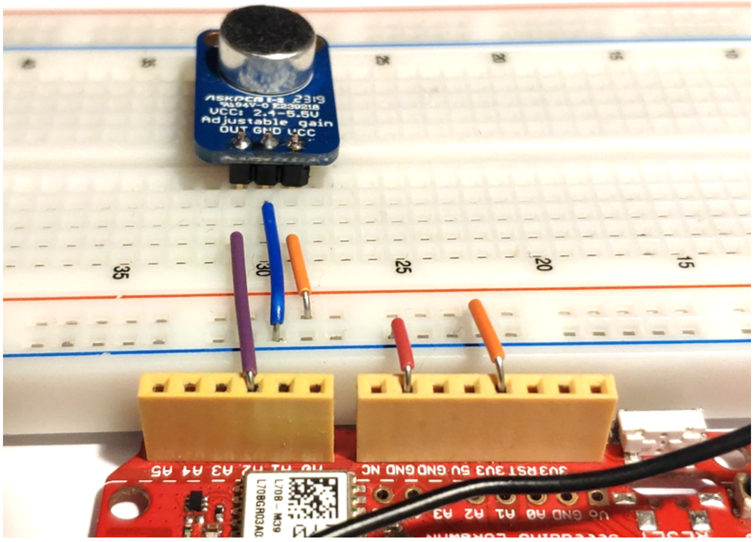


Figure 4-22 - Adafruit electret microphone with adjustable gain module connected to the Seeeduino LoRaWAN w/GPS on A2 port

Following the same methodology as before, the comparison graphs are plotted in Figure 4-23 and Figure 4-24. Using the same methodoloty, these are frequency graphs with the number of samples in the yy axis and interval considered in the xx axis. These allow a different view of the performance of the module.

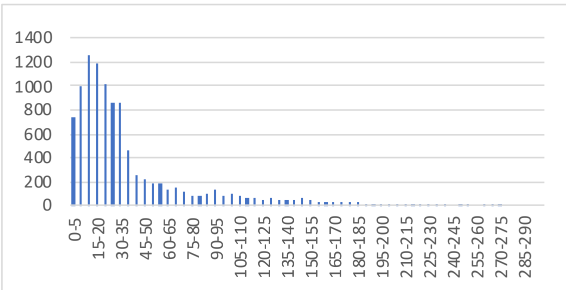


Figure 4-23 - Output from electret microphone with LM393 and 6mm capsule at 30dB

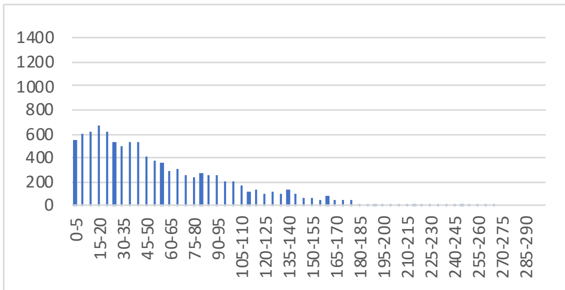


Figure 4-24 – Output from electret microphone with LM393 and 9.6mm capsule at 60dB

Comparing these results to the previous modules we can verify that this module is capable of detecting differences in sound pressure level between both environment conditions. Also, by the results in Table 4-7, this module produces amplification of the signal at the different settings as the average value does not change much, but the $\Delta Input$ does. Another aspect of notice is that the values in Table 4-7 clearly indicate that the module identified a higher SPL as the variation between maximum and minimum increases with the increase in noise level (415 at 30db and 472 at 60dB). This is also observable in the results from formula (4), presented in Table 4-8.

Table 4-7 - Performance of the Adafruit electret microphone with adjustable gain

| Adafruit adjustable | Mininum setting 30 dB | Mininum setting 60dB | Medium setting 30 dB | Medium setting 60dB | Max setting 30 dB | Max setting 60dB |
|---|---|---|---|---|---|---|
| Max Input | 561 | 603 | 664 | 666 | 1023 | 1023 |
| Min Input | 420 | 409 | 249 | 194 | 123 | 102 |
| Δinput | 141 | 194 | 415 | 472 | 900 | 921 |
| Average Input | 516 | 514 | 524 | 528 | 540 | 537 |

Table 4-8 - ∑|ΔInput| comparison for 30dB and 60dB for Adafruit electret microphone with adjustable gain at medium setting

| ∑\|ΔInput\| | Medium setting 30 dB | Medium setting 60dB | Variation (%) |
|---|---|---|---|
| Adafruit adj. | 375832 | 560880 | 49% |

This module also exhibits no artefact as the Makeblock as can be shown in Figure 4-25. As before, the yy axis is represented the value reported by the ADC and in the xx axis is the number of the sample.

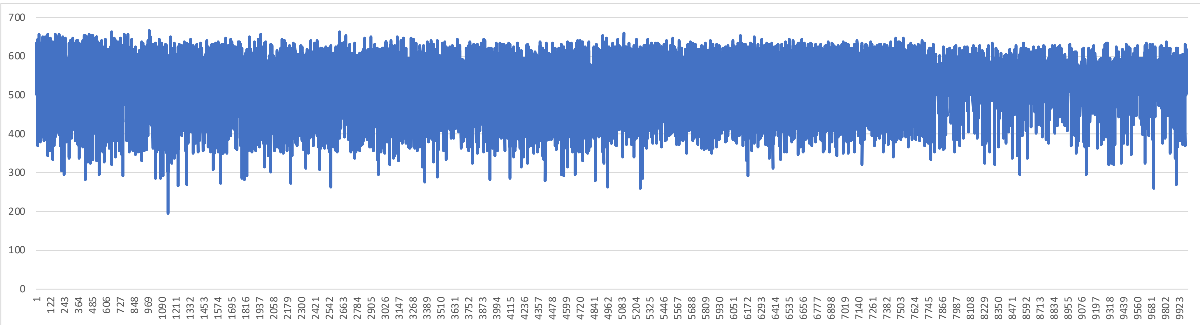


Figure 4-25 - Graph of the results obtained from the Adafruit electret microphone with adjustable gain at 60dB

### 4.3.5 Adafruit I²S MEMS Microphone Breakout (3421)

This module is different from the previous ones as it uses a digital interface, namely I²S [108]. This requires a different program, available in Annex C. The ADC is inside the MEMS microphone. As such, the board receives the converted values, and it will not use its own ADC.

Connecting the module to the Seeeduino LoRaWAN is simple. However, since it has no labels related to the I²S and the documentation does not reference it, a study of the schematic [109][99] allowed to map the pinout of the controller chip to the pinout of the board. The results are the following: the BCLK (clock) connects to pin 1, DOUT (signal) connects to pin 9, LRCL (left/right clock) connects to pin 0. More documentation about the functionality of the microphone can be obtained from the datasheet of the included MEMS microphone [91].

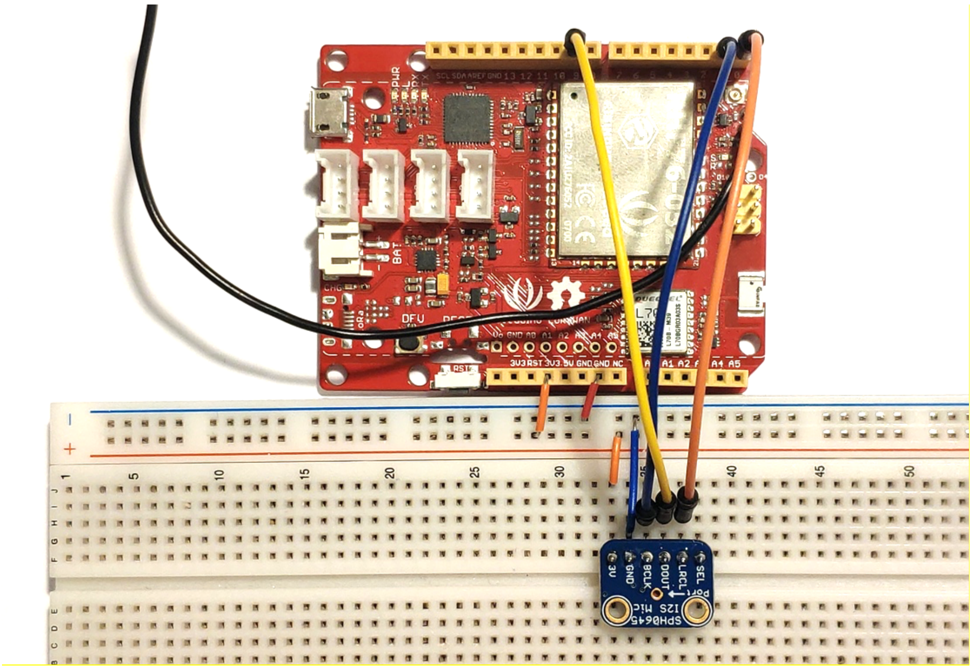

Figure 4-26 - Adafruit I²S MEMS microphone to the Seeeduino LoRaWAN w/GPS with I²S

An important difference between this module and all the previous ones, is that the analogue to digital conversion is taking place inside the microphone, and not at the ADC in the microcontroller. Another key difference is that the ADC of the microcontroller has 10-bit resolution, and the microphone package has 32-bit resolution. This is several orders of magnitude difference. In fact, the difference is given by:

$$\frac{2^{32}}{2^{10}} = 4194304 \ (5)$$

With such a massive difference, the values captured have to be converted in order to be comparable to the previous modules. All values are converted by the factor given in formula (5). Taking this into account, the information is summarized in Table 4-9, Figure 4-27 and Figure 4-28, following the same methodology as before.

Table 4-9 – Performance of the Adafruit I²S MEMS Microphone breakout - SPH0644L4H (3421).
**Note:** converted values

| Adafruit SPW2430 | 30 dB | 60dB |
|---|---|---|
| Max Input | 498,27 | 501,61 |
| Min Input | 498,01 | 495,21 |
| Δinput | 0,26 | 6,41 |
| Average Input | 498,12 | 501,61 |

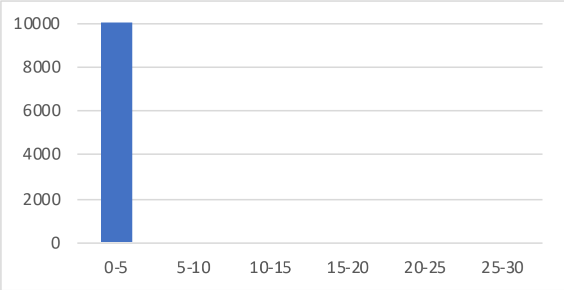


Figure 4-27 – Output from Adafruit I$^2$S MEMS Microphone Breakout at 30dB. Converted values.

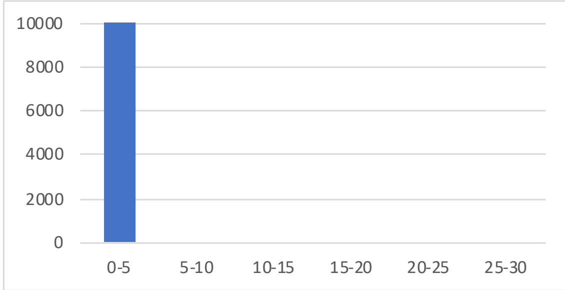


Figure 4-28 – Output from Adafruit I$^2$S MEMS Microphone Breakout at 60dB. Converted values.

Comparing the information from Figure 4-27 and Figure 4-28 alone, we conclude that there is no difference and this microphone board does not fit the purpose. However, a deeper look into other data allows to draw a very different conclusion. A hint is present in Table 4-9, as the values captured at 30 dB and 60 dB are quite different. This is even more visible when the values of $\sum |\Delta Input|$ are compared.

Table 4-10 - ∑|ΔInput| comparison for 30dB and 60dB for Adafruit I2S MEMS Microphone Breakout

| ∑\|ΔInput\| | 30 dB | 60dB | Variation (%) |
|---|---|---|---|
| Adafruit 3421 | 330,13 | 7251,68 | 2097% |

This is the higher variation value so far. As the microphone has much higher resolution it also means that it can detect much higher and much lower values of ambient noise. This makes it a prime candidate for the luminaire node. It does, however, require a microcontroller with a I$^2$S interface.

This microphone also presents no artifact as can be shown in Figure 4-29. In this graph the yy axis presents the input at their absolute value (i. e., not converted).

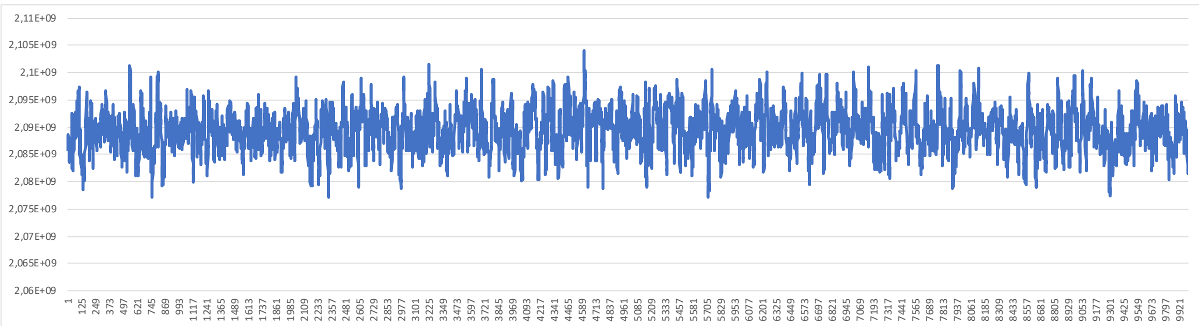

Figure 4-29 - Graph of the results obtained from the Adafruit I$^2$S MEMS Microphone Breakout at 60dB

### 4.3.6 Adafruit Silicon MEMS Microphone Breakout

This is another sensor with good documentation [95]. The documentation states that it can be powered by the 3.3V or the 5V rail so 3.3V was selected. Vin was then connected to the 3.3V rail from the Seeeduino. The DC output of the module was connected to the analogue input A2 of the controller board as shown in Figure 4-30 – Adafruit Silicon MEMS microphone breakout module connected to the Seeeduino LoRaWAN w/GPS on A2 port. This module has no adjustment and after the measurements it was clear that it was not suitable for the purpose. The variation is too small to be detected by the ADC. This is very clear in Table 4-11 as the results are very similar for each environment condition. This is also present in the results from formula (4), presented in Table 4-12, with a variation of 0%.

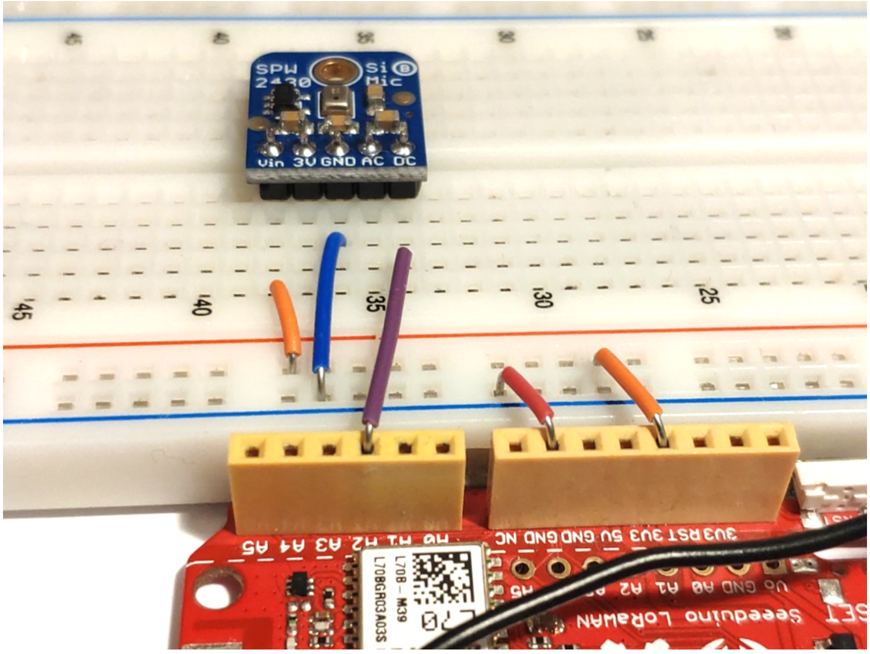


Figure 4-30 – Adafruit Silicon MEMS microphone breakout module connected to the Seeeduino LoRaWAN w/GPS on A2 port

Table 4-11 - Performance of the Silicon MEMS microphone breakout

| Adafruit SPW2430 | 30 dB | 60dB |
|---|---|---|
| Max Input | 249 | 250 |
| Min Input | 244 | 245 |
| Δinput | 5 | 5 |
| Average Input | 246 | 246 |

Table 4-12 - ∑|ΔInput| comparison for 30dB and 60dB for Adafruit Silicon MEMS microphone

| ∑\|ΔInput\| | Medium setting | | Variation (%) |
|---|---|---|---|
| | 30 dB | 60dB | |
| Adafruit SPW24 | 5208 | 5208 | 0% |

The documentation of this microphone board states that:

> *"No additional opamp is included, the output peak-to-peak voltage has a 0.67V DC bias and about 100mVpp (peak-to-peak) when talking near the microphone, which is good for attaching to something that expects 'line level' input without clipping. The peak-to-peak can be as high as 1Vpp if there's a very loud sound."*

This is consistent with the results, as 100mVpp is a very low value. This module requires amplification in order to be suited for the purpose of building a luminaire node capable of evaluating the environment noise in order to report it to the application.

## 4.4 Connection to TTN

The data captured by the device must be sent over a wireless connection and made available in the Application Programming Interface (API) as presented in Figure 4-31. By using the TTN, the data is encoded using the LoRa protocol and received by one or more TTN gateways. These gateways forward the message to the TTN network server and then to the configured application. From here it can be made available to an outside application. In order to accomplish this, the node devices must be configured, the TTN application created and an integration selected. These steps were tested and are demonstrated next. It's important to mention that although the tests are done with a Seeeduino LoRaWAN w/ GPS board, they are also valid for other boards with a few adjustments.

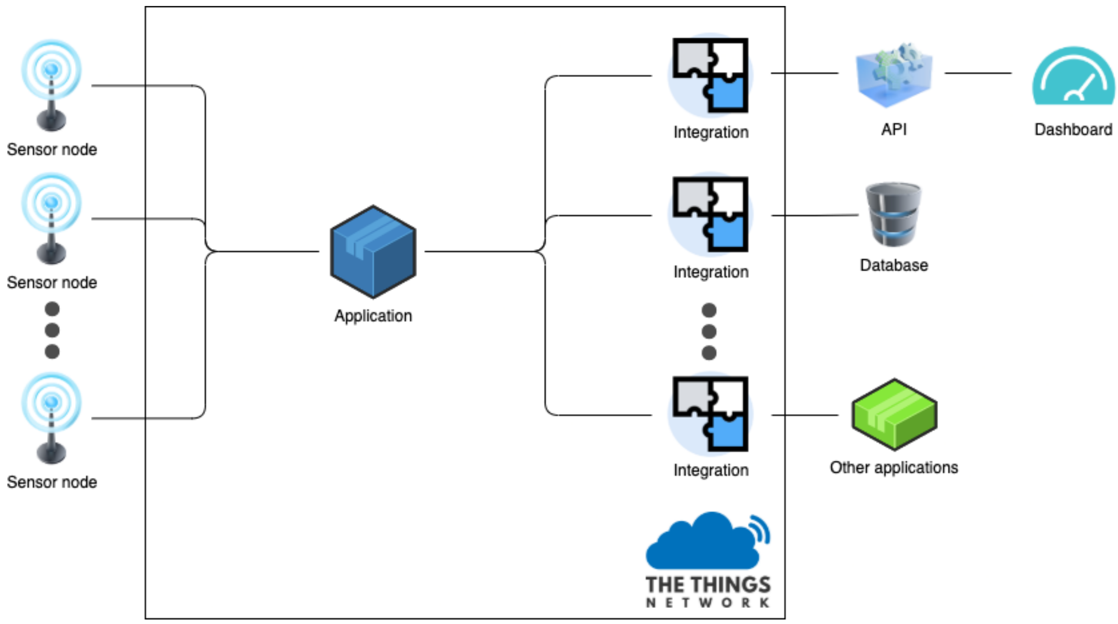


Figure 4-31 - TTN application: from device nodes to the integrations with other services

### 4.4.1 Setting up the communication

A simple and ready to use C++ library is available to set up the communication with TTN. Using it makes the connection to the TTN as simple as filling in a few parameters for any Microchip RN2483 and RN2903 modules. This library is directly available from the Arduino IDE through the Manage Libraries functionality. The library is simply named TheThingsNetwork [110].

