POLITECNICO DI TORINO

Master of Science

in Communication and Computer Networks Engineering

Master's Thesis

LoRa© Indoor application using an Arduino compatible industrial board

Supervisors

Prof. Ladislau Matekovits

Prof. Marco Allegretti

Candidate

Lizet Perez

2020 - 2021

ACKNOWLEDGMENTS

I would like to thank my family for the support they gave me during these years while working on my master’s. To my forever friends and colleagues who always believed in me and encouraged me to arrive at the end of this journey. To Politecnico di Torino for giving me the opportunity of achieving my dream. I would like also to thanks my supervisor professors L. Matekovits and M. Allegretti for introducing me to new technologies such as the Internet of Things and LoRa. Finally, I would like to thank professor I. Bordi for her guidance and support during the process of writing this thesis work.

SUMMARY

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Internet of Things is revolutionizing our world into a smarter place, it emerged as a concept 20 years ago and now is making the news around the world. This emerging technology connects billions of smart devices ("things" that are embedded with sensors, software, and other technologies) with other devices and systems over the internet.

According to Statista, an online portal providing data on the global digital economy, there are about 21.5 billion interconnected devices in the world and this number is expected to explode in the following years. By the end of 2021, it is predicted that the number of IoT devices will reach 46 billion, and by 2030 this number will reach 125 billion. Most of these smart devices will not be at home but in factories, hospitals, schools [1].

At present, global organizations are using IoT to boost productivity and efficiency, 72% of organizations have adopted IoT technologies into the workplace, from air conditioning and illumination systems to mobile devices. In chemical industrial sectors, IoT devices such as chemical sensors and picking systems have been used to decrease operational exposure. Also, healthcare companies are already using IoT devices, sensors are being used to monitor and maintain medical devices. To build smart cities of the future, governments are already connecting security systems in buildings, streetlights, and vehicles to create a coherent setting, being the most popular application of IoT the remote monitoring and control of devices within the city boundaries [2].

The success of IoT is completely dependent on reliable communication to the many IoT devices. This remains a huge obstacle across industrial and commercial industries. These industries are faced with complex or remote environments where traditional wired and wireless connectivity options cannot meet the cost, coverage, and power requirements needed for IoT applications. Cellular networks are too expensive and very power-intensive and Wi-Fi experiences problems with scalability, power consumption, and coverage when it comes to indoor and remote locations. Mesh networks cannot scale beyond medium-range applications because network setup and management are very complex. Bluetooth is designed only for short-range applications [3].

In this scenario, a new class of wireless technology emerged Low Power Wide Area Networks (LPWANs). LPWAN technology gained prominence as the preferred choice for IoT applications in 2015 when the GSMA wireless industry association defined a series of LPWAN standards to help network operators meet the specific cost, coverage, and power consumption needs of IoT applications [4].

Among LPWAN technologies, Long Range (LoRa), has attracted considerable attention from academia and industry. One of the most important characteristics of the LoRa technology is the long-range communication capabilities along with ultra-low power requirements, which makes this technology ideal for battery-operated devices. The LoRa technology was developed by Nicolas Sornin, Olivier Seller, and François Sforza together they started the company Cycleo, then acquired and patented by Semtech.

THESIS OBJECTIVES

In this work, we focused on the LoRa physical layer proprietary technology. The long-range and low-power characteristics of LoRa make it an interesting candidate for smart sensing technology in civil infrastructures (such as smart metering, environment monitoring, health monitoring, etc.) as well as in industrial ones (such as predictive maintenance, industrial automation, etc.). The goals of this thesis are two:

The first goal is to assess the Lora communication link subjected to interference caused by objects in an indoor environment. Interference weakens wireless signals and therefore is an important consideration when you are planning a wireless network.

The second objective aims to model the LoRa signal propagation in an indoor environment and evaluate which indoor propagation model suits best.

METHODS AND RESULTS

We carry out a Packet Loss Rate estimation, which is one of the simplest quality estimators, and an RSSI estimation which is very popular due to its low cost and its simplicity.

THESIS OUTLINE

This Thesis is structured as follows:

In Chapter 1 we provide a technical overview of LoRa giving special attention to its physical layer on which this study focuses. Then we presented major applications and adopted solutions of LoRa for the Industry field.

Some consolidated empirical propagation models are presented in Chapter 2. Among them, we consider Indoor propagation models which will be helpful to study the measurements obtained from the experiments in the indoor environment.

Then, the architecture implementation and configuration of boards and other tools for the experiments are shown in Chapter 3 and in Chapter 4 we present the steps we follow to carry out the experiments in an indoor environment.

Chapter 5 collects the results and analyzes the robustness of LoRa in the indoor scenario which can easily be an industrial environment. Finally, in Chapter 6 we present the conclusions.

CHAPTER 1

LORA PROTOCOL

Nowadays, many companies from different sectors are adopting the Internet of Things (IoT) to operate more efficiently. The growth of IoT devices has been remarkable but predictions expect that growth to further accelerate over the following years. Currently, IoT-connected devices are reaching 21.5 billion from and this number will continue growing. LPWAN technologies are the preferred choice for IoT applications.

Traditional wireless networks technologies are not considered the best convenient networks especially for the low mobility communication devices transmitting short messages. For this reason, LPWAN technologies are the preferred choice for IoT applications.

Due to the gap between cellular networks (like GSM and LTE) and short-range networks (like Wi-Fi, Bluetooth). There is a growing demand to design LPWAN technologies to fill the gap and to accomplish the requirements of IoT devices and applications.

LPWAN stands for Low Power Wide Area Network, it refers to a class of network technology designed to communicate wirelessly over long distances with low power consumption compared to other technologies such as Wi-Fi, satellite communications, and others. These two important characteristics made it suitable for the specific need of IoT devices and machine-to-machine devices M2M. LPWAN can improve the efficiency of IoT devices since it has economical low-power transceivers operating for long durations.

Wireless technologies can be classified based on their range, the range is often measured in terms of receiver sensitivity that is the lowest signal power for a message to be detected and demodulated. This higher receiver sensitivity is achieved by reducing the signal bandwidth or adding processing gain, both come at the cost of lower data rates. Also, the use of sub-GHz frequency bands in most LPWAN solutions further improves range and penetration capability. As the wavelength is inversely proportional to free space path loss, radio waves in sub-GHz systems can travel over Km in open areas [3]. Figure 1 presents a classification of wireless technologies based on their range and their power efficiency.

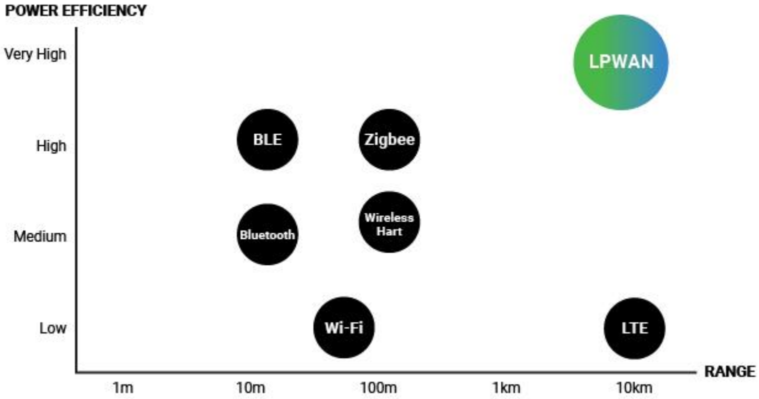


Figure 1. Wireless technologies based on range and power efficiency [3]

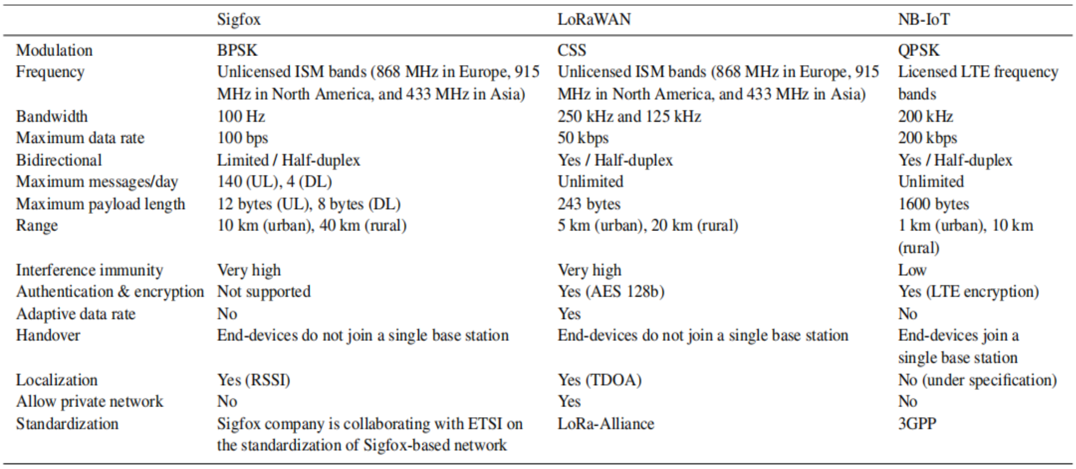
Bluetooth and Zigbee are known as Wireless personal area network (WPAN) ranges from 10 to 100 meters, Wi-fi is a Wireless local area network (WLAN) it ranges from 100 to 1000 meters. Finally, wireless wide area network (WWAN) with a range up to 100km, like cellular (2G,3G,4G LTE) and LPWAN.

LPWANs adopt various approaches to optimize power efficiency. The simplest approach is that transceivers can be put in sleep mode to consume minimal power. In bidirectional communications, transceivers can be awakened only at predefined times or as soon as after an uplink is sent to receive the downlink message. Moreover, many LPWANs employ lightweight asynchronous protocol at the MAC layer to minimize data overhead. Finally, the one-hop star topology introduces great power benefits, while mesh solutions like Zigbee and Wireless HART consume more power than LPWAN solutions. This is because, in a multi-hop mesh topology, the device must spend extra energy on listening for and relaying messages from other devices. On the other hand, devices in a star network stay most of the time in sleep mode [3].

Among LPWAN technologies, we consider Sigfox, LoRa, and NB-IoT as they are today the three leading LPWAN technologies.

As shown in Table 1, Sigfox and LoRa employ unlicensed spectra and asynchronous communication protocols while NB-IoT employs licensed spectrum an LTE- based synchronous protocol. Sigfox and LoRa cannot provide the same QoS provided by NB- IoT. This is the reason because NB-IoT is preferred for applications that require guaranteed quality of service [5]. NB- IoT technology is more costly considering the cost of the end device, network deployment, and the spectrum cost which is free for LoRa and Sigfox.

NB-IoT devices consume more energy because compared to the other two technologies because of synchronous communication, QoS handling and its OFDM/FDMA access mode require more current [5].



	Sigfox	LoRaWAN	NB-IoT
Modulation	BPSK	CSS	QPSK
Frequency	Unlicensed ISM bands (868 MHz in Europe, 915 MHz in North America, and 433 MHz in Asia)	Unlicensed ISM bands (868 MHz in Europe, 915 MHz in North America, and 433 MHz in Asia)	Licensed LTE frequency bands
Bandwidth	100 Hz	250 kHz and 125 kHz	200 kHz
Maximum data rate	100 bps	50 kbps	200 kbps
Bidirectional	Limited / Half-duplex	Yes / Half-duplex	Yes / Half-duplex
Maximum messages/day	140 (UL), 4 (DL)	Unlimited	Unlimited
Maximum payload length	12 bytes (UL), 8 bytes (DL)	243 bytes	1600 bytes
Range	10 km (urban), 40 km (rural)	5 km (urban), 20 km (rural)	1 km (urban), 10 km (rural)
Interference immunity	Very high	Very high	Low
Authentication & encryption	Not supported	Yes (AES 128b)	Yes (LTE encryption)
Adaptive data rate	No	Yes	No
Handover	End-devices do not join a single base station	End-devices do not join a single base station	End-devices join a single base station
Localization	Yes (RSSI)	Yes (TDOA)	No (under specification)
Allow private network	No	Yes	No
Standardization	Sigfox company is collaborating with ETSI on the standardization of Sigfox-based network	LoRa-Alliance	3GPP

Table 1. Overview of LPWAN technologies: Sigfox, LoRa, NB-IoT [5].

Therefore, cellular technologies like NB-IoT are more suited to situations where there is cellular coverage, and where the quality of service, low latency, and larger amount of data are important. While Sigfox and LoRa are more suited for scenarios where you have very small data to transfer, and you want to transmit over a long range with low power.

Additionally, LoRa allows you to set up a private network with ease. When using LoRaWAN protocol, it offers end-to-end device encryption AES 128b and a Class C enabling lower latency [6].

In conclusion, LoRa is suited for applications where small data is sent and also allows you to configure and manage your network. These characteristics make it suitable for an industrial application as well as many others. For this reason, LoRa was selected for the development of this thesis work.

1.1 LoRa Modulation Properties

LoRa is based on the Compressed High-Intensity Radar Pulse (chirp) spread spectrum (CSS) modulation. As a spread spectrum technique, LoRa CSS uses its entire bandwidth to transmit signals which make it robust to channel noise

The signal modulation in LoRa mainly relies on the chirp pulse signal, which is used to encode the information. A chirp signal refers to a signal with constantly increasing or decreasing frequency that sweeps across a predefined bandwidth B over a defined period Ts.

At first, the bits generated from the MAC layer are divided into subsequences of consecutive bits defined as symbols. The Spreading Factor (SF) represents the number of bits per symbol with $SF \in \{7, \dots, 12\}$. Hence the number of possible symbols is equal to $M = 2^{SF}$ [7].

Then to each m-th symbol, where $m \in \{0, \dots, M-1\}$, a chirp is associated. Chirps can be up-chirps when the frequency changes from highest to lowest and down-chirps when the frequency changes from lowest to highest. The bandwidth of the chirp is equal to the bandwidth B of the CSS signal [7]. In LoRa, the bandwidth B can assume the values $B \in \{125\ Mhz, 250\ Mhz, 500 Mhz\}$.

The chirp associated with the m-th symbol is obtained by applying a delay $\tau_m = \frac{m}{B}$ to a raw chirp. A raw chirp is a chirp up or down over an entire symbol period Ts.

The raw chirp outside the interval $\left[-\frac{T_s}{2}, \frac{T_s}{2}\right]$ is cyclically brought back into $\left[-\frac{T_s}{2}, -\frac{T_s}{2} + \tau_m\right]$ as shown in Figure 2 where $f_c(t) = \pm \frac{B}{T_s} t$ [7].

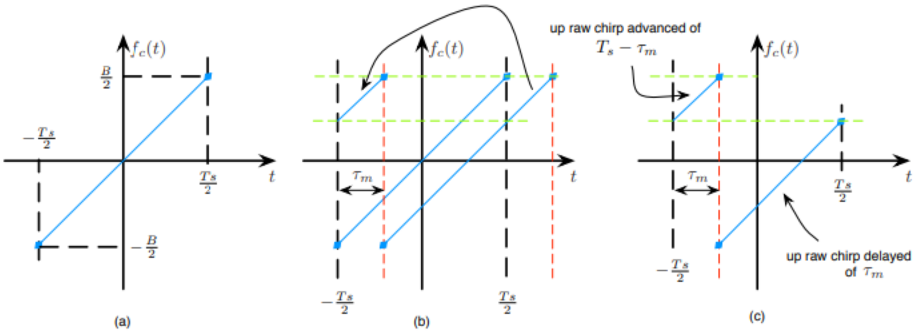


Figure 2. Symbol → chirp association process - (a) up raw chirp - (b) process illustration - (c) chirp associated with the m-th symbol [7].

Therefore, the modulated chirp related to the transmission of the m-th symbol can be written into two parts [7]:

1) for $t \in \left[-\frac{T_s}{2}, -\frac{T_s}{2} + \tau_m\right)$, the ramp of the raw chirp (up or down) advanced in time by $(T_s - \tau_m)$,

2) for $t \in \left[-\frac{T_s}{2} + \tau_m , \frac{T_s}{2}\right]$, the ramp of the raw chirp (up or down) delayed in time by τ_m

Moreover, the physical layer LoRa utilizes Hamming Code as forwarding error correction scheme which further improves the robustness of the transmitted signal at cost of redundancy [8].

The error correction scheme depends on the parameter known as Code Rate (CR). To every 4 bits, CR bits are associated with error correction.

Finally, the standard LoRa frame is shown in Figure 3. The B and SF are constants for a frame. The LoRa frame begins with a preamble that allows the receiver to synchronize to the data sent.

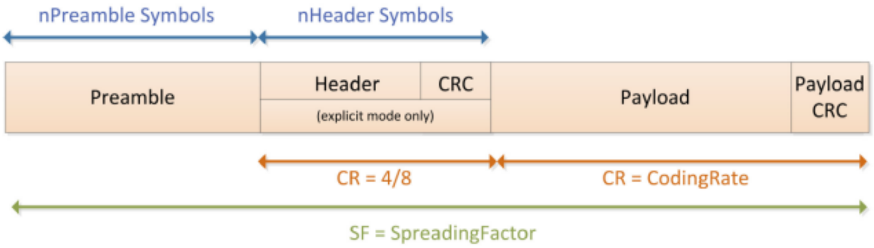


Figure 3. Standard LoRa frame [8].

As shown in Figure 4, the preamble starts with a sequence of constant up chirps that occupy the whole bandwidth. The last two up-chirps in the preamble define a sync word. This one-byte value is used to distinguish LoRa networks that use the same frequency bands. After the sync word, there are two and a quarter down chirps which have a duration of 2.25 symbols. The preamble can be configured to have a duration between 10.25 and 65,539.25 symbols [9].

