People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Power and Control

Final Year Project Report Presented in Partial Fulfillment of the Requirements for the Degree of

MASTER

In **Electronics**

Option: **Computer Engineering**

Title:

Design and Implementation of IOT System Based LoRa

Presented by:

- **Djadir Mohammed Ibrahim Mansour**

Supervisor:

Dr. Maache Ahmed

Registration Number:………/2020

Abstract

The internet of things (IOT) is one of the most growing technologies in the last few years. However, this technology increases the demand for connected devices. Various standards are currently contending to gain an edge over the competition and provide the massive connectivity that will be required by a world in which everyday objects are expected to communicate with each other. Among these standards, Low-Power Wide Area Networks (LPWANs) are continuously gaining momentum, mainly thanks to their ability to provide long-range coverage to devices, exploiting license-free frequency bands.

LoRa is emerging as one of the most promising LPWAN, since it enables the energy-constraint devices distributed over wide areas to establish affordable connectivity. However, how to implement a cost-effective and flexible LoRa network is still an open challenge.

This project aims to design and implement a prototype for a low cost LoRa network for IOT application. The network prototype will be designed based on the five-layer conventional IOT architecture. Our design will include hardware and software implementation of a LoRa end node, single-channel LoRa gateway and a network server. Furthermore, we implement a new transmission protocol between the end node and the gateway based on the variation of the LoRa module parameter. In addition, our implementation also includes a data handling system that collects the data from the network server and save it to the cloud.

Dedication

I dedicate this project to my Dear parent whose believe in me and support me during all my studies. I would like to thank them for every thing they did for me without them i would not be able to to complete my studies.

Acknowledgement

This thesis is the largest academic project I have done so far in my life. It took me several months to complete it; Starting from researching the topic, implementing the system and writing the thesis itself. This process allowed me to live one of the most motivating and exciting experiences of my academic life; I received a lot of support from Multiple people during this process that I would like to show appreciation to them.

First of all I would like to praise almighty ALLAH for giving us the power to realize this project.

I would also thank my supervisor Dr. Maache who guided and helped me during the different phases of the project. Many thanks are given to all the professors and workers at the IGEE for their valuable encouragement and precious pieces of advice.

I would like, as well, to show my deepest gratitude to my beloved parents because they kept fueling my energy with their constant support during all the tough times.

A particular dedication is attributed to Wameedh scientific club for providing me with the necessary resources and an adequate workplace.

Last but not least, I would like to thank all my family members and friends; they have been a source of inspiration and encouragement for years.

Table of contents

List of figures

Chapter 1: IOT System Modelling

Chapter 2: LoRa Modulation Technique

Chapter 3:Design and Implementation

Chapter 4: Experimental Results and Discussion

List of tables

List of Abbreviations

BLE: bleutooth low energy

BW: Bandwidth

CoAP :Constrained Application Protocol

CIOT:Cellular IOT

CSS:Chirp Spread Spectrum

CR:Correction Rate

CRC: Cyclic Redundancy Check

DUP:Duplicate flag

E2E: end to end

FCR:Forward Correction Rate

HAL: Hardware Abstraction Layer

HTTP:Hypertext Transfer Protocol

IOT: internet of things

LR-WPAN:Low-Rate Wireless Personal Area Network

LoRa: Long range

LoRaWAN :LoRa for Wide Area Networks

LPWAN:Low Power Wide Area Network

M2M :Machine-to-Machine

MAC: medium access control

MCU: Micro-controller Unit

MQTT:Message Queuing Telemetry Transport

NF: Noise Figure

RSSI:Received Signal Strength Indication

SF: Spreading Factor

SNR: Signal to Noise Ratio

SPI: Serial Peripheral Interface

TCP:Transmission Control Protocol

TSDB: time series data base

PCB: Printed Circuit Board

General Introduction

General Introduction:

The Internet of Things (IOT) is a collection of connected objects, embedding electronics, software, sensors, and wireless connectivity protocols that collect and exchange information with applications through wireless networks connected to the Internet. Nowadays internet of things became a trending topic as the number of connected devices is expanding rapidly with estimates stating that 75.44 billion IOT devices or more will be active this by 2025 [1]. This technology affects many industries and services such as automotive, smart building, smart agriculture, smart cities, industrial control, e-health, and supply chain.

The growth of connected devices poses some key challenges in terms of connectivity technology that need to be upgraded to handle and solve the following challenges:

- The high density of connected devises and achieve the best possible efficiency in the face of packet collision.

- Power consumption of the connected devices should be optimized so that these devices, which run on batteries, achieve long autonomy.

- These devices must support long-range communication.

The capability of an IOT network protocol to fulfil these requirements needs to be carefully investigated before a massive deployment can be implemented. This has drawn a lot of attention to the new wireless IOT connectivity family named Low-Power Wide-Area Network (LPWAN). The latter is well suited to support services and use cases which need long-range communication (dozens of km) to reach devices which must have a low power consumption budget in order to operate several years remotely on a single battery pack. The major use cases of LPWAN include Smart Cities, supply chain management with asset tracking & condition monitoring, smart grids, smart agriculture.

One of the prominent LPWANs is LoRa, LoRa is radio modulation technology invented in 2010 by the French startup Cycleo and then acquired in 2012 by Semtech (a semiconductor company). A MAC layer has been added to standardize and extend the LoRa physical communication layer onto internet networks. This MAC layer is called the LoRaWAN (LoRa for Wide Area Networks) specification. The specification is open sourced and supported by the LoRa Alliance. The LoRaWAN protocol also includes several key wireless networks feature such as E2E encryption and security, adaptive data rate optimization, quality of service, and other advanced communication applications.

The objective of this project is to implement an IOT system based on LoRa. This system will consist of a LoRa node to act as end devices, a gateway to receive data and send it via wifi to our server, the simulation of the LoRa modulation was done using an open source simulator Pathos.

This report consists of four chapters. Chapter 1 represents a general background about IOT technology. Chapter 2 explains the LoRa modulation technique and its important parameter. This chapter will include also simulation of an end to end LoRa communication and a discussion about the obtained results. Chapter 3 will include the software and hardware design of the IOT node and the server-side of our system. Chapter 4 is dedicated for the implementation and testing of our system. Finally, conclusions and future work are outlined at the end of the report.

Chapter I IOT System Modelling

1.1 IOT overview :

The idea of connecting things to the Internet has been discussed and researched extensively for the last two decades as the new technology breakthrough. Connecting embedded devices and sensors to the internet is predicted to have a great impact on the industry revolution, health care, and energy production.

M2M (Machine-to-Machine) with a similar concept of connecting things has been applied in different industries since the early 1990s [2]. However, M2M and IOT are two distinct concepts: IOT is wider and comes with many new options and touch different sectors. It also relies on new technologies in terms of connectivity, application, and storage. The main purpose of M2M is to maintain the connectivity between a specific machine and the remote host in a fixed, proprietary installation, which is generally configured to monitor and control only those specific types of machines.

Hence, M2M provides end-to-end connectivity to exchange data between machines. It manages point-to-point systems, such as elevator remote control and a vending machine.

However, the IOT concept broadens the idea of M2M to create a new Internet of connected things horizontally rather than vertical solutions. To establish the horizontal interaction between IOT entities, things connected to the network send their data to the cloud, from which humans, computer systems, and other things read the data, interact with each other and integrate with other standalone applications/solutions. Thus, IOT aims at creating an open, scalable, standards-based, service-oriented network in which very large amount of nodes can communicate and interact with each other.

One of the main purposes of IOT is the construction of a horizontal, versatile, scalable, and accessible data architecture where a large number of devices and humans interact with each other.

However, the vast amount and distribution of the devices over the network create technical challenges in the network architecture framework.

1.2 IOT Architecture:

IOT architecture comprises a collection of physical objects, sensors, cloud services, developers, actuators, communication layers, users, business layers, and IOT protocols. Because of the wide domain of internet objects, there is no single consensus on IOT architecture, which is universally agreed. Different researchers proposed different architectures. According to most researchers' views, a conventional IOT architecture is considered as three layers [3], which depicted in Figure1.1.

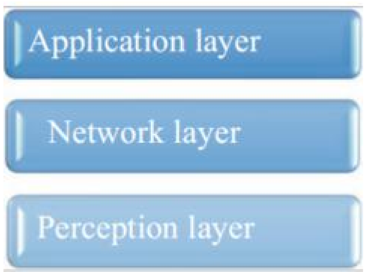


Figure 1.1: Conventional IOT architecture

1.2.1-Perception Layer :

The perception layer is also known as the sensing layer or the device layer. This is the lowest layer of conventional IOT architecture. It is mainly responsible for collecting and transforming useful information/data from devices. The devices in this layer can be sensors and actuators controlled by an embedded systems such as micro-controllers and have the capability to collect data in real-time. They support low power, generally, they use wireless connectivity protocol such as Zigbee BLE and lPWAN [4].

1.2.2-Network Layer :

The network layer or gateway layer. This layer's primary responsibility is to transmit data between the application and the IOT architecture's perception layer. This layer mainly gathers information and provides a layer of perception with multiple applications and servers. This layer is essentially an internet-based convergence of communication and networks.

This layer consist of Gateway, which ensures unique addressing and routing capabilities for the unified integration of countless devices into a single cooperative network. Various types of technologies are supported by this layer such as wired, wireless, and satellite, and diverse type of protocols such as MQTT and HTTP, ,CoAP.... [5]

1.2.3-Application layer :

The application layer is considered as a top layer of conventional IOT architecture. This layer provides personalized services based on the user's needs. The primary responsibility of this layer is to link the wide gap between users and applications [6]. This IOT layer brings the industry together to achieve high-level intelligent applications such as disaster monitoring, health monitoring, transposition, medical, and ecological environments and manages global management for all smart applications.

It should be also noted that some researchers consider that IOT consists of five architecture layers as seen in Figure 1.2.

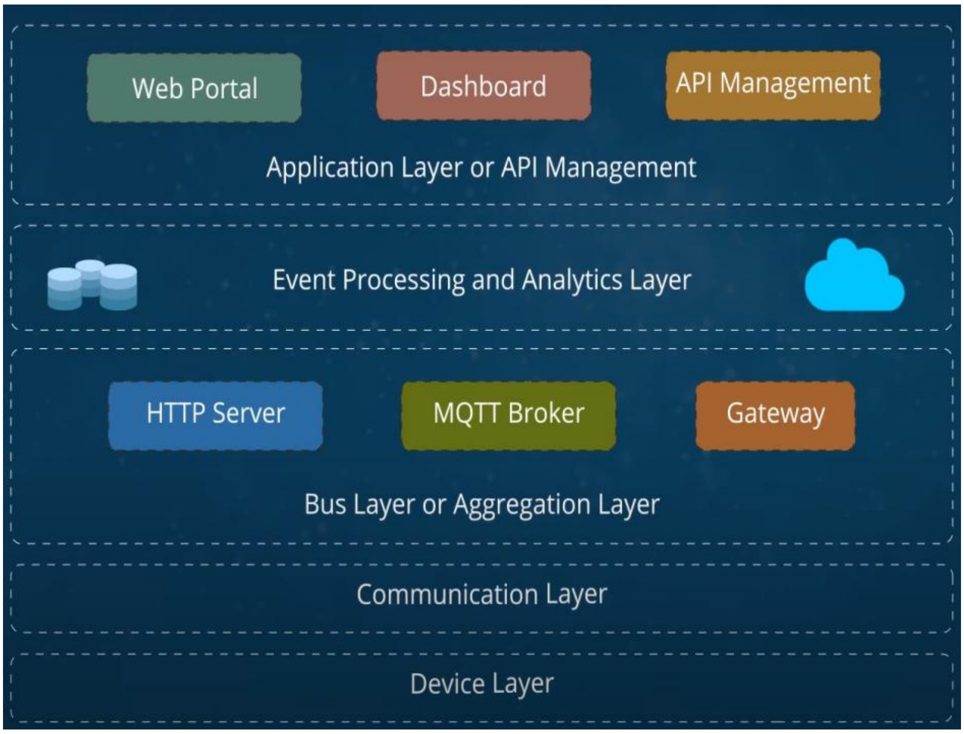


Figure 1.2: IOT five layer reference architecture

In this architecture, there are two additional layer between the communication and the application layer. An aggregation layer consists of webserver or broker to send data from the communication layer to the next layer, which is the analytical layer. The latter is where the data get processed on cloud and stored onto the databases.