With this library it is possible to set up the communication with the TTN. A few parameters are required to start using it. As the communication between the main controller and the LoRa module is done by means of a serial port and serial ports are abstracted as Stream

Objects, two streams are required: one to send data to the the LoRa module and the other to receive the debug information from it. Another required parameter defines the frequency plan. As mentioned in Section 3.3, this is country dependent as detailed in the TTN documentation [30][111]. There are also other optional parameters. The constructor instantiates a *TheThingsNetwork* object and is Figure 4-32:

```
TheThingsNetwork ttn(Stream& modemStream, Stream& debugStream, fp_ttn_t frequencyPlan,
  uint8_t sf = spreadingFactor, uint8_t fsb = frequencySubBand);
```

Figure 4-32 - TheThingsNetwork constructor

This constructor receives the parameters:

- *Stream& modemStream* – stream to send data to the LoRa module;
- *Stream& debugStream* – stream to receive debug data from the LoRa module;
- *fp_ttn_t frequencyPlan* – frequency plan to use, options are: *TTN_FP_EU868*, *TTN_FP_US915*, *TTN_FP_AS920_923*, *TTN_FP_AS923_925* or *TTN_FP_KR920_923*;

The remaining parameters are optional and define other details for the communication such as the spreading factor and the frequency sub-band:

- *uint8_t sf* – spreading factor to use, influencing the airtime and, consequently, the per-message power consumption, varies between 7 and 12 (except for the United States where it varies from 7 to 10);
- *uint8_t fsb* – frequency sub-band to use;

As mentioned in Section 3.3, in order to register this device to the TTN, a unique *devEUI* is required. Fortunately, this library provides a simple way of obtaining this information from the module. By using the method *showStatus()*, this information is return over the *debugStream*. For now, no code is required in the *loop()* section (Figure 4-33).

This application will return the information over the serial port over to the Arduino IDE. The information returned is presented next. The contents of *DevEUI* is the required information to register the device in the TTN.

```
#include <TheThingsNetwork.h>

#define modemStream Serial1
#define debugStream Serial
#define frequencyPlan TTN_FP_EU868

TheThingsNetwork ttn(modemStream, debugStream, frequencyPlan);

void setup() {
  modemStream.begin(57600);
  debugStream.begin(9600);

  while (!debugStream);       // wait for the Serial Monitor to open

  debugStream.println("-- STATUS");
  ttn.showStatus();
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 4-33 - Example of *showStatus()* use

```
08:42:26.324 -> -- STATUS
08:42:26.324 -> EUI: 0004A30B001E1258
08:42:26.324 -> Battery: 3294
08:42:26.357 -> AppEUI: 0000000000000000
08:42:26.357 -> DevEUI: 0004A30B001E1258
08:42:26.357 -> Data Rate: 0
08:42:26.357 -> RX Delay 1: 1000
08:42:26.393 -> RX Delay 2: 2000
```

Figure 4-34 - Output from the example of *showStatus()* use

This example uses a more generic LoRaWAN library provided by SeeedStudio. It contains an empty constructor and uses the *extern* keyword to define the *lora* object. It also contains a method for obtaining the *DevEUI* (Figure 4-35).

```
/**
 *  \brief Read the ID from device
 *
 *  \param [in] *buffer The output data cache
 *  \param [in] length The length of data cache
 *  \param [in] timeout The over time of read
 *
 *  \return Return null.
 */
void getId(char *buffer, short length, unsigned char timeout = DEFAULT_TIMEOUT);
```

Figure 4-35 - *getId* method

It is now possible to adapt the previous source code for the Seeeduino LoRaWAN w/GPS. This adaptation will be able to provide us with the *DevEUI* to register the device in TTN (Figure 4-36). Similar procedure should be made to any other board, adapting the code to the particular set of peripherals used.

```
#include <LoRaWan.h>

#define debugStream SerialUSB

char buffer[256];

void setup(void)
{
    debugStream.begin(115200);

    while(!debugStream);      // wait for the Serial Monitor to open

    lora.init();

    memset(buffer, 0, 256);
    lora.getId(buffer, 256, 1);
    debugStream.println("-- STATUS");
    debugStream.print(buffer);
}

void loop(void) {
  // put your main code here, to run repeatedly:

}
```

Figure 4-36 – Program using the *getId* method

The output of the application is very similar to the previous one. Here we can find the *DevEUI* to register the device in TTN, so that is what will be demonstrated next.

```
09:09:08.082 -> -- STATUS
09:09:08.082 -> +ID: DevAddr, 01:29:A5:F6
09:09:08.082 -> +ID: DevEui, 00:AA:65:8E:57:E6:76:DD
09:09:08.082 -> +ID: AppEui, 70:B3:D5:7E:D0:01:BE:B1
```

Figure 4-37 - Output from the application using the *getId* method

### 4.4.2 Register the device in TTN

To start using a device within the network a few simple steps are needed. Fist, a user registration in TTN is required. After logging in the web site www.thethingsnetwork.org go to “console” and then “applications”. From there an application can be created.

The first step aims to create an application that will receive the data. To add an application (Figure 4-38) it is required a unique “Application ID”, a description, a “Application EUI” (assigned by the TTN), and the selection of the handler where the application will be registered. The “Application ID” is an identifier, it must be unique and will displayed in the applications list along with the description.

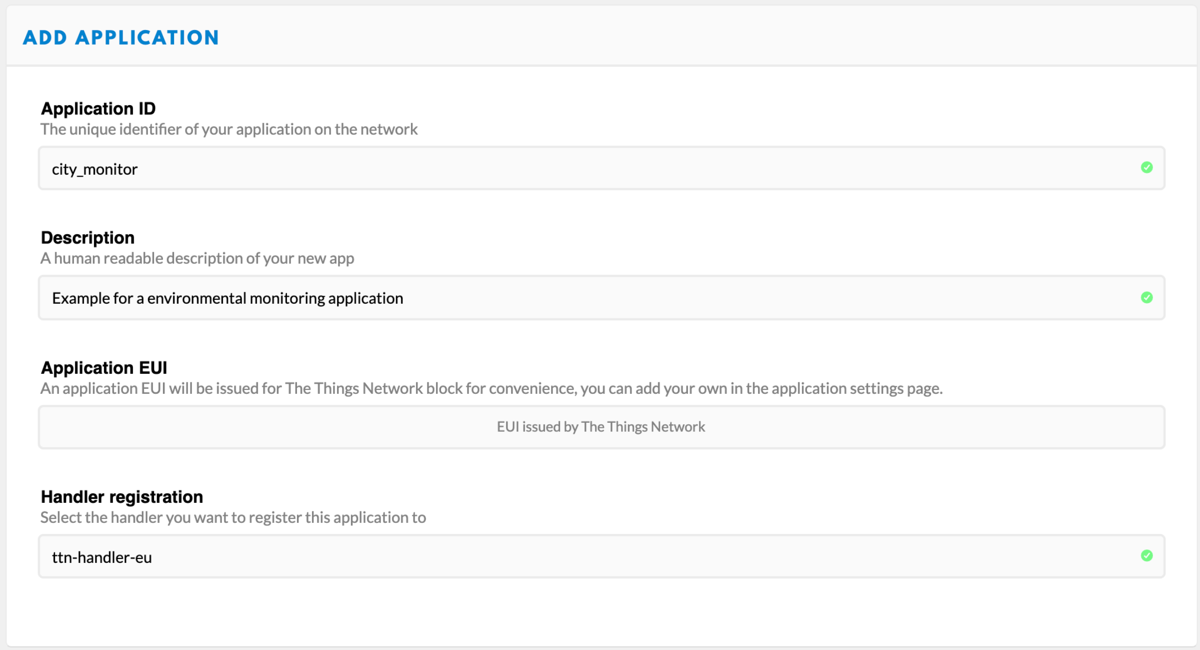


Figure 4-38 – TTN console: Add application

After the creation of the application, the devices can be added. In the Application Overview the user must select “register device” and then fill in the “Device ID” and the device EUI. The App Key and App EUI will be automatically be filled. The “Device ID” must be unique within an application and the device EUI must correspond to the one on the device. In Figure 4-39 the data from the output of the application from the Seeeduino LoRaWAN was added.

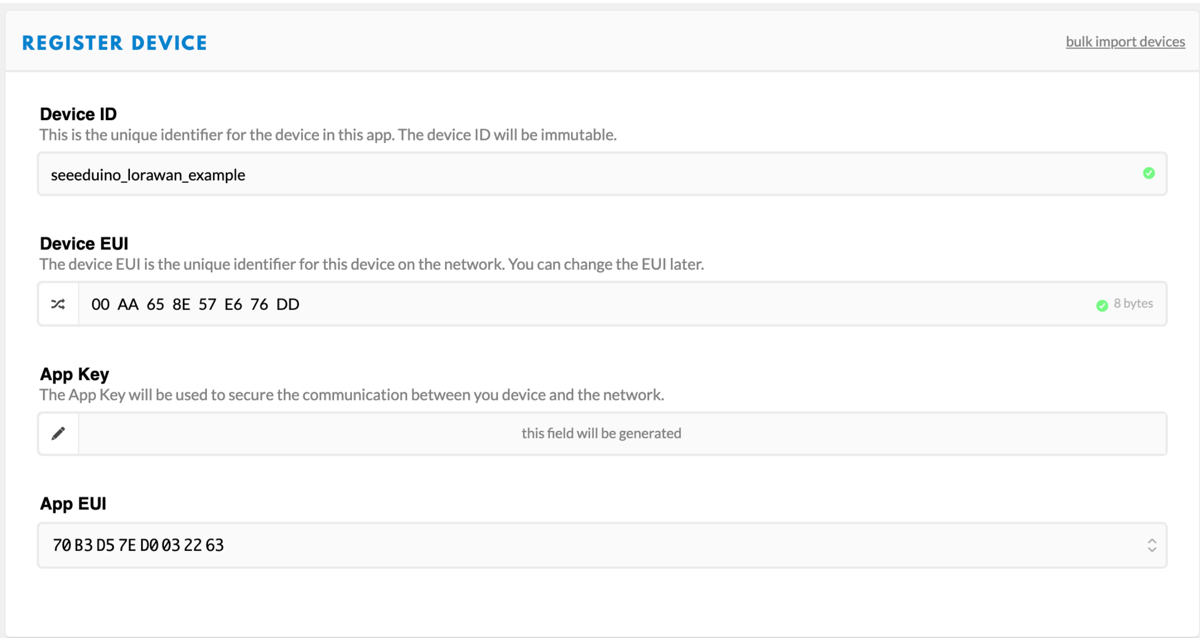


Figure 4-39 – TTN console: Register device

Next, the device is ready to communicate with the application.

### 4.4.3 Send data to the TTN

The first thing to do in order to send data to the TTN is to join the network. After this step the mentioned libraries can be used to send data. Starting example provided by the manufacturer, a simple "Hello World" program is implemented. This is done for demonstration purposes only because, as mentioned earlier, the payload should be kept to a minimum and text is not the ideal means of transmitting data. The result of this process can be confirmed in the gateway traffic (Figure 4-40), application data (Figure 4-41), application message details (Figure 4-42 and Figure 4-43) and the serial output from the Arduino IDE (Figure 4-44). It is important to notice that the gateway has access to the "Physical Payload", encrypted at the device and it is unable to decrypt it. The communication is secured from the device to the application.

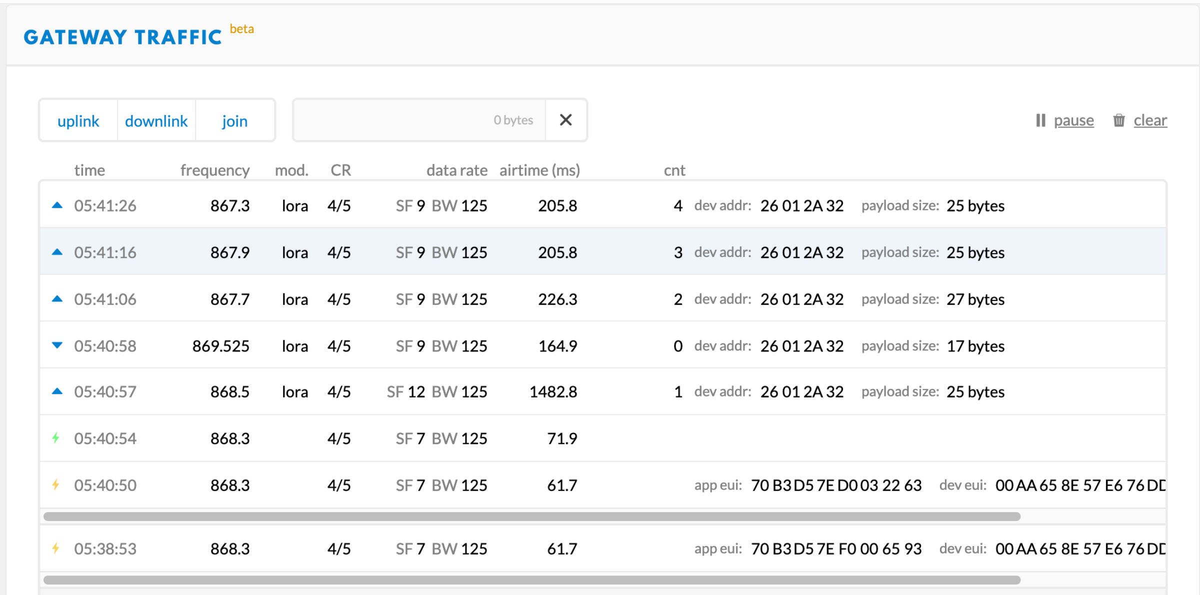


Figure 4-40 - Gateway traffic

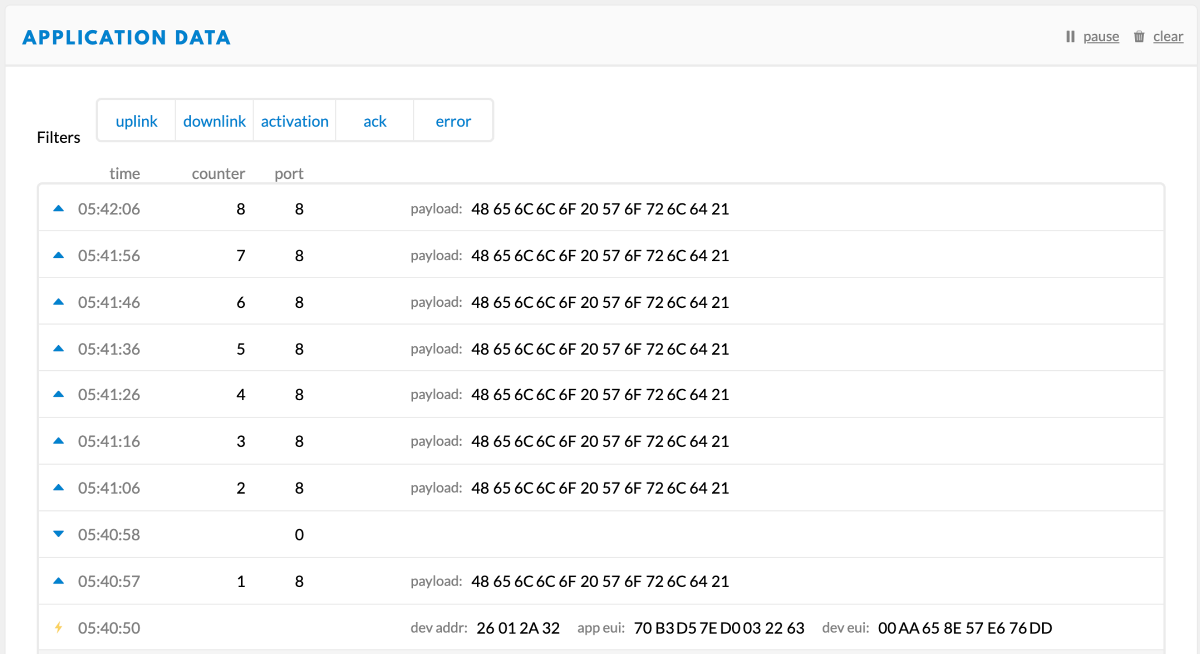


Figure 4-41 – Application: data

In the application data we can see the payload transmitted. Decrypted but encoded in hexadecimal form. Decoding the payload using any available tool results in the message "Hello World".

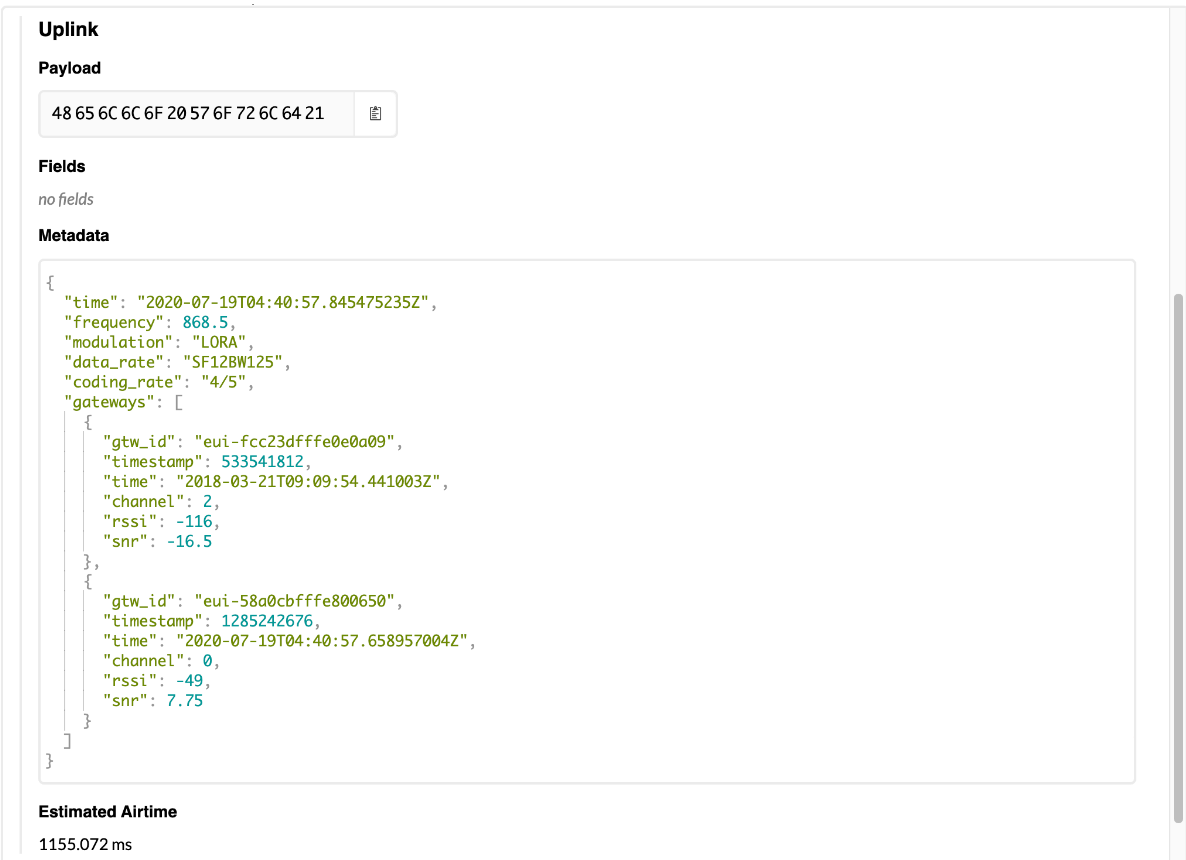


Figure 4-42 – Application: message details

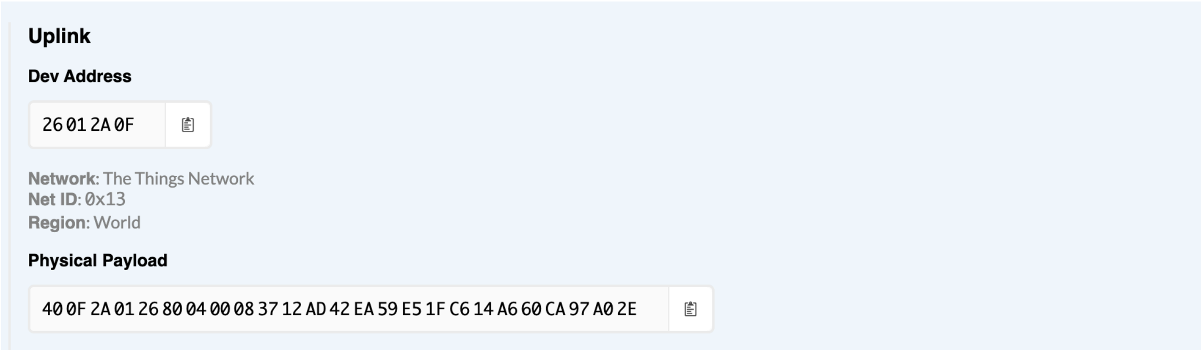


Figure 4-43 - Extract from the gateway message detail

