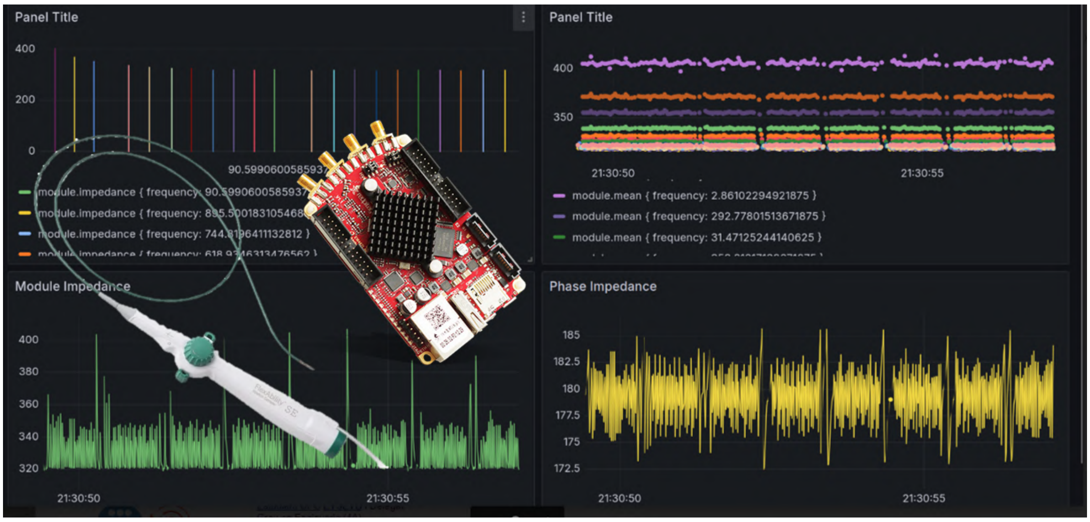
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

telecos
BCN

Application for control, visualization and analysis of Fast-EIS measurements for medical applications



Bachelor's **Degree Thesis**
Submitted to the Faculty at the
Escola Tècnica Superior
d'Enginyeria de Telecomunicació de Barcelona
de la Universitat Politècnica de Catalunya
by

Josep Mencion Seguranyes

In partial fulfillment
of the requirements for the
Degree in Electronic Engineering

Advisor: **Ramon Bragós Bardia**

Application for control, visualization and analysis of Fast-EIS measurements for medical applications

Abstract

This project involves developing an application to control a Fast Electrical Impedance Spectroscopy system, optimizing the high-speed capture, processing, and real-time visualization of impedance data from cardiac and pulmonary tissues. These measurements are taken using a specialized catheter, and the system enables direct measurements within the body, facilitating the differentiation of electrical properties between healthy and pathological tissues, such as tumors, based on their resistance and capacitance.

The integration of real-time signal processing and communications technologies in this system provides a tool for detailed analysis of tissue response at different frequencies and over time, essential for the precise identification of anomalies. Additionally, the data is stored in a database for subsequent analysis and extraction of conclusions. This approach aims to improve accuracy in the localization and diagnosis of internal lesions, utilizing principles of biomedical engineering, electronics, and telecommunications.

Aplicativo de control, visualización y análisis de medidas Fast-EIS para aplicaciones médicas

Resumen

Este proyecto consiste en el desarrollo de una aplicación para controlar un sistema de Espectroscopía de Impedancia Eléctrica Rápida, optimizando la captura, procesamiento y visualización en tiempo real de datos de impedancia de tejidos cardíacos y pulmonares. Estas medidas se realizan utilizando un catéter especializado, y el sistema permite medidas directas dentro del cuerpo, facilitando la diferenciación de propiedades eléctricas entre tejidos sanos y patológicos, como tumores, basándose en su resistencia y capacitancia.

La integración de tecnologías de procesamiento de señales en tiempo real y comunicaciones en este sistema proporciona una herramienta para el análisis detallado de la respuesta del tejido a diferentes frecuencias y en el tiempo, esencial para la identificación precisa de anomalías. Además, los datos se almacenan en una base de datos para su posterior análisis y extracción de conclusiones. Este enfoque tiene como objetivo mejorar la precisión en la localización y diagnóstico de lesiones internas, utilizando principios de ingeniería biomédica, electrónica y telecomunicaciones.

Aplicatiu de control, visualització i anàlisi de mesures Fast-EIS per a aplicacions mèdiques

Resum

Aquest projecte consisteix en el desenvolupament d'una aplicació per controlar un sistema d'Espectroscòpia d'Impedància Elèctrica Ràpida, optimitzant la captura, processament i visualització en temps real de dades d'impedància de teixits cardíacs i pulmonars. Aquestes mesures es realitzen utilitzant un catèter especialitzat, i el sistema permet mesures directes dins del cos, facilitant la diferenciació de propietats elèctriques entre teixits sans i patològics, com tumors, basant-se en la seva resistència i capacitança.

La integració de tecnologies de processament de senyals en temps real i comunicacions en aquest sistema proporciona una eina per a l'anàlisi detallada de la resposta del teixit a diferents freqüències i en el temps, essencial per a la identificació precisa d'anomalies. A més, les dades s'emmagatzemen en una base de dades per a la seva posterior anàlisi i extracció de conclusions. Aquest enfocament té com a objectiu millorar la precisió en la localització i diagnòstic de lesions internes, utilitzant principis d'enginyeria biomèdica, electrònica i telecomunicacions.

Logic will get you from A to Z. Imagination will get you everywhere.
Albert Einstein

The greatest glory lies not in never falling, but in rising every time we fall.
Nelson Mandela

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my professor and advisor, **Ramon Bragós Bardia**, for providing me with the opportunity to work on this project and for dedicating countless hours of guidance and support. Dr. Bragós holds a Ph.D. in Telecommunication Engineering (1997) from UPC and is a professor in the Department of Electronic Engineering at UPC, where he is a member of the Electronic and Biomedical Instrumentation research group. Since 2009, he has served as the Deputy Director for Educational Innovation at the School of Telecommunications Engineering (ETSETB), where he is responsible for implementing the CDIO methodology in the curriculum and coordinating project-based courses.

I would also like to extend my sincere appreciation to **Sergi Valverde Valverde**, CEO of TensorMedical, a startup based in Girona specializing in medical image analysis with artificial intelligence for early detection of multiple sclerosis. Their insights and support have been invaluable to this project.

Special thanks to **Miguel Ángel Cámara**, Biomedical Engineer, *RD* Data Scientist and AI Engineer, for providing me with valuable guidance on how to approach my Bachelor's thesis and for introducing me to the possibility of incorporating data analysis models.

I am grateful to **Xavier Rosell Ferrer**, currently a Professor at the Department of Electronic Engineering and the Biomedical Research Center of UPC, where he leads the Biomedical Instrumentation group, and **Juan Ramos Castro** for their guidance and mentorship throughout this journey. Professor Ramos Castro has been a lecturer at UPC since 1998, teaching courses on electronic and biomedical instrumentation. He is currently co-director of the Catalonia Institute for Space Studies. Their expertise and encouragement have been instrumental in shaping this thesis.

Finally, I would like to thank my family for their unwavering support, especially my parents, my brother, and my grandparents.

Contents

List of Figures

List of Tables

Acronyms

ADCs	Analog-to-Digital Converters
DACs	Digital-to-Analog Converters
EIS	Electrical Impedance Spectroscopy
FEIS	Fast Electrical Impedance Spectroscopy
FFT	Fast Fourier Transform
FPGAs	Field-Programmable Gate Arrays
GUI	Graphical User Interface
HSP	Hospital de la Santa Creu i Sant Pau de Barcelona
IEB	Electronic and Biomedical Instrumentation Group
MSps	Mega Samples per Second
RC	Resistor-Capacitor
RMS	Root Mean Square
UPC	Universitat Politècnica de Catalunya
SDG	Sustainable Development Goals
BT	Bachelor's Thesis

Chapter 1

Introduction

The aim of this project is to develop an application for controlling a Fast Electrical Impedance Spectroscopy system, optimizing the capture, processing, and visualization of impedance data from cardiac and pulmonary tissues. These measurements are taken using a specialized catheter, allowing for direct in-body measurements and facilitating the differentiation of electrical properties between healthy and pathological tissues, such as tumors, based on their resistance and capacitance. The integration of real-time signal processing and communication technologies into this system would provide a tool for detailed analysis of tissue response at different frequencies, essential for precise identification of anomalies. This approach aims to enhance accuracy in the localization and diagnosis of internal lesions, employing principles of biomedical engineering, electronics, and telecommunications.

1.1 Motivation

The primary motivation for this project comes from the Electronic and Biomedical Instrumentation Group (IEB) at the Universitat Politècnica de Catalunya (UPC) aiming to enhance an existing electrical impedance spectroscopy (EIS) system based on a FPGA board. The current system was equipped with a rudimentary application that displays some sample points in real-time and saves data in a binary format known as raw data. While functional, this application lacks the capability for real-time visualization and comprehensive data analysis post-measurement. The goal is to optimize the data analysis of impedance data after collection in order to extract estimations about the tissue health.

In addition to that, there is a growing need within the biomedical engineering community for advanced tools for tissue characterization. Electrical impedance spectroscopy is a powerful technique for distinguishing between healthy and pathological tissues based on their passive electrical properties. By improving the existing EIS system, would facilitate research and clinical applications in areas such as pathological cells detection which can cause arrythmias or other problems.

1.2 Objectives

The objectives of this project are designed to address the challenges in the development of a dashboard for tissue characterization using electrical impedance. Leveraging expertise from multiple disciplines including biomedical engineering, signal processing, and machine learning, the following objectives aim to provide a comprehensive solution for optimizing impedance data capture, processing, and analysis.

1.2.1 Development of the Dashboard Interface

The first objective is to design and develop an intuitive and dashboard interface. This dashboard will allow medical users to interact with impedance data, modify parameters, and visualize results in a clear and understandable manner. The aim is to create a tool that is accessible to clinicians and researchers alike.

1.2.2 Integration of Real-Time Data Processing and Analysis

The second objective is to integrate real-time data processing and analysis capabilities into the dashboard. This involves being able to see the data in real time while making the in-vivo measurement. This will allow doctors to make informed decisions quickly and accurately.

1.2.3 Optimization for Clinical Use and Post-Data Analyzing

The third objective is to optimize the dashboard for clinical use, ensuring that it meets the requirements and standards of medical practice. This implies that it complies with the following points:

- **Impedance Magnitude as a function of Time** ($|Z(t)|$)
- **Impedance Phase as a function of Time** ($\theta(t)$)
- **Impedance Magnitude as a function of Frequency** ($|Z(f)|$): this graph has to be related to the Impedance vs. When someone selects a temporal measure, the frequency graph has to be generated using the nearest 26 frequencies.
- **Impedance Phase as a function of Frequency** ($\theta(f)$): this graph has to act as the last one.
- Storage of a predetermined segment of the acquired data after catching a specific control.