1.3 IOT Communication and Messaging protocol:

In this part, we will discuss communication protocols and messaging protocols and list the characteristic of those protocols. Finally we will compare these protocols.

1.3.1 Connectivity protocols:

Connectivity protocols facilitate the establishment of the connection between devices and gateway, they can be divided into two main categorizes wired and wireless. The ability of those protocols are affected by some parameters such as range data rate and power consumption.

According to the previous parameter, we will list some of the very popular and important connectivity protocols and do a comparison to select which is the better to use for which application.

1.3.1.1 Low-Rate Wireless Personal Area Networks

Low-Rate Wireless Personal Area Network (LR-WPAN) technologies create small networks, typically covering and interconnecting the devices owned by an individual or operating in a house. These standards provide low data rates and short-range communication, in order to focus on efficient battery use.

The range of a single hop for these devices is between 10 m to 100 m, with a raw data rate between 20 and 250 kbit/s. The common technologies used for this category are BLE (Bluetooth low energy) and Zigbee[4].

1.3.1.2 Cellular IOT

Cellular IOT (CIOT) standards will operate in licensed bands and leverage the already existing cellular network coverage to provide internet access to IOT devices. The fact that the infrastructures for the network is already installed is a great benefit and will make deployment time very short.

Currently, three different standards have been proposed [7]: EC-GSM, LTE-M, and NB-IOT. EC-GSM is designed to leverage and improve on legacy EDGE and GPRS systems to provide better coverage and range, with limited power requirements. LTE-M will integrate with LTE to make use of its capacity and performance and bring new power-saving options to increase device battery life.

Finally, the new NB-IOT standard will focus on ultra-low-end IOT applications, once again leveraging the existing LTE infrastructures [7]. Future 5G networks are also expected to provide connectivity to IOT devices.

1.3.1.3 Low Power Wide Area Networks

LPWANs have recently been emerging as an alternative to LR-WPANs and CIOT, mainly thanks to the range limitations of LR-WPANs and to the fact that CIOT is still in a very early stage of deployment. These networks provide wireless connectivity using a star topology and long-range transmission in the unlicensed sub-GHz frequency bands [8]. The other great benefit brought by LPWANs is an increased power efficiency: many of these technologies so far have made the claim of being able to sustain a device for 10 years on a couple AA batteries.

One of the main competitors among standards for this architecture is LoRa. It is a technology that exploits a new spread spectrum design that enables a higher receiver sensitivity in order to trade data rate for coverage, decreasing the former to increase the latter.

LoRa and LoRaWAN are, respectively, a proprietary modulation developed and owned by Semtech Corporation [9] and a network standard, focused on leveraging useful properties of the LoRa modulation, proposed by the LoRa Alliance [10]. The LoRa modulation allows for very good receiver sensitivities at a contained chip cost, thus achieving long-range transmissions (up to 13 miles in a rural environment) at the price of a reduced data rate, in the 0.3 to 50 kbps range. At the same time, the LoRaWAN standard that allows multiple LoRa devices to communicate together aims at shifting the burden of administering the network towards a central control point. This allows devices to be as simple as possible, and gives a central coordinator the power of easily tuning each device's parameters in order to accommodate new nodes in the network.

After the connectivity have been studied let's compare these technologies in order to chose the best solution for our system. Figure 1.3 [11] compares the connectivity technologies in term of range and data rate.

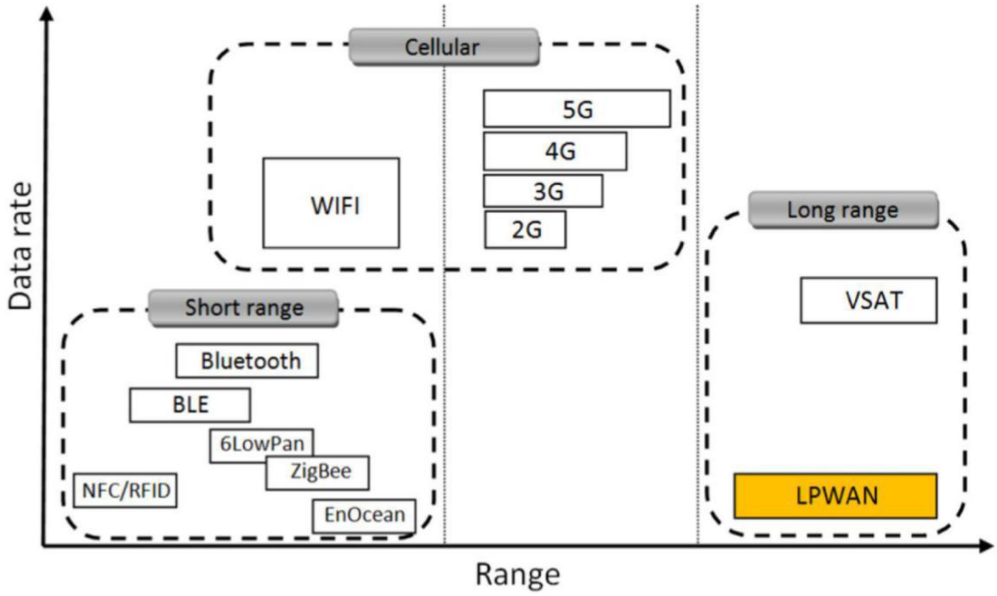


Figure 1.3: Required data rate vs. range capacity of radio communication technologies

The figure shows the ability of LPWAN technology to provide a very wide range compared to other technologies in addition to very low power consumption, which makes it the best connectivity solution for transmitting light-weight data for long distances. In the next chapter, the LoRa technology is explored more.

1.3.2Messaging protocols:

Messaging protocols can operate in both application layer and communication layer. Their main role is to facilitate the data transmission between devices; they define how each device will be referred and standardizing messages.

Three of the dominants messaging protocols in IOT are HTTP, COAP, and MQTT. In this section we will discuss the features of each of them.

1.3.2.1 HTTP:

HTTP (Hypertext Transfer Protocol) is one of the widely used application layer protocols for transferring data over the Internet. HTTP architecture defines a server (generally referred to as webserver), which serves the data and a client (generally a web browser), which requests the data from the server. Hence, HTTP follows a request-response scheme from the client is sent to the server, and the server replies with a response to the client. Since HTTP is an application layer protocol, it can be used on top of any reliable transport level protocol such as TCP or RTP [12].

HTTP is designed for communication between 2 systems only at a time. While this works fine for requesting resources from the web as a user, it does not fulfil the needs of IOT setup. where there are a large number of sensors that are generating data at the same time and want to push this ahead to the server at the same time as well. Hence, HTTP does not fulfil the need for one-to-many communication between sensors and the server. HTTP may be used to communicate between the server and the cloud.

1.3.2.2 CoAP:

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. It decreases the effects of the difficulties created by the constrained nature of these networks (e.g. low-power, lossy). CoAP utilizes a request/response based architecture between a CoAP client and a CoAP server. CoAP uses UDP to avoid the overhead created by connection-oriented protocols such as TCP.

Each CoAP message consists of a four-byte binary header followed by a sequence of options and payload (see Figure 1.4). The header contains 2-bit CoAP version, 2-bit message type indicator, 4-bit token length indicator for the variable-length token, 8-bit message code and 16-bit message ID. The header is followed by the token value, which is used to correlate requests and responses. Token value is followed by options if there are any. Following the options comes the Payload Marker (0xFF) to indicate the end of options and the start of optional payload. This compact header design decreases the overhead significantly, which leads to decreased energy consumption and response time for constrained devices [13].

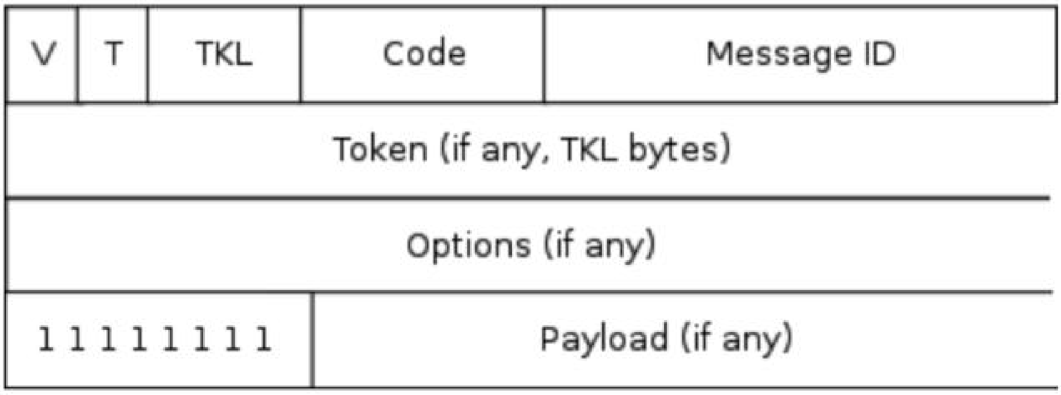


Figure 1.4: CoAP Message Format

The efficient and conservative characteristics of CoAP can enable devices operating in poor signal quality to send their data reliably. Despite CoAp's ability to run on small devices, it supports networks with billions of nodes.

1.3.2.3 MQTT:

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol that provides resource-constrained network clients with a simple way to distribute telemetry information. The protocol, which uses a publish/subscribe communication pattern. It was created as a cost-effective and reliable way to connect monitoring devices used in the oil and gas industries to remote enterprise servers [14].

MQTT is one of the most commonly used protocols concerning IOT. It enables resource-constrained IOT devices to send, or publish, information about a given topic to a server that functions as an MQTT message broker. The broker then pushes the information out to those clients that have previously subscribed to the topic. To a human, a topic looks like a hierarchical file path. Clients can subscribe to a specific level of a topic's hierarchy or use a wild-card character to subscribe to multiple levels.

The packet consists of 2-byte fixed header + a variable header and a payload (see figure 1.5). Fixed Header consists of two bytes: the first byte contains the message type and control flags such as the Duplicate flag (DUP), QoS (security level) and RETAIN flag. The second field contains the remaining length field which is, length of variable header + the length of the payload.

A variable header is not present in all the MQTT packets. Some MQTT commands or messages use this field to provide additional information or flags and they vary depending on the packet type. A packet identifier is common in most of the packets types.

In the end, the packet may contain a payload. Even the payload is optional and varies with the type of packet. This field usually contains the data which is being sent.

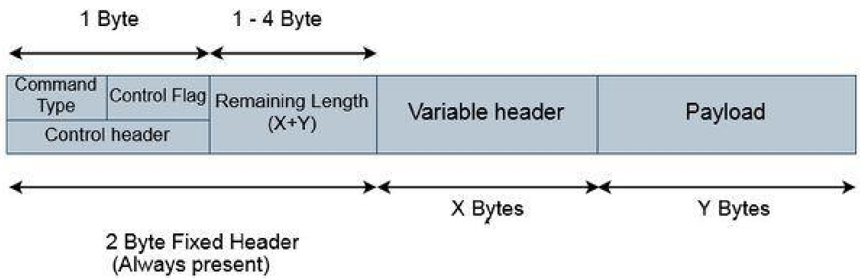


Figure 1.5: MQTT Message Format

The benefits of using MQTT in IOT are:

- Lightweight code footprint: Devices need only a few lines of code in order to get up and running with the MQTT protocol.
- Minimized data packets: MQTT is very energy-efficient. This is great if a device is battery-powered or has little CPU power.
- Speed: MQTT operates in real-time, with no delays outside of QoS.
- Ease of implementation: MQTT already has libraries in programming languages such as C and Python.
- Last will and testament: If a client unexpectedly disconnects, you can set message instructions to be sent to all subscribers in order to remedy the situation.
- Retained messages: Each topic can have one retained message that a client automatically receives when it subscribes.