```
05:40:35.874 -> +VER: 2.0.10
05:40:36.879 -> +ID: DevAddr, 01:29:A5:F6
05:40:36.879 -> +ID: DevEui, 00:AA:65:8E:57:E6:76:DD
05:40:36.879 -> +ID: AppEui, 70:B3:D5:7E:F0:00:65:93
05:40:36.952 -> +ID: AppEui, 70:B3:D5:7E:D0:03:22:63
05:40:37.027 -> +KEY: APPKEY 24 42 80 47 05 9A 15 92 9B 00 B7 80 3E 17 12 8F
05:40:38.142 -> +MODE: LWOTAA
05:40:39.230 -> +DR: EU868
05:40:40.282 -> +DR: DR0
05:40:40.282 -> +DR: EU868 DR0  SF12 BW125K
05:40:41.396 -> +CH: 0,868100000,DR0:DR5
05:40:42.483 -> +CH: 1,868300000,DR0:DR5
05:40:43.601 -> +CH: 2,868500000,DR0:DR5
05:40:44.705 -> +RXWIN1: 0,868100000
05:40:45.791 -> +RXWIN2: 869500000,DR3
05:40:46.880 -> +LW: DC, OFF, 0
05:40:47.990 -> +LW: ERROR(-2)
05:40:49.076 -> +POWER: 14
05:40:50.239 -> +JOIN: Start
05:40:50.239 -> +JOIN: NORMAL
05:40:56.273 -> +JOIN: Network joined
05:40:56.273 -> +JOIN: NetID 000013 DevAddr 26:01:2A:32
05:40:56.273 -> +JOIN: Done
05:41:06.296 -> +MSG: Start
05:41:06.296 -> +MSG: TX "Hello World!"
05:41:06.296 -> +MSG: MACCMD: "03 31 FF 00 01 "
05:41:06.296 -> +MSG: RXWIN2, RSSI -48, SNR 8.8
05:41:06.296 -> +MSG: Done
```

Figure 4-44 - Output from the serial port in Arduino IDE

As mentioned before, the message is received in the application, already decrypted, but encoded in the hexadecimal format. The TTN has an incorporated decoder so the message can be decoded in the TTN application or sent as a hexadecimal format to an external application. In order to use the decoder, in the Application, select the “Payload Formats” tab and insert the decoder code. In this case, the decoder code only converts the hexadecimal value to the corresponding character and returns the result. The result is the original message, properly decoded as presented in Figure 4-45. It is important to notice that it also provides a converter, validator and encoder functions. The encoder function is used for downlink messages, that is, to send data to the nodes. The others can be used cumulatively and are used for uplink messages.

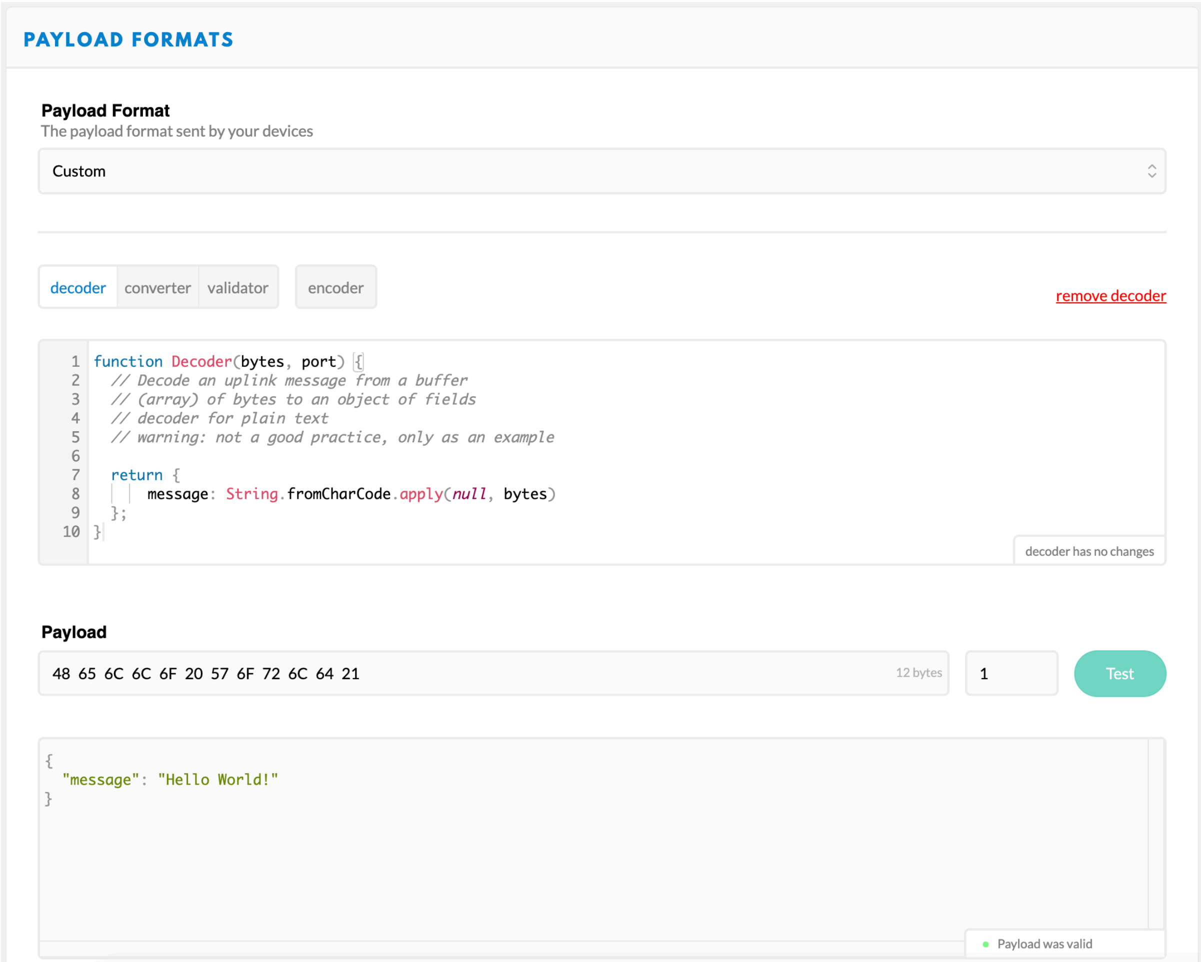


Figure 4-45 - TTN decoder function

Using these functions, the very optimized byte stream sent from the device can be converted into something more human readable like a JSON object [112]. However, to send this data to an external application the MQTT protocol [113] or an integration must be used. Integrations are one way of sending the collected data out of the TTN. These use the same API's and SDK's that an external application could use directly. The TTN provides a list of useful integrations (Figure 4-46) such as Data Storage, HTTP Integration , IFTTT [114] and TTN Mapper [115], just to name a few.

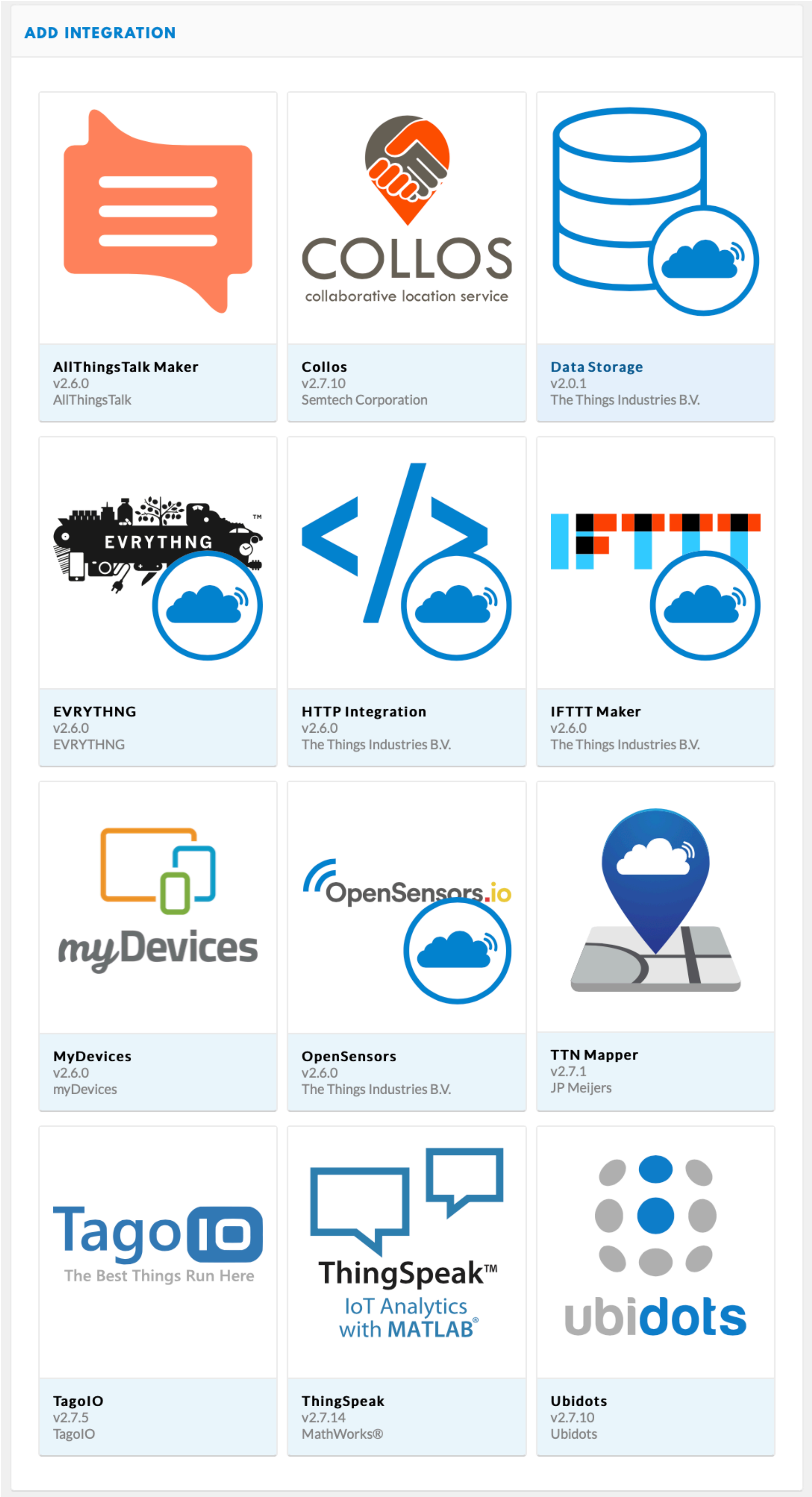


Figure 4-46 – TTN Console: available integrations. Source: [116]

### 4.4.4 Configure a gateway

The data sent over the LoRa protocol is received by properly configured TTN gateways that forward the message to the TTN network servers. This is accomplished by a packet forwarder in the gateway as described in Figure 2-5. A packet forwarder is a program, running on a gateway, that receives and transmits data from and to the LoRa hardware and forwards them to/from the network servers. In practical terms, the configuration process of a gateway is simple, requiring registration on the TTN and on the device.

In the TTN website, inside "Console", go to "Register Gateway" and fill in the gateway ID, a description, a frequency plan, the router and the location of the antenna. The frequency plan is geography dependent and the router selected should be the one with the closest location relative to the gateway. If the gateway has no GPS location, the location information is also required (Figure 4-47)

On the gateway, depending on the brand/model, it may be required to setup the packet forwarder. The installation instructions for the gateway should contain information about how this can be accomplished. If not, the TTN has currently 3 options for this setup [117]. These are the Semtech UDP packet forwarder [118], the TTN packet forwarder [119] and the Semtech Basic Station packet forwarder [120]. However, the Semtech UDP packet forwarder is deprecated as it contains several flaws and the development of the TTN packet forwarder is currently on hold. The currently best option is the new Semtech Basic Station which is based on WebSocket protocol [121].

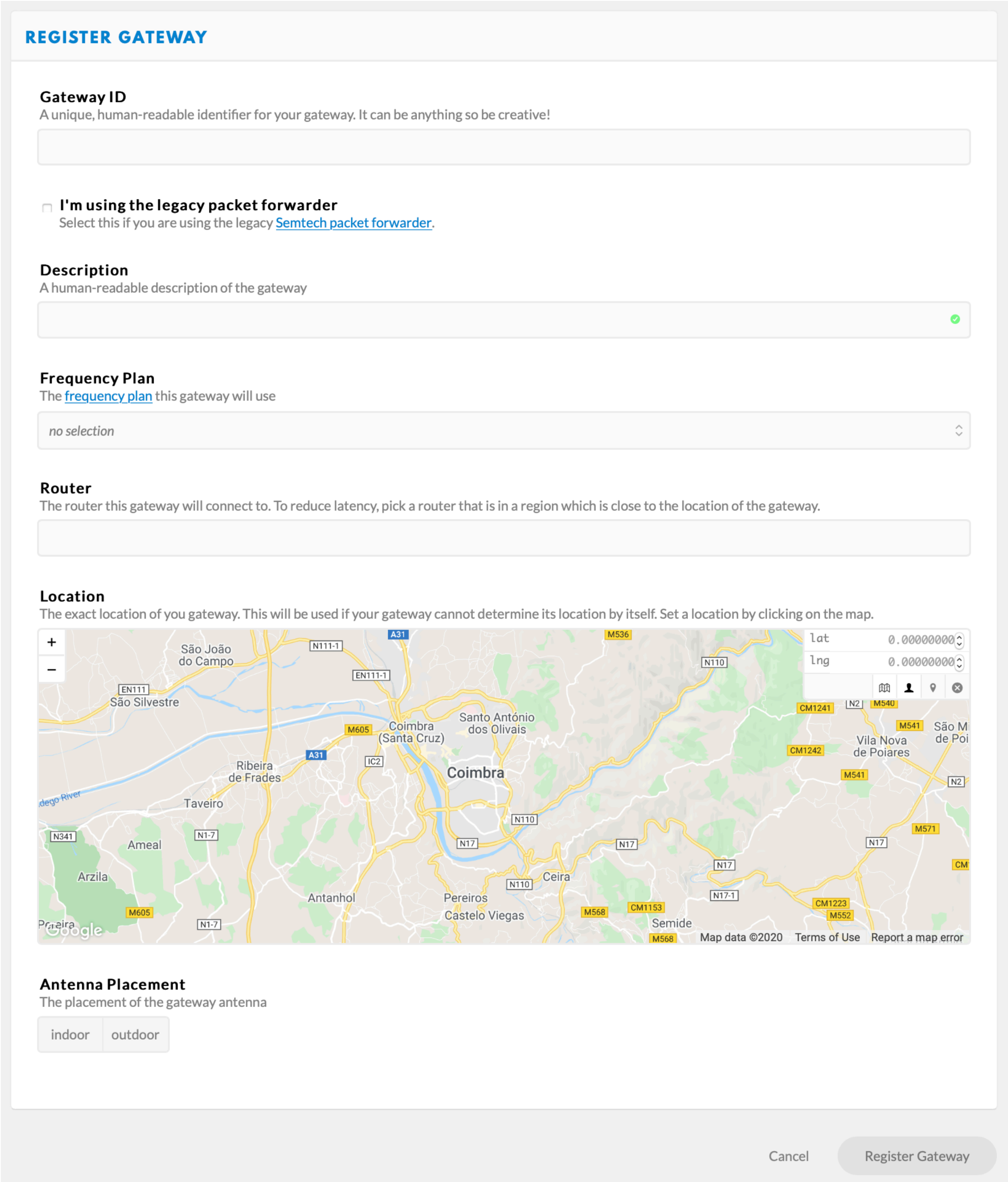


Figure 4-47 – TTN console: register gateway. Source: [116]

## 4.5 Application

The application is the core of the system. It collects and manages the data from the sensors, configures luminaires and actuators, manages users and alarms. It has all the business logic that can be tailor-made to the specific needs of the client. The definition of an application is proposed in this section. This solution follows the three-tier architecture [122]. This is a very modular approach (presentation, business logic and data layers) to software providing the ability to be adapted to several solutions.

### 4.5.1 Data model

The data is collected at the sensors located in the luminaires. This data is related to the health status of the luminaire as well as environmental data. All this data is stored in a database, along with other information such as the users and actuators. The proposed database model consists in 6 tables: users, luminaires, sensors, actuators, actuator events, as well as historic data for the sensors and actuators. The logic model is specified in Figure 4-48.

Some of the settings are not available to all users and this can also be tailor made to the needs of a client. It makes sense that the city inhabitants have access to the environmental data provided by the sensors. City officials will have access to the luminaires health status and will be able to configure the schedule of the lights, for instance. The system also allows to power on the lights by schedule or by light intensity, globally, per luminaire or groups of luminaires.

A user can register several luminaires and a luminaire can have several users. However, a luminaire does not exist without a user. In the case of a collective, several users have access to the same luminaire. The remainder of the database model is pretty straightforward, the luminaires have sensors and actuator with their historic data. The actuators can be configured with different events, by time and date. An example of an actuator is a switch that controls the nearby public space irrigation for instance, based on a configured schedule and on a rain and/or soil humidity sensor. These sensors can even be located in different luminaires. The actuator events can also be driven by the actions of a user. This model considers that the sensors that monitor the health of the node (temperature, light intensity, power) are stored in the same table as the sensors for environmental data. The distinction between them is based on their types.

This generic logic data model is converted to a physical model as presented in Figure 4-49. As a consequence, an auxiliary table, connecting users to luminaires is needed.

The data from the sensors is collected directly from the TTN and stored in the database by the application. This application also manages the access to the data, users and permissions, luminaires and owners, alarms, as well as firmware updates. The firmware updates allow for the luminaire nodes to be reconfigured and updated when new functionalities are developed. It also offers a REST API allowing the data to be made available for other applications, including frontends (i.e., user applications) for different types of devices. This modular approach allows for much easier maintenance and optimizes development effort.

As mentioned earlier, not all luminaire nodes will be equal, and their functionality can be extended after installation. This is made possible by the modularity of the nodes. As such, the sensors and actuators can be defined in a per-luminaire-basis.

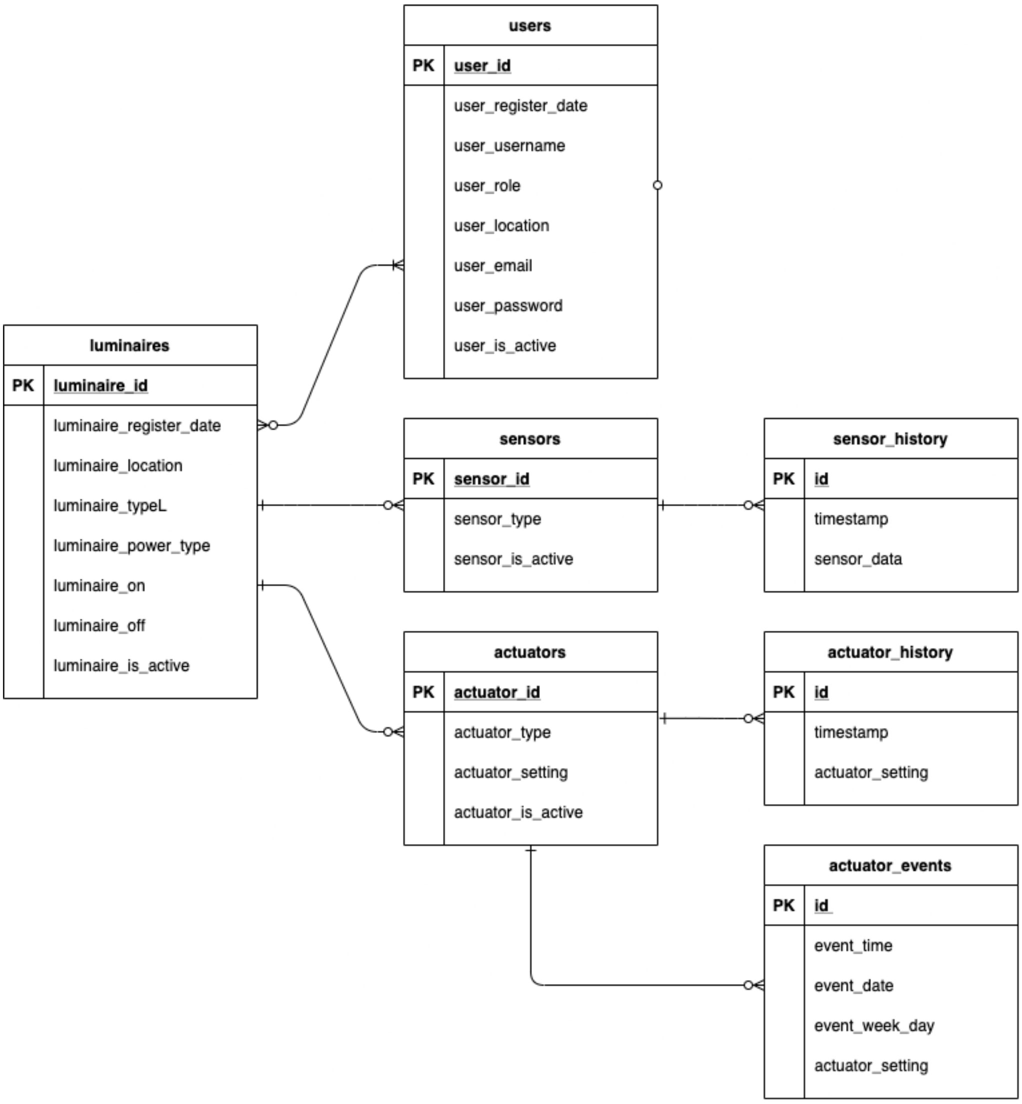


Figure 4-48 - Logic data model

The data collected by the sensors is analysed in the application and alarms are triggered when a configured condition occurs. These alarms can be related to the health status of a luminaire, such as power being on, but no light beig detected, or the environment data exceeds a given threshold.

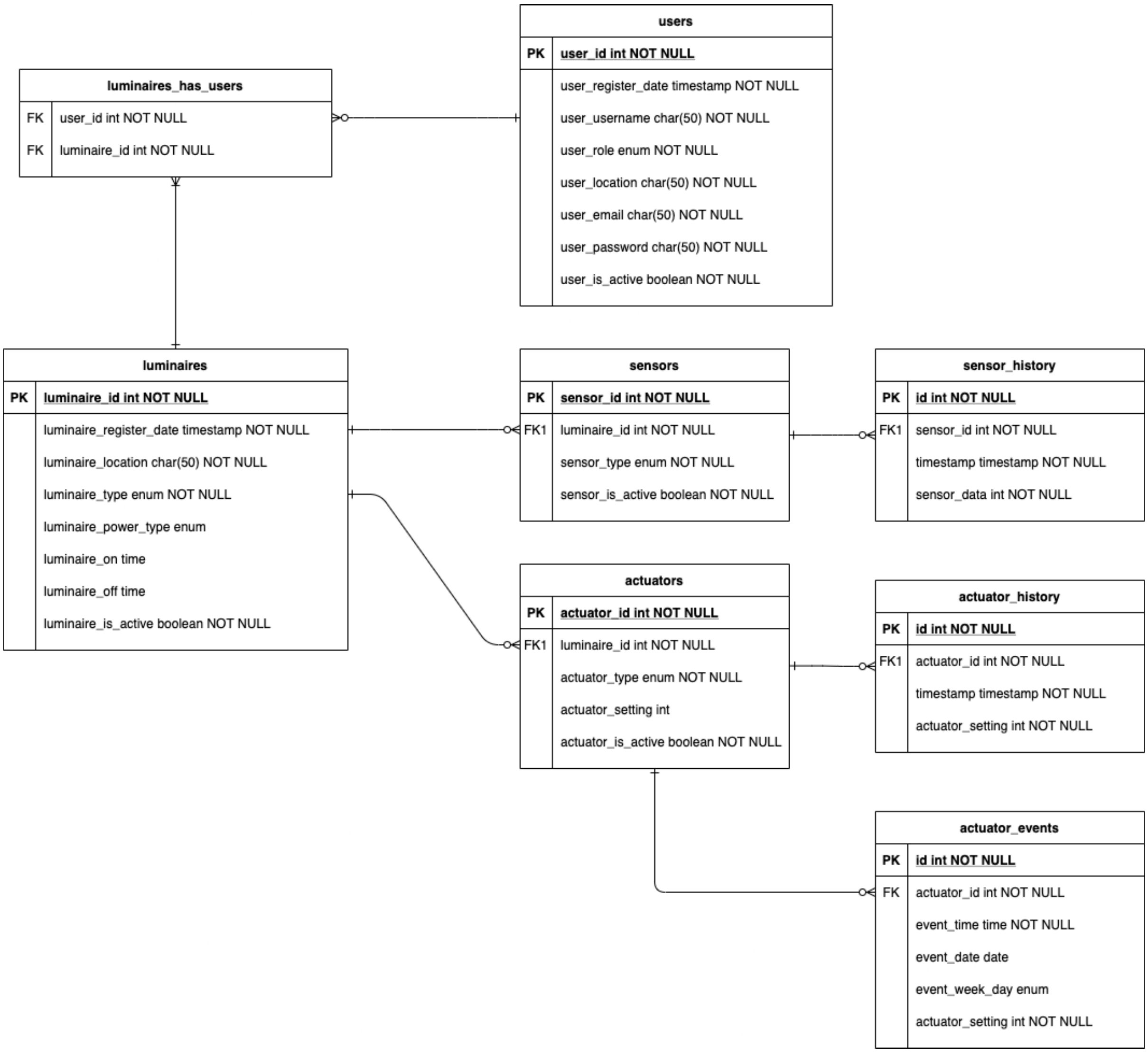


Figure 4-49 – Physical data model

### 4.5.2 REST API

The proposed REST API [123][124] provides access to common needed information using an access key in the request. The information made available by the REST API is presented inFigure 4-50, and Figure 4-51. This Swagger API framework [125] was used to produce this description. It is important to notice that the proposed REST API do not present a *delete* endpoint when a soft-delete solution is preferred.

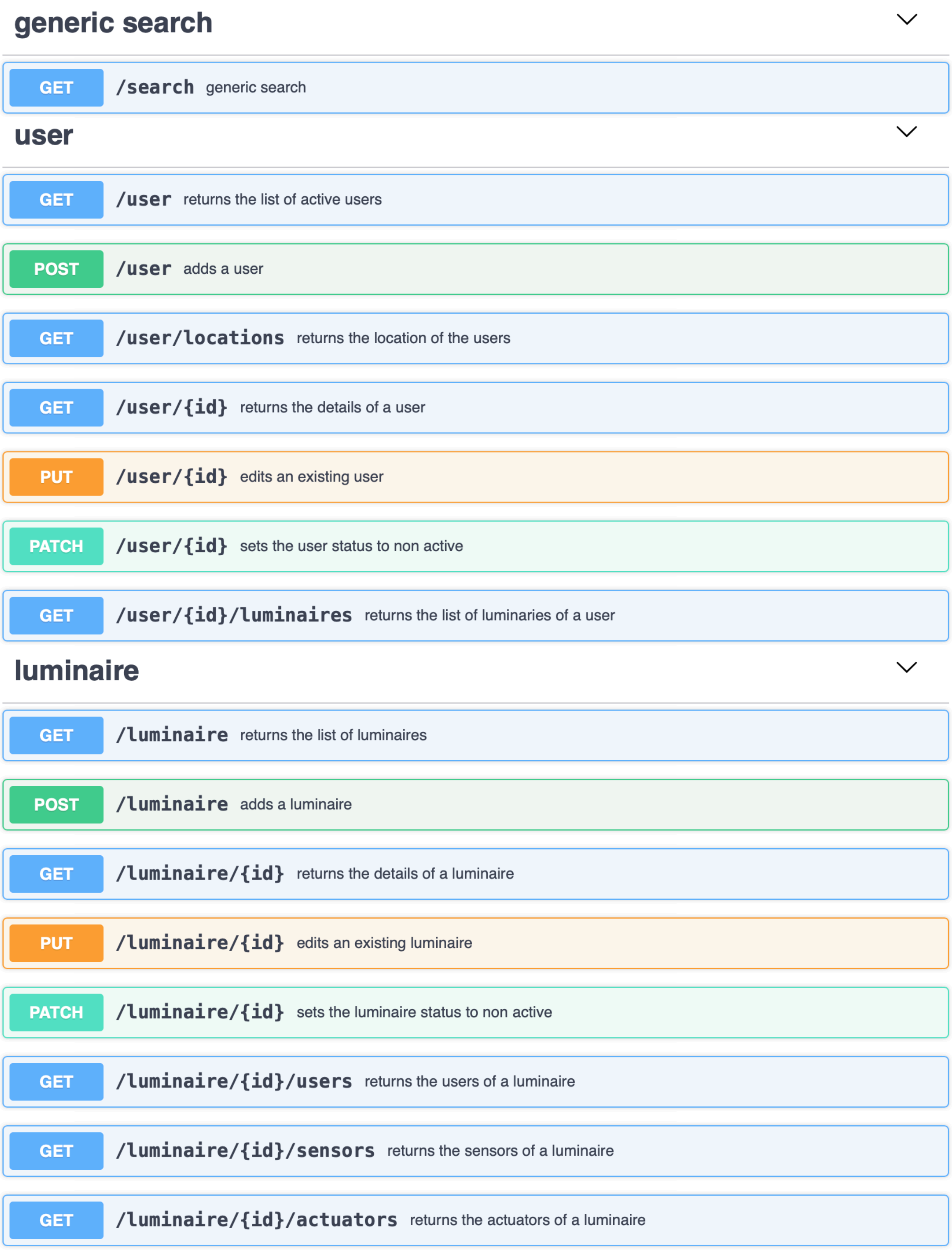


Figure 4-50 - Generic search, user and luminaire endpoints

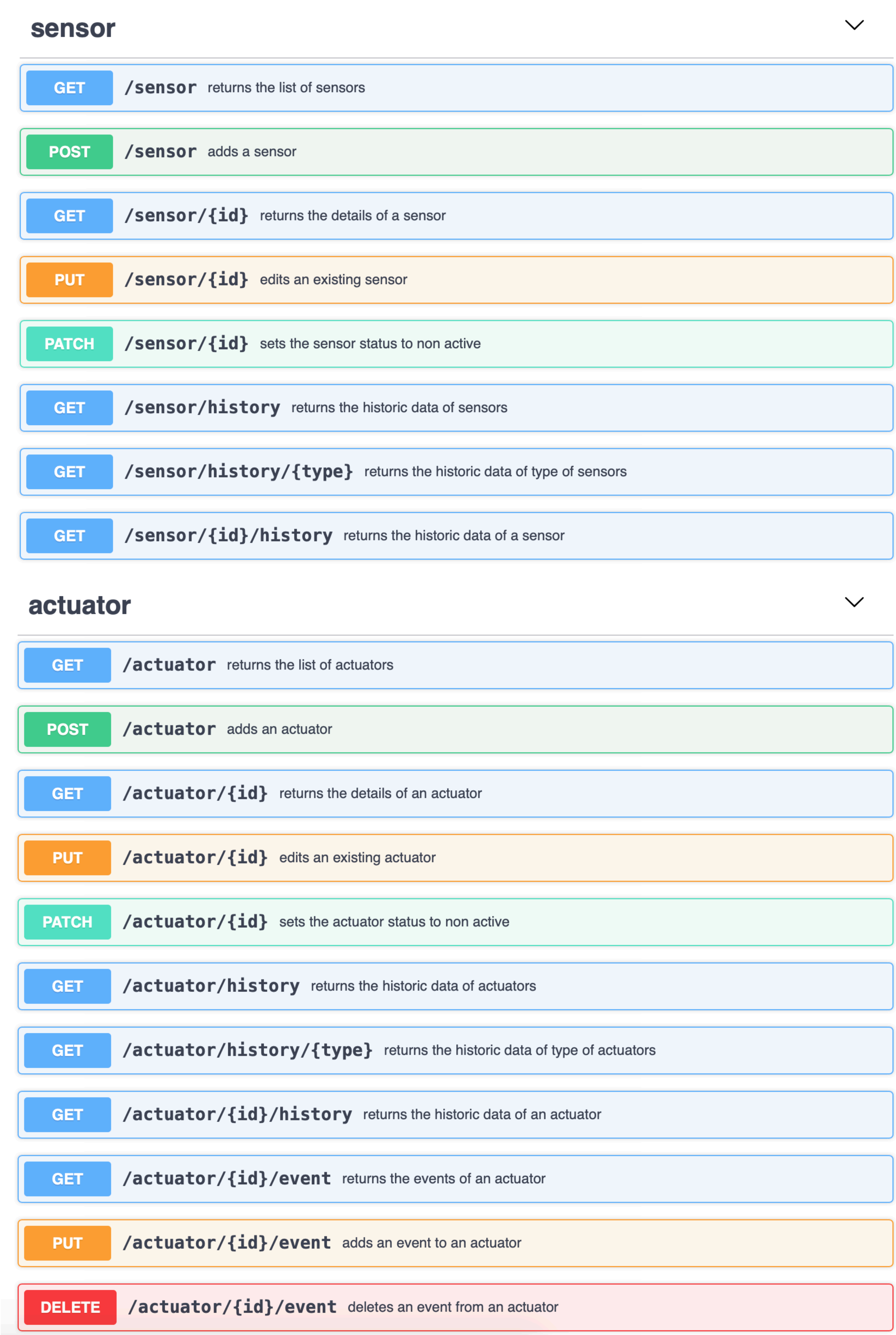


Figure 4-51 - Sensor and actuator endpoints

### 4.5.3 User interface example

The proposed solution concerning the frontend (user application or user interface) is one of many that can be used to access the data and configure the system. The options presented to the user depend on the login credentials. The role of the logged user defines the access to the different areas. Also, the user has the capability of configuring his own luminaires and the others he has been given access to.

The following user interface example mockups are presented as web pages but the same can be applied to be used in a mobile application or even through an application with a text user interface. The landing page contains the login and register options to access the webpages as indicated in Figure 4-52. Here, a user can register to access the application by the use of a username and password. During the registration process, an email is required as well as the user location.

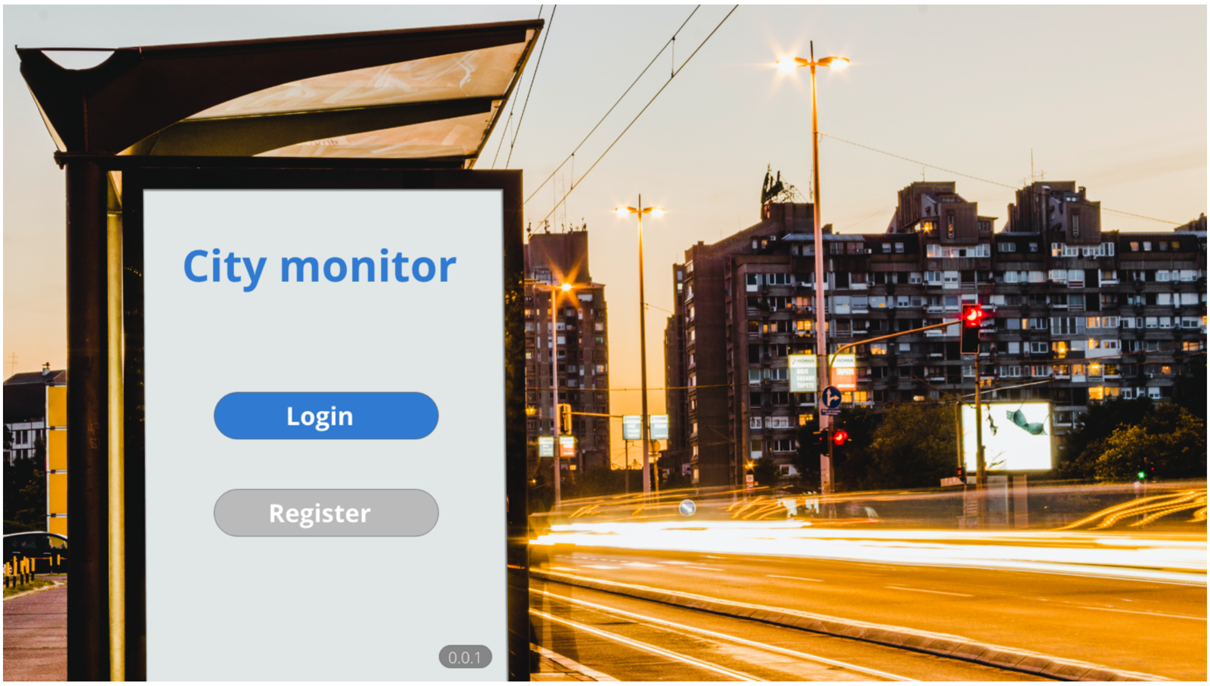


Figure 4-52 – Application login page

After logging in, the default screen presents a map, that can be zoomed in and out, allowing to search for a specific smart luminaire. This is represented in Figure 4-53. On the top left, a search bar exists allowing to search for any field. This search can be used to search for road, street, city park, train station or any other landmarks that can be localized on a map.

Below this search bar, there are several icons related to each type of data that can be measured by the system. This information includes, but is not limited to, the following:

- Noise level;
- Streetlight intensity;
- Relative humidity;

- Wind speed;
- Sunlight intensity;
- Ambient temperature;
- Nitrogen dioxide;
- Sulfur dioxide;
- Carbon dioxide;
- Particulate matter, 10µm.

