Hugo Miguel Couchinho Marcos

LOW-POWER ELECTRICAL IMPEDANCE SPECTROSCOPY DEVICE TO ASSESS PLANTS PHYSIOLOGICAL STATE BASED ON LORAWAN WIRELESS NETWORKS

Master's Dissertation in Electrical and Computer Engineering – Specialization in Automation oriented by PhD Professor Rui Cortesão and co-oriented by PhD Professor José Pedro Amaro and presented to the Faculty of Science and Technology of the University of Coimbra.

September of 2019

Master's Dissertation

LOW-POWER ELECTRICAL IMPEDANCE SPECTROSCOPY DEVICE TO ASSESS PLANTS PHYSIOLOGICAL STATE BASED ON LORAWAN WIRELESS NETWORKS

Hugo Miguel Couchinho Marcos

Master's Dissertation in Electrical and Computer Engineering – Specialization in Automation oriented by PhD Professor Rui Cortesão and co-oriented by PhD Professor José Pedro Amaro and presented to the Faculty of Science and Technology of the University of Coimbra.

September of 2019

UNIVERSIDADE D
COIMBRA

Acknowledgements

Developing this thesis was a long, difficult and sometimes discouraging enterprise. It was not possible to complete this journey into unknown territory, without the figurative light of many different people, helping me navigate in the right direction.

First, i would like to thank my supervisors Professor Rui Cortesão and Professor Pedro Amaro, for all the support, patience and guidance when I need it the most.

Second, I must express my profound gratitude to my family, for all of the unconditional support and encouragement throughout my entire academic journey and the process of developing this thesis. To my girlfriend, Lia, who even in the most difficult and stressful moments, always endured my bad humor and patiently stood by me, encouraging me and motivating me.

To all of my new colleagues at Instituto Pedro Nunes - Laboratory of Automation and Systems for making me part of the this great team and for supporting this project.

To Lourenço, for the pertinent advice and skills that helped clarify this document.

This important mark would not have been possible without the help of all this people. Thank you!

Abstract

Diseases that affect the vegetable tissues are promoted by both biotic and abiotic factors. These are characterized by a fast spreading rate and an absence of symptomatology, causing a negative impact both in the worldwide economy and in the plant ecosystem.

The standard procedure to detect diseases and respective symptoms is restricted to a visual symptoms observation and performed only by skilled and trained personnel. Additionally, and regardless of a possible lack of capable personnel, plant visual observation is only effective in the last stages of the diseases. It becomes clear that the lack of equipment with analysis and characterization capabilities of the physiological state of vegetable tissues on-site, leaves room for the emergence of new devices.

These systems should be able to analyze the physiological state and diagnose plant diseases under different environmental conditions. Also, the absence of maintenance requirements and systematic routines are some desired characteristics for new devices, all the while being transparent to the plant growth and development.

In this thesis a complete system to diagnose plant diseases is proposed. The developed system is able to analyze the physiological state of plants based on Electrical Impedance Spectroscopy (EIS) technique. The system transfers collected data and results via Long Range Wide Area Network (LoRaWAN) wireless protocol, while having specific procedures to ensure low power consumption. In this sense, its current profile and energy consumption analysis are also presented in this work.

Keywords — Electrical Impedance Spectroscopy, LoRaWAN, Energy Consumption

Keep your face always toward the sunshine and shadows will fall behind you

— Walt Whitman,

Contents

List of Acronyms

3GPP	Third Generation Partnership Project.
ABP	Activation by Personalization.
ADC	Analog-to-Digital Converter.
ADR	Adaptive Data Rate.
AES	Advanced Encryption Scheme.
AFE	Analog Front End.
AIO	All-in-One.
AppSKey	Application Session Key.
AT	Attention.
BIS	Bio-Impedance Spectroscopy.
BPSK	Binary Phase Shifting Keying.
BW	Bandwidth.
CDMA	Code Division Multiple Access.
CE	Counter Electrode.
CR	Code (or coding) Rate.
CSS	Chirp Spread Spectrum.
DAC	Digital-to-Analog Converter.
DDS	Direct Digital Synthesis.
DEEC	Departamento de Engenharia Electrotécnica e de Computadores.
DFT	Discrete Fourier Transform.
DIP	Dual In-line Package.
DSP	Digital Signal Processing.

DSSS	Direct-Sequence Spread Spectrum.
EC-GSM-IoT	Extended Coverage-GSM-IoT.
EDGE	Enhanced Data rates for GSM Evolution.
eGPRS	Enhanced General Packet Radio Service.
EIS	Electrical Impedance Spectroscopy.
ETSI	European Telecommunications Standard Institute.
FDMA	Frequency Division Multiple Access.
FEC	Forward Error Connection.
FSK	Frequency Shift Key.
GF	Gain Factor.
GSM	Global System for Mobile communications.
GUI	Graphical User Interface.
HPC	High Pin Count.
HVAC	Heating, Ventilation and Air Conditioning.
IC	Integrated Circuit.
IDE	Integrated Development Environment.
IEEE	Institue of Electrical and Electronics Enginners.
IIP3	Third Order Input Intercept Point.
IoT	Internet of Things.
ISEC	Instituto Superior de Engenharia de Coimbra.
ISM	Industrial, Scientific and Medical.
LoRa	Long Range.
LoRaWAN	Long Range Wide Area Network.
LPWAN	Low-Power Wide Area Network.
LTE	Long Term Evolution.
LTN	Low Through-put Networks.

MAC	Medium Access Control.
MCC	MPLAB Code Configurator.
MCU	Micro-Controller Unit.
MI	Myocardial electrical Impedance.
MIC	Message Integrity Code.
MSSP	Master Synchronous Serial Port.
NB-IoT	Narrow Band Internet of Things technology.
NwSKey	Network Session Key.
OFDMA	Orthogonal Frequency Division Multiple Access.
OOK	On-Off Keying.
OSI	Open System Interconnection.
OTAA	Over-the-Air Activation.
PGA	Programmable Gain Amplifier.
PHY	Physical Layer.
PRB	Physical Resource Block.
QPSK	Quadrature Phase Shift.
RE	Reference Electrode.
RPMA	Random Phase Multiple Access.
SF	Spreading Factor.
SNO	Sigfox Network Operator.
SNR	Signal to Noise Ratio.
SSOP	Shrink Small-Outline Package.
SUT	Sample Under Test.
TIA	Trans-Impedance Amplifier.
TOA	Time On Air.
TRB	Transaction Request Block.

TTN	The Things Network.
UART	Universal Asynchronous Receiver/Transmitter.
UNB	Ultra Narrow Band.
WE	Working Electrode.
WSE	Working Sensing Electrode.

List of Figures

List of Tables

1 Introduction

The steady stream of developments regarding wireless communications is one of the main promoters of technological improvements across different industries all over the world, stimulating the global economy and leading to the increase of connected portable devices with numerous functions. Each passing year, millions of new users and devices connect themselves to the Internet, using the available infrastructures. The number of devices that are online have already passed the human population, and the trend indicates that they will continue to grow year after year [1].

Something that a few years ago started as an interaction between individuals and a global network to exchange data and knowledge, has evolved into the connectivity capabilities of almost every electronic device, ranging from HVAC (Heating, Ventilation, and Air Conditioning) to the monitoring control devices used in smart homes. This current state conceptualizes the idea of the Internet of Things (IoT), doting the traditional devices with new capabilities of image processing, sound capture and communications between them to share information and perform specific jobs.

1.1 Motivation

Exploring the "smart" and "sensorial" capabilities of a device in order to improve home or business applications is a technological area that has evolved quite a lot in the past few years. In [1] and [2] it is possible to perceive the economic growth in services and application related with IoT technologies.

This noticeable evolution is also pushed by a number of challenging issues that modern society is facing, mostly related with sustainable development, climate change, heavy urbanization and disruption of the food supply chain. Focusing on the food supply, more specifically on agriculture, problems related with the fast spread of asymptomatic forest dis-

[1] Count of worldwide connected devices, from 2015 to 2025 (in billions): https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

eases arise, most of them with no cure [3]. Adding to this already complex problem is the fact that, often, the hand-labour, equipment and techniques required for the early diagnosis of diseases are almost nonexistent. This further compromises the food supply chain and consequently the global economy [4].

There are some examples which beckon the need of advanced technical solutions to monitor and diagnose pathologies in plants, like the diseases of the Pinus Pinaster (a.k.a. Pine Trees) which are affecting wood production (for example in Portugal) [5] and the increased levels of hydric stress that vineyards have been subjected to, a result of severe climate changes which reduce the productive capacity of the plants [6].

Electrical impedance spectroscopy (EIS) is a method that studies the electrical properties of a tissue, based on the response of a sample to the injection of an electrical signal. Several studies like [7] or [8] have already proven that EIS is a powerful tool to analyze and characterize vegetable tissues. In [4] and [3] it is also possible to confirm that this impedance analysis can be used to study in-situ the physiological state of plants and evaluate the potential to being infected with diseases.

1.2 Contributions

The work presented in this thesis was subsequently incorporated and condensated, resulting in the following publication:

- J. Pedro Amaro, Hugo Marcos, Rui Cortesão, "An energy study of a LoRaWan based Electrical Impedance Spectroscopy module for tree health monitoring", 45th Annual Conference of the IEEE Industrial Electronics Society (IECON)

1.3 Objectives

This project intends to combine the monitoring and diagnosing capabilities of the status of plant species based on EIS with the long-range and free radio transmissions that Long Range (LoRa) modulation is capable of delivering in a single, portable, low-cost and low-power device. The main goal is to create a system capable of performing an impedance analysis every hour or so, 24 hours a day, with a very large battery life which can last up to 4 weeks or even some months.

In terms of conception, the system will be based on a single processing unit, which will control all of the needed computations necessary for it to be able to run and perform all of

the needed activities and procedures. The identified existing systems are based on a LoRa transceiver coupled with a power inefficient Arduino or tend to incorporate more than one processing unit, for that reason, the proposed system will present a gap in relation to its more direct competitors.

At the end of this project, the goal is to have a package, including hardware and software, that is capable of communicating with a LoRa network and perform EIS analysis, making use of only one micro-processor unit and specific chips and components.

1.4 Thesis Outline

This thesis is structured in 6 chapters. The first chapter provides a contextualization of the project that will be developed by describing the motivation and main goals of the work to be developed, and detailing the main structure of the present document.

In the second chapter the main guidelines and some details about the existing technologies for low-power wide area networks are presented. Also in this chapter, the LoRa and Long Range Wide Area Network (LoRaWAN) protocols are presented detailing the technical and functional aspects of both technologies, as a starting point for the developed work. At the end of this second chapter there is also a brief overview of the EIS technique, the most used measurement configuration and the existing relation between EIS and vegetable life.

The third chapter explains the proposed system architecture and some of its main features, along with both hardware and software choices taken during the conception phase of the project and their reasoning.

The chosen configuration was then tested and the description of this phase is presented in chapter four, where all of the implementation details related to both hardware and software aspects are comprehensively explained and detailed. The test subjects are organized in three main groups, each one characterized by some of the main core functionalities of the system like the wireless connection, impedance analysis and cyclic behaviour.

In chapter five, the most important results of the system are displayed and analyzed. The main results related to the wireless capability to communicate with a LoRaWAN gateway and some of the console print-outs of the system are presented in the first two subsections. The sections 5.3 and 5.4 illustrate the results of the calibration procedure applied to the system and display the values obtained by performing an EIS analysis to some predefined and known loads. Chapter five ends with an overview of the consumed current during a normal run of the system in order to explain the low-power approaches taken.

Finally, in chapter six the main conclusions of the work developed are presented, along with the description of both successes and possible shortcomings of the projected work. Afterwards, a specific set of possible modifications and improvements to the system will be presented, which should be addressed in later revisions and updates to the system developed during the work which culminated in the present thesis.

2 Background

This chapter presents an in-depth analysis and description of some of the most important concepts and ideas, important to the development of the proposed solution.

The first section is dedicated to introducing and explaining some of the existent low - power Wide Area Networks (LPWAN) like Sigfox and Long Term Evolution (LTE) LPWAN technologies, and detailing some of the most important technical aspects of each technology. Chapter 2.2 is exclusively dedicated to LoRa modulation and its inherent technical aspects that makes it one of the most used protocols for long distance communications. Still on the topic of long range communications, chapter 2.3 discusses the main ideas and specifications of one of the available MAC protocols that run on top of LoRa radio modulation. In this case, the focus will be directed mainly to LoRaWAN.

The general concept of EIS is explained in section 2.4, including a brief summary about impedance measurement and an analysis of some of the most used configurations to detect and measure impedance from a generic sample under test (SUT). In addition, the Appendix B is dedicated to briefly summarize some of the developed EIS systems.

2.1 Existing LPWAN technologies

2.1.1 Sigfox

Sigfox is a proprietary technology/operator for LPWANs, widely implemented in IoT applications. It offers end-to-end communication between IoT devices based on proprietary base stations equipped with software-defined radio modules, provided by Sigfox Network Operators (SNOs), and a backhaul connection to servers using an IP-based network. The end-devices that are connected to base stations use a binary phase-shift keying (BPSK) modulation in an ultra-narrow band (100 Hz) of the industrial, scientific and medical (ISM) band carrier. The frequency bands used by Sigfox are in the unlicensed spectrum of ISM bands, and differ between continents. In Europe the frequency of 868 MHz is used, in

North America the system is implemented for 915 MHz, while in Asia the 433 MHz carrier frequency is used.

This technology allows for the use of the available bandwidth in an efficient way which, allied to the low levels of noise experienced, leads to a low-cost design with high receiver sensitivity. The only significant constraint is the limited throughput of only 100 *bps*.

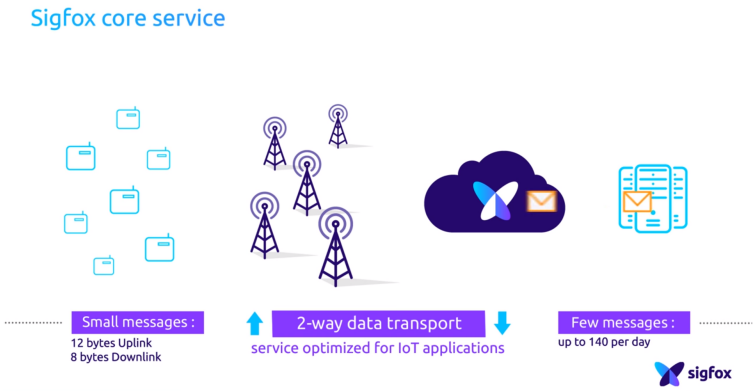

Figure 2.1: Sigfox architecture overview [9]

Previously, Sigfox only allowed up-link communication, from the end-devices to the stations, however, it later started to support the down-link communication, although with a large difference between the information sent and received. Nowadays, Sigfox offers different plans based on the amount of information that needs to be sent and received, allowing up to 140 up-link messages per day, each one with a max payload of 12 bytes [10]. Regarding to down-link communication, it is only possible to send four messages per day with a payload of eight bytes, making it impossible for the end-device to acknowledge all the successfully received messages.

The down-link communication is only possible after an up-link transmission and the end device must wait for a response from the base station. In order to ensure communication reliability between nodes and stations, Sigfox uses time and frequency diversity, as well as transmission duplication, allowing each end-device message to be transmitted multiple times (three times by default) over different frequency channels. Since the technology uses a very narrow band of 100 Hz to exchange messages, this allows for the slotting of 40 kHz of ISM band into 400 orthogonal channels [11]. Given that base stations can receive messages simultaneously over all channels, the end device can then randomly choose a slotted channel to reliably transmit its messages and expect them to be received without the need to exchange acknowledgements (ACKs). This simplifies the end-device design and consequently reduces its cost.

To summarize the main aspects of this wireless technology, figure A.1, displays the most important concepts of Sigfox, along with some key aspects and the main drivers of this LPWAN.

2.1.2 NB-IoT

NB-IoT is a Narrow Band IoT technology for LPWANs, specially conceived to work with existent cellular technologies, like global system for mobile communications (GSM) and LTE under the licensed frequency bands (700 MHz, 800 MHz or 900 MHz).

This technology occupies one GSM channel of 200 kHz or consumes one physical resource block (PRB) of LTE, which consists of a slot of 180 kHz, depending on the operation mode selected for deployment. NB-IoT can be configured in three different modes, displayed in figure 2.2, enhancing the deployment flexibility based on the available spectrum and the required application. [12].

Figure 2.2: Configurable modes for NB-IoT deployment [13]

For up-link purposes, NB-IoT is based in a Frequency Division Multiple Access (FDMA) over a single carrier frequency and enables a data-rate as high as 20 *kbps*. While in down-link it employs an orthogonal FDMA, known as OFDMA, and offers a transfer-rate of 200 *kbps*. The signals in both directions are modulated in a quadrature phase-shift (QPSK), and each message supports a maximum payload of 1600 bytes [14].

The standalone mode works based on the currently available GSM bands, acting as a replacement for one or more GSM carriers by redirecting their main application for IoT purposes. However, the Third Generation Partnership Project (3GPP), who are the developers of NB-IoT, recommend the integration of NB-IoT simultaneously with LTE cellular networks. This integration can be achieved by updating the software of previous LTE equipments. The simultaneous use of these systems allows for other operation modes such as in-band operation and guard-band operation.

For in-band operation, one or more PRBs of an LTE carrier are reserved. In this reserved region, NB-IoT signals must obey to some restrictions and should not be transmitted in time-

frequency resources reserved for regular LTE signals. Despite the separate nature of NB-IoT and LTE, these can be supported using the same eNB – hardware device connected to the network which communicates with mobile handsets [13].

When NB-IoT is configured in guard-band operation mode, it will be deployed within the guard-band of an LTE carrier. The NB-IoT carrier is contained within the guard-band and its center frequency can have at most a 7.5 kHz offset from the 100 kHz channel raster. To separate the NB-IoT and LTE signals, a sub-carrier spacing is used, while maintaining the orthogonality with LTE.

This communication protocol is similar to the LTE protocol, consisting of a lighter version with minimum functionalities, directed to enhance IoT applications. It uses the LTE back-end to broadcast information for all end devices within a cell. This system is optimized to consume the least amount of power possible from the different devices, as a result of the small messages exchanged with infrequent time-stamps and the removal of features that are not needed in IoT context, for example measurements to monitor the channel quality, carrier aggregation, and dual connectivity. These optimizations make the end-devices efficient and low-cost, achieving a battery lifespan of up to 10 years for a transmission rate of 200 bytes per day [15]. The main aspects of this technology are condensed in the figure A.2.

2.1.3 Ingenu RPMA

Ingenu, also known as On-Ramp Wireless, is a proprietary LPWAN technology that operates in a 2.4 GHz ISM band available worldwide, and, unlike other LPWAN technologies, does not rely on its propagation properties by trading range for throughput and network capacity. This, coupled with less strict regulations, enables devices to send large amounts of information when compared with other sub-GHz competitors. This capability is achieved thanks to the use of a patented physical (PHY) and medium (MAC) access scheme called Random Phase Multiple Access (RPMA) and is based on the use of direct-sequence spread spectrum (DSSS) with multiple access, consisting in a variation of Code Division Multiple Access (CDMA).

The information that needs to be sent is encoded at an 1/2 rate, and the resulting stream is modulated using a D-BPSK scheme and spread with a spreading factor (SF) that can vary in powers of 2, from 512 to 8192. The SF variation is in direct relation with the processing gain – each time the SF is doubled, the gain is increased in 3 dB [16].

Despite of being used for up-link transmissions, it enables multiple transmitters to share

a specified time slot by widening the time slot of a traditional CDMA and scattering the channel access, adding a random offset for each transmitting device.

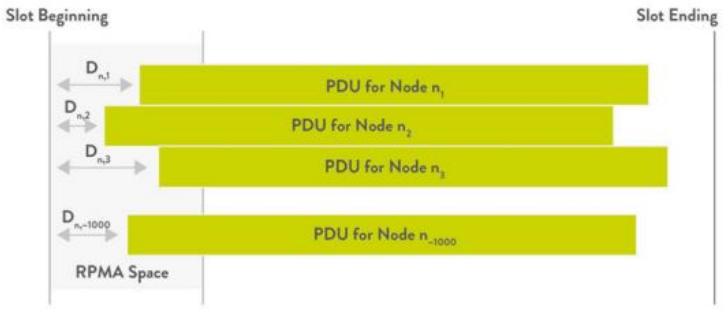

Figure 2.3: Implementation concept of RPMA scheme [17]

In figure 2.3 it is possible to see that the RPMA protocol does not grant access to the communication channel at the beginning of the time slot for all connected devices, reducing the information overlap between concurrent transmissions and increasing the signal to noise ratio [17]. On the base stations, the time shifted signals are received using multiple demodulators. Ingenu guarantees that even if half of the transmitted message packets drop out, the sent messages can still be decoded, ensuring the demodulation of 1200 overlapping signals [17].

This LPWAN technology also has down-link capabilities, albeit with link asymmetry. In down-link mode, the base stations spread the signals for individual end-devices and broadcast them using a CDMA scheme.

Performance-wise, RPMA is capable of achieving up to -142 dBm receiver sensitivity and 168 dB link budget [17]. The end-devices have the ability of adjusting the transmission power in order to communicate with base station without introducing high levels of interference to nearby devices. This is done by deciding, unilaterally, the optimal spreading factor to transmit, based on measurements of the down-link signal strength [18].

2.1.4 Telensa

Telensa is another available solution for LPWAN applications which provides a bi-directional complete package for control and monitoring of IoT applications. The communication between end-devices and base station is modulated using a proprietary Ultra Narrow Band (UNB) technique, which operates in a license-free Sub 1 GHz ISM band at low data rates. One base station can serve up to 5000 nodes/end-devices and cover a radius of $2kM$ for urban environments and $4kM$ for rural areas.

Employing a smart lighting application, even if the end-device loses its connection to the base station, it continues to operate as programmed [19]. Telensa estimates that a node should be capable of achieving a battery lifetime up to 20 years and aims to standardize the technology using European Telecommunications Standards Institute (ETSI) Low Throughput Networks (LTN) specifications which allow for easy integration within applications [20].

2.1.5 EC-GSM-IoT

Extended coverage GSM IoT (EC-GSM-IoT) is another licensed solution for a wide-area network technology that works in the same licensed frequency spectrum as detailed in 2.1.2 and was developed by 3GPP. It is based on Enhanced General Packet Radio Service (eGPRS), and is designed to improve the existing GSM networks by bringing Enhanced Data rates for GSM Evolution (EDGE) and enabling improvements in signal coverage of up to 20 dB. Despite the advantages in signal coverage, this technology also shifts the complexity of the solution to the communication infra-structure, reducing the complexity of end-devices while facilitating a reduced power consumption and low latency communication.

This IoT protocol can be implemented in the existing GSM networks, by performing a software upgrade on the radio and core networks and defining new control/data channels mapped over legacy GSM [21], and allowing for the support of up to 50,000 devices per cell, in a single transceiver .

The dichotomy between legacy GSM and EC-GSM-IoT allows for a progressive emergence of IoT concepts in the existing GSM network without requiring any allocation of resources, while making use of striking some of the most important aspects of IoT like global coverage and cost advantages [22]. The key concepts about this technology are displayed in a synthetic manner on the table of figure A.3.

2.2 LoRa - Long Range Communication

Long Range communication (LoRa), is a modulation technique developed by Semtech, that uses unlicensed radio spectrum in the ISM radio bands to transfer information with a low transfer rate at very long distances. [23] This technology, developed primarily for IoT networks, aims to eliminate the use of signal repeaters, improve the network capacity and support a large number of connected devices. The connected devices should have long lasting battery life under normal usage conditions, and a reduced production costs.[24]

LoRA uses a specific DSSS technique called Chirp Spread Spectrum modulation (CSS), where each chirp generated corresponds to one bit sent. This way, LoRa differs from other wireless technologies which use a different modulation principle, based on Frequency Shift Key (FSK). When employing this type of modulation, there is a trade-off between transmission capacity and long distance sensitivity – decreasing the former to boost the latter. This is done by encoding the information in a chirp signal that varies the frequency over time, ensuring low power requirements in transmission and robustness to signal degradation. An advantage of this method is the reduced complexity of the receiver, since the existing offsets in time and frequency modelling are the same for both. [25]

Signal transmission uses a large frequency band to reduce interference and handle frequency offsets caused by low cost crystals [26]. The transmitted signal is encrypted and it is resistant to multipath fading and Doppler effects, robust to interferences and jamming attacks; this makes its decoding difficult for any device other than the receiver. This receiver is capable of receiving signals 19.5 dBs below noise level [25]. The frequencies which are generally used for LoRa communications are displayed in figure 2.4, for a myriad of different countries.

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 +8	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kpbs				
Link Budget Up	155dB	154dB				
Link Budget Dn	155dB	157dB				

Figure 2.4: Frequency Band specific for LoRa use.[27]

In order to ensure the proper use of a LoRa module and regardless of the selected frequency band, three additional parameters other than carrier frequency need to be set. These are the code rate (or coding rate), SF and bandwidth (BW). Selecting and setting them properly will dictate the power consumption, transmission range and resistance to noise of the end device, thus the performance of the device may be optimized by fine-tuning these three selectable parameters.

2.2.1 Code Rate

The LoRa modulation contains a Forward Error Correction (FEC) mechanism to deal with data corruption and interference, known as Code Rate (CR). It can assume values between 0 and 4 in direct correspondence with LoRa coding rates, displayed in table 2.1.

Table 2.1: Code Rate (CR) and respective Coding Rates

Code Rate (CR)	1	2	3	4
Coding Rate	$\frac{4}{5}$	$\frac{2}{3}$	$\frac{4}{7}$	$\frac{1}{2}$
Redundant Bits	1	2	3	4

The CR controls the number of redundant bits used in the transmission. Selecting CR=0 means that is are no redundant bits hence no FEC.

Contrarily, a CR=4 represents a coding rate of $\frac{1}{2}$ –2 bits are being transmitted, one contains information to be sent and the other one redundant data. It can therefore be seen that while a higher CR offers more data robustness and protection, enabling the receiver to detect and correct possible errors in the message, it however increases the time-on-air per data-set packet, reducing the effective data rate. The same conclusion can be reached by analyzing figure C.1, where the effective data rate of the transmission is related with different values of CR.

2.2.2 Spreading Factor (SF)

One other parameter that needs to be adjusted for correct data transmission to be achieved is the Spreading Factor (SF) which establishes the ratio between bit data rate and chirp rate. It can be seen as the selected duration of the chirp to send a given symbol – a shorter chirp in time enables a shorter time-on-air for each packet sent.

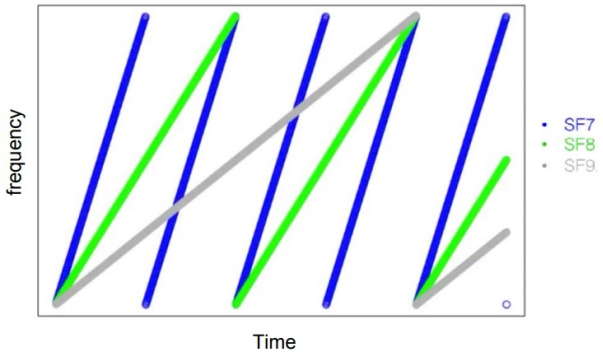

Figure 2.5: Relation between spreading factor and chirp time.[28]

In figure 2.5, it can be seen how increasing the SF value will increase the chirp time – each step in the SF scale will double the time necessary to transmit the same amount of data. A higher SF will increase the signal to noise ratio (SNR) and sensitivity, resulting in a better transmission range, however, the amount of information sent is reduced and time-on-air is increased.[26]

The number of chips needed to encode a symbol is given by 2^{SF} and for the case of LoRa spreading factors between 6 and 12 are accepted. The use of lower SF requires special coding, like the use of implicit headers [26]. The most used SFs and respective chirp length are detailed in table C.1.

Table 2.2: Spreading Factor and Chirp Length in LoRa

Spreading Factor (SF)	Chirp Length 2^{SF}
7	128
8	256
9	512
10	1024
11	2048
12	4096

This process of encoding information, more specifically symbols, in frequency chirps means that the SF is directly related with transmission data rate.

$$R_b = SF * \frac{1}{\frac{2^{SF}}{BW}} [bits/sec] \tag{2.1}$$

In formula 2.2.2, SF represents the spreading factor and it may vary between 7 and 12, and BW is the bandwidth of the communication which can assume values of 125 kHz, 250 kHz or 500 kHz. These two parameters need to be chosen so as to balance the effective data rate and transmission range [29], according to what is desired. A study on how changes to the SF impact the transmission data rate can be observed in figure C.2

2.2.3 Bandwidth (BW)

The bandwidth (BW) in LoRa in one of the most important parameters because it will define the frequency band used for the exchange of information. The usual bandwidth values vary geographically (from country to country) but the most common bands are 125 kHz, 250 kHz and 500 kHz.

According to figure 2.6, a higher BW enables more chirps per second to be sent, this is

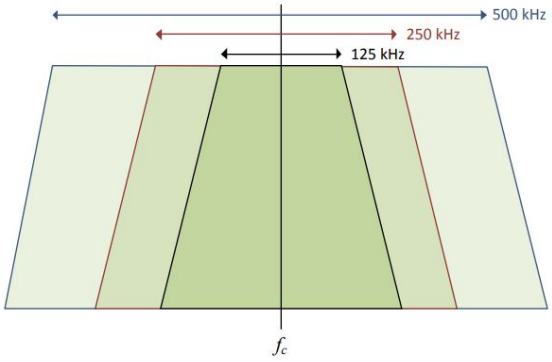

Figure 2.6: LoRa Bandwidth (BW) diagram [30]

because data is transmitted at a chirp rate equal to the selected BW. For example, if the transceiver is programmed with a BW of 250 kHz, this means that it will generate 250 $kcps$ (kilo-chirps-per-second) when communicating.

A larger BW means more chirps generated per second, leading to a higher data rate and consequently, a shorter time in the air for sent information. However, an increase in BW leads to the integration of additional noise power in the channel and leads to a reduction in sensitivity on the receiver side [30].