1.2.4 Validation and Testing

The fourth objective is to validate the performance of the impedance characterization system. This involves testing each one of the steps, including patient creation, measure taking, database data uploading, visualizing and analyzing data.

1.3 Work Plan and Gantt Chart Analysis

The Gantt chart clearly illustrates the timeline for each work package (WP), showing the start and end dates as well as the duration of each task. The tasks are spread over the timeline from Week 1 (Februrary 12, 2024) to Week 18 (June 17, 2024), ensuring a structured and phased approach to project completion. This structured timeline helps in tracking progress, identifying potential delays, and ensuring that each task is completed within the allocated timeframe. The overlap of certain tasks indicates parallel activities, which optimize resource utilization and project efficiency.

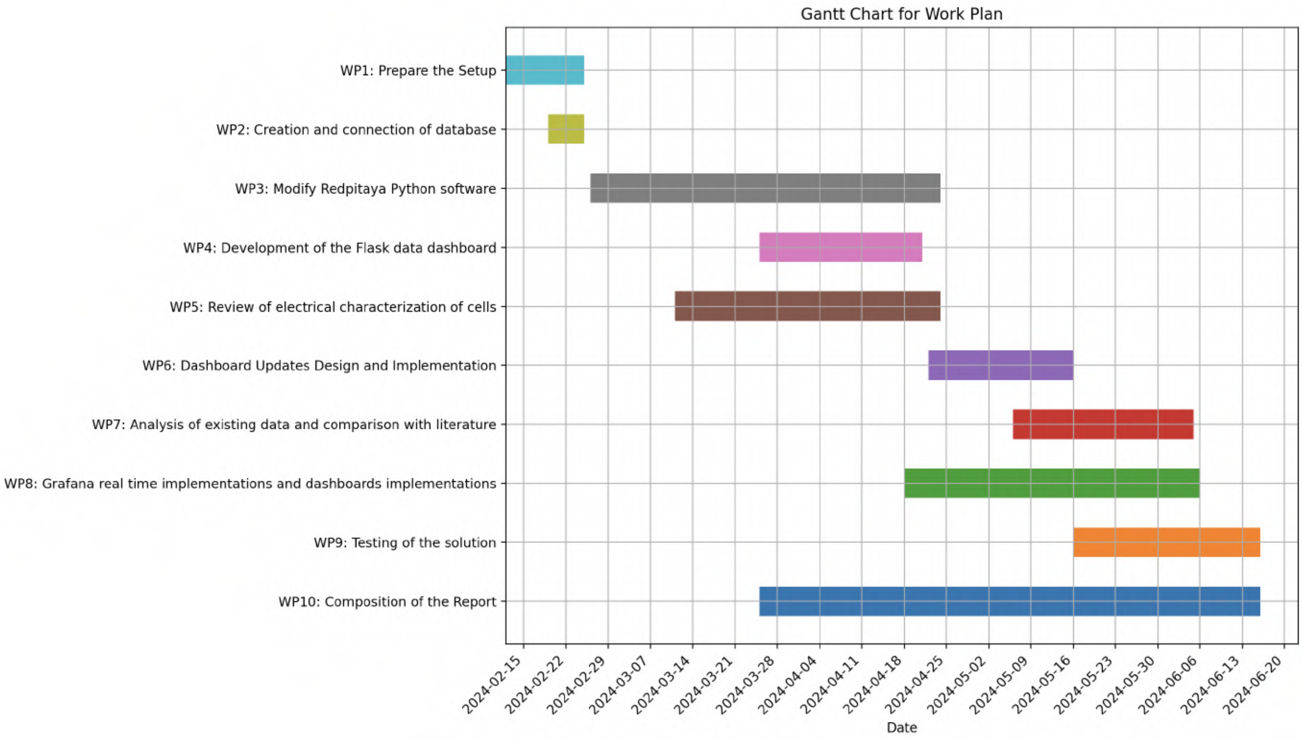


Figure 1.1: Gantt Diagram

Source: Own

WP1: Prepare the Setup

This work package consists of preparing the computer in which all tests will run. It involves installing all the dependencies and necessary software.

- **Internal Task 1:** Set up Ubuntu.
- **Internal Task 2:** Install InfluxDB, a software which allows us to create databases.
- **Internal Task 3:** Install Grafana, an advanced software which allows us to visualize data in many ways.
- **Internal Task 4:** Set up Redpitaya and the connection via Ethernet with the computer.

WP2: Creation and connection of database

To analyze the data in the future, we need to save all the measurements in one place and make them easily accessible. This is achieved by creating a database based on InfluxDB.

- **Internal Task 1:** Create an InfluxDB database.
- **Internal Task 2:** Design a function to upload the data in real time to the database.
- **Internal Task 3:** Modify the program to achieve a frequency of 15.625 MHz in submitting data to the database.
- **Internal Task 4:** Modify the Python application software to send data in real time to the database.

WP3: Modify Redpitaya Python software

We need to receive data from the catheter at the sample frequency, which is 15.625 MSamples/s. The challenge is ensuring that we do not lose data and that all data is stored in an easily accessible and processable format.

WP4: Development of the Flask data dashboard

Medical professionals need an easy way to understand the data. This involves designing a dashboard that receives real-time data from the Redpitaya device through Ethernet and displays it in various formats.

- **Internal Task 1:** Choose the basis for the application, which will be a web app.
- **Internal Task 2:** Design and create the backend code.
- **Internal Task 3:** Design and create the frontend code.
- **Internal Task 4:** Connect the dashboard to the Python application that takes the data in real time.
- **Internal Task 5:** Connect the dashboard to Grafana to show graphs.
- **Internal Task 6:** Create queries related to the needed graphs for the medical sector.
- **Internal Task 7:** Test if the whole system is working properly.

WP5: Review of electrical characterization of cells

This involves a literature review focused on the electrical characterization of cells, particularly tumor tissues and cancer, to compile a State-of-the-Art document guiding the project's scientific approach.

- **Internal Task 1:** Identify and review relevant articles and studies.

- **Internal Task 2:** Compile findings and methodologies.
- **Internal Task 3:** Create the State-of-the-Art document.
- **Internal Task 4:** Interview surgeons, interventional radiologists, and cardiologists to understand the most important data to display on the dashboard for medical conclusions.

WP6: Dashboard Updates Design and Implementation

This task involves designing and implementing updates to the dashboard to display real-time data from the Red Pitaya device via Ethernet, making it easy for medical professionals to access data in various formats.

- **Internal Task 1:** Document conditions of existing data collection.
- **Internal Task 2:** Create a roadmap of views needed and how to implement them.
- **Internal Task 3:** Implement the coding for these updates.

WP7: Analysis of existing data and comparison with literature

Analyze existing experimental data, document the conditions under which the data were collected, and compare these findings with the literature to define protocols for future data collection.

WP8: Grafana real time implementations and dashboards implementations

This involves implementing real-time Grafana solutions and configuring dynamic dashboards, including setting up Grafana instances, designing dashboards, integrating data sources, and optimizing performance.

- **Internal Task 1:** Initial Grafana Configuration.
- **Internal Task 2:** Design and Creation of Dashboards.
- **Internal Task 3:** Integration of Data Sources.
- **Internal Task 4:** Real Time modifications.

WP9: Testing of the solution

This work package involves testing the entire solution to ensure that all components work together seamlessly and meet the project requirements.

WP10: Composition of the Report

This task involves compiling the project documentation, including all research, development, and implementation details, into a comprehensive project report.

Chapter 2

Bioimpedance background theory

Bioimpedance refers to the electrical impedance of biological materials and its passive electrical properties.

In biological materials, ions dissolved in the fluids within and around cells serve as the carriers of electrical charge. To allow the flow of current, these ions must move within the tissue. These electrolyte solutions exhibit conductive behavior, adhering to Ohm's law. In addition to that, biological cell membranes act as insulators, known as dielectrics, possessing bound charges. They exhibit both high resistance to current flow and high capacitance.

When biological tissue is stimulated by an alternating signal, the result is a behavior that **varies with the frequency of the signal**, which provides **valuable insights into the physiological conditions and structure of the tissue**.

2.1 Electrical Properties of Tissue

Tissues consist of extracellular fluid and cells. Extracellular fluid includes all body fluids outside of cells, comprising mostly water and ions.

Cells are composed of a lipid bilayer plasma membrane. Enclosed within this membrane, cells house intracellular fluid, or cytosol, along with organelles and the cell's nucleus.

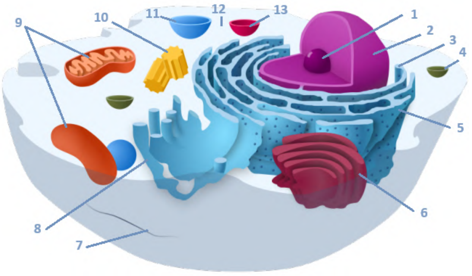


Figure 2.1: Living cell: (1) nucleolus (2) nucleus (3) ribosome (4) vesicle (5) rough endoplasmic reticulum (6) Golgi apparatus (7) Cytoskeleton (8) smooth endoplasmic reticulum (9) mitochondria (10) centrioles (11) vacuole (12) cytoplasm (13) lysosome

Source: ***Wikipedia - Animal Cell*** [1]

2.2 Tissue as Dispersive Medium

The electrical impedance of animal tissues presents several deviations along the frequency, known as dispersions.

- **α-dispersion:**

  Occurs within the low-frequency range, from below 1 Hz to 1 kHz. Although we know some factors driving this frequency dependency, three main causes are: the endoplasmic reticulum's influence, channel proteins in the plasma membrane, and relaxation of counter-ions on the charged cellular surface.

- **β-dispersion:**

  The range of β-dispersion [2] is from 1 kHz to 1 MHz. This phenomenon comes from the low conductivity and capacitive properties inherent in the cell membrane and other internal membrane structures, as well as their interactions with extracellular and intracellular electrolytes.

- **γ-dispersion:**

  Comes from the abundant presence of water in cells and tissues. Its relaxation frequency hovers around 20 GHz, akin to that of regular water, with its dispersion range spanning from hundreds of MHz to several GHz.