The sensors that can be adapted to this system can capture other environmental parameters such as particulate matter of 2.5µm, precipitation, soil moisture, UV index and wind direction, to name a few. It can also collect data about the performance of the light module. Besides the streetlight intensity it can also collect data related to the module temperature and energy consumption for instance. This information can be used to detect malfunctions and to alert the city officials.

Besides the information related to the sensors, this list can also include the information regarding actuators, such as the time the streetlights will turn on or the irrigation of the public park will start. By making this information public, the citizens can plan if they will sit on the grass at a particular time or avoid it, since the irrigation will start soon. This is just one of the examples of what is possible to achieve with this system. Since that not all information can be made public, that configuration is done in the application by type of sensor or actuator.

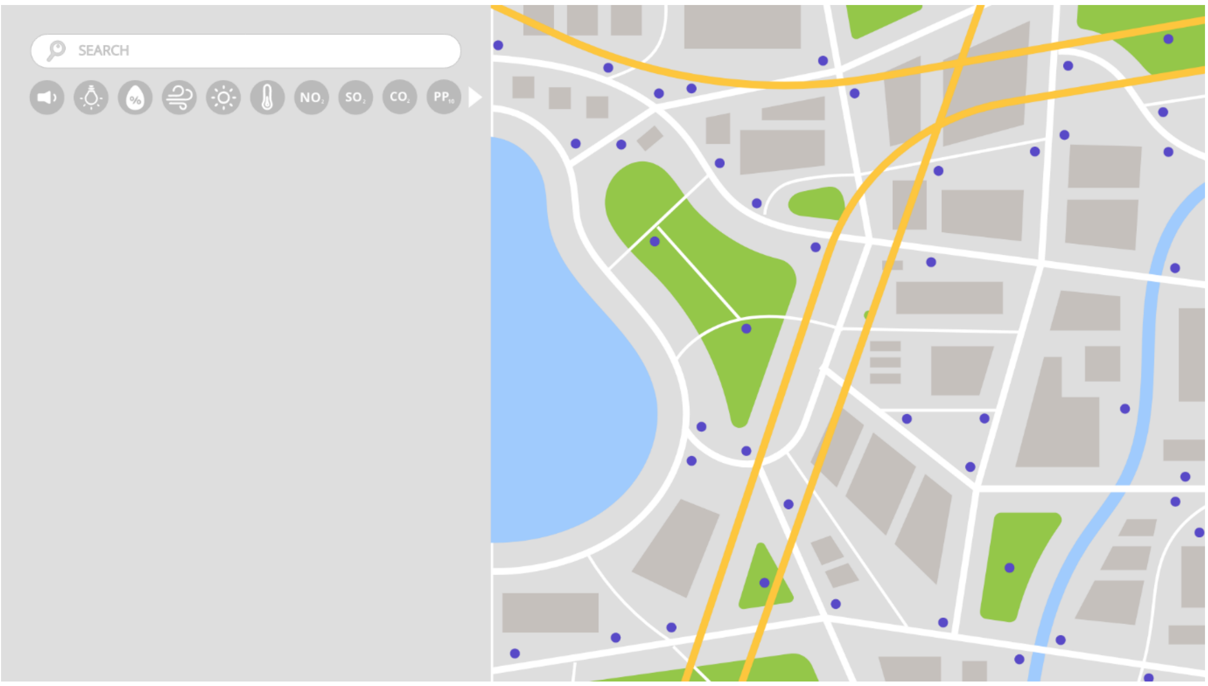


Figure 4-53 – Application default screen

Clicking on one of these parameters the map of the city presents graphically the values currently being read from the sensors. In the bottom of the left side pane, a scale is shown allowing to map the colors shown in the map to the values captured by the sensor. This is represented in Figure 4-54. The colors also place the values captured by the sensors in context, translating from an absolute value to a qualitative indication of what that value means. This is preferable since all citizens can have access to this information and might not have the knowledge to understand the meaning of a particular absolute measurement. As such, the scale goes from green to red, meaning from good to bad.

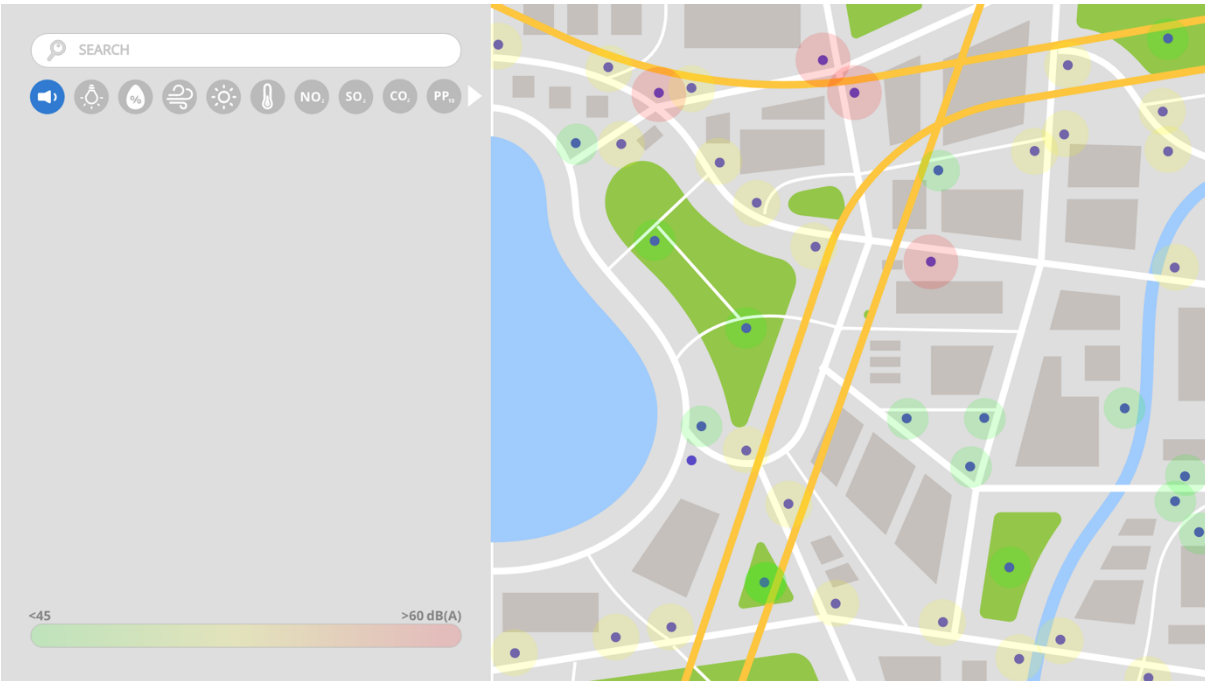


Figure 4-54 – Application: sound map selected

Clicking on one of the blue points on the map, the information about that smart luminaire is presented on the left pane, as presented in Figure 4-55. This information includes the data most recently collected by all its sensors and the configuration of its actuators. As said before, the access to the different types of information and configurations can be limited based on the proper user role. Typically, the citizens will not have access to any node configuration placed by the city officials. However, the modularity of the system allows for this possibility where it makes sense.

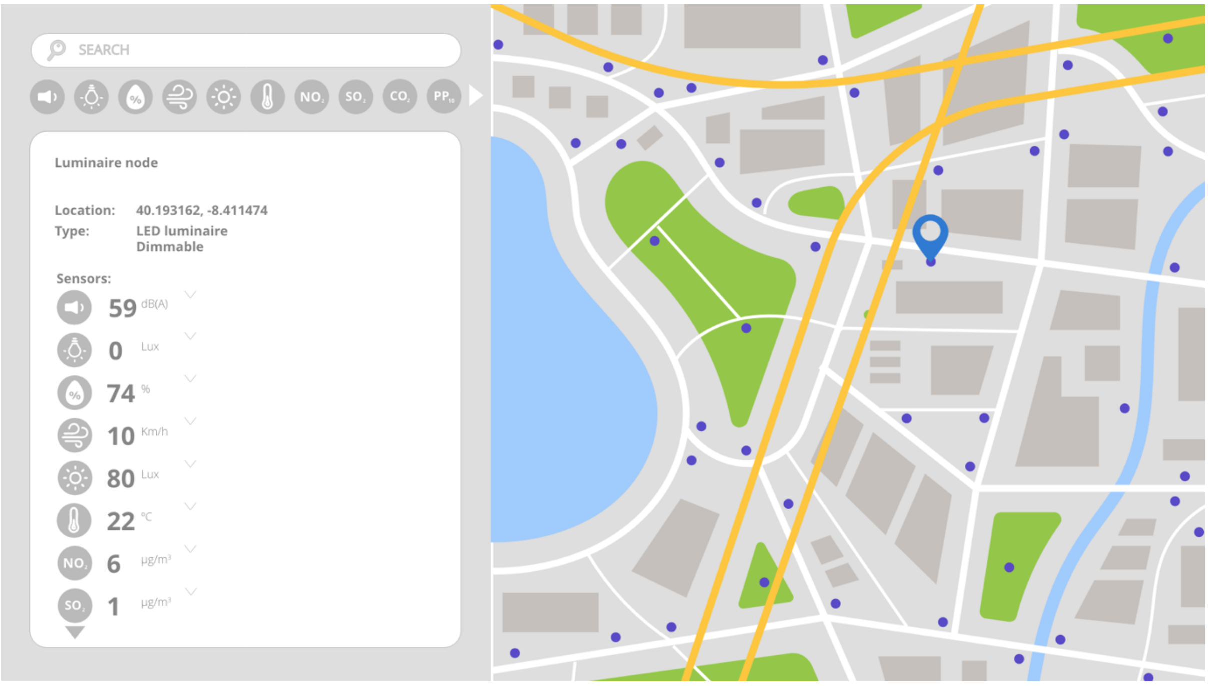


Figure 4-55 – Application: node selected

Besides the most recent value of the sensors and actuators, by expanding one of the types of data (sound level, for instance, as represented in Figure 4-56), an history of the values captured by that sensor is shown. This history can be adjusted by zooming in and out and allows for a more complete view of variation of the parameter in time. This can help in a situation of excess noise at night, for instance. Although the system is not planned to be accurate enough in order to be of legal value, it can give an idea if the claim is legitimate.

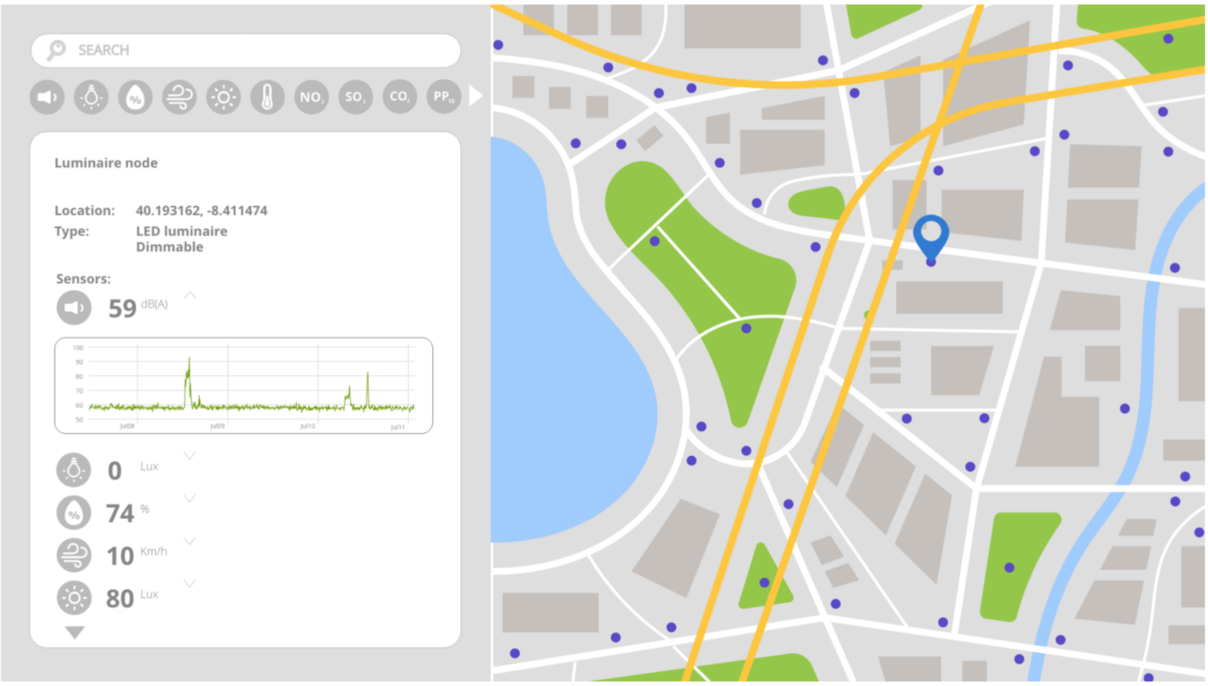


Figure 4-56 - Application: node sound history

Selecting one of the actuators, the information about what can be configured based on the user role is shown. This is the case of Figure 4-57, where the streetlight configuration can be set. In this example the streetlight can be powered on, its power level can be adjusted when it is powered on or, a schedule can be defined, or a twilight sensor can be used. The modularity of the system allows for the streetlights to be controlled at an individual, street, or city level, depending on the preferences.

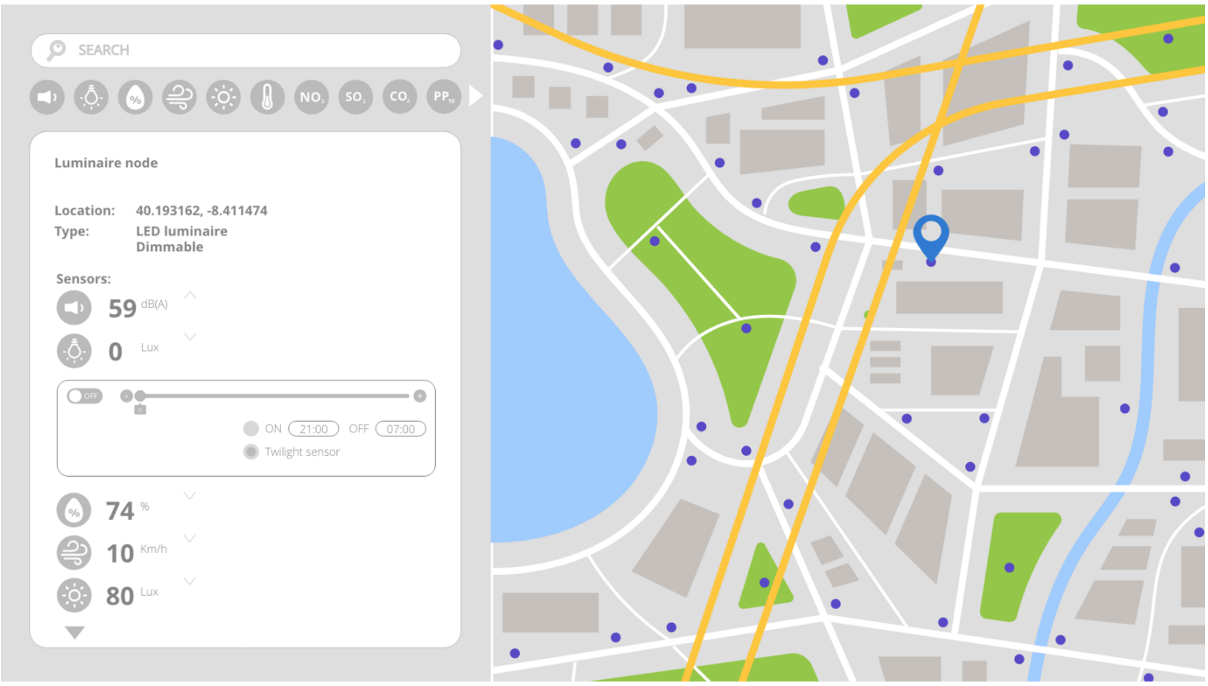


Figure 4-57 - Application: streetlight power configuration

# 5 Conclusions

The environment is nowadays an ever more concern. The change from the traditional public lightning solutions to an eco-friendlier LED based solution is a step in the right direction. However, greater savings can be achieved by adding a small smart module to each luminaire or group of luminaires. This module allows finer control and can be controlled remotely. As such, this work focuses on the opportunity this path creates, in particular to the noise pollution monitoring problem. Adding a more complete monitoring solution of the health status of the luminaire and also of the surrounding environment. This allows for a more connected city but, even more important, clear advantages to the environment that all citizens can benefit from. Without monitoring, it is impossible to act upon the problems that emerge in a timely manner. Furthermore, relying on a manual monitoring solution is expensive and slow.

## 5.1 Contributions

This work focused on how to build a solution for near-real-time noise pollution monitoring while providing the modularity needed to incorporate other sensors as well as to control the public illumination.  The search for a solution was divided in three main aspects: the communication, the sensors and the software. The goal is to develop a system that takes the opportunity of this change in technology to greatly improve the capability of a city to optimize its resources and to monitor the environment, namely noise pollution.

In order to achieve this, several solutions were studied as well as their advantages and disadvantages. Smart cities typically use information collected by several devices scattered around the city that transmit this data using a communication network. For this communication network several solutions are available and in this work the most common ones were considered.

Leveraging the need of electronic modules in the luminaires for energy optimization and control, a modular solution was proposed. This solution manages the public illumination but also other environmental sensors or actuators. It provides the following benefits:

- Controls the luminaire, turning it on, off or dimming. This allows much finer control over the public illumination and allows for further savings in energy;
- Monitors the health of the luminaire, that is, temperature, power consumption, light flux, or other parameters and allows much faster actions in case of a malfunction. With the inclusion of a battery it can also report in the case of a power outage;
- Monitors the environment with several sensors. Although the work focused on noise level metering, the solution is modular, allowing other sensors to be installed. The

nodes in a luminaire can be fitted with the sensors that make sense to install in that particular location. This allows for a near-real-time picture of the environment of the city and allows this data to be shared with the citizens. Alarms can be configured to trigger when a specific parameter is out of limits such as a high noise levels at night;

- Actuates on the surrounding environment, allowing it to double as public irrigation controllers, for example, saving on costs of additional controllers and adds remote control over these infrastructures.

These modules are proposed to be built using very common microcontrollers and modules in order to keep the costs down. Furthermore, the deployment of a large number of modules, even considering the added cost of modularity, will drive the prices down as they are built in volume.

This solution is ambitious and allows to create a system that can provide much more information about the environment, at a much faster rate, allowing economic and resource savings while providing the means to protect the environment. By being open, it can be adapted to the particular needs of a city or community. Furthermore, the communication network is free to access and will also serve other purposes and all citizens can expand and/or benefit from it.

## 5.2 Limitations and future research

This work focused on the system's concept. Further work is required to develop the hardware and software for a prototype. Three key areas of further development are identified: the development of the modular nodes, the implementation of the REST API and the development of the frontend application.

At the sensor level, this work focused on the most dynamic environmental parameter, the noise level. The study of the microphones allowed for a much clear view that not all modules are suitable for the intended purpose. Several examples can be found that provide sound pressure level metering with these modules, but their limited sensibility does not allow for accurate measurements. This was made very clear by the tests that were presented in this work. These tests show that quality modules must be used in order to collet this data with a minimal level of accuracy.

It is important to mention that the system is not intended to be regulation level accurate. However, if properly calibrated, it can provide a clear view of the noise levels in the city. The noise level data collection can be tailor made to the specifications of the client, that is, by using mean values per time period, reporting the maximum SPL level or analysing the sound waves by means of an FFT function. That said, the other environment sensors must be studied as well. The optimization of the messages is important, as the payload influences the airtime and, by consequence, the number of messages that can be exchanged between the nodes and the LoRaWAN gateways.

The implementation of the proposed REST API (webservice) should provide for different accesses to the system based on user profiles. The city officials will have access to configure and manage the system while the citizens will be able to access the environmental data.

This work also opens the possibility of creating Open Source nodes, that can be employed in buildings and their surroundings, for instance. Since the system is modular, this is also an opportunity for economic savings and to reduce the pressure on the environment. It can also allow citizens to deploy their own nodes and help to expand the system. The information captured by these nodes could pass through a filter for eliminating false data. The implementation of artificial intelligence algorithms can also be explored.

# References


[1] EURELECTRIC, 'Electricity for More Efficiency: Electric Technologies and their Energy Savings Potential', European Commission, Brussels, 2004. Accessed: Nov. 09, 2019. [Online]. Available: http://www.uie.org/sites/default/files/REPORT_FINAL_July_2004.pdf.

[2] Su-Chin Huang, Li-Ling Lee, Ming-Shan Jeng, and Yao-Ching Hsieh, 'Assessment of energy-efficient LED street lighting through large-scale demonstration', in *2012 International Conference on Renewable Energy Research and Applications (ICRERA)*, Nov. 2012, pp. 1–5, doi: 10.1109/ICRERA.2012.6477444.

[3] 'Instituto Português do Mar e da Atmosfera', 2020. https://www.ipma.pt/pt/educativa/observar.tempo/index.jsp?page=ema.index.xml (accessed Jun. 10, 2020).

[4] 'QualAR - Qualidade do AR', 2020. https://qualar.apambiente.pt/ (accessed Jun. 10, 2020).

[5] 'Município de Lisboa - Qualidade do Ar', 2020. https://www.lisboa.pt/cidade/ambiente/qualidade-ambiental/ar (accessed Jun. 10, 2020).