Using the same SF the system can either make use of a narrow BW, which maximizes the channel sensitivity but compromises the time on air, or adopt a wider frequency band, resulting in a faster transmission with data getting less Time on Air (TOA) and subsequently dealing with the reduced sensitivity of the device. The influence of transmission BW in information TOA can be analyzed in table C.1.

2.2.4 LoRa Packet Structure

In order to enable the correct communication between LoRa devices, the information must be compiled in a defined structure in order for the devices to decode the received bytes. To achieve this, the data transmitted via LoRa modulation is segmented and encapsulated into small data packets with a specific structure composed by 4 major fields: Preamble, Header, Payload and Payload CRC (which will be referred simply as CRC for simplicity reasons).

Preamble

The preamble field is used for synchronization purposes, for example synchronizing the receiver with the data being transmitted. By default, the preamble is configured with 12.25 symbol sequence, divided into 8 configurable symbols and 4.25 default symbols. Since it is a configurable variable field, the length of the preamble can be changed according to desired

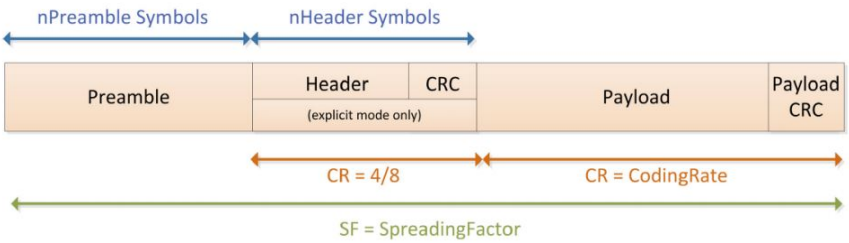

Figure 2.7: LoRa default packet structure [30]

application.

Header

This specific field can have different configurations and data contents depending on the chosen mode of operation. The configuration can be made in explicit, the default mode, or in implicit mode.

In explicit mode, the payload of the header field contains several bytes of data representing different parameters like payload length, FEC data rate and CRC. This last field is optional and composed by 2 bytes. In this mode, the header field is transmitted with FEC capability, using a CR of 4. With this CR, 4 redundant bits of information are being sent, allowing the receiver to discard both invalid headers and correct header information, if necessary [31].

The explicit mode is the default configuration for a LoRa transmission, however, in some specific applications the information to be transmitted has always the same structure and composition, meaning that parameters like payload, CR and CRC have a fixed size and/or are known in advance. These last circumstances allow for the use of the implicit mode, however, when selecting a SF equal to 6, this mode also needs to be employed.

In implicit mode the header is removed from the packet, meaning that all the needed parameters must be configured on both sides – transmitter and receiver- prior to transmission. Due to header removal the packet is smaller, thus reducing the transmission time.

Payload

The Payload field includes information specific to each type of sensor application that needs to be sent via LoRa modulation. This field has a payload capacity between 2 and 255 bytes and contains additional categories inside of it, related with medium access control (MAC) protocol like the MAC header, MAC payload and Message Integrity Code (MIC). Because all of these fields are related with MAC, implemented on top of LoRa modulation,

the composition of the payload field will be further discussed in the LoRaWAN section.

CRC

Payload CRC is an optional field composed by 2 bytes of data, used by the receiver to detect if the sent message has an error or is corrupted. If CRC is active, the receiver can know if the received message has the expected information and report the result back to the sender [32].

2.3 LoRaWAN

In the previous section LoRa modulation was addressed and its main inherent characteristics presented. Establishing a parallel with the Open System Interconnection (OSI) model for long range communications, the LoRa modulation is the first layer which must implement the physical layer in order to enable the dialog with LoRa end-devices.

The PHY layer, which is at the hardware level, defines the electrical specifications necessary for data transfer. On the other hand, the data link layer is responsible for working with the lower layer to enable the sending and receiving of information via PHY layer.

The LoRaWAN is a MAC protocol, developed by the LoRa Alliance and is mainly used for LPWANs to wirelessly connect battery operated devices, with low power requisites. It can be mapped to the second and third layer of the OSI model, as can be seen in figure 2.8.

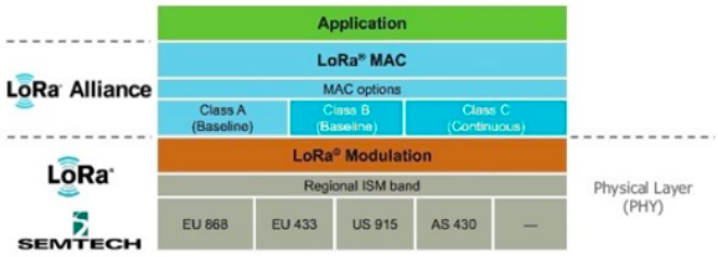

Figure 2.8: Relation between LoRa, LoRaWAN and OSI model[33]

Distinguishing between LoRa and LoRaWAN is important because other companies, like Link Labs, develop different MAC protocols and create hybrid technologies based on LoRa modulation and unrelated to LoRaWAN. The work developed by [26] is an example of a MAC protocol implementation, called LoRaBlink, which differs from LoRaWAN but is still based on LoRA modulation; this protocol enables multi-hop communication in a network of battery operated devices.

The implementation of LoRaWAN allows the devices to be mobile, low cost and bi-directional, central aspects for smart cities and industrial applications. Besides that, it is optimized to reduce the battery consumption to the minimum possible and designed to be scalable from a single gateway to a large network with millions of end devices. This scalability is only possible due to the network configuration, which is based on the star topology, visible in figure 2.9, and composed by several different entities.

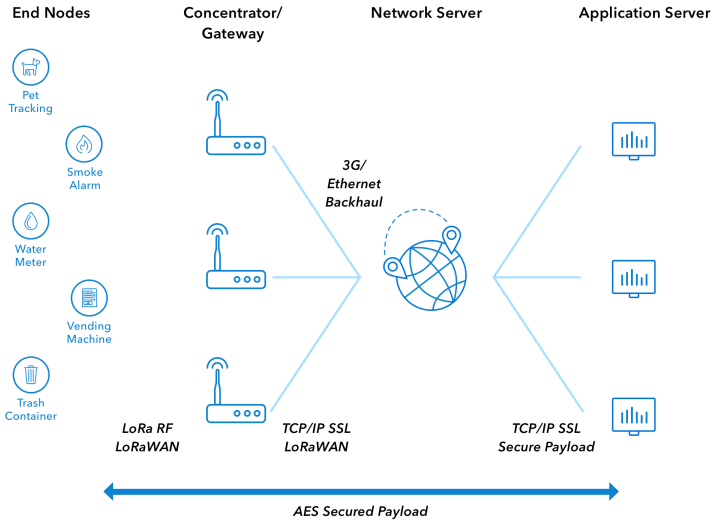

Figure 2.9: LoRaWAN star-topology arquitecture [34]

- End-Nodes: Typically IoT devices, which perform the environmental measurements are. These have embedded sensors and a LoRa module which works side by side with a microcontroller unit (MCU) to give the node the ability to send the collected information to a LoRaWAN module.

- LoRaWAN gateway: Always-on device responsible for capturing the sent LoRa packet from the end-node and delivering it to the network server.

- LoRaWAN network server: Where all of the complexity is located, making it the core of the LoRa architecture. It is responsible for managing and performing many additional tasks to ensure the correct operation of the network.

- Remote device: Can control the behaviour of the end-nodes and browse collected data. It can be a computer or any other device.

A more detailed analysis of each one of the LoRaWAN components and their respective functions in the networks structure is presented in section D.2.

2.3.1 Battery Lifetime

One of the LoRaWAN MAC features is the capability of the end-devices to be asynchronous and communicate only when they need to send information to the network, similar to the ALOHA method [27]. Other technologies like those of cellular type, require that end-nodes wake-up, periodically, and synchronize with the network to check for new messages. This approach is more power consuming that LoRaWAN because the nodes need to be transmitting in a regular basis, even if they do not have any information to send. The work developed by [35] compares the power consumption of several different communication protocols, displayed in the figure D.1. The results indicate that LoRaWAN technology has the advantage of consuming 3 to 5 times less power than concurrent technologies, making it a viable asset in IoT environments.

2.3.2 Network Capacity

The capability of the network to handle multiple devices and a large volumes of information sent and received by the gateway, is an important factor when it comes to viability and scalability of LoRaWAN. For that reason, a LoRaWAN network has specific features like Adaptive Data Rate (ADR) and a multi-channel multi-modem transceiver in gateways to improve data transfer between nodes [36]. This allows for multiple messages from the end-nodes to be simultaneous received on multiple LoRa channels. However, there are some factors that can affect the number of active concurrent channels like the data-rate, payload length, and node transmit cadence [27].

As discussed in section 2.2, LoRaWAN in based in CSS which makes the signals with different spreading factors almost orthogonal. As the SF changes, the effective data-rate also changes, this relation is used by the gateway to be able to receive multiple data-rates on the same channel at the same time, dividing the channel in virtual sub-channels.

This principle, enables transfer data-rates between the network and devices between 0.3 *kbps* and 50 *kbps*, and works best if the nodes are configured with correct SF. If an end-node is relatively close to a gateway and has a good link, best practices advise to use the highest data-rate possible. Contrarily, using a lower data-rate and filling up the available spectrum longer than it needs reduces the potential space for other nodes to transmit [27].

In order to maximize overall network capacity, the LoRaWAN network has the ability to manage the SF used for each of the connected devices employing an ADR. This functionality is only used if the end-device, connected to the network, requests it and not by network

initiative. The use of ADR is not recommended if the end device is not moving, otherwise it is advised to have it enabled.

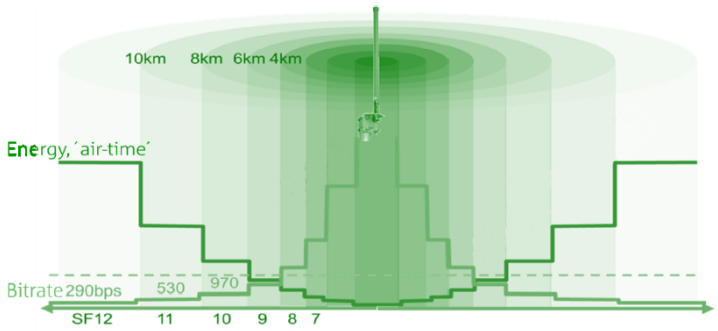

Figure 2.10: Relation between end-node distance and SF, used by ADR [37]

The mathematical procedure within the ADR algorithm analyses the last 10 received up-link packets, computes their SNR and compares the obtained value with a reference – usually the limit SNR for each SF plus a margin due to fading [37].

2.3.3 End-Devices Classes

End nodes in IoT can serve different applications, which may have different requirements related to battery life and communication latency. In order to optimize these applications, LoRaWAN enables three different device classes according to the trade-off between communication latency and battery lifetime. A large latency in communication means that the device will consume more power during transmission stage thus shortening battery lifetime. The three MAC implementations are displayed in figure D.2.

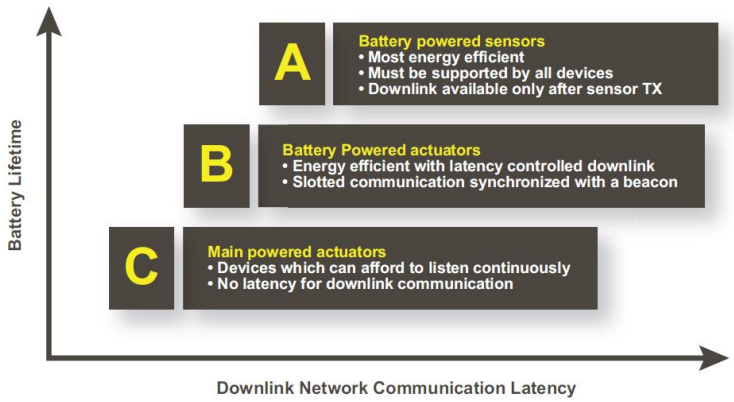

Figure 2.11: Trade-off between communication latency and battery life for available device classes [27]

Class A devices are generally used in battery powered sensors, allowing for bi-directional communication where the up-link transmission, from end-device to network, is followed by

two short timing receive windows. The first receive window comes exactly 1 second after the up-link transmission, while the second receive window is open 2 seconds after the device finishes the transmission. If the network communicates in either receive window, it can only establish communication with the device after the next up-link transmission. Otherwise, the server can respond using one of the receive windows but not both.

The Class B devices have a similar behaviour to the class A devices with the exception of an extra receive window at scheduled times. These devices, usually battery powered actuators, join the network as regular class A devices and then switch to Class B devices [10]. After that, and to inform the device to open the extra receive window at scheduled time, the gateway sends a beacon to synchronize all end devices. When the equipment receives the ping, it opens a short reception window called "ping slot", enabling the server to know when the devices are listening.

Class C devices are the main powered actuators and do not have any battery consumption constraints. This class of devices is characterized by having the minimum latency in down-link communication, when compared with other two classes. This is because they not only open two receive windows, as Class A, but also open a continuous receive window after the transmission window. These devices are only used in applications where there are not any power availability constraints and low latency communication is needed.

The three classes described above are synthesized in figure D.2, alongside the available receive windows and timings for each.

2.3.4 Network Security

The network security is an important aspect in LPWANs used in IoT, since it ensures the operation of the device in an uninterrupted and safe way. LoRaWAN implementation has two layers of security, one at the network level and one at the application level. The network layer solution comes with an authentication protocol and a security protocol, based on the Advanced Encryption Standard (AES) 128 encryption scheme, to ensure the authenticity of the node in the network. On the other hand, the application layer of security ensures the network operator does not have access to the end user's application data.

All of the security layers are considered in two security keys, all with 128 bits length, which are the Network Session Key (NwkSKey) and Application Session Key (AppSKey). The NwkSKey is used for check the validity of messages (via the message's MIC) and to map a specific device with a specific application, enabling the interaction between the end-device

and server. The AppSKey is the key that is used for the end device to encrypt the payload that will be sent to the network and for the receiver to decode the received message. This key ensures that no one but the proprietary of the application, will be able to decode/read the message uploaded by the end-node. Based on the activation protocol established between the end-device and the network, both of these keys can change in every device connection or remain static until the user changes them.

Both session keys (NwkSKey and AppSKey) are unique per device and per session. If the device is configured in Over-the-air Activation (OTAA) mode, these keys are re-generated upon every activation. This behaviour does not occur if the devices join the network via Activation by Personalization (ABP) mode, in this case the keys stay unchanged until the developer requests it otherwise.

Dynamically activated devices, which employ OTAA, use the AppKey to generate the two session keys during the activation procedure. In The Things Network (TTN) (an online platform where technical aspects related to LoRaWAN can be managed), a default AppKey will be used to activate all devices, alternatively a customized AppKey can be set per device. The algorithm used for this is the AES-128, similar to the algorithm used in the 802.15.4 standard. The NwkSKey is used to validate the integrity of each message by its MIC. For this, LoRaWAN uses AES-CMAC.

2.4 Electrical Impedance Spectroscopy (EIS)

Electrical Impedance Spectroscopy (EIS) is a technique widely used in several different applications such as microbiological analysis [38], food products screening [39][40], corrosion monitoring [41], quality control of substances [42], characterization of solid electrolytes [43], human body analysis [44] and in assessing the physiological state of plants (artigo).

Despite its broad range of applications, the working principle remains the same. There are two basic formulations for the EIS, the so-called potentiostat EIS and the galvanostat EIS [45].

The first one consists of applying an alternate voltage

$$V(t) = \bar{V} + \hat{V} \cdot sin(\omega t) \tag{2.2}$$

to an SUT and registering the alternate induced current in the sample obtained with

$$I(t) = \bar{I} + \hat{I} \cdot sin(\omega t + \phi) \quad . \tag{2.3}$$

Based on the previous two equations, it is possible to calculate the complex impedance by

$$Z(j\omega) = \frac{V(j\omega)}{I(j\omega)} = \frac{\hat{V}}{\hat{I}} \cdot e^{-j\phi} = \mid Z \mid \cdot e^{j \cdot Arg(Z)} \quad , \tag{2.4}$$

which can be represented in the rectangular form by

$$Z(j\omega) = Re(Z) + j \cdot Im(Z) \quad . \tag{2.5}$$

Here $\hat{V}$ and $\hat{I}$ represent the amplitude of voltage and current respectively, $\bar{V}$ and $\bar{I}$ are the direct (DC) values of voltage and current, f is the frequency of the test, $w = 2\pi f$ is the angular frequency, ϕ is the phase offset between voltage $V(t)$ and $I(t)$ and $V(j\omega)$ and $I(j\omega)$ the Steinmetz transforms, also known as phasor, of $V(t)$ and $I(t)$, respectively.

The galvanostat EIS is one other EIS method which is done by stimulating the test sample with an alternate current, instead of injecting an alternate voltage like the above method, and registering the voltage drop across the sample. Despite the small change in the procedure, the impedance can still be calculated using formula 2.4. In most cases, using alternate voltage or alternate current dues not produce any considerable differences in results. However, in some specific applications, like in the case of corrosion analysis, the galvanostat EIS ensures the measurement is carried out at the true corrosion potential [46].

2.4.1 Standard EIS Assemblies for Impedance Measurement

Relatively to measurement configurations, EIS can be obtained with different implementations, namely two-, three- and four-electrode implementations, displayed in 2.12.

The simpler configuration is composed by two electrodes, a working electrode (WE) and a counter electrode (CE), and both electrodes are used in both the stimulus and sensing stages. This means that, when computing the impedance $Z(j\omega)$ of the sample, the result will include contributions from the sample interface at both electrodes [45]. In order to mitigate this interference generated by the electrodes, it is recommended to add an extra electrode, also called reference electrode (RE). This leads to the configuration (b) represented in figure 2.12.

This implementation is used when is necessary to perform analysis in electrochemical

systems and differs from the two-electrode assembly because the alternate signal used to stimulate the sample is injected between the WE and the RE, while the signal sensing is measured at the CE. This leads to a lower interference contribution, since there is no current drawn at the RE. In this case the detectable interference is only caused by the WE.

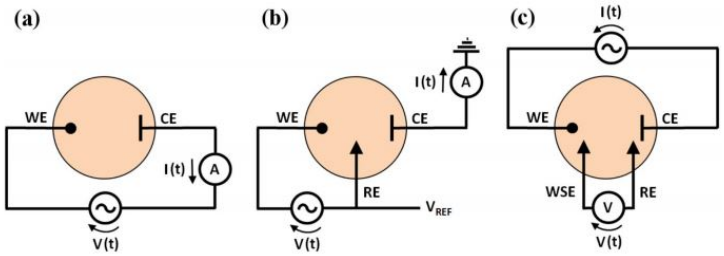

Figure 2.12: Most known configurations for EIS measurement [45]

In case of galvanostat EIS, it is necessary to add a fourth electrode, called working sensing electrode (WSE), to the previous configuration which results in a four electrode assembly – (*c*). The WE and CE are used to excite the SUT with sine-wave current, while the WSE and RE are used to capture the AC voltage generated by the sample. This setup allows for an interference-free measurement at the cost of increasing the assembly's complexity, since the there is no current drawn at both WSE and RE [45]. The main goal for the system developer is to find the best compromise between the signal interference and configuration complexity.

Other important aspect to take into consideration when performing an EIS analysis is the excitation amplitude of the signal that is transmitted to the sample. Most of the described applications apply a low amplitude signal to work in the linear range of impedance [47], however, some studies like [48] describe the behaviour of impedance in the non-linear range.

2.4.2 Impedance Representation

Some of the test samples used to perform EIS have a complex behaviour, which can be translated by a variation of $Z(j\omega)$ with frequency. In order to properly describe the impedance of a SUT, best practices advise to perform a sweep over different frequencies to check if the response of sample impedance is immutable with frequency change.

One way to graphically visualize the change of impedance response with frequency is by representing the values in a Nyquist plot.

Horizontally, the Nyquist plot enables the representation of the real part of the measured impedance, directly related with the amount of dissipated energy, while the vertical axis

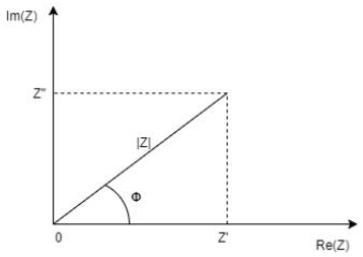

Figure 2.13: Nyquist Plot [49]

$Im(Z)$ is directly related with stored energy [49]. Both these mentioned parameters can be obtained by

$$Re(Z) =\mid Z \mid \cdot cos(\phi) \tag{2.6}$$

and

$$Im(Z) =\mid Z \mid \cdot sin(\phi) \tag{2.7}$$

Despite providing a visual tool to observe the two elements of impedance, the Nyquist plot is not ideal when it comes to represent impedances since it masks the impedance dependency on frequency, omitting important information for an EIS analysis [45]. The alternative is to plot a bode diagram, where the real ($Re(Z)$) and imaginary ($Im(Z)$) components of the impedance can also be plotted, but in this case versus the frequency of analysis, presented on a logarithmic scale.

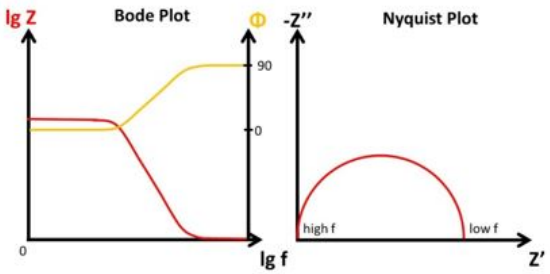

Figure 2.14: Side by Side comparison between Bode and Nyquist Plot [50]

The bode diagram of figure 2.14 allows for the perception of the change in behaviour of impedance with frequency, something the Nyquist plot does not permit. On the other hand, the bode plot is not very sensitive to changes in the measured system if the fundamental behavior of the system does not change. Both of the obtained curves are used to display the measured impedance, which depends on the electrical model employed to model it.. This equivalent model depends on the EIS application, but usually is composed by simple components, like resistors or capacitors, however, more complex elements can be used to

obtain the desired equivalent circuit in some specific applications. The work developed in [45] analyzes and explores several equivalent electric models to evaluate food quality.

Relatively to the implemented equivalent electric model in the scope of this project, the work developed in [4] and [3] will serve as a base to be used for impedance modulation.

2.4.3 Equivalent electric model of vegetable tissues

Each vegetable tissue has its own composition and biological arrangement which can be linked with its specific and unique electrical signature, mainly due to the influence of cell membranes and both intra- and extra-cellular fluids. When the tissue is affected by physiological changes or even diseases, it can be detected in the electrical field by the impedance spectrum [3].

Representing biological tissue electrically can be done based on an equivalent circuit that consists of a parallel arrangement between the resistor R_1 which represents the extra-cellular fluid with a second resistor R_2, used to model the intra-cellular fluid, in series with a capacitor C, simulating the membrane capacitance [3] [51].

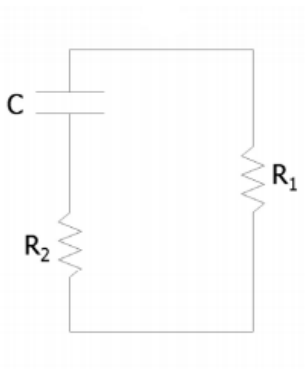

Figure 2.15: Simplified Hayden model for electrical representation of vegetable tissues [8]

The capacitive value of the modeled membrane and the resistive value of both intra- and extra-cellular fluids can be extracted using the theoretical Cole model, where the impedance spectrum obtained is called the Cole-Cole plot [52]. Other ratios like phase angle and interrelated indexes, such as the ratio between the impedance value obtained at the system's highest possible frequency versus the impedance measure at the lowest possible frequency, are widely used to extract information about biological materials and their inherent physiological condition [4].

In this thesis, it is intended to implement the equivalent electric circuit, displayed in 2.15 to conduct an impedance analysis across a wide range of frequencies. In addition, it is proposed to define the ratio between the impedance at highest frequency versus the impedance at lowest frequency as a measurement of the physiological condition of plants.

3 System Architecture

In this chapter the proposed system architecture will be described, detailing the main devices that system should have in order to fulfil the project objectives. Afterwards, the choices of components to use will be listed and analyzed to substantiate the development of a system capable of performing EIS and send the gathered data via TTN using a LoRaWAN MAC protocol. A brief analysis to the most used software tools in the context of this project are presented in Appendix I.

3.1 Proposed System Architecture

The device to be developed in the scope of this project will be composed by an MCU, an impedance analyzer chip with an AFE, a LoRa transceiver module, an external circuit with timer and "wake-up" capabilities, and a power supply circuit, consisting mainly of a low-dropout voltage regulator. A diagram of the proposed architecture may be found in Figure 3.1.

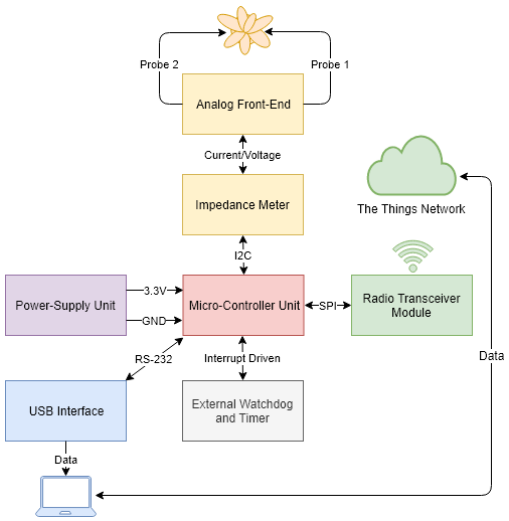

Figure 3.1: Diagram of proposed system architecture

A USB interface has also been included in the system to enable information print out via terminal. This eases the debugging process of the whole system. The USB interface

is designed so that it may be physically detached, eliminating the power drawn by this component and minimizing the current consumption, hence extending battery lifetime.

The "brain" of the system will be an MCU and it will be responsible both for the wireless communication with gateway via LoRa modulation and the communication with the impedance meter. Besides these two main functions, it should also stream data via USB interface when the device is connected to a computer.

To meet the compactness and low-power requirements of the system, the selected impedance meter should include all of the needed peripherals to allow for an impedance analysis with minimal to no additional hardware. Taking into account the information in section 2.4, the impedance meter device should include, if possible, the configurability of the excitation signal, sampling and representation of the signal generated by the SUT. These elements help minimize the complexity of the system that will be developed and guarantee that all of the components are compatible.

Considering the goal of the project is conduct an EIS analysis in (mostly) plants, some type of interface circuit between the sample and the measuring device is needed to ensure that the sample is properly excited and the generated signal is measured correctly. This is an important aspect taking into account that the quality of the results will inevitably depend on the quality of the gathered data.

In order to explore the low-power capabilities of the micro-controller, it was decided to include an external device with very low current consumption, responsible for “waking up” the device at specific time intervals. This permits the programming of the MCU in the most efficient low-power mode possible by disabling all internal peripherals like oscillators, timers and so on, leading to an expected increase in battery lifetime.

3.2 Hardware

3.2.1 AD5933 - Integrated Impedance Analyzer

The AD5933 is an IC which makes use of a fully integrated impedance measurement system that is configurable through an I2C interface. In its “off the shelf” configuration, the AD5933 is capable of measuring impedances ranging from 100 Ω to 10 $M\Omega$, when considering a frequency sweep between 1 kHz and 100 kHz. The chip allows for a frequency resolution of 0.1 Hz and claims a measurement accuracy of 0.5 % [53].

The operation procedure that is implemented in AD5933 to measure Z is similar to the

procedure described in section 2.4, which consists in exciting an unknown complex impedance with a known controlled voltage at a well-defined frequency and measuring the current flowing through the sample. The components of Z are determined by computing the real and imaginary parts of the measured current via a Discrete Fourier Transform (DFT) in order to obtain the coefficients. By controlling the commands sent to the I2C interface, it is possible to select the desired frequency range to perform the EIS analysis. Additionally, a single frequency analysis can also be programmed.

To measure the impedance of an object it is necessary to generate an alternate voltage signal. To achieve this, the AD5933 includes a programmable direct digital synthesis (DDS) module, responsible for synthesizing a sinusoidal signal at a defined frequency [53]. The digital generated signal is then converted into an analog wave by the integrated DAC and sent to a programmable gain stage, defining the amplitude of the signal that is injected in the sample by the voltage output pin of the IC, called V_{out}. The output impedance of the AD5933 is directly related with the amplitude of the generated voltage signal, varying between 200 Ω and 2.4 $k\Omega$ [54].

When a voltage signal is applied to the sample it will generate a current signal. In the measurement phase, that current signal is forwarded to a current-to-voltage amplifier, presenting at the voltage input pin of the IC, identified as V_{in}, a virtual ground with a DC value set to half the supply voltage ($V_{DD}/2$). The current flow enters via a V_{in} pin and develops a voltage signal at the amplifier output, with a gain externally defined by the feedback resistance (R_{FB}). The voltage signal is generated between R_{FB} and V_{in} pins. The programmable gain amplifier (PGA) inside the AD5933 package allows the output to be multiplied by a factor of either 1 or 5, based on the set configuration [55].

After amplification, the signal is then filtered by a low-pass filter and then presented to the input of a 12-bit ADC. The result of the conversion is then forwarded to the digital signal processing (DSP) engine to be characterized. Both the R_{FB} value and the PGA gain must be carefully chosen in order to maintain the signal within the admissible range between 0 V and V_{DD} V, and avoid saturation of the amplifier.

Other element that needs to be configured prior to analysis is the settling time of the system. This value is the time interval elapsed between sending of the command for the DDS module to generate a waveform with a specific frequency, and the moment that ADC triggers the sampling acquisition. This period, which has direct correspondence with the number of output excitation cycles that are allowed to pass through the unknown impedance, can be quantified by

$$\Delta t_{settling} = \frac{1}{f} \cdot N_s \quad , \tag{3.1}$$

where f is the frequency of waveform and N_s is the number of cycles configured in the device.

The range of allowed frequencies f depends on the system clock. AD5933 has an integrated oscillator that runs as 16.776 MHz, which can be translated in to a frequency sweep that can vary between 1 kHz and 100 kHz. If desired, the chip can also be fitted with and external oscillator, thus unlocking impedance measurements at lower frequencies.