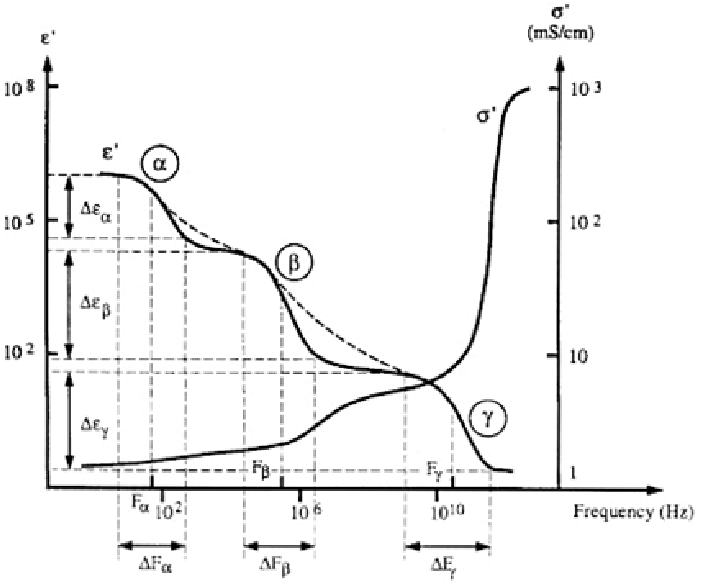


Figure 2.2: Frequency Tissue Dispersion Graphic

Source: *J. R. Bourne, Bioelectrical impedance techniques in medicine [3]*

The dispersion that we use in bioimpedance measurements is β-dispersion. This is because **β-dispersion, occurring between 1 kHz and 1 MHz, is significantly influenced by**

the cell membrane's conductive and capacitive properties, along with interactions between extracellular and intracellular electrolytes.

The β-dispersion range is particularly relevant for bioimpedance because it provides crucial information about the **cellular structure and composition**. This frequency range is **sensitive to the state of the cell membranes, which is critical in distinguishing between different types of tissues, including normal and pathological tissues such as tumors**. In the case of tumors, the structure and behavior of cell membranes can be markedly different from those of normal cells. Tumor cells often exhibit altered membrane properties, including differences in ion channel activity and membrane capacitance, which can be detected through changes in bioimpedance measurements within the β-dispersion range. These differences are used to identify and characterize tissue types, providing a non-invasive diagnostic tool in medical applications.

2.3 Bioimpedance Tissue Characterization

Bioimpedance analysis has emerged as a critical technique in medical diagnostics, particularly useful for differentiating pathological tissues in organs such as the lungs and heart. This document delves into the nuances of bioimpedance measurements and how they assist in the detection and treatment of diseases based on changes in electrical properties of tissues.

2.4 Lung Tissue Analysis

2.4.1 Bioimpedance in Normal and Tumorous Lung Tissue

Properties of Normal Lung Tissue

The lung's primary function is gas exchange, and it is structurally characterized by a network of air sacs and bronchial tubes, surrounded by soft tissue. This structural composition results in a high bioimpedance due to the air, which is a poor conductor of electricity.

- **Air Content**: High bioimpedance due to air-filled spaces.
- **Cellular Structure**: Spread-out cells surrounded by air, contributing to impedance.

Tumorous Tissue Impedance

Unlike healthy lung tissue, tumorous tissue shows a packed arrangement of cells, significantly reducing the air content and thereby decreasing impedance. This provides a stark contrast in bioimpedance measurements between healthy and cancerous tissues.

- **Cell Density**: Increased due to tumor cells.
- **Electrical Conductivity**: Lower impedance due to reduced air spaces.

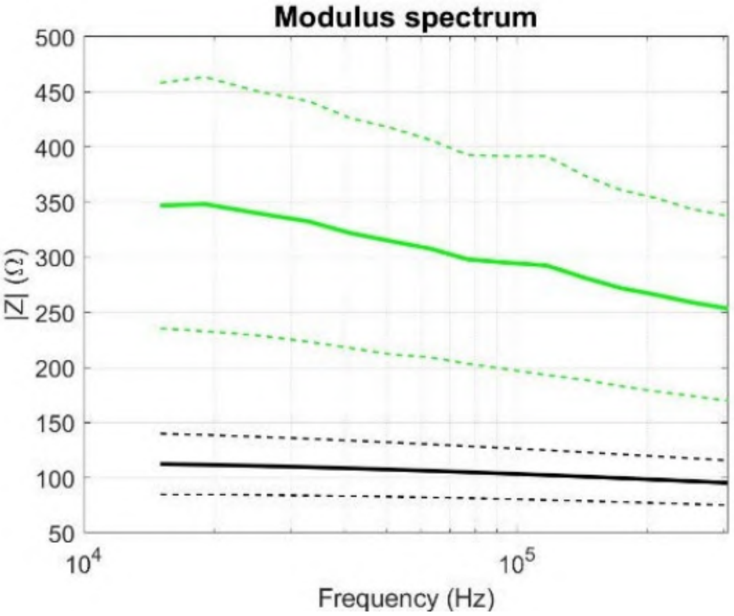


Figure 2.3: Modulus Spectrum
Source: Georgina Company Se et al, 2022 (green is healthy lung, black is tumor) [4]

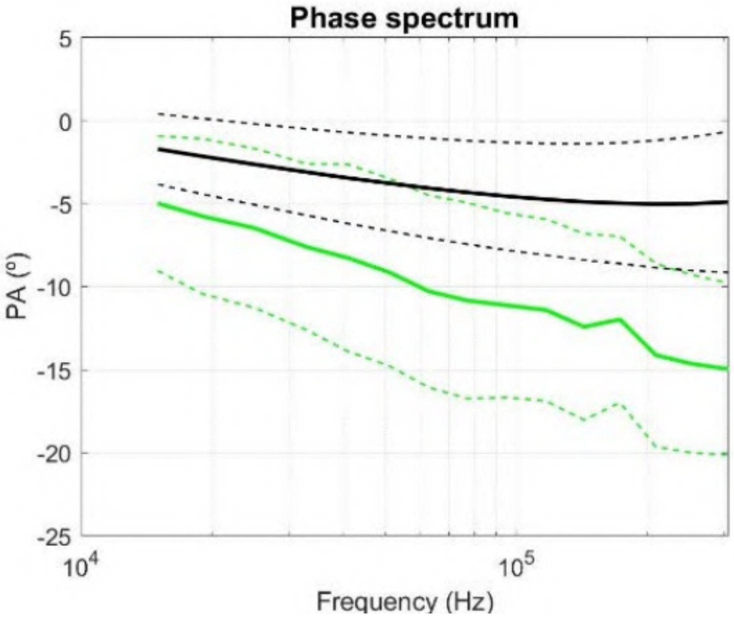


Figure 2.4: Phase Spectrum
Source: Georgina Company Se et al, 2022 (green is healthy lung, black is tumor) [4]

2.4.2 Experimental Measurement Techniques

Measurement of lung tissue bioimpedance involves experimental setups where electrodes are placed strategically to capture the impedance variations between healthy and tumorous tissues.

- **Threshold Definition**: Establishing a bioimpedance threshold to distinguish between healthy and pathological tissues.
- **Geometry Consideration**: Adjustments based on individual anatomical differences, particularly in the bronchial network.

2.5 Heart Tissue Analysis

2.5.1 Bioimpedance in Healthy Heart Tissue

Structural Composition

The heart's muscular walls, particularly those of the ventricles, are thick and robust, designed to withstand and generate high pressure to pump blood effectively. This structural feature significantly influences the bioimpedance measurements.

- **Muscular Walls**: Thick ventricular walls contribute to higher impedance.
- **Coronary Arteries**: Presence of arteries on the surface that supply blood to the heart muscle itself.

2.5.2 Pathological Changes Due to Coronary Obstruction

Infarction and Bioimpedance

A blocked coronary artery can lead to myocardial infarction, where the affected region undergoes significant cellular and structural changes, detectable by shifts in bioimpedance.

- **Immediate Response**: Cells initially increase their fluid content to survive, **lowering the extracellular fluid** and then raising the impedance.
- **Long-term Changes**: Necrosis leads to a thinning of the tissue and replacement by non-cellular collagen, reducing impedance.

2.5.3 Therapeutic Monitoring via Bioimpedance

Catheter-based impedance measurements are crucial in managing heart pathologies, especially to monitor and treat areas affected by infarction.

- **Real-Time Monitoring**: Using catheters to deliver and measure impedance at various frequencies to detect and localize necrotic tissue effectively.

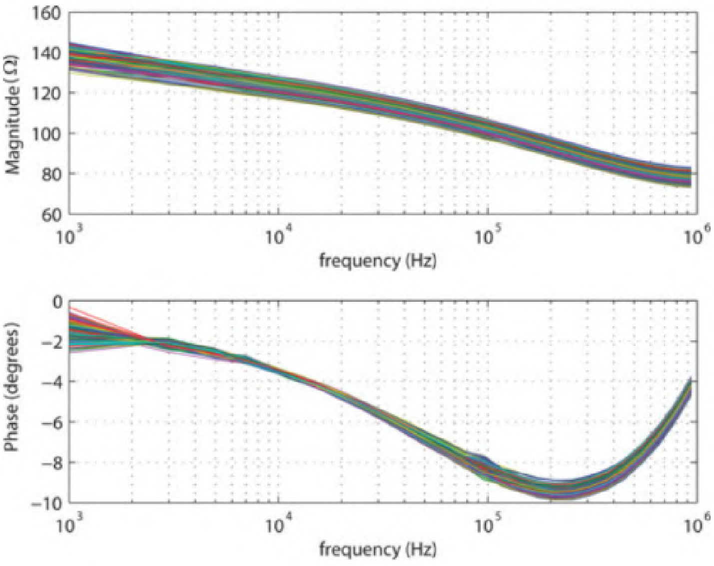


Figure 2.5: Changes in the healthy myocardium impedance spectrum magnitude (top) and phase (bottom) within the cardiac cycle during 10 s (22 spectra/s, four multisine periods). Source: Benjamin Sanchez et al, 2011 [5]

- **Ablation Therapy**: Targeted ablation of malfunctioning tissue monitored through changes in impedance, ensuring only pathological cells are destroyed.

The application of bioimpedance in medical diagnostics serves as a promising tool in electronic engineering for health applications, particularly in enhancing the accuracy and efficacy of disease detection and treatment in the lungs and heart. This advanced characterization technique provides engineers and healthcare professionals with critical insights into the electrical properties of tissues, enabling better patient outcomes.

The measurements of electrical impedance spectroscopy performed with a catheter in the miocardium or in the lungs, which are dynamic tissues, provide time series of impedance spectra.

While for research purposes, capturing and visualizing all the time-frequency evolution of the impedance parameters may make sense, in a clinical tool, the visualization should be simple and should allow identifying if the measurement is being taken in a right place and acquire a data segment of a few seconds and provide an estimation of the tissue state.