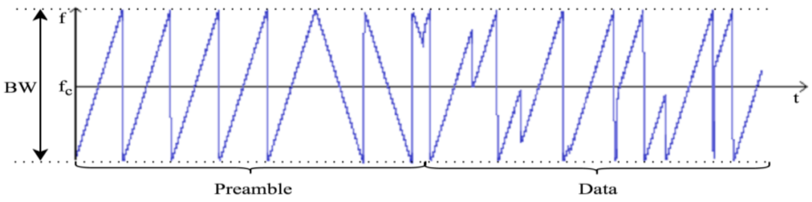


Figure 4. Frequency variation over time of a sample signal emitted by a LoRa transmitter. Fc is the central frequency of the channel, and BW is the bandwidth [9].

After the preamble, an optional header is transmitted with a code rate of 4/8 when present. It indicates the size of the payload in bytes which can be 255 bytes maximum. It also indicates the code rate, and whether a 16-bit cyclic redundancy check (CRC) for the payload is present at the end of the frame. At the receiver side, the CRC is used to discard packets with invalid headers. [9].

Received Signal Strength Indication (RSSI) is the received signal power in milliwatts but is measured in dBm. This value can be used to measure how well a receiver can hear a signal from the transmitter.

In the case of the LoRa signal, the typical LoRa RSSI values are -30 dBm if the signal is strong and -120dBm if the signal is very weak meaning that the closer this value is to 0 the better the signal is. Another parameter useful to evaluate the received signal is the Signal to Noise Ratio (SNR) which is the ratio between the receiver power signal and the noise floor power level. The typical LoRa SNR values are between -20dB and +10dB, when the SNR is positive means that the signal is operating above the noise floor and when the SNR is negative means that the signal is operating below the noise floor. However, LoRa can demodulate signals below the noise floor with SNR between the -7.5dB and -20dB [10].

1.2 LoRa and LoRaWAN

LoRa denotes the physical layer, or “bits “layer implementation as defined by the OSI Network model, as shown in Figure 5. It is a proprietary protocol owned by Semtech that uses the spread spectrum modulation technique derived from the Chirp Spread Spectrum (CSS). LoRa CSS can provide communication up to 3 km in dense urban areas and up to 10 km or more in rural areas. In addition, it provides also high-quality connections. The LoRa network management is open which means anyone can deploy LoRa networks.

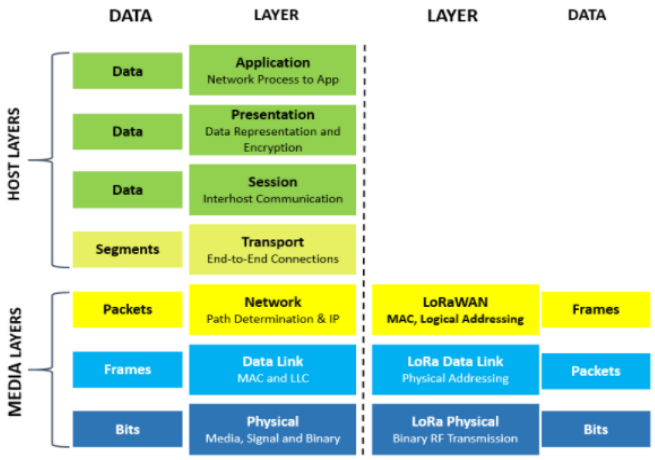


Figure 5. OSI seven-layer network model [11].

LoRaWAN is the most popular standard which is implemented by the LoRa alliance for lower power consumption and network device compatibility of LoRa terminals. It defines the communication protocol and the system architecture. More specifically it's a MAC layer protocol with some Network Layer components.

LoRaWAN networks use unlicensed frequencies which are specific and different for every country. This implies that you don't have to pay for data usage if you own a LoRaWAN network.

Compared to the proprietary LoRa, LoRaWAN software is open source. This means it's easy to build up your LoRaWAN network, and many vendors sell the network components like servers and gateways, all of which are interoperable.

LoRaWAN also allows you to have coverage almost anywhere in the world, you may find a service provider with infrastructure in that location or you can build the infrastructure by hand. However, to have the guarantee that your sensors will be connected to the network is better to build the infrastructure yourself.

Figure 6 shows (highlighted in yellow) the countries with LoRaWAN Network activity which include open communities, private networks, or networks driven by an operator. A network operator is any LoRaWAN network intended to openly monetize connectivity or end-to-end services to third parties. Network operators, who are members of the LoRa Alliance, are considered LoRaWAN Public Network Operators while LoRaWAN open communities are considered developer communities [12].

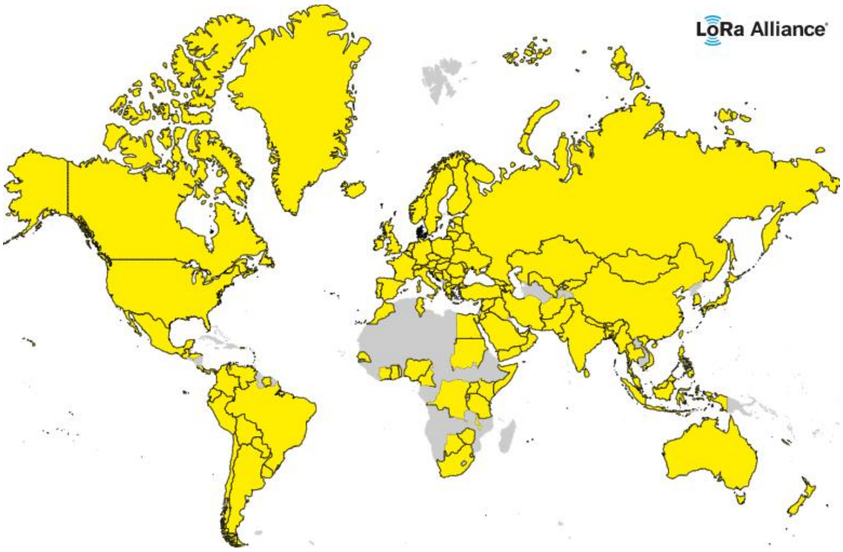


Figure 6. LoRaWAN global network coverage in 2021 [12].

1.3 LoRa Network Elements

The LoRa Network consists of end devices, gateways, and servers. Servers can be network servers, application servers, and joining servers. The network architecture is deployed in a star topology as depicted in Figure 7, at the center we find the network server which is in charge of managing the network and dispatch data to other servers. The gateways transmit messages between the end devices and the servers. They can have access to the network through a standard IP connection.

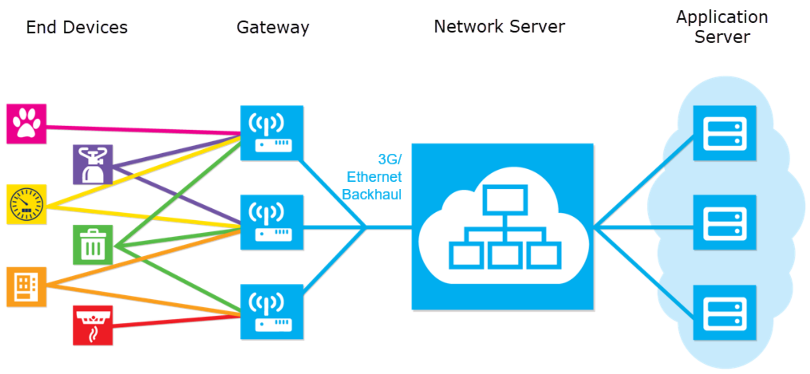


Figure 7. LoRaWAN architecture [13].

A very famous open community network is The Things Networks, commonly known as TTN, which is an open-source infrastructure that aims to provide free LoRaWAN network cover. The TTN is developed by many communities across the world.

A TTN free account can be created from the TTN website, this account allows you to create different TTN applications. A TTN application allows you to register devices, configure the settings for the application, etc.

The TTN network as shown in Figure 8, consists of end devices, gateways, a router, a broker, a handler, and the application servers. The router, the broker, and the handler are microservices.

The end device is a LoRaWAN enabled device, the gateway is in charge to receive the LoRa messages and sends them to the router using a very secure packet forwarder that allows remote configuration. The router gets the messages from the gateway and forwards them to a broker. Then, the broker identifies the device, deduplicates the traffic, and forwards the message to the handler where the application is registered. Finally, the handler encrypts and decrypts the payloads and publishes them to message brokers to be used by applications [14].

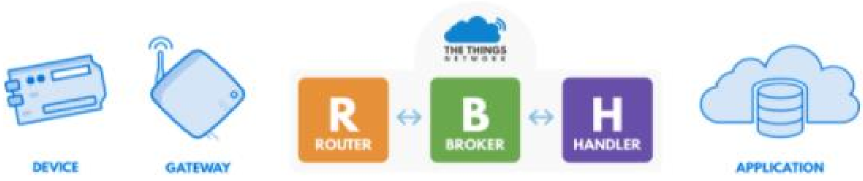


Figure 8. Key components of the things network [14]

1.3.1 End devices

End devices consist of sensors that are used to monitor the environment and generate the sensing data, a microcontroller to which the sensing data is sent, and an RF module that enables wireless communication.

Sensors can detect changes in the environment such as temperature, light, motion, and pressure. There are many types of sensors, and they have a variety of use cases. In Figure 9, a humidity sensor and an accelerometer are depicted. Humidity sensors measure the amount of power vapor in the atmosphere of air or other gases, they are used in conditioning, heating, vent systems. Another popular sensor is the accelerometer which can be used as anti-theft protection to alert if some object that should be static is moved.

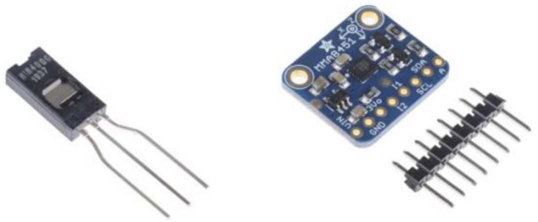


Figure 9. Honeywell HIH-4000-001 humidity sensor and Adafruit mma8451 accelerometer

The analogs signals produced by the sensors based on the observed phenomenon are converted to digital signals by an analog-to-digital converter and sent to a microprocessor for further processing.

The transceiver component links the node to the network. Typically, transceivers used in industrial scenarios operate in four different modes: transmit, receive, idle, and sleep. Power consumption in idle mode is equal to the power consumed in received mode. Thus, to save power transceivers should be turned off instead of running in idle mode when the transceiver is not transmitting or receiving.[15]

Several low-cost transceivers are available in the market, they can be controlled via SPI/UART connection or other standards. LoRa transceivers from brands such as Adafruit, RF Solutions, Microchip technology, and others, are based on Semtech chips that feature the patented modulation technique LoRa.

On the other hand, the LoRaWAN transceiver modules support LoRaWAN classes to optimize the balance between the network latency and battery life on end devices.

To each LoRaWAN end device is assigned an identifier when manufactured, these are used to guarantee the safe transport of packets over the network and to deliver encrypted data to the Cloud. The class of the device determines when it can receive downlinks as all LoRaWAN end devices can send uplinks anytime and it also determines the energy efficiency of the device [16].

Class A is the lowest power operating mode, as these end devices spend most of their time in sleep mode. A class A end device can send messages at any time to the servers, and every uplink is followed by two short downlink windows, allowing bidirectional communication. Communication is always started by the end device, and it's fully asynchronous. Differently to Class A end devices, Class B end devices use periodic beacons to synchronizes to the network and open downlink windows at scheduled times. This allows the network to send downlink communication with a deterministic latency, at cost of power consumption. Class C end device receiver is always open even if the device is not transmitting meaning a lot of power will be consumed. For this reason, Class C end devices require a constant power source rather than relying on a battery [16].

All LoRa end devices must be registered first before sending data to the network through a procedure known as activation. There are two methods of activation available:

- Over-The-Air-Activation (OTAA): Every device must perform a join procedure to join the network. Then a dynamic address is assigned to it and the security keys are discussed.

- Activation By Personalization (ABP): Configuration of encryption keys is done manually, there is no need to perform a handshake procedure to exchange keys like in OTAA. For these reasons, this method is less reliable than OTAA.

1.3.2 Gateways

Gateways are located in the local intranet that is also being used by the end devices. Both LoRa and LoRaWAN end devices can send messages to the gateways. A gateway consists of a microprocessor and an RF module and a LoRa module when using LoRa technology. They are useful to perform some storage and processing on the data received by the end device, thus end devices do not need to have a high processing power. When a gateway has access to the internet it can work as an access point that will allow the end devices to communicate with the cloud. Figure 10 shows a LoRa gateway for industrial purposes and another LoRa gateway with 4G backhaul.

Once it is connected to the internet the gateway can communicate with the LoRaWAN network. In the LoRaWAN network, each message sent by an end device is received by all gateways that can see it which will send them to the cloud. When these messages are received by the network server, it will select the gateway that received the message with the best RSSI, in this way it can communicate more efficiently with the end device.

Figure 10. LoRaWAN industrial rugged gateway and cloud cell 4G gateway

1.3.3 Servers

Gateways access to internet and send the data to the Servers in the network. In the LoRa network there are 3 types of servers: a Network server, Application servers and Join servers.

The network servers oversee the managing of the entire LoRaWAN network. It controls the network parameters dynamically to adapt the system which is always changing and establish secure 128-bit AES connection for the transport of end-to-end data which goes from the LoRaWAN end device to the end-users

Application in the Cloud as well as for the control of traffic that goes from the LoRaWAN end device to the Network Server and back.

It also guarantees the incorruptness of each message and the reliability of every sensor on the network.

Some functions of the Network Server are [17]:

- Over-The-Air-Activation (OTAA): Every device must perform a join procedure to join the network. Then a dynamic address is assigned to it and the security keys are discussed.
- Message deduplication: eliminates duplicate messages received by multiple gateways.
- Message routing: transfers payloads from the uplink application to the appropriate application servers, transfer downlink payloads coming from any application server to the end device, forwards Join-request and Join-accept messages between the devices and the join server [17].
- Adaptative Data Rate (ADR): ADR can optimize device power consumption and communication reliability. The ADR allows the network server to communicate to the end device to reduce transmission power or increase the data rate. End devices that are close to gateways should use a lower spreading factor and higher data rate, while devices further away should use a high spreading factor because they need a higher link budget [18].
- Acknowledgment of messages: it provides acknowledgment of confirmed data messages and some MAC commands.

The Application servers are responsible for managing application data messages received from devices and interpreting the collected data.

Finally, the Join Server manages the OTAA process for the end devices. It communicates the Network Session Key of the end-device to the network server and the Application Session Key to the corresponding application server. To perform these functions the join server must have the following information for each end device under its charge [17]:

- Dev EUI (end device serial unique identifier)
- AppKey (application encryption key)
- NwkKey (network encryption key)

- Application Server identifier
- End-device Service Profile

1.3.4 Security

The LoRa physical layer is a public protocol that works on an unlicensed frequency band, these two aspects make LoRa vulnerable to attacks.

For example, an attacker could overhear the address of a legal end device and generate false packets to the gateway. Also, an attacker could make use of a LoRa device to send maximum length packets to occupy the channel causing congestion.

On the contrary, LoRaWAN considers network security in its design. Security in LoRaWAN networks is based on the following key elements [17]:

- Frame encryption on uplink and downlink, the AES128 encryption scheme is used.
- Cryptographic keys used to encrypt frames are different for each device.
- Keys are never transmitted over the air so they cannot be intercepted by hackers.

The Join Server is in charge to derive Network Session keys and Application Session keys and providing them to Network Server and Application Server.

The session keys are computed when a device connects to the Network for the first time, the so-called Join Procedure. The Join procedure can be carried out by Over-The-Air-Activation (OTAA) or by Activation by personalization.

All these features ensure that only authorized devices can connect to the LoRaWAN network, the network traffic cannot be listened to, captured, and replayed, or altered.

1.4 Real-world deployments of LoRa

LoRa is ideal for a wide number of applications where long-range communication and very low power consumption are needed. Some applications of LoRa technology are Smart agriculture, Smart buildings, Smart Cities, Smart Metering, Smart Healthcare, Smart Industrial Control, Smart Retail, Smart Supply Chain, and Logistics [19].

- Smart agriculture: The long-range and low power characteristics of LoRa enable the use of low-cost sensors in a wireless network, which send data from the farm to the Cloud where it can be analyzed to improve operations. The National Narrowband Network Communications LoRaWAN wireless network covers a farm of one million acres in South Wales, sensors in the farm send real-time information about the soil moisture, rainfall, crops, water level, and livestock [20].

- Smart buildings: LoRa characteristics of low power and the capability to penetrating heavy building materials make it suitable for IoT-connected smart homes and building devices. Some applications are lighting control, temperatures remote control, HVAC control systems. LDT Inc. company integrated LoRa devices into its smart fire prevention system. Sensors and a network camera are used to detect flames, smoke, passive infrared, and dangerous changes in temperature in real-time [21].

- Smart cities: Many cities around the world are using LoRa technology to transform and enhance their services in different areas such as Health, Education, and Transportation. Amsterdam was the first city covered by the LoRa technology. Other cities around the world followed, cities such as Boston, Ottawa, Mexico City, Sao Paolo, Montevideo, Buenos Aires, Pekin, Manchester, and Zurich. Zurich uses IoT LoRa to determine places with forgotten bikes and zones with distribution grid power failure, and real-time occupancy rate monitoring in public transportation. Meanwhile, Amsterdam has developed projects using IoT focused on solving traffic-related problems by installing sensors that monitor the state of the traffic flow and parking availability. Such solutions have reduced the time of searching available parking lots by 43% [22].

- Smart metering: thanks to the ability of LoRa to penetrate deep indoors and a remarkable immunity to multipath and interference, utility companies have chosen LoRa as the technology allows the collection of data from water meters which are usually found in basements. Vision Metering, a leading developer of IoT utility metering solutions has incorporated the LoRa technology into its solution for advanced metering infrastructures (AMI) for water and gas. Vision's new solution enables the upgrade of legacy metering solutions to long-range AMI, for hourly or daily data transfer over LoRaWAN networks for efficient utility management and billing [23].