To determine the value of the measured current, the chip performs a DFT over 1024 samples for each analyzed frequency, implementing the algorithm in 3.2.1 for determining the spectral power of current.

$$X(f) = \sum_{n=0}^{1023} \Big(x(n) \times (cos(n) - jsin(n)) \Big) \tag{3.2}$$

In subsection 3.2.1, $X(f)$ represents the signal power at frequency f, $x(n)$ is the ADC output and $cos(n)$ and $sin(n)$ represent sample vectors provided by DDS module that carry information related with output excitation signal [55].

The digital representation of the $Re(Z)$ and $Imag(Z)$ components of the result for each frequency point, are encoded in a two's complement format in 4 specific 8-bit registers in AD5933 memory, two dedicated to the real part, and two to the imaginary part, all available via I2C communication. These values, however, do not represent the resistive and reactive components of the impedance under test. To obtain the actual impedance value, the magnitude given by 3.2.1 must be multiplied by a gain factor (GF), which varies for each voltage excitation/gain combination. This yields the admittance which is then inverted to give impedance [54].

$$Magnitude = \sqrt{Re(Z)^2 + Imag(Z)^2} \tag{3.3}$$

Analog Front End (AFE)

Taking what was referred in the previous section, the AD5933 impedance analyzer can, after calibrations and startup configurations, be made ready to perform EIS sweeps without any additional hardware. This can be accomplished by simply connecting the SUT to V_{in} and V_{out} pins, represented in E.1.

If the application requires highly precise and highly repeatable results, the stand-alone

AD5933 configuration possesses some elements that can be enhanced in order to achieve this. One of them is the DC level re-biasing. This must be employed because, when connecting the impedance between V_{in} and V_{out}, the DC bias voltage has a slightly difference between the transmit and receive stages.

Analyzing the standard configuration of the AD5933, presented in figure E.2, it is possible to see that on the receiver side, the DC offset is connected to the ADC midscale and exhibits a reference value of $V_{DD}/2$ as it is connected to the non-inverting pin of the Trans-impedance amplifier (TIA) - responsible for performing the current to voltage conversion. During transmission stage there is no set reference value and it depends on the selected output voltage. This means that, the DC level is amplified by R_{FB}, reducing the dynamic range of AD5933's embedded ADC [54].

The internal output impedance (Z_{out}) of the impedance meter is another aspect to consider which can decrease the performance of the results. This value depends on the amplitude of the voltage excitation signal and can be as high as 2.4 $k\Omega$, having a considerable impact in the measurements. The total measured impedance

$$Z_{total} = Z_{unknown} + Z_{system} \quad , \tag{3.4}$$

is the sum of the unknown impedance from the SUT with the output impedance of AD5933 based on the selected parameters and configuration. When measuring small impedances, adding the system output impedance may dramatically increase the range thus increasing the total measurable impedance. This has a direct impact by reducing the output current. To offset this phenomenon, the calibration resistor R_{FB} must have a higher value, leading to a decreased SNR and a lower system sensitivity [54].

Both the issues related with the DC level between the output and input stages, and the output impedance of the system, can be minimized by using an interface circuit between the impedance meter and the load, also called AFE which is displayed in figure 3.2.

In order to re-bias the DC level of the output, the AFE circuit is fitted with a 47 nF capacitor, two 50 $k\Omega$ resistors and an operational amplifier $A1$. This first stage of the conditioning circuit is composed by a passive high-pass filter, responsible for removing the DC bias offset from the AD5933 voltage output signal. The result is then buffered by the operational amplifier $A1$ with low input noise and high input impedance, which acts as a current supplier to the unknown impedance, preventing the current from being drawn from the AD5933. This circuit filters the DC component of the current and acts like a simple

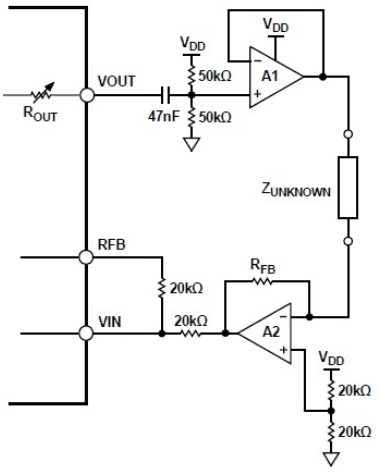

Figure 3.2: Analog Front End for AD5933 [54]

single supply biasing method, using the two biasing resistors with the same value to set the voltage on the non-inverting input of $A1$ equal to $V_{DD}/2$, avoiding the need for an extra power supply to generate negative voltages.

To enhance the sensitivity of the system and include the ability to measure low impedances, the output impedance is reduced by using another operational amplifier $A2$ with low output impedance configured as trans-impedance. This setup, also known as a current-to-voltage converter, enables the conversion of the current that crosses the unknown impedance, to a proportional voltage with gain given by R_{FB}. The voltage value presented at the output of the TIA can be given by

$$V_{TIA} = \frac{V_{DD}}{2} - R_{FB} \times i_{Load} \quad , \tag{3.5}$$

where i_{Load} is the current that crosses the SUT, R_{FB} is the voltage/current gain resistor and V_{DD} is the supply voltage. All these modifications allow AD5933 to have increased sensitivity when measuring lower impedances, while having a low output impedance, which translates into the extraction of more precise results.

3.2.2 SX1276 Transceiver

The SX1276 is a transceiver developed by Semtech which provides ultra-long range spread spectrum communication and high interference immunity, based on FSK/on-off keying (OOK) modulation, while minimizing current consumption. It packs Semtech's proprietary modulation, LoRa, which gives SX1276 a sensitivity of over -148 dBm using a low cost crystal and cheap materials. This increased sensitivity combined with the integrated +20 dBm power amplifier determines the Semtech transceiver as the right choice for applications that require

range and communication robustness.

Other relevant characteristics include compatibility with Institute of Electrical and Electronics Engineers (IEEE) 802.15.4g and WMBus, exceptional phase noise, selectivity, receiver linearity and Third Order Input Intercept Point (IIP3) which enhances the low current consumption of the device, when compared with its competitors. The SX1276 provides several bandwidth options ranging from 7.8 kHz to 500 kHz. Regarding SFs, the transceiver supports SFs ranging from 6 to 12, covering all available frequency bands [56]. Some of the most important features of this Semtech transceiver are exhibited in section F.0.2.

There are several commercially available LoRa modules which have the SX1276 transceiver embedded in their packaging. In section F.0.3 two of the current options available in the market are reviewed. Considering the objectives of the present work, the RN2483 modem was discarded since it has an embedded micro-controller which leads to a slightly higher current consumption when compared with the HopeRF modem. In addition, the current prototype architecture will include a processing unit, which will perform both the functions of the unit embedded in RN2483 and other ones related with EIS. This architecture, with a single micro-controller unit to control all of the peripheral, is preferred over having an extra micro-controller to manage the communication issues. Based on that, the RFM96W was the selected modem to be used in this project.

3.2.3 Micro-Controller Unit (MCU)

In the scope of this project, the choice of the micro-controller most suited for the desired application is directly influenced by the requirements of the Microchip LoRaWAN stack that will be used to develop the long range communication. The minimum requirements for this stack are:

- 8-bit PIC device;
- 32 kB of Flash memory;
- 3 kB of RAM memory;
- 1 SPI interface;
- 6 GPIOS, in which 3 of them must be interrupt capable.

In addition to the stack requirements, an overall analysis of the system was performed in order to enumerate some extra requirements that should be met. After the preliminary

analysis, these requirements were:

- Low-Power consumption in sleep and stand-by mode;
- 1 I2C interface for communication with AD5933;
- 1 SPI interface for communication with LoRa transceiver;
- 1 UART interface for FTDI;
- Low-Cost.

Joining the two main groups of requirements it was observed that a variety of options existed, mainly in the PIC18 family. However, considering that the compatibility between Microchip stack and PIC18F46K22 is assured by the manufacturers *a priori*, the latter was the selected micro-controller. In section G.0.2 some of the most important features of the chosen Programmable Intelligent Computer (PIC) device are detailed.

Once the processing unit was chosen, one other problem emerged, related with programming and debugging functionalities of the PIC18F46K22. One of the possible solutions was to employ a PIC kit 3, displayed in G.2 – an in-circuit debugger/programmer that uses in-circuit debugging logic incorporated into each chip with Flash memory to provide a low-cost hardware debugger and programmer.

Although this is a solution that is able to program a large range of PIC devices and may be used outside of the project scope, its high cost (about 80 euros) was determinant in its elimination as a possible alternative. The alternate solution was also a Microchip device: the Curiosity HPC development board, shown in the figure G.3.

The main direct advantage of Curiosity HPC over PIC kit 3, in addition to its lower cost (35 euros), is the existing fully integrated 8-bit development platform targeted for prototyping in a compact and robust environment that is seamlessly integrated with MPLAB X Integrated Development Environment (IDE). The selected device also has the programmer/debugger functionalities, thanks to an integrated PIC Kit 3, and some additional hardware like potentiometers, LEDs, buttons and mikroBus sockets for plug-n-play modules. Microchip Curiosity devices are available in two versions, one for 8, 14 and 20 pin PIC chips, and other version, the High Pin Count (HPC), targeted for 40 pin PICs, similar to PIC18F46K22.

3.2.4 TPL5010 Timer

The Texas Instruments TPL5010, is an integrated circuit with timer and watchdog capabilities. It is designed for battery-powered applications, mainly IoT, with low-power constraints that requires a periodical "wake-up" or watchdog function. The "wake-up" function may be understood as a switch between sleep mode and the normal running mode, for simplicity reasons this process will be referred as waking the device up throughout the document.

Many of IoT applications contain a micro-controller unit that is usually kept in a low power mode to maximize current savings, only waking up within certain time intervals to perform specific tasks. One of the possibilities is to use one of the internal timers of the micro-controller, if available, for this task. This can, however, lead to a current consumption of some μA, having a negative impact in battery lifespan of the system.

One of the main advantages of TPL5010 is the very low current consumption of about 35 nA, offering power savings of almost two orders of magnitude, making it the ideal choice to replace the integrated micro-controller timer functionality. The chip can be easily deployed in a simple hardware configuration, presented in figure H.1, and can be set to wake-up at different time intervals, which can range between 100 ms to 7200 s, making it simpler to implement in the μC side, with the configuration of an external switch used by TPL5010 to wake-up the device [57].

One of the main goals of this project is to have a device with very low power consumption, hence the need to put the μC in sleep mode. The chosen micro-controller, detailed in section 3.2.3 has six Timers that can be configured in several different modes. Despite this, only Timer 1 gathers the requirements to wake-up the device from sleep mode. Under normal conditions, this would be one of the options to adopt, but due to the fact that Timer 1 is used for LoRaWAN stack implementation and the relatively high current consumption of the device even in sleep mode discards this possibility. All of this led to the choice of an external timer, more specifically the Texas Instruments TPL5010.

4 Implementation

This chapter will describe all of the procedures and steps taken to create a functional prototype, capable of performing EIS analysis and which can transmit the data to TTN, with low-power requirements. It will detail all of the individual approaches taken prior to the final assembly and the overarching solution with all of the modules integrated into a functional prototype. The chapter will be divided according to the different components to be fitted into the system like LoRaWAN communication, impedance meter (EIS), periodic execution and Low-power constraints, respectively.

4.1 Wireless Communication

Before the development of firmware and hardware, the necessary conditions for the prototype to be able to "talk" with the LoRa network and Cloud must be created. Chiefly among them are the configurations that should be implemented in the end-node side, since the connection will be made to an already existent network.

At the time when this project was developed, there were only two LoRaWAN gateways in Coimbra: one at the Departamento de Engenharia Electrotécnica e de Computadores (DEEC) and another one at Instituto Superior de Engenharia de Coimbra (ISEC). Thankfully, these overlapped perfectly with the development sites of the project. Because the coverage of LoRaWAN is not at full potential in the region where this project was carried out, its development was based in another portable LoRaWAN gateway, previously configured and provided by ISEC, in order to cover additional locations where the project was also worked on.

4.1.1 TTN end-node registration and deployment

All of the communication between the prototype and the cloud will be mediated by TTN infrastructure, hence the first steps to be taken are related with the registration of the applica-

tion and the device in TTN system. Before the registration process, it is necessary to register an account in TTN website, which can be found in *https* : *//www.thethingsnetwork.org/*. The sign up process is pretty simple and it requires only a username, password and an e-mail address. After logging in with the created account, the Console section presents two options: registering a gateway or registering an application.

Selecting the "Applications" option opens up the application manager, which is blank before the creation of an application. Clicking the "add application" button, a window similar to figure 4.1 is shown.

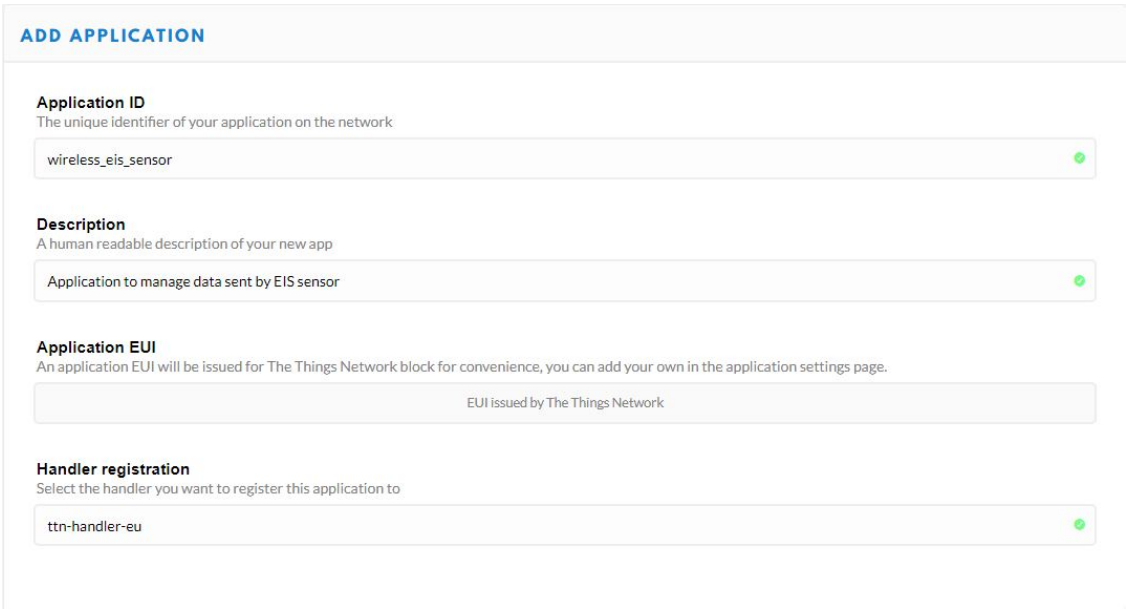

Figure 4.1: Example of creating a TTN application

In it, the user only needs to fill out the "Application ID" – the unique identifier of the application – withing user account and an optional description, used to help contextualize the purpose of the application. The handler responsible for managing the communications should be selected according to the region were the application will be deployed. Since the prototype will run in Europe, the European handler $ttn - handler - eu$ was selected. Regarding the "Application EUI", no further action is needed as it is managed by TTN.

At this point, an application has been created, but it is not yet ready to communicate as there are no devices registered, yet. In order to do that, the user is required to select the "Devices" tab, and select the "register device" button, under the same tab. This action generates a new window with some specific fields that need to be filled out, like the "Application ID", which is mandatory and has a function similar to the "Application ID" described above, among others.

The "Device EUI" field is used to unambiguously identify the device on the network, hence

should be unique and non-repeatable. Besides this field there is the "App key" which is a 16-byte encryption key employed to secure the communication between the end-device and the network, ensuring that the data exchanged between the two end-points cannot be visible or accessed by other entities. These two fields, can be customized by the user or set to be generated by TTN, which is safer. The latter was the adopted option. After these steps, the basic settings of TTN's platform have been successfully configured and the communication with the device that will be developed has been enabled.

4.1.2 Firmware Development

The Firmware Development will be conducted making use of the Microchip software and LoRaWAN library, described in section I. First of all, it is necessary to include the LoRaWAN plug-in in the MPLAB IDE, in order to enable the code required to deploy the wireless communication. The steps to achieve this are detailed in [58].

After that, it is possible to start configuring the necessary aspects to implement the basic functions of the LoRaWAN MAC in MPLAB Code Configurator. The implementation of the LoRa MAC involves the configuration of several elements like hardware and firmware, which can be set-up via Graphic User Interface (GUI) provided by MCC. However, depending on the region where the prototype will be running, some additional aspects need to be taken into consideration like the ISM band or the device class, for example. Despite the large amount of configurations, Microchip provides a generic guide, detailed in [58], to create a project in MPLAB with the LoRaWAN API pre-configured to the selected micro-controller.

The project generated by MCC can be divided in four main layers:

1. The Application Layer, where the programmer will be developing the custom code;
2. The LoRaWAN Layer, which contains the Class A application for the device, along with the transceiver interface and some software timers;
3. The abstraction layer, with provides the interface between the peripheral drivers and the application layer;
4. The peripheral drivers library, which contain the drivers for the selected peripheral, be they custom coded or generated by MCC.

The main structure of the developed firmware to communicate via LoRaWAN contains several layers, which can be visualized in figure J.1.1 of section J.1.

The hardware libraries can be either generated automatically, based on the settings configured in MCC, or coded manually. In the present case auto generation was selected for testing purposes. After adding the remaining features of the system, these libraries will be completed with custom code. The data encryption layer comes pre-deployed with the selected stack however, due to security policies and regulations, some of the functions have to be manually configured.

The first step in the creation of the section of the firmware responsible for wireless communication is to develop a custom main program. This program should be able to make all the relevant configurations and initializations required to successfully transmit a packet to the network. After analyzing some examples of codes with similar functions, a preliminary code was developed in order to test the communication between the end node and the network.

No further configurations regarding hardware or pin configurations were altered since the LoRaWAN API configures all of the peripherals and respective I/O pins necessary to communicate with the SX1276 transceiver. This configuration is done according to the input provided by the user in the LoRaWAN API generation stage. Section J.1.2 describes the first version of the firmware and preliminary functions, it also presents a flowchart which synthesizes the execution flow of the code developed.

After the initial deployment of the first iteration of the firmware, new functions and procedures to make the procedure more robust and reliable in sending data are developed. The created code originates a new layer, located above of the MAC layer assured by LoRaWAN stack, which is responsible for managing several aspects related with the application, working in direct communication with the MAC layer.

This was a complex implementation step due to the lack of support material which could provide a baseline to start the development. This is because most of the LoRaWAN applications use an all-in-one (AIO) chip that does not need any low-level interaction, and, additionally, the example provided by Microchip is very basic. After examining the manual of the LoRaWAN stack, and analyzing almost all of the functions provided by the stack, the extra functionalities developed and posteriorly refined were:

- Receive data from TTN - The device is ready to receive data from the cloud network, regardless the number of sent bytes. In an initial stage it only prints out to universal asynchronous receiver/transmitter (UART) the received bytes, but it can be used to trigger specific actions in the end-node via TTN data.

- Scan Active Channels - The firmware has the capacity to scan all possible channels and register, in a specific variable, which one of those is active.

- Adaptive Spreading Factor - One of the key aspects to ensure the best connection possible between the node and the gateway is the SF. The PIC18 firmware can perform a spreading factor adaptation at each system start-up. The device starts by establishing communication with the network using the lowest SF possible, which is 7. If it fails the communication is retried twice. Should it fail every time, the SF is incremented and the linking is attempted again. This ensures that the device always uses the lowest SF, which has benefits in communication data-rate and power consumption.

- Data Packet Control - The code controls all of the packets that are sent to the network in order to verify if they were successfully sent or not.

- Data Resend Procedure - Using the data packet control, the system has an implemented procedure to re-send packets that were not previously sent. The resend procedure can be configured to rerun a specified number of times. If it fails an additional time, the system proceeds with the execution and will send the previous information only when new data is available.

Some of the implemented functionalities can be found in the commercial and AIO modules dedicated to LoRaWAN communication. Besides the characteristics described above, the power consumption was also taken into account. This lead to some extra procedures and functions which are described in the following sections.

In figure J.3, the execution flow of the firmware which enables the system to successful transmit and receive data is represented. In the diagram all of the previous detailed features are considered and it is possible to see what processes the system executes in order to support the described functionalities. It should also be considered that this execution flow is embedded in a more complex system that combines LoRaWAN communication, impedance analysis and low-power consumption. This means that the displayed structure might not be as clear in the complete firmware due to the necessary adaptation to combine the different parts in the PIC18F46K22 firmware.

The first step in developing the system's wireless capability was the firmware, as it depends on the LoRaWAN stack, which is responsible for assigning some of the needed pins for the communication with the selected transceiver. For that reason, it was illogical to start with the wiring if the pins to be used for the LoRaWAN stack are still unknown. With that

issue addressed, the following stage is to couple the SX1276 transceiver with the processing unit.

4.1.3 Hardware Setup and Wiring

The first thing that needs to be done before performing the wiring between the transceiver module and PIC19F46K22 is to get the pin assignment and respective pin functions of the RFM95, the selected module, in order to successfully link them with the corresponding I/O's on the micro-controller side. In figure 4.2 it is possible to see the pin assignments for the RFM95 LoRa transceiver. Based on the pin-out of the module, a table with each pin name and its respective function was created to better understand the behaviour of each pin and check which should be connected to the processing unit. The results are displayed in table K.1.

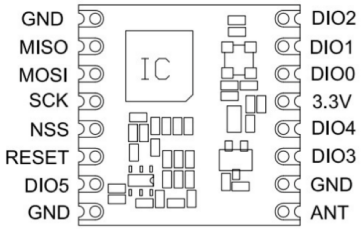

Figure 4.2: Pin functions of RFM95 transceiver module (courtesy of Sparkfun)

With the device and the list of the transceiver pins and respective functions, presented in table K.1 of section K.1.1, it is possible to start configuring and linking the chip I/O pins with their respective pin in the micro-controller. An issue encountered, as can be seen in figure F.3, is that the chip does not allow an easy and stable physical connection without soldering the wires directly to the device. Since this is a prototype in early development stage, this approach was avoided because it do not allow the wires to be easily connected and disconnected.

The adopted procedure was to buy a specific shield for the RFM95 transceiver which has a configuration that enables the use of pin headers for easy connection and disconnection of wires and an SMA connector to append a proper antenna. The shield, presented in figure K.1, is available online at Dycon's website, accessible at www.diycon.nl, and costs only 2.95 euros.

The integration of the transceiver with the selected shield was straightforward, despite some initial difficulties in soldering the chip correctly to the board.

In Table K.1 it is possible to see the pin attributions between the PIC device and the wireless module.

Table 4.1: RFM95 transceiver used pins and respective functions

PIC Pin Number	PIC Pin Name	RFM95 Pin
17	RC2	RESET
19	RD0	SCK
20	RD1	MISO
37	RB4	DIO2
35	RB2	DIO1
34	RB1	DIO0
33	RB0	DIO5
27	RD4	MOSI
22	RD3	NSS

4.2 Impedance Meter

After implementing the first version of the communication with TTN, the next iteration in the development of the prototype was the development of the impedance sensing capabilities, making use of the AD5933 already addressed in subsection 3.2.1. The first stage of development was dedicated to linking the hardware and integrating it with the existing components of the system, namely the micro-controller unit and the LoRa transceiver module. After addressing the physical part, the development of an upgraded version of the firmware, initially developed in chapter 4.1, was started.

4.2.1 Hardware Configuration and Deployment

In order to properly connect the AD5933 with the existing circuit, it is necessary to study all the pins and their respective functions. The layout and the connections of this device may be seen in figure 4.3.

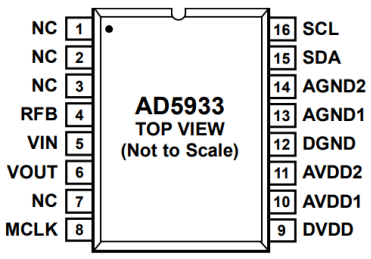

Figure 4.3: Layout and connections of AD5933 [53]

Analyzing the data-sheet [53] provided by the manufacturer, it was possible to create a table with the basic functions of each pin in AD5933. This proved to be a good approach as it helps with the wiring procedure. The gathered information is presented in table K.2.

The AD5933 is a chip that is made on a single package, in this case a 16-lead shrink small-outline package (SSOP). This is ideal for surface mounting but, in the case of prototyping and development, it makes the wiring more difficult. Since the chosen development board does not have any sockets for this kind of package, a shield developed by Roth Elektronik was also bought, presented in figure K.2. This product fulfilled all the technical requirements and was rapidly available, albeit with a slightly higher cost of 7.69€.

With the package assembled, the pins of the chip were connected based on their function. The integration of the impedance meter with the remaining system was done in two stages. In the first stage, the power supply lines and the two pins responsible for I2C were connected, so as to attempt to establish a successful data transfer between the chip and the processing unit. Once this attempt was successful, the sensing circuit was integrated with the AD5933.

For the first stage, the I2C data pins were connected to the micro-controller with 10 $k\Omega$ resistors as recommended and following the pin assignments described in table 4.2, seen below.

Table 4.2: Correspondence between AD5933 pin and selected micro-controller pins

PIC Pin Number	**PIC Pin Name**	**AD5933 Pin**
23	RC4	SDA
18	RC3	SCL

Once the first round of successful communications between the PIC18 and the AD5933 via I2C bus was finished, interface circuit was to be implemented, as described in section 3.2.1. This is a particular circuit that requires some previous computations in order to determine the optimal system parameters for a specific range.

Two parameters that must be tuned are R_{FB} – the feedback resistor that sets the gain of the TIA – and R_{CAL} – the resistor that needs to be used to calibrate the system. Choosing a value for R_{FB} depends on several other aspects, this dependence is defined in the expression 4.2.1.

$$R_{FB} = \frac{(\frac{V_{DD}}{2} - 0.2) \times Z_{MIN}}{(V_{PK} + \frac{V_{DD}}{2} - V_{DCOFFSET})} \times \frac{1}{GAIN} \tag{4.1}$$

In this formula, V_{PK} is the peak voltage for the selected output voltage range, Z_{MIN} is

the minimum impedance measurable by the system, $GAIN$ is the configured PGA system gain (times 1 or times 5), V_{DD} is the supply voltage and $V_{DCOFFSET}$ represents the DC offset voltage for the selected voltage range.

The choice of the most adequate value for the peak voltage must take into account the signal-to-noise ratio of the measurement, which is degraded when lower amplitudes are chosen. For this initial development stage, the peak voltage V_{PK} was set to 0.98V just to test the overall behaviour of the system. After all of the preliminary tests completed, the objective is to increase the V_{PK} to the maximum allowed value, to enhance the SNR.

This fact, alongside the offset voltage generated by AFE to avoid dual power supplies, led to a choice of 0.98V as the final value for the peak voltage.

When setting the lower value for impedance sensing, it was initially decided to employ the lowest admissible value of the chip – $1k\Omega$. The chip's internal programmable gain can have a unitary gain (times 1) or amplify five times the signal sampled by AD5933. Initially, for development purposes, the PGA gain was configured to 1. Since the system will be fitted with an AFE the $V_{DCOFFSET}$ is fixed and exhibits a constant value of $V_{DD}/2$. The whole system is powered at 3.3V.

One other parameter that must be fine-tuned is the calibration resistor, which is dependant both on the minimum impedance value expected to be measured, and on the maximum admissible value. Defining the most adequate calibration resistor value will dictate the quality and accuracy of the obtained results since the GF, obtained during system calibration, is used afterwards to extract the impedance value of the sample and is directly dependent on the data provided by the AD5933. This calibration resistor value can be obtained using the expression presented in 4.2.1.

$$R_{CAL} = (Z_{MIN} + Z_{MAX}) \times \frac{1}{3} \tag{4.2}$$

The maximum value of impedance that the system is able to measure was set, in an initial stage, at 250 $k\Omega$, so as to explore a relatively large band. Should the defined range prove to be insufficient for the objective of the project, new values must be chosen accordingly.

Since the defined impedance range is quite large, special considerations must be taken into account when developing the circuitry. The used AFE, which interfaces with AD5933, sets a maximum allowed ratio for the maximum and minimum impedance specifically 45, for the present case [54]. This means that the maximum measurable Z value must be lower or equal to 45 times the minimum value defined. This is limited by the ADC resolution,

supply, and the DC offset of the selected output and the offset voltage [54]. Taking these parameters into consideration, the computed ratio is 250, well above the maximum allowed.

$$\frac{Z_{MAX}}{Z_{MIN}} = 45 \tag{4.3}$$

To address this issue, the impedance range was divided into shorter sub-segments so as to reduce the difference between the maximum and minimum values. This segmentation and changing of the minimum values of impedance requires the addition of an extra feedback resistor R_{FB} to the circuit, calculated by the expression 4.2.1.

There are several multiplexers or switches that are able to perform the task of switching inputs based on the state of a selector input however, in general, these either need dual supply voltages, consume several μA when in operation or have very high on-resistances which lead to changes in the measured impedance value. The ADG839 by Analog Devices, was the selected device because it exhibits very low on-resistance (about 0.6Ω), supports a single supply voltage of up to 3.6 V, is capable of handling currents up to 300 mA and has a rather low power consumption (below 0.1μW), fundamental to help maintain the battery lifetime of the prototype as high as possible.

The changes in the proposed architecture to support two feedback resistors so as to be able to measure a wider impedance range can be observed in figure K.3. Both R_{FB} and R_{CAL} may be computed some simple math however, in order to later address possible changes in design, a simple Matlab script was created to quickly obtain both parameters based on the impedance measurement range desired. This script is presented in listing K.1.

Despite an AFE being used for AD5933, the R_{FB} can also be obtained by applying formulation in 4.2.1 however it must be taken into consideration the fact that $V_{DCOFFSET}$ has a constant value of $VDD/2$, due to the re-bias of the DC level of the output signal.