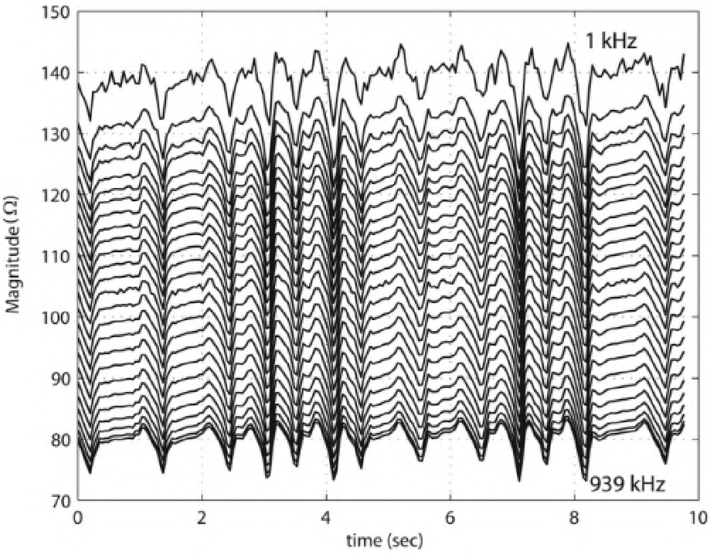


Figure 2.6: Myocardium tissue impedance magnitude (22 spectra/s, four multi-sine periods) G time course at the exciting frequencies [see (21)]. The excited frequencies appear in descending order from the lowest (1 kHz) to the highest (939 kHz)

Source: Benjamin Sanchez et al, 2011 [5]

Chapter 3

Problem Statement

3.1 Presentation of the Problem

In a previous project [6], a Fast Electrical Impedance Spectroscopy (FEIS) acquisition was developed. Given that the primary focus was on hardware development, particularly the FPGA core, the software component was intentionally kept simple, just to display that the hardware part was woking.

This approach resulted in significant limitations, particularly in real-time data analysis and visualization, which did not allow managing data on the graphs. These constraints highlighted the need for an enhanced and more efficient solution.

3.2 Resolution of the Problem

The old desktop version of the Python program displays some samples in real-time, but we do not have the ability to rewind or zoom into a measurement once it has been made. We can only view and export in binary format. This requires complex data processing each time the data needs to be analyzed. In this file, the data are only saved in binary format.

3.3 Major Issues in the Old System

3.3.1 Data Storage and Management

The old system stored each sample separately in a `.bin` file format, leading to several issues:

- **Scattered Data Files:** Each sample was saved as an individual `.bin` file, resulting in numerous scattered files. This made it difficult to manage and organize the data efficiently.

- **Difficulty in Data Analysis:** Direct analysis of the `.bin` files was challenging, requiring additional steps to convert and process the data. This added complexity to the data analysis workflow.

3.3.2 Live Visualization Limitations

The live visualization provided by the old system was a static image with significant limitations:

- **No Zoom Functionality:** Users could not zoom in on specific sections of the data, limiting their ability to closely inspect detailed variations and trends.
- **Fixed Time Interval:** The time interval displayed in the live visualization was fixed, preventing users from adjusting the time window to focus on different periods of interest.

3.3.3 Lack of Data Organization and Patient Information Integration

There was no integrated system to register samples related to patients, which posed several problems:

- **Disorganized Data:** Without a centralized registry, correlating data samples with specific patients was cumbersome and prone to errors.
- **Lack of Contextual Information:** The absence of a system to relate data with patient information hindered comprehensive analysis and reduced the clinical relevance of the data.

3.3.4 Real-Time Processing and Conclusion Extraction

The old system lacked real-time processing capabilities, which had critical implications:

- **Delayed Data Processing:** Data processing was not performed in real-time, delaying the extraction of meaningful insights and conclusions.
- **Inability to Make Immediate Decisions:** The absence of real-time processing made it difficult for clinicians to make immediate decisions based on the latest data, potentially affecting patient outcomes.

3.4 Application Architecture

The architecture of the new application is designed to overcome the limitations of the old system, providing a more robust and efficient solution for FEIS data acquisition, processing, and visualization.

3.4.1 System Components

The new system architecture includes the following key components:

- **Data Acquisition Module:** Captures bioimpedance signal using an analog front-end composed by a current source, a differential amplifier and a current-to-voltage converter.

- **Processing Unit:** Utilizes Red Pitaya's FPGA and ADC capabilities for real-time data acquisition and processing.
- **Data Storage:** Stores data in a centralized database (InfluxDB) for efficient management and retrieval.
- **Visualization Interface:** A web-based interface (Grafana) for real-time data visualization, with features like zoom and adjustable time intervals.

3.4.2 Data Flow

The data flow in the new system is designed for efficiency and real-time processing:

- Data is acquired by the sensors and transmitted to the Red Pitaya.
- The Red Pitaya processes the data in real-time and sends it to InfluxDB.
- Grafana retrieves data from InfluxDB and displays it in real-time on the user interface.

3.5 Methodology

The methodology outlines the steps and tools used to develop the new system, ensuring a structured and efficient approach to problem-solving.

3.5.1 Work Segmentation

The work was segmented into several phases to ensure systematic progress:

- **Requirement Analysis:** Identifying the limitations of the old system and defining the requirements for the new system.
- **Design and Development:** Developing the new system architecture and components.
- **Testing and Validation:** Testing the system to ensure it meets the defined requirements.
- **Deployment:** Deploying the system for real-world use and monitoring its performance.

3.5.2 Use of Python

Python was chosen as the primary programming language due to its versatility and the availability of robust libraries for data processing and visualization.

3.5.3 App Framework

The application framework includes Flask for the backend and Grafana for the frontend visualization. Flask provides a flexible and efficient way to develop the server-side components, while Grafana offers powerful tools for real-time data visualization.

3.5.4 Tools and Resources

Several tools and resources were utilized in the development of the new system:

- **Red Pitaya:** Used for real-time data acquisition and processing.
- **InfluxDB:** A time-series database used for storing and managing data.
- **Grafana:** Used for creating interactive and real-time visualizations.
- **Python Libraries:** Libraries such as NumPy, Pandas, and Flask were used for data processing and application development.

This comprehensive methodology ensures that the new system addresses the limitations of the old system and provides a robust solution for FEIS.

Chapter 4

Solution

The solution to the current problem that has been planned is the development of a web dashboard developped to display and manage cardiac or pulmonary bioimpedance data obtained via the Red Pitaya FPGA-based system, which must comply various requirements.
The system has to be intuitive and easy to use for medical personnel, and will allow real-time monitoring of impedance measurements, both magnitude and phase, from cardiac or pulmonary samples.

4.1 Dashboard Requirements

1. **Patient Registration:** Before starting any sample, the system should allow entering the name, reference, and observations related to the patient. This step is crucial to properly record the measurement session and **associate the measurements with the clinique data of the patient and the hystopathological analysis of the tissues**.

2. **Sample Registration:** For each new sample taken from the patient, the system should automatically generate a unique ID. Additionally, a description of the sample and an optional field for additional observations should be entered.

3. **Start and Real-Time Visualization:** There should be a "Start" button which, when activated, begins data reading from the Red Pitaya and displays the impedance data in real time in two separate graphs: one for the impedance **magnitude** and another for the **phase**, with a refresh rate between 50 and 100 ms.

4. **Stopping the Measurement:** A "Stop" button will allow the data collection to end. When activated, the system should continue collecting data for an additional 5 seconds before completely stopping the measurement.

5. **Viewing Previous Measurements:** The system should include an index or list of all the measurements taken. Upon selecting a measurement, detailed graphs of the impedance, both magnitude and phase, organized by time should be displayed.

6. **Frequency Graphs:** When selecting a time interval on the magnitude or phase graphs, the system should display a bar chart of the corresponding frequencies, au-

tomatically selecting the 25 samples around the point of interest, starting from the first frequency of 0.95 kHz.

4.2 Technical Note

All the data which is transmitted from Red Pitaya, which is voltage and current, is initially stored in a .bin file with records of voltage and intensity, which is approximately 900 samples per second. Due to a real-time limitation, only 30 of these 9000 samples are processed to calculate the impedance, which is then uploaded to InfluxDB. This allows us to visualize using Grafana, in real time but decimated.

4.3 System Architecture

Here it can be observed a graph in which is described the components which are being taken into account in the dashboard system.

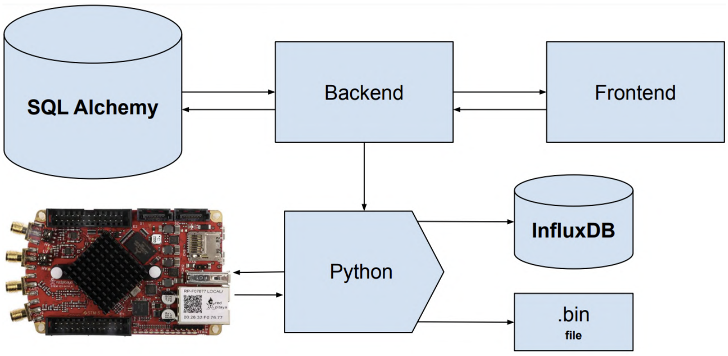


Figure 4.1: Schema of the dashboard system
Source: Own

4.4 Ploting impedance versus time information

Grafana is a very powerful tool which allows us graph information in real time. From the outset it is not designed to have a refresh interval less than 5 s, which is too high to graph bioimpedance measures in real time, which have a total duration of approximately 5 s. That's why it has been necessary to make a modification on the path code of Grafana to allow us making a 50 ms refresh, which is almost real time.

4.5 Application Operations and Electronic System Description

The application is designed to control a Fast Electrical Impedance Spectroscopy (FEIS) device, optimizing high-speed capture, processing, and real-time visualization of impedance

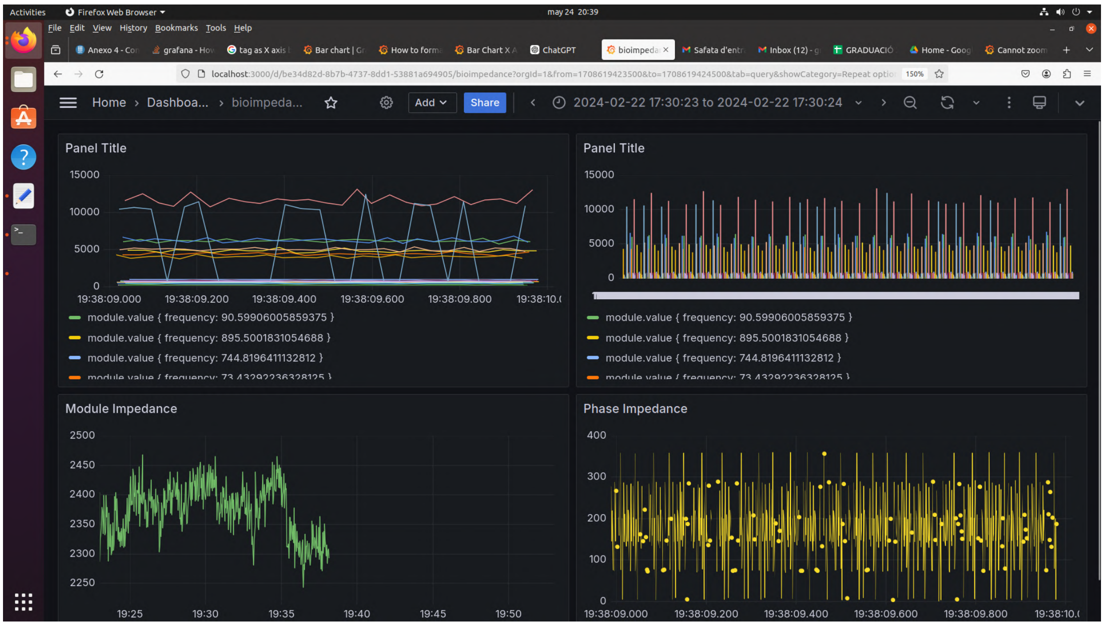


Figure 4.2: Schema of a dashboard in Grafana system

Source: Own

data from cardiac and pulmonary tissues. The data is acquired using a specialized catheter connected to a Red Pitaya, the electronic system used for data acquisition and processing.