[6] 'Ruído - MUNICÍPIO', 2020. https://www.lisboa.pt/cidade/ambiente/qualidade-ambiental/ruido (accessed Jun. 10, 2020).

[7] ADAI - Associação para o Desenvolvimento da Aerodinâmica Industrial, 'Coimbra_Ln_A3.pdf', *Mapa de Ruído da Cidade de Coimbra, Ln*, Mar. 2011. http://apambiente.pt/_zdata/DAR/Ruido/SituacaoNacional/MapasRuidoMunicipais/Coimbra_Ln_A3.pdf (accessed Jun. 14, 2020).

[8] F. R. Pinto, M. Guedes, and M. J. Leite, 'Projeto-piloto de demonstração de mapas de ruído', Agência Portuguesa do Ambiente, Mar. 2004. [Online]. Available: https://apambiente.pt/_zdata/DAR/Ruido/NotasTecnicas_EstudosReferencia/Proj_Pil_Map_Ruido.pdf.

[9] Agência Portuguesa do Ambiente, 'Guia prático para medições de ruído ambiente - no contexto do Regulamento Geral do Ruído tendo em conta a NP ISO 1996', Agência Portuguesa do Ambiente, Oct. 2011. Accessed: Jun. 10, 2020. [Online]. Available: https://apambiente.pt/_zdata/DAR/Ruido/NotasTecnicas_EstudosReferencia/GuiaPraticoMedicoesRuidoAmbiente.pdf.

[10] Agência Portuguesa do Ambiente, 'Directrizes para Elaboração de Mapas de Ruído, Versão 3', Agência Portuguesa do Ambiente, Dec. 2011. Accessed: Jun. 10, 2020. [Online]. Available: https://apambiente.pt/_zdata/DAR/Ruido/NotasTecnicas_EstudosReferencia/DirectrizesMapasDez2011_todo_2.pdf.

[11] H. Ritchie and M. Roser, 'Urbanization', *Our World in Data*, Jun. 2018, Accessed: Jun. 11, 2020. [Online]. Available: https://ourworldindata.org/urbanization.

[12] '68% of the world population projected to live in urban areas by 2050, says UN', *UN DESA | United Nations Department of Economic and Social Affairs*, May 16, 2018. https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html (accessed Jun. 11, 2020).

[13] M. Deakin, Ed., *Smart Cities: Governing, Modelling and Analysing the Transition*, 1 edition. Abingdon, Oxon; New York, NY: Routledge, 2013.

[14] A. Caragliu, C. Bo, and P. Nijkamp, 'Smart cities in Europe', presented at the 3rd Central European Conference in Regional Science (CERS), 2009.

[15] A. Caragliu, C. D. Bo, and P. Nijkamp, ‘Smart Cities in Europe’, *Journal of Urban Technology*, vol. 18, no. 2, pp. 65–82, Apr. 2011, doi: 10.1080/10630732.2011.601117.
[16] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, ‘Internet of Things for Smart Cities’, *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014, doi: 10.1109/JIOT.2014.2306328.
[17] H. Bauer, M. Patel, and J. Veira, ‘Internet of Things: Opportunities and challenges for semiconductor companies | McKinsey’. https://www.mckinsey.com/industries/semiconductors/our-insights/internet-of-things-opportunities-and-challenges-for-semiconductor-companies (accessed Jan. 15, 2019).
[18] ‘State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating’, 2018. https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/ (accessed Jan. 15, 2019).
[19] ‘Best Fit Cases for LPWANs - White Paper from ABI Research’, *Ingenu*, 2016. https://www.ingenu.com/portfolio/best-fit-use-cases-for-lpwans/ (accessed Jan. 15, 2019).
[20] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, ‘A Study of LoRa: Long Range & Low Power Networks for the Internet of Things’, *Sensors*, vol. 16, no. 9, p. 1466, Sep. 2016, doi: 10.3390/s16091466.
[21] ‘Who We Are | Wi-Fi Alliance’. https://www.wi-fi.org/who-we-are (accessed Jan. 15, 2019).
[22] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, ‘LoRa and LoRaWAN testbeds: A review’, in *2017 IEEE AFRICON*, Sep. 2017, pp. 1496–1501, doi: 10.1109/AFRCON.2017.8095703.
[23] ‘Technology’, *Ingenu*, 2018. https://www.ingenu.com/technology/ (accessed Feb. 05, 2019).
[24] ‘About LoRaWAN™ | LoRa Alliance™’. https://lora-alliance.org/about-lorawan (accessed Jan. 15, 2019).
[25] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, ‘Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios’, *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, Oct. 2016, doi: 10.1109/MWC.2016.7721743.
[26] W. Ejaz *et al.*, ‘Internet of Things (IoT) in 5G Wireless Communications’, *IEEE Access*, vol. 4, pp. 10310–10314, 2016, doi: 10.1109/ACCESS.2016.2646120.
[27] ‘Sigfox Technology Overview | Sigfox’. https://www.sigfox.com/en/sigfox-iot-technology-overview (accessed Jan. 15, 2019).
[28] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, ‘Understanding the Limits of LoRaWAN’, *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, Sep. 2017, doi: 10.1109/MCOM.2017.1600613.
[29] ‘Home page | LoRa Alliance®’. https://lora-alliance.org/ (accessed Jul. 05, 2020).
[30] ‘Frequency Plans’, *The Things Network*, Jun. 09, 2020. https://www.thethingsnetwork.org/docs/lorawan/frequency-plans.html (accessed Jun. 11, 2020).
[31] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, ‘On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology’, in *2015 14th International Conference on ITS Telecommunications (ITST)*, Dec. 2015, pp. 55–59, doi: 10.1109/ITST.2015.7377400.
[32] T. Network, ‘LoRa World Record Broken: 832km/517mi using 25mW’, *The Things Network*. https://www.thethingsnetwork.org/article/lorawan-world-record-broken-twice-in-single-experiment-1 (accessed Jun. 11, 2020).

[33] T. Network, ‘LoRaWAN® distance world record broken, twice. 766 km (476 miles) using 25mW transmission power’, *The Things Network*. https://www.thethingsnetwork.org/article/lorawan-distance-world-record (accessed Aug. 24, 2019).
[34] ‘What is LoRa? | Semtech LoRa Technology | Semtech’. https://www.semtech.com/lora/what-is-lora (accessed Jan. 15, 2019).
[35] ‘The Things Network’, *The Things Network*. https://www.thethingsnetwork.org/docs/ (accessed Jan. 15, 2019).
[36] ‘LoRaWAN Architecture’, *The Things Network*, Jun. 09, 2020. https://www.thethingsnetwork.org/docs/lorawan/architecture.html (accessed Jun. 11, 2020).
[37] ‘2. How is sound measured?’ https://ec.europa.eu/health/scientific_committees/opinions_layman/en/hearing-loss-personal-music-player-mp3/l-3/2-sound-measurement-decibel.htm (accessed Feb. 14, 2020).
[38] T. S. Perry, ‘San Diego’s streetlights get smart’, *IEEE Spectrum*, vol. 55, no. 1, pp. 30–31, Jan. 2018, doi: 10.1109/MSPEC.2018.8241729.
[39] C. Mydlarz, C. Shamoon, and J. P. Bello, ‘Noise monitoring and enforcement in New York City using a remote acoustic sensor network’, Dec. 07, 2017. https://www.ingentaconnect.com/contentone/ince/incecp/2017/00000255/00000002/art00064 (accessed Jan. 15, 2019).
[40] Hammer Monica S., Swinburn Tracy K., and Neitzel Richard L., ‘Environmental Noise Pollution in the United States: Developing an Effective Public Health Response’, *Environmental Health Perspectives*, vol. 122, no. 2, pp. 115–119, Feb. 2014, doi: 10.1289/ehp.1307272.
[41] *Local Laws of the City of New York for the year 2005, no. 113*. 2005.
[42] ‘SONYC – Sounds of New York City’. https://wp.nyu.edu/sonyc/ (accessed Jan. 15, 2019).
[43] J. P. Bello *et al.*, ‘SONYC: A System for the Monitoring, Analysis and Mitigation of Urban Noise Pollution’, *arXiv:1805.00889 [cs, eess]*, May 2018, Accessed: Feb. 05, 2019. [Online]. Available: http://arxiv.org/abs/1805.00889.
[44] C. Mydlarz, J. Salamon, and J. P. Bello, ‘The implementation of low-cost urban acoustic monitoring devices’, *Applied Acoustics*, vol. 117, pp. 207–218, Feb. 2017, doi: 10.1016/j.apacoust.2016.06.010.
[45] N. Xie, ‘LoRa Technology: Remote and Real-time Acoustic Monitoring’, *Semtech*, p. 4, 2018.
[46] S. N. Sakib, T. Ane, N. Matin, and M. S. Kaiser, ‘An intelligent flood monitoring system for Bangladesh using wireless sensor network’, in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, May 2016, pp. 979–984, doi: 10.1109/ICIEV.2016.7760145.
[47] M. Ibrahim, A. Elgamri, S. Babiker, and A. Mohamed, ‘Internet of things based smart environmental monitoring using the Raspberry-Pi computer’, in *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, Oct. 2015, pp. 159–164, doi: 10.1109/ICDIPC.2015.7323023.
[48] J. C. Ferreira, J. A. Afonso, V. Monteiro, and J. L. Afonso, ‘An Energy Management Platform for Public Buildings’, *Electronics*, vol. 7, no. 11, p. 294, Nov. 2018, doi: 10.3390/electronics7110294.
[49] S. Mdukaza, B. Isong, N. Dladlu, and A. M. Abu-Mahfouz, ‘Analysis of IoT-Enabled Solutions in Smart Waste Management’, in *IECON 2018 - 44th Annual Conference of*

*the IEEE Industrial Electronics Society*, Oct. 2018, pp. 4639–4644, doi: 10.1109/IECON.2018.8591236.
[50] F. Folianto, Y. S. Low, and W. L. Yeow, 'Smartbin: Smart waste management system', in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Apr. 2015, pp. 1–2, doi: 10.1109/ISSNIP.2015.7106974.
[51] S. Zavare, 'Smart City Waste Management System Using GSM', vol. 5, no. 3, p. 5, 2017.
[52] S. S. Navghane, M. S. Killedar, and D. V. M. Rohokale, 'IoT Based Smart Garbage and Waste Collection Bin', vol. 5, no. 5, p. 3.
[53] S. V. Kumar, T. S. Kumaran, A. K. Kumar, and M. Mathapati, 'Smart garbage monitoring and clearance system using internet of things', in *2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, Aug. 2017, pp. 184–189, doi: 10.1109/ICSTM.2017.8089148.
[54] N. M. Yusof, A. Z. Jidin, and M. I. Rahim, 'Smart Garbage Monitoring System for Waste Management', *MATEC Web Conf.*, vol. 97, p. 01098, 2017, doi: 10.1051/matecconf/20179701098.
[55] G. Prajakta, J. Kalyani, and M. Snehal, 'SMART GARBAGE COLLECTION SYSTEM IN RESIDENTIAL AREA', 2015.
[56] M. Kalpana and J. Jayachitra, 'Intelligent Bin Management System for Smart City using Mobile Application', vol. 1, no. 5, p. 4, 2017.
[57] 'SWACHH : An Effective Real Time Solid Waste Management System for Municipality - Semantic Scholar'. https://www.semanticscholar.org/paper/SWACHH-%3A-An-Effective-Real-Time-Solid-Waste-System-Ghate-Kurundkar/efbd333fab608d7c875c05000ad24aa3bdb85428 (accessed Feb. 05, 2019).
[58] S. Thakker and R. Narayanamoorthi, 'Smart and wireless waste management', in *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Mar. 2015, pp. 1–4, doi: 10.1109/ICIIECS.2015.7193141.
[59] K. Mahajan and J. S. Chitode, 'Waste Bin Monitoring System Using Integrated Technologies', vol. 3, no. 7, p. 5, 2007.
[60] A. Papalambrou, D. Karadimas, J. Gialelis, and A. G. Voyiatzis, 'A versatile scalable smart waste-bin system based on resource-limited embedded devices', in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, Sep. 2015, pp. 1–8, doi: 10.1109/ETFA.2015.7301466.
[61] M. Al-Jabi and M. Diab, 'IoT-enabled citizen attractive waste management system', in *2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes Systems (IT-DREPS)*, Dec. 2017, pp. 1–5, doi: 10.1109/IT-DREPS.2017.8277804.
[62] C. J. Baby, H. Singh, A. Srivastava, R. Dhawan, and P. Mahalakshmi, 'Smart bin: An intelligent waste alert and prediction system using machine learning approach', in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2017, pp. 771–774, doi: 10.1109/WiSPNET.2017.8299865.
[63] M. Aazam, M. St-Hilaire, C. Lung, and I. Lambadaris, 'Cloud-based smart waste management for smart cities', in *2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, Oct. 2016, pp. 188–193, doi: 10.1109/CAMAD.2016.7790356.