With all this in mind, two impedance ranges were defined. The first one starts at the low admissible value allowed by the chip in the proposed configuration, 1 $k\Omega$, and ends at 32 $k\Omega$. The second impedance band is bounded by the lower value of 32 $k\Omega$ and a maximum of 250 $k\Omega$. There is some slight overlap in the impedance range, thankfully this is not a problem because the R_{FB} commutation is controlled by the MCU. This allows for the selection of which range the system will be working on, when inside the overlapping interval. The values were also adjusted in order to have primarily a feedback resistor with a standard value, in order to avoid potentiometers or a resistor combination.

Table 4.3: Calculated values for R_{FB} and R_{CAL}

Range	Z_{MIN}	Z_{MAX}	R_{FB}	R_{CAL}
1	$1k\Omega$	$32k\Omega$	1450Ω	$11k\Omega$
2	$32k\Omega$	$250k\Omega$	$46.4k\Omega$	$94k\Omega$

4.2.2 Firmware Development

The firmware development was one of the lengthier steps during implementation, in order to have a running firmware capable of interacting with the impedance meter. The complexity of this step was within the analysis of the methodology used for PIC18F in I2C transactions and its posterior modulation in order to allow it to work with the AD5933.

One of the main steps is the configuration, in PIC, of the peripheral capable of performing I2C communication, in this case the Master Synchronous Serial Port (MSSP). This peripheral can operate with different communication protocols including RS-232, RS-485, SPI or I2C, among others, assuming that it is properly configured.

To configure the MSSP1 peripheral of the PIC18F46K22 the graphical interface provided by the MPLAB Code Configurator was used. This method was chosen due to its simplicity in generating the code needed to configure and interact with the peripheral, based on the input provided by the user. The configured parameters are displayed in figure J.4.

At this point, the PIC was parameterized and the code needed for configuring and interfacing with the MSSP1 was generated. To start developing the custom library for communicating with AD5933, the generated I2C driver file for the MSSP1 peripheral was used. The samples of generated code by MCC have an approach that differs from the examples that Microchip provides for other PICs, because it uses an I2C communication design based on Transaction Request Blocks (TRBs). This approach does not require the creation of a communication stage with the I2C device based on elementary operations like start, repeated start, or stop; instead it is necessary to construct a transaction in a specific format and with the needed parameters to perform any given I2C action.

To implement the communication with the AD5933, the TRBs and some of the provided functions had to be studied and analyzed. The most used function prototypes are detailed in table J.1.

In the present case, a TRB write transaction cannot be linked with a “write” value on the I2C device because they not map the same operation. A “write” transaction means that something is being written on the I2C bus. To write a value on the device, a series of “read” and “write” TRBs, which are grouped in lists, must be performed. The PIC peripheral

processes a given list as a string of I2C restarts and, once all TRBs are completed, an I2C stop condition is issued to the I2C bus and the flag is set with the correct condition code. Afterwards, if there is any other list in the queue, the process is restarted and this new list is processed in an identical manner. In J.1 it is possible to see a custom TRB transaction included inside a function to read one byte of a specific register in AD5933.

The TRB procedure was extensively analyzed and a custom library with all of the needed functions to interact with AD5933 and explore the full capacity of the chip was created. In table J.2 of the appendix, each of the developed AD5933 functions is described. With the developed library it is possible to communicate with the AD5933 and configure and read all of the needed registers, enabling the construction of the main part of the PIC18F firmware, which conducts the EIS and gathers the relevant data.

The objective of this project is the study of the impedance of a SUT under a very large frequency spectrum, between 5 kHZ and 95 kHZ, however an issue arises considering the equipment being used. An intuitive way to perform EIS is to do a single frequency sweep, configuring the chip to start at a frequency of 5 kHz, end at a frequency of 95 kHz and use a frequency step of 1 kHz, this seems achievable on a superficial level since the impedance analyzer, the AD5933, is capable of measuring impedance up to 100 kHz.

Problems arise because the AD5933 has a finite frequency response, which leads to a gain factor that changes with frequency [53]. The impedance calculus is based on GF, which is computed for each system configuration. In the present case the GF is not constant and varies with frequency, affecting the impedance measurements. This error can be minimized by reducing the range of the frequency sweep to the allowable minimum, within the scope of the project. For this reason, the EIS analysis was compartmentalized in five smaller blocks, with a maximum delta of 10 kHz.

Having enumerated and delimited all of the frequency ranges that will be used in the prototype, the next step in the current development project, is the definition of the programming parameters needed to configure the chip. The main formula used to get the values to be input into the device is based on the selected frequencies and is displayed in 4.2.2.

$$FrequencyCode = \left(\frac{DesiredFrequency(Hz)}{\frac{MasterClock(Hz)}{4}} \right) \times 2^{27} \tag{4.4}$$

Since the system will perform five frequency sweeps, and several parameters must be encoded in the hexadecimal values to be written in the impedance meter, a brief Matlab script was created to quickly return the correct values. The used code may be seen in J.2.

The values that were inputted into the AD5933 to configure each of the frequency sweeps, using the script, are presented in the table 4.4.

Sweep	Start (Hz)	Hex	Stop (Hz)	Delta Increment (Hz)	Hex	Increments	Hex
1	5000	0x2710C	15000	250	0x1F41	41	0x29
2	20000	0x9C42E	30000	250	0x1F41	41	0x29
3	45000	0x15F968	55000	250	0x1F41	41	0x29
4	70000	0x222EA2	80000	250	0x1F41	41	0x29
5	85000	0x2981C5	95000	250	0x1F41	41	0x29

Table 4.4: Frequency hexadecimal codes wrote in AD5933

Regarding the custom firmware developed for the AD5933, the main execution flow of the program is detailed in figure J.5. This solution combines all of the previously referred aspects and is also able to support the dual gain capability based on the two feedback resistors. It is integrated into a larger solution combining EIS capabilities and the SX1276's ability to send the data to the TTN cloud.

This part of the firmware was developed in a blockwise structure to permit the adjustment of several parameters related with the impedance analysis that the AD5933 performs. It is possible to configure the start frequency, increment and stop frequency for each frequency sweep block by simply changing the input parameters of each configuration function developed. Besides that, the code is structured so as to readily deploy any number of frequency sweep blocks desired. Within this project it was configured for five frequency sweeps, but this number can be easily altered. Finally, it is possible to define the most important data-points, extracted in each frequency analysis, which should be sent to the TTN platform.

4.2.3 AD5933 Calibration Methods

After the initial configuration of the system and computation of relevant parameters, it must be calibrated in order to determine the GF. This value depends on the internal configurations of the device and both R_{FB} and R_{CAL} resistors, and is used to determine the value of the impedance measured. Based on the information presented in section 4.2, two impedance ranges were defined, shown in table 4.3. Two impedance ranges require at least two different calibrations, one for each calibration resistor.

The system may be calibrated using different approaches, namely based on a single impedance and a single frequency, or on multipoint frequencies for a single impedance [54].

Besides the single point calibration, it is also possible to realize a two-point calibration of the AD5933 [53].

In the first calibration procedure, an impedance with a known value, usually R_{FB}, is excited with a single frequency equal to the median frequency of the frequency sweep. This is a quick process and can be readily stored in the micro-controller memory, however it can compromise the quality of the results obtained, mainly due to the DFT procedure embedded in the AD5933. Rather than analyzing the entire spectrum and calculating the energy for each given frequency, the algorithm returns a single bin that contains multiple frequencies, at approximately 976.56 Hz at 1 $MSPS$ [54]. Considering a practical example of a measurement at 1 kHz, the referred bin will contain stored energy from a frequency band ranging from 976 Hz to 1952 Hz. This, combined with the fact that noise can be generated by some devices at different frequencies, means that more energy is stored inside a specific bin than the energy measured only by the impedance.

To increase the accuracy, the calibration using multipoint frequencies for a single impedance can be used. This approach generates a GF value for each frequency and is generally used in applications where the frequency span is very wide. This method helps in reducing errors related with energy bins described above and errors due to op-amp bandwidth [54]. There are two possible ways to implement this calibration procedure, either storing in memory a table with the GF for each frequency, or calibrating the system "on-the-fly". Storing the generated amount of data may prove to be difficult for an MCU with low memory, for that reason, when there isn't enough memory, the "on-the-fly" procedure may present itself as a solution to this problem.

Using this process might, however, lead to some changes in hardware, because a device with the capability to switch between different inputs is needed. Usually multiplexers are used, where the calibration resistors and the unknown impedance to be analyzed are connected in a configuration similar to the one represented in figure 4.4.

When conducting an "on-the-fly" calibration, the system, before performing the impedance analysis on the unknown sample, changes the multiplexer input to the calibration resistor, conducts a sweep and determines the GF. The impedance of the SUT in then determined for that specific frequency having in consideration the obtained GF. Unlike the previous method, the sweep phase needs to be repeated twice, once with R_{CAL} and a second time with the unknown impedance.

Other possible calibration procedure consists of a two-point calibration. This approach assumes that the error in gain varies linearly with the frequency, making it possible to

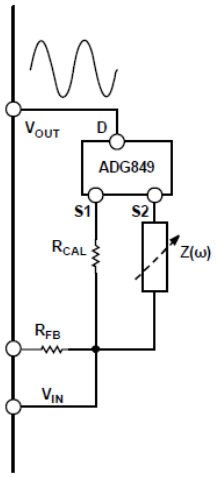

Figure 4.4: Architecture example for On-the-fly calibration [54]

reduce it by adjusting the GF. To implement this calibration method, the GF must initially be obtained for each end of the frequency sweep. Afterwards a linear regression is performed on the data, represented in 4.2.3, to obtain the gain for each of the values inside the frequency band.

$$GF_{DesiredFreq} = \left(\frac{\Delta GF}{\Delta F} \times (DesiredFreq - LowerFreq) \right) + GF_{LowerFreq} \tag{4.5}$$

To obtain the raw data the calculations to determine the GF are all the same, regardless of the selected method. After obtaining this data, the first step is to determine the magnitude of the DFT at the intended frequency point, given by

$$Magnitude = \sqrt{Re^2 + Im^2} \tag{4.6}$$

where Re is the 16-bit value representing the real part of impedance value, stored in the 0x94 and 0x95 registers of the AD5933, and Im represents the imaginary part of impedance, stored in the 0x96 and 0x97 registers.

After obtaining the magnitude of both real and imaginary parts of impedance, the final step is to combine that result with the value of the known impedance measured during the calibration phase. The GF of the analyzed frequency is then obtained using 4.2.3.

$$GainFactor = \left(\frac{Admittance}{Code} \right) = \frac{\left(\frac{1}{KnownImpedance} \right)}{Magnitude} \tag{4.7}$$

In the scope of this project, the simpler method was first implemented – exciting a known impedance with multiple frequencies. For the present case, five different calibrations

were required – one for each of the frequency sweeps previously defined. This was the chosen method considering a preliminary phase because it ensures a good tradeoff between quality of results and implementation complexity. However, in the next iterative phase of development, it is expected for the two-point calibration to be implemented for each frequency range, using on-the-fly procedures.

4.3 System Periodic Execution

The periodic execution of the system tasks was one of the last features implemented. This was because without knowing the exact procedure to communicate with TTN or to perform a EIS sweep, it was not possible to structure the firmware to be interrupt-driven and periodic. Embedding the TPL5010 external timer in the previously developed circuit proved to be simple thanks to the chip's lack of need for any configurations besides the resistor, which sets the time interval, and its small amount of pins to be linked with the MCU. In the present case, the first step was to establish the physical connections between the device and the rest of the circuit.

4.3.1 Hardware Configurations and Setup

The TPL5010 is only available in a single package, specifically a 6-pin small outline transistor (SOT23) package. This rather small package is an essential component to have when the upgraded version of the prototype is to be developed, however it raised an issue when it came to connect the device to the breadboard circuit developed so far. The chosen approach, similar to the other chips, was to buy an adapter shield to convert the SOT23 package to a dual in-line package (DIP) mount, compatible with the breadboard. The Aries adapter, represented in figure K.4, was the selected choice.

The configuration of the time interval for the wake-up of the MCU is done based on the value of the external resistor (R_{EXT}), placed between the respective terminals in TPL5010. The magnitude of the resistor for the desired time can be obtained from the formula 4.3.1. In addition, some of the most used intervals can be seen in [57]. It was decided to employ an interruption around every 8 minutes, as it is the best compromise between a reasonable delay for testing and a standard resistor value, $10k\Omega$ in this case.

$$R_{EXT} = 100 \times \left(\frac{46.9861 - \sqrt{46.9861^2 + 4 \times 0.1284(-2561.8889 - 100 \times 8)}}{2 \times 0.1284} \right) \tag{4.8}$$

The connection between the WAKE pin of the device and the MCU is straightforward and does not require any additional hardware. The same approach is valied for the DONE pin, used by the MCU to indicate that the wake-up (WAKE) signal was successfully received. For the reset signal coming out of the TPL, a pull-up resistor is needed before connecting to PIC, since the output is open-drain and does not have the ability to provide power. All of the above considerations lead to the connections between the MCU and the external timer presented in 4.5.

Table 4.5: TPL5010 used pins and respective attributions in PIC18F

TPL5010 Pin	PIC Pin Name	PIC Pin Number (PDIP)
WAKE	RB7	40
DONE	RB6	39
RST	RE3	1

4.3.2 Firmware Changes

In order to explore the functionalities of the external timer, the firmware required some minor adjustments mostly related with pin configurations. One of them was the configuration of the RB7 to trigger an interruption signal when a state transition occurs in that pin. This behaviour is needed because the PIC18 should resume the program execution right after performing the last instruction in the previous execution cycle. Along with configuration of RB7, it was also necessary to develop a routine to pulse a signal in the RB6 pin when the RB7 interruption is executed so that the TPL knows that MCU is awake. In figure J.6 the interaction between the external timer and the MCU can be seen, along with the periodic execution behaviour of the developed firmware when the external timer wakes up the micro-controller.

4.4 Low-Power Capabilities

The implementation of procedures to reduce the prototype energy consumption was one of the last steps taken in the development since it is illogical trying to optimize the power consumption of the different processes, without before accounting for and developing all of the system features. Essentially, the focal point was the firmware of the PIC device in order to take advantage of the sleep/low-power modes that the implemented modules, like the AD5933 or the SX1276 transceiver have. Since some of the most important features of the

SX1276 are implemented via LoRaWAN stack, the first stage consisted of exploring the radio functions present in the stack to find any procedure capable of enabling the sleep mode of the transceiver. This desired was however not found. Because of this the datasheet of the SX1276 device was also looked into.

After a detailed analysis, it was found that the SX1276 has a specific register (0x01) that is responsible for controlling the operational mode of the device. The $RegOpMode$ is a 8-bit register which allows for the configuration of some modulation aspects but also has 3 bits for controlling the transceiver modes, of which the sleep mode is one. To trigger that state, it is necessary to change the three least significant bits to 0. Having enabled the sleep mode of SX1276, a simple procedure was created which makes use of some of the methods available in the stack and was embedded in the radio library of the stack. The result is displayed in listing J.3.

Like the SX1276, the impedance meter from Analog Devices also has the possibility to be put in low-power mode, consuming only 5 μA [53], by putting the device in power-down mode. To explore this low-power functionality, the power-down mode was implemented in order for the device to consume less current. Unlike in the case of the SX1276 no new method was developed. This was possible because when the AD5933 library for PIC18F device was developed, a function called $ChangeMode$ was created to switch between the different operational modes of AD5933, including the power-down mode.

Relatively to the PIC18 MCU, some routines to stimulate the low-power consumption from the device were developed. Firstly the device was configured for the power mode that reduces the current consumption to the minimum possible – the Sleep Mode. In this mode, the MCU, when in sleep mode, disables the primary oscillator and some of the peripherals, extraordinarily reducing the amount of current drawn to a few μA.

When a peripheral module is not being used or inactive, it will continue to draw some amount of power, even with the device in sleep mode. To reduce the amount of power consumed, all the system peripherals are disabled right after the MCU enters in sleep mode, enabling only the ones that are used during the normal execution of the firmware after the wake-up. Apart from this measure, the pins that are used are also being configured as high Z to ensure that no power is being drawn from PIC when the low-power mode is deployed.

The combination of these procedures helps in saving current that otherwise would be wasted. This issue may be further explored and some other measures could be taken to achieve even lower currents. This is one of the various details that should be addressed in the next versions of the prototype.

In the following chapter, the current consumption of the device will be thoroughly analyzed, along with the power savings obtained by using the low-power measures described. In the same chapter, the results of impedance measurements and calibration procedures will also be presented and analyzed.

5 System Evaluation and Results

Testing a developed system is one of the most important phases in the creation of a prototype. The real capabilities of the developed system are determined through testing, which allows for a critical evaluation of the viability of the whole project. This chapter presents the analyses ran on the prototype, as well as all of the results. The main components and features of the system like wireless communication, impedance measurement, cyclic behaviour and power consumption, were examined and relevant results and conclusions will be addressed in the following sections.

5.1 Sending Data to TTN Application

One of the most important aspects that the system should present is the transmission of information using LoRa modulation. Should the testing of this capability return errors or if it fails completely, the whole structure must be re-evaluated.

To test the wireless capabilities, the prototype data will be used to transmit to the TTN application, previously created in 4.1. Firstly, the gateway logs are checked to verify if it is receiving any information sent by the node, in order to validate this gateway. The device backlog is available since this gateway has been previously registered by ISEC. If this step returns positive results, it is assessed whether the application created in the TTN server is receiving data from the end-device.

From the end-node side, whose terminal output is represented in the listing L.1, the device tries to connect to the network using the pre-set values for App EUI, Dev EUI and App Key, while using the OTAA method. This activation method was adopted for testing and initial development phases because it triggers a sequence of messages and "hand-shakes" between the two ends of the communication tunnel. This sequence is relevant for debugging and the firmware's proof of concept. Without this tool, this process would be impossible since the ABP method does not usually involve direct communication between the end-node

and the gateway. In ABP, the node has the necessary keys stored internally and skips the join procedure stage by connecting to the network without generating visible "messages".

After successfully joining the network, represented by *Connection Status 1*, the device sends the raw impedance values of the predefined sequences to TTN and enters the sleep mode. At this point, from the prototype side, everything was working correctly. This means that the other end of the communication tunnel, the TTN, now becomes the focal point of the analysis.

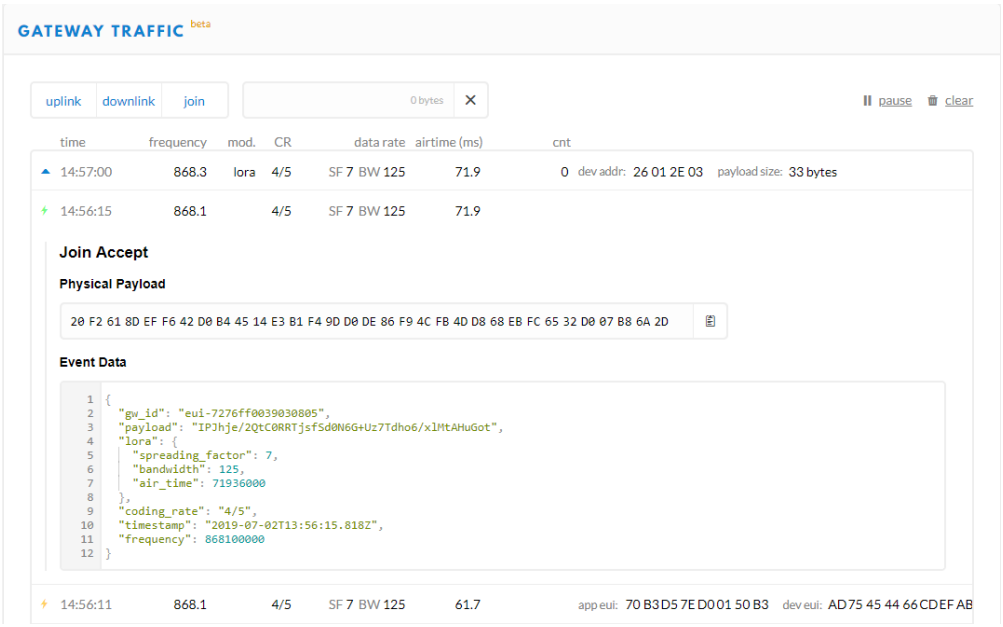

Figure 5.1: Received packets at the gateway

In the gateway logs, presented in figure 5.1 a normal exchange of LoRa packets between the end-node and the gateway are shown, indicating that the devised system is sending data properly. In the line characterized by an orange logo, the packet sent by the prototype requesting to join to the network is represented. This is an expected result, since the LoRaWAN communication is being deployed with OTAA activation. After receiving the packet, the gateway responds with a “join” accept message to the end-node, signalled by the green logo, indicating that the “join” request was authorized. At this point, the prototype is successfully connected to the network and is ready to send data. The blue icon indicates that the gateway has received a data packet, with 33 bytes of data, from the end-node. It is not possible to view the message sent by the prototype in the gateway backlog as it is encoded. Having received datasets from the node, it is now becomes necessary to verify if the data sent is correctly redirected to the TTN application created.

As expected, the packets sent from the node are correctly redirected to the application data, as figure 5.2 shows. In the application area it is also possible to check the “join” packet

sent, represented by an orange symbol, and the EIS-related data sent. Unlike the gateway backlog, in the application console, the message appears decrypted and readable, easing the interpretation of the received information and enabling the debugging of the communication and impedance acquisition procedure.

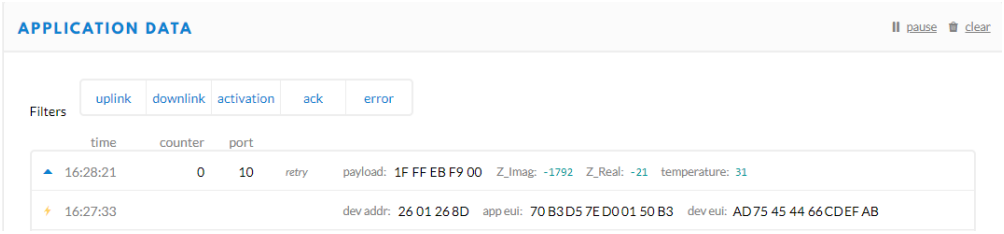

Figure 5.2: Received data in the TTN application

Decoding the information that arrives to the TTN application is achieved through the use of a payload function. In the present case, a simple payload function was created to read the first 5 bytes of information, perform the needed computations, and output the decimal value corresponding to the modulus of the real and imaginary part of the measured impedance along with the temperature read by the AD5933 chip. Placing the output values of TTN application console and the raw console data generated by the end-device side-by side, allows for the conclusion that the wireless section of the prototype is not only capable of successfully communicate with the needed LoRaWAN infrastructure devices, but also of sending formatted custom data without errors and corrupted information.

With these results, one of the main project goals was achieved: the successful deployment of a LoRaWAN communication with only one processing unit, PIC18F in this case, and without using any of the ready to roll AIO LoRa modules.

5.2 Receiving data from TTN Application

After validating that the uplink capability is working as expected, the procedure was inverted in order to simulate sending data from the TTN application to the prototype. Sending messages to the end-node is also an important functionality because it can be used to trigger some specific action on the end-device. This wasn't expanded on in this stage of development, but it is something that merits an upgrade in further revisions of the project. Hence, at this stage of development only the downlink capabilities of the system were developed and subsequently tested. To achieve that, the web-interface from TTN was employed to schedule a transmission of 3 bytes from the network to the device. TTN was also enabled to send

data to the end-device by using a specific set of external services and APIs developed by third parties.

Figure 5.3: Pack of bytes sent to the prototype via TTN

In figure 5.3, the layout of the TTN web-interface and the sample data that was sent to the prototype is displayed. The typed data is not instantly sent, because the LoRa protocol does not allow a gateway to directly contact with an end-device unless the communication has been started by the device. This means that the information is only sent in predefined time slots right after a device transmission, as explained in section 2.3 and visible in figure D.2. Due to this, the devices should first conduct all of the procedures and then send the collected data, for the sent bytes from TTN network to be received and decoded.

As described, the device exits sleep mode, executes all of the five frequency sweeps predefined, sends the compiled information and, after that, the three bytes scheduled in TTN web platform are received and decoded. The received bytes can be seen in lines 37 to 39 of the listing L.2, right after the uplink transmission. Comparing the received data with the inputted data on the cloud reveals a successful implementation of the down-link capabilities in the prototype's firmware.

5.3 Impedance Meter Calibration

Like discussed in section 4.2, the calibration of the AD5933 was set-off by using the calibration method which analyses all of the frequencies contained in each one of the frequency sweeps defined. The calibration resistors used was a 10 $k\Omega$ resistor for the impedance range between 1 $k\Omega$ and 32 $k\Omega$, and a 100 $k\Omega$ for impedances contained between 32 $k\Omega$ and 250 $k\Omega$. The generated results of both analysis were outputted to the terminal via FTDI.

In L.3 it is possible to examine the results produced by the system while performing a complete frequency sweep. The two comma separated outputted values are directly related with the real and imaginary part of the impedance Z analyzed. Both values are the result of the representation of the respective two 8 bit registers in two's complement format.

The results generated by the device while performing the analysis of the two chosen calibration resistors were used to calculate the GF, necessary for the determination of the unknown impedances that the system will measure. It was decided to create yet another Matlab script, represented in Listing L.4, to calculate the GF for each one of the frequencies analysed in the five intervals.

Based on the script and using the values outputted by the device for the 10 $k\Omega$ and 100 $k\Omega$ calibration resistors, the obtained gain factors for each one of the two measurement ranges are displayed. In Figure 5.4 the GFs calculated for impedances magnitudes ranging between 1 $k\Omega$ and 32 $k\Omega$ are represented. The calibration for range 2 is presented in Figure 5.5.

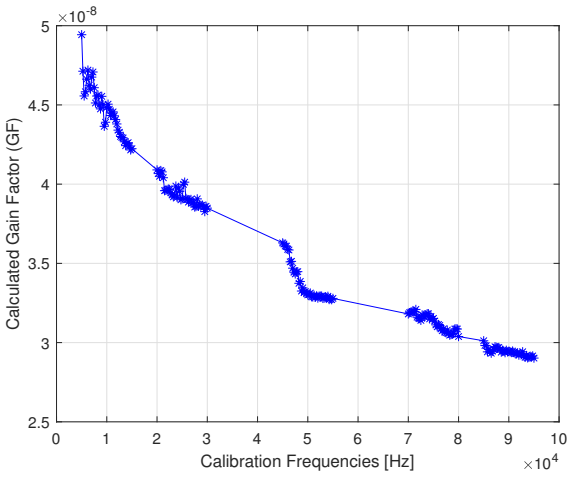

Figure 5.4: Calculated Gain Factors for range 1

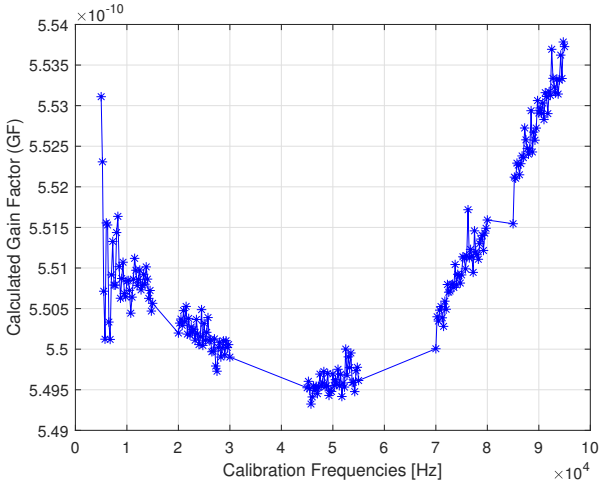

Figure 5.5: Gain factors calculated for range 2

On both figures it can be seen that the GF does not have a fixed relation with the frequency of the analysis, hence the importance of the calibration procedure. Both of the analyses are based only in one frequency sweep, which can lead to some inconsistencies in the calculated GF, visible in figure 5.5. One of the possible ways to minimize that effect is to calculate the GF based on median value of several acquisitions. This was not the implemented approach since it requires a procedure that, at the current stage of development, is difficult and time-consuming to implement.

Once the system was calibrated, it became possible to perform a EIS sweep and determine the magnitude of the unknown impedance.

5.4 Impedance Meter Measurements

In this section some of the results obtained by the prototype developed are discussed, for the measurement of different types of impedances. All of the results displayed were obtained

with the preliminary version of the prototype developed in the breadboard. The preliminary and prototypical nature of this device is relevant since this may have a significant influence in the quality and repeatability of the results. Despite this fact, it is also a good benchmark for proof of concept, considering this as the worst case scenario. All of the results were also based in a single sweep over all frequencies, which may compromise the precision of the results.

At this point, none of the computations needed to obtain the value of the impedance were conducted with the prototype, in order to keep the firmware's simplicity to still be able to debug and change certain procedures, if needed. Some of the necessary values are, at the current stage, being calculated in the PIC however, once the first version of the firmware reaches the first development iteration, the objective is migrate all of these calculations to the device itself.

For debugging and development purposes, another Matlab script was created, with the capability to read the calibration values stored in a file and generate a representation of the impedance measured by the device, using the raw data outputted by this very device. The script for determining the impedance value based on the previous calibration is listed in Listing L.5.

Three impedance analysis were conducted for range 1. At this stage, only the simpler impedance electrical model, consisting of one resistor, was tested, as the system's behaviour was previously unknown. First, a 1 $k\Omega$ resistor was studied, followed by a 2.2 $k\Omega$ and 10 $k\Omega$ resistors. All of the results are represented in 5.6, 5.7 and 5.8. For range 2, three more sweeps were taken. The three chosen resistors for this analysis were: 47 $k\Omega$, 100 $k\Omega$ and 220 $k\Omega$. These can be seen in Figure 5.9, Figure 5.10 and Figure 5.11, respectively.