The system facilitates the differentiation of electrical properties between healthy and pathological tissues, such as tumors, based on their resistance and capacitance in a sort of electronic biopsy. The integration of real-time signal processing and communication technologies in this system provides a tool for detailed analysis of tissue response at different frequencies and over time. The data is also stored in a database for subsequent analysis and extraction of conclusions.

4.5.1 Overview of the Electronic System

The electronic system forms the backbone of the FEIS device, enabling accurate data acquisition and processing. The primary component of this system is the Red Pitaya platform, a versatile and powerful tool designed for high-speed data acquisition and real-time signal processing.

Red Pitaya Platform

The Red Pitaya platform is equipped with analog-to-digital converters (ADCs) and digital-to-analog converters (DACs), which are crucial for capturing the impedance signals from the specialized catheter. The key features of the Red Pitaya that make it suitable for this application include:

- **High-Speed ADCs:** The ADCs on the Red Pitaya have a sampling rate of up to 125 MSps, allowing for precise capture of high-frequency signals.

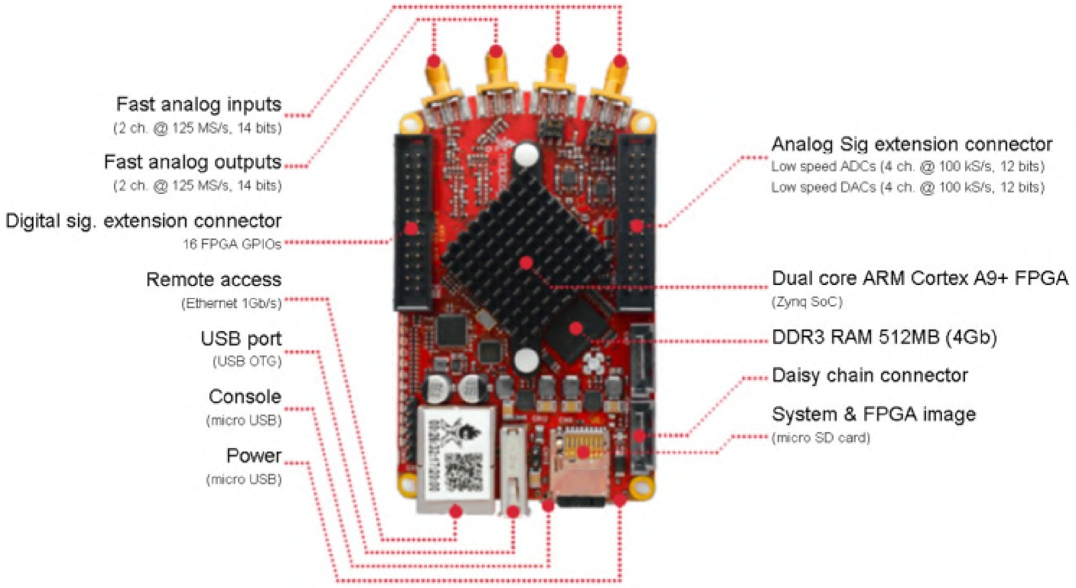


Figure 4.3: Red Pitaya

Source: ***FPGA Red***

- **Flexible DACs:** The DACs provide the necessary output signals for the excitation of tissues during impedance measurements.
- **FPGA Integration:** The Field-Programmable Gate Array (FPGA) on the Red Pitaya allows for customizable and high-speed processing of the acquired data.
- **Connectivity:** The platform supports various communication protocols, including Ethernet, which is used for data transfer to the main application for further processing.

4.6 Data Acquisition using Ablation Catheter

The data acquisition process is a critical component of the application, involving the capture of impedance signals from pulmonary tissues using an ablation catheter connected to the Red Pitaya platform.

4.6.1 Ablation Catheter: Design and Functionality

The specialized catheter used in this system is designed to perform cardiac ablation and is approved for human use. It is equipped with multiple electrodes to ensure accurate and localized impedance measurements. The key aspects of the catheter include:

- **Multi-electrode Configuration:** The catheter incorporates a multi-electrode array, enabling detailed spatial resolution of impedance measurements. Each electrode pair can capture localized impedance data, providing a comprehensive map of tissue properties.

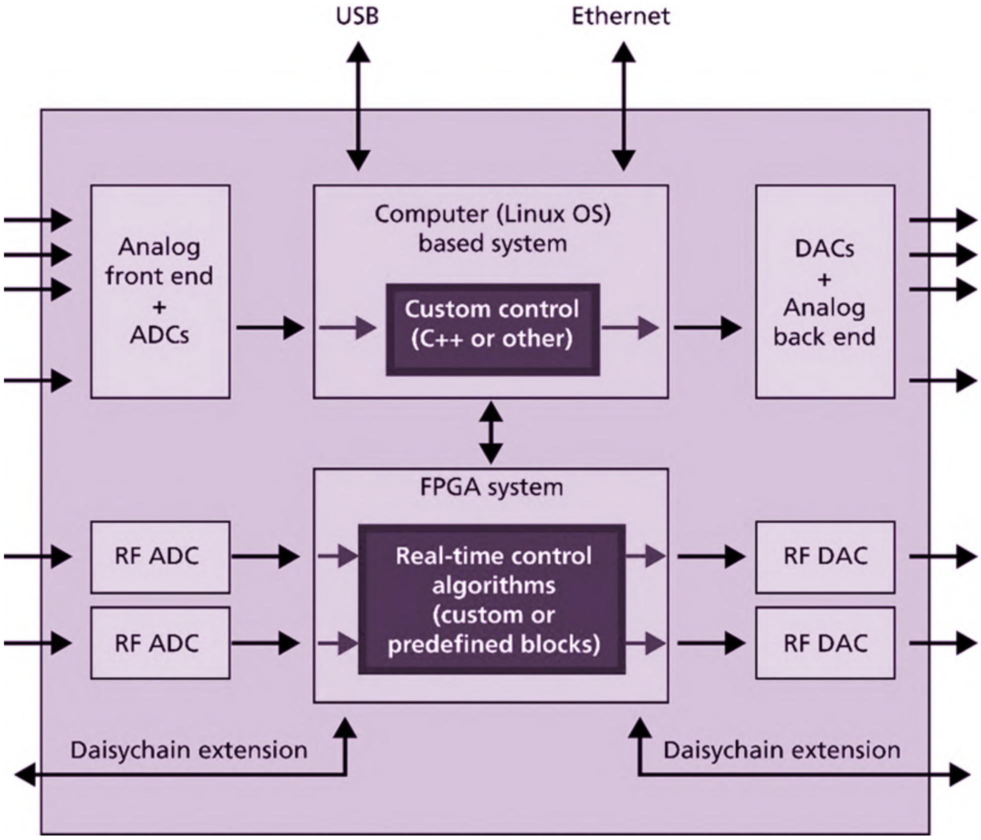


Figure 4.4: Red Pitaya Diagram

Source: ***FPGA Red***

Figure 4.5: Ablation Cathter

Source: ***Abbot - Cardiovascular [7]***

In some configurations, the 4 electrodes at the top are used to perform the impedance measurements. In some others, only the tip electrode and 2 external electrodes are used.

- **Biocompatible Materials:** The catheter is constructed using biocompatible materials to ensure safety and compatibility with human tissues. This minimizes the risk of adverse reactions during medical procedures.
- **Flexible and Durable:** The catheter is designed to be flexible, allowing it to navigate through the pulmonary pathways with ease, although is a single use device.

Its functionality is described below.

- **Signal Capture:** The electrodes capture the voltage signals from the tissue. These signals represent the electrical response of the tissue to an applied current, varying with tissue composition and health.
- **Signal Conditioning:** Integrated signal conditioning circuits in the front-end ensure that the captured signals are within the optimal range for the Red Pitaya's ADC inputs. This includes amplification and filtering to enhance signal quality.

System Integration

The integration of the specialized catheter with the Red Pitaya platform is critical for the seamless operation of the FEIS device. The catheter is connected to the front-end, which is connected to the Red Pitaya, which captures the impedance signals and processes them in real-time. The processed data is then transmitted to the main application via Ethernet for visualization and further analysis.

Component	Specification
Red Pitaya ADC	14-bit, 125 MSps
Red Pitaya DAC	14-bit, 125 MSps
Catheter Electrodes	Multi-electrode array
Communication Protocol	Ethernet

Table 4.1: Specifications of the Electronic Components

4.6.2 Data Acquisition Process

The data acquisition process involves the capture of impedance signals from the pulmonary tissues using the specialized catheter. These signals are then digitized and processed by the Red Pitaya platform.

Signal Capture

The catheter injects a current and voltage signals captures the voltage signals, which are then fed into the ADCs of the Red Pitaya. The high sampling rate ensures that even the subtle variations in impedance are accurately captured.

Real-Time Processing

The FPGA on the Red Pitaya processes the digitized signals in real-time and calculates the impedance values by obtaining the FFT of both the voltage and current acquisitions and separating the real and imaginary components of the impedance. The calculated impedance is then transmitted to the main application database for further processing and visualization.

Data Transmission

The processed data is transmitted to the main application over Ethernet. This ensures that the data is available for real-time visualization and analysis, providing immediate feedback on the tissue properties.

- **Sample Rate:** The ADCs sample the signals at 125 MSps, providing high-resolution data. With an 8x decimation performed by the FPGA, this results in a sampling frequency of 15.625 MHz.
- **Data Format:** The processed data is formatted for compatibility with the main application, including magnitude and phase information.
- **Latency:** The low-latency communication ensures that the data is available in real-time for immediate analysis.

The program performs some calculations before using the data, next the operations involved in data acquisition are explained, and the specifics of the electronic components used in the system. The measurements are taken at frequencies ranging from 0.95 kHz to 897.85 kHz, as detailed in Appendix C.

4.6.3 Challenges and Solutions in Data Acquisition

The data acquisition process presents several challenges, particularly in ensuring accurate and reliable measurements.

Real-Time Data Processing:

- **Challenge:** Real-time processing of high-speed data requires efficient algorithms and robust hardware. The computational complexity of FFT and impedance calculations can lead to latency issues.
- **Solution:** The FPGA on the Red Pitaya is utilized for its parallel processing capabilities, allowing for efficient real-time calculations. It was necessary to modify the Python program to avoid processing all the samples in real time, as it is not feasible for the human eye to perceive such a high data rate per second. However, all data is still stored for future analysis.