- Smart healthcare: LoRa's reliable performance and low cost make it suitable for critical smart healthcare applications. LoRa based sensors and gateways can monitor high-risk patients safeguarding their health and medical safety. Everynet's LoRaWAN-based network enables the real-time monitoring or runner safety at the annual Tor des Géants ultramarathon, providing network connectivity uncovered by Cellular networks [24].

- Smart retail: It refers to the hybridization between traditional shopping and modern technologies. EasyReach, a company that enables IoT applications for the retail and industrial domain, has developed a new energy monitoring solution based on LoRa devices and LoRaWAN protocol. The EasyPlug LoRa-based sensor deploys to the power supply cord of appliances in few minutes and starts collecting energy consumption data immediately, more information about the appliance is such as status (on/off), and location are also collected. A web-based platform creates reports on the collected data for trend analysis, this allows improving the energy efficiency of their applications. Additionally, automated alerts generated by the system notify managers of changes in equipment used to prevent theft, overuse, or abnormalities [25].

- Smart supply chain and logistics: LoRa's long-range, low power consumption and PS-free geolocation abilities allow vehicles, cargo, and other assets to be monitored over large zones and within harsh environments. Of, a Chinese bike-sharing company uses LoRa technology on its bicycles to track them, measure the demand for bicycles, and reduce costs [26].

1.4.1 Smart industrial networks

LoRa due to its outstanding low power capabilities is suited for the industrial automation world. LoRa end devices may monitor several industrial parameters such as pollution monitoring, fire detection, leakage detection, and so on. This technology is employed as a solution for many industrial applications around the world, some examples are listed below:

- In India, Easy Reach Solutions is a startup specializing in smart IoT solutions. It has incorporated LoRa into its industrial and smart vehicle monitoring products. Easy Reach's LoRa based application includes sensors for steam traps, concrete mixers, forklifts, diesel tankers, and trunks. The startup has over 600 deployments to date [27].

- Machine Max is a leading provider of smart solutions for fleet management, construction, and mining application. It has used LoRa technology to monitor machine status from anywhere on the construction or mining site. Data from the sensor is presented to site managers in real-time so that managers can use this data to identify problems in their area. The Machine Max solution can be easily deployed onto fleet machines in under a minute [28].

- Transco Industries Inc. has implemented LoRa into its conveyor belt applications for better monitoring and to reduce costs. This solution enables miners to monitor conveyor belts remotely, thus reducing maintenance costs and helping to avoid belt failure [29].

- AIUT, a hardware and software provider specializing in IoT- based solutions in the oil and gas markets, has incorporated LoRa devices and the LoRaWAN protocol into its new solution for measuring the volume of liquefied petroleum gas (LPG) in tanks. Thanks to the line of intelligent sensors, gas consumption can be monitored remotely and automatically improving the process of supply and reserve management for distributors to reduce operating costs [30].

- Advantech, a leading supplier of industrial communication solutions, has collaborated with Semtech Corporation to expand its line of wireless products with the launch of the Wzzard LRP node and SmartSwarm 243 gateway, both integrated with LoRa technology. The Wizzard LRP node is a highly integrated sensing platform for applications. The LoRa-based node allows the reduction of maintenance costs while increasing productivity. Meanwhile, the SmartSwarm 243 gateway can work in remote areas and harsh environments [31].

- Cisco Industrial Asset Vision is a simple solution by Cisco for remote resource management. It is equipped with a Cloud-based dashboard to better monitor and manage the condition of the resources. This solution can be deployed in minutes using a simple QR code to onboard sensors and devices. [32].

CHAPTER 2

EMPIRICAL PROPAGATION MODELS

Radio waves propagation can experience different path changes due to external environmental factors. For example, propagation characteristics in a forest will be different than in a desert since a forest presents many trees and bushes. Different weather conditions will also affect the propagation characteristics of the environment. The population density in the area is another important factor, a dense urban area will present different propagation characteristics from a rural area. All these factors make an accurate characterization of propagation radio waves to be difficult.

In radio network planning, propagation models also called path-loss calculation are used to model the environment. To calculate the path loss with maximum accuracy it is important to use a reliable and suitable propagation model as it will help to optimize cell coverage size, use the correct transmitted powers, and eliminate interference problems of other radio transmitters.

Performance metrics are used to evaluate the accuracy and efficiency of the models relative to the measured path loss. These metrics are the Mean Prediction Error (MPE) which gives the biasness of the predictive model, The Root Mean Squared Error (RMSE) which indicates the variance in the error, the Standard Deviation Error (SDE) which indicates the degree of deviation from the mean.

Propagation models can be roughly categorized into three types: stochastic, deterministic, and empirical.

Stochastic models are the least accurate as they require the minimum information about the environment. Therefore, mostly deterministic, and empirical models are used for modeling radio waves propagation.

Deterministic models estimate the field strength directly from the profile of the terrain between the transmitter and the receiver. These methods take account of the earth's curvature, free space losses, and losses due to diffraction. Thus, these models require detailed knowledge of the location, dimension, and parameters for every tree or building and terrain feature in the area to be covered. Examples of deterministic models are Ray tracing and Ray launching.

Empirical models are based on measurement data, they are mostly used by researchers to model a path loss equation as they are in general simple and fast in terms of computation with acceptable accuracy. Vast measurements and statistical analysis allowed to derive parameters for these models such as received signal strength, frequency, antenna heights, and terrain profiles. Therefore, they are only valid for specific environments (rural, urban, suburban, sea, etc.) and short ranges of frequency.

Since deterministic models require high computation time, in this thesis the focus has been directed towards empirical models to find a radio propagation suitable for prediction in an indoor environment.

2.1 INDOOR PROPAGATION MODELS

The Indoor environment model estimates the path loss inside a closed area, thus an area confined by walls. Features such as the layout of the building, the type of construction materials used, and the building type (office area, residential home, factory) strongly influence the propagation within the building. In general, radio waves in the indoor environment can be classified as either line of sight (LOS) or obstructed (NLOS).

Mechanisms such as reflection, diffraction, and scattering have to be considered when studying signal propagation in an indoor environment.

Reflections occur when a propagating wave impacts a very large object whose dimension is larger compared to the wavelength of the propagating wave. The electromagnetic waves during propagation usually encounter obstacles such as walls, doors, windows, pillars in the indoor environment, and depending on the obstacle surface the reflected signal suffers attenuation.

Diffraction happens when an impenetrable obstacle obstructs the signal propagation between the transmitter and receiver, for example, an obstacle made of metal or concrete materials. Propagating waves break on the edge of the obstacle, then they continue to propagate attenuated. Diffractions allow signals to propagate behind obstacles and are one of the main factors why signals are received where there is no line of sight between transmitter and receiver. Scattering occurs when the signal impinges on an object that has a rough surface causing the signal to scatter.

All these factors are called multiplicative noise. It is conventional to subdivide these factors as path loss, shadowing or slow fading, and fast fading or multipath fading.

Multipath propagation occurs as radio waves are reflected or diffracted or scattered by obstacles, radio waves establish various transmission paths from the transmitter to the receiver antennas. Many reflections are produced in an urban environment because of the obstacles the radio waves encounter. Figure 11 shows an example of multipath propagation.

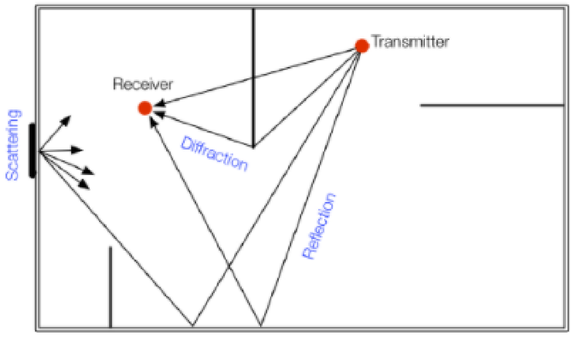


Figure 11. Indoor multipath radio propagation [33].

2.1.1 Free space model

Free space path loss is used to measure the decrease of the transmitted signal power due to spreading in free space environment. This spreading loss is the most important type of loss experienced by an electromagnetic wave when propagates. The free space model is given by the equation:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\Pi)^2\, d^2 L}$$

Where is the transmitted signal power, and are the antenna gains of the transmitter and the receiver respectively, L is the path loss and λ is the wavelength.

The free space model reveals that the path loss is a function of TX-RX separation when the transmitter and receiver are within the LOS range in a free space environment. The free space path loss equation provides valid results only if the receiving antenna is in the far-field of the transmitting antenna.

The log-normal path loss model is a generalization of the free space Friis equation. The model is expressed as:

$$p_r(d)_{dB} = \bar{p}_r(d_0)_{dB} - 10n\, log\left(\frac{d}{d_0}\right) + x\,, d > d_0$$

Where $\rho_r(d)_{dB}$is the received power at a distance d meters from the transmitters, n is the path loss exponent that defines the rate of decay of power with respect to distance, x is a Gaussian random variable with zero mean and variance defined in dB, $\bar{p}_r(d_0)_{dB}$ is the ensemble average over all possible received power values for a given reference distance denoted as d_0 meters. Distance d_0 should be small as possible while being in the far-field region from the transmitter. The values n and x are obtained from empirical data.

2.1.2 Dominant path loss model 1

The dominant path model was proposed to enhance the accuracy of a simple empirical model such as one slop which assumes a direct propagation path between transmitter and receiver not true for indoor environments. In the indoor environment, the transmitted signal may encounter obstacles such as elevators that cannot penetrate. In this situation, the contribution of received signal strength by transmission and reflection will significantly reduce, and most of the time the radio waves will reach the receiver by bypass diffraction. Thus, the received signal strength does not depend on the distance between transmitter and receiver but is related to the length L of the dominant path [34].

Due to the multipath effect, all the received rays are superimposed at the receiver to reconstruct the signal. Some of them will carry most of the signal energy since their path does not penetrate many obstacles that will attenuate the signal while the others will be highly attenuated due to the number of walls penetrated. The main idea in the dominant path is to approximate the rays that carry most of the signal energy and replace them with one dominant path.

The dominant path loss model is expressed as follows:

$$P(L) \quad = P \quad (d_0) \quad - 10n\, log\left(\frac{L}{d_0}\right)$$

Where the $P(L)$ is the received signal strength of users when the length of the dominant path is L. And $P \quad (d_0)$ is the received signal strength of users when the length of the dominant path is d_0 (d_0 *is equal to* $1m$). The parameter n is the path loss index which depends on the indoor environment [34].

2.1.3 Dominant path loss model 2

The attenuation of the level of signal strength is different when the signal propagates in the line-of-sight environment and non-line-of-sight (NLOS) environment. Therefore, the LOS propagation and NLOS propagation in this model is considered to improve the accuracy of the dominant path model [34].

The Dominant path model 2 is described as follows:

$$p(L) = \begin{cases} p(d_0) - 10n_{LOS}\, log\left(\frac{L}{d_0}\right) \\ p(d_0) - 10n_{NLOS}\, log\left(\frac{L}{d_0}\right) \end{cases}$$

Where n_{LOS} is the path loss index in the LOS environment and n_{NLOS} is the index path loss in the NLOS environment. The other parameters are the same as in the dominant path loss 1 [34].

Figure 12 shows an indoor environment divided into two separate parts. First, a room part which includes three small rooms and three big rooms. This room part is approximated the LOS environment in which the path-loss index. The second part is the corridor part where the elevator is a bigger obstacle made up of metal and concrete materials and cannot be penetrated by radio waves. Thus, the corridor part is a typical NLOS environment in which the path-loss index. Radio waves cannot penetrate the thicker concrete walls that is why all the radio waves from AP reach the sampling points R1and R2 by bypass diffraction, that is, the radio waves firstly reach the point S1, then reach the point S2, and finally arrive at points R1 and R2 [34].

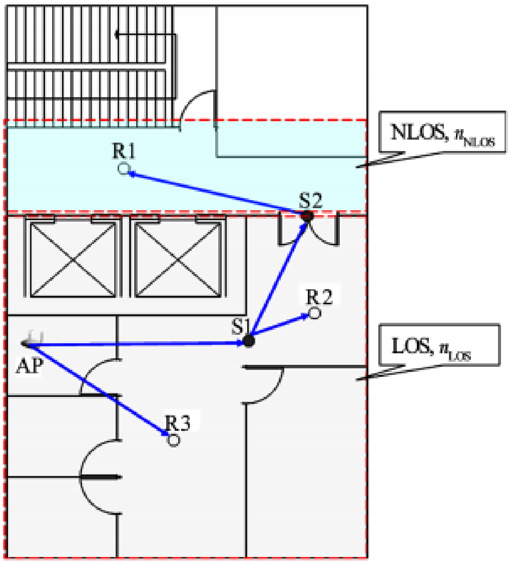


Figure 12. Dominant path 2 in indoor scenarios [34].

2.2 OUTDOOR PROPAGATION MODELS

In outdoor propagation models, radio waves propagate over uneven ground. This uneven terrain profile may vary from a simple curved earth profile to a very mountainous profile. The propagating radio waves may encounter obstacles such as buildings, trees, cars and so on which must be considered for the path loss computation. As for indoor environments, the outdoor environment is dominated by factors such as diffraction, reflection, and scattering. The communication between TX-RX depends on the propagation environment where the transmitted signal wave may be reflected, diffracted, or scattered and multiple copies of the signal may arrive at the receiver with different amplitude, phase creating constructive or destructive interference (multipath propagation). When the line-of-sight path is blocked by obstacles as shown in Figure 13, then the signal is arriving at the receiver by non-line-of-sight (NLOS) propagation since is no longer a direct path.

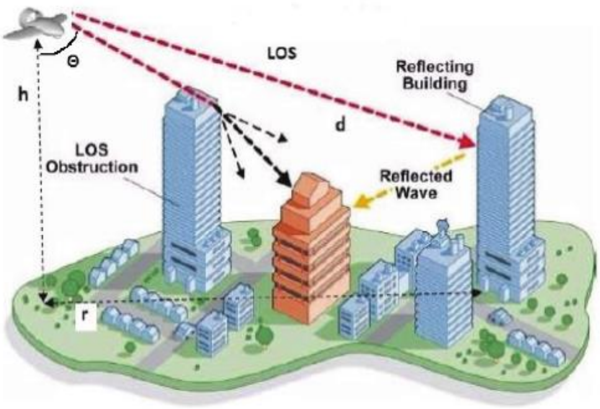


Figure 13. Example of LOS and NLOS links [35].

This thesis project focuses on signal propagation in indoor environments still some relevant outdoor models will be also mentioned below.

2.2.1 Okumura-Hata model

Is an empirical propagation model used to predict the path loss in outdoor environments.

The model was made by Okumura in Tokyo in 1968. Okumura's measurements were later incorporated into a mathematical formula by Hata.

This model is valid for frequencies from 150 to 1500 MHz and base station antenna height is between 30m to 200 m, the mobile station height from 1 to 10 meters, and the distance from 1 to 20Km.

The Okumura-Hata formula for urban areas follows:

$$L_P = 69.55 + 26.16 \log f \quad -13.82 \log h_T - a(h_R) + (44.9 - 6.55 \log h_T) \log d \quad [dB]$$

where f is the frequency (in MHz), h_R is the height of the base station (in m), $h_R$is the height of the mobile terminal (in m), $a(h_R)$correction parameter due to area type and d is the distance (in Km)

The antenna correction factor $a(h_R)$ is given by:

$a(h_R) = 3.2[\log_0(11.75h_R)]^2 - 4.97$ for big cities.

$a(h_R) = [1.1 \, log\, f - 0.7]h_R - [1.56 \, log\, f - 0.8]$ for small-medium cities

Okumura-Hata for suburban areas and rural areas are given by L_{path} calculated starting from L_p :

For sub-urban: $L_{path} = L_p - 2\left[log_{10}\frac{f}{28}\right]^2 - 5.4$

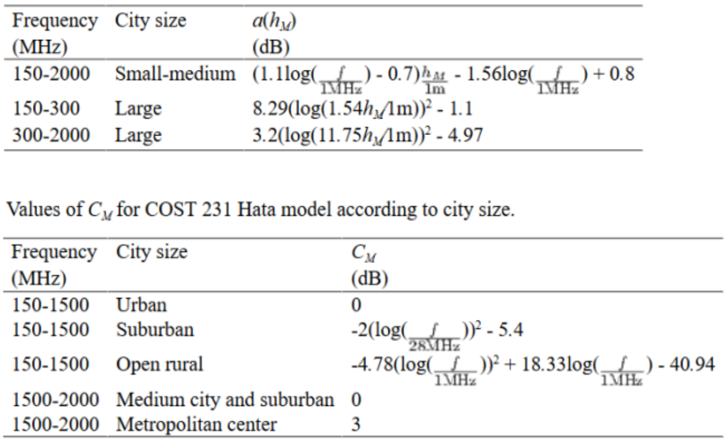
For rural: $L_{path} = L_p - 4.78[log_{10} f]^2 + 18.33 \, log_{10} f - 40.94$

2.2.2 COST 231 model

The COST 231 Model is an improved version of the Hata Model which cover the frequency bands from 1500-2000 MHz which covers the bands allocated to the 3G networks, mobile station antenna height from 1 to 10m, base station antenna height from 30 to 200m and link distance from 1 to 20 km. The path loss is given by:

$$L_p = 46.3 + 33.9 \, log\, f - 13.82 \, log\, h_b - a(h_m) + (44.9 - 6.55 \, log\, h_b) \, log\, d \quad + C_m$$

This model is not valid forh_b<=h_{roof} (base station below roof height). Values for $a(h_m)$ are listed in Table 2.



Frequency (MHz)	City size	$a(h_M)$ (dB)
150-2000	Small-medium	$(1.1\log(\frac{f}{1MHz}) - 0.7)\frac{h_M}{1m} - 1.56\log(\frac{f}{1MHz}) + 0.8$
150-300	Large	$8.29(\log(1.54h_M/1m))^2 - 1.1$
300-2000	Large	$3.2(\log(11.75h_M/1m))^2 - 4.97$

Values of C_M for COST 231 Hata model according to city size.