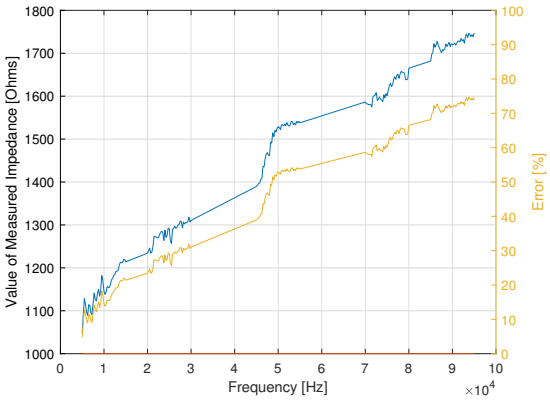

Figure 5.6: 1 $k\Omega$ impedance analysis

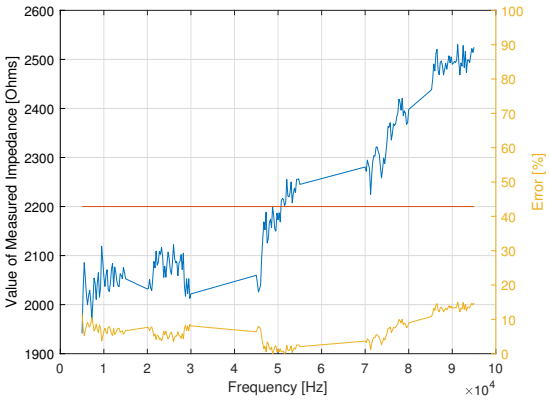

Figure 5.7: 2.2 $k\Omega$ impedance analysis

The analyses show that the system is not as accurate as would be desired in measur-

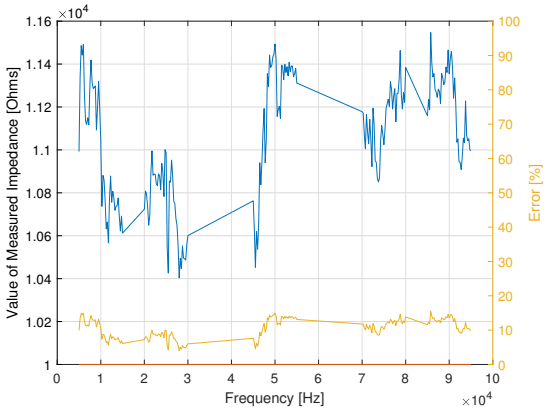

Figure 5.8: 10 $k\Omega$ impedance analysis

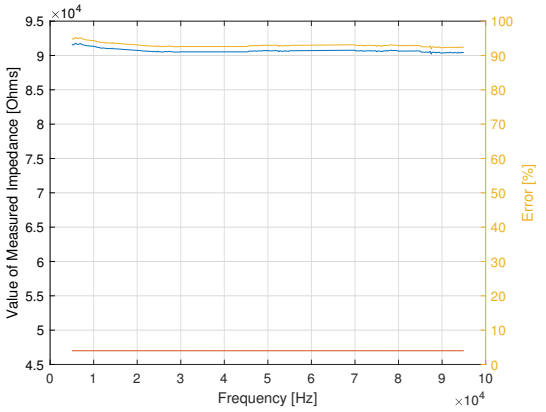

Figure 5.9: 47 $k\Omega$ impedance analysis

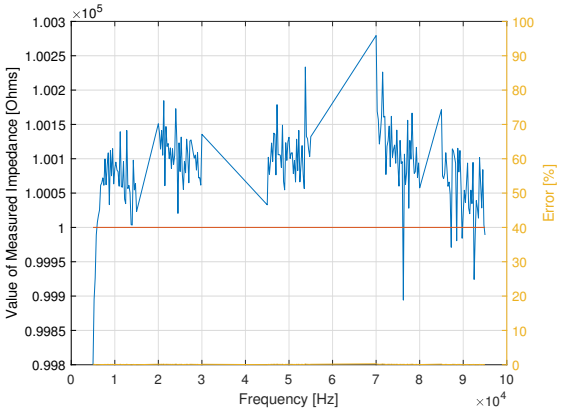

Figure 5.10: 100 $k\Omega$ impedance analysis

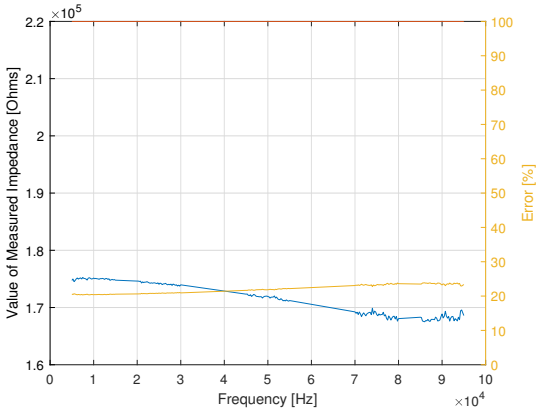

Figure 5.11: 220 $k\Omega$ impedance analysis

ing unknown impedances, exhibiting errors higher than the ones described in the AD5933 datasheet. The exception is the 100 $k\Omega$ analysis which produced very low errors, in the same conditions of the other tests. On the other hand, the 47 $k\Omega$ analysis is very poor in terms of results, exhibiting errors of large orders of magnitude.

One factor that contributes to higher errors and reduced precision is the deployment of the prototype in the breadboard. This type of assembly is prone to unreliable connections and the fact that the connection may not have been properly made, between the SUT and the different components of the system, plays a significant role in reducing the accuracy and precision of the results. A single deviated wire was proven to be a cause for the appearance of an alteration in the outputted results, for example. Besides that, the wires may also induce errors and interferences in the measurements, as described in [3]. The influence of the wires in the measurement can be seen in Figure 5.6, where the impedance measured is increasing when the frequency rises. This trend can be linked with a inductive behaviour which is related with the large amount of wires used in the prototype.

All of these aspects were took in consideration and a new schematic and respective printed

circuit board (PCB) were developed in the scope of this project, as a possible measure for improving the device and to properly test the impedance measuring capabilities of the system. However , due to several factors, the migration to this new platform was impossible to complete within the project timetable. The developed schematic and PCB are presented in section M.

The next step is to conduct error analysis on the new PCB system design and compare with the preliminary results displayed above and verify if the results improve. New improvements can only be considered after that analysis.

5.5 Power Consumption Analysis

The analysis of the device's power consumption was one of the last issues addressed, since it only makes sense for it to be analyzed once the prototype is in a stable version of development, with all of the main features already implemented. When this analysis was ran, the device was already able to communicate via LoRaWAN protocol and conducting EIS analysis, in a periodic time basis.

The main objective was to measure and trace the amount of current consumed by the device at different operating modes, in order to analyze the impact of each one of the functions like network join, message transmission, EIS sweep and sleep mode in the current consumption. To do that, a resistor of known value was placed at the beginning of the power supply line and the voltage drop at the resistor was measured. This measured voltage can be converted into a current measurement by using Ohm's law and the magnitude of the resistor used.

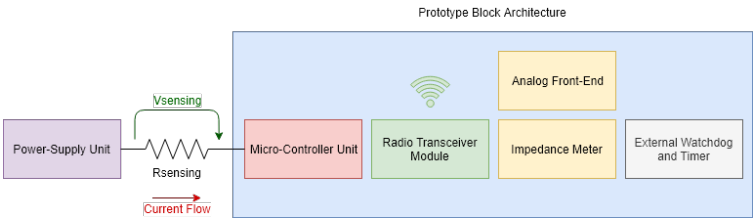

Figure 5.12: Relation between calculated gain factor and calibration frequency

In figure 5.12 it is possible to see the circuit layout implemented to measure the current. To be able to precisely quantify it, a change had to be made in order to isolate the PIC MCU from the development board which was being used for the development. This was necessary because the Curiosity HPC does not have any procedure to isolate the MCU from the rest of other peripherals. This means that if the current consumption is measured with the

microcontroller in the board, the total current of the prototype will be the value obtained, alongside the current consumed by all of the peripherals of Curiosity, like the programmer and the debugger, among others.

To be able to log the device current consumption to visualize the overall behaviour, the Digilent Analog Discovery 2 USB Oscilloscope was employed. The results, displayed in figure 5.13, cover a wake-up cycle together with an EIS measurement phase and a transmission stage of the LoRaWAN node. This system behaviour is required every time an EIS analysis is performed and its results sent through the network.

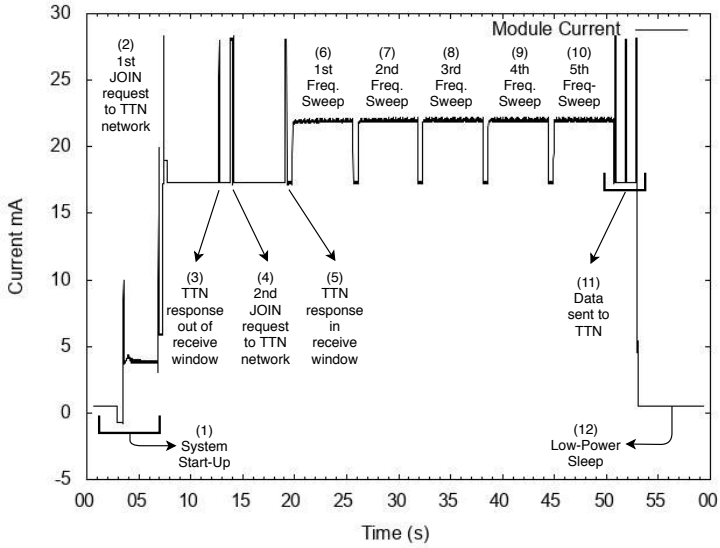

Figure 5.13: Detailed current consumption analysis of device functions

By analyzing the figure, it is possible to see the influence of the system start-up, represented by (1) in the total current consumption. After the start-up, the system tries to join the network, using Over-the-Air activation (OTAA), which is represented in the first spike of the current waveform (2). Following this, the module opens a time limited window and waits for the response of the network within that time interval. If the network response is outside the receive window, the module starts a new joint request, indicated by (4).

The request to rejoin the network might be related to traffic congestion in the wired network where the gateway was installed, which delays the response sent to the device. Once the response falls within the receive window (5), the module starts the impedance analysis process, the influence of this procedure can be seen by an increase in the overall current consumption, during the sample excitation time.

When the frequency sweep and EIS analysis are complete, seen in the numbers (6) through (10), the system processes the received information and generates the real and imaginary part of each frequency of interest, storing it inside a buffer that will be used in the transmission

phase (11). When the data exceeds the LoRa maximum packet size, the data is split into multiple packages. After the confirmation that the data was successfully sent, the prototype starts the pre-sleep procedures, forcing the low-power mode on all the peripherals, namely the impedance meter and the transceiver module, and redefining the inputs and outputs to a high impedance state to minimize the leak currents. Subsequently, the processing unit of the system enters its sleep mode and waits for the next switch on signal, generated by the wake signal from the TPL5010 external timer. In this lethargic mode, the current consumption of the prototype assumed values of $400\mu A$.

Besides the study of the current consumption in one wake-up cycle, the periodic capabilities of the device to trace a current consumption over a longer period were also analyzed. Figure 5.14 illustrates a 20 minute interval during which four sequences of wake-up, join, EIS, transmit data and sleep stages occurred.

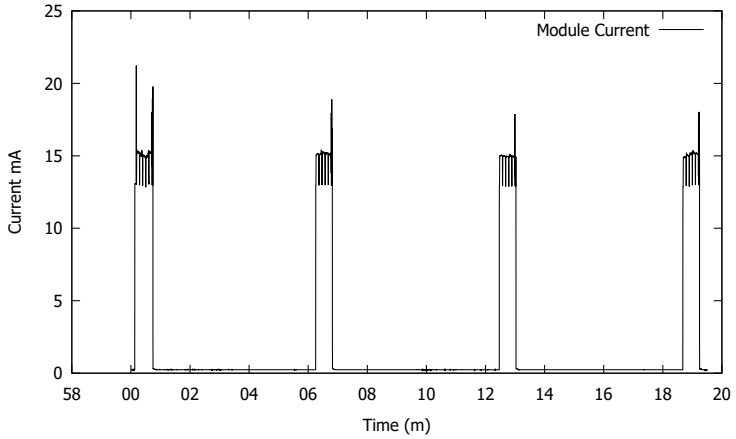

Figure 5.14: Overall current consumption analysis of device during normal cyclic execution

All in all, the results presented look promising in terms of current consumption, and consequently regarding battery lifetime. Despite that, there is still room for improvements and the possibility to achieve an even lower sleep current, with some adjustments both in firmware and hardware.

6 Conclusions

This thesis addressed the issue of growth diseases that affect vegetable tissues which are caused by a variety of external and internal factors. Its main goal was the creation and development of a low-power and wireless technological solution capable of studying a plant's physiological state and infer about the vegetable health condition. The development process revealed itself to be quite challenging and often trying, due to many problems in the early stages of development. The main concept was to develop a system based on a Texas Instruments microchip, the MSP430, because of its lower cost and good low-power skills. However, the LoRaWAN communication's integration with MSP430 revealed to be a problematic issue since, at the date of development, no information or integrations between the chosen transceiver and the MSP430 were available. The two possible solutions were to either port the LoRaWAN stack, available for other platforms, to the MSP430, or change the core system to another architecture with more support to LoRaWAN. The latter was chosen as it appeared to be the one that had the highest probability of completion within the project timeframe, already reduced by then.

Despite the major setback during the project, the main objectives were accomplished. A complete solution capable of sending and receiving information based on the innovative LoRaWAN MAC protocol and perform impedance analysis on a SUT was developed, taking into account the low-power objectives.

During testing, the system was satisfactory in communicating with LoRaWAN gateways, successfully sending and receiving packets without errors and unexpected behaviours. The impedance analysis performed in several magnitude defined resistors proved that the system can interact with the AD5933 and retrieve valid impedance measurements. This feature was not without defects however, and significant errors were introduced, in the conducted measurements, ranging from 1% to upwards of 60%. Although further tests need to be performed on the impedance measurement capabilities in order to evaluate if the detected errors are due to the embryonic implementation, or if the impedance meter of the system

was incorrectly designed.

All in all, the developed solution proved the viability of the initial concept developed and its possible introduction into a marketable solution, contributing to healthier plants and a more sustainable ecosystem.

6.0.1 Future Work

Despite the initial proof of concept, the developed solution still came up short in a variety of ways and therefore has room for improvement. This is needed in order to mature the overall design and make it market ready. Some of the focal points that a possible revision of this concept should consider are:

- Migrating the architecture from breadboard to the developed PCB – This step will be important to diagnose If the system has any major errors in design that need to be corrected;
- In-depth impedance analysis – The device impedance measuring skills should be more detailed and studied to evaluate both the magnitude and the phase of the measured impedance;
- Incorporating "on-the-fly" calibration – This type of calibration is less prone to interferences and external factors as it calibrates the system before data-acquisition, enhancing the results. Besides that, is has less memory costs than allocating all the GFs for different frequencies in MCU memory;
- Comparing the EIS results with an equivalent market solution – To prove the validity of the data, the impedance measurements should be compared with a market established solution, the Discovery 2, for example;
- Testing the LoRaWAN coverage – Validate if the LoRaWAN coverage is in line with the objectives targeted for the device;
- EIS acquisition on live tissues – Test the prototype in vegetable tissues over long periods of time and compare the results with experimental data gathered in equivalent conditions;
- Development of a front end application – All the data exchanged between the end device and the network should be visible to the user, along with the expected health status of the plant.

Ultimately this prototype aims to cover a gap in the market. While the results may seem promising, some work must be done before a marketable solution may be developed and this tool used to its fullest. As is the case with every project, there are setbacks and wrong turns which prove to be invaluable opportunities to learn and adjust. This acquired experience and knowledge is a powerful tool in being able to carry the project to its term. Due to the, already referred, time constraints, the work which originated this thesis was sadly unable to do this. However, it remains a record of the background information, ideas and concepts; of the possible methods and tools that a solution might employ; of the issues found and its possible solutions, and of the intellectual process behind all of it. As suchs this thesis may prove itself a powerful tool to whomever wishes to take upon himself to turn this concept into a concrete and produceable tool, the author included.

Analysing the physiological status of vegetable tissues, without the need for specialized labour, is a job which, if done right, can achieve significant gains in both time savings, disease detection and prevention, and even cost managements from the side of the producers. As such the successful development of this concept into a physically compact tool is a goal with environmental and economical consequences which can range from the regional and local level up to a global scope, hence, its relevance is not to be understated.

7 Bibliography

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 2347–2376, Fourthquarter 2015.

[2] S. Popli, R. K. Jha, and S. Jain, "A survey on energy efficient narrowband internet of things (nbiot): Architecture, application and challenges," *IEEE Access*, vol. 7, pp. 16739–16776, 2019.

[3] E. Borges, M. Sequeira, A. F. V, C. H. C. Pereira, J. Cardoso, C. Correia, T. M. Vasconcelos, I. M. Duarte, and P. D. Coimbra, "Assessment of physiological states of plants in situ an innovative approach to the use of electrical impedance spectroscopy."

[4] E. Borges, M. Sequeira, A. F. V, J. Cardoso, C. Correia, T. M. Vasconcelos, I. M. Duarte, and P. D. Coimbra, "Bioimpedance parameters as indicators of the physiological states of plants in situ."

[5] L. Martins, J. Castro, W. Macedo, C. Marques, and C. Abreu, "Assessment of the spread of chestnut ink disease using remote sensing and geostatistical methods," *European Journal of Plant Pathology*, vol. 119, pp. 159–164, Oct 2007.

[6] C. van Leeuwen, O. Trégoat, X. Choné, B. Bois, D. Pernet, and J.-P. Gaudillère, "Vine water status is a key factor in grape ripening and vintage quality for red bordeaux wine. how can it be assessed for vineyard management purposes?," *OENO One*, vol. 43, pp. 121–134, Sep. 2009.

[7] M. I. N. ZHANG and J. M. Willison, "Electrical impedance analysis in plant tissues: impedance measurement in leaves," *Journal of Experimental Botany*, vol. 44, pp. 1369–1375, 08 1993.

[8] E. Azzarello, S. Mugnai, C. Pandolfi, E. Masi, and S. Mancuso, *Stress Assessment in Plants by Impedance Spectroscopy*, vol. 3. 01 2006.

[9] S. Services, "Sigfox technology overview."

[10] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on lpwa technology: Lora and nb-iot," *ICT Express*, 2017.

[11] L. C. Distribuidas, "Waspmote sigfox networking guide."

[12] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT Express*, 2019.

[13] N. Mangalvedhe, R. Ratasuk, and A. Ghosh, "Nb-iot deployment study for low power wide area cellular iot," *IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2016.

[14] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3gpp narrowband internet of things," *IEEE Communications Magazine*, 2017.

[15] A. Adhikary, X. Lin, and Y.-P. E. Wang, "Performance evaluation of nb-iot coverage," *IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016.

[16] C. Goursaud and J.-M. Gorce, "Dedicated networks for iot : Phy / mac state of the art and challenges," *EAI Endorsed Transactions on Internet of Things*, 2015.

[17] Ingenu, "Rpma technology for the internet of things."

[18] J. N. Nine and S. Boudaud, "Lpwan as enabler for widespread geolocation solutions - a lora device for advanced positioning assets tracking," *Embedded World*, 2017.

[19] J. Finnegan and S. Brown, "A comparative survey of lpwa networking," *Zhejiang University International Doctoral Students Conference*, 2017.

[20] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys Tutorials*, 2017.

[21] G. M. I. I. A. group, "3gpp low power wide area technologies - gsma white paper."

[22] Ericsson, "Cellular networks for massive iot," *Ericsson White Paper*, 2016.

[23] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia, and N. Strachan, "Evaluation of lora and lorawan for wireless sensor networks," *IEEE SENSORS*, pp. 1–3, 2016.

[24] O. Khutsoane, B. Isong, and A. M. Abu-Mahfouz, "Iot devices and applications based on lora/lorawan," *43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 8–11, 2017.

[25] S. Corporation, "An1200.22 lora™ modulation basics."

[26] M. Bor, J. Vidler, and U. Roedig, "Lora for the internet of things," *EWSN '16 Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, 2016.

[27] L. Alliance, "Lorawan™ what is it? - a technical overview of lora® and lorawan™," *Technical Marketing Workgroup 1.0*, 2015.

[28] E. Pietrosemoli, "Lora details."

[29] A. Lavrica and V. Popa, "Internet of things and loratm low-power wide-area networks: A survey," *International Symposium on Signals, Circuits and Systems (ISSCS)*, 2017.

[30] S. Corporation, "An1200.13: Sx1272/3/6/7/8: Lora modem designer's guide."

[31] S. Corporation, "Sx1272/73 datasheet."

[32] U. Noreen, A. Bounceur, and L. Clavier, "A study of lora low power and wide area network technology," *3rd International Conference on Advanced Technologies for Signal and Image Processing*, 2017.

[33] D. Purves, "Low power wireless sensor network technologies and standards for the internet of things."

[34] T. T. Network, "Background information about lorawan."

[35] M. S. Mahmoud and A. Mohamad, "A study of efficient power consumption wireless communication techniques/ modules for internet of things (iot) applications," *Advances in Internet of Things*, 2016.

[36] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, "Lora and lorawan testbeds: A review," *IEEE Africon*, 2017.

[37] L. Nordin, "Spreading factor (sf), time on air and (adaptive) data rate," 2018.

[38] N. Ramirez, A. Regueiro, O. Arias, and R. Contreras, "Electrochemical impedance spectroscopy: An effective tool for a fast microbiological diagnosis," *Biotecnologia Aplicada, 72-7*, 2008.

[39] M. Grossi, R. Lazzarini, M. Lanzoni, and B. Riccò, "A novel technique to control ice cream freezing by electrical characteristics analysis," *Journal of Food Engineering, 347-354, Vol. 106, Iss. 4*, 2011.

[40] M. Grossi, G. D. Lecce, T. G. Toschi, and B. Riccò, "Fast and accurate determination of olive oil acidity by electrochemical impedance spectroscopy," *IEEE Sensors Journal, 2947-2954, Vol. 14, Iss. 9*, 2014.

[41] H. Zhu, H. Luo, D. Ai, and C. Wang, "Mechanical impedance-based technique for steel structural corrosion damage detection," *Measurement, 353-359, Vol. 88*, 2016.

[42] A. Amirudin and D. Thieny, "Application of electrochemical impedance spectroscopy to study the degradation of polymercoated metals," *Prog. Org. Coat, 1-28. Vol. 26*, 1995.

[43] M. Z. Iqbal and Rafiuddin, "Preparation, characterization, electrical conductivity and dielectric studies of na2so4 and v2o5 composite solid electrolytes," *Measurement, 102-112, Vol. 81*, 2016.

[44] F. Clemente, P. Arpaia, and C. Manna, "Characterization of human skin impedance after electrical treatment for transdermal drug delivery," *Measurement, 3493-3501, Vol. 46, Iss. 9*, 2013.

[45] M. Grossi and B. Riccò, "Electrical impedance spectroscopy (eis) for biological analysis and food characterization: a review," *Journal of sensors and sensor systems*, 2017.

[46] A. Guyader, F. Huet, and R. P. Nogueira, "Polarization resistance measurements: Potentiostatically or galvanostatically?," *CORROSION*, vol. 65, no. 2, pp. 136–144, 2009.

[47] M. Grossi, M. Lanzoni, R. Lazzarini, and B. Riccò, "Automatic ice-cream characterization by impedance measurements for optimal machine setting," *Measurement*, vol. 45, pp. 1747–1754, 08 2012.

[48] S. Ramanathan and F. Fasmin, "Review - nonlinear electrochemical impedance spectroscopy," *Journal of The Electrochemical Society*, vol. 164, pp. H443–455, 05 2017.

[49] E. C. Duarte, "Electrical impedance spectrometry - application to pet (polyethylene terephthalate) relative humidity detection and preliminary studies towards peloids characterization," *Master Degree Thesis - University of Coimbra*, 2018.

[50] P. Sens, "Bode and nyquist plot."

[51] U. G. Kyle, I. Bosaeus, A. D. D. Lorenzo, P. Deurenberg, M. Elia, J. M. Gómez, B. L. Heitmann, L. Kent-Smith, J.-C. Melchior, M. Pirlich, H. Scharfetter, A. M. Schols, and C. Pichard, "Bioelectrical impedance analysis—part i: review of principles and methods," *Clinical Nutrition*, vol. 23, no. 5, pp. 1226 – 1243, 2004.

[52] S. Grimnes and Ø. G. Martinsen, *Bioimpedance and bioelectricity basics*. 2008.

[53] A. Devices, "1 msps, 12-bit impedance converter, network analyzer - ad5933 datasheet."

[54] M. Usach, "An1253 application note - how to configure the ad5933/ad5934."

[55] N. Capela, "Impedance cardiography," *Master Degree Thesis - University of Coimbra*, 2013.

[56] S. Corporation, "Sx1276/77/78/79 - 137 mhz to 1020 mhz low power long range transceiver - datasheet."

[57] T. Instruments, "Tpl5010 nano-power system timer with watchdog function."

[58] M. T. Inc., "Lorawan™ library plug-in for mplab code configurator user's guide."

[59] A. Labs, "Iot cellular networksr."

[60] R. Dzwonczyk, A. Hartzler, and A. Liu, "A new apparatus and method for measuring the myocardial electrical impedance spectrum," *m, Proceedings of Computers in Cardiology*, 1992.

[61] M. A. Atmanand, V. J. Kumar, and V. G. K. Murti, "A microcontroller-based scheme for measurement of l and c," *Measurement Science and Technology, 5, Vol. 6*, 1995.

[62] M. A. Atmananda and V. J. Kumar, "Microcontroller based lcr meter," *Microprocessors and Microsystems, 20, 297-301*, 1996.

[63] Y. Yang, J. Wang, G. Yu, and F. Niu, "Design and preliminary evaluation of a portable device for the measurement of bioimpedance spectroscopy," *Physiological Measurement*, 2006.

[64] P. Bogónez-Franco and P. J. Riu, "Implantable bioimpedance system for measuring the impedance of kidney," *13th International Conference on Electrical Bioimpedance and the 8th Conference on Electrical Impedance Tomography*, 2007.

[65] T. Piasecki, KonradChabowski, and K. Nitsch, "Design, calibration and tests of versatile low frequency impedance analyser based on arm microcontroller," *Measurement, 155-161, Vol. 91*, 2016.

[66] M. Simic, "Realization of complex impedance measurement system based on the integrated circuit ad5933," *21st Telecommunications Forum Telfor (TELFOR)*, 2013.

[67] L. Breniuc, V. David, and C.-G. Haba, "Wearable impedance analyzer based on ad5933," *International Conference and Exposition on Electrical and Power Engineering (EPE)*, 2014.

[68] J. Hoja and G. Lentka, "Portable analyzer for impedance spectroscopy," *XIX IMEKO World Congress Fundamental and Applied Metrology*, 2009.

[69] J. de Carvalho Silva, J. Rodrigues, A. M. Alberti, P. Solic, and A. Aquino, "Lorawan - a low power wan protocol for internet of things: a review and opportunities," *2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, 2017.

[70] P. Ram, "Part i - understanding the lpwan, lora and lorawan technology."

[71] M. T. Inc., "Rn2483 - low-power long range lora® technology transceiver module."

[72] HopeRF, "Rfm95/96/97/98(w) - low power long range transceiver module v1.0."

Appendix A

Existing LPWAN technologies

A.1 Overview analysis to Sigfox technology

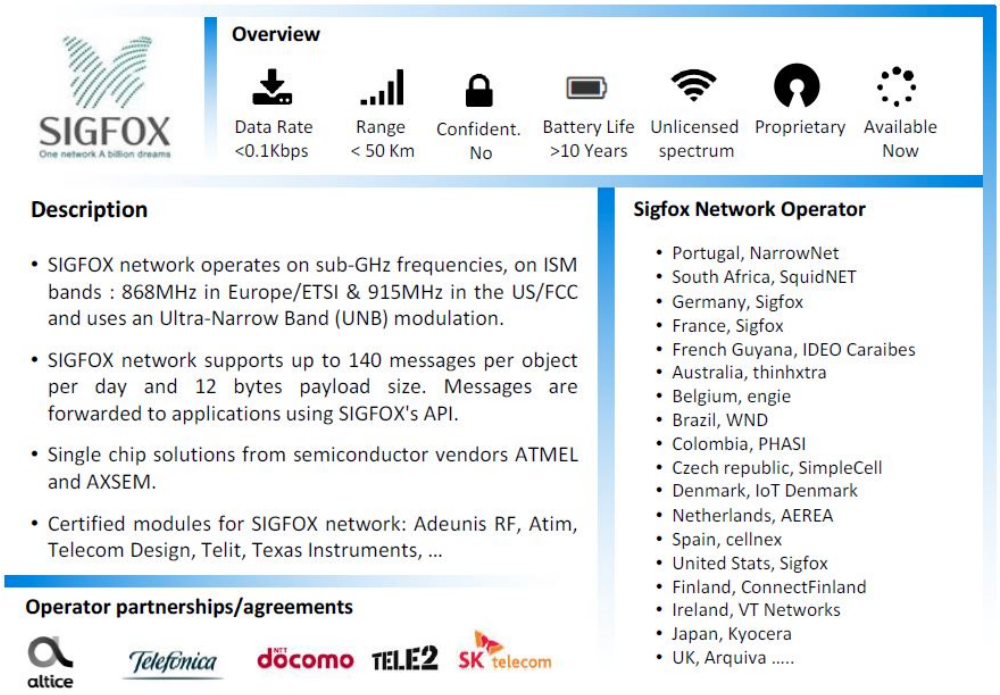

Figure A.1: Overall analysis to Sigfox technology [59]

A.2 Global analysis of NB-IoT characteristics

Figure A.2: Overall analysis to NB-IoT technology and main partners [59]

A.3 General description and overview of EC-GSM-IoT

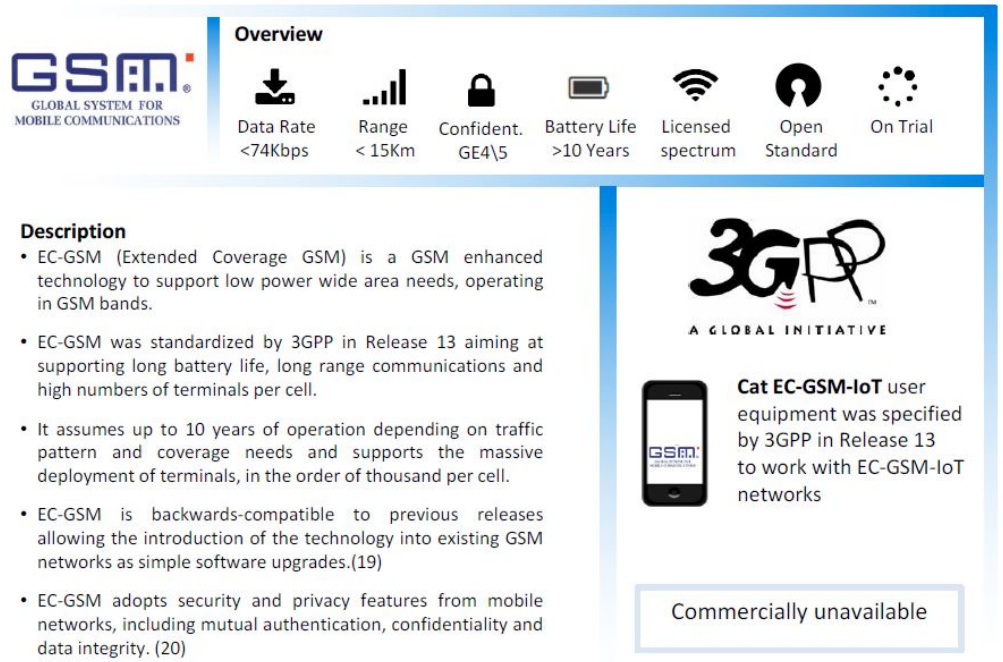

Figure A.3: Overall analysis to EC-GSM-IoT technology [59]

Appendix B

Existing impedance measurement systems

The recent improvements in methodologies and general electronics have enabled the emergence of several projects and studies to perform EIS, packing new chips in portable solutions with a low cost design.