Data Integrity and Transmission:

- **Challenge:** Ensuring data integrity during transmission from the Red Pitaya to the main application is crucial. Any data loss or corruption can compromise the accuracy of the analysis.

- **Solution:** The system uses reliable communication protocols and error-checking mechanisms to ensure data integrity. The Ethernet connection provides a stable and high-bandwidth channel for data transfer.

Component	Function
Ablation Catheter	Signal capture and conditioning
Red Pitaya ADC	Signal digitization (125 MSps)
Red Pitaya FPGA	Real-time processing (FFT, impedance calculation)
Ethernet	Data transmission to main application

Table 4.2: Key Components and Their Functions in Data Acquisition

4.7 Mathematical Calculations and Signal Processing

The core functionality of the application involves mathematical calculations and signal processing techniques to accurately determine the impedance characteristics of the tissue.

4.7.1 Impedance Calculation

The impedance Z of the tissue is a complex quantity defined as the ratio of the voltage V to the current I. Mathematically, it can be expressed as:

$$Z = \frac{V}{I} \tag{4.1}$$

Given that both voltage and current are complex quantities, impedance can be separated into its real part (resistance R) and imaginary part (reactance X):

$$Z = R + jX \tag{4.2}$$

Where j is the imaginary unit. The magnitude $|Z|$ and phase θ of the impedance are given by:

$$|Z| = \sqrt{R^2 + X^2} \tag{4.3}$$

$$\theta = \tan^{-1}\left(\frac{X}{R}\right) \tag{4.4}$$

These calculations are fundamental to the application as they allow the differentiation between healthy and pathological tissues based on their electrical properties.

4.7.2 Fourier Transform and Frequency Analysis

The application utilizes the Fast Fourier Transform (FFT) to analyze the frequency components of the captured signals. The FFT is an efficient algorithm to compute the Discrete Fourier Transform (DFT) and its inverse. For a signal $x[n]$ of length N, the DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn} \tag{4.5}$$

The inverse DFT is given by:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{j\frac{2\pi}{N}kn} \tag{4.6}$$

By applying the FFT to the sampled signal, both voltage and current, the application can identify the frequency components present in the tissue impedance. This frequency analysis is crucial for understanding the tissue response at different excitation frequencies.

Phase and Magnitude Calculation

The phase θ and magnitude $|Z|$ of the impedance are calculated from the FFT results. For each frequency component k, the magnitude and phase are computed as follows:

$$|Z[k]| = \sqrt{(\Re(X[k]))^2 + (\Im(X[k]))^2} \tag{4.7}$$

$$\theta[k] = \tan^{-1}\left(\frac{\Im(X[k])}{\Re(X[k])}\right) \tag{4.8}$$

Where $\Re(X[k])$ and $\Im(X[k])$ are the real and imaginary parts of the FFT output for frequency component k. These values are then normalized to ensure consistency across different measurements.

4.7.3 Data Smoothing and Filtering

To improve the accuracy of the impedance measurements, the application employs data smoothing and filtering techniques. These techniques reduce noise and enhance the signal quality. One common approach is the use of a low-pass filter, which can be mathematically represented as:

$$Y[n] = \sum_{k=0}^{M-1} h[k] \cdot x[n-k] \tag{4.9}$$

Where $Y[n]$ is the filtered signal, $h[k]$ is the filter coefficient, $x[n]$ is the input signal, and M is the filter order. The choice of filter coefficients $h[k]$ is critical and is typically designed using techniques such as the window method or the frequency sampling method.

4.7.4 Real-Time Processing and Data Transmission

The processed impedance data is transmitted in real-time to the main application for visualization and analysis. The application uses efficient data structures and algorithms to ensure that the processing and transmission are performed with minimal latency. This real-time capability is essential for providing immediate feedback during medical procedures.

The mathematical models and algorithms described here form the foundation of the signal processing capabilities of the application. They enable accurate and efficient analysis of the impedance data, facilitating the identification of pathological tissues with high precision.

Operation	**Equation**
Impedance	$Z = \frac{V}{I}$
Magnitude	$\lvert Z \rvert = \sqrt{R^2 + X^2}$
Phase	$\theta = \tan^{-1}\left(\frac{X}{R}\right)$
DFT	$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}$
IDFT	$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] \cdot e^{j\frac{2\pi}{N}kn}$
Filtered Signal	$Y[n] = \sum_{k=0}^{M-1} h[k] \cdot x[n-k]$

Table 4.3: Key Equations Used in the Application

4.8 Program Operations and Algorithms

The *Python* application has been developed to achieve accurate and efficient processing of impedance data. The primary focus is on the programmatic aspects, including data handling, real-time processing, and the mathematical models integrated within the application.

4.8.1 Receiver Class and Data Handling

The application incorporates a specialized `Receiver` class designed to handle data acquisition in a separate process, preventing the main GUI from freezing. This class is responsible for establishing connections, receiving data packets, and storing the data for further processing.

Initialization and Connection: The `Receiver` class initializes with several parameters, including host, port, file name, data size, stop flag, queue, and frequency values. These parameters configure the class for efficient data handling.

```
class Receiver(multiprocessing.Process):
    def __init__(self, host, port, file, size, stop, queue,
   frequencies_val):
        super().__init__()
        self.host = host
        self.port = port
        self.file = file
        self.size = size
        self.stop = stop
        self.queue = queue
        self.frequencies_val = frequencies_val
```

Listing 4.1: Initialization of class with Required Data

Data Reception and Storage: The `run` method establishes a socket connection to the server and continuously receives data packets until the stop flag is set. The data is written to a file for persistent storage and simultaneously processed in real-time.

```
def run(self):
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((self.host, self.port))
        with open(self.file, 'wb') as f:
            while not self.stop.is_set():
                data = s.recv(self.size)
                f.write(data)
                self.process_data(data)
```

Listing 4.2: Connection of Red Pitaya

4.8.2 Real-Time Data Processing

The `Receiver` class processes the received data packets to extract impedance information. This involves converting the byte stream into a structured format, and calculating impedance values from the voltage and current coefficients.

Byte Stream Conversion: The data received from the catheter is in the form of a byte stream. This byte stream is converted into an array of complex numbers representing the raw signal.

```
def process_data(self, data):
    num_frames = len(data) // (52 * 8)
    for frame_start in range(0, num_frames * (52 * 8), 52 * 8):
        data_frame = data[frame_start:frame_start + (52 * 8)]
        read_frame = np.frombuffer(data_frame, dtype='<i2', count=
   52*4)
        self.calculate_impedance(read_frame)
```

Listing 4.3: Process and Save Data on a Buffer

Impedance Calculation: The *calculate_impedance* method separates the real and imaginary parts of the signal, sorts them, and calculates the impedance. The impedance magnitude and phase are then normalized and transmitted to the main application.

```
def calculate_impedance(self, read_frame):
    xk_index = read_frame[2::4]
    frame_re = read_frame[0::4]
    frame_im = read_frame[1::4]
    frame = frame_re + 1j * frame_im
    sort_samples = np.argsort(xk_index)
    frame_sorted = frame[sort_samples]
    frame_CH1 = [(np.conj(frame_sorted[-i-1]) + frame_sorted[i])/2
   for i in range(26)]
    frame_CH2 = [1j*(np.conj(frame_sorted[-i-1]) - frame_sorted[i])/2
    for i in range(26)]
    z = np.divide(np.array(frame_CH2), 1e-3 * np.array(frame_CH1))
    z_mag = np.abs(z)
    z_phase = np.mod(np.arctan2(np.imag(z), np.real(z)) * 180 / np.pi
    + 360, 360)
```

```
    self.transmit_data(z_mag, z_phase)
```

Listing 4.4: Impedance Calculus

4.8.3 Data Transmission and Integration with Main Application

The processed data, including the impedance magnitude and phase, is transmitted to the main application using a queue. This ensures seamless integration and real-time visualization of the data.

Queue Transmission: The impedance data is placed into a queue, which is continuously monitored by the main application. This design ensures that the data is available for real-time analysis and visualization.

```
def transmit_data(self, z_mag, z_phase):
    for i in range(len(self.frequencies_val)):
        self.queue.put([z_mag[i], z_phase[i]])
```

Listing 4.5: Process the impedance magnitude and phase for each frequency

Main Application Integration: The main application retrieves the data from the queue and updates the GUI with real-time impedance information. This involves plotting the impedance magnitude and phase across different frequencies, providing immediate feedback to the user.

4.9 Data Flow and Visualization: InfluxDB and Grafana

The data flow begins with acquisition, followed by uploading to InfluxDB, and connecting to Grafana for real-time visualization. The process involves capturing impedance data, storing it in a time-series database, and visualizing it using Grafana. Additionally, we discuss the integration with a Flask-based dashboard for managing measurements.

4.9.1 Uploading Data to InfluxDB

The impedance data is uploaded to InfluxDB, a time-series database optimized for handling high-write loads and real-time queries. The following code snippet demonstrates the process of formatting the data and uploading it to InfluxDB.

Data Formatting: Each measurement is formatted using the InfluxDB Line Protocol, which includes a measurement name, tags, fields, and a timestamp. The frequency at which the measurement is taken is used as a tag, ensuring that data can be queried efficiently based on frequency.

```
# Create the data points in Line Protocol format
module_line = f"module,frequency={self.frequencies_val[i]} value={
   float(z_mag[i])}"
phase_line = f"phase,frequency={self.frequencies_val[i]} value={float
   (z_phase[i])}"

# Write the data points to InfluxDB
```

```
self.client.write([module_line, phase_line], {'db': 'bioimpedance'}, 204, 'line')
print("Data uploaded to InfluxDB for the specified frequency.")
```

Listing 4.6: Create and upload to InfluxDB the data points in Line Protocol format

Data Upload Process: The data is uploaded to InfluxDB in real-time as it is processed. This ensures that the database is continuously updated with the latest measurements, making them available for immediate visualization and analysis.

```
def transmit_data(self, z_mag, z_phase):
    for i in range(len(self.frequencies_val)):
        # Create data points
        module_line = f"module,frequency={self.frequencies_val[i]} value={float(z_mag[i])}"
        phase_line = f"phase,frequency={self.frequencies_val[i]} value={float(z_phase[i])}"

        # Upload data to InfluxDB
        self.client.write([module_line, phase_line], {'db': 'bioimpedance'}, 204, 'line')
        print("Data uploaded to InfluxDB for the specified frequency.")
```

Listing 4.7: Write the data to InfluxDB database

4.9.2 Saving the data on an InfluxDB database

The previous program displayed about 30 samples per second at all frequencies regarding impedance, both in magnitude and phase. This means that the Python program, in addition to saving in real time the voltage and intensity of all samples taken by the Red Pitaya, of the approximately 900 it performs, calculates the impedance in magnitude and phase on 30 equispaced samples (performing the mathematical operation), and displays the result on screen.