Frequency (MHz)	City size	C_M (dB)
150-1500	Urban	0
150-1500	Suburban	$-2(\log(\frac{f}{28MHz}))^2 - 5.4$
150-1500	Open rural	$-4.78(\log(\frac{f}{1MHz}))^2 + 18.33\log(\frac{f}{1MHz}) - 40.94$
1500-2000	Medium city and suburban	0
1500-2000	Metropolitan center	3

Table 2. Values of $\boldsymbol{a(h_m)}$ for COST – 231 Hata model according to the city size.

2.2.3 COST 231 Walfisch Ikegami model

The Walfisch Ikegami Bertoni model, also revised the COST 231 Model into a COST 231 Walfisch Ikegami model. It considers reflection and scattering above and between buildings in urban environments.

It considers both line of sight (LOS) and non-line of sight (NLOS) situations. It's valid for frequencies from 800 MHz to 2GHz, for base station heights from 4 to 50 m, mobile heights from 1 to 3 m and cell sizes up to 5 Km. It is especially convenient for prediction in urban corridors.

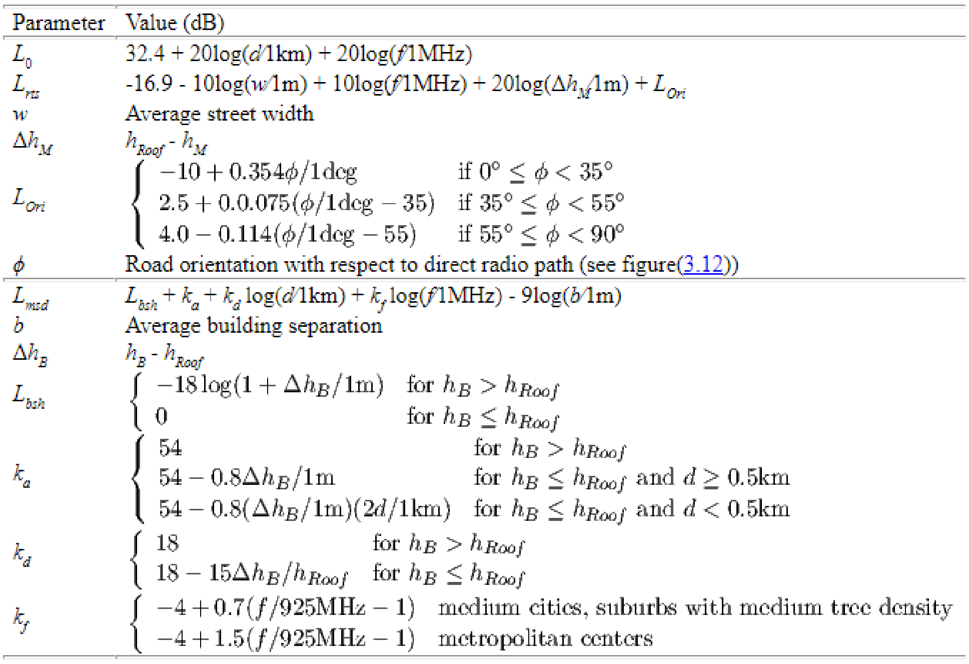
The line-of-sight case is approximated using the free space model with a link distance greater than 20m:

$$L_{LOS} = 42.6 + 26\, log\left(\frac{d}{1Km}\right), +20\, log\left(\frac{f}{1MHz}\right) \quad for\ d \geq 20m$$

The model for non-line of sight takes into consideration various scattering and diffraction properties for the surrounding buildings:

$$L_{NLOS} = L_0 + max\{0, L_{rts} + L_{msd}\}$$

Where L_0 is the free space loss, L_{rts} is a correction factor representing diffraction and scatter from roof top to street, and L_{msd} represents multiscreen diffraction due to urban rows of buildings. These terms that vary with street width, building height, and separation, angle of incidence are detailed in Table 3.



Parameter	Value (dB)
L_0	$32.4 + 20\log(d/1\text{km}) + 20\log(f/1\text{MHz})$
L_{rts}	$-16.9 - 10\log(w/1\text{m}) + 10\log(f/1\text{MHz}) + 20\log(\Delta h_M/1\text{m}) + L_{Ori}$
w	Average street width
Δh_M	$h_{Roof} - h_M$
L_{Ori}	$\begin{cases} -10 + 0.354\phi/1\text{deg} & \text{if } 0° \leq \phi < 35° \\ 2.5 + 0.0.075(\phi/1\text{deg} - 35) & \text{if } 35° \leq \phi < 55° \\ 4.0 - 0.114(\phi/1\text{deg} - 55) & \text{if } 55° \leq \phi < 90° \end{cases}$
ϕ	Road orientation with respect to direct radio path (see figure(3.12))
L_{msd}	$L_{bsh} + k_a + k_d\log(d/1\text{km}) + k_f\log(f/1\text{MHz}) - 9\log(b/1\text{m})$
b	Average building separation
Δh_B	$h_B - h_{Roof}$
L_{bsh}	$\begin{cases} -18\log(1 + \Delta h_B/1\text{m}) & \text{for } h_B > h_{Roof} \\ 0 & \text{for } h_B \leq h_{Roof} \end{cases}$
k_a	$\begin{cases} 54 & \text{for } h_B > h_{Roof} \\ 54 - 0.8\Delta h_B/1\text{m} & \text{for } h_B \leq h_{Roof} \text{ and } d \geq 0.5\text{km} \\ 54 - 0.8(\Delta h_B/1\text{m})(2d/1\text{km}) & \text{for } h_B \leq h_{Roof} \text{ and } d < 0.5\text{km} \end{cases}$
k_d	$\begin{cases} 18 & \text{for } h_B > h_{Roof} \\ 18 - 15\Delta h_B/h_{Roof} & \text{for } h_B \leq h_{Roof} \end{cases}$
k_f	$\begin{cases} -4 + 0.7(f/925\text{MHz} - 1) & \text{medium cities, suburbs with medium tree density} \\ -4 + 1.5(f/925\text{MHz} - 1) & \text{metropolitan centers} \end{cases}$

Table 3. Parameters value for COST 231 Walfisch Ikegami model [36]

CHAPTER 3

EXPERIMENTAL SETUP

The system setup integrates three LoRa nodes, they can transmit and receive data. The LoRa nodes were built and set up in Lab, then they were used to perform experiments in an indoor environment. The three LoRa nodes were built employing an Arduino Uno board and an Ind. I/O board. The Arduino Uno and Ind. I/O boards were programmed and configured using the Arduino Integrated Development Environment (IDE). Thus, these C/C++ programs written on the IDE were then loaded to the microprocessors to execute them.

All hardware components we employed to build the LoRa nodes are listed below along with their technical specifications. Also, software components as the Arduino IDE, to configure and set up the component are mentioned. The procedure to build and configure the nodes to perform wireless communication is explained step by step.

The interfacing and wiring between the microcontroller Arduino Uno and Ind. I/O and the LoRa modules are described in detail. Then the setup code for the microcontrollers and interfacing between the computer and the microcontrollers is explained. The communication parameters such as communication channel, wireless rate, and address we set for the LoRa modules are mentioned as well as the different communication modes the module can work on.

In the experiments the transmitter and receiver LoRa nodes communicated in point-to-point communication. This architecture is further clarified, and other types of possible communication are mentioned and explained.

3.1 COMPONENTS

3.1.1 Arduino board

Arduino is an open-source computing platform on a simple I/O board and software. An Arduino board can be purchased pre-assembled or built by hand (open-source hardware). Users can adapt the boards to their needs update and distribute their versions. The pre-assembled Arduino board consists of a microcontroller (MCU), which is programmed using Arduino programming language and the Arduino development environment Arduino. The IDE platform allows you to program electronic components using the Arduino programming language which is a simplified version of C/C++.

Arduino Uno shown in Figure 14 is the most popular Arduino board, it is provided with an ATMEGA 328 microcontroller and it is compatible with various shields.

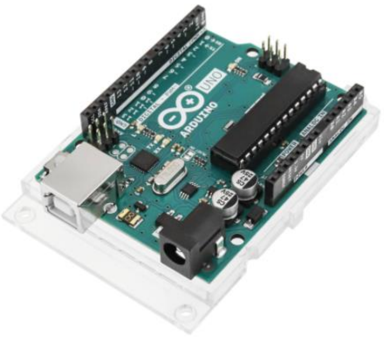


Figure 14. Arduino Uno board

Shields can enhance the Arduino board by adding more capabilities. Shields can be easily plugged over the Arduino board. There are many shields compatible with the Arduino Uno board such as the Ethernet Shield shown in Figure 15 which is used to connect the Arduino board to the internet. The Ethernet Shield is based on the Wiznet W5500 Ethernet chip and the microSD Shield equips the board with mass-storage capability.

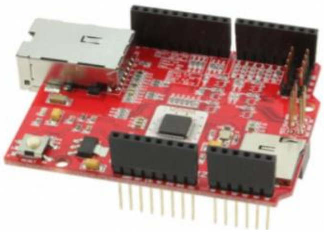


Figure 15. Wiznet w5500 ethernet shield

Arduino Uno board has useful features to connect to sensors, like the serial port (UART) shared with USB port, the capability of generating software serial, SPI, and I2C communication protocols. Power supply, memory, and other details for Arduino Uno are mentioned in Table 4.

TECHNICAL DETAILS	
Dimensions	80mm x60mm x25mm
Weight	50g
Microcontroller	ATmega328 8 bit
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O pins	14 (of which 6 provide PWM output)
PWM Digital I/O pins	6
Analog Input pins	6
DC Current per I/O pin	20mA
DC Current for 3.3V pin	50mA
Flash Memory	32KB (ATmega328P) of which 0.5KB used by bootloader.
SRAM	2KB (ATmega328P)
EEPROM	1KB (ATmega328P) Programming lock for software security
Clock speed	16 MHz
LED_BUILTINT	13

Table 4. Arduino Uno R3 technical details [37]

Arduino Uno Pin Diagram is depicted in Figure 16, an explanation for all pins will follow:

- Analog pins: Pins A0 to A5 are used as analog input
- Digital pins: Pins 0 to 13 are used as digital input or output
- Serial pins: They are used for serial communication between the Arduino board and the computer or other devices. The pin1 is used to transmit data and pin 2 to receive it [38].
- PWM pins: They are used to convert the digital signal into an analog signal by varying the width pulse. Pins 3,5,6,9,10 ad 11 gives 8-bit PWM by the function of analogWrite() [38].

- SPI pins: These pins are used to maintain SPI communication with the help of an SPI library. They include SS (pin 10 used as Slave Select), MOSI (pin 11 is used as Master Out Slave In), MISO (pin 12 is used as a Master In Slave Out), SCK (pin 13 is used as Serial Clock) [38].

- LED pin: This is an Inbuilt LED using digital pin 13. It turns on when the digital pin is put to a high level.

- Reset: This pin is used to reset the board.

- External Interrupt pins: Pins 2 and pin 3 are used to produce external interrupt.

- Vin: This is the input voltage pin of the Arduino board.

- 5V: This pin is used as a regulated power supply voltage and it gives a supply of 5V.

- 3.3V: This pin is used as a regulated power supply voltage and it gives a supply of 3.3V.

- AREF pin: This pin is used to provide a reference voltage from an external power supply.

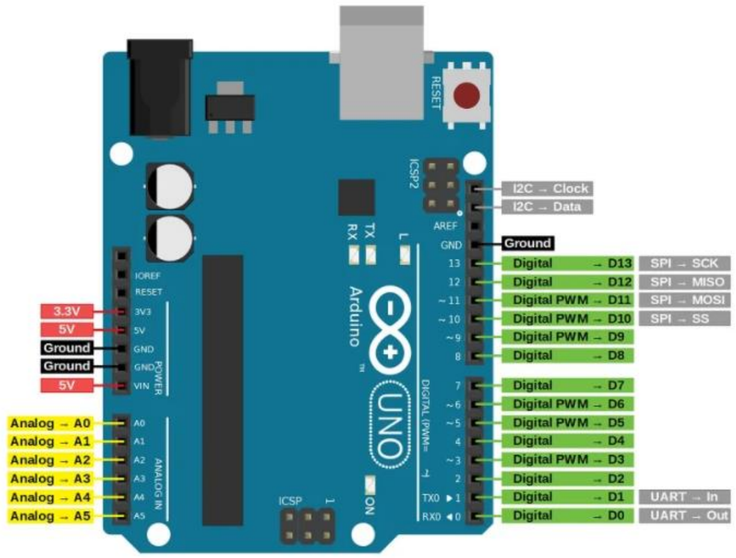


Figure 16. Arduino Uno pin diagram [38]

Another popular Arduino board are shown in Figure 17:

- Arduino Mega: This board provides more input/output ports than Arduino Uno which means that more devices such as sensors can be connected. It has 54 digital input/output and 4 embedded hardware serial ports.

- Arduino Nano: is a tiny board based on ATmega328. It works with a mini-B USB cable and there is no jack adapter, so a Vin pin must be used to turn it on. It offers the same connectivity as the Arduino Uno board but in a tiny form. Differently from UNO, in Arduino nano board male headers are used which allows to place it on a breadboard. A disadvantage of this board is that is incompatible with standard Arduino shields.

- LilyPad Arduino: is a board designed for wearables and e-textiles. It is based on ATMega168V, a low-power version of the ATMega168 or the ATMega328V.

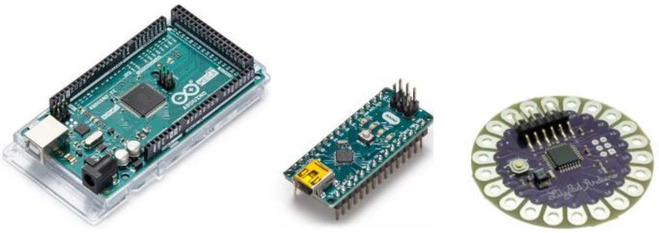


Figure 17. Arduino Mega, Arduino Nano, and Lilypad Arduino.

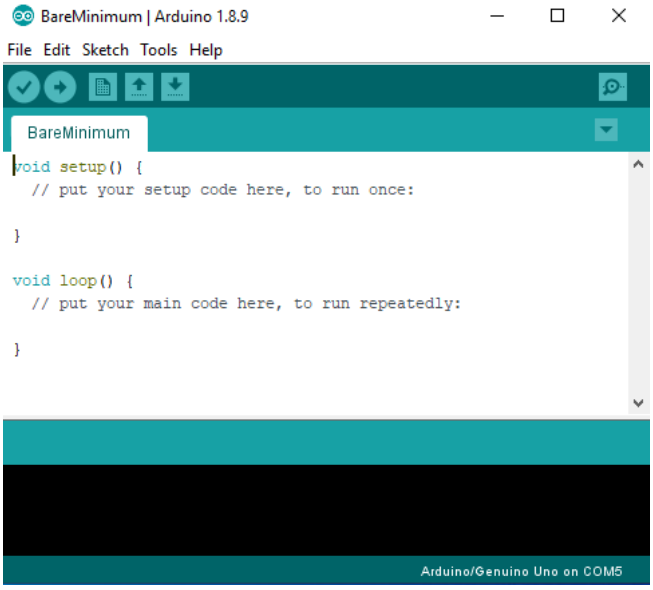
3.1.2 Arduino IDE

Is an open-source software mainly used for writing and compiling the code into the Arduino Uno. This official Arduino software is available for Windows OS, Linux OS, Mac OS and runs on the Java Platform.

The main code is known as a sketch. At the end of compilation, a Hex File will be generated for that sketch, then the file is transferred and uploaded in the controller on the board. The IDE environment broadly contains two basic parts: Editor and Compiler where at first it is used for writing the required code and the latter is used for compiling and uploading the code into the Arduino board. The environment supports C and C++ languages.

The Serial monitor which can be seen in the top right corner is an independent terminal that lets the programmer both send messages to the Arduino board and receive messages from the Arduino. It can be found also in the Tools board. The Arduino should be connected to the computer by USB cable to activate the Serial Monitor. The baud rate selected must be the same as that of the Arduino board using at that moment. The baud rate of 9600 is the standard baud rate usually used in Arduino.

As shown in Figure 18, the sketch is divided into two program parts: setup and loop. On the header, the necessary libraries are included, and global variables are declared. There is a list of libraries that can be added by clicking the Sketch button and going to the Include Library.


BareMinimum | Arduino 1.8.9
File Edit Sketch Tools Help
BareMinimum
Arduino/Genuino Uno on COM5

```
void setup() {
  // put your setup code here, to run once:

}

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

}
```

Figure 18. Arduino IDE

To select the boards, by clicking on tools there is a board section where you can select the board you are working on.

After selecting the board and the serial port, you can go to the Sketch section and verify/compile the sketch. The sketch written on the text editor is saved with the file extension .ino.

In the output panel, the compilation status of the code and the errors found in the code are shown. Before uploading the generated hex file into the Arduino board, all the errors found must be corrected. The LEDs TX and RX on the board will blink indicating that the program is running successfully.

3.1.3 LoRa module

Many LoRa transceivers radio modules based on Semtech chips are available in the market. This work will focus on a LoRa module based on the SX1278 chip.

The SX1278 transceiver makes part of the series SX1276/77/78/79, it supports Semtech's patented LoRa modulation technology. Thanks to the spread spectrum modulation, LoRa can reach longer ranges than existing systems based on FSK or OOK modulation, provides better selectivity and blocking performance improving also communication reliability. LoRa allows the user to decide on the spread spectrum modulation bandwidth (BW), spreading factor (SF), and error correction rate (CR). Moreover, each SF is orthogonal which allows multiple transmitted signals to occupy the same channel without interfering [39].

The LoRa module chosen for this work is the Atk-LoRa-01-V3.0 which is depicted in Figure 19. The LoRa module Atk-LoRa-01-V3.0 is based on SX1278 and allows serial communication (TTL), it operates at 433 MHz frequency and uses a half-duplex method for SPI communication. The module can achieve a range of 5-10Km which makes it suitable for industrial and commercial use. Moreover, the module presents an external antenna of 3 dB that expands the range of communication.