One of the first developments was performed in [60], where the authors created a system based in an Intel 80C196 micro-controller, eight 10-bit analog-to-digital converter (ADC) channels and an RS-232 port to measure the Myocardial electrical impedance (MI) in a clinically applicable way with two cardiac pacing leads.

Three years later, in 1995, the authors of [61] developed a portable LCR meter prototype with an Intel 8751 processing unit and a digital-to-analog converter (DAC) to be able to determine values of some basic electronic components, like inductance (L), capacitors (C) and resistors (R). This architecture was adjusted and perfected, leading to a new prototype, presented in 1996, in [62].

In 2006 Yang, in collaboration with other investigators, developed in [63] a portable bio-impedance spectroscopy (BIS) device based on the magnitude-ratio and phase-difference detection method constituted by a PIC18F452 micro-controller, developed by Microchip, and a AD933 integrated circuit. The integrated circuit (IC) was responsible for generating a test signal within a frequency range from 20 kHz to 1 MHz, and measure the impedance from the unknown sample, which can vary between 9 Ω and 5.7 $k\Omega$. Bogonez and Riu, in 2007, also used a PIC18F2420 micro-controller to develop [64]. The proposed system was designed to be implantable and have the capacity to measure bio-impedance. The range of impedance measured varies between 0.07 Ω and 1400 Ω, considering a frequency sweep from

100 Hz to 2 MHz.

Few years later, in 2016, Piasecki in cooperation with other developers, created in [65] a simple device based in ARM architecture to perform EIS measurements. The proposed impedance analyzer only contains a STM32 micro-controller unit and simple operational and instrumentation amplifiers to perform the EIS sweep. To generate the excitation signal required, and to measure the response of the object, the device only uses the internal DAC and ADCs in the μC. Relatively to signal conditioning for the excitation signal and voltage/current measurements the authors implemented an analog front-end (AFE) to perform the needed operations on the analog signals.

The developments in the EIS area lead to improvements in integrated circuits to enable the impedance measurement in more compact packages. One of the cases of success was the IC from Analog Devices, the AD5933. This chip was used by Simic in [66], where an impedance analyzer based on the microcontroller ATMega128 and AD5933 was presented. The proposed device was able to perform impedance reading in the band comprised between 1 $k\Omega$ and 5 $M\Omega$, considering a limited frequency sweep, starting in 1 kHz and ending at 100 kHz. The generated data is displayed in an attached LCD display and sent to a micro SD card to be stored in a format that is compatible with MS Excel. In the datasheet of AD5933 is possible to verify that, using the internal oscillator of the chip it is not possible to perform frequency sweeps for frequencies lower than 1 kHz. In order to address this limitation, [67] designed a new approach by using a DS1077 external oscillator to generate frequencies lower than 1 kHz and control the master clock of AD5933.

Hoja and Lentka developed in [68] an innovative idea to enhance the characteristic of a portable device for impedance spectroscopy, based on the implementation of two AD5933 IC's. This solution allows for the measurement of impedance from a test sample ranging from 100 Ω to 10 $G\Omega$ in a very wide frequency range, between 0.01 Hz and 100 kHz.

The review of some of the proposed architectures to perform EIS analysis is key for the design process and eventual prototyping the proposed system.

Appendix C

LoRa Modulation

C.0.1 Code-Rate Analysis

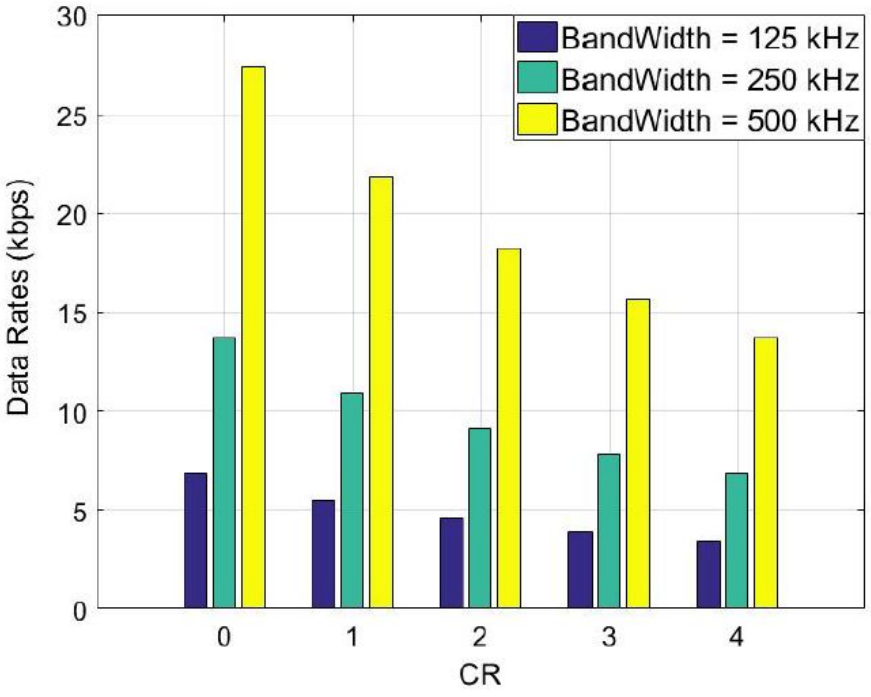

Figure C.1: Effect of CR in data transmission with different bandwidths and fixed SF=7.[32]

C.0.2 Spreading Factor Analysis

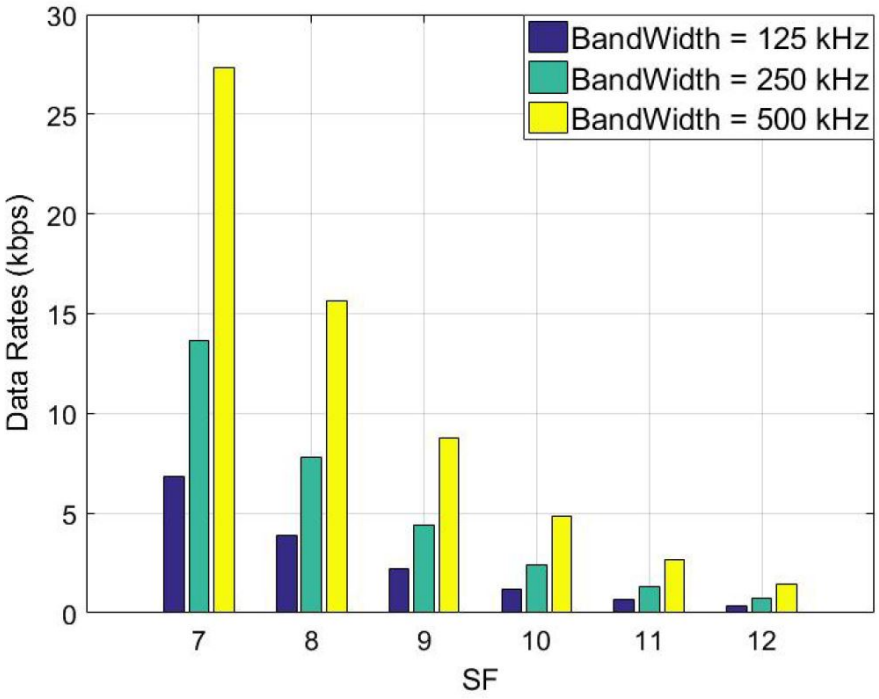

Figure C.2: Relation between data rate and spreading factor in LoRa.[32]

C.0.3 Bandwidth influence in TOA

Table C.1: Influence of bandwidth in information air time and system sensitivity [30]

Bandwidth $[kHz]$	Time on air $[ms]$	Sensitivity [dBm]
125	264.2	-132
250	132.1	-129
500	66	-126

Appendix D

LoRaWAN

D.1 Main characteristics of LoRaWAN MAC protocol

Table D.1: Summary of main LoRaWAN characteristics [69]

Characteristics	LoRaWAN
Topology	Star
Modulation	SS Chirp
Data-Rate (DR)	290bps - 50$kbps$
Link Budget	154 dB
Packet Size	154 dB
Battery Lifetime	8-10 years
Power Efficiency	Very High
Security / Authentication	Yes (32 bits)
Range	5km urban - 15km suburban
Interference Immunity	Very High
Scalability	Yes
Mobility / Localization	Yes

D.2 Detailed analysis of LoRaWAN architecture components

The end-nodes are typically IoT devices with some restrains regarding power consumption and do not perform any kind of heavy computation. Their main task is to measure variables related to the environment, perform some calculations and then send the information generated to a LoRa modem to be broadcasted, without any constraints related to gateway selection. The broadcasted information will be captured by multiple gateways within signal range, which can be any value between 1 km and 10 km, depending on the surrounding environment.

The transmitted information is then received by one or more gateways, which receive the RF packets generated by the end-nodes and convert them to IP packets to be sent to the network server, acting like a transparent bridge [70]. This communication link between the gateways and the server can be connected using Ethernet, cellular or any other wired or wireless telecommunication technologies [27]. Unlike the end-devices, the gateways must be always-on and ready to receive any message at any time, which depends solely on an end-node and is therefore unpredictable.

The core of the LoRaWAN architecture is in the network server. These servers can be based on cloud solutions like TTN or LoRIOT, and are responsible for managing the communication with gateways and rerouting the packets generated by the end-nodes to the correspondent application. They can operate in up-link or down-link mode, sending data from the sensor to the application or redirecting the data from the application to the end-device, respectively.

This approach concentrates all of the computing needs and system complexity in the network server, the core of the architecture. While being able to filter the redundant information received by the gateways, the servers can perform many other tasks like send confirmation messages to end-nodes, selecting the best gateway for a data transmission, adapt the data rate among others. The mains aspects of the LoRaWAN MAC protocol are synthesized in table D.1.

D.2.1 Battery Consumption Analysis

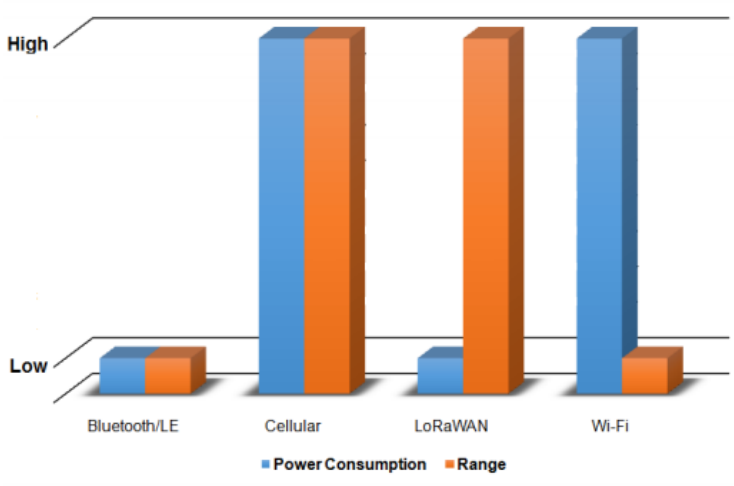

Figure D.1: Analysis of power consumption vs range for the main communication technologies [69]

D.2.2 Available classes for end-devices

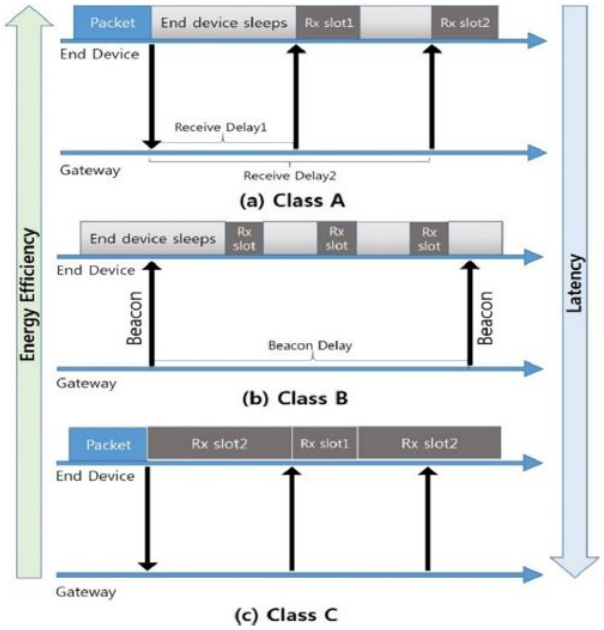

Figure D.2: MAC implementation for each one of the available device classes [10]

Appendix E

AD5933 Impedance Analyzer

E.0.1 Block Diagram of Internal Architecture

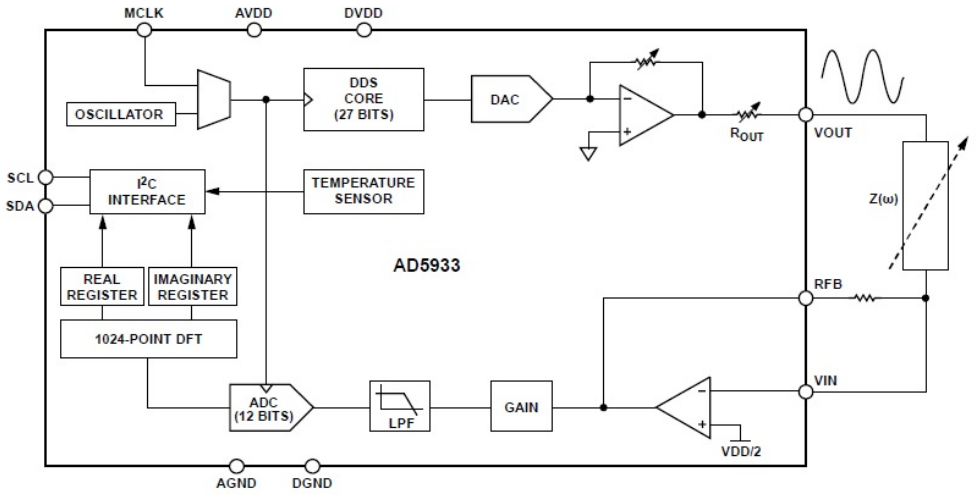

Figure E.1: Block diagram with all of the components of AD5933 [53]

E.0.2 Standard Configuration

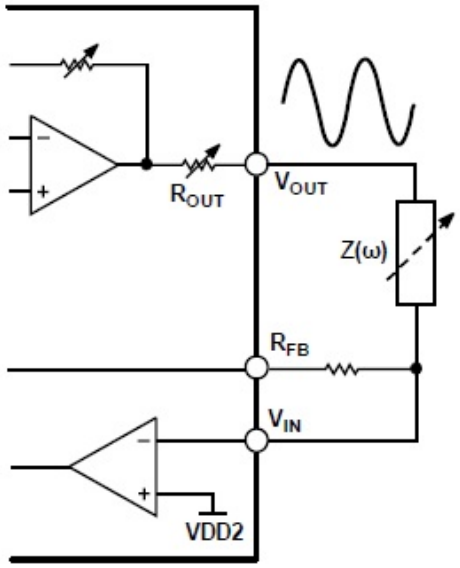

Figure E.2: Out-of-the box configuration for AD5933 [54]

Appendix F

SX1276 Transceiver

F.0.1 Block diagram of internal structure

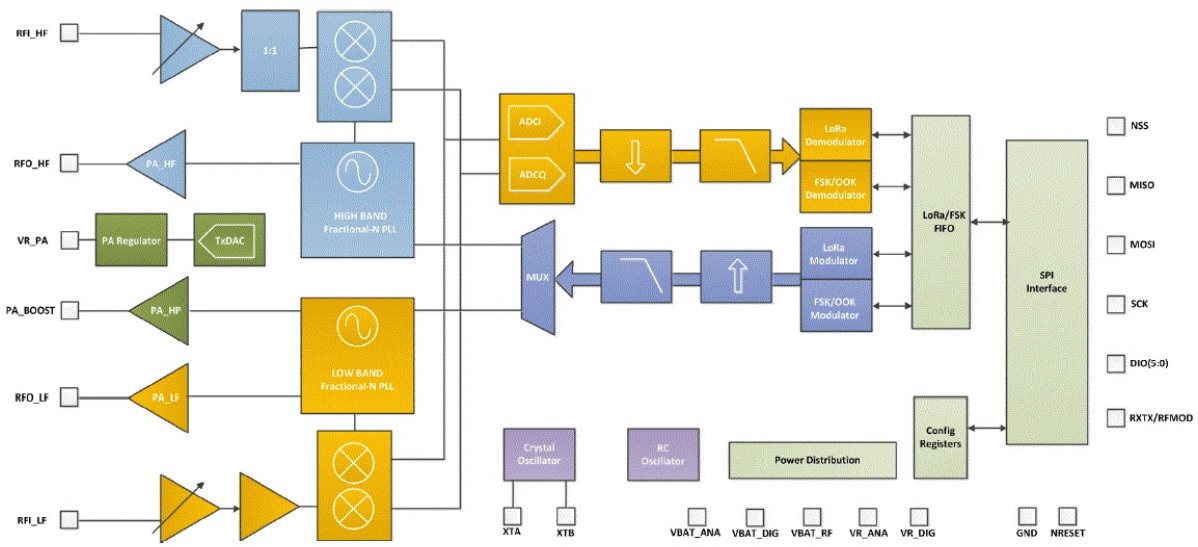

Figure F.1: Block diagram of SX1276 architecture [56]

F.0.2 Main Characteristics of SX1276 chip

- 168 dB maximum link budget
- Max. transmit power of +20 dBm
- +14 dBm high efficiency PA
- Configurable bit-rate up to 300 *kbps*
- High signal sensitivity up to -148 dBm

- Low receive current of 9.9 mA

- FSK, GFSK, MSK, GMSK, LoRa and OOK modulation

- Built-in temperature sensor and low battery indicator

F.0.3 Available chips with SX1276 device

The LoRa powered chips market is quite large and offers lots of models available. Some of them, like the Microchip RN2483, also contain a processing unit that communicates with the transceiver and does all of the computations needed to have a fully functional, easy to use device for long range wireless communications. Microchip's module integrates RF, a base-band controller and a micro-controller unit. The module can be configured and deployed by using Attention (AT) commands, which are sent via UART communication.

Figure F.2: Microchip RN2483 [71]

Other of the available solutions is the HopeRF RFM96W which, like the RN2483, also packs the SX1276 chip, however it does not include any processing unit, being the main difference between the two referred chips. Despite that, the RFM96W contains all the components needed to have a complete LoRa modem, which can be configured via SPI interface. Unlike RN2483, the RFM96 cannot be considered a fully functional device since all of the computations, configurations and implementations of MAC protocols, needed to successfully communicate over long distances with LoRa modulation, must still be conducted. This may impose a problem if a quick and easy deployment is desired.

Figure F.3: HopeRF RFM96W LoRa modem [72]

Appendix G

Microcontroller Unit

G.0.1 General architecture of a micro-controller

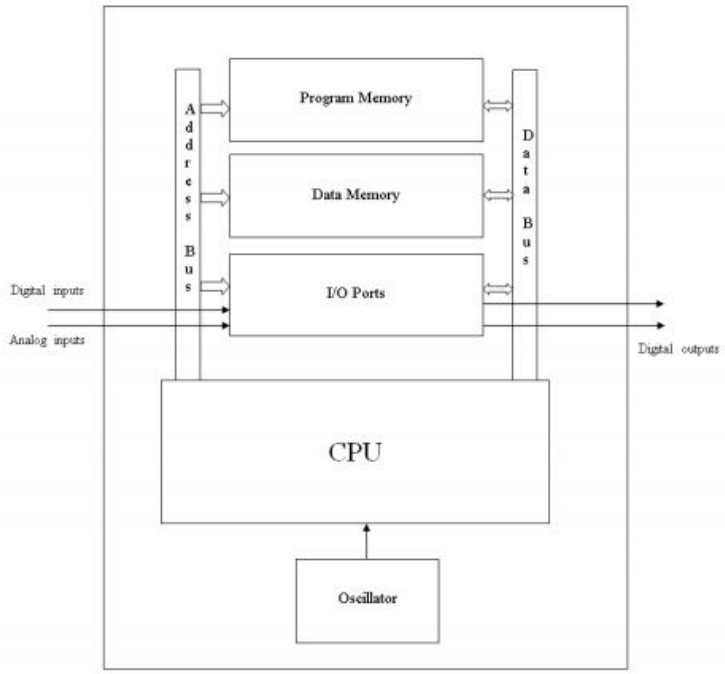

Figure G.1: Block diagram of the main architecture of a micro-controller [**?**]

G.0.2 Main Features of Chosen PIC device

- 64 kBytes of program memory
- Up to 16 MIPS Operation
- 8 x 8 Single-Cycle Hardware Multiplier
- 16 MHz precision Internal Oscillator

- ADC with 10-bit resolution, up to 30 external channels
- Extreme Low Power features
- Up to 35 I/O Pins
- 2 EUSART modules (RS-485 or RS-232)
- 2 MSSP modules (SPI or I2C)

G.0.3 Debugger/Programmers for PIC18F

Figure G.2: Microchip PIC Kit 3 Programmer/Debugger (obtained from Microchip's website)

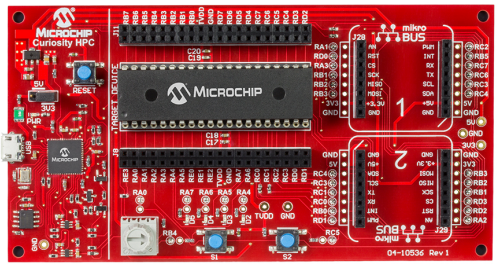

Figure G.3: Microhip Curiosity HPC development board (obtained from Microchip's website)

Appendix H

External Timer

H.0.1 Standard Hardware Configuration

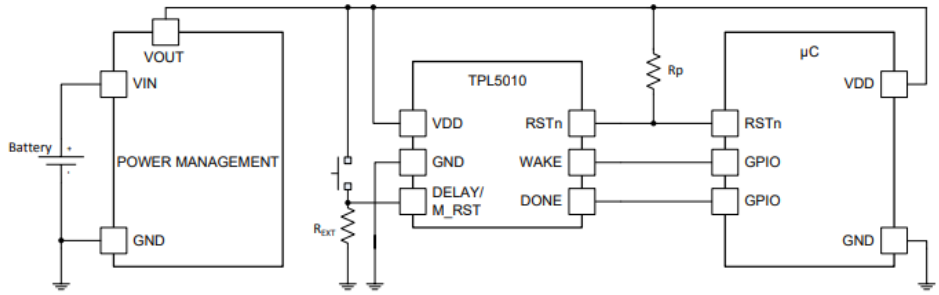

Figure H.1: Schematic of Standard TPL5010 Implementation [57]

Appendix I

Software

I.0.1 MPLAB X Integrated Development Environment

This project aims to develop a custom firmware to enable EIS sweep analysis and send the generated data via LoRaWAN MAC protocol, using the selected Microchip micro-controller. In order to program and debug their chips, Microchip offers a complete development solution called MPLAB X IDE. This IDE can run in all major platforms like Windows, Linux or Mac OS and allows the user to develop the embedded code, either in C or Assembly language, permits the deployment and debugging of the generated code instruction by instruction, enables the analysis of memory scope and variables stored in processing unit and even allows for the simulation of the developed code, without the need for a micro-controller unit.

The compilation of the developed code is performed by the compiler attached to MPLAB X, the XC8 Compiler. This software is responsible for "converting" the C language used to program the device into machine code that can be interpreted by the micro-controller. It incorporates additional features like code optimization among others, however, those functionalities are only available in pro versions. Thankfully, Microchip provides the free version of the compiler which has proven to more than suffice for the creation of the prototype firmware.

MPLAB X IDE can be fitted with extra integrated capabilities by installing optional software components, also developed by Microchip. MPLAB Code Configurator (MCC) is one of those components, designed to provide a free graphical programming environment capable of generating samples of C source code that can be inserted in the firmware project. It presents an intuitive interface, directed to the rapid configuration and initialization of some sets of peripherals and functions according to the targeted PIC architecture. Nowadays,

MCC supports the large majority of 8, 16 and 32-bit PIC micro-controllers. In figure I.1 it is possible to see the development environment to program the PIC devices, along with the MPLAB Code Configurator IDE.

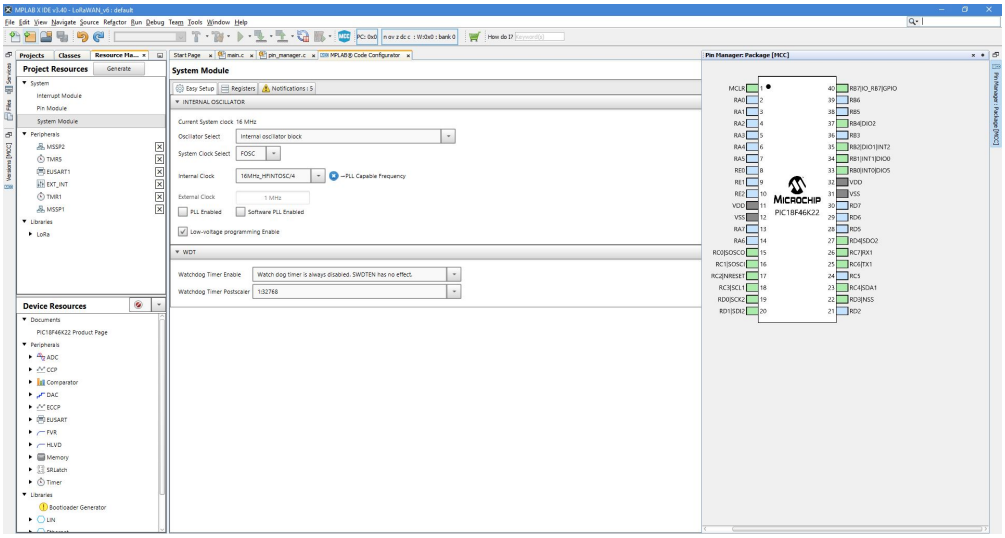

Figure I.1: Microchip MPLAB Code Configurator

Despite Microchip providing regular updates to all its software, the versions used in this project were not the most up-to-date. This was due to requirements of some of the plug-ins used, namely the LoRaWAN stack. The existing constraints generated the following list of software requirements:

- MPLAB X IDE 3.40;
- MPLAB Code Configurator v3.25;
- XC8 compiler v1.38.

I.0.2 LoRaWAN Plug-In Library for MCC

The LoRaWAN Library is a so called plug-in, aimed for integration with MCC, which enables developers to generate snippets of C code of Microchip's LoRaWAN stack solution for LoRa end-devices. In the core of Microchip's LoRaWAN stack is the code developed for RN2483 chip which implements a Class A communication link based on LoRa modulation, with specific transceivers like SX1276.

This library only supports the integration with 8 bit PICs and is supplied by Microchip as an archive to be added to MCC in order to be used. After that, the graphical interface provided by Code Configurator gives to the user the possibility to [58]:

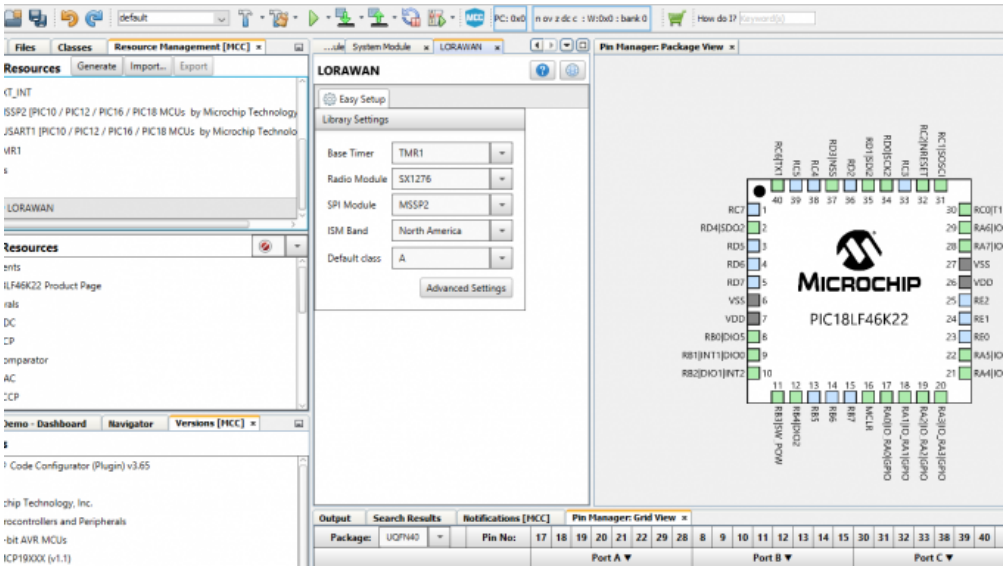

Figure I.2: GUI of MPLAB Code Configurator with LoRaWAN Stack (courtesy of BeyongLogic)

- Select the requirements for the LoRaWAN stack from the device resources (modules present on the device);
- Define the communication channels for the EU 433/868 MHz ISM band;
- Enable/Disable communication channels for the NA 915 MHz ISM band;
- Adjust various parameters of the LoRaWAN stack regarding the communication with the server;
- Generate the necessary C code to program a PIC micro-controller.