[64] ‘Microphone sensitivity’, *Biamp Systems*, Mar. 25, 2014. https://support.biamp.com/General/Audio/Microphone_sensitivity (accessed Jun. 24, 2020).
[65] ‘APA - Políticas > Ruído > Normalização Aplicável’. https://apambiente.pt/index.php?ref=16&subref=86&sub2ref=530 (accessed Jul. 14, 2020).
[66] B. Guo, S. Song, A. Ghalambor, and T. R. Lin, ‘Chapter 17 - An Introduction to Condition-Based Maintenance’, in *Offshore Pipelines (Second Edition)*, B. Guo, S. Song, A. Ghalambor, and T. R. Lin, Eds. Boston: Gulf Professional Publishing, 2014, pp. 257–297.
[67] *Instrumentation Reference Book*. Elsevier, 2010.
[68] P. Malcovati and A. Baschirotto, ‘The Evolution of Integrated Interfaces for MEMS Microphones’, *Micromachines (Basel)*, vol. 9, no. 7, Jun. 2018, doi: 10.3390/mi9070323.
[69] G. Boré and S. Peus, *Microphones*. Georg Neumann GmbH, Berlin, 1989.
[70] ‘Dynamic Microphone: Moving Coil Microphone » Electronics Notes’. https://www.electronics-notes.com/articles/audio-video/microphones/moving-coil-dynamic-microphone.php (accessed May 17, 2020).
[71] ‘Home Page draft 1 – Master Mix Live’. http://mastermixlive.com/home-page/ (accessed May 18, 2020).
[72] J. Lewis, ‘Analog and Digital MEMS Microphone Design Considerations’, Analog Devices, Technical Article MS-2472, 2013.
[73] ‘defining_dBu.pdf’. Accessed: May 31, 2020. [Online]. Available: https://web.sonoma.edu/esee/courses/ee442/archives/sp2019/supp/defining_dBu.pdf.
[74] G. Feiertag, M. Winter, and A. Leidl, ‘Flip chip packaging for MEMS microphones’, *Microsystem Technologies*, vol. 16, pp. 817–823, May 2010, doi: 10.1007/s00542-010-1039-3.
[75] A. Industries, ‘Electret Microphone Amplifier - MAX9814 with Auto Gain Control’. https://www.adafruit.com/product/1713 (accessed Jan. 08, 2020).
[76] ‘The Things Network Airtime Calculator’. https://thethingsnetwork.org/airtime-calculator (accessed Jul. 31, 2020).
[77] Qoitech, ‘How Spreading Factor affects LoRaWAN device battery life’, *The Things Network*. https://www.thethingsnetwork.org/article/how-spreading-factor-affects-lorawan-device-battery-life (accessed Jul. 18, 2020).
[78] ‘Frequency Plans by Country’, *The Things Network*, Jun. 09, 2020. https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country.html (accessed Jun. 28, 2020).
[79] ‘en_30022001v020401o.pdf’. Accessed: Jul. 05, 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_en/300200_300299/30022001/02.04.01_40/en_30022001v020401o.pdf.
[80] ‘lm2903.pdf’. Accessed: Jan. 08, 2020. [Online]. Available: http://www.ti.com/lit/ds/symlink/lm2903.pdf.
[81] ‘KY-038 Microphone sound sensor module - SensorKit X40 Wiki’. http://sensorkit.en.joy-it.net/index.php?title=KY-038_Microphone_sound_sensor_module (accessed Jan. 08, 2020).
[82] ‘Keyes Microfone detecção de som Módulo Sensor de Arduino’, *Hobbyking*. https://hobbyking.com/pt_pt/keyes-microphone-sound-detection-sensor-module-for-arduino.html (accessed May 03, 2020).
[83] ‘lm386.pdf’. Accessed: Jan. 08, 2020. [Online]. Available: http://www.ti.com/lit/ds/symlink/lm386.pdf.

[84] ‘Keyes Brick-Sensor de Som Para Kingduino’, *Hobbyking*. https://hobbyking.com/pt_pt/keyes-brick-sound-sensor-for-kingduino.html?queryID=&objectID=44295&indexName=hbk_live_magento_pt_pt_products (accessed Jun. 27, 2020).
[85] ‘Me Sound Sensor · GitBook’. http://docs.makeblock.com/diy-platform/en/electronic-modules/sensors/me-sound-sensor.html (accessed May 20, 2020).
[86] ‘Me Sound Sensor V1’. https://www.funduinoshop.com/Me_Sound_Sensor_V1_en (accessed Jun. 27, 2020).
[87] A. Industries, ‘Electret Microphone Amplifier - MAX4466 with Adjustable Gain’. https://www.adafruit.com/product/1063 (accessed Jan. 23, 2020).
[88] ‘MAX4466 Low-Cost, Micropower, SC70/SOT23-8, Microphone Preamplifiers with Complete Shutdown - Maxim’. https://www.maximintegrated.com/en/products/analog/audio/MAX4466.html (accessed Jan. 23, 2020).
[89] ‘CMA-4544PF-W.pdf’. Accessed: Mar. 08, 2020. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/CMA-4544PF-W.pdf.
[90] ‘1063 Adafruit Industries LLC | Development Boards, Kits, Programmers | DigiKey’. https://www.digikey.pt/products/en?mpart=1063&v=1528 (accessed Jun. 28, 2020).
[91] ‘ph0645lm4h-b-datasheet-rev-c-1525723.pdf’. Accessed: May 20, 2020. [Online]. Available: https://pt.mouser.com/datasheet/2/218/ph0645lm4h-b-datasheet-rev-c-1525723.pdf.
[92] ‘I2SBUS.pdf’. Accessed: Jan. 22, 2020. [Online]. Available: https://www.sparkfun.com/datasheets/BreakoutBoards/I2SBUS.pdf.
[93] ‘3421 Adafruit Industries LLC | Development Boards, Kits, Programmers | DigiKey’. https://www.digikey.pt/products/en/development-boards-kits-programmers/evaluation-and-demonstration-boards-and-kits/787?k=adafruit%20i2s (accessed Jun. 28, 2020).
[94] A. Industries, ‘Adafruit I2S MEMS Microphone Breakout - SPH0645LM4H’. https://www.adafruit.com/product/3421 (accessed Jan. 23, 2020).
[95] A. Industries, ‘Adafruit Silicon MEMS Microphone Breakout - SPW2430’. https://www.adafruit.com/product/2716 (accessed Jan. 23, 2020).
[96] ‘Top Port SiSonicTM Microphone - SPW2430HR5H-B.pdf’. Accessed: Jan. 24, 2020. [Online]. Available: https://www.knowles.com/docs/default-source/model-downloads/spw2430hr5h-b.pdf.
[97] ‘MIC5225 - Power Management - Linear Regulators’. https://www.microchip.com/wwwproducts/en/MIC5225 (accessed Jan. 24, 2020).
[98] ‘2716 Adafruit Industries LLC | Development Boards, Kits, Programmers | DigiKey’. https://www.digikey.pt/product-detail/en/adafruit-industries-llc/2716/1528-1362-ND/5604375 (accessed Jun. 28, 2020).
[99] ‘Seeeduino LoRaWAN - Seeed Wiki’. http://wiki.seeedstudio.com/Seeeduino_LoRAWAN/ (accessed May 10, 2019).
[100] Semtech Corporation, ‘SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver - Datasheet’, Jan. 2019. https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001OKs/Bs97dmPXeatnbdoJNVMIDaKDlQz8q1N_gxDcgqi7g2o (accessed Feb. 03, 2020).
[101] ‘The Things Uno’, *The Things Network*, Dec. 12, 2019. https://www.thethingsnetwork.org/docs/devices/uno/ (accessed Dec. 18, 2019).
[102] ‘ESP32 Overview | Espressif Systems’. https://www.espressif.com/en/products/socs/esp32/overview (accessed Jul. 31, 2020).
[103] ‘LoRa SODAQ Board - ExpLoRer’, *SODAQ*. https://shop.sodaq.com/explorerrn2483a-eu.html?source=facebook (accessed Jul. 16, 2020).

[104] 'RAK811 LPWAN Evaluation Board', *RAKwireless Store*. https://store.rakwireless.com/products/rak811-lpwan-evaluation-board (accessed Jul. 25, 2020).
[105] 'Arduino - Software'. https://www.arduino.cc/en/main/software (accessed Mar. 23, 2020).
[106] 'Sound Meter:decibel meter,whisper & noise detector – Aplicações no Google Play'. https://play.google.com/store/apps/details?id=com.infostack.soundmeter&hl=pt_PT (accessed Apr. 03, 2020).
[107] D. I. S. Pigeon, 'The Ultimate White Noise Generator • Design Your Own Color'. https://mynoise.net/NoiseMachines/whiteNoiseGenerator.php (accessed Apr. 03, 2020).
[108] J. Lewis, 'Common Inter-IC Digital Interfaces for Audio Data Transfer', p. 3.
[109] 'ATSAMD21G18 - 32-bit SAM Microcontrollers'. https://www.microchip.com/wwwproducts/en/ATsamd21g18 (accessed Jan. 23, 2020).
[110] The Things Network, 'TheThingsNetwork.h', *GitHub*. https://github.com/TheThingsNetwork/arduino-device-lib (accessed Jul. 18, 2020).
[111] 'Frequency plans by Country', *The Things Network*, Jan. 22, 2019. https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country.html (accessed Jan. 15, 2019).
[112] 'ECMA-404 Standard: The JSON Data Interchange Syntax'. Accessed: Jul. 20, 2020. [Online]. Available: https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf.
[113] 'MQTT'. https://mqtt.org/ (accessed Jul. 20, 2020).
[114] IFTTT, 'IFTTT Platform'. https://ifttt.com (accessed Jul. 20, 2020).
[115] 'TTN Mapper'. https://ttnmapper.org/ (accessed Jul. 20, 2020).
[116] 'The Things Network Console'. https://console.thethingsnetwork.org/applications/add (accessed Feb. 23, 2020).
[117] 'Packet Forwarders', *The Things Network*, Feb. 18, 2020. https://www.thethingsnetwork.org/docs/gateways/packet-forwarder/ (accessed Feb. 24, 2020).
[118] The Things Network, 'Semtech UDP Packet Forwarder'. https://www.thethingsnetwork.org/docs/gateways/packet-forwarder/semtech-udp.html (accessed Jul. 20, 2020).
[119] The Things Network, 'TTN Packet Forwarder', *The Things Network*, Jun. 29, 2020. https://www.thethingsnetwork.org/docs/gateways/packet-forwarder/ttn.html (accessed Jul. 20, 2020).
[120] Semtech Corporation, 'Basic Station 2.0.5 June 2020 documentation'. https://doc.sm.tc/station/ (accessed Jul. 20, 2020).
[121] I. Fette and Melnikov, A., 'The WebSocket Protocol', Dec. 2011. https://tools.ietf.org/html/rfc6455 (accessed Jul. 20, 2020).
[122] H. Schuldt, 'Multi-Tier Architecture', in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 1862–1865.
[123] R. T. Fielding, 'Architectural Styles and the Design of Network-based Software Architectures', University of California, Irvine, 2000.
[124] 'RESTful Web Services', *W3schools*, Jan. 26, 2019. /restful-web-services/intro/ (accessed Jul. 15, 2020).
[125] 'World's Most Popular API Framework', *Swagger*. https://swagger.io/ (accessed Feb. 24, 2018).

# Annex A - Public presentations

## A.1 GDG DevFest Coimbra'18

Citing the organization, the DevFest Coimbra'18 is a technological conference that will be held in Coimbra, on November 24th. Organized by Google Developers Group (GDG) Coimbra, as part of a cycle of international Conferences annually promoted by Google, this event aims to bring the best of technology to the center of Portugal: Coimbra. It's the perfect synergy of technological innovation with the city known as the Portuguese capital of knowledge.

Focusing in mobile, web, design and hardware this conference combines talks by renowned speakers with technological stimulating workshops, providing a hands-on-learning opportunity.

This event took place in the 24$^{th}$ November and in a workshop, representing the TTN community of Coimbra, I presented the theory of operation and a practical example of configuring a gateway and a node device.

## A.2 Bot Olympics'19 Coimbra

BotOlympics is a robotics competition of the University of Coimbra. It has educational and formative purposes and has as the main goal to present the world of robotics to the most curious. This event took place in the Pólo II of the University of Coimbra, in the 21$^{st}$ to 24$^{th}$ of February 2019.

It's composed by several simultaneous challenges, evolving several academic degrees, during a weekend. Simultaneously, several workshops also take place and it was in one of these that representing the Coimbra community of TTN I presented the network, the theory of operation and how an application can be built, registered and used free of charge by using this network. A practical demonstration of how a gateway and a device can be configured and added to the network also took place.

## A.3 ArduinoDay2019@IPT

The ArduinoDay is a global and simultaneous event celebrated in several communities around the world. This was born in the Arduino community and celebrated by enthusiasts in a 24-hour event where demonstrations based on this open-source platform take place.