1.4 Data handling and storage:

IOT devices have limited computing power and storage space and, thus, they are not suitable to store and process the data continuously generated by themselves. The data generated by the IOT devices and exchanged through the communication infrastructure should be stored in proper information system. However, the information system able to handle this data differs from traditional ones and it has a specific requirement: capability to handle thousands of parallel input streams, real-time data, and fast access to storage. To satisfy this requirement new type of database systems was introduced and referred to as Time-Series Database TSDB.

TSDBs are databases optimized for time-stamped or time-series data. Time series data are simply measurements or events that are tracked, monitored, sampled, and aggregated over time [15]. The main properties distinguishing TSDBs are the time-stamp data storage and compression, data lifecycle management, data summarization, ability to handle large time series dependent scans of many records, and time series aware queries.

1.5 Conclusion

In this chapter, we have introduced the IOT technologies' principals. These include the current architectures of IOT and the use case of many connectivity solution in IOT. LPWAN was selected to be the best connectivity solution for long-range application. Furthermore, the time-series databases were proposed as storage solution for IOT.

Chapter II LoRa

Modulation Technique

1.1 Introduction :

One of the most important connectivity protocols are the LPWAN due to the important benefit that offers for long-range application. As explained in the previous chapter, LoRa is the dominant technologies in the LPWAN market. What differentiates these technologies is the modulation technique.

In this chapter, we will explore LoRa modulation, based on the Chirp Spread Spectrum (CSS), and introduce the following critical elements that control it: radio band frequency, channel bandwidth, transmission power, correction rate, and spreading factor. At the end of this the chapter, we will simulate an end-to-end communication system to evaluate the performance of this technology and the effect of varying the controlling parameters.

1.2 LoRa's Chirp Spread Spectrum

The idea behind CSS is that a sinusoidal signal of linearly varying frequency and fixed duration, called chirp (see Figure 2.1), can be employed to “spread” information over a wider spectrum than it would normally need to occupy. This uniform distribution of a symbol over a larger bandwidth provides resistance to frequency-selective noise and interferes, at the price of lower spectral efficiency. Using some additional precautions, CSS can also be more resilient to multi-path interference and Doppler effect than other more conventional modulations. Let’s assume that the available frequency band for transmission is B = [f0, f1]. A chirp can be constructed so that it increases linearly in frequency from a starting frequency fs ∈ B to that same frequency, wrapping around from f1 to f0 when hitting the end of the available band. In LoRa, a chirp’s starting frequency inside the available bandwidth is used to represent a symbol [16]. The number of bits that LoRa encodes in a symbol is a tuneable parameter, called SF. This means that a chirp using spreading factor SF represents 2 power SF bits using a symbol, and there are M = 2^SF possible starting frequencies for a chirp. A transmission’s spreading factor is also used to determine the duration of a symbol, according to equation 2.1 :

$$T_s = \frac{2^{SF}}{B} \qquad (2.1)$$

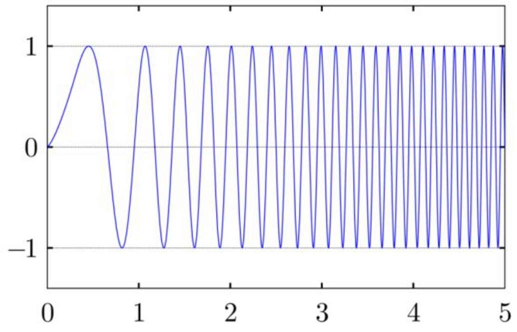


Figure 2.1:chirp signal in time domain

This means that assuming the modulation is using a fixed bandwidth, an increase of spreading factor of 1 will yield symbols that last twice the duration. Analogously, a bigger bandwidth increases the rate at which chirps are transmitted, and consequently the bitrate of the modulation. An increase in the transmit time for a chirp (i.e., a symbol) gives the message a higher robustness to interference or noise. On the other hand, this effect may be partially balanced by the fact that for higher spreading factors the number of possible symbols increases, making the occurrence of symbol errors more likely: the reason for this is that achieving synchronicity between the receiver and the signal especially critical when low data rates are employed. Another disadvantage of transmitting longer messages is the increased probability of collisions. Because of the reasons above, the choice of SF affects receiver sensitivity, which is defined by equation 2.2 :

$$S = 174 + 10 \log10 (B) + \text{NF} + \text{SNR dB}, \quad (2.2)$$

where the first term is due to thermal noise at the receiver in 1 Hz of bandwidth, NF is the noise figure at the receiver (which is fixed for a given hardware setup), and SNR is the signal to noise ratio required by the underlying modulation scheme.

Beside SNR, Receiving Signal Strength Indication RSSI is another measurement that allow us to evaluate the sensibility. Table 2.1 represents SNR and RSSI values for different spreading factors [17].

Table 2.1: SNR and RSSI values for different spreading factors

SF	*SNR(dB)*	*RSSI*
7	*-7.5*	*-123*
8	*-10*	*-126*
9	*-12.5*	*-129*
10	*-15*	*-132*
11	*-17.5*	*-134.5*
12	*-20*	*-137*

From equation 2.3, we can now get the bitrate for a certain pair of SF and B using a simple computation:

$$R_b = \frac{SF}{Ts} \tag{2.3}$$

The bitrates for a range of spreading factors and bandwidths are found in Table 2.2.

Table 2.2: Bitrate [bits/s] for a range of spreading factors and bandwidths

SF	125KHz	250KHz	500KHz
7	6835	13671	27343
8	3906	7812	15625
9	2197	4396	8793
10	1220	2441	4882
11	671	1342	2685
12	366	732	1464

2.2 LoRa data packet

LoRa frame format can be either implicit or explicit. It consists of three parts (see figure2.2) [16]: the preamble is used to detect the start of the packet by the receiver, the header is the default mode of operation it provides the payload length, the forward correction rate (FCR) and flag for the presence of the cyclic redundancy check (CRC).

This header is optional it exists only in the explicit mode packet. The third part is the payload it contain the data to be transferred and payload CRC.

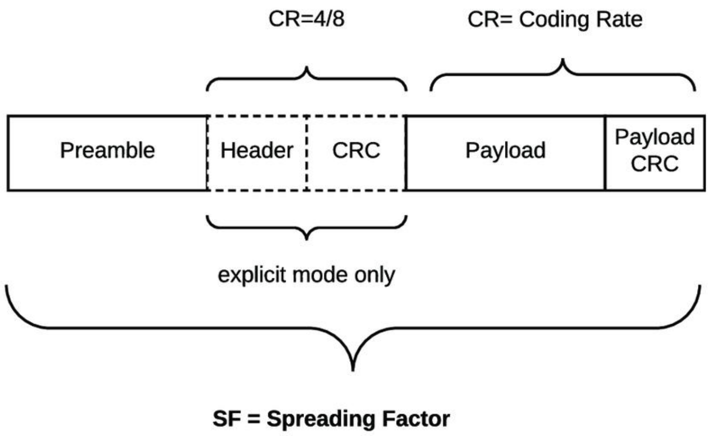


Figure 2.2: LoRa packet structure

An example of LoRa modulated packet is seen in Figure 2.3 [16]

Figure 2.3: Spectogram representation of LoRa Packet

The on-air time is the amount of time that the packet spend to reach the receiver, it can be computed with the following formula [18]:

$$T_{packet} = T_{preamble} + T_{payload} \tag{2.4}$$

where $T_{preamble}$ is the time it takes to transmit the preamble, and $T_{payload}$ is the time to transmit the actual data. These two entities have the following equations:

$$T_{preamble} = (n_{preamble} + 4.25) * T_s \tag{2.5}$$

$$T_{payload} = n_{payload} * T_s \tag{2.6}$$

where $n_{preamble}$ is a configurable parameter that affects the number of symbols in the preamble (and thus the probability that a receiver will detect an incoming packet, at the cost of longer time on-air).

The computation of $n_{payload}$ is more complicated, since it depends on Packet Length PL (number of bytes), Implicit Header IH, Low data rate optimization DE and coding Rate CR = {1 : 4/5, 4 : 4/8}. IH is 0 if the header is enabled, otherwise 1.

Implicit header reduces packet size by using predefined CR and CRC configurations, otherwise, these values are included in the frame header. If a low data rate is enabled DE value is set to 1.

$$n_{payload} = 8 + \max\left(ceil\left[\frac{8PL - 4SF + 28 + 16CRC - 20H}{4(SF - 2DE} \right] (CR + 4), 0 \right) \tag{2.7}$$

Using equation 2.7, we can evaluate the transmission performance of LoRa for different spreading factors by choosing fixed bandwidth and coding rate. Figure 2.4 show the simulation result of varying payload length over time using different spreading factors with BD=125khz and CR=4/5.

We can conclude that LoRa frame time on-air is highly depended on SF values, the higher SF the slower data flow.

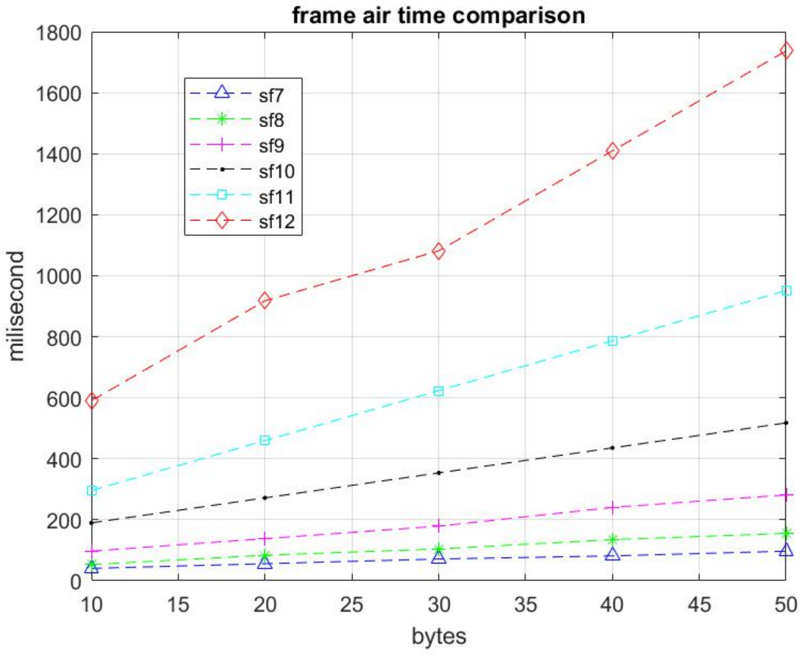


Figure 2.4: LoRa frame air time comparison for CR = 4/5 BW = 125 kHz.

2.3 Spreading factor orthogonality :

One very powerful feature of the LoRa modulation is that different spreading factors are pseudo-orthogonal, even when the same center frequency and bandwidth settings are used. This allows a receiver to correctly detect a packet using spreading factor i even if it is overlapping in time with another transmission employing spreading factor j, as long as i≠j and the received packet's Signal to Interference plus Noise Ratio (SINR) is above a certain threshold (also called isolation) that depends on both i and j. This pseudo-orthogonality between different packets allows a network employing LoRa devices to exploit different spreading factors to achieve a higher throughput with respect to more traditional modulation schemes, in which a collision can cause the incorrect reception of both the intended packet and the interferer.