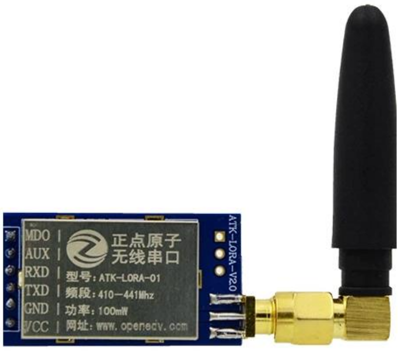


Figure 19. Atk-LoRa-01-v3.0 module with antenna

The module presents 6 pins: VCC, GND, RX, TX, AUX, MD0, where pin AUX and pin MD0 are used to configure the module.

- **Pin VCC:** This pin should be connected to 3.3-5V. Communication with a 5V single-chip microcomputer may require level adaptation.
- **Pin GND:** It should be connected to the ground.
- **Pin TX:** It is the TTL serial input of the module it should be connected to the external TX output pin. High-level TTL is greater than 2.4V and Low-level TTL is smaller than 0.4V.
- **Pin RX:** It is the TTL serial output of the module it should be connected to the external RX output pin.

- **Pin MD0:** This pin is put to a high level to enter the parameter setting. When is powered on, it cooperates with pin AUX to enter in firmware upgrade mode. When floating, pin MD0 is at a low level.
- **Pin AUX:** It is used to indicate the working status of the module; it wakes up the external MCU.
  It cooperates with pin MD0 to enter in the firmware upgrade mode when the power is on. When floating, pin AUX is at a low level.

  Relevant technical details and functionalities of the Atk-LoRa-01-V3.0 are shown in Tables 5 and 6.

TECHNICAL DETAILS	
Working frequency	410 - 441 MHz
Communication distance	3000 m
Transmission power	100 mW (20 dB)
Working Voltage	3.3V to 5V DC
Working temperature	-40˚C - 85˚C
Receiver Sensitivity	-136 dB
Antenna	3dB

Table 5. Atk-LoRa-01-V3.0 technical details

FUNCTIONALITIES	
Configuration with AT commands	AUX=0 MD0=1
Wireless Communication	AUX=0 MD0=0
Firmware upgrade	AUX=1 MD0=1

Table 6. Atk- LoRa-01-V3.0 functionalities

Under the wireless communication function, the module can operate in four modes:

- **Normal mode:** Data sent through normal mode can be received by receiving modules in normal mode or wake-up mode. The module can transmit packets 58 bytes long (buffer size). Transmission starts once the user input data reaches 58 bytes, then the module will continue sending packets. The AUX pin on the module will be put at a high level when sending the first packet, then after all the data is sent through the RF chip, the AUX will be put at a low level. At this point, the user can continue to send data up to 512 bytes. Once all data packets are received the AUX pin on the receiving module is put to a high level. Then after 2-3ms of delay, the receiving module will start sending the received data through the serial port TX pin. Finally, after all data is sent through the serial port to the MCU, the module puts the AUX pin to a low level.

- **Wake-up mode:** The module starts transmitting data as in normal mode, a wake-up code is added before each data packet is sent. The wake-up code is used to wake up the receiving module that is in power saving mode (sleep mode).

- **Power Saving mode:** The serial port on the module will be closed and will not be able to receive serial data from the MCU, so is not transmitting. The module working in power saving mode is in sleep-monitor mode. The receiving module which is working in power saving mode will monitor the wake-up code from the transmitting module working in Wake-up mode. When the receiving module receives a valid wake-up code, it will stay in the receiving state until the entire valid packet is received. Once the entire packet is received, the module will put the AUX pin to a high level. Then, it will wait 2-3ms to open the serial port to send the received data through TX. Finally, the module will put the AUX pin to a low level. The receiving module will continue to work in sleep-monitor mode. The wake-up times can be set so the module can have different receiving response delays and power consumption.

- **Signal strength mode:** This function can check the signal strength of both nodes in communication. It offers in output the SNR and the RSSI indicator which are useful for evaluating the quality of the transmission. The module will transmit packets to the module in Signal strength mode, the output of the receiving module will be:

```
SNR:7 RSSI: -47.740600
SNR:7 RSSI: -46.674000
SNR:7 RSSI: -46.674000
SNR:7 RSSI: -45.607400
SNR:7 RSSI: -46.674000
```

3.1.4 Industruino board

Industruino is a fully featured Arduino compatible board housed in a DIN-rail mountable case. It consists of a topboard and a baseboard. The topboard is the brain and the user interface of Industruino. It presents a microcontroller and an LCD screen. All signals connect from this board to the Baseboard via a 40pin FPC cable [40].

There are three topboards available: Topboard 1286-EOL, Topboard 32u4-EOL, and the newest Topboard D21G. We employed the topboard D21G shown in Figure 20.

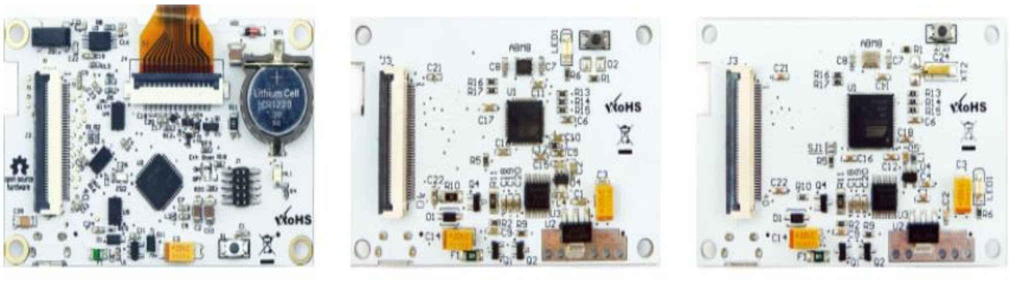


Figure 20. Topboard D21G, topboard 32u4-EOL, and topboard 1286-EOL [40]

Topboard D21G hosts the SAMD21G18 microcontroller with 256KB of flash memory. Other important features of the D21G are [40]:

- Clock speed 48 MHz
- Flash 256 KB (of which 16 KB used by USB/Ethernet bootloader)
- SRAM 32 KB
- Digital I/O Pins: 19
- PWM Pins: 12
- Analog Input Pins: 12

To use Industruino's onboard LCD screen and button panel, two different libraries can be used: The UC1701 library and the U8G library which has more features but it is heavier. The UC1701 library includes the following example sketches:

- **Hello world**: displays a string "Hello world"

- **Thermometer**
- **IndustruinoDemoCode:** a demo code that allows you to read/write all pins of the Industruino.

The baseboard link Industruino to the external elements. Currently, there are 2 types of baseboards: Proto & Ind. I/O, which are shown in Figure 21.

Proto stands for “Prototyping”. The standard Industruino comes with a prototyping baseboard that will offer protection to your project from the outside world.

Ind. I/O baseboard stands for “Industrial level I/O”. The board offers many interface options and insolated power supplies for each of its three functional zones. All field peripherals are isolated from the microcontroller via digital isolators and communicate via the I2C protocol. This board can be programmed with the Arduino IDE software, it is supplied with Arduino libraries allowing it to easily interface with industrial-level sensors and actuators. This board is an interfacing solution to bridge the gap between Arduino compatibility and the world of PLC, robust industrial sensors, and actuators [41].

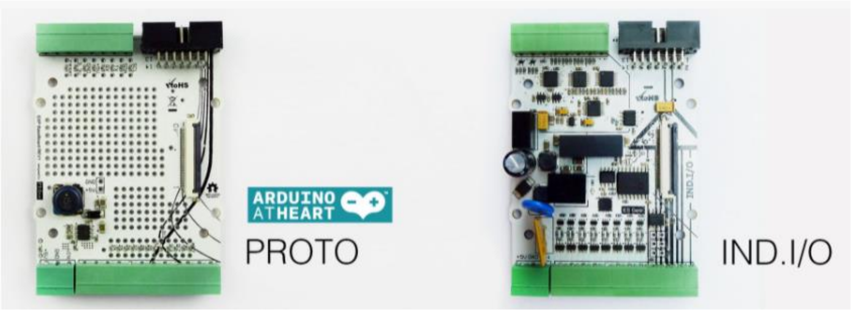


Figure 21. Proto board and Ind. I/O [40]

Along with the Topboard D21G, we employed Ind. I/O as the baseboard. The Ind. I/O board includes the features listed in Table 7 [42]:

Datasheet features	
Supply Voltage	
Standard input voltage	12/24V
Permissible range, lower limit (DC)	6.5V
Permissible range, upper limit	32 V

(DC)	
Dimensions	
Width, Height, Depth, Weight	71.5 mm, 87 mm, 58 mm, 150 g
Digital inputs	
Number of digital inputs	8
Type of digital input	Galvanically isolated serializer with interrupt
Input voltage range	0-28V
Logic HIGH voltage	>11V
Logic LOW voltage	>3V
Maximum trigger frequency	10KHz
Protection of digital outputs	Short-circuit, over-current, over-temperature, ESD, transients.
Digital outputs	
Number of digital outputs	8(shared with digital outputs
Type of digital output	Galvanically isolated high-side drive (Change pump NFET)
Output voltage range	Tied to supply voltage (6.5-32V)
Maximum current per output	2.6 A
Maximum total current	6.5 A
Maximum switching frequency	400 Hz
Protection of digital outputs	Short-circuit, over-current, over-temperature, ESD, transients.
Analog inputs	
Number of analog inputs	4
Types of analog inputs	Buffered ADC
Range of voltage measurement	0-10V
Range of current measurement	0-20mA
Switching of voltage/current mode	Automatic-in software
Resolution	18Bit
Conversion rate	18bit: 3.7 Hz - 16bit: 15Hz - 14bit: 60Hz - 12bit: 240Hz
Protection of analog inputs	ESD, transients.
Analog outputs	
Number of analog outputs	2
Types of analog outputs	Buffered DAC
Range of output voltage	0-10V
Range of output current	0-20mA

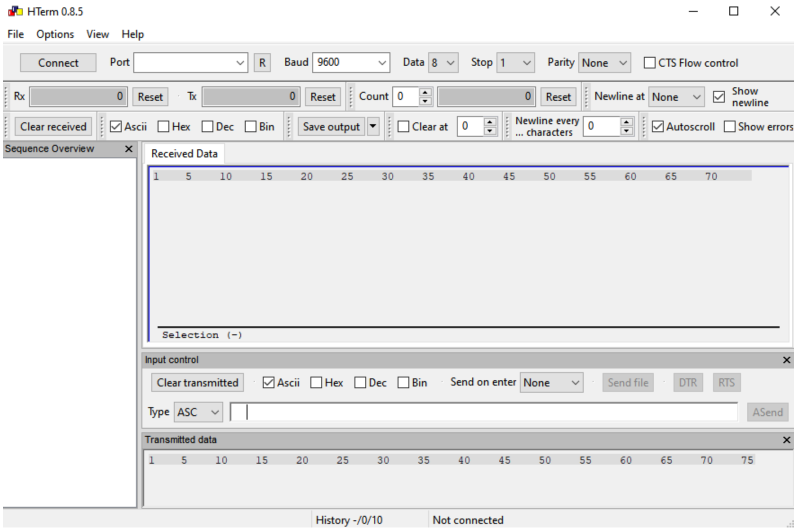
Switching of voltage/current mode	Automatic-in software
Resolution	12Bit
Update rate	20 kHz
Protection of analog inputs	Short-circuit, over-current, over-temperature, ESD, transients.
COMMUNICATION PORTS	
RS485	
Isolation topology	Isolated from MCU and analog field section
Duplex type	Half-duplex
Number of receivers on bus	32
Data rate	1 Mbps
Expansion port (direct MCU control)	
Isolation topology	Isolated from digital and analog field section
Number of pins	14
Voltage level	5V
Protocols supported	SPI, I2C, UART, 9 GPIO's
Protection of expansion port	ESD, transients.
User interface	
LCD	128x64 pixel FSTN with dimmable backlight
Push buttons	3- push button membrane panel
Environmental	
Protection class	IP20
Ambient operating temperature	0 – 55°C

Table 7. Datasheet features [42]

3.1.5 Hterm Software

Hterm software was used to enable serial communication between the LoRa module and the laptop to change the configuration of the LoRa module and obtain the RSSI measured values. Hterm is a terminal software that allows serial communication with devices connected via USB as it supports virtual serial ports (RS232) and supports also all available baud rates. HTerm 0.8.5 last version for Windows is shown in Figure 22. Other features of HTerm:

- Input and output can be made in ASCII, hexadecimal, binary, and decimal.
- Files can be sent and received.
- Parity and flow-control
- Customization via XML configuration file



Baud 9600 | Data 8 | Stop 1 | Parity None | ☐ CTS Flow control

Rx 0 | Tx 0 | Count 0 | 0 | Newline at None | ☑ Show newline

☑ Ascii ☐ Hex ☐ Dec ☐ Bin | ☐ Clear at 0 | Newline every ... characters 0 | ☑ Autoscroll ☐ Show errors

☑ Ascii ☐ Hex ☐ Dec ☐ Bin | Send on enter None

Type ASC

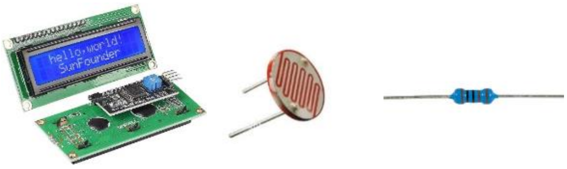
History -/0/10 | Not connected

Figure 22. Hterm software 0.8.5

3.1.6 Other components

- Power Supply:

- 16x2 LCD with i2C, a photoresistor (sensor), a 1K resistor

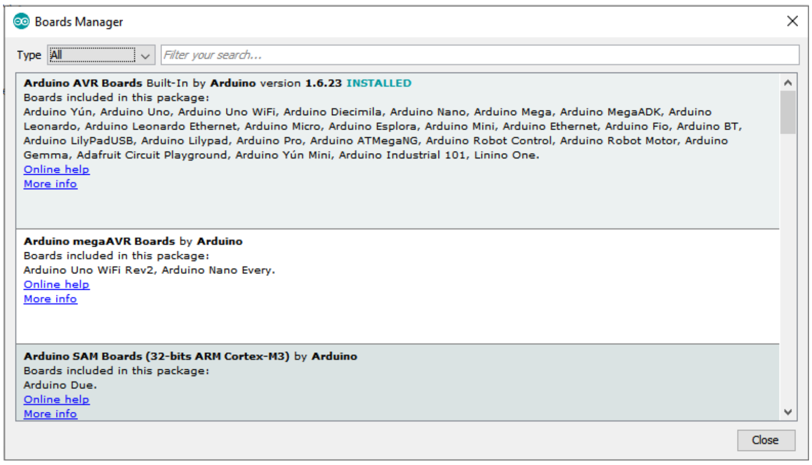


- USB-UART converter (ATK-USB-UART-V1.2), Breadboard, and jumpers

3.2 LoRa nodes

3.2.1 Setup

Arduino IDE 1.8.15 software for Windows was downloaded and installed. Then, from the IDE environment, we installed the board packages for Arduino and Ind. I/O (Arduino AVR boards package and Industruino SAMD Boards package). The board manager in Arduino IDE is shown in Figure 23.



Arduino AVR Boards Built-In by **Arduino** version **1.6.23** INSTALLED
Boards included in this package:
Arduino Yún, Arduino Uno, Arduino Uno WiFi, Arduino Diecimila, Arduino Nano, Arduino Mega, Arduino MegaADK, Arduino Leonardo, Arduino Leonardo Ethernet, Arduino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATMegaNG, Arduino Robot Control, Arduino Robot Motor, Arduino Gemma, Adafruit Circuit Playground, Arduino Yún Mini, Arduino Industrial 101, Linino One.
Online help
More info

Arduino megaAVR Boards by **Arduino**
Boards included in this package:
Arduino Uno WiFi Rev2, Arduino Nano Every.
Online help
More info

Arduino SAM Boards (32-bits ARM Cortex-M3) by **Arduino**
Boards included in this package:
Arduino Due.
Online help
More info

Figure 23. Board manager on Arduino IDE

To build the LoRa nodes the RF module Atk-LoRa-1-V3.0 LoRa was used and MCU boards: Arduino Uno and Ind.I/O were employed.

The LoRa modules were working under wireless communication function (MD0=Low level, AUX=Low level) with default parameters:

- **Baud rate:** 9600 bps
- **Wireless data rate:** 19.2 Kbps
- **Module address:** 0
- **Communication channel:** 23
- **Transmission power:** 20 dB

The first two LoRa nodes based on Arduino Uno were built, a transmitter and a gateway node. The MCU boards and LoRa module were connected as shown in Figure 24.

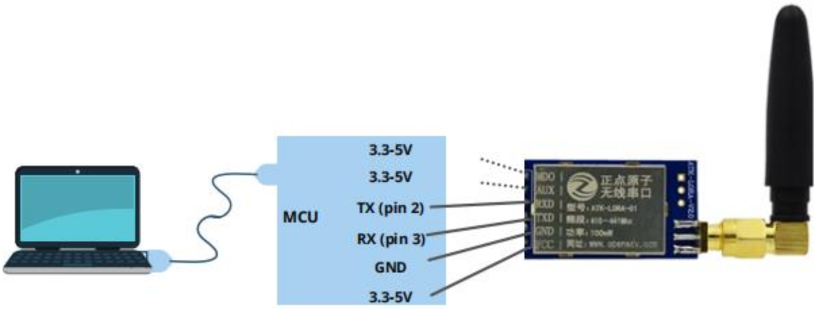


Figure 24. Wiring connection between Arduino Uno and LoRa module Atk-LoRa-01-v3.0

Next, the Arduino Uno board was configured. It was connected to the laptop with a standard USB, then from Tools->boards we selected the Arduino Uno board and from Tools->ports, we selected the Serial port corresponding to the board.