The participation in this event took place in the 18th March, in the Polytechnic Institute of Tomar and I presented the technology involved in TTN and how the IoT projects developed in the Arduino community can benefit from it.


Figure A-0-1 - Presemtation of TTN at the ArduinoDay'19 in Tomar

## A.4 Ciclo de Eventos - Dinâmicas para a Inovação

Promoted by ANI (Agência Nacional de Inovação), a number of initiatives about valuing knowledge for the National Strategy for the Intelligent Specialization were organized. These events took place throughout the country and were composed of two moments: knowledge transfer workshops and themed focus group meetings. The purpose of the workshops was to drive the National Network of Technological Infrastructures through the discussion of the challenges to the next decade. As for the focus group meetings the purpose was to present disruptive technologies that answer to challenges of today.

The event took place on the 6th February, in the IT – Telecomunications Institute, in Aveiro. I have presented how The Things Network and LoRaWAN can help in solving some of the challenges of today in the context of Smart Cities. I also participated in a technical session of questions and answers about the subject.

## A.5 Smart Planet

Smart Planet is a technology publication focused on Smart Cities, communication technologies, buildings, energy, security and others. It covers the challenges presented by a constant changing world and the initiatives that provide the citizens better services, better businesses, better working spaces and a lower economic and environmental cost. The key audience are the public and private decision makers, that are confronted with these challenges and search for a comprehensive and accessible way to understand the available solutions.

I've been interviewed in April for an article based on the community driven network based on TTN that is being developed in Coimbra and how it can help to develop IoT projects not only in Coimbra but in Portugal and the world. The article was published in Smart Planet on the 29th of April: "O poder da IoT nas mãos das Comunidades"[2] (The power of IoT in the hands of the communities).

## A.6 ENEI 2019

The ENEI is the national encounter of informatics students (Encontro Nacional de Estudantes de Informática) and is one of the biggest events in Portugal involving informatics students of all the colleges of Portugal. It is recognized by the quality and variety of subjects and activities as well as the contact between the participants and companies.

In this event, that took place in ISEC on the 14th April, I presented the community of TTN in Coimbra as well as others in the country and in the world, the theory of operation. I also provided contact with several devices where the students could experiment to code and to register using this technology.

## A.7 Critical Summer Camp

Organized by Critical Software, Critical Summer Camp is three-week event, almost fully remote, where the core value of "Engineering Ingenuity" is presented. This event is composed of a series of workshops and presentations where the most innovative technologies are showcased. 15 students from 6 institutions participate in this event.

[2] https://www.smartplanet.pt/news/iot-and-redes/o-poder-da-iot-nas-maos-das-comunidades

As part of this event, in the 30th of July, I presented the context of TTN, LoRaWAN, the theory of operation and best practices as well as a live demo of configuring an application from configuring the device to receiving the results in a database and through a REST API.

# Annex B – Hardware testing

For the purpose of comparing the similar devices a simple setup was prepared. This included a Seeeduino LoRaWAN, a breadboard and a set of the previously selected devices.

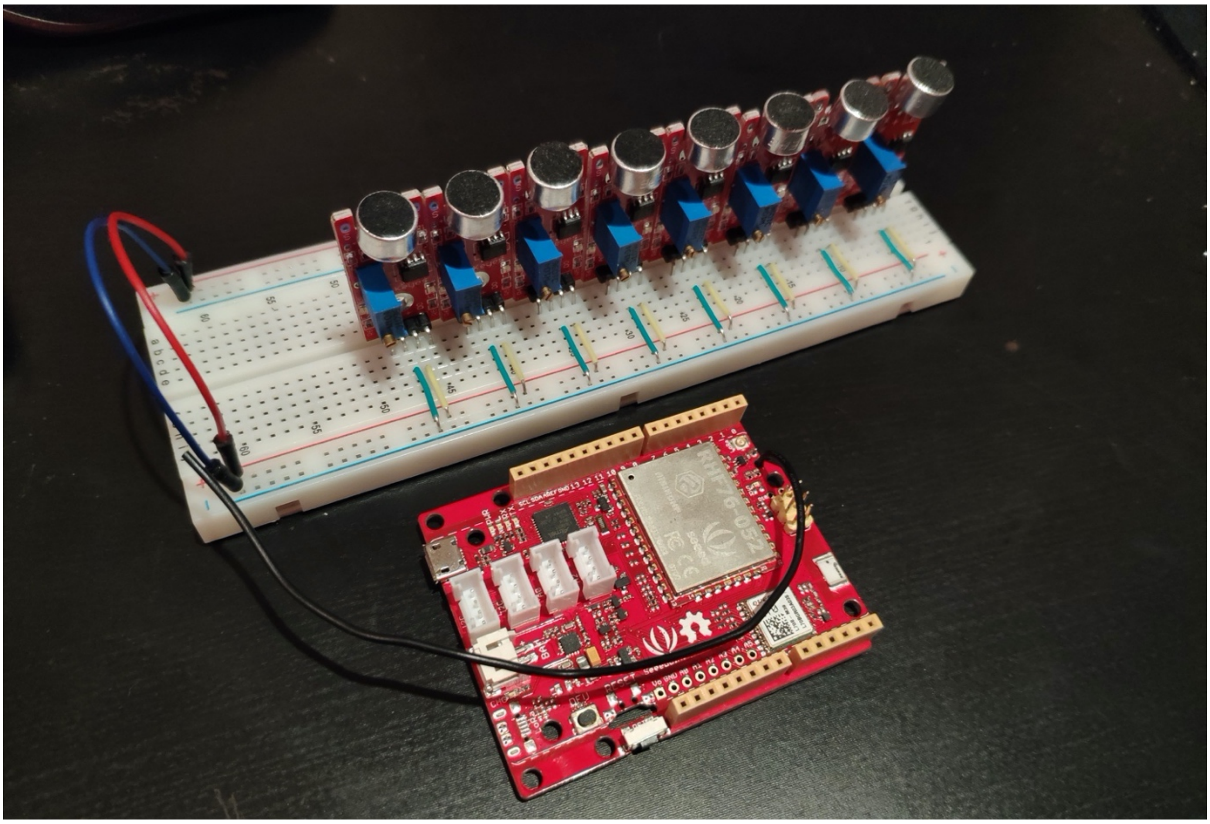

Figure B-0-1 - Prepared setup

Connecting the 6 devices to analogue ports A0 to A5 the output of the Arduino IDE presents a clear artifact in the readings from analogue input A5:

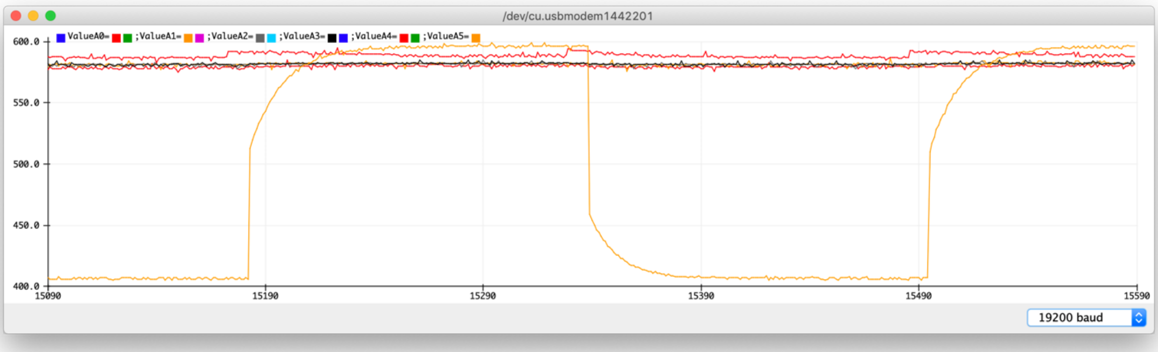


Figure B-0-2 – Serial Plotter output from Arduino IDE with all 6 channels connected

To isolate the problem to the controller board or to the sensor, the sensor was disconnected from the board and the result presents a clear digital signal at constant intervals:

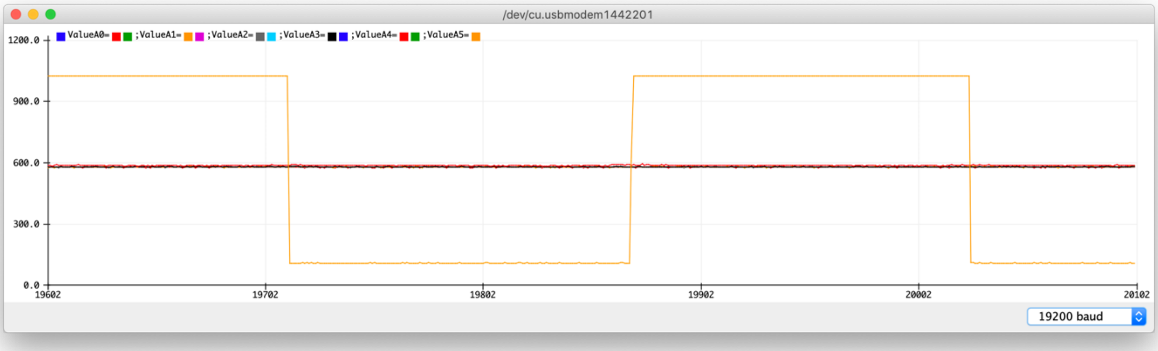


Figure B-0-3 - Serial Plotter output from Arduino IDE after disconnection of the A5 position device

Disconnecting all devices from the board, the output from the Arduino IDE serial plotter still presents the digital signal:

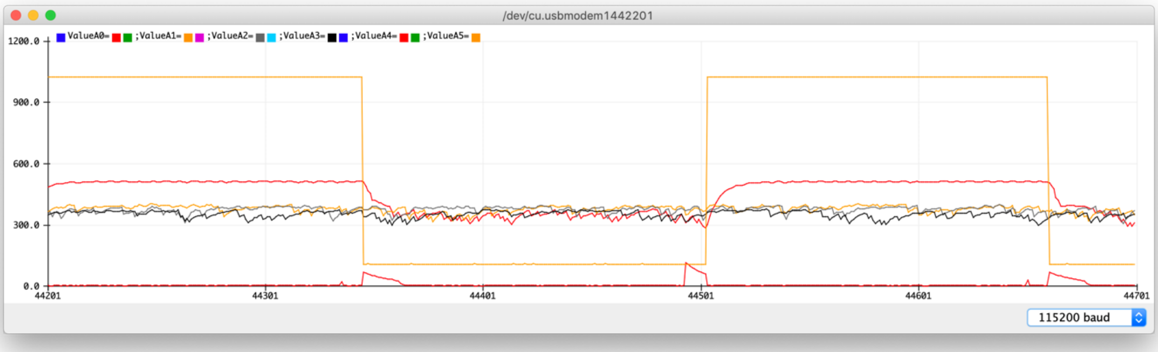


Figure B-0-4 – Serial Plotter output from Arduino IDE: noise readings in analogue inputs from A0 to A5

Clearly there is a signal present in the Seeeduino LoRaWAN analogue input A5. The datasheet of the mentioned development board indicates a battery charge status in A5. This can be also seen in the board schematic. After these observations, the A5 input port of the Seeeduino LoRaWAN cannot be used without a board modification. The analogue input A4 also doubles as a reading from the battery and further experimentation is required. Removing the reading related to the A5 analogue input the readings are more consistent:

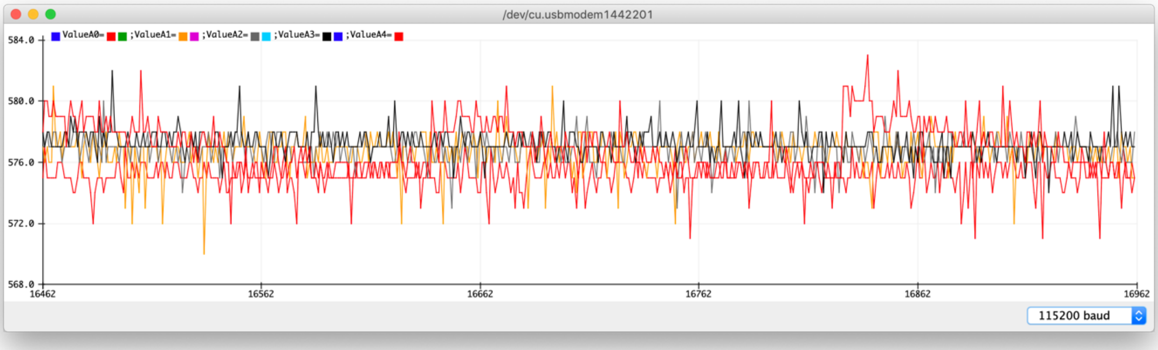


Figure B-0-5 - Serial Plotter output from Arduino IDE after removal of the reading from the analogue A5 input

After a closer inspection of the results, an artifact is still present. The readings from the analogue input A4 also present an artifact at regular intervals. This effect can be seen in the next figure:

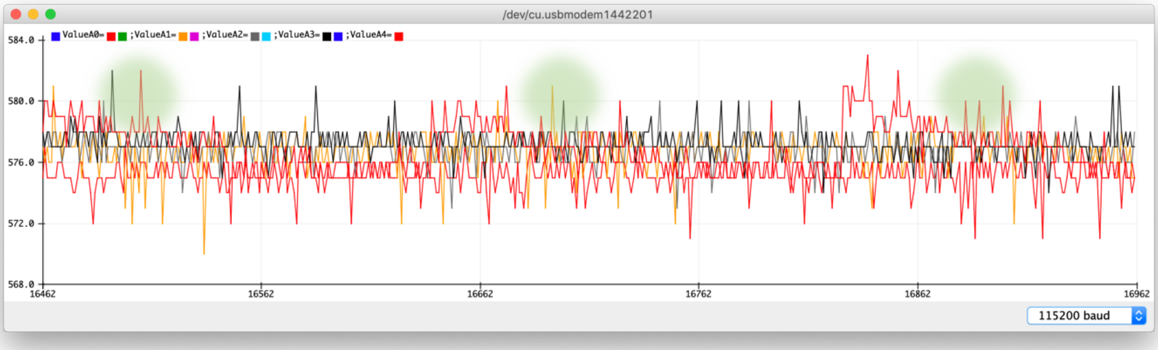


Figure B-0-6 - Serial Plotter output from Arduino IDE showing apparent artifact in readings from A4

Again, to isolate the problem, the connection to the analogue input A4 was removed. The result presents the artifact in the reading. The analogue port A4 reads 0 (instead of noise) and has an artifact at regular intervals. This was already present in Figure B-0-2.

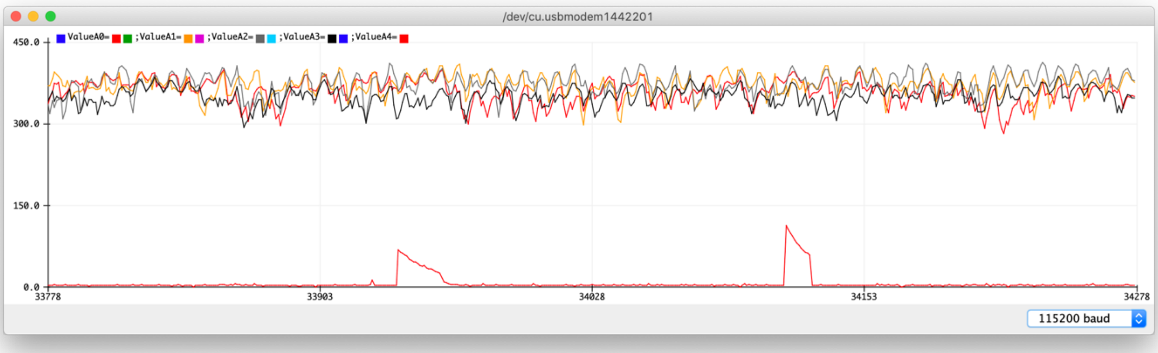

Figure B-0-7 - Serial Plotter output from Arduino IDE: noise readings in analogue inputs from A0 to A4 with no device connected to the A4 input.

There are therefore, 4 analogue ports available to receive data from the microphones. Using these 4 ports a more consistent result is obtained. The result has a significant amount of noise present.

Connecting all inputs (A0 to A3) to the same device allows for a view of the variation of readings between analogue inputs.

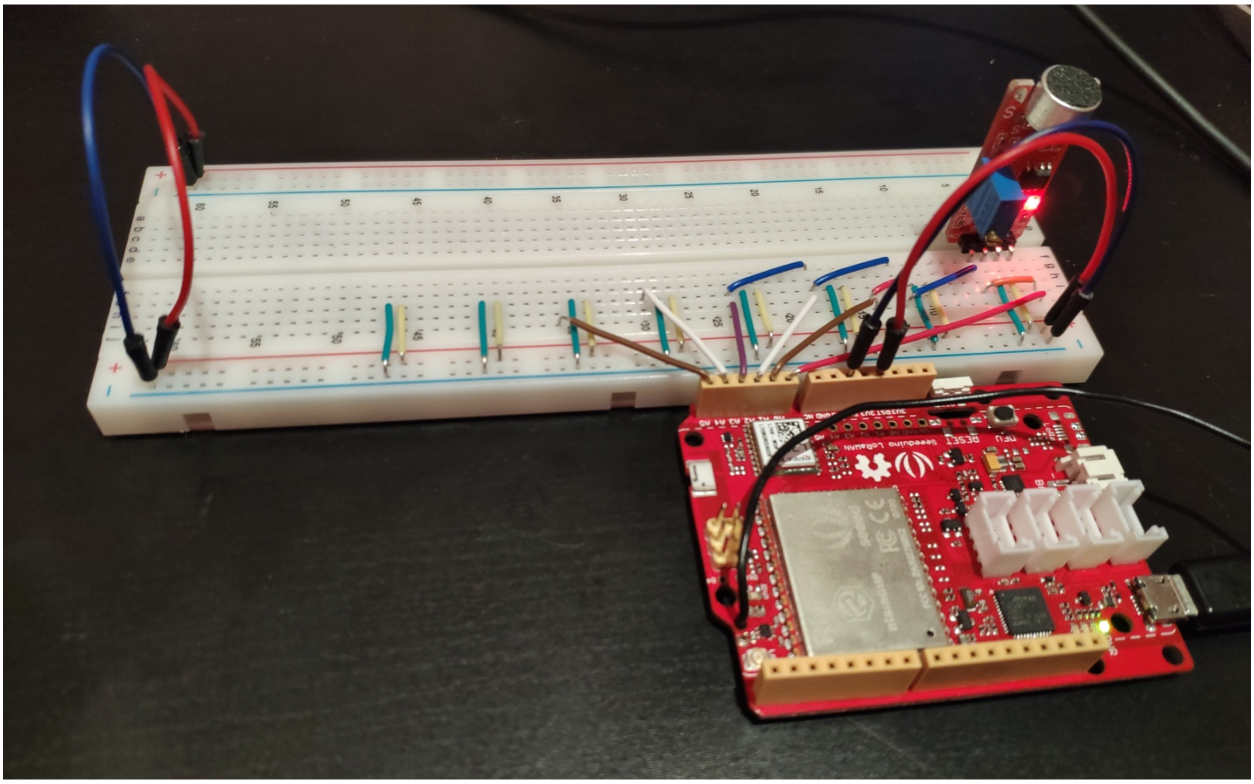

Figure B-0-8 - Modified setup, connecting analogue inputs (A0 to A3) to the same device

There are still some variations present in the output from the Serial Plotter of the Arduino IDE.

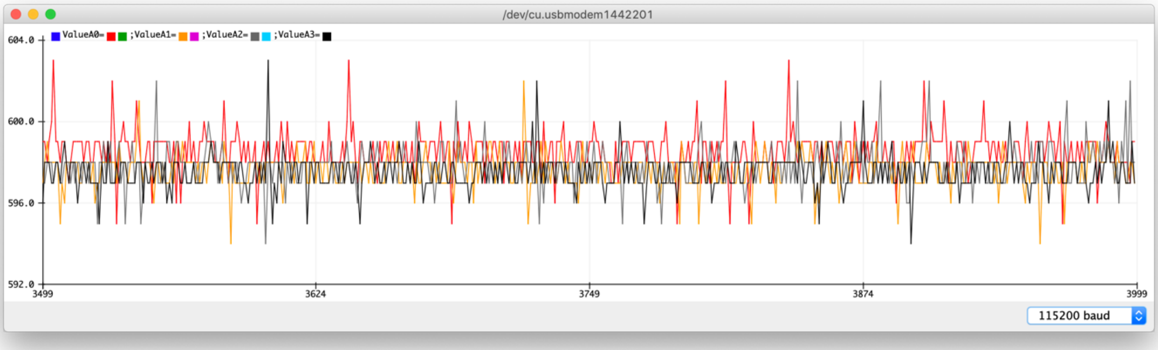


Figure B-0-9 - Serial Plotter output from Arduino IDE: readings from analogue inputs (A0 to A3) all connected to the same device

In order to determine the variation of the readings from the analogue inputs in the presence of different environmental noise levels a modification of the code was prepared. This code takes 10 000 readings and outputs the result of the maximum value recorded as well as the minimum and the difference (delta) between these two values. The setup was exposed to an environment of 30dB and 60dB and the results were recorded.

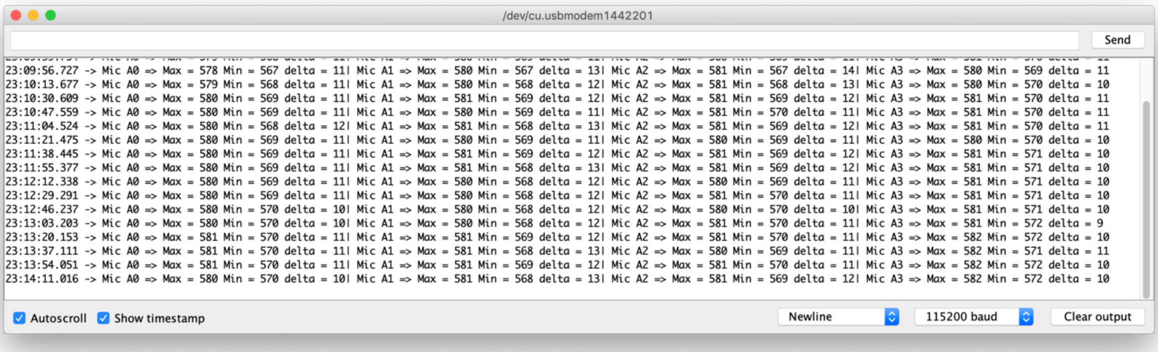


Figure B-0-10 - Measured values in a environment of around 30dB using 4 similar devices

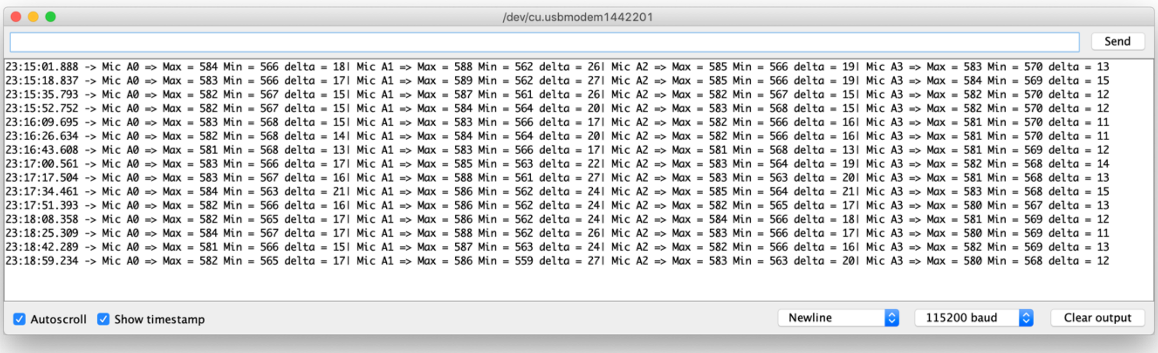


Figure B-0-11 - Measured values in an environment of around 60dB using 4 similar devices

These tests present small changes in the values recorded in the presence of this different environmental noise levels as well as significant differences between devices.

The same setup as before was prepared using the The Things Uno. This will allow to determine if changing the controller also changes the values recorded.

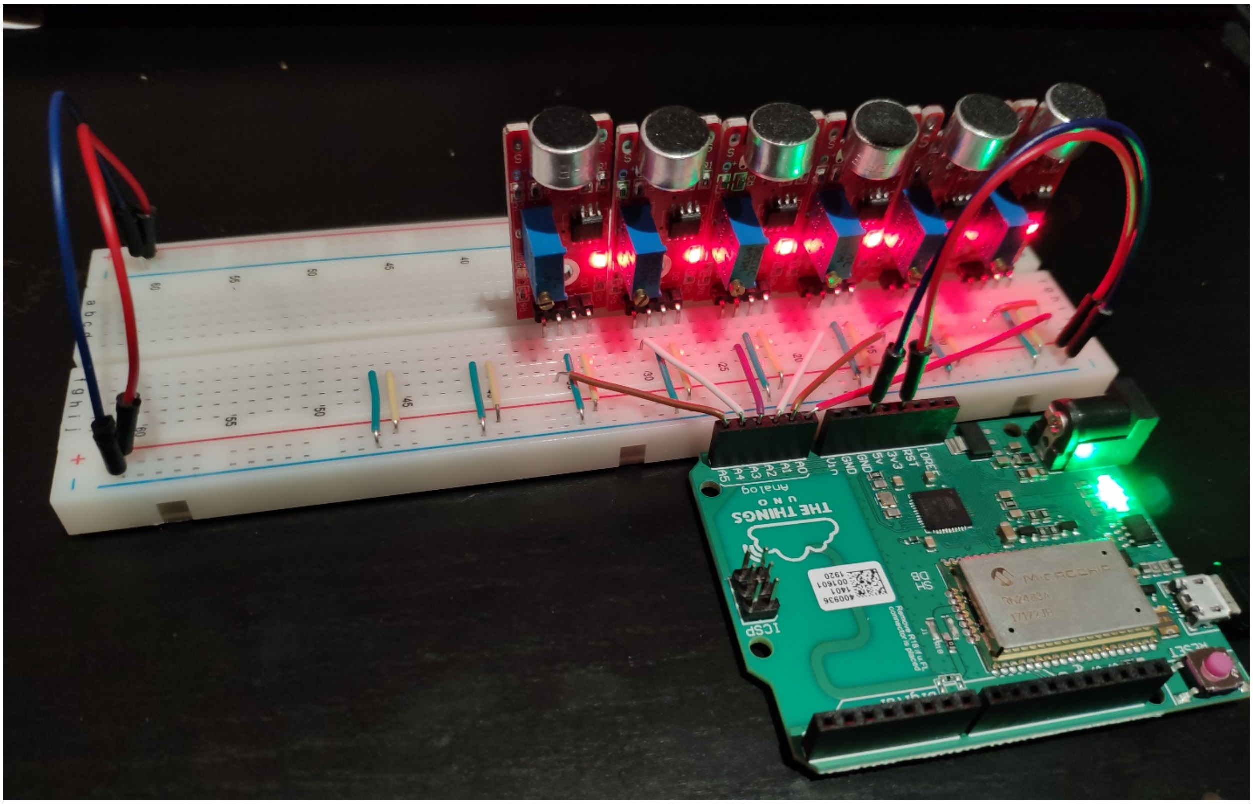

Figure B-0-12 - Setup using the The Things Uno

The first version of the program was run, reading the values captured from the analogue inputs A0 to A5. The results presented in the output from the Serial Plotter of the Arduino IDE show a more consistent result with less noise. There is a constant change in values in one or more inputs. Further testing is required.

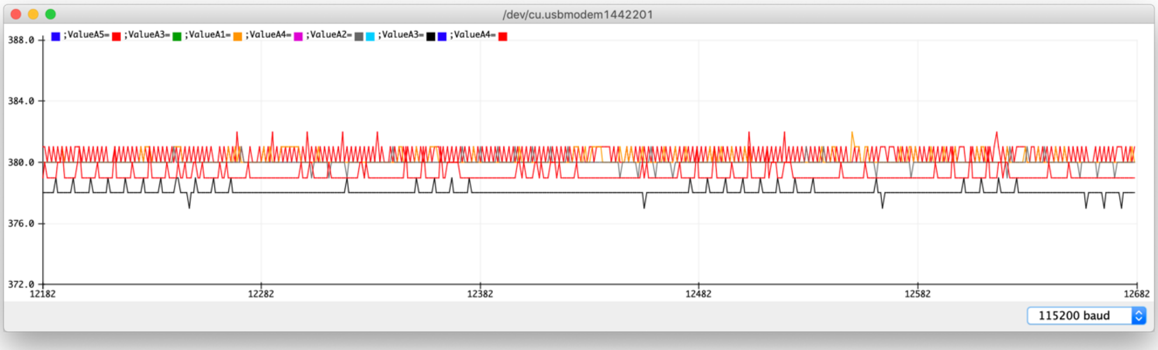


Figure B-0-13 - The Things Uno - Serial Plotter output from Arduino IDE with all 6 channels connected

Disconnecting all devices from the board, the results show a variation from 0 to 6, contrasting with the results from the Seeeduino LoRaWAN prototyping board. The same constant change in values mentioned before is present in the output.

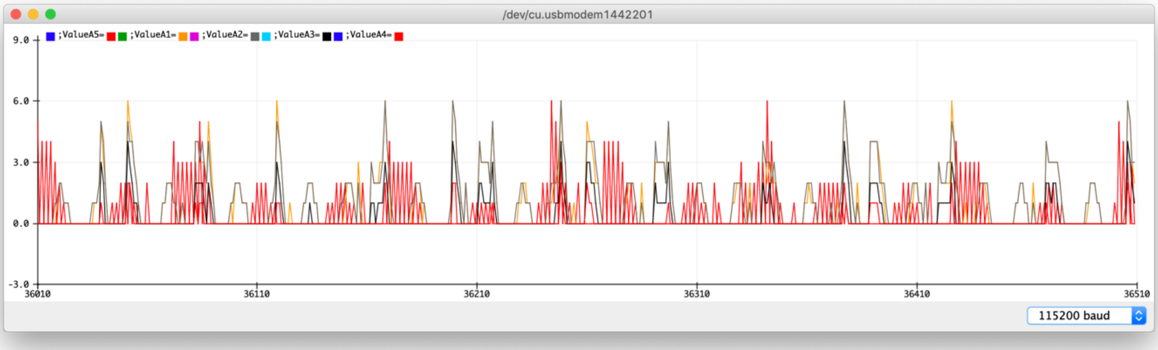


Figure B-0-14 - The Things Uno - Serial Plotter output from Arduino IDE: noise readings in analog inputs from A0 to A5

Connecting the devices and tapping in the sensor connected to the analogue input A0 the variation is even more noticeable. The same behaviour is present if the sensor tapped is the one connected to the analogue input A5. Tapping a device connected to the inputs A1 to A4 shows a different result.

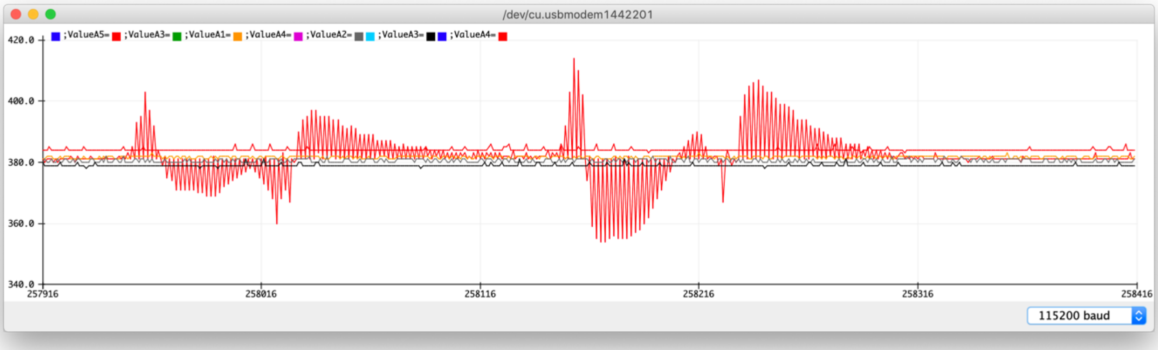


Figure B-0-15 – The Things Uno – Serial Plotter output from Arduino IDE with identical microphones connected to inputs A0 to A5 – tapping in microphone connected to A0

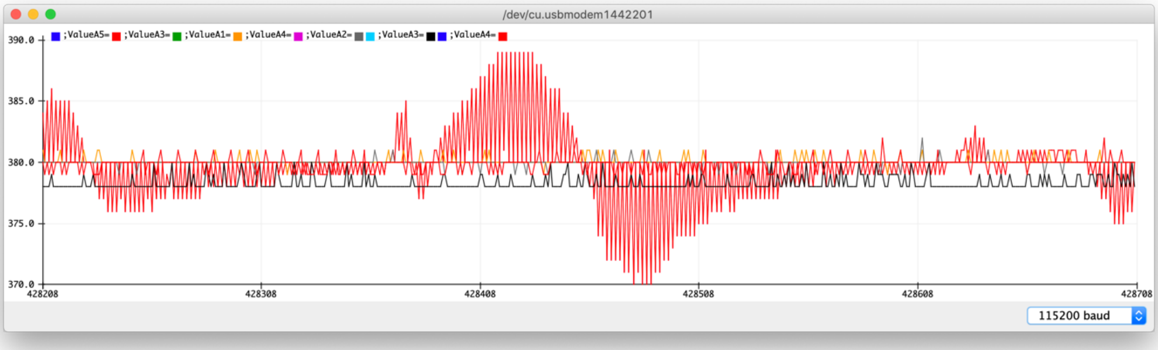


Figure B-0-16 – The Things Uno – Serial Plotter output from Arduino IDE with identical microphones connected to analog inputs A0 to A5 – tapping in microphone connected to A5

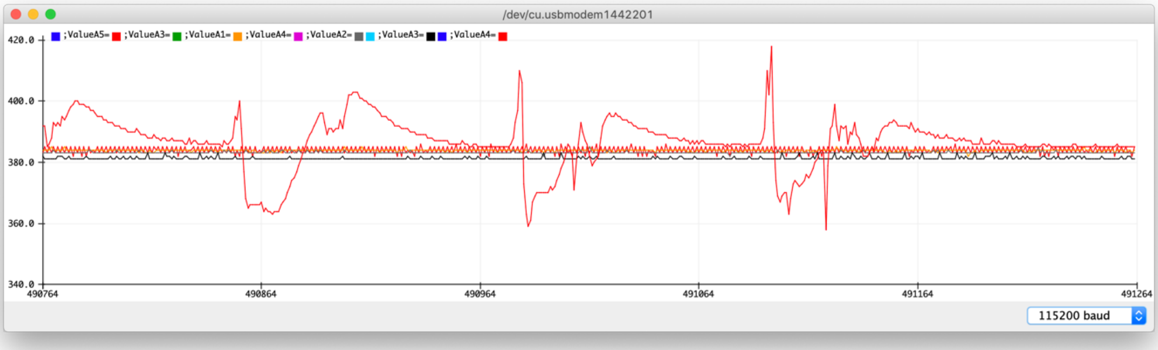


Figure B-0-17 – The Things Uno – Serial Plotter output from Arduino IDE with identical microphones connected to analog inputs A0 to A5 – tapping in microphone connected to A4

Using the second version of the code, for the inputs A1 to A4, the setup was exposed to an environment noise of 30dB and 60dB and the values recorded.

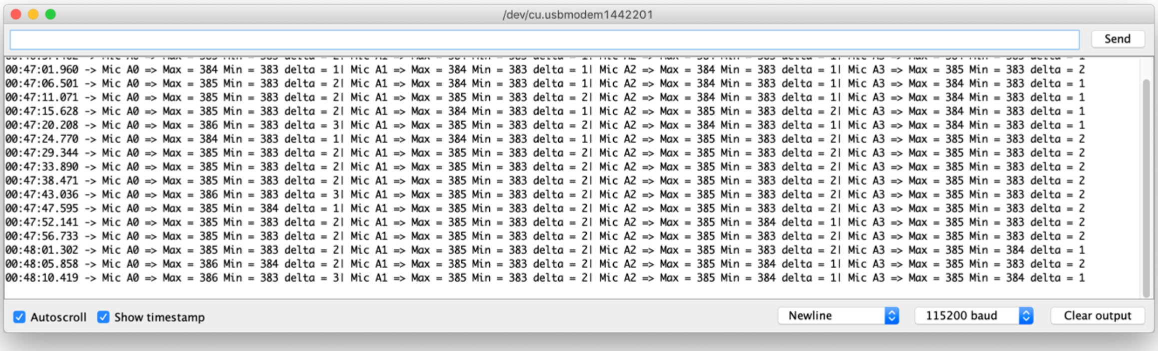


Figure B-0-18 – The Things Uno – measured values in an environment of around 30dB using 4 similar devices, bconnected to ports A1 to A4

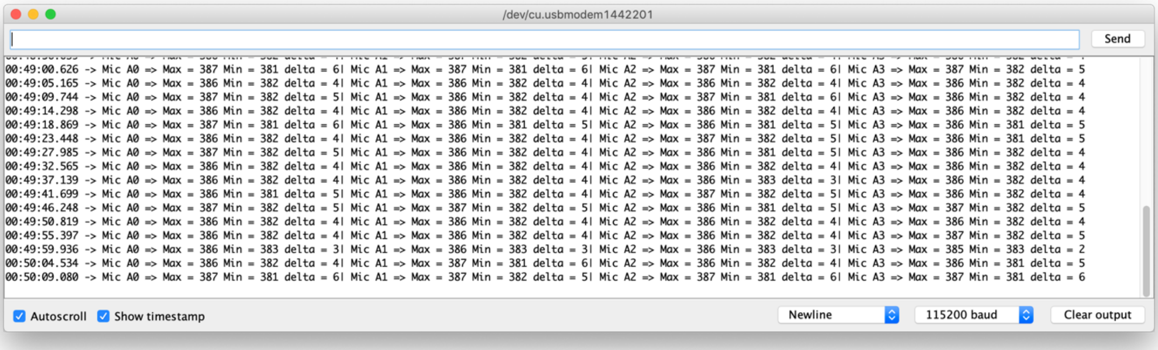


Figure B-0-19 – The Things Uno – measured values in an environment of around 60dB using 4 similar devices, connected to analog inputs A1 to A4


Figure B-0-20 – The Things Uno – measured values in an environment of around 30dB using one device, connected to ports A1 to A4

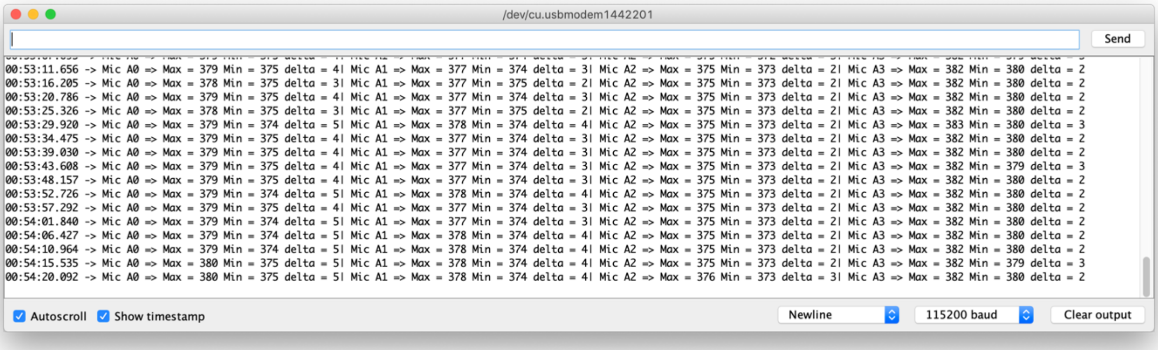


Figure B-0-21 – The Things Uno – measured values in an environment of around 60 dB using one device connected to analogue inputs A1 to A4

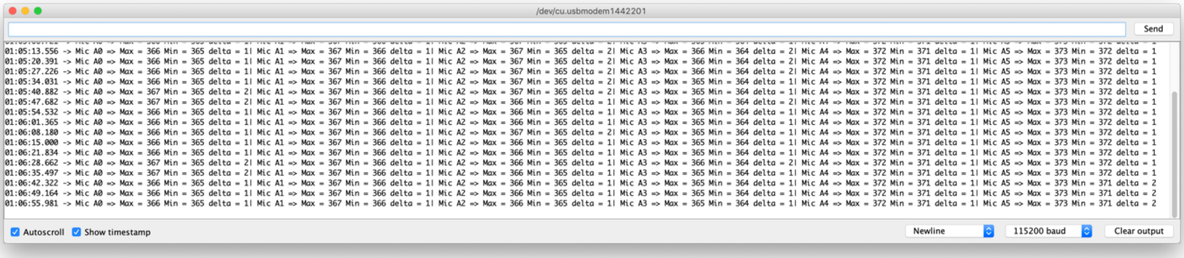


Figure B-0-22 - The Things Uno – measured values in an environment of around 30dB using one device, connected to analogue inputs A0 to A5

# Annex C – Code for testing analog microphones