2.4 Simulation of LoRa end-to-end system:

This section is devoted to simulate an end-to-end communication to evaluate the performance of LoRa in which the open source software Pothos is used.

The Pothos is a complete data-flow framework for creating topologies of interconnected processing blocks. Topologies can be designed and tested graphically, and deployed over a network.

2.4.1 System description

The design of our system is a simple node-to-node communication using LoRa transmitter and receiver. Figure 2.5 shows block diagram for our system.

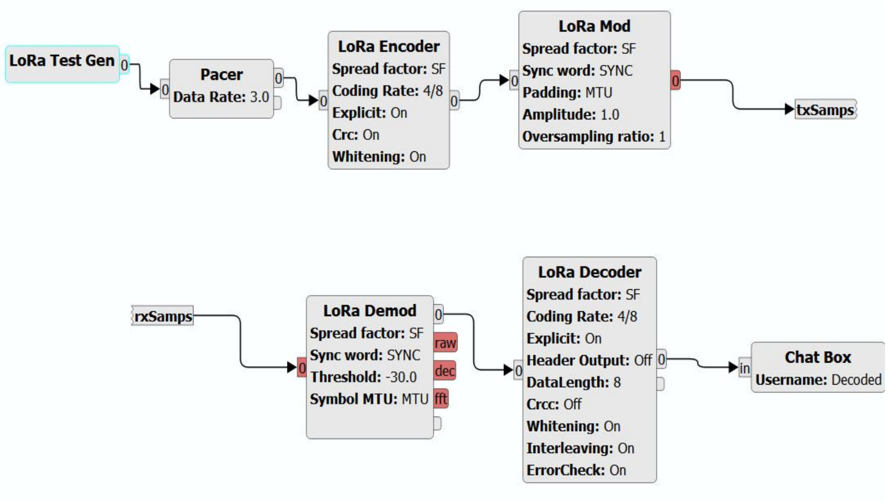


Figure 2.5:Pothos model for LoRa node-to-node transmission

The transmitter part consists of a text message generator. This is generating an incremental message for the purpose of the test. The second block is pacer that forwards the data generated by the first block to the encoder with the possibility of slowing the data flow due to the limitation in the data rate. The third block is for encoding, which transforms the data to LoRa modulation symbol. The encoding process adds the explicit header, the correction rate, and the preamble part using the defined bit rate and spreading factor. The last block is responsible for modulation. It accepts a packet of symbols for transmission and performs the chirp modulation, where detect and frame synchronization symbols are also inserted.

In the receiver part, we have demodulation block that dechirp the signal by multiplying it with the conjugate chirp. This turns each modulated symbol into regions of constant frequency. Next, take the FFT of each region. The location of the peak bin in the FFT will tell us the value of the symbol. The next block is the decoder, which is basically the reverse process. Finally, a chatbox monitors the received message.

2.4.2 Simulation Results:

The simulation result of our system is shown in Figure 2.6. It shows that transmitted and received signals are almost same only with slight degradation of the received signal.

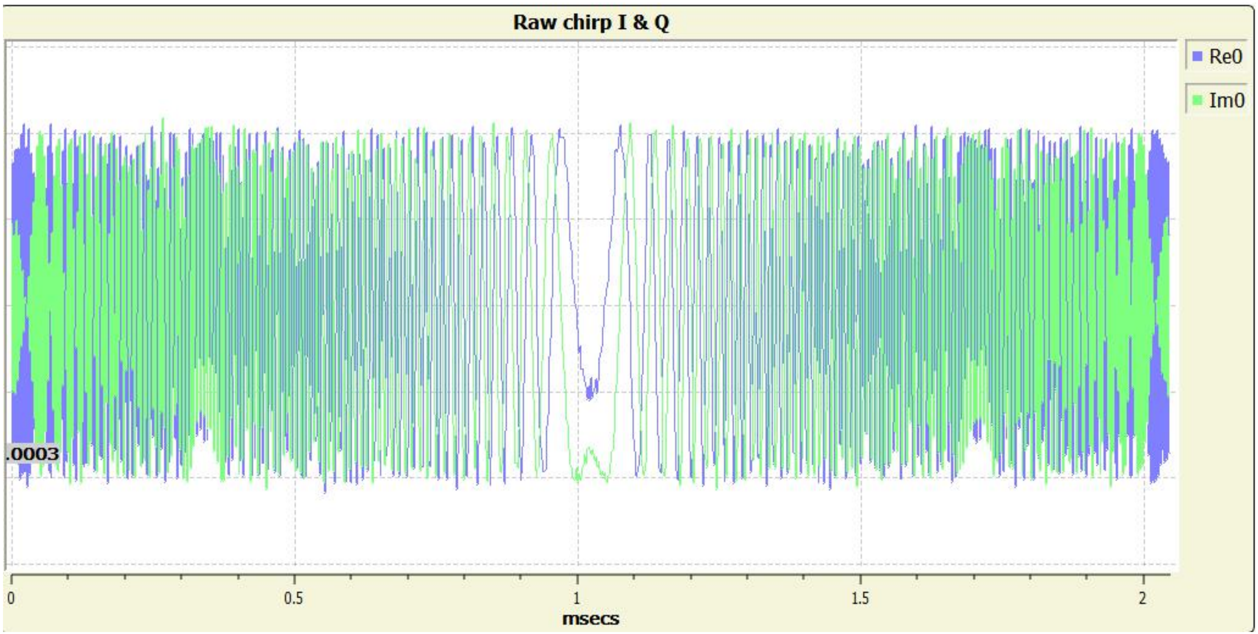


Figure 2.6: Transmitted and received signal

After comparing both the input and the output signals, another scenario is tested by varying the spreading factor values and comparing the transmitted and received signal. Figure 2.7 shows the received signal in the frequency domain in order to make reading it easier.

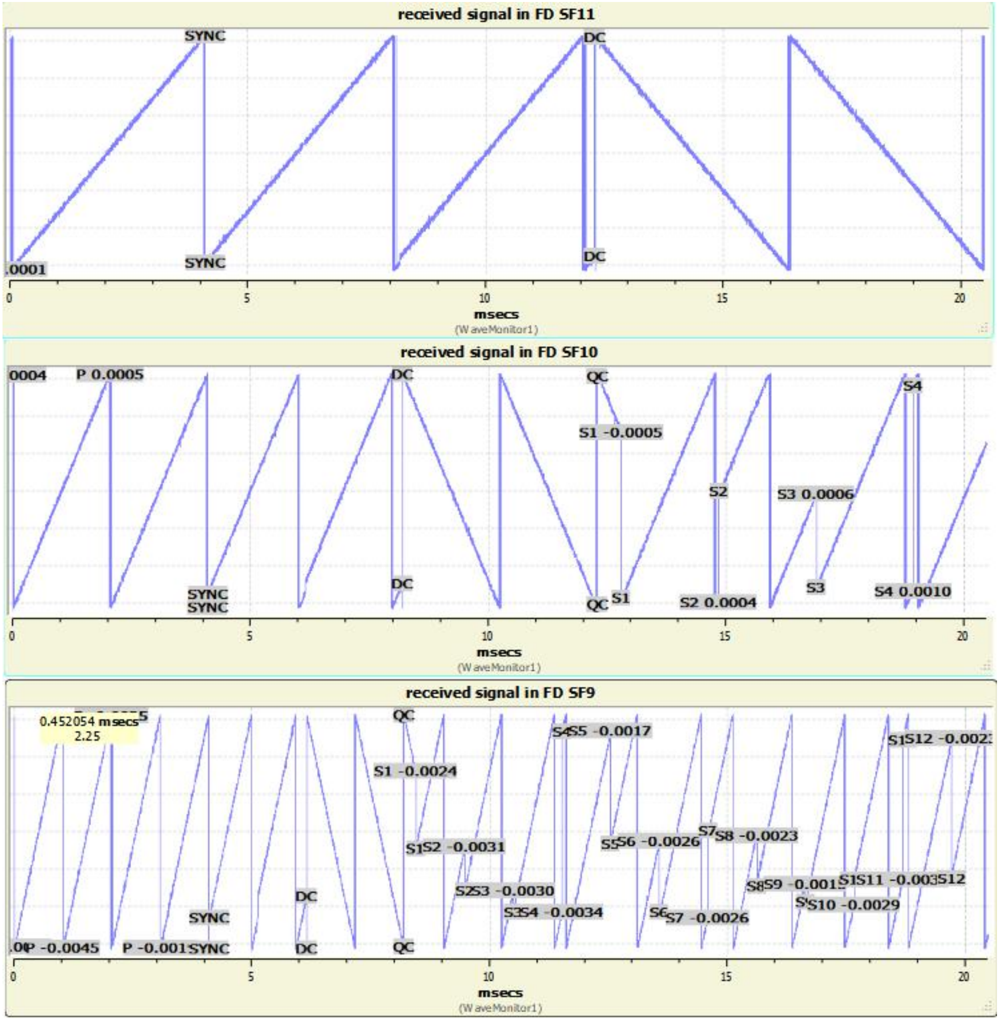


Figure 2.7: Modulated LoRa packet in frequency domain with sf=9, 10, 11

As seen in the previous figure, decreasing the spreading factor by one step doubles the number of chirps in the same time frame. This confirms the results shown in Table 2.2 and obtained using the mathematical model in the previous part of this chapter. Hence, the greater the spreading factor the slower the data rate.

2.5 Conclusion

In this chapter, we covered the chirp spread spectrum modulation technique and its application in the LoRa technology. The characterization of this modulation was studied including data rate sensitivity and time-on-air. Finally, we performed an end-to end transmission simulation to study the effect of varying the spreading fact

Chapter III Design and Implementation

3.1 Introduction:

In this chapter, we will design and implement an IOT system based on LoRa. This work will focus on the communication between the sensing layer using LoRa to benefit from its long range and low power capabilities. Then, the MQTT messaging protocol is used to send the data from the gateway to the broker. The broker will use an agent to update the database and send the data to the cloud.

3.2 System architecture:

The system is designed based on the 5 layer architecture discussed in the first chapter. The overall system architecture is demonstrated in Figure 3.1. In the bottom layer, we will design an IOT node that uses LoRa transceiver. The end nodes of the system will operate on different spreading factors to benefit from the orthogonality of this parameter. In the communication layer, a single channel LoRa gateway is used. This gateway will send an acknowledgement signal periodically. Each message is transmitted using a different spreading factor. The end node with the corresponding factor will receive the acknowledgement signal and send the data to the gateway. With this messaging technique, we can connect up to six end nodes to a single channel gateway. The latter will send the data via MQTT to the network server that contains an MQTT Broker and an agent that stores the data as time series and send it to the cloud.