To program the board so that we were able to send and receive messages through the LoRa module, a library was downloaded from the library manager. Among the many libraries written for LoRa modules available in the market, the best fitting our LoRa module Atk-LoRa-1-V3.0 was the SoftwareSerial library as it is simple to use. The SoftwareSerial library is used to simulate a serial port through software, it enables serial communication with a digital pin other than the serial port.

Finally, a sensor (photoresistor) was added to the transmitter node and 16x12 LCD with I2C was added to the gateway node so that the sensing data from the photoresistor was printed on the LCD. The resulting transmitting and receiving nodes are shown in Figure 25. Another two simple sketches were written for the two LoRa nodes respectively.

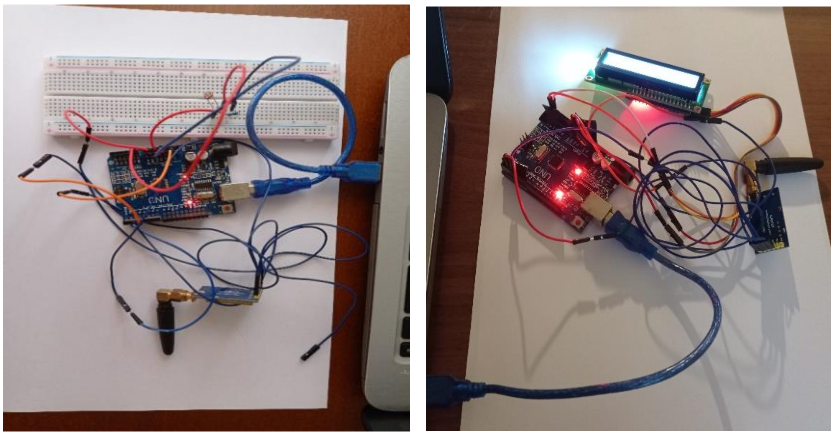


Figure 25. Transmitter node (left) and gateway node (right)

Moreover, another two LoRa nodes were built. A transmitter node based on Ind. I/O board and a gateway node based on Arduino Uno board. The gateway node was built in the same way as the earlier gateway node.

The Atk-LoRa-1-V3.0 LoRa modules were working under wireless communication function (MD0=Low level, AUX=Low level) with default parameters:

- **Baud rate:** 9600 bps
- **Wireless data rate:** 19.2 Kbps
- **Module address:** 0
- **Communication channel:** 23
- **Transmission power:** 20 dB

To set up the Arduino IDE environment for Ind. I/O. From the File-> Boards, the Industruino D21G was selected. The board was connected to the laptop with a USB mini cable, then from Tools->port, the corresponding serial port was selected.

Then we installed the Indio library which comes with demo codes. These codes were used as a starting point to write the sketches to control de I/O on the Industruino. Also, the UC1701 display library was downloaded from the library manager from which we used the demo codes: Hello World and IndustruinodemocodeD21G.

The IndustruinodemocodeD21G sketch comes with a menu that can be used by the user to control the I/O of Ind. I/O. The loaded menu is depicted in Figure 26.

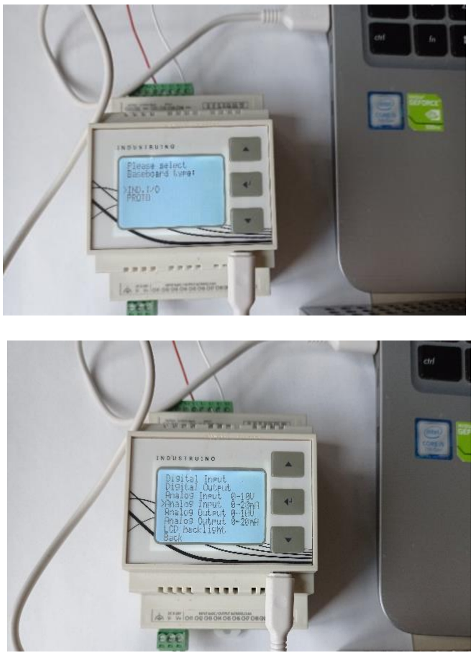


Figure 26. Ind. I/O board menu

The SoftwareSerial library is not supported by the Industruino, but the board has an inbuilt hardware serial pin that can be used directly.

Serial ports onboard D21G are SerialUSB for USB (Serial Monitor), Serial for hardware serial on D0/D1, and Serial1 for hardware serial on D10/D5.

We connected the LoRa module to Serial1. The pins of the Atk-LoRa-1-V3.0 LoRa module were welded to the MCU pins on the breadboard as depicted in Figure 27. Then the LoRa module was placed onto the baseboard inside Ind. I/O shell.

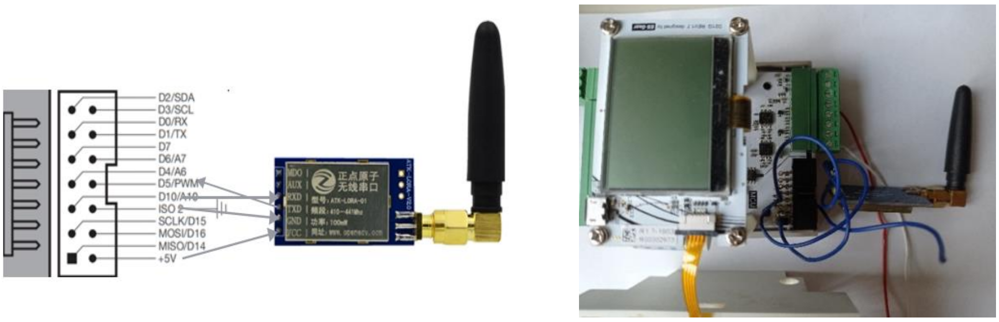


Figure 27. Connection diagram Atk-LoRa-01 and Ind. I/O

We power up the breadboard with a 12V external power supply. The power supply was connected to pins V+ and V- on the baseboard as shown in Figure 28.

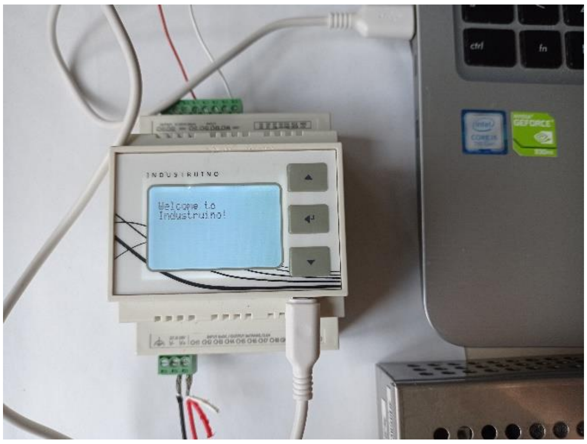
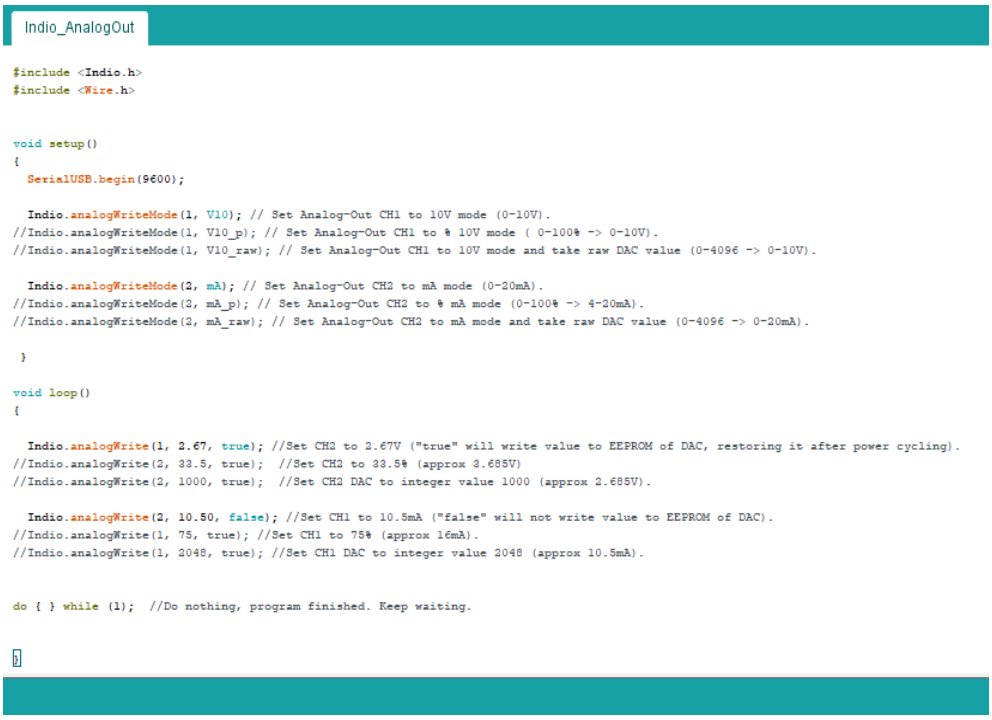


Figure 28. Ind. I/O powered up by a 12V power supply

Once Ind. I/O board is powered up, we tested some features of Ind. I/O board. We set the board to give power in output and then to receive power.

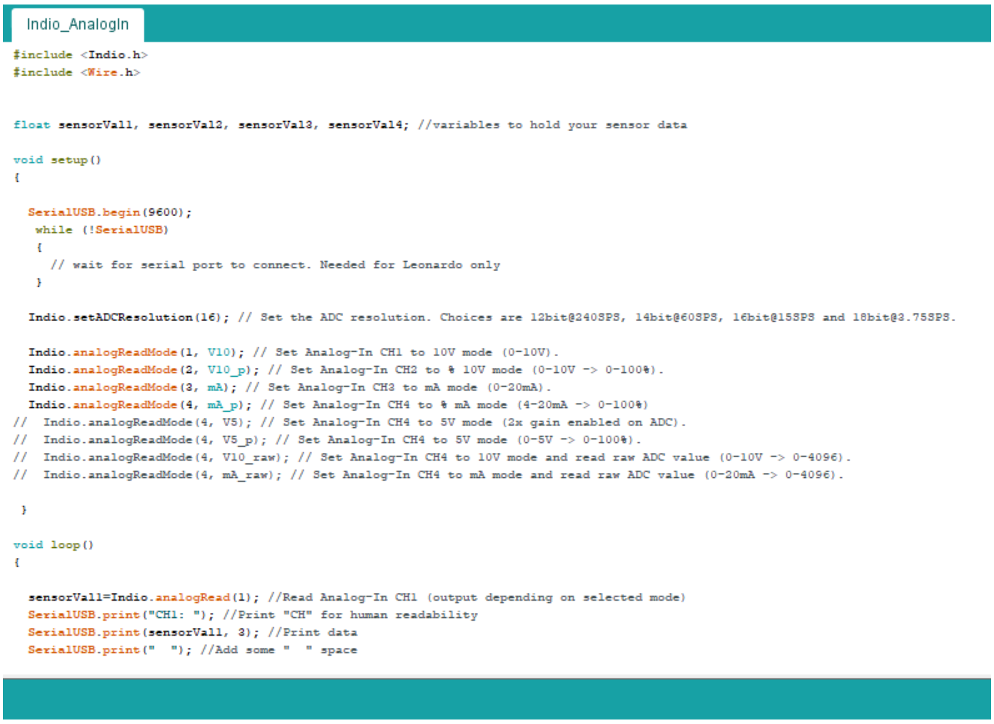
We loaded to the board the Indio_AnalogOut demo code shown in Figure 29. In the sketch, pins are initialized to give analog voltage in output. We can choose the output voltage, channel (1, 2), and the mode (10V mode, mA mode).



```
#include <Indio.h>
#include <Wire.h>


void setup()
{
  SerialUSB.begin(9600);

  Indio.analogWriteMode(1, V10); // Set Analog-Out CH1 to 10V mode (0-10V).
//Indio.analogWriteMode(1, V10_p); // Set Analog-Out CH1 to % 10V mode ( 0-100% -> 0-10V).
//Indio.analogWriteMode(1, V10_raw); // Set Analog-Out CH1 to 10V mode and take raw DAC value (0-4096 -> 0-10V).

  Indio.analogWriteMode(2, mA); // Set Analog-Out CH2 to mA mode (0-20mA).
//Indio.analogWriteMode(2, mA_p); // Set Analog-Out CH2 to % mA mode (0-100% -> 4-20mA).
//Indio.analogWriteMode(2, mA_raw); // Set Analog-Out CH2 to mA mode and take raw DAC value (0-4096 -> 0-20mA).

 }

void loop()
{

  Indio.analogWrite(1, 2.67, true); //Set CH2 to 2.67V ("true" will write value to EEPROM of DAC, restoring it after power cycling).
//Indio.analogWrite(2, 33.5, true);  //Set CH2 to 33.5% (approx 3.685V)
//Indio.analogWrite(2, 1000, true);  //Set CH2 DAC to integer value 1000 (approx 2.685V).

  Indio.analogWrite(2, 10.50, false); //Set CH1 to 10.5mA ("false" will not write value to EEPROM of DAC).
//Indio.analogWrite(1, 75, true); //Set CH1 to 75% (approx 16mA).
//Indio.analogWrite(1, 2048, true); //Set CH1 DAC to integer value 2048 (approx 10.5mA).


do { } while (1);  //Do nothing, program finished. Keep waiting.


}
```

Figure 29. Indio_analogout sketch

The demo code Indio_AnalogIn shown in Figure 30, the value read from the Analog_In channel is then printed on serial monitor (SerialUSB). From the sketch, we can set the ADC resolution (12bit@240SPS, 14bit@60SPS, 16bit@15SPS, and 18bit@3.75SPS), the Analog_In channel (1 to 4), and the mode (V10, V10_p, mA, mA_p, and other options).



```
#include <Indio.h>
#include <Wire.h>


float sensorVal1, sensorVal2, sensorVal3, sensorVal4; //variables to hold your sensor data

void setup()
{

  SerialUSB.begin(9600);
   while (!SerialUSB)
   {
     // wait for serial port to connect. Needed for Leonardo only
   }

  Indio.setADCResolution(16); // Set the ADC resolution. Choices are 12bit@240SPS, 14bit@60SPS, 16bit@15SPS and 18bit@3.75SPS.

  Indio.analogReadMode(1, V10); // Set Analog-In CH1 to 10V mode (0-10V).
  Indio.analogReadMode(2, V10_p); // Set Analog-In CH2 to % 10V mode (0-10V -> 0-100%).
  Indio.analogReadMode(3, mA); // Set Analog-In CH3 to mA mode (0-20mA).
  Indio.analogReadMode(4, mA_p); // Set Analog-In CH4 to % mA mode (4-20mA -> 0-100%)
//  Indio.analogReadMode(4, V5); // Set Analog-In CH4 to 5V mode (2x gain enabled on ADC).
//  Indio.analogReadMode(4, V5_p); // Set Analog-In CH4 to 5V mode (0-5V -> 0-100%).
//  Indio.analogReadMode(4, V10_raw); // Set Analog-In CH4 to 10V mode and read raw ADC value (0-10V -> 0-4096).
//  Indio.analogReadMode(4, mA_raw); // Set Analog-In CH4 to mA mode and read raw ADC value (0-20mA -> 0-4096).

 }

void loop()
{

  sensorVal1=Indio.analogRead(1); //Read Analog-In CH1 (output depending on selected mode)
  SerialUSB.print("CH1: "); //Print "CH" for human readability
  SerialUSB.print(sensorVal1, 3); //Print data
  SerialUSB.print("  "); //Add some "  " space
```

Figure 30. Indio_analogin sketch

3.3 SYSTEM ARCHITECTURE

Two point-to-point system architectures were employed for the experiments, on each of the antennas were working in transparent transmission and normal mode. Transparent transmission means that you will get what you send. If A node sends data: AA BB CC DD to B node, B node will receive AA BB CC DD.

Both LoRa modules transmitting and receiving had the same default parameters:

- **Baud rate:** 9600 bps
- **Wireless data rate:** 19.2 Kbps
- **Module address:** 0
- **Communication channel:** 23
- **Transmission power:** 20 dB

In the first architecture, packets are wirelessly sent using LoRa protocol to a receiver /gateway that forwards the data to the computer for further elaboration. The transmitter node and receiver node are based on an Arduino board. The first architecture was used to set up the communication between the transmitter node and receiver node. The first configuration is depicted in Figure 31.

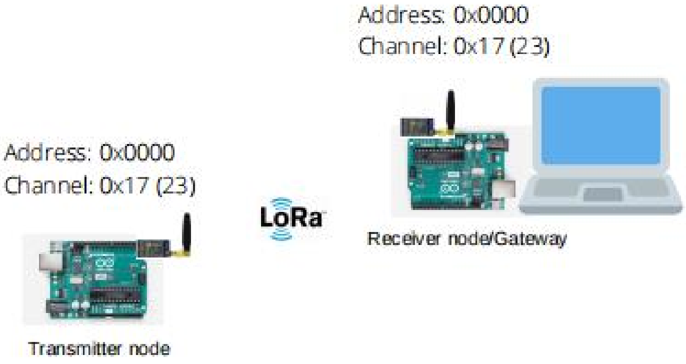


Figure 31. System architecture 1

In the second architecture, packets are wirelessly sent using LoRa protocol to a receiver/gateway that forwards the data to the computer for further elaboration.

The transmitter node is based on Ind. I/O board while the receiver/gateway node is based on an Arduino board. The second architecture is depicted in Figure 32.