In the scope of this project, the main reason for using this library to implement the LoRaWAN MAC layer is due to the out-of-scope work and prohibitive amount of time needed to implement this MAC from the ground up.

Appendix J

Firmware Development

J.1 RFM95

J.1.1 Conceptual Structure of LoRaWAN firmware

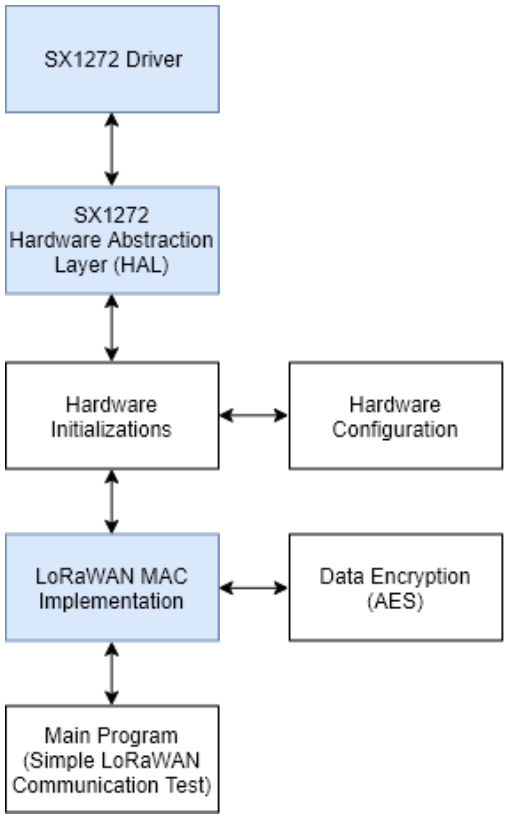

Figure J.1: Used libraries and structure of the project

The blocks with blue background represent the main functions that are implemented by the LoRaWAN stack.

J.1.2 First iteration of firmware

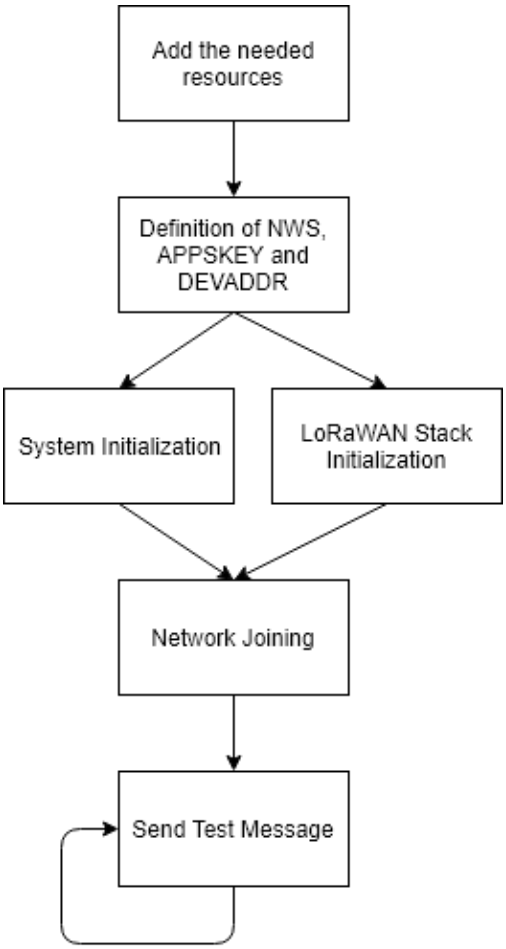

Figure J.2: Flowchart of the communication firmware developed

In figure J.2 it is possible to see the execution line of the program developed to test the LoRaWAN communication. It starts by including the needed libraries and resources into the program, followed by the inline definition of network specific parameters. After that, the main function is called, which starts to configure the system (peripherals, timers, etc..), followed by the initialization of the communication stack.

When everything is configured, the system tries to establish communication and join the network. In order to control this process of attempted connection between the end-node and network, a function called *RxJoinResponse* was customized to simply toggle a variable based on the join response status. If the control variable is valid, the program calls the send function of the LoRaWAN stack and sends data to the network.

J.1.3 LoRaWAN capabilities in the developed Firmware

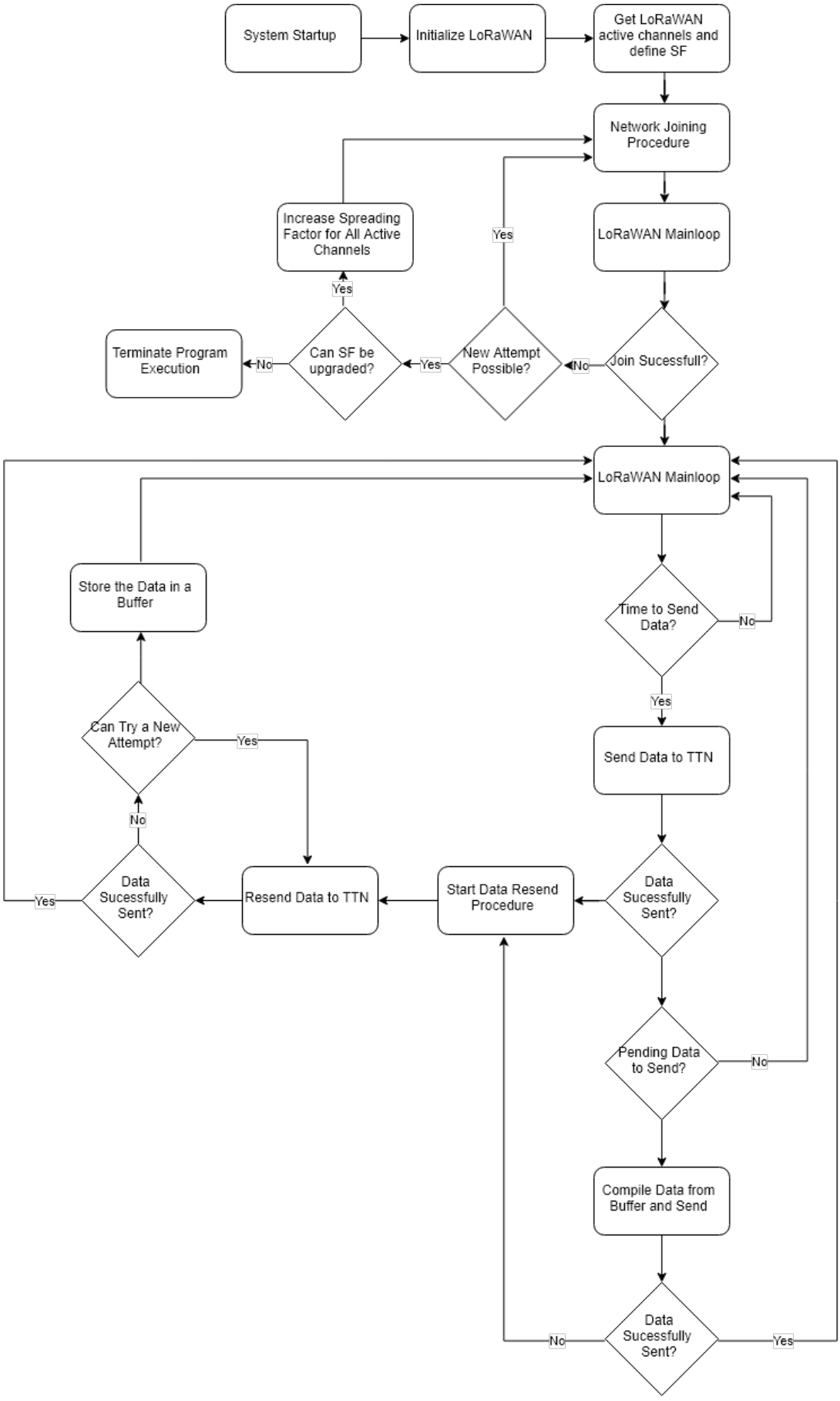

Figure J.3: Flowchart of the specific firmware execution for LoRaWAN communication

J.2 AD5933

J.2.1 MCU interface configuration with MCC for AD5933

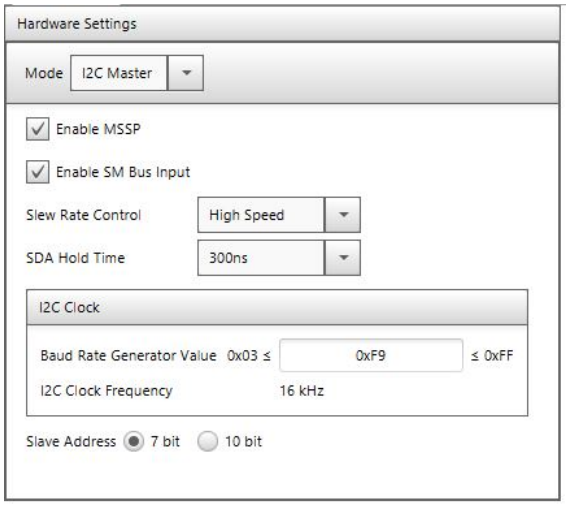

Figure J.4: Hardware configurations for MSSP1, inputted in MPLAB MCC

J.2.2 Most Used I2C function in PIC18

Table J.1: Definition of some of the most used functions to establish I2C communication

I2C Function	Main Function
Initialize	Function where all of the hardware related aspects are configured, based on the information provided in the configuration stage of the peripheral in MCC. I2C mode, SDA Hold time or clock frequency are some of the aspects configured.
MasterReadTRBBuild	This function is responsible for constructing a TRB that performs a read operation in I2C lines. The needed parameters must be supplied in the correct form as input arguments and the function, when called, allocates the respective structures with the supplied parameters.
MasterWriteTRBBuild	It has a similiar function to I2C1_MasterReadTRBBuild but instead it configures a “write” transaction which will lead to a “read” operation in I2C lines.
MasterTRBInsert	This function aggregates the TRBs defined by the user with the two previous functions and inserts them in a I2C transaction queue along with a pointer to the completion flag.

J.2.3 Example of Custom TRB used in I2C communication

Listing J.1: Example of custom function for AD5933, based on TRBs

```
1
2  //function to read one byte of AD5933 using TRB approach
3  uint8_t AD5933_ReadByte(uint8_t reg, uint8_t *pData)
4  {
5      I2C1_MESSAGE_STATUS status = I2C1_MESSAGE_PENDING;
6      static I2C1_TRANSACTION_REQUEST_BLOCK trb[2];
7
8      uint8_t I2CMasterBuffer[2];
9      //before we read a byte, we need to positioning the read pointer of the device.
10     //to do that, the command 0b10110000 needs to be sent.
11     I2CMasterBuffer[0] = 0b10110000;
12     //the register we need to read
13     I2CMasterBuffer[1] = reg;
14
15     I2C1_MasterWriteTRBBuild(&trb[0], &I2CMasterBuffer, 2, AD5933_ADDRESS);
16     //after the write, the i2c interface signals a repeated start and send the
17     //device address with read bit, and waits for ad5933 data.
18     I2C1_MasterReadTRBBuild(&trb[1], pData, 1, AD5933_ADDRESS);
19     //starts the i2c transaction.
20     I2C1_MasterTRBInsert(2, &trb[0], &status);
21
22     //if the message is pending, block execution
23     while(status == I2C1_MESSAGE_PENDING);
24     //returns 1 if the procedure completed succesfully.
25     return (status == I2C1_MESSAGE_COMPLETE);
26 }
```

J.2.4 List of Developed Functions and Respective Behaviour

J.2.5 Matlab Script to Generate Frequency Hex Values

Listing J.2: Matlab script for determine the hexadecimal values to program AD5933

```
1  clear
2  clc
3  close all
4  %%Frequency Parameters
5  %frequency in hertz
6  startFrequency=40e3;
7  stopFrequency=60e3;
8  incrementFrequency=0.25e3;
9  clock=16.776e6;
10 %%Frequency Formula
11 res1=clock/4;
12 val=(startFrequency/res1)*(2^27);
13 %calculate HEX to write
14 frequencyHEX=dec2hex(uint32(val))
15 valInc=(incrementFrequency/res1)*(2^27);
16 incrementFreqHEX=dec2hex(uint32(valInc))
17 %determine the number of increment needed to achieve the stop frequency
18 buffer=0;
19 j=0;
20 while(buffer<stopFrequency)
21     frequencyPoints(j+1)=startFrequency+j*incrementFrequency;
22     buffer=frequencyPoints(j+1);
23     j=j+1;
24 end
25 numberIncrements=j
26 numberIncrementsHEX=dec2hex(numberIncrements)
```

Table J.2: Explanation of the developed functions for AD5933

Name	Objective
ReadByte	This function, when called, reads a byte of an AD5933 register, defined as a function input, and stores the raw data in a register, whose pointer is also inputted into the function.
WriteByte	Perfoms a "write" operation in the chip. Takes as input the address of the register to write and 8-bit data to be written.
GetTemperature	Communicates with the AD5933 and triggers a temperature measurement. Gets the two 8-bit data stored in the temperature specific registers and performs bitwise operations to combine the read registers in order to output a decimal value representing the temperature.
PointerPositioning	Function to position the memory pointer of the AD5933 in a specific address. Used when we want to write more than one byte in a single transaction.
WriteStartFrequency	Writes in the register containing the start frequency of EIS analysis, the values passed to the function as input. It takes three 8-bit values, which combined represent the decimal value of the start frequency.
WriteIncFrequency	It has a similar behaviour as WriteStartFrequency, differing only in the addresses of the registers where the information will be written.
WriteIncNumber	Function similar to the two previous defined functions. Takes as input two 8-bit registers representing the number of frequency increments needed in the EIS analysis.
WriteSettlingTime	The amount of time that the sample requires after the excitation phase and before the acquisition phase is inputted as an 8-bit variable, which will be written in the respective register of the AD5933.
ChangeMode	This function is responsible for changing the operational state of the AD5933. It takes as input the decimal number representing the desired state and computes the I2C transaction needed to do that.

J.2.6 Flowchart of the developed Firmware

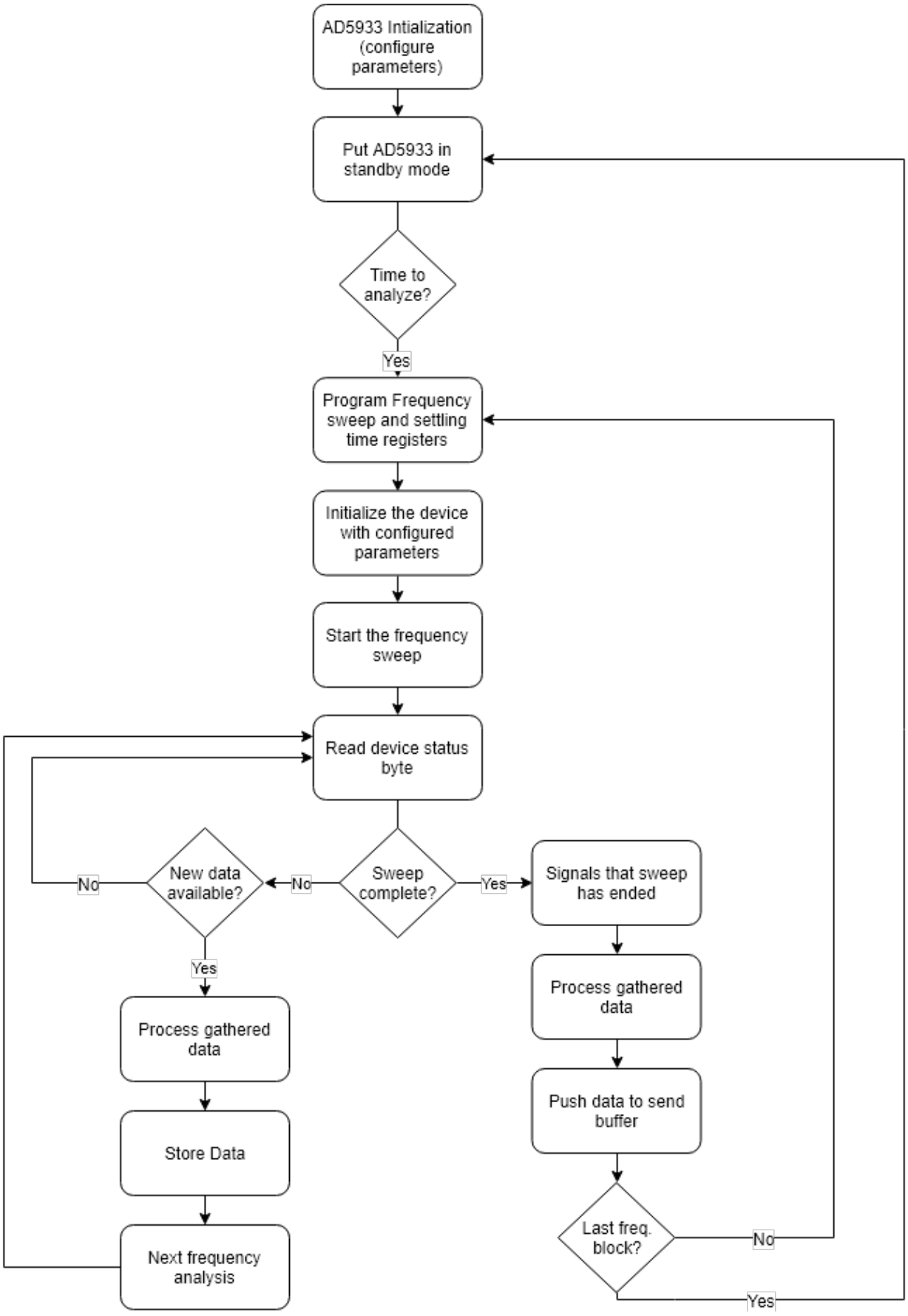

Figure J.5: Flowchart of the developed firmware for a complete EIS analysis

J.3 TPL5010

J.3.1 System Behaviour for Periodic Executions

Figure J.6: Updated firmware execution flow to allow MCU periodic awakenings

J.3.2 Function to enable Sleep Mode of SX1276

Listing J.3: Developed C method to deploy the sleep mode in SX1276

```
1 void RADIO_Sleep(void)
2 {
3     //Variable to store the actual configurations and operational mode of the device
4     uint8_t opMode;
5     //get the data from the device via SPI interface
6     opMode = RADIO_RegisterRead(REG_OPMODE);
7     //Perform bitwise operation with a mask composed by 0b11111000 and writes the result in the device
8     RADIO_RegisterWrite(REG_OPMODE, opMode & (~0x07));
9 }
```

When called, the procedure reads the current value of the register and stores it in a variable. After that, a bitwise operation is performed, executing a logical conjunction between the read value and the mask $0b11111000$. This operation changes only the three least significant bits of the register and leaves the other five bits untouched, preserving the configuration. Finally, the value is written in the device in order to trigger the sleep mode.

Appendix K

Hardware

K.1 RFM 95 Transceiver

K.1.1 Pin Functions of RFM95 Transceiver

Table K.1: RFM95 transceiver used pins and respective functions

RFM95 Pin Name	Pin Function
GND	Ground for voltage and logic functions
MISO	Master In Slave Out pin responsible for sending data from the transceiver to the selected processing unit. $3.3V$ logic
MOSI	Master Out Slave In pin used for send information from microcontroller to the transceiver module. It is $3.3V$ logic like MISO
SCK	SPI interface clock pin. Used so synchronize the data transmission between Master and Slave. It is a input to the chip
RESET	Radio reset pin. It is an input and it is pulled HIGH to reset the device. When LOW the radio is turned on
DIO5, DIO2, DIO1	IRQ pins used for notifications or specific functions. They can be configured in the chip to trigger an interrupt when a specific event occurs. Like the remaining pins, they are $3.3V$ logic
DIO3, DIO4	Not used
$3.3V$	Power supply pin. The chip is powered at $3.3V$, and the power supply module of the system should be capable of delivering currents up to $150mA$, since the radio peaks can be high.
ANT	Antenna signal Pin

K.1.2 Selected Shield for RFM95

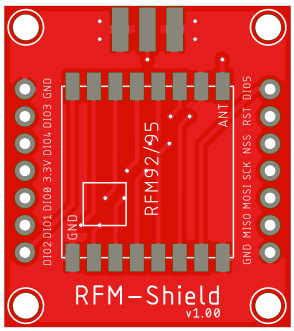

Figure K.1: Selected RFM95 shield (courtesy of Diycon)

K.2 AD5933 Chip

K.2.1 AD5933 Pin List and Functions

Table K.2: AD5933 pin names and respective functions

AD5933 Pin Name	Pin Function
NC	Not Connected
R_{FB}	Set the gain of the current-to-voltage amplifier embedded in AD5933 in receiver side. A External Feedback Resistor is connected between R_{FB} and V_{IN} to set the gain
V_{IN}	Input pin to the Transimpedance Amplifier in receiver side. It presents a virtual earth voltage of $VDD/2$
V_{OUT}	Pin responsible for providing the excitation signal used to stimulate the SUT
MCLK	Supplys the master clock signal defined by the user to the device
DVVD	Digital supply voltage
AVDD1, AVDD2	Analog voltage supply
DGND	Digital ground
AGND1, AGND2	Analog ground pins
SDA	Data line used in the I2C communication. It is a open-drain pin that requires a pull-up resistor
SCL	I2C clock line. It is a open-drain pin, like SDA so it also requires a pull-up resistor

K.2.2 Selected Shield for AD5933

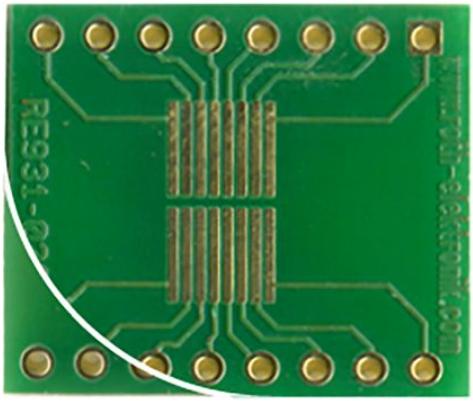

Figure K.2: Adopted shield for AD5933 development

K.2.3 Proposed architecture for Dual Feedback Resistors (R_{FB})

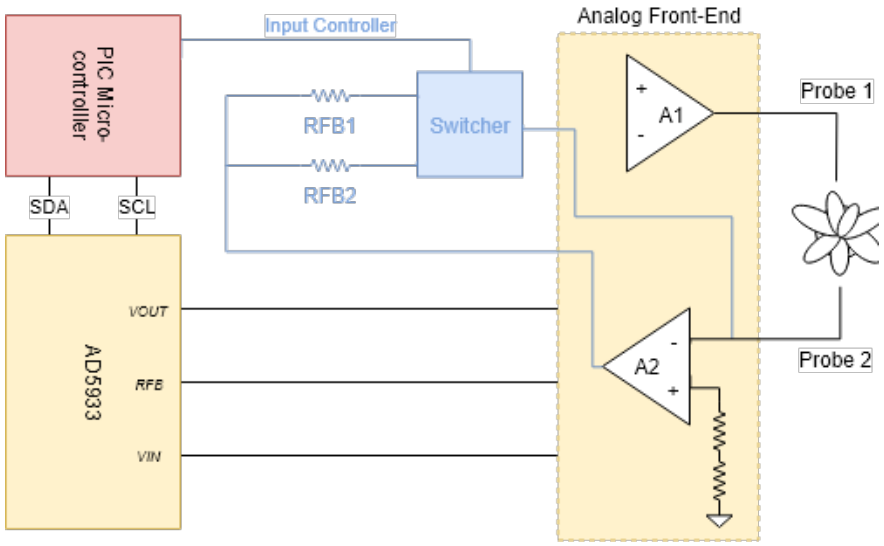

Figure K.3: Circuit architecture to support the active change of R_{FB}

K.2.4 Matlab Script for Calculation of system resistors

Listing K.1: Matlab developed code for calculating R_{FB} and R_{CAL}

```
%RFB calculation based on impedance range

%VPK is the peak voltage of the selected output range.
vpk=1
%ZMIN is the minimum impedance.
zmin=[1e3 14e3]
%zmax values mus obey to max allowed ratio in datasheet. if we use aditional external hardware
%for rebiasing, the max ratio is x45
zmax=[15e3 200e3]
%GAIN is the selected PGA gain, 1 or 5.
gain=1
```

```
%VDD is the supply.
vdd=3.3
%VDCOFFSET is the dc offset voltage for the selected range shown in Table 1.
vdoff=vdd/2

[l,c]=size(zmin)
for r = 1:c
    num=((vdd/2)-0.2)*zmin(r);
    den=(vpk+(vdd/2)-vdoff);
    rfb=(num/den)*(1/gain);
    X = ['RFB for Zmin of ',num2str(zmin(r)),' Ohm = ',num2str(rfb),' Ohms'];
    disp(X)

    rcal=(zmin(r)+zmax(r))*(1/3);
    X = ['Rcal for impedance range between ',num2str(zmin(r)),' Ohm and ',num2str(zmax(r)),' Ohm = ',
        num2str(rcal), 'Ohms'];
    disp(X)
end
```

K.3 TPL5010

K.3.1 Selected Shield

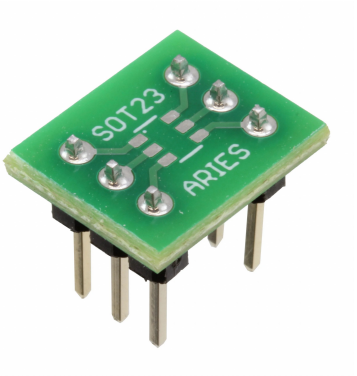

Figure K.4: Used adapter for TPL5010 (courtesy of Digikey)

Appendix L

Real Data

L.1 Wireless Evaluation

L.1.1 Terminal output of prototype, when sending data to TTN

Listing L.1: Terminal output generated by the prototype

```
Activacao OTAA
 -- DETALHES --
APP EUI: 70 b3 d5 7e d0 1 50 b3
DEV EUI: ad 75 45 44 66 cd ef ab
APP KEY: e f6 ac 27 22 8c 4b 44 49 5b f2 b7 40 4d 0 13
Canal 0 ACTIVO | Freq: 868100000 | DR: 55
Canal 1 ACTIVO | Freq: 868300000 | DR: 55
Canal 2 ACTIVO | Freq: 868500000 | DR: 55
[INFO AD5933] - Chip in Standby
Estabelecendo Conexao com Rede
Connection Status 1
END!
Pontos Adquiridos: 41
0x80 value: 177
[INFO LORAWAN] - Sent Successfull
[INFO AD5933] - Power-Down
[INFO PIC] - Enter Sleep
```

L.1.2 End-device console report with down-link capabilities

Listing L.2: Console output of a periodic system execution with downlink message terminal

```
[INFO PIC] - Running Mode
[INFO] - Temperature Read: 28
Config Mode
First Frequency Sweep Selected: 1->10 KHz
[INFO AD5933] - Device Initialized
[INFO AD5933] - Frequency sweep initialized
END!
Pontos Adquiridos: 37
0x80 value: 177
Config Mode
Second Frequency Sweep Selected: 20->30 KHz
[INFO AD5933] - Device Initialized
[INFO AD5933] - Frequency sweep initialized
```

```
END!
Pontos Adquiridos: 41
0x80 value: 177
Config Mode
Third Frequency Sweep Selected: 45->55 KHz
[INFO AD5933] - Device Initialized
[INFO AD5933] - Frequency sweep initialized
END!
Pontos Adquiridos: 41
0x80 value: 177
Config Mode
Fourth Frequency Sweep Selected: 70->80 KHz
[INFO AD5933] - Device Initialized
[INFO AD5933] - Frequency sweep initialized
END!
Config Mode
Fifth Frequency Sweep Selected: 85->95 KHz
[INFO AD5933] - Device Initialized
[INFO AD5933] - Frequency sweep initialized
END!
Pontos Adquiridos: 41
0x80 value: 177
[INFO LORAWAN] - Sent Successfull
BYTE 1 Recebido: ab
BYTE 2 Recebido: ff
BYTE 3 Recebido: dc
[INFO AD5933] - Power-Down
[INFO PIC] - Enter Sleep
```