This is quite an archaic method because it simply serves to view the measurement on the screen, as the impedance is not saved when storing, but rather a binary file with all 900 measurements per second of voltage and intensity. To address this issue, it has been decided to install the **InfluxDB** software and set up a database over which Python will communicate. The Python code has been modified so that these 30 samples per second, once the impedance in magnitude and phase has been calculated, are saved in the database, adding a label or 'tag' with the frequency at which the measurement was made.

4.9.3 Connecting to Grafana for Visualization

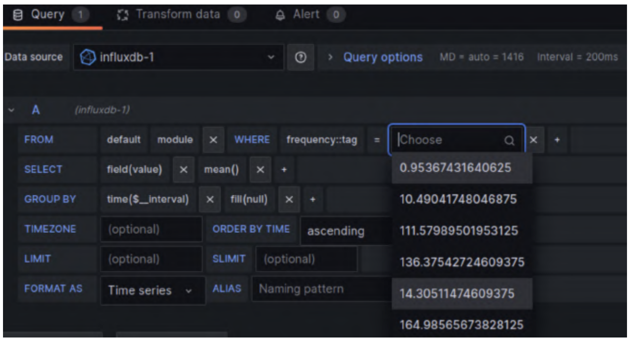
Grafana is used to visualize the impedance data stored in InfluxDB. Grafana's powerful query capabilities and interactive dashboards provide a comprehensive view of the impedance measurements over time and frequency.
It is pointless to store data in a database if we cannot view, configure visualizations, and analyze them. That is why we have chosen to use **Grafana**, a data visualization software

that is open-source under the Apache 2.0 license, which allows for the visualization and formatting of metric data. It has been linked with the database to view the data.

Figure 4.6: Data on Grafana
Source: Own

Thus, now if we want, we can filter by frequency and zoom into the areas of interest.

After modifying the code so that when saving the sample, it includes the label with the corresponding frequency.



Data source: influxdb-1

MD = auto = 1416 Interval = 200ms

FROM default module WHERE frequency::tag =

SELECT field(value) mean()

GROUP BY time($__interval) fill(null)

TIMEZONE (optional) ORDER BY TIME ascending

LIMIT (optional) SLIMIT (optional)

FORMAT AS Time series ALIAS Naming pattern

0.95367431640625
10.49041748046875
111.57989501953125
136.37542724609375
14.30511474609375
164.98565673828125

Figure 4.7: Frequency Selection on Grafana
Source: Own

Grafana Setup: To set up Grafana for visualizing the impedance data, follow these steps:
1. **Add InfluxDB as a Data Source:** In Grafana, configure InfluxDB as a data source by providing the connection details, including the database name and authentication credentials.
2. **Create Dashboards:** Create dashboards in Grafana to visualize the impedance data. Each dashboard can include multiple panels, each displaying different aspects of the data, such as impedance magnitude over time or phase over frequency.
Querying Data: Grafana queries the data from InfluxDB using its query language. For frequency-based graphs, the following query is used to fetch the last 26 data points for

different frequencies:

```
SELECT LAST("value") AS "impedance", "frequency"
FROM "autogen"."module"
WHERE $timeFilter
GROUP BY "frequency"::tag
ORDER BY time DESC
LIMIT 26
```

Listing 4.8: QUERY for generating the frequency graph in Grafana

Integration with Flask Dashboard: The application includes a Flask-based dashboard to control the measurements and view the recorded data. The dashboard interacts with Grafana to display real-time visualizations and provides a user-friendly interface for managing the measurement process.

4.9.4 Impedance Magnitude Over Time

One of the key visualizations is the impedance magnitude over time. This graph provides insights into how the impedance magnitude changes over the duration of the measurement.
Graph Configuration: The graph is configured in Grafana to display the impedance magnitude for each frequency over time. The time range can be adjusted to focus on specific periods, and the data points are plotted to show trends and anomalies.
Query Used: The following query is used to fetch the impedance magnitude data from InfluxDB:

```
SELECT "value"
FROM "module"
WHERE $timeFilter
```

Listing 4.9: QUERY for generating the time graph in Grafana

Analysis: This visualization helps in identifying any significant changes in impedance magnitude, which could indicate changes in tissue properties. For instance, a sudden increase in magnitude could signify the presence of pathological tissue.

4.9.5 Impedance Phase Over Time

The phase of the impedance is another critical parameter that is visualized over time. This graph helps in understanding the phase variations in the tissue's electrical response.
Graph Configuration: Similar to the magnitude graph, the phase graph is configured to display the phase of the impedance for each frequency over time. Grafana's interactive features allow for detailed examination of specific time periods.
Query Used: The following query is used to fetch the impedance phase data from InfluxDB:

```
SELECT "value"
FROM "phase"
WHERE $timeFilter
```

Listing 4.10: QUERY for generating the phase time graph in Grafana

Analysis: Phase variations can provide additional information about the tissue's electrical properties. Significant phase shifts may indicate abnormalities in the tissue, aiding in diagnosis.

4.9.6 Impedance Magnitude over Frequency

This graph shows the impedance magnitude as a function of frequency. It is particularly useful for analyzing the frequency response of the tissue.
Graph Configuration: The graph is configured to display the last 26 frequency points, providing a snapshot of the impedance magnitude across different frequencies. This helps in identifying frequency-dependent properties of the tissue.
Query Used: The following query fetches the data points required for this graph:

```
SELECT LAST("value") AS "impedance", "frequency"
FROM "autogen"."module"
WHERE $timeFilter
GROUP BY "frequency"::tag
ORDER BY time DESC
LIMIT 26
```

Listing 4.11: QUERY for generating the module frequency graph in Grafana

Analysis: Frequency-dependent variations in impedance magnitude can reveal critical information about the tissue. For example, certain pathological tissues may exhibit distinct impedance characteristics at specific frequencies.

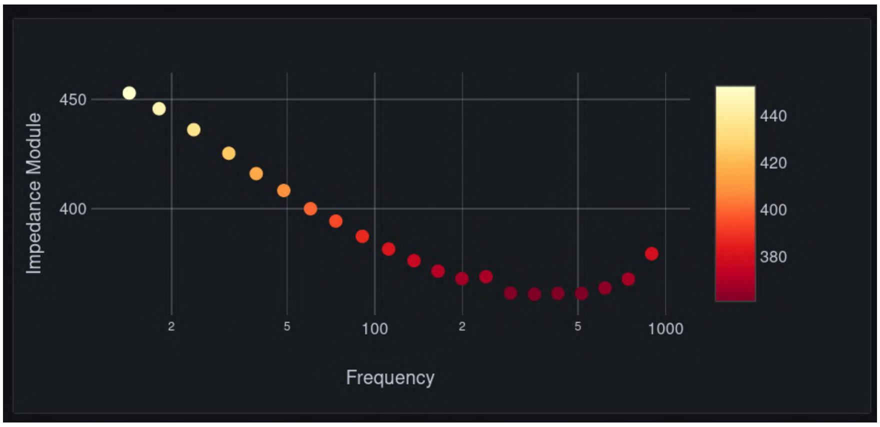


Figure 4.8: Impedance Magnitude over Frequency
Source: Own

4.9.7 Impedance Phase Over Frequency

This graph shows the impedance phase as a function of frequency, complementing the magnitude over frequency graph.
Graph Configuration: The phase graph is configured similarly to the magnitude graph, displaying the phase across the last 26 frequency points.
Query Used: The following query fetches the data points for the phase graph:

```sql
SELECT LAST("value") AS "impedance", "frequency"
FROM "autogen"."phase"
WHERE $timeFilter
GROUP BY "frequency"::tag
ORDER BY time DESC
LIMIT 26
```

Listing 4.12: QUERY for generating the phase frequency graph in Grafana

Analysis: Analyzing the phase response across frequencies helps in gaining a complete understanding of the tissue's electrical properties. This can be crucial for identifying pathological changes that are frequency-dependent.

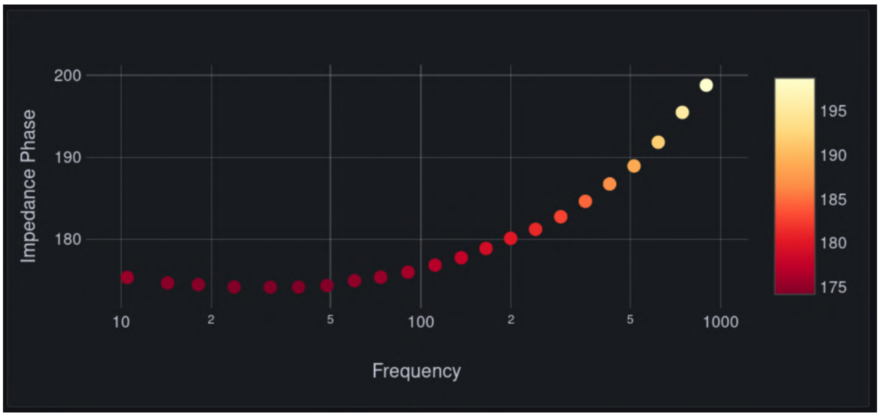


Figure 4.9: Impedance Phase over Frequency
Source: Own

Visualization	Parameter	Query
Magnitude over Time	Impedance Magnitude	SELECT "value" FROM "module" WHERE $timeFilter
Phase over Time	Impedance Phase	SELECT "value" FROM "phase" WHERE $timeFilter
Magnitude over Frequency	Impedance Magnitude	SELECT LAST("value") AS "impedance", "frequency" FROM "autogen"."module" WHERE $timeFilter GROUP BY "frequency"::tag ORDER BY time DESC LIMIT 26
Phase over Frequency	Impedance Phase	SELECT LAST("value") AS "impedance", "frequency" FROM "autogen"."phase" WHERE $timeFilter GROUP BY "frequency"::tag ORDER BY time DESC LIMIT 26

Table 4.4: Queries Used for Grafana Visualizations

Chapter 5

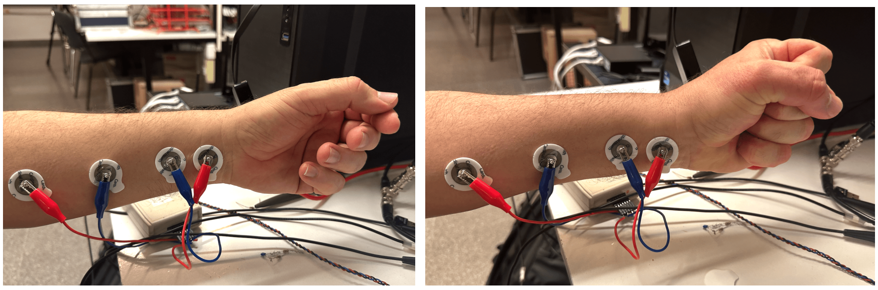
Results

Some representative results of the work are presented below, focusing on the interface we created for data management and visualization, and the experiments conducted in the laboratory. The interface includes a Flask dashboard for patient management and a Grafana dashboard for real-time data visualization. We conducted our tests in the laboratory using an RC network to simulate tissue and monitored the electrical impedance using electrodes on the hands. Additionally, we have used anonymous measurement files, which include data from healthy tissue, bronchial tissue, and tumors.