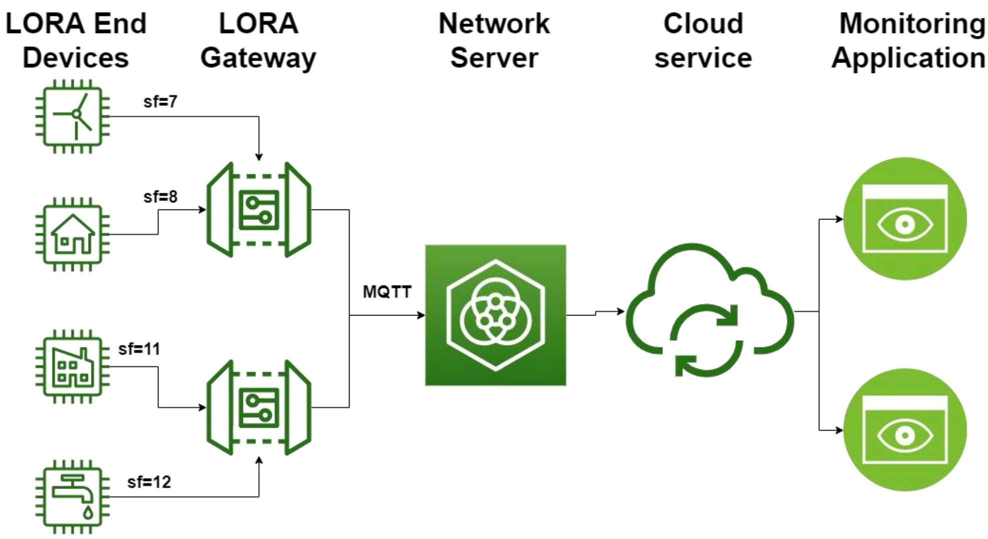


Figure 3.1: Overall system architecture

3.3 Hardware design:

In this section, we will describe the hardware component used in the design of the end nodes and the gateway:

3.3.1 LoRa node :

The hardware design of the LoRa node consists of two main part the computing module that is the Micro-controller and the LoRa modem, for the best performance we chose the following hardware:

Microcontroller:

The choice of Microcontroller is important for attaining the required results out of an embedded system and while most embedded systems developed for IOT applications consist of Low power CPUs for longevity of the device. For this reason, the STM32L4 microcontroller was chosen due to its low power consumption.

The STM32L496xx devices are the ultra-low-power microcontrollers based on the high performance Arm® Cortex®-M4 32-bit RISC core operating at a frequency of up to 80 MHz. The Cortex-M4 core features a Floating point unit (FPU) single precision, which supports all Arm® single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security [19].

This device offer up to three 12 bit ADCs , two DAC channels low power RTC and many advanced communication interfaces such as: four I2c, three SPIs, two low power UART, two audio interfaces, two CAN, and camera interface.

The variety of communication interfaces of the STM32L4 allow the user of our end node to connect many sensor and actuators.

LoRa modem:

The LoRa modem used in our node is the SEMTECH SX1278. It is half duplex low IF transceiver that offers bandwidth option from 7.8 kHz to 500kHz with spreading factor ranging from 6 to 12 covering only the lower UHF (Ultra High Frequency) frequency bands (410MHz-525Mhz) [20].

All the parameter are configurable via SPI interface that gives the access to configuration registers. These includes a mode auto sequencer that oversees the transition and calibration of the SX1278 between intermediate modes of operation in the fastest time possible. Figure 3.2 show simplified block diagram of the SX1278.

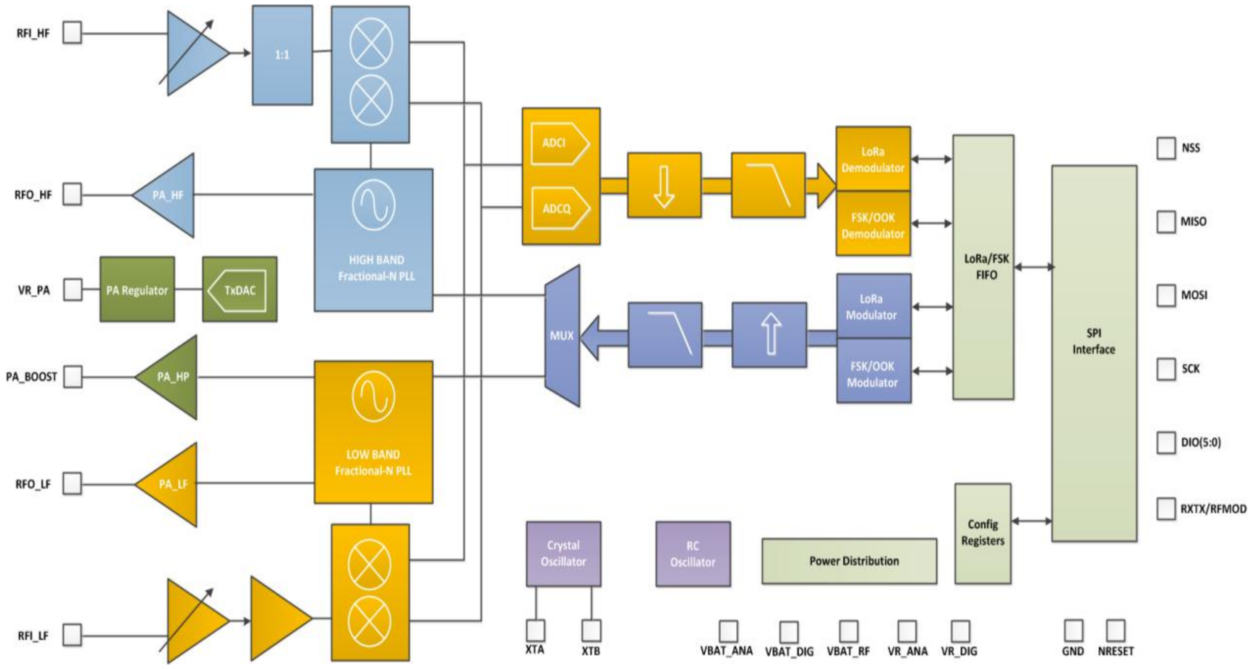


Figure 3.2: Simplified block diagram of SX1278

After selecting the hardware, a schematic of the end-node was created as seen in Figure3.3. The board contains the following:

- A USB connector dedicated for the serial communication
- A connector for the external power source or the battery
- A debugging connector used for debugging and programming the ship
- Switches to select the operation mode
- An SMA connector for the antennas
- A power regulation circuit because both the microcontroller and the LoRa module operate on 3.3v.

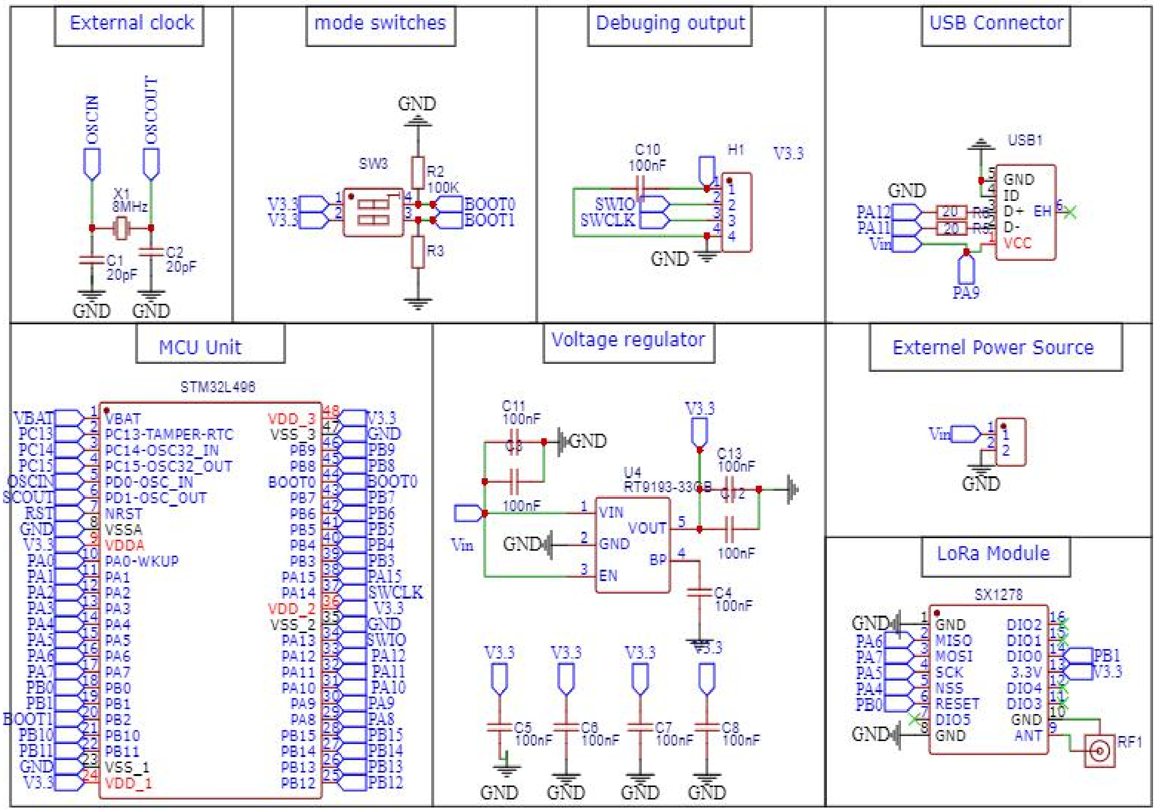


Figure 3.3: The end-node schematic

The PCB layout of the node is shown in Figure 3.4.

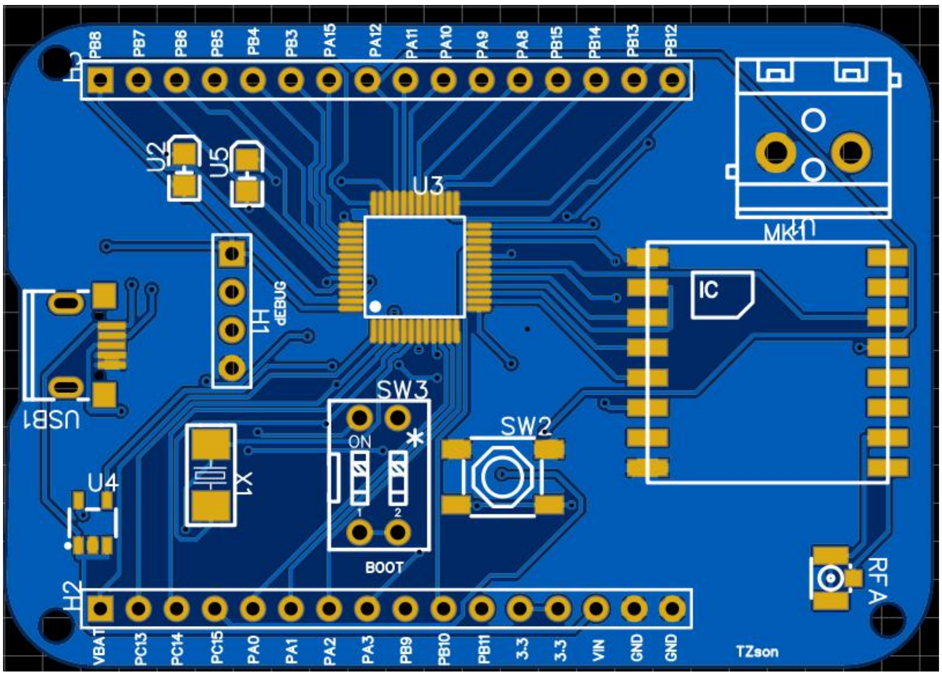


Figure 3.4: The end-node PCB layout

3.3.2 LoRa Gateway:

The gateway is primarily used in a stationary host location. Hence, low power is not a hard requirement, which makes the hardware choice easier. For this reason, a wifi gateway is used in our project. It consists of two parts: the SX1278 module and the ESP32 microcontroller that contains a wifi module to connect to the network server. The ESP 32 will be connected to the SX1278 through SPI and the DIO0 flag to indicate the existence of received message. The schematic of the gateway is shown in Figure 3.5.