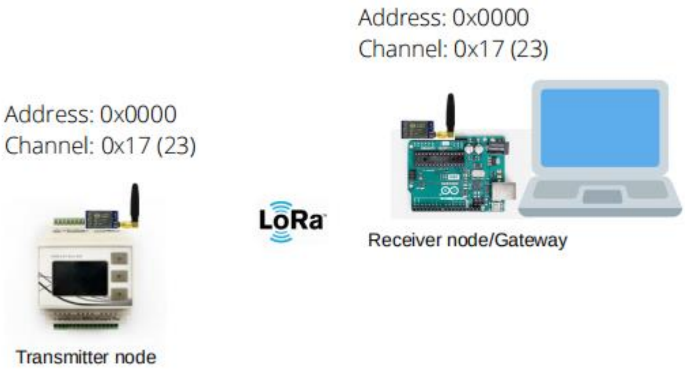


Figure 32. System architecture 2

The Fixed transmission is another type of communication that can be also employed on Atk-LoRa-V3.0 modules. On a Fixed transmission, messages can be sent to a specific device on a specific channel.

To change the default configuration mode on the Atk-LoRa-V3.0 module, the module had to be put in configuration mode. To enter in configuration mode, pin MD0 was put to a high level and pin AUX to a low level (idle). Then a USB-UART adapter was used to interface the LoRa with the computer, by using the Hterm software we established a serial communication with the LoRa module. On the Hterm terminal, the following parameters were set to enter in configuration mode:

- **Baud rate**: 115200 bps
- **Data bits**: 8
- **Stop bit** :1
- **No parity bit**

Once the serial communication was established, the configuration on the LoRa modules can be changed from the Hterm software. The LoRa module responds to AT commands, the list of AT commands that works for the Atk-LoRa-V3.0 module is shown in Table 8.

AT commands	
AT	Test module respond conditions
AT+ MODEL?	Check device model
AT+CGMR?	Software version number
AT+ UPDATE	Check if the device is in firmware update mode
ATE1	Enable Echoing
ATE0	Disable echoing
AT+RESET	Reset (Restart)
AT+DEFAULT	Restore factory settings
AT+FLASH=	Parameter saving
AT+ADDR=?	Query device address configuration range
AT+ADDR?	Query the device address
AT+ADDR=	Configuring device address
AT+TPOWER=?	Query transmitting power configuration range
AT+TPOWER=?	Query transmitting power
AT+TPOWER=	Configuring transmitting power
AT+CWMODE=?	Query working mode configuration range
AT+CWMODE?	Query working modes
AT+CWMODE=	Configuring the working mode
AT+TMODE=?	Query transmitting mode configuration range
AT+TMODE?	Query transmitting mode
AT+TMODE=	Configuring the transmitting mode
AT+WLRATE=?	Query the wireless rates and channels
AT+WLRATE?	Query wireless rate and channel set
AT+WLRATE=	Configuring the wireless rate and channel
AT+WLTIME=?	Check the configuration sleep time range
AT+WLTIME=?	Check sleep time
AT+WLTIME=?	Configure sleep time
AT+UART=?	Query serial port baud rate range
AT+UART?	Check the configured serial port baud rate
AT+UART=	Configure serial port

Table 8. AT set of instructions

CHAPTER 4

MEASUREMENTS

4.1 COMMUNICATION TESTS

We performed preliminary tests to establish first communication on a point-to-point architecture. These tests were performed using the first point-to-point configuration.

All tests were carried out in the laboratory. On the first test, the transmitter node and receiver/gateway node were placed at 1 m from each other with no obstacles. Figure 33 shows the first test set up with the first architecture.

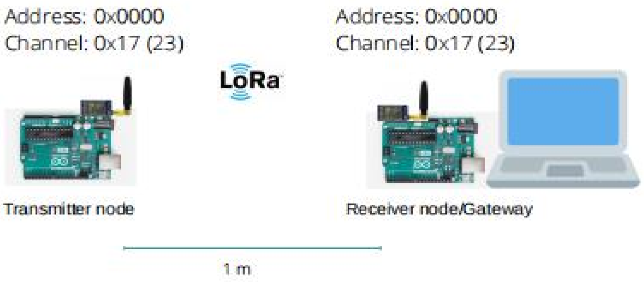


Figure 33. Test 1 - System architecture 1

Two simple sketches were developed to test the communication between the nodes.

-The transmitter was programmed to send packets every 3 seconds; packets were 4 digits long.

-The receiver/gateway was programmed to receive packets on the Serial monitor.

We tested the communication sending 200 packets, 400 packets, and finally 800 packets. For each, the test was run 4 times, then an average was calculated, and finally the percentage of packet loss. The percentages of packet loss are shown in Table 9.

	200 packets	400 packets	800 packets
Packet loss (%)	0	0	0.031

Table 9. Packet loss (%) - Test 1

On the second test, the transmitter node and receiver/gateway node were placed at 2,5 meters from each other with a wall between them. Figure 34 shows the second test setup.

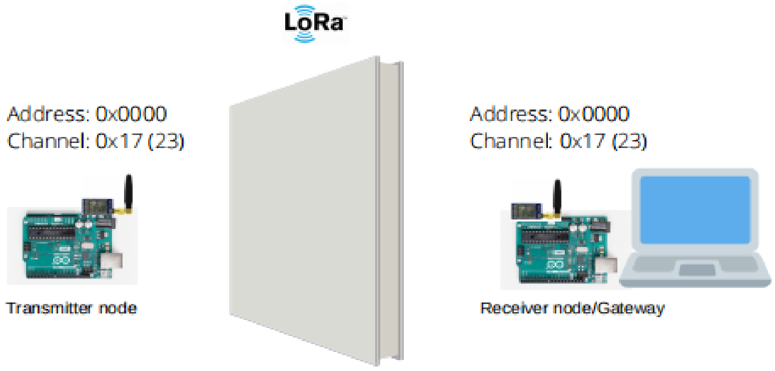


Figure 34. Test 2 - System architecture 1

Two simple sketches were written to test the communication between the nodes.

-The transmitter was programmed to send packets every 3 seconds, for a total of 800 packets.

-The receiver/gateway was programmed to receive packets on the Serial monitor.

We tested the communication sending packets of lengths 4 ("0123"), 8 ("0123467"), and finally 12 ("012345678901"). The test was run 4 times for each packet length then an average was calculated and finally the percentage of packet loss. The percentages of packet loss are shown in Table 11.

	Packet length 4	Packet length 8	Packet length 12
Packet loss (%)	0.031	0.062	0.094

Table 10. Packet loss (%) - Test 2

Finally, a third test was performed, the transmitter and receiver/gateway nodes were disposed at 6 meters from each other with an elevator as an obstacle between them. The elevator is made up of metal and concrete materials. The disposition of the transmitter node and gateway node is shown in Figure 35.

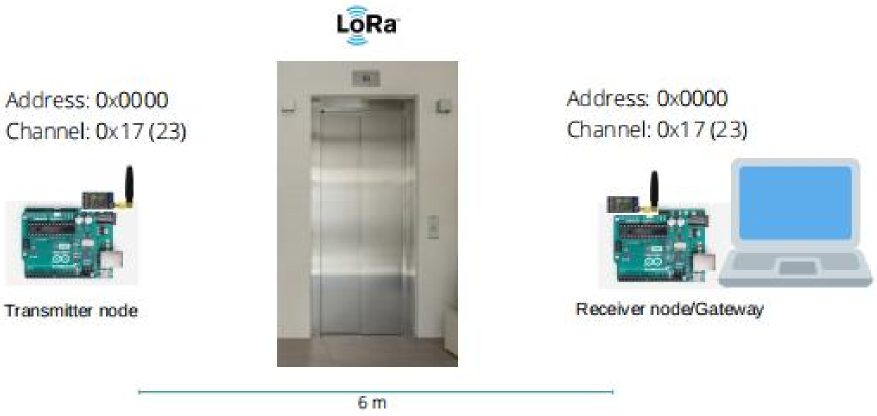


Figure 35. Test 3 (6m) - System architecture 1

Simple sketches were written to test the communication between the nodes:

-The transmitter was programmed to send packets every 3 seconds, for a total of 800 packets.

-The receiver/gateway was programmed to receive packets on the Serial monitor.

We tested the communication sending packets of lengths 4 (“0123”), 8 (“0123467”), and finally 12 (“012345678901”). The number of packets lost is shown in Table 11.

	Packets length 4	Packet length 8	Packet length 12
Packet loss (%)	0.031	0.125	0.187

Table 11. Packet loss (%) - Test 3 (6 m)

The same test was repeated with the transmitter and receiver/gateway node placed at 12 meters distant from each other, as shown in Figure 36, with the same elevator as the obstacle between them.

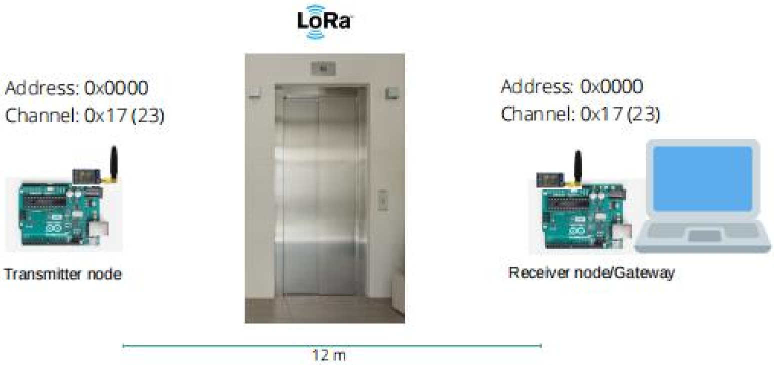


Figure 36. Test 3 (12m) - System architecture 1

The resulting number of packets lost is shown in Table 12.

	Packet length 4	Packet length 8	Packet length 12
Packet loss (%)	0.062	0	0.062

Table 12. Packet loss (%) - Test 3 (12 m)

As expected, the percentage of packet loss when there were no obstacles between the nodes is close to zero. Considering a wall as an obstacle, the radio waves can easily penetrate the wall that is why results are still very low remaining very close to zero.

Radio waves cannot penetrate obstacles made of metals; most of the radio energy is reflected. The signal arrives at its destination by diffraction. Packet loss percentage stays very close to zero also when considering a large metal obstacle like an elevator; they also show that losses are fewer when transmitting at 12 m than when transmitting at 6 m. This may be due to the signal having more space to propagate to arrive at the receiver instead of being soon reflected by the metal and concrete walls. By looking at the last three experiments, sending packets of larger doesn't influence packet loss.

4.2 INDOOR POINT-TO-POINT MEASUREMENTS

The results from the first communication tests showed that communication between the nodes was stable, so we proceeded to carry out the experiments using the second architecture. In the second architecture, Industruino Ind. I/O at the transmitter node that sends packets to the receiver/gateway node based on Arduino Uno.

We placed the transmitter and the receiver node at 1 m distant from each other to perform the first experiment. The first experimental setup with the second architecture is shown in Figure 37. Simple sketches were developed to assess the communication.

-The transmitter was programmed to send packets every 3 seconds; packets were 4 digits long.

-The receiver/gateway was programmed to receive packets on the Serial monitor.

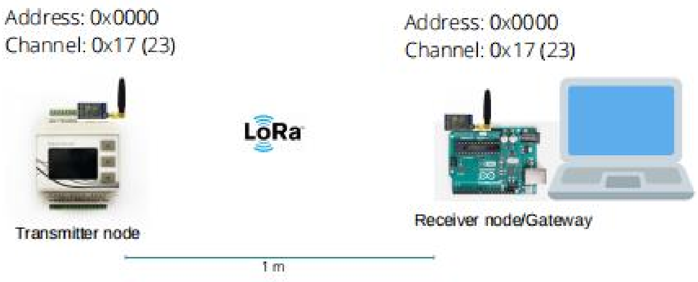


Figure 37. Experiment 1

We examined the communication sending 200 packets, 400 packets, and finally 800 packets. For each, sketches were run 4 times, then an average was calculated, and finally the percentage of packet loss.

Proceeding with the second experiment, the transmitter and receiver/gateway node were positioned at 2.5 meters from each other with a wall between them. Figure 38 shows the second experimental setup with the second architecture.

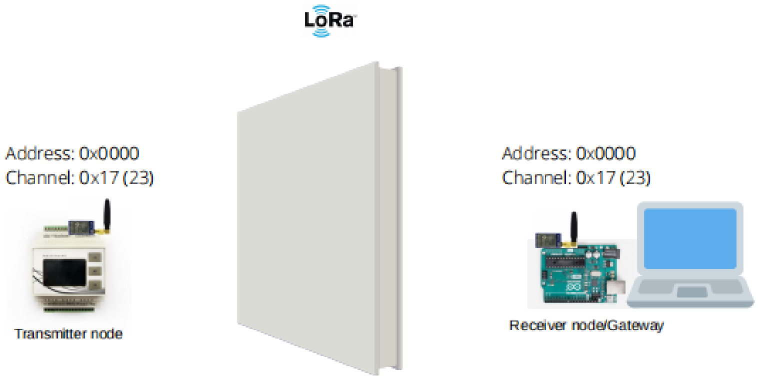


Figure 38. Experiment 2

Simple sketches were written to assess the communication between the nodes.

-The transmitter was programmed to send packets every 3 seconds, for a total of 800 packets.

-The receiver/gateway was programmed to receive packets on the Serial monitor.

We examined the communication sending packets of lengths 4 ("0123"), 8 ("01234567"), and finally 12 ("012345678901"). Sketches were run 4 times for each packet length then an average was calculated and finally the percentage of packet loss.

Lastly, the third experiment was done; the transmitter and receiver/gateway nodes were disposed at 6 m from each other with a large metal obstacle between them. The disposition of nodes is shown in Figure 39.

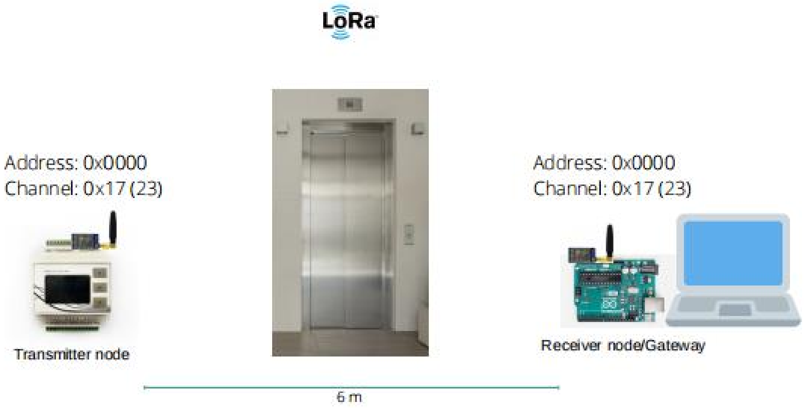


Figure 39. Experiment 3 (6 m)

Simple sketches were written to assess the communication between the nodes.

-The transmitter was programmed to send packets every 3 seconds, for a total of 800 packets.

-The receiver/gateway was programmed to receive packets on the Serial monitor.

We examined the communication sending packets of lengths 4 (“0123”), 8 (“01234567”), and finally 12 (“012345678901”). Sketches were run 4 times for each packet length then an average was calculated and finally the percentage of packet loss.

Later we placed the nodes at 12 m distance from each other with the same obstacle in between them, as shown in Figure 40, and then we performed the same previous experiment.

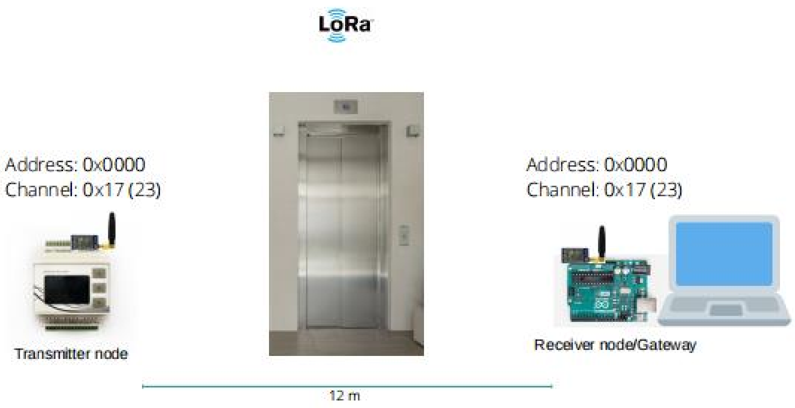


Figure 40. Experiment 3 (12 m)

After finishing performing the experiments with the second architecture, we carry on with the second part of the experiments using this same architecture. The transmitter node was placed in a fixed position in a room and the receiver node was placed on different points in the laboratory. The laboratory includes 4 rooms and has a dimension of 65 m^2 .

Two different scenarios in the indoor environment were taken into consideration. In the first scenario, the signal propagation will experience attenuation by the presence of walls, thus a Line-of-sight environment. The second scenario also considers the presence of an elevator, so we expected higher attenuation of the signal power.

The survey trail and transmitter node placement of the experimental testbed for the first and second scenarios are shown in Figure 41 and Figure 42, respectively. The minimum distance between two locations in the survey trail was

1.3 m. At each location, we calculated the average of 50 RSSI values and the scanning frequency was set to 0.5 seconds. The RSSI values were obtained by the receiving LoRa module working in signal strength mode. To change the configuration of the module, the LoRa module was first set in configuration mode then the configuration was changed with AT + CWMODE= 3 command.