L.2 Complete frequency sweep for calibration purposes

Listing L.3: Console fragment output of the prototype with raw data from all of the frequency sweeps, used for determining the gain factor for each frequency bin analysis

```
[INFO PIC] - Running Mode
[INFO] - Temperature Read: 23
Config Mode
First Frequency Sweep Selected: 1->10 KHz
[INFO AD5933] - Device Initialized
[INFO AD5933] - Frequency sweep initialized
-415,-1980;
-517,-2058;
-491,-2139;
-421,-2141;
-416,-2104;
-452,-2070;
-461,-2113;
-409,-2135;
-395,-2102;
-416,-2083;
-439,-2125;
-420,-2176;
-406,-2157;
-417,-2152;
-407,-2185;
-398,-2199;
-378,-2163;
-384,-2188;
-404,-2255;
-372,-2247;
-353,-2202;
-357,-2191;
-351,-2208;
-333,-2231;
-306,-2227;
```

```
 32  -300,-2224;
 33  -305,-2236;
 34  -307,-2248;
 35  -295,-2264;
 36  -285,-2286;
 37  -288,-2293;
 38  -282,-2308;
 39  -268,-2313;
 40  -266,-2315;
 41  -251,-2330;
 42  -246,-2344;
 43  -232,-2338;
 44  -221,-2336;
 45  -219,-2348;
 46  -206,-2363;
 47  -189,-2360;
 48  END!
 49  Pontos Adquiridos: 41
 50  0x80 value: 176
 51  [INFO AD5933] - Device Reset
 52  Config Mode
 53  Second Frequency Sweep Selected: 20->30 KHz
 54  [INFO AD5933] - Device Initialized
 55  [INFO AD5933] - Frequency sweep initialized
 56  14,-2445;
 57  20,-2459;
 58  34,-2469;
 59  42,-2447;
 60  48,-2452;
 61  69,-2474;
 62  73,-2525;
 63  85,-2523;
 64  96,-2521;
 65  114,-2517;
 66  120,-2518;
 67  132,-2531;
 68  152,-2539;
 69  161,-2546;
 70  175,-2538;
 71  183,-2501;
 72  181,-2547;
 73  203,-2507;
 74  233,-2518;
 75  229,-2553;
 76  249,-2547;
 77  261,-2487;
 78  275,-2477;
 79  291,-2541;
 80  303,-2544;
 81  302,-2555;
 82  322,-2544;
 83  326,-2541;
 84  338,-2554;
 85  351,-2552;
 86  364,-2568;
 87  370,-2565;
 88  381,-2531;
 89  396,-2560;
 90  412,-2550;
 91  416,-2554;
 92  437,-2548;
 93  449,-2555;
 94  457,-2573;
 95  473,-2547;
 96  483,-2553;
 97  END!
 98  Pontos Adquiridos: 41
 99  0x80 value: 176
100  [INFO AD5933] - Device Reset
```

```
101 Config Mode
102 Third Frequency Sweep Selected: 45->55 KHz
103 [INFO AD5933] - Device Initialized
104 [INFO AD5933] - Frequency sweep initialized
105 1255,-2453;
106 1271,-2451;
107 1286,-2454;
108 1302,-2446;
109 1319,-2455;
110 1329,-2453;
111 1345,-2512;
112 1360,-2501;
113 1375,-2533;
114 1394,-2547;
115 1409,-2549;
116 1411,-2536;
117 1429,-2524;
118 1460,-2579;
119 1466,-2564;
120 1483,-2616;
121 1496,-2588;
122 1522,-2604;
123 1532,-2591;
124 1542,-2597;
125 1563,-2595;
126 1577,-2581;
127 1577,-2574;
128 1597,-2587;
129 1612,-2579;
130 1621,-2562;
131 1647,-2559;
132 1651,-2547;
133 1663,-2554;
134 1678,-2531;
135 1684,-2527;
136 1698,-2517;
137 1715,-2522;
138 1728,-2511;
139 1732,-2498;
140 1750,-2491;
141 1765,-2472;
142 1774,-2489;
143 1785,-2474;
144 1803,-2469;
145 1812,-2453;
146 END!
147 Pontos Adquiridos: 41
148 0x80 value: 176
149 [INFO AD5933] - Device Reset
150 Config Mode
151 Fourth Frequency Sweep Selected: 70->80 KHz
152 [INFO AD5933] - Device Initialized
153 [INFO AD5933] - Frequency sweep initialized
154 2491,-1918;
155 2509,-1882;
156 2514,-1869;
157 2512,-1866;
158 2533,-1839;
159 2537,-1830;
160 2537,-1809;
161 2577,-1836;
162 2585,-1834;
163 2597,-1823;
164 2617,-1814;
165 2603,-1774;
166 2610,-1773;
167 2629,-1760;
168 2641,-1720;
169 2630,-1720;
```

```
2636,-1711;
2663,-1729;
2666,-1696;
2681,-1704;
2690,-1681;
2713,-1676;
2736,-1686;
2749,-1688;
2753,-1653;
2759,-1656;
2779,-1666;
2801,-1652;
2799,-1631;
2817,-1643;
2838,-1615;
2828,-1597;
2838,-1622;
2848,-1631;
2856,-1608;
2859,-1598;
2867,-1573;
2866,-1513;
2869,-1504;
2882,-1482;
2926,-1507;
END!
Pontos Adquiridos: 41
0x80 value: 176
[INFO AD5933] - Device Reset
Config Mode
Fifth Frequency Sweep Selected: 85->95 KHz
[INFO AD5933] - Device Initialized
[INFO AD5933] - Frequency sweep initialized
3076,-1251;
3103,-1271;
3117,-1242;
3153,-1271;
3139,-1258;
3161,-1244;
3174,-1244;
3175,-1200;
3168,-1187;
3156,-1161;
3169,-1136;
3174,-1137;
3180,-1109;
3192,-1112;
3215,-1111;
3208,-1091;
3219,-1077;
3232,-1073;
3221,-1051;
3238,-1036;
3238,-1020;
3248,-1001;
3260,-982;
3254,-964;
3269,-945;
3277,-950;
3283,-935;
3282,-912;
3304,-896;
3295,-879;
3302,-873;
3292,-846;
3324,-848;
3337,-834;
3329,-811;
3348,-806;
```

```
3349,-786;
3343,-776;
3352,-762;
3351,-730;
3366,-736;
END!
Pontos Adquiridos: 41
0x80 value: 176
[INFO LORAWAN] - Sent Successfull
[INFO AD5933] - Power-Down
[INFO PIC] - Enter Sleep
AD5933 Hard-Sleep
SX1276 Forced Sleep
```

L.3 Matlab script to calibrate the device

Listing L.4: Matlab script to calculate gain factor based on a single frequency sweep

```
clear
clc
close all

if exist('bothCalibrations.eps', 'file')==2
  delete('bothCalibrations.eps');
end
if exist('centerFrequencies.eps', 'file')==2
  delete('centerFrequencies.eps');
end
if exist('allFrequencies.eps', 'file')==2
  delete('allFrequencies.eps');
end
if exist('calibrationValues.mat', 'file')==2
  delete('calibrationValues.mat');
end


%parameters needed
meanOfAcqs=0;

freqSweeps=5;
startFreqs=[5e3,20e3,45e3,70e3,85e3];
stopFreqs=[15e3,30e3,55e3,80e3,95e3];
calibFreq=[10e3,25e3,50e3,75e3,90e3];
incrementFrequency=0.25e3;
impedanceCalib=100e3;

if(meanOfAcqs==1)
    load('meanAcqs.mat')
else

values1=[
647,-18068;
655,-18094;
768,-18142;
889,-18156;
981,-18104;
1014,-18103;
1091,-18138;
1203,-18138;
1286,-18106;
1333,-18089;
1396,-18102;
1472,-18096;
1551,-18068;
1613,-18056;
1673,-18071;
```

```
1756,-18076;
1836,-18060;
1888,-18048;
1961,-18053;
2036,-18046;
2111,-18031;
2176,-18023;
2201,-18024;
2292,-18022;
2365,-18006;
2415,-17992;
2499,-17972;
2555,-17969;
2612,-17967;
2670,-17956;
2731,-17943;
2816,-17938;
2879,-17926;
2932,-17913;
2987,-17908;
3098,-17882;
3144,-17879;
3200,-17877;
3265,-17862;
3354,-17854;
3411,-17840;
]

values2=[
4621,-17578;
4677,-17559;
4734,-17542;
4810,-17524;
4885,-17497;
4941,-17486;
4999,-17463;
5088,-17449;
5164,-17420;
5208,-17414;
5286,-17387;
5319,-17379;
5374,-17361;
5453,-17341;
5501,-17317;
5536,-17312;
5603,-17295;
5645,-17278;
5713,-17244;
5786,-17235;
5832,-17210;
5906,-17192;
5957,-17171;
6019,-17143;
6073,-17134;
6122,-17116;
6195,-17095;
6249,-17075;
6300,-17051;
6364,-17039;
6455,-17007;
6535,-16966;
6581,-16946;
6636,-16931;
6702,-16901;
6747,-16880;
6812,-16860;
6846,-16840;
6890,-16825;
6955,-16797;
```

```
118  7033,-16770;
119  ];
120
121  values3=[
122  10265,-15026;
123  10365,-14954;
124  10399,-14934;
125  10470,-14892;
126  10516,-14856;
127  10528,-14842;
128  10604,-14790;
129  10665,-14744;
130  10769,-14672;
131  10751,-14682;
132  10822,-14623;
133  10886,-14581;
134  10922,-14553;
135  10970,-14511;
136  11050,-14458;
137  11082,-14433;
138  11118,-14399;
139  11139,-14394;
140  11218,-14330;
141  11239,-14314;
142  11263,-14286;
143  11350,-14223;
144  11360,-14212;
145  11408,-14176;
146  11444,-14139;
147  11532,-14076;
148  11549,-14056;
149  11625,-14005;
150  11668,-13962;
151  11673,-13960;
152  11718,-13902;
153  11770,-13872;
154  11837,-13805;
155  11896,-13760;
156  11933,-13720;
157  11972,-13702;
158  12065,-13620;
159  12137,-13561;
160  12151,-13537;
161  12232,-13462;
162  12240,-13462;
163  ]
164
165  values4=[
166  14715,-10679;
167  14725,-10643;
168  14761,-10596;
169  14762,-10587;
170  14813,-10513;
171  14861,-10453;
172  14798,-10548;
173  14924,-10351;
174  14946,-10325;
175  14999,-10230;
176  15013,-10215;
177  15082,-10112;
178  15095,-10088;
179  15132,-10032;
180  15162,-9989;
181  15185,-9937;
182  15254,-9846;
183  15274,-9807;
184  15313,-9747;
185  15335,-9718;
186  15379,-9642;
```

```
15414,-9572;
15475,-9474;
15482,-9471;
15499,-9433;
15482,-9425;
15573,-9312;
15594,-9270;
15630,-9215;
15665,-9168;
15677,-9114;
15717,-9062;
15755,-8998;
15820,-8887;
15747,-9003;
15750,-8994;
15810,-8885;
15826,-8869;
15847,-8817;
15930,-8662;
15930,-8655;
]


values5=[
16480,-7559;
16503,-7463;
16539,-7384;
16581,-7274;
16634,-7154;
16672,-7075;
16687,-7028;
16710,-6965;
16741,-6892;
16733,-6880;
16770,-6802;
16790,-6762;
16824,-6683;
16849,-6614;
16843,-6587;
16855,-6602;
16881,-6513;
16914,-6436;
16939,-6356;
16971,-6238;
16991,-6198;
16987,-6203;
17001,-6171;
17035,-6064;
17041,-6067;
17062,-5975;
17088,-5905;
17121,-5830;
17132,-5770;
17160,-5691;
17173,-5592;
17195,-5562;
17225,-5480;
17242,-5434;
17244,-5409;
17256,-5391;
17278,-5301;
17287,-5237;
17325,-5143;
17310,-5142;
17339,-5050;
]

end
```

```
256
257
258
259
260
261
262 for pointer=1:freqSweeps
263     %frequency points calculation
264     buffer=0;
265     j=0;
266     while(buffer<stopFreqs(pointer))
267         frequencyPoints(j+1)=startFreqs(pointer)+j*incrementFrequency;
268         buffer=frequencyPoints(j+1);
269         j=j+1;
270     end
271     %change name of vars
272     valuesEIS=eval(sprintf('values%d',pointer));
273     %get array size
274     [l,c]=size(valuesEIS);
275     %perform conversions in all lines of array
276     for i = 1:l
277         %16bit decimal value of real part
278 %         unsignedReal=swapbytes(typecast(uint8(valuesEIS(i,1:2)),'uint16'));
279 %         real(i)=double(typecast( uint16(unsignedReal), 'int16'));
280         real(i)=valuesEIS(i,1)
281         %16bit decimal value of imag part
282 %         unsignedImag=swapbytes(typecast(uint8(valuesEIS(i,3:4)),'uint16'));
283 %         imag(i)=double(typecast( uint16(unsignedImag), 'int16'));
284         imag(i)=valuesEIS(i,2)
285     end
286
287     %calculate magnitude
288     magnitudePoints = sqrt(real.*real+imag.*imag);
289
290     %Center Frequency Calibration
291     %get index of selected calibration frequency
292     idx=find(frequencyPoints==calibFreq(pointer));
293     %calculate magnitude
294     magnitudeCenterFreq=sqrt(real(idx)^2+imag(idx)^2);
295     %calculate gain factor
296     GF=(1/impedanceCalib)/(magnitudeCenterFreq);
297
298     %All Frequencies Calibration
299     gains=(1/impedanceCalib)./(magnitudePoints);
300     if(pointer==1)
301         frequencysAnalised=frequencyPoints;
302         GF_AllFreqs=gains;
303         GF_CenterFreqs=GF*ones(1,l);
304     else
305         GF_AllFreqs=horzcat(GF_AllFreqs,gains);
306         GF_CenterFreqs=horzcat(GF_CenterFreqs,GF*ones(1,l));
307         frequencysAnalised=horzcat(frequencysAnalised,frequencyPoints);
308     end
309 %     GF_AllFreqs(pointer,1:length(gains))=gains;
310
311 end
312         data(1,:)=frequencysAnalised;
313         data(2,:)=GF_CenterFreqs;
314         data(3,:)=GF_AllFreqs;
315
316
317
318
319  plot(data(1,:),data(2,:),'r*-')
320  grid on
321  ylabel('Calculated Gain Factor (GF)')
322  xlabel('Calibration Frequencies [Hz]')
323  hold on
324  plot(data(1,:),data(3,:),'b*-')
```

```
 legend('Center Frequency','All Frequencies')



 answer = questdlg('Which configuration would you like to export', ...
        'Export Data', ...
        'Center Frequencies','All Frequencies','Center Frequencies');

print -depsc bothCalibrations
close all
% Handle response
switch answer
    case 'Center Frequencies'
        disp([answer ' coming right up.'])
        plot(data(1,:),data(2,:),'r*-')
        grid on
        ylabel('Calculated Gain Factor (GF)')
        xlabel('Calibration Frequencies [Hz]')
        print -depsc centerFrequencies
        calibration(1,:)=data(1,:);
        calibration(2,:)=data(2,:);
        save('calibrationValues.mat','calibration');

    case 'All Frequencies'
        disp([answer ' coming right up.'])
        plot(data(1,:),data(3,:),'b*-')
        grid on
        ylabel('Calculated Gain Factor (GF)')
        xlabel('Calibration Frequencies [Hz]')
        print -depsc allFrequencies
        calibration(1,:)=data(1,:);
        calibration(2,:)=data(3,:);
        save('calibrationValues.mat','calibration');
end
```

L.4 Matlab script to determine unknown impedance

Listing L.5: Script to obtain the value of an unknown impedance

```
clear
clc
close all

startFreqs=[5e3,20e3,45e3,70e3,85e3];
stopFreqs=[15e3,30e3,55e3,80e3,95e3];
calibFreq=[10e3,25e3,50e3,75e3,90e3];
incrementFrequency=0.25e3;

usedImpedance=47e3;



values1=[
-8731,-17697;
-8826,-17702;
-8944,-17698;
-8872,-17710;
-8856,-17672;
-8924,-17664;
-8976,-17684;
-8981,-17672;
-8860,-17693;
-8916,-17678;
-8951,-17692;
-9029,-17661;
```

```
-8993,-17666;
-9101,-17614;
-9045,-17672;
-9053,-17682;
-9034,-17687;
-8969,-17724;
-8940,-17752;
-8983,-17736;
-8921,-17758;
-8975,-17751;
-8874,-17801;
-8914,-17816;
-8918,-17809;
-8848,-17835;
-8816,-17859;
-8808,-17882;
-8765,-17900;
-8673,-17942;
-8634,-17962;
-8621,-17985;
-8574,-18008;
-8515,-18039;
-8439,-18070;
-8357,-18106;
-8331,-18132;
-8224,-18166;
-8254,-18177;
-8198,-18208;
-8143,-18229;
]

values2=[
-6894,-18806;
-6855,-18815;
-6747,-18858;
-6688,-18884;
-6641,-18898;
-6556,-18939;
-6531,-18945;
-6435,-18988;
-6326,-19029;
-6247,-19050;
-6114,-19106;
-6099,-19111;
-6030,-19132;
-5861,-19187;
-5881,-19181;
-5746,-19229;
-5688,-19243;
-5515,-19297;
-5450,-19316;
-5447,-19305;
-5375,-19338;
-5283,-19366;
-5170,-19402;
-5081,-19422;
-4971,-19452;
-4891,-19474;
-4829,-19489;
-4749,-19505;
-4668,-19531;
-4624,-19535;
-4517,-19564;
-4429,-19584;
-4314,-19612;
-4154,-19653;
-4138,-19656;
-4015,-19685;
-4027,-19684;
```

```
 97   -3962,-19687;
 98   -3753,-19742;
 99   -3841,-19713;
100   -3791,-19727;
101   ];
102
103   values3=[
104   2079,-19994;
105   2304,-19964;
106   2400,-19952;
107   2273,-19952;
108   2507,-19920;
109   2565,-19919;
110   2672,-19899;
111   2824,-19874;
112   2968,-19859;
113   2702,-19889;
114   2789,-19886;
115   2892,-19861;
116   2889,-19860;
117   2643,-19891;
118   2744,-19879;
119   2765,-19886;
120   2894,-19842;
121   3080,-19828;
122   3126,-19824;
123   3162,-19805;
124   3280,-19796;
125   3276,-19790;
126   3339,-19784;
127   3504,-19763;
128   3549,-19742;
129   3487,-19756;
130   3764,-19704;
131   3817,-19697;
132   3886,-19684;
133   4029,-19656;
134   4168,-19646;
135   4061,-19642;
136   4197,-19618;
137   4487,-19566;
138   4769,-19498;
139   4679,-19512;
140   4816,-19489;
141   4782,-19500;
142   4694,-19513;
143   4788,-19498;
144   4799,-19481;
145   ]
146
147   values4=[
148   9972,-17372;
149   10098,-17312;
150   10155,-17274;
151   10188,-17251;
152   10295,-17187;
153   10496,-17064;
154   10530,-17042;
155   10597,-17010;
156   10516,-17049;
157   10583,-17010;
158   10553,-17022;
159   10783,-16877;
160   10959,-16753;
161   10951,-16761;
162   11080,-16669;
163   11175,-16610;
164   11299,-16530;
165   11323,-16517;
```

```
166  11561,-16341;
167  11434,-16429;
168  11834,-16183;
169  11412,-16436;
170  11441,-16428;
171  11756,-16212;
172  11764,-16197;
173  11938,-16068;
174  11834,-16141;
175  11856,-16112;
176  12065,-15971;
177  12076,-15951;
178  12104,-15932;
179  12299,-15777;
180  12469,-15634;
181  12433,-15652;
182  12572,-15546;
183  12530,-15583;
184  12571,-15541;
185  12616,-15509;
186  12702,-15439;
187  12732,-15420;
188  12836,-15339;
189  ]
190
191
192  values5=[
193  13883,-14396;
194  14021,-14287;
195  13960,-14346;
196  14000,-14294;
197  14081,-14226;
198  14154,-14147;
199  14190,-14108;
200  14246,-14057;
201  14246,-14052;
202  14262,-13976;
203  14458,-13909;
204  14135,-14155;
205  14313,-13984;
206  14203,-14086;
207  14415,-13879;
208  14509,-13785;
209  14494,-13802;
210  14466,-13808;
211  14552,-13740;
212  14718,-13548;
213  14808,-13477;
214  14712,-13560;
215  14887,-13370;
216  14948,-13300;
217  15000,-13240;
218  15255,-12922;
219  15362,-12811;
220  15425,-12716;
221  15312,-12855;
222  15243,-12949;
223  15455,-12670;
224  15563,-12541;
225  15653,-12433;
226  15632,-12472;
227  15588,-12536;
228  15816,-12214;
229  15839,-12195;
230  16001,-11985;
231  16019,-11936;
232  16016,-11924;
233  15934,-12045;
234  ]
```

```
235
236
237
238 load('calibrationValues.mat');
239
240 for pointer=1:5
241 valuesEIS=eval(sprintf('values%d',pointer));
242
243 [l,c]=size(valuesEIS);
244
245 for i = 1:l
246     %16bit decimal value of real part
247 %     unsignedReal=swapbytes(typecast(uint8(valuesEIS(i,1:2)'),'uint16'));
248 %     real(i)=double(typecast( uint16(unsignedReal), 'int16'));
249     real(i)=valuesEIS(i,1);
250     %16bit decimal value of imag part
251 %     unsignedImag=swapbytes(typecast(uint8(valuesEIS(i,3:4)'),'uint16'));
252 %     imag(i)=double(typecast( uint16(unsignedImag), 'int16'));
253     imag(i)=valuesEIS(i,2);
254 end
255
256 %calculate magnitude
257 magnitudePoints = sqrt(real.*real+imag.*imag);
258
259 if(pointer==1)
260     allMagnitudes=magnitudePoints;
261 else
262     allMagnitudes=horzcat(allMagnitudes,magnitudePoints);
263 end
264
265 end
266 allImpedances=1./(calibration(2,:).*allMagnitudes(1,:));
267 %
268 %
269 % GF=arrayGF(pointer);
270 %
271 %
272 %
273 %
274 % impedance=1./(GF*magnitudePoints);
275 % % figure
276 % % subplot(2,1,1)
277 % % plot(frequencyPoints,impedance)
278 % % hold on
279 % % plot(frequencyPoints,repmat(impedanceCalib,1,l))
280 % % hold off
281 % % title( ['Frequency sweep between ' num2str(startFreqs(pointer)) ' and ',num2str(stopFreqs(pointer)),'
        KHz, calibrated at ',num2str(calibFreq(pointer)),' KHz'] )
282 %
283 %calculate errors
284 error=allImpedances-usedImpedance;
285 percentualError=(error/usedImpedance)*100;
286 %
287 % % subplot(2,1,2)
288 % % bar(frequencyPoints,percentualError)
289 % % xlabel('Frequency')
290 % % ylabel('Percentual Error')
291 % % title(['Impedance Measurement Error for calibration at ',num2str(calibFreq(pointer)),' KHz'])
292 figure
293 plot(calibration(1,:),allImpedances(1,:))
294 hold on
295 plot(calibration(1,:),repmat(usedImpedance,1,length(allImpedances)))
296 xlabel('Frequency [Hz]')
297 ylabel('Value of Measured Impedance [Ohms]')
298
299 yyaxis right
300 plot(calibration(1,:),abs(percentualError))
301 ylabel('Error [%]')
302 ylim([0 100])
```

```
303  grid on
304
305  mean(percentualError)
306  % std(allImpedances)
```

Appendix M

System Hardware Upgraded Version

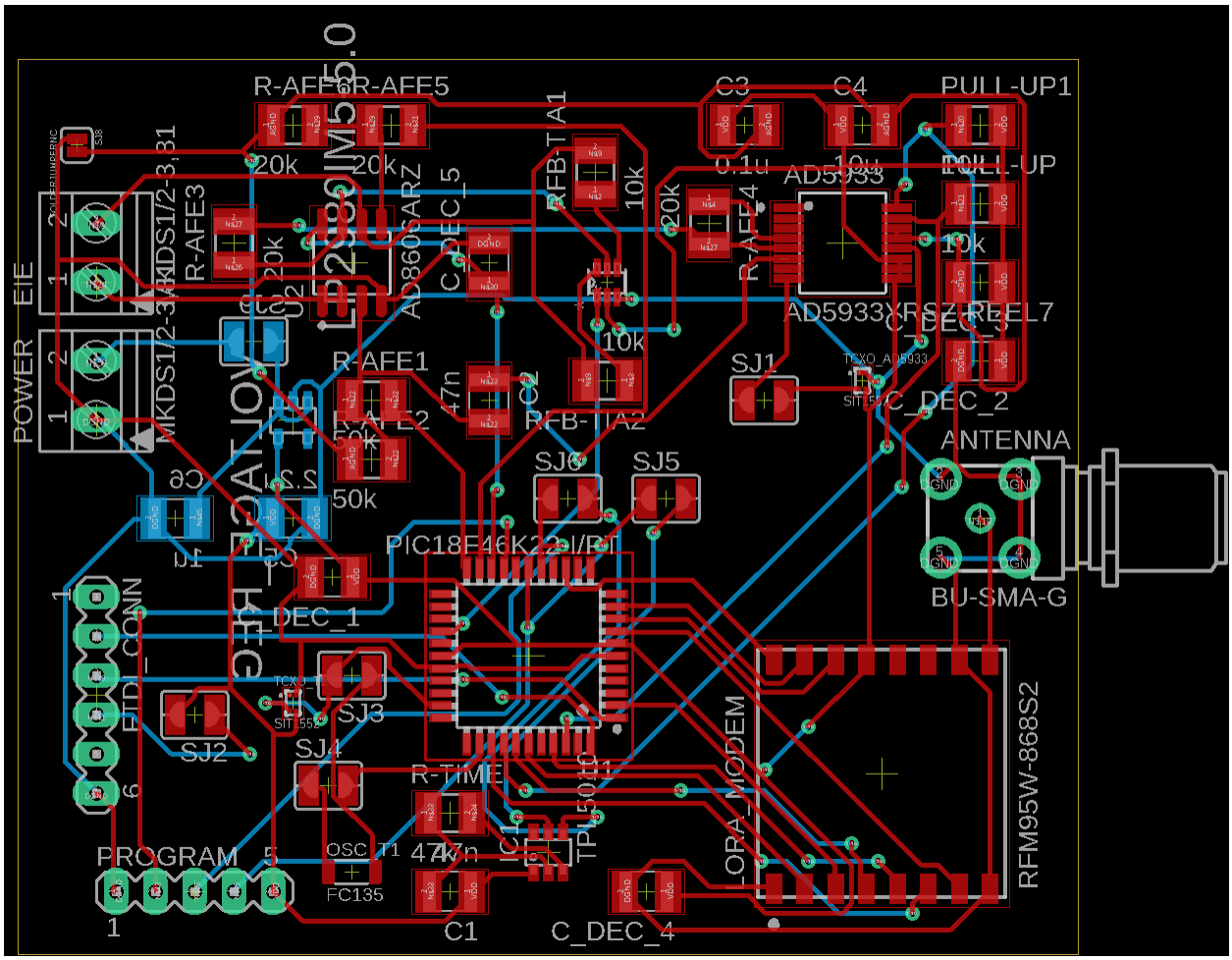

Figure M.1: New PCB design with all of the modules incorporated

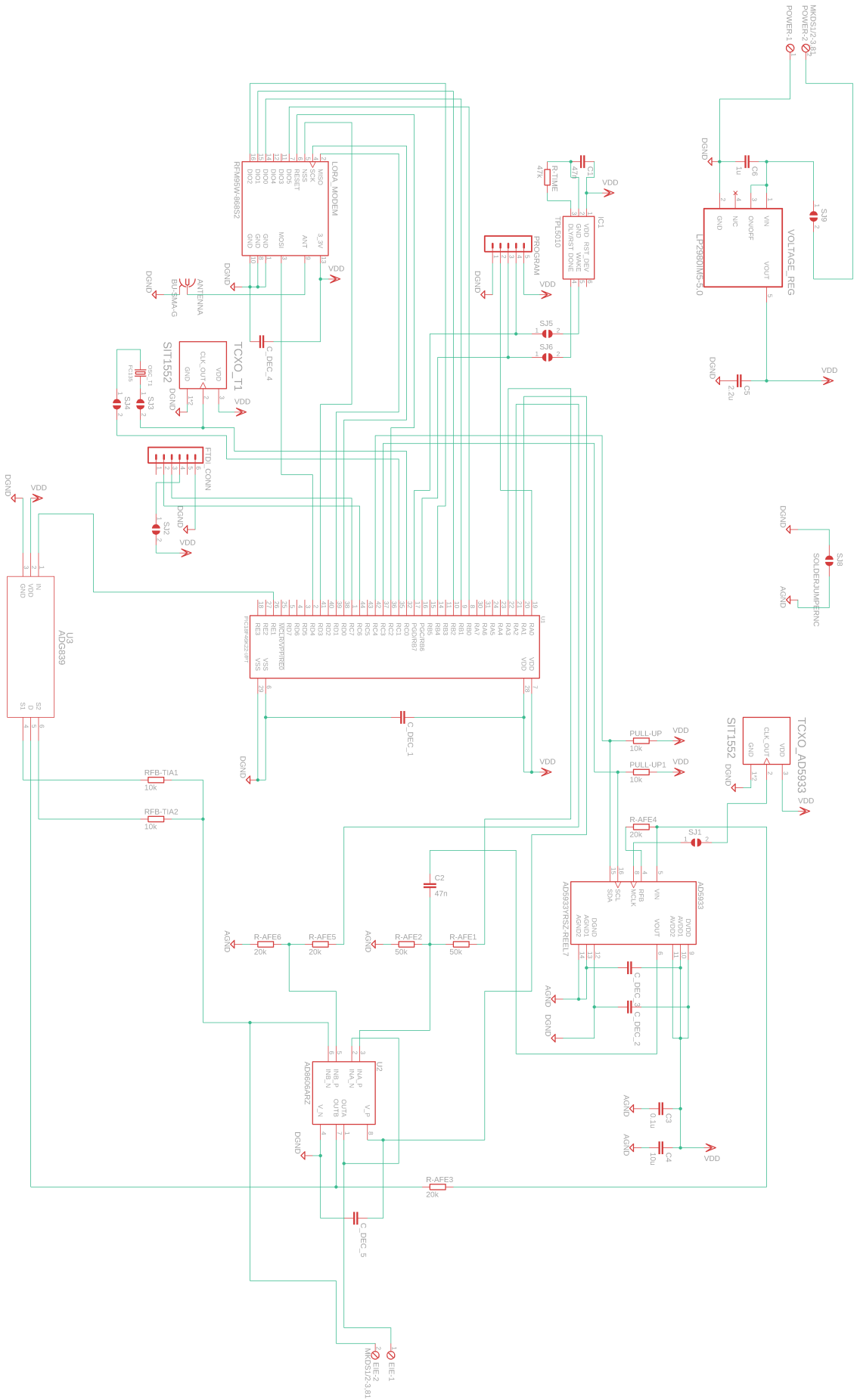

Figure M.2: Schematic of the new board revision