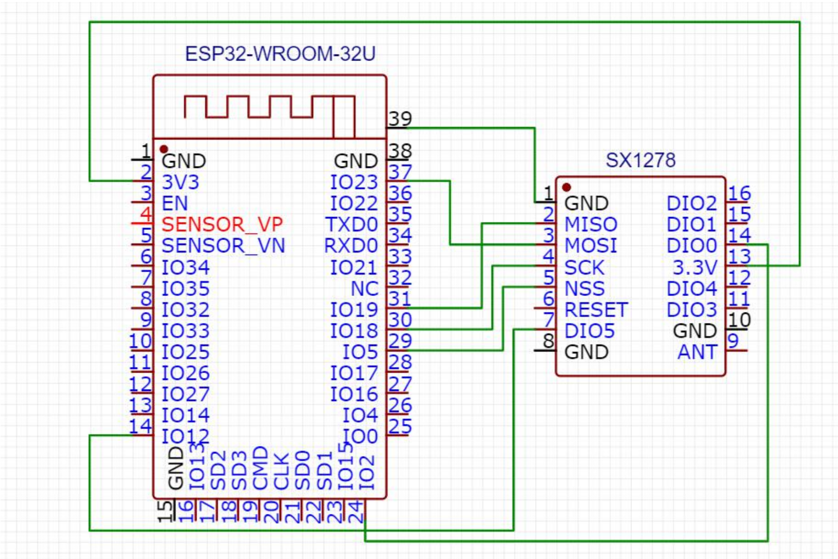


Figure 3.5:Gateway schematic

3.4 Software design:

First of all, we developed a driver for interfacing the SX1278 LoRa module to the STM32L4 where STmicroelectronics Cube IDE was used.

The operation of developing a driver was very challenging as the LoRa module has 112 registers where 68 of them should be configured to operate in the LoRa modulation mode. The other registers are dedicated for the other modulation offered by this module. The interfacing was done via SPI. The read/write operations are done in two modes; the first is called one time which read/write from the FIFO register one byte. This mode is used to access configuration registers. The second one is the burst mode, which is used in reading or writing the data packets. The number of bytes depends on the data received/transmitted. All the functions implemented in the driver of the SX1278 module are listed in the Table 3.1 with their descriptions:

Table 3.1: LoRa module driver functions and descriptions

Function	**Description**
SX1278_hw_init	Hardware pin assignment of the microcontroller for the SPI and the DIO0 flag using HAL library
SX1278_SPIRead	SPI Single byte read from fifo register
SX1278_SPIWrite	SPI burst write to fifo register
SX1278_SPIBurstWrite	SPI burst read from fifo register
SX1278_DIO0_Interrupt Handler	Interrupt configuration by reading DIO0 Pin State and clear it after each interrupt
SX1278_entryLoRa	Selection of LoRa mode in the modem over the FSK and OOK
SX1278_clearLoRaIrq	The clearance of the flag when the packet received or in timeout
SX1278_LoRaEntryRx	Configure the module for the receiving mode and the payload length
SX1278_LoRaEntryTx	Configure the module to transmitting mode and select the length of message and the sent signal power
SX1278_LoRaRxPacket	Reading the received data from the fifo register and indicating the end of the message
SX1278_LoRaTxPacket	Writing the message to be sent on the fifo register of the module
SX1278_begin	Configure the operating frequency, bandwidth and the spreading factor
SX1278_RSSI	Read the sensitivity of the received signal
SX1278_standby	Set the radio module to the standard operation mode
SX1278_sleep	Set the radio module to the low power mode, generally used during configuration.

3.4.1 The end-node software :

The software firmware of the LoRa node was developed using the drive in the previous section. The flowchart of the program is shown in Figure 3.7.

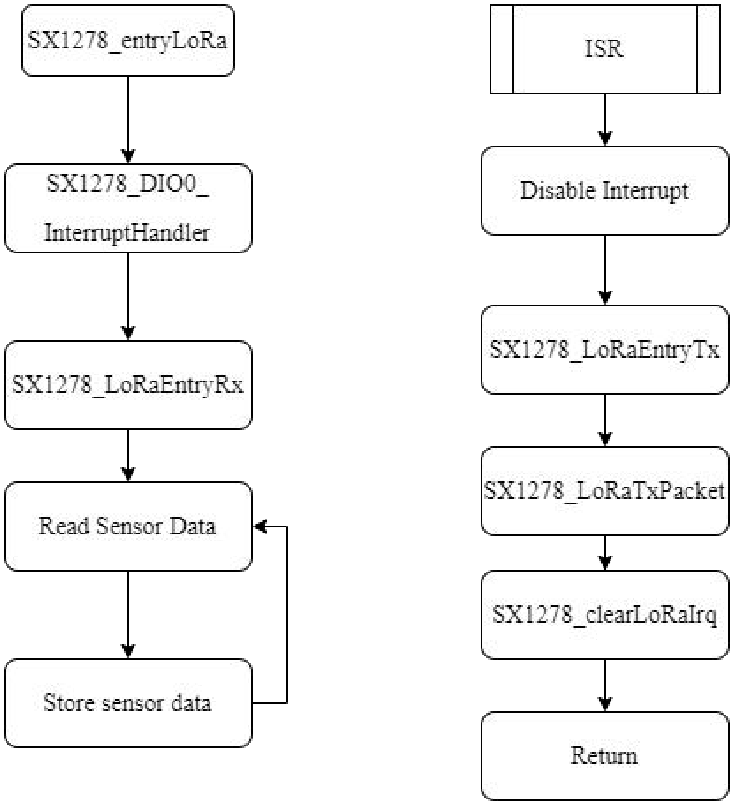


Figure 3.6: LoRa node software flowchart

Our program first configures the LoRa module as a receiver and enables the interrupt. Then, it enters into an infinite loop reading the sensors' data, saving it into an array, and waiting for an acknowledgement message from the gateway. When receiving the message an interrupt is generated and the ISR will switch the node to transmitter mode and send the sensors' data saved in the array to the gateway.

3.4.2 The Gateway software:

A flowchart of the software controlling the gateway is illustrated in Figure 3.8. First, we connect to the server via wifi and configure the gateway as an MQTT publisher. Then, we send an acknowledgment message with a specific spreading factor and wait for receiving data from the node. If the node did not transmit the data within the timeout period, the gateway will send an acknowledgment message to another node with the next spreading factor value. If the gateway get an response from the node before timeout an interrupt will be generated ,in the ISR it will switch to receiver mode, read the data from the LoRa modem, and send it to the MQTT server through the corresponding channel.

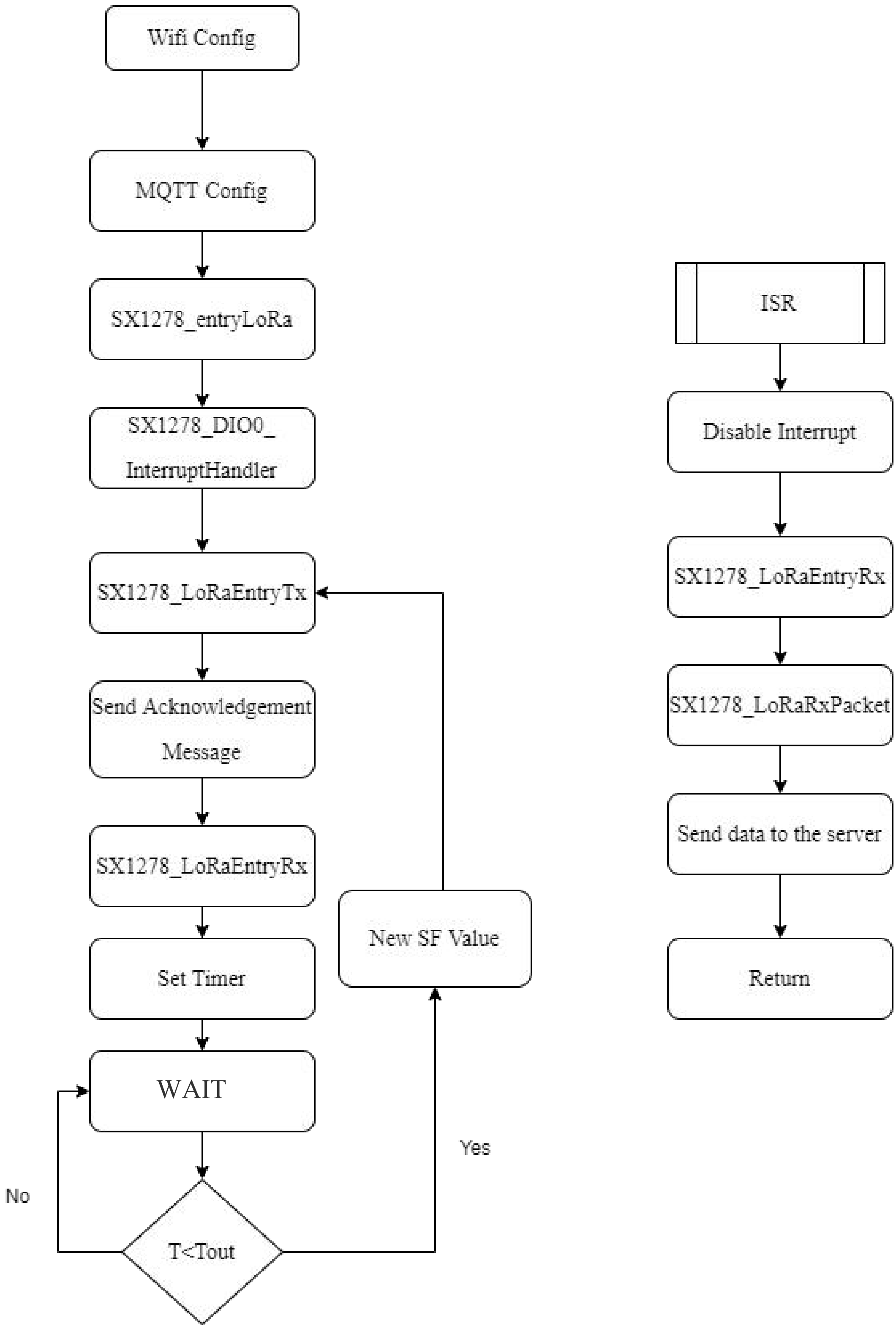


Figure 3.7: LoRa gateway software flowchart

3.5 Data Collection and storage:

In this section, we will discuss how the server collects data from the gateway and stores it as TSDB. We used an MQTT broker to collect the data from the gateway. Then, a server agent processes this data and saves it as a time series database in the cloud. The Figure below illustrate this operation.

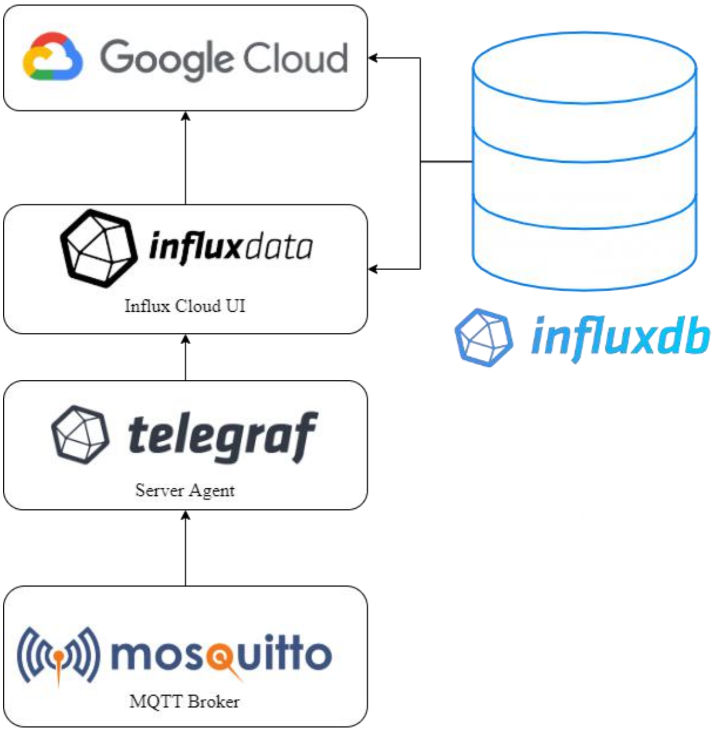


Figure 3.8: Data storage system software structure

The tools used in this system are:

- Mosquitto: is an open source message broker that implements the MQTT protocol.
- Influx Cloud: is a data service platform that relies on the time series data. We used this platform because it offers many services for IOT real-time data storage and monitoring.
- InfluxDB: is an open source time series databases that is part of the influx cloud stack. It is designed to handle high write and query loads and provides an SQL-like query language called InfluxQL for interacting with data.
- Telegraf: is a server agent for collecting and reporting data. It is the first part of the influxcloud slack that has many input plugins which is in our case MQTT.