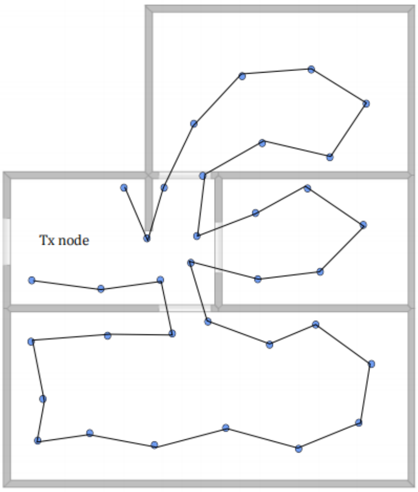


Figure 41. Survey trail and transmitter placement - Scenario 1

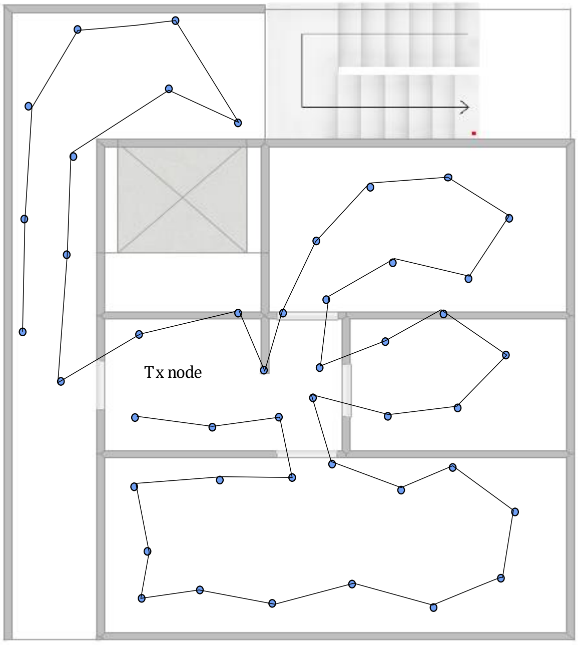


Figure 42. Survey trail and transmitter placement - Scenario 2

CHAPTER 5

RESULTS AND ANALYSIS

Results for the first experiment are shown in Table 13. The first results show that in a non-obstructed LOS propagation, communication between the nodes is quite robust as packet loss is very low. By sending 200, 400, and 800 packets, packet loss stay below 1%.

	200 packets	400 packets	800 packets
Packet loss (%)	0	0.062	0.125

Table 13. Packet loss (%) - Experiment 1

Table 11 shows the results of the experiments in the obstructed environment. When radio wave propagation is obstructed by a wall packet loss stays below 1%.

In the third experiment, being the elevator, an obstacle made of metal and concrete materials, the signal will arrive at the destination by diffraction and reflection from walls. Packet loss stays below 1% also in this case. Results also show that we have better performance when transmitting at 12 meters than when transmitting at 6 meters. This is due to the signal having more space to propagate to the receiver.

By looking at the results of Table 14, changing the size to packets from 4 to 8 and 12 doesn’t affect packet loss.

Comparing results from the preliminary tests and the experiments performed with nodes from the second architecture, packet loss performances are alike when transmitting with Arduino Uno board and Industruino Ind. I/O board.

Obstacle	Packet length 4	Packet length 8	Packet length 12
Wall	0.187	0.156	0.218
Elevator (6m distance)	0.250	0.094	0.156
Elevator (12m distance)	0	0.094	0.062

Table 14. Packet loss (%) - Experiments 2, 3

To estimate the path loss for the first indoor scenario, the Log-normal path loss model was used to model the LOS propagation. We set d_0 to 1 meter and $p_r(d_0)_{dB}$ was set to -28.0 dB.

$$p_r(d)_{dB} = \bar{p}_r(d_0)_{dB} - 10n\, log\left(\frac{d}{d_0}\right),d > d_0$$

The MMSE estimate for the path loss index n was found by using the formula :

$$J(n) = \sum_{i=1}^{k}(P_r - \overline{P_r})^2$$

Where P_r is the computed value, RSSI average value measured at a location and $\overline{P_r}$ is the calculated value at each location.

Being $J(n)$ the sum of squared errors between measured and estimated values, we calculated the value of n that minimize the mean square error by equating derivative of $J(n)$ to 0.

The RSSI estimation results of the Path Loss model are shown in Figure 43, and the resulting path-loss index of the Path Loss model is 2.49. From Figure 43, we can see that the radio waves propagation matches with the Path loss model since the propagation environment approximate a LOS environment.

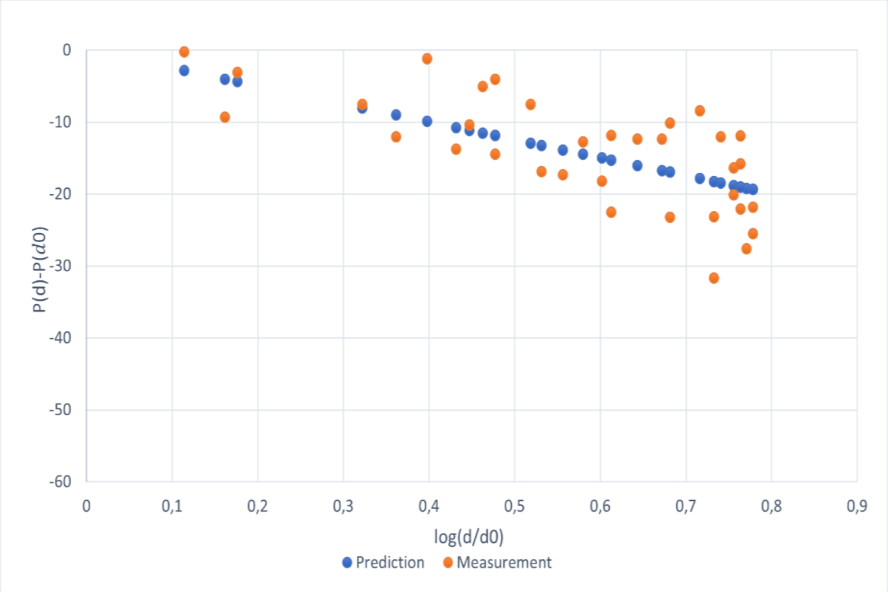


Figure 43. The RSSI estimation results of path loss model - Scenario 1

Signal propagation in the second indoor scenario was modeled with the Log-normal path loss model, Dominant Path Model 1, and Dominant Path Model 2. The path loss indexes for the three models are shown in Table 15. The RSSI estimation results of the three propagation models are shown in Figure 44, Figure 45, and Figure 46.

	Path-loss index n_{LOS}	**Path-loss index** n_{NLOS}
Path Loss Model	2.75	-
Dominant Path Model 1	2.55	-
Dominant Path Model 2	2.32	3.62

Table 15. The path-loss index of three propagation models

From Figure 44, we can see that the path loss model approximates the radio waves propagation inside the rooms and in the corridor part but behind the elevator, the attenuation level of signal strength is more significant.

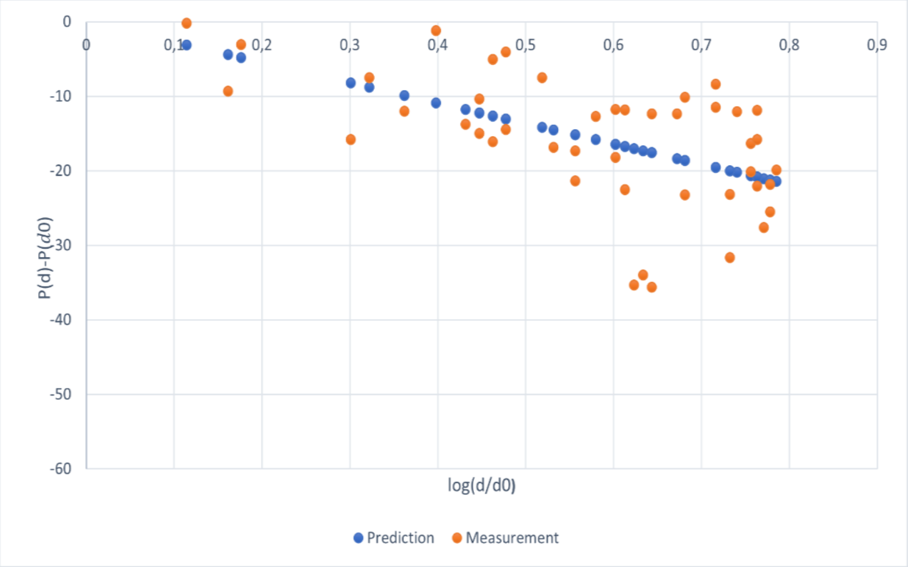


Figure 44. The RSSI estimation results of path loss model - Scenario 2

From the experimental results in Figure 45, we can see that the RSSI estimation results are better when considering L, the Dominant Path. Comparing the results of the Path Loss Model, the path loss index with the Dominant Path Model decreases from 2.75 to 2.55. Thus, the dominant path approximates better the radio wave propagation than the Path Loss Model.

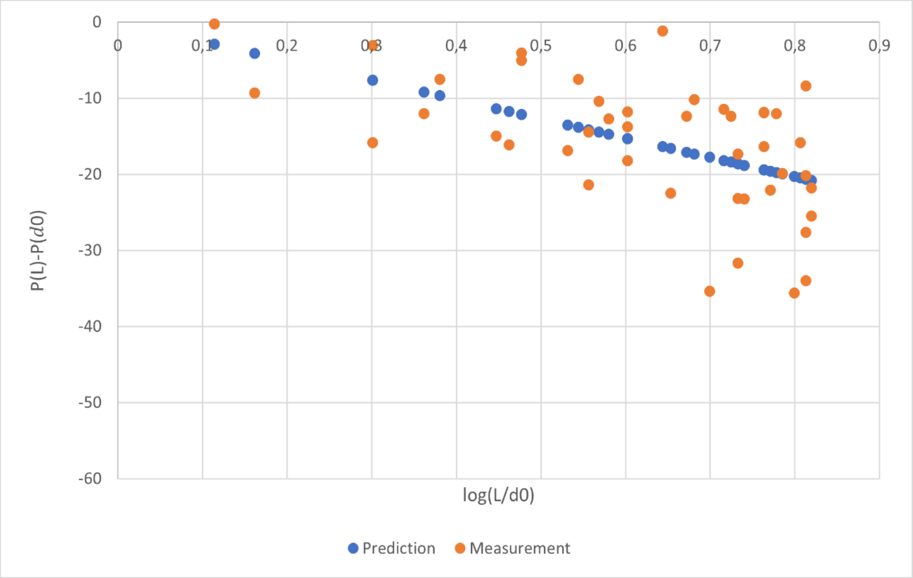


Figure 45. The RSSI estimation results of dominant path model 1 - Scenario 2

Finally, from Figure 46 we can see that the RSSI estimation results of Dominant Path Model 2 additionally improves since the Dominant Path Model 2 analyzes LOS propagation and NLOS propagation separately. The path-loss index $nLOS$ for the Dominant Path Model 2 is 2.32 in the LOS environment and the path-loss index $nNLOS$ is 3.62.

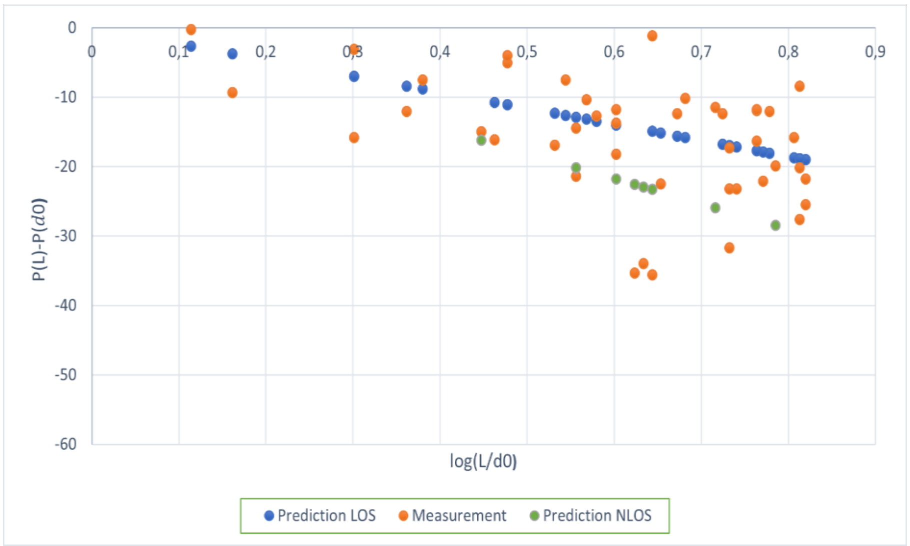


Figure 46. The RSSI estimation results of dominant path model 2 - Scenario 2

CHAPTER 6

CONCLUSIONS

LoRa describes the proprietary protocol for the physical layer and does not describe the functions of the upper layers, denoted as LoRaWAN. Even if the LoRa specification is proprietary, some parts of the protocol are well documented and others not. In the first part of this work has been presented as much information as it could be gathered about LoRa.

Some applications of LoRa in the real world have also been mentioned, focusing on Lora applications in the industrial sectors. Typically, in industrial applications, reliability and security are the most important system attributes. Industrial sectors such as agriculture, transportation, automobile factories, logistics, and manufacturing are characterized by heavy equipment and metals that are highly reflective to radio waves. Thus, these forms of indoor environment present a harsh condition for wireless communications. The precise placement of sensors in these harsh environments is critical; the knowledge gained from in-service monitoring can allow customers to predictively maintain critical equipment, thus avoiding expensive interruptions.

The first goal of this work was to assess the LoRa modulation considering obstacles that are present in a harsh environment typical of industrial applications. To this aim, a testbed has been built, a LoRa network with a single transmitter and a single gateway was deployed indoors.

An Industruino Ind. I/O, an Arduino compatible board was chosen as MCU board at the transmitter node, as it can handle high voltage digital and analog input/output typically used in both factories and build automation to enable smarter sensing.

The first results show that sources of interference such as metallic structures not significantly influenced the quality of the link. Moreover, sending longer data packets didn’t influence the Packet error rate values. Ind. I/O met the MCU board functions satisfactorily at the transmitter node.

The second goal was to predict the radio conditions in a typical indoor scenario with obstacles such as walls, doors, metal, and other furniture. To this aim, we selected three consolidated indoor propagation models: Path Loss Model, Dominant Path Loss 1, and Dominant Path Loss 2.

Based on the measurements, we can say that using a LoRa module transceiver working at 433MHz we can cover the area under consideration with average RSSI values above 65 dB at all measurement locations. The comparison between the measured values and theoretical values computed with the mentioned three indoor propagation models shows that the Dominant Path loss Model 2 is the most suited to be used for the first design of a LoRa based communication in an indoor industrial environment.

REFERENCES

[1] https://techjury.net/blog/how-many-iot-devices-are-there/#gref

[2] http://chiefit.me/wp-content/uploads/2017/03/HPE-Aruba_IoT_Research_Report.pdf

[3] https://behrtech.com/lpwan-technology/

[4] https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/resources/innovation-technology/low-power-wide-area-technology

[5] Mekki K, Bajic E, Chaxel F, Meyer F, A comparative study of LPWAN technologies for large-scale IoT deployment, 2019, ICT Express 5(1):1-7

[6] URL: https://lora-developers.semtech.com/library/tech-papers-and-guides/lpwan-technologies

[7] Ferré G, Giremus A, LoRa Physical Layer Principle and Performance Analysis, ICECS 2018 25th IEEE International Conference on Electronics Circuits and Systems, 2018, pp.65-68

[8] SX1272/3/6/7/8: LoRa Modem Designer’s Guide AN1200.13

[9] Augustin J, Yi T, Clausen T, Townsley W, A Study of LoRa: Long Range & Low Power Networks for the Internet of Things, 2016, Sensors 16(9)

[10] https://lora.readthedocs.io/en/latest/

[11] https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/

[12] URL: https://lora-alliance.org

[13] https://tech-journal.semtech.com/understanding-the-lorawan-architecture

[14] https://www.thethingsnetwork.org/article/the-things-network-architecture-1

[15] Vehbi C. G, Gerhard P.H, Industrial wireless sensor networks: Challenges, Design principles, and technical approaches, 2009, IEEE Transactions on industrial electronics, vol. 56, no. 10

[16] https://lora-developers.semtech.com/library/tech-papers-and-guides/lorawan-class-a-devices/

[17] https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/

[18] https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate/

[19] https://www.semtech.com/lora/lora-applications

[20] https://www.semtech.com/lora/lora-applications/smart-agriculture

[21] https://www.semtech.com/lora/lora-applications/smart-homes-buildings

[22] Andrade O, Sang Guun Yoo, A comprehensive study in the use of LoRa in the development of smart cities, 2019 p.24

[23] https://www.semtech.com/lora/lora-applications/smart-utilities

[24] https://www.semtech.com/lora/lora-applications/smart-healthcare

[25] https://www.semtech.com/company/press/semtechs-lora-devices-monitor-equipment-energy-use-with-connected-plugs

[26] https://www.semtech.com/lora/lora-applications/smart-supply-chain-logistics

[27] https://www.semtech.com/company/press/semtechs-lora-technology-enables-smarter-management-of-industrial-applications

[28] https://www.semtech.com/company/press/semtechs-lora-technology-enables-more-efficient-construction-and-mining-machines

[29] https://www.semtech.com/company/press/semtechs-lora-technology-enabling-smarter-mining-solutions-from-transco-industries-inc

[30] https://www.semtech.com/company/press/semtech-and-aiut-optimize-petroleum-gas-management-with-lora-devices

[31] https://www.semtech.com/company/press/advantech-delivers-innovative-iot-solutions-using-semtechs-lora-technology

[32] https://www.semtech.com/company/press/semtech-supports-new-iot-solutions-for-cloud-managed-asset-and-facility-monitoring-developed-by-cisco

[33] https://www.intechopen.com/books/new-approach-of-indoor-and-outdoor-localization-systems/one-stage-indoor-location-determination-systems

[34] Zhao Y, Li M, Shi F, Indoor radio propagation model based in the dominant path, Int. J. Communication Network and System Sciences, 2010, p.3, pp.330-337

[35] Sharma N, Magarini M, Dossi L, Reggiani L, Nebuloni R, A study of channel model parameters for aerial base station at 2.4GHz in different environments, 2018, 15th IEEE Annual Consumer Communication & Networking Conference (CCNC).

[36] http://morse.colorado.edu/~tlen5510/text/classwebch3.html

[37] https://diyi0t.com/arduino-uno-tutorial/

[38] https://store.arduino.cc/arduino-uno-rev3

[39] SX1276-7-8-9 Datasheet

[40] https://www.industruino.com

[41] https://industruino.com/page/baseboards

[42] https://static.industruino.com/downloads/datasheets/Industruino_INDIO_Specification_June'2017.pdf