```
void setup()
{
    SerialUSB.begin(baudRate);
    while(SerialUSB);            // wait for serial to be ready
    pinMode(mic, INPUT);         // set mic port to input
    delay(initialDelay);         // wait
}

void loop()
{
    // read mic input
    readMic = analogRead(mic);

    // update max input value
    maxInput = max(readMic, maxInput);

    // update min input value
    minInput = min(readMic, minInput);

    // calculate sum (for average calculation)
    sumInput += readMic;

    // set present time - initial delay
    time = millis() - initialDelay;
    SerialUSB.println(inputLabel + readMic + sampleLabel
      + sample + timeLabel + time);
    sample++;

    // at the end present the results
    if (sample == maxSamples) {

      // calculate average
      averageInput = sumInput/maxSamples;
      SerialUSB.println(maxInputLabel + maxInput + minInputLabel
        + minInput + averageInputLabel + averageInput);

      // stop
      while(true);
    }
}
```

# Annex D – Code for I²S microphone

```
/*
NOTE: Use board Arduino Zero (Native USB) instead of Seeeduino LoRaWAN
*/

#include <I2S.h>

const int baudRate = 115200;      // baud rate for USB serial port

const int mic = A2;               // analog port selected for mic
const int initialDelay = 5000;    // delay before initialization
const int maxSamples = 20000;     // max number of samples before stop

int sample = 0;
int time = 0;
int readMic = 0;
int samples[maxSamples];
int maxInput = 0;
int minInput = 1023;
int sumInput = 0;
int averageInput = 0;

const String inputLabel = "Input = ";
const String sampleLabel = " Sample = ";
const String timeLabel = " Time = ";
const String maxInputLabel = "\r\nMaxInput = ";
const String minInputLabel = "\r\nMinInput = ";
const String averageInputLabel = "\r\nAverageInput = ";

void setup()
{
    SerialUSB.begin(baudRate);                // initialize serial port
    while(SerialUSB);                         // wait for serial to be ready
    pinMode(mic, INPUT);                      // set mic port to input
    I2S.begin(I2S_RIGHT_JUSTIFIED_MODE, 44100, 32); // configure I2S for 44.1 kHz/32-bit sample
    delay(initialDelay);                      // wait

}

void loop()
{
    readMic = I2S.read();                     // read mic input
    if (readMic) {                            // filter 0 values
        maxInput = max(readMic, maxInput);    // update max input value
        minInput = min(readMic, minInput);    // update min input value
        sumInput += readMic;                  // calculate sum (for average calculation)

        time = millis() - initialDelay;       // set present time - initial delay
        SerialUSB.println(inputLabel + readMic + sampleLabel + sample + timeLabel + time);
        sample++;                             // increment sample number
    }


    if (sample == maxSamples) {               // at the end present the results
      averageInput = sumInput/maxSamples;     // calculate average
      SerialUSB.println(maxInputLabel + maxInput + minInputLabel + minInput
        + averageInputLabel + averageInput);
      while(true);                            // stop
    }

}
```