This agent will collect data from the mosquitto broker and process this data by transforming it to influxdb line protocol by adding a time stamp and measurement name. This data will be sent to the cloud via HTTP. A configuration file for our project of the telegraf is found in appendices.

The Figure below shows the user interfaces for the influx cloud service.

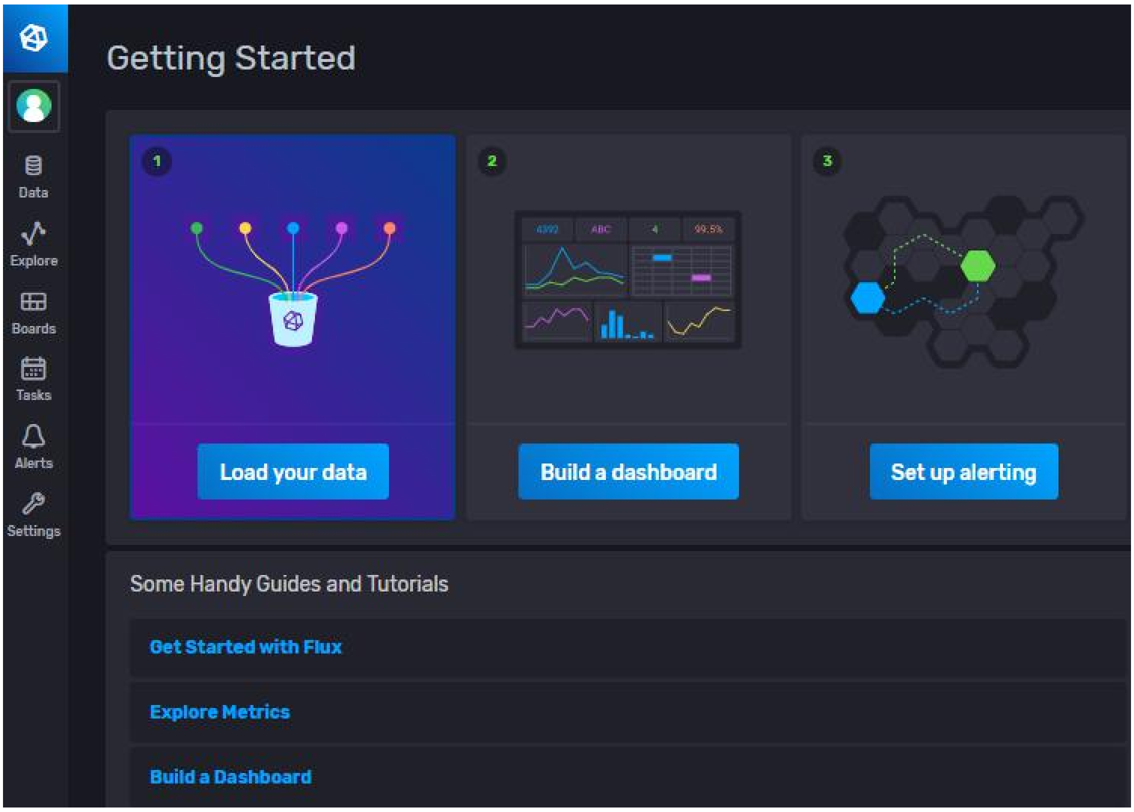


Figure 3.9: Influx cloud service user interface

3.6 Conclusion

In this chapter we covered the hardware and the software design of the network, all the hardware component used were described in details.In the software design we developed a library of the LoRa module for the STM32,the software of both the node and gateway were described.

Chapter VI

Experimental Results and Discussion

4.1 Introduction

In this chapter, we will evaluate the performance of the LPWAN system described in the previous chapter. The first experiment will test the functionality of the driver and the massaging protocol between the gateway and the end-node. The second will test the performance of the LoRa module for long-range applications.

4.2 Node-To-Gateway Test (close-range)

In this section, the node to gateway communication is tested in order to ensure the correct functionality of the software developed earlier. In this experiment, a node is within the gateway’s range in order to test the response of the node. Figure 4.1 shows the hardware used in the experiment.

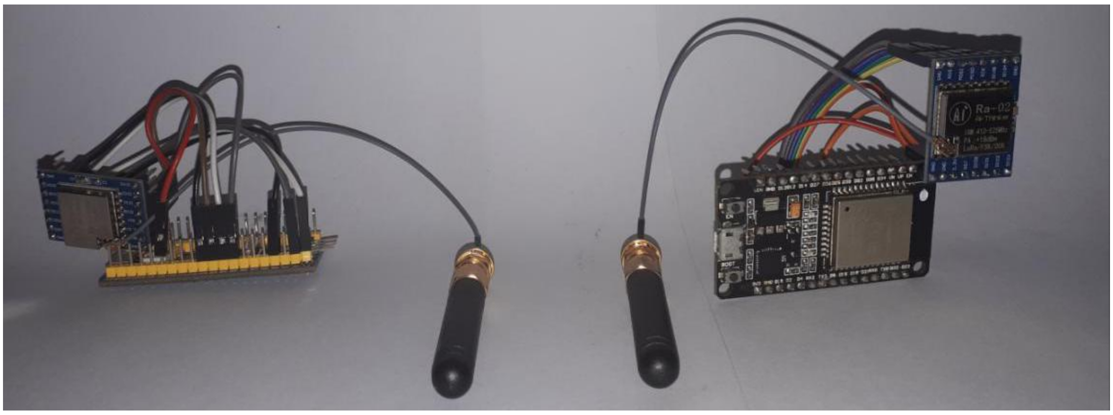


Figure 4.1: Node (left) to gateway (right) communication test (close-range)

The node designed in the previous chapter (PCB) was sent for manufacturing in china. However, due to the pandemic, the board did not reach us on time. Therefore, we replaced it by a similar STM32 MCU board and SX1278 module.

The nodes were configured to operate with SF=10, both gateway and node will operate on the frequency 433Mhz. to illustrate the communication process we will connect the gateway to computer via Serial. Figure 4.2 show reading serial data from the gateway. It can be seen that the gateway gets a response only when an acknowledgement with SF=10 is sent. Thus is because the only node available in the range operate in that SF. This experiment confirms the proper functionality of our driver and the messaging protocol. It also confirmed the principle of avoiding interference based on the spreading factor orthogonality

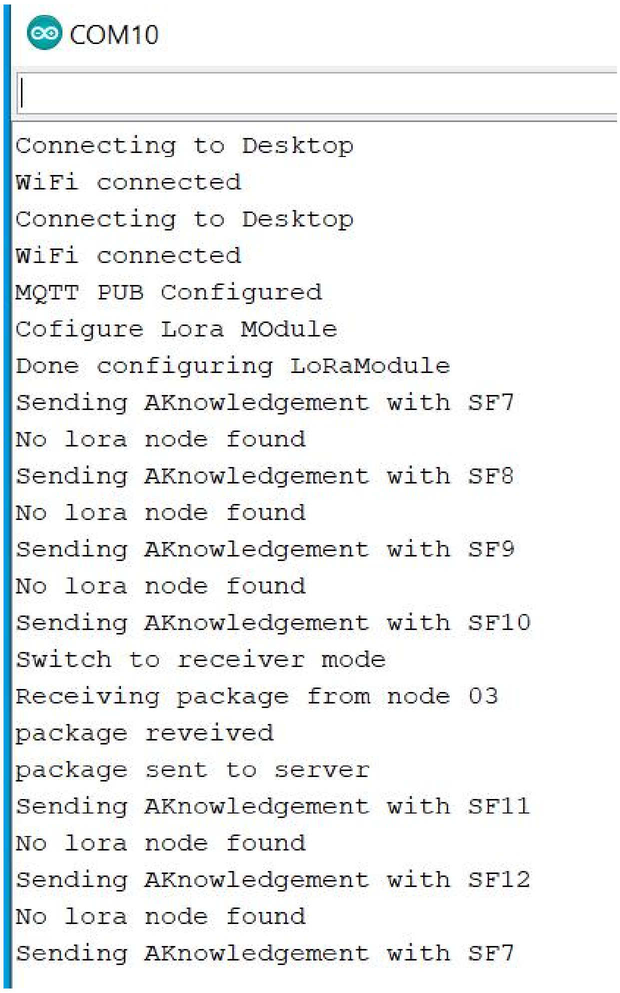


Figure 4.2: Reading serial reading from the gateway

4.3 LoRa long-range Test

The test was performed in a light urban area in Oued Rhiou town. We collected data from 3 locations such that the distance between the three locations and the receiver are approximately 540m, 1100m and 1600 , the receiver was placed at roof of a building approximately 10m higher than relative ground level. A map of the three locations and the receiver's location is shown in Figure 4.3.

At each location, we transmitted one hundred "hello" messages. Then, we counted the number of the received messages to get the Packet Error Ration (PER), whereas the receiver provided us with the RSSI.

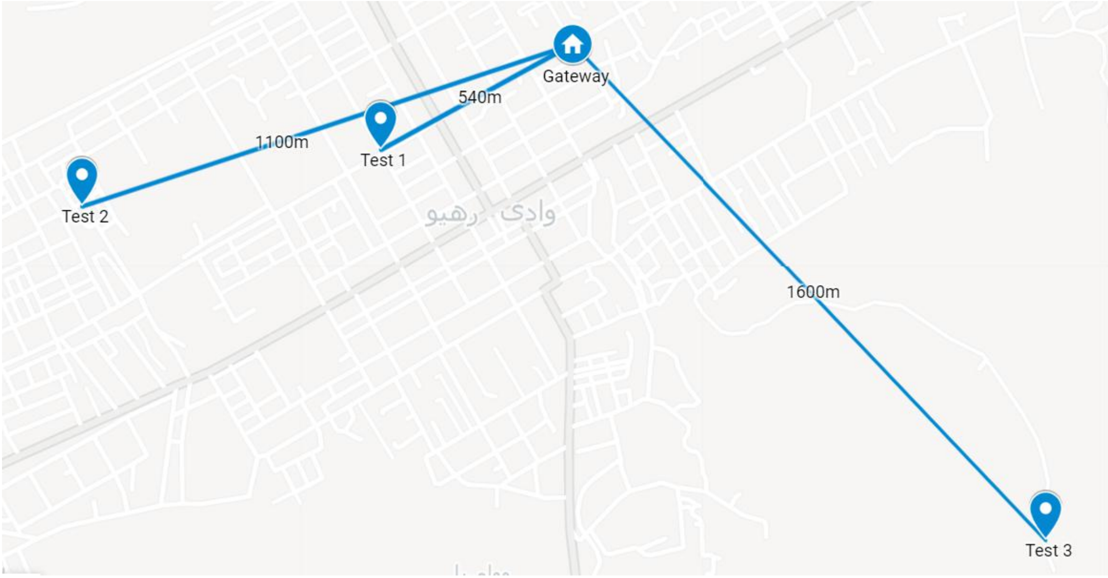


Figure 4.3: Map of the three test locations (long-range)

The table below shows the configuration used in the test:

Table 4.1: Parameters used in the long-range test

Frequency	Power	CR	BW
433Mhz	20dBm	4/5	125Khz

The measurement results are given in Tables 4.1-4.3. RSSI values given in tables are average values.

Table 4.2: Test results at distance 540m

Spreading factor	**PER%**	**Avg RSSI [dBm]**
7	2%	-115.12
8	2%	-114.25
9	5%	-117.66
10	4%	-123.33
11	7%	-130.17
12	7%	-133.66

Table 4.3: Test results at distance 1100m

Spreading factor	PER%	Avg RSSI [dBm]
7	4%	-119,60
8	3%	-117.45
9	6%	-123.14
10	6%	-124.73
11	12%	-132.65
12	11%	-135.5

Table 4.4: Test results at distance 1600m

Spreading factor	PER%	Avg RSSI [dBm]
7	/	/
8	/	/
9	9%	-127.78
10	11%	-130.23
11	11%	-134.35
12	13%	-137

The results presented confirms that our system functions correctly. The results also confirms the conclusion of chapter II, where the bigger the spreading factor the bigger error. In test location 3 (1600m), the receiver lost connection with the node at spreading factor 7 and 8. The RSSI measured values are closer to the values given by the manufacturer in Table 2.1.

It can be seen clearly that the PER increases when we use higher spreading factor in addition to the slow data rate which causes several problems when dealing with sensitive and critical data.

4.4 Discussion

- The first experiment shows the functionality of a single channel gateway. This method offers multiple node communication with a single frequency channel which make the nodes setup easier but is has some limitation in the number of nodes
- The single channel gateway receives data from one node at time , so the gateway will send acknowledgement message periodically to different nodes as consequence each node can only send data each 5 minutes waiting for the other nodes .
- The low data transmission frequency allows for long intervals where the nodes will be in receiving mode. Therefore, it saves battery and prolongs the lifetime of the node.
- The usage of high spreading factor (10, 11, 12) offers higher range coverage. But, it has a negative effect on the performance as the rate of loss is very high in addition to low data rate and higher power consumption.

4.5 Conclusion

The aim of this chapter was to test the functionality and the performance of an IoT system based on LoRa. The first experiment confirms the proper functionality of our software design, and the second experiment proves the long range capability of LoRa technology and studied the effect of spreading factor.

Conclusions

Conclusions:

The presented project dealt with the design and implementation of a prototype IoT system based on LoRa technology for sensor data analysis and monitoring. The goal of this system was to facilitate the use of LoRa technology as connectivity solution for IOT and develop method to maximize the performance with low cost and propose data storage model for this system. Both hardware and software of the system was simplified. Influx data used as data storage solution. This report provides the guidelines for designing a simple LoRa connectivity solution.

The system was tested using two experiments. The first one proved the functionality of the transmission protocol we developed to avoid collusion and simplify the configuration. The obtained results were accurate. In the second one, we tested the performance of LoRa by varying different parameters .The results were compared the theoretical values from the second chapter. This comparison showed some limitations in the case when operating in long-range.

Improvement in the design provided can be made by studying the positioning of the nodes and the gateway and enhance the transmission protocol by developing full duplex system and adding encryption to secure the data.

References

[1] Statista Research Department "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025". Nov. 27, 2016. [Online]. Available:https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/[Accessed: 29/3/20].

[2] Jan Höller et al. From Machine-to-Machine to the Internet of Things:

Introduction to a New Age of Intelligence, 1st ed. Oxford, The UK: Academic

Press, 2014.

[3] Mohammad, A., Bahareh, B., Arash, H., Parisa, A., and Farhad, N.

Towards the Internet of Things: Architectures, Security, and Applications,1 ed.

Springer International Publishing, 2019.[E-book] Available: Springer.

N. DUCROT , D. RAY , A. SAADANI "LoRa device Developer guide"

Orange Connected Object and Partnership, [Online]. Available: https://developer.orange.com/wp-content/uploads/LoRa-Device-Developer-Guide-Orange.pdf. [Accessed: 04/04/2020].

[5] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin. "A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration". ACM Comput. Surv. February 2019.

[6] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal and Q. Z. Sheng, "IoT Middleware: A Survey on Issues and Enabling Technologies," in IEEE Internet of Things Journal, .Feb. 2017.

[7] Ericsson, White paper: Cellular networks for massive iot, January 2016.

Available:https://www.ericsson.com/en/reports-and-papers/white-papers/cellular-networks-for-massive-iot--enabling-low-power-wide-area-applications (Accessed:10/4/20).

[8] M. Centenaro, L. Vangelista, A. Zanella and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," in IEEE Wireless Communications, October 2016.

[9] Semtech Corporation, “Semtech Corporation Website.” . Available:

http://www.semtech.com (Access date:24/02/20).

[10] LoRa Alliance, “LoRa Alliance Website.”. Available: https://www.lora-alliance.org/ (Access date:24/02/20).

[11] Kais M., Eddy B., Frederic C. and Fernand M. ‘A comparative study of LPWAN technologies for large-scale IoT deployment’ ICT Express. January ,2018.

[12] Grigorik, Ilya. . Making the web faster with HTTP 2.0. Communications of the ACM. , 2013.

[13] W. Colitti, K. Steenhaut, N. De Caro, B. Buta and V. Dobrota, "Evaluation of constrained application protocol for wireless sensor networks," 2011 18th IEEE Workshop on Local & Metropolitan Area Networks (LANMAN), Chapel Hill, NC, 2011.

[14] MargaretRouse “MQTT (MQ Telemetry Transport)” December, 2015) .[Online]

Availabale :’https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport’(Accessed:8/04/20)

[15] Zhang, Shuai & Zeng, Wenxi & Yen, I-Ling & Bastani, Farokh. Semantically Enhanced Time Series Databases in IoT-Edge-Cloud Infrastructure. 2019.

[16] M. Knight and B. Seeber, ‘Decoding LoRa: Realizing a modern LPWAN with SDR,’ september 2016, the 6 th GNU Radio Conference.

[17]Semtech Corporation, “AN1200.22 LoRa Modulation Basics,” May 2015.

[Online]. Available: http://www.semtech.com/images/datasheet/an1200. 22.pdf

[18] Ertürk, M.A.; Aydın, M.A.; Büyükakkaşlar, M.T.; Evirgen, H. (october, 2019)

‘A Survey on LoRaWAN Architecture, Protocol and Technologies’. Future Internet .

[19]STMicroelectronics,”Ultra-low-power Arm® Cortex®-M4 32-bit MCU” STM32L496 datasheet. Janury2020

[20]SEMTECH “SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver” SX1278 datasheet. May 2020.

Appendices

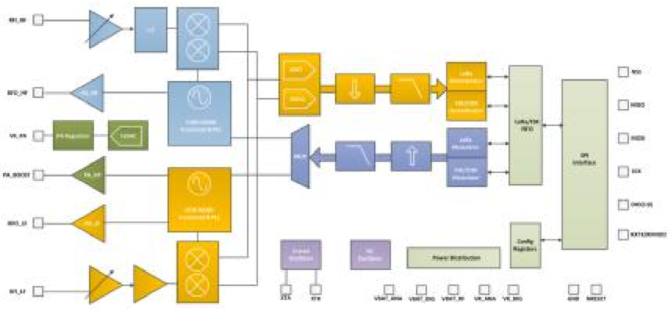
Appendix A SX1278 datasheet

SEMTECH

SX1276/77/78/79

WIRELESS & SENSING PRODUCTS **DATASHEET**

SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver



GENERAL DESCRIPTION

The SX1276/77/78/79 transceivers feature the LoRa® long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption.

Using Semtech's patented LoRa modulation technique SX1276/77/78/79 can achieve a sensitivity of over -148dBm using a low cost crystal and bill of materials. The high sensitivity combined with the integrated +20 dBm power amplifier yields industry leading link budget making it optimal for any application requiring range or robustness. LoRa provides significant advantages in both blocking and selectivity over conventional modulation techniques, solving the traditional design compromise between range, interference immunity and energy consumption.

These devices also support high performance (G)FSK modes for systems including WMBus, IEEE802.15.4g. The SX1276/77/78/79 deliver exceptional phase noise, selectivity, receiver linearity and IIP3 for significantly lower current consumption than competing devices.

ORDERING INFORMATION

Part Number	Delivery	MOQ / Multiple
SX1276IMLTRT	T&R	3000 pieces
SX1277IMLTRT	T&R	3000 pieces
SX1278IMLTRT	T&R	3000 pieces
SX1279IMLTRT	T&R	3000 pieces
SX1276WS[1]	Wafer Form	1 Wafer (2000 dies)

1. For Wafer deliveries, refer to the corresponding "Wafer Delivery Specification"

KEY PRODUCT FEATURES

- LoRa® Modem
- 168 dB maximum link budget
- +20 dBm - 100 mW constant RF output vs. V supply
- +14 dBm high efficiency PA
- Programmable bit rate up to 300 kbps
- High sensitivity: down to -148 dBm
- Bullet-proof front end: IIP3 = -11 dBm
- Excellent blocking immunity
- Low RX current of 9.9 mA, 200 nA register retention
- Fully integrated synthesizer with a resolution of 61 Hz
- FSK, GFSK, MSK, GMSK, LoRa® and OOK modulation
- Built-in bit synchronizer for clock recovery
- Preamble detection
- 127 dB Dynamic Range RSSI
- Automatic RF Sense and CAD with ultra-fast AFC
- Packet engine up to 256 bytes with CRC
- Built-in temperature sensor and low battery indicator

APPLICATIONS

- Automated Meter Reading.
- Home and Building Automation.
- Wireless Alarm and Security Systems.
- Industrial Monitoring and Control
- Long range Irrigation Systems

Appendix B

Telegraf Agent Configuration File

```
nf
outputs.influxdb_v2]]
## The URLs of the InfluxDB cluster nodes.
##
## Multiple URLs can be specified for a single cluster, only ONE of the
## urls will be written to each interval.
## urls exp: http://127.0.0.1:9999
urls = ["https://us-central1-1.gcp.cloud2.influxdata.com"]

## Token for authentication.
token = "$INFLUX_TOKEN"

## Organization is the name of the organization you wish to write to; must ex
organization = "brahimdjadir@gmail.com"

## Destination bucket to write into.
bucket = "brahimdjadir's Bucket"
inputs.mqtt_consumer]]
## MQTT broker URLs to be used. The format should be scheme://host:port,
## schema can be tcp, ssl, or ws.
servers = ["tcp://127.0.0.1:1883"]

## Topics that will be subscribed to.
topics = [
  "telegraf/Node01/Senror",
  "telegraf/+/mem",
  "sensors/#",
]

## The message topic will be stored in a tag specified by this value.  If set
## to the empty string no topic tag will be created.
# topic_tag = "topic"

## QoS policy for messages
##   0 = at most once
##   1 = at least once
##   2 = exactly once
##
## When using a QoS of 1 or 2, you should enable persistent_session to allow
## resuming unacknowledged messages.
  qos = 0

## Connection timeout for initial connection in seconds
  connection_timeout = "30s"
```

```
## Username and password to connect MQTT server.
  username = "telegraf"
  password = "metricsmetricsmetricsmetrics"

## Optional TLS Config
# tls_ca = "/etc/telegraf/ca.pem"
# tls_cert = "/etc/telegraf/cert.pem"
# tls_key = "/etc/telegraf/key.pem"
## Use TLS but skip chain & host verification
# insecure_skip_verify = false

## Data format to consume.
## Each data format has its own unique set of configuration options, read
## more about them here:
## https://github.com/influxdata/telegraf/blob/master/docs/DATA_FORMATS_INPUT.m
data_format = "influx"
```

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA - Boumerdes

Institute of Electrical and Electronic Engineering

Department of Electronics

Authorization for Final Year Project Defense

Academic year: 2019/2020

The undersigned supervisor: Dr. MAACHE Ahmed
authorizes the student(s):

DJADIR Mohammed Ibrahim Mansour Option: Computer Engineering.

to defend his / her / their final year Master program project entitled:

Design and Implementation of an IOT System Based LoRa

during the ☑ June ☐ September session.

Date: 02/11/2020

The Supervisor

The Department Head