5.1 Phenomenon in Electrical Impedance Measurement

Some observations regarding the phenomenon that occurs when measuring electrical impedance in the arm using a four-electrode method have been made in an experiment: two electrodes for current injection and two for voltage measurement.

When the arm is compressed, it has been observed that the impedance module increases to 462 ohms. This increase is attributed to the compression of the muscle fibers, which results in higher impedance. As the muscle fibers are compressed, there is less space between them, leading to greater resistance to the electrical current.



(a) Relaxed hand

(b) Contracted hand

Figure 5.1: Contraction and expansion of the hand to see changes in impedance

Source: Own

Upon releasing the compression, the impedance drops back to 450 ohms. This fluctuation is directly related to the physical changes in the structure of the muscle tissue. Compressing the arm again causes the impedance to rise back to 462 ohms.

These observations highlight the sensitivity of electrical impedance measurements to changes in muscle fiber density and spacing, providing valuable insights into muscle physiology and the effects of physical pressure on tissues.

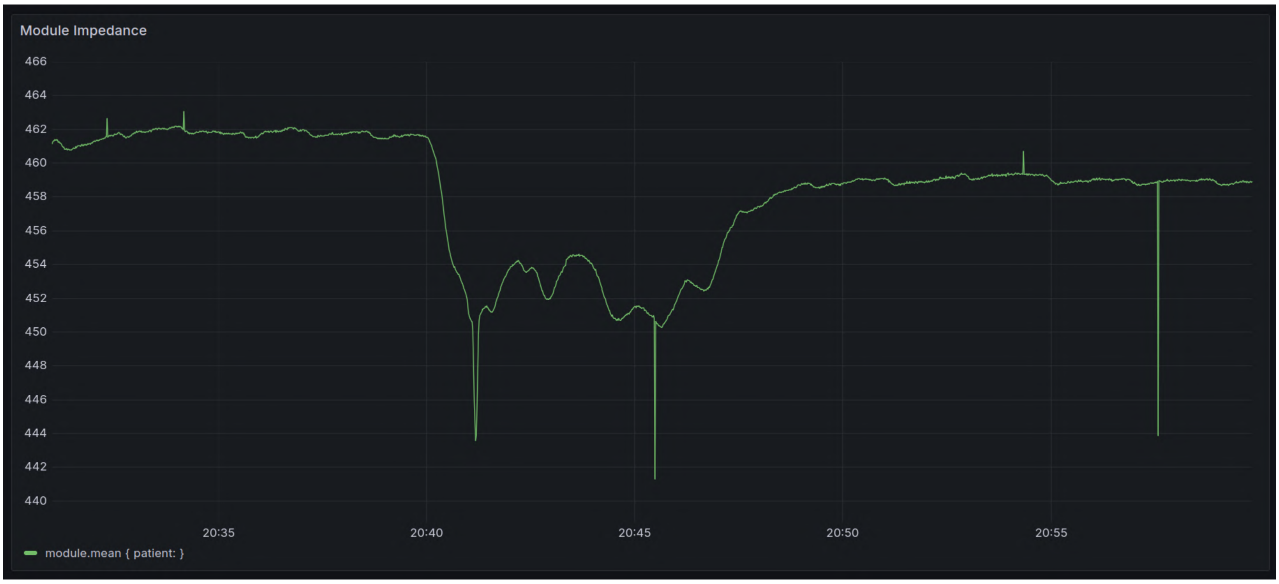


Figure 5.2: Time Graphic in Hand Dilatation and Contraction

Source: Own

5.2 Observation of Respiratory Cycles

In this experiment, the electrical impedance between two electrodes placed on each hand has been measured. The recorded data, visualized in the provided graph, shows clear cyclic variations corresponding to the subject's breathing cycles.

A four-electrode configuration has been used: two electrodes for injecting a small alternating current and two for measuring the resultant voltage. This setup minimizes the impact of electrode-skin contact impedance, allowing for more accurate readings of the body's internal impedance. Changes in the impedance can be attributed to physiological variations, such as those caused by breathing.

5.2.1 Bioimpedance and Respiratory Cycles

Breathing causes **rhythmic expansions and contractions of the chest and abdominal cavities**, which in turn influence the **distribution and properties of the tissues within these regions**. These changes affect the **electrical impedance of the body**, as the composition and density of tissues alter with each breath.

The graph shown in Figure 5.3 illustrates the module of impedance over time. The waveform exhibits **periodic peaks and troughs, which correspond to the inhalation and exhalation phases of the respiratory cycle**.

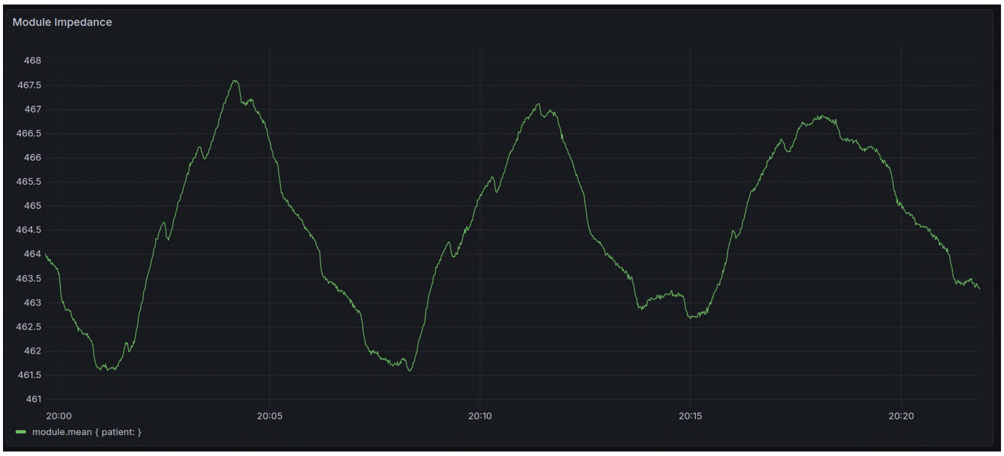


Figure 5.3: Impedance module measurement showing respiratory cycles.

5.2.2 Relation to Respiratory Phases

During inhalation, the lungs expand, and the volume of air increases, causing a slight decrease in tissue density. This results in higher impedance, observed as the peaks (467.6 Ω) in the graph. Conversely, during exhalation, the lungs contract, air volume decreases, and tissue density increases, leading to lower impedance, seen as the valleys (461.6 Ω) in the graph.

The periodicity of the waveform corresponds directly to the breathing rate of the subject. Each complete cycle in the graph (one peak and one trough) represents one full breath, consisting of both inhalation and exhalation phases.

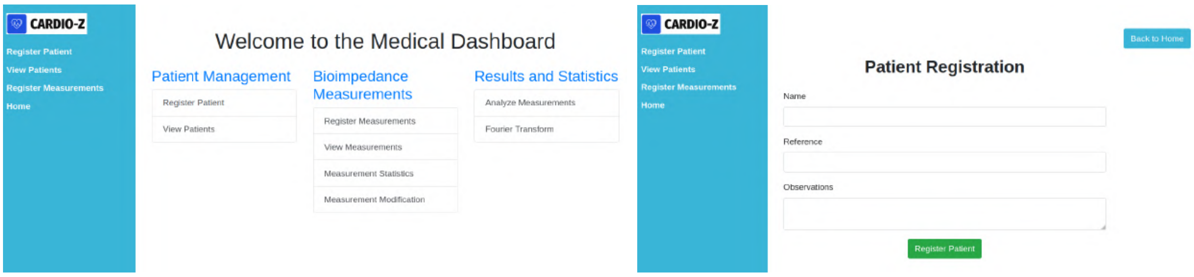
5.3 Interface Description

5.3.1 Flask Dashboard

The Flask dashboard allows us to register patients and create measurements for each of them, noting observations that can be related in the future. This interface is user-friendly and ensures that all patient data is systematically organized.

- **Patient Registration:** The interface provides fields for entering patient information such as name, age, and medical history.

- **Measurement Creation:** For each patient, we can create multiple measurements, annotating observations that will be useful for future analysis.



(a) Dashboard Home (b) Patient Registration

Figure 5.4: Flask Dashboard
Source: Own

5.3.2 Grafana Dashboard

The Grafana dashboard provides real-time visualization of data. We can view the data on both temporal and frequency axes, allowing for comprehensive analysis.

- **Real-Time Data Visualization:** The dashboard displays real-time data from the measurements, showing variations over time.
- **Frequency Analysis:** Parallel to the real-time data, a program allows us to analyze the frequency response of the data retrieved from Grafana.

Figure 5.5: (1) Top right: Temporal graph of impedance modulus with series for different frequencies, distinguished by colors according to the legend at the bottom. (2) Top left: Frequency graph of the measurement with a scale. (3) Bottom right: Impedance phase. (4) Bottom left: Impedance modulus without frequency differentiation.

5.4 Laboratory Experiments

5.4.1 Experimental Setup

The experiments were conducted in the IEB laboratory. These tests have not yet been performed on humans but used an RC network to simulate tissue.

5.4.2 RC Network Simulation

A RC network comprising a 10 kΩ resistor in parallel with a series combination of a 2 kΩ resistor and a 3.3 nF capacitor has been used. This network simulates the electrical properties of biological tissue.

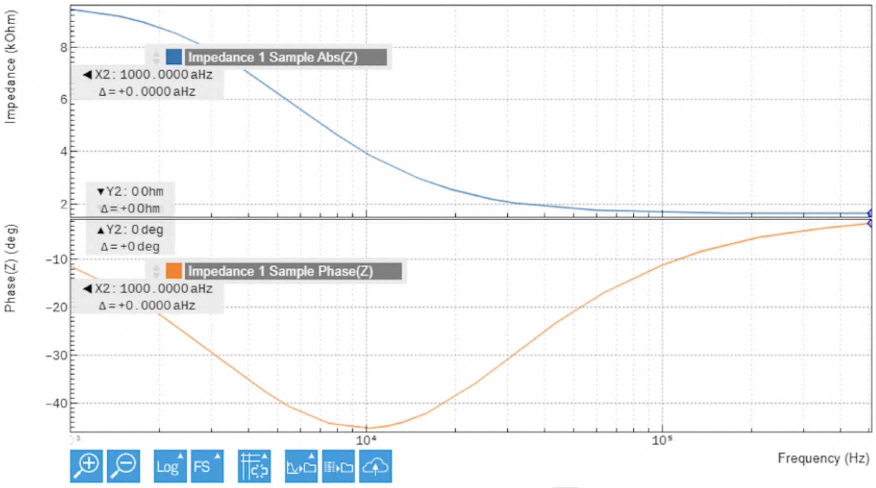


Figure 5.6: RC Network: 10k ohm resistor in parallel with a series combination of a 2k ohm resistor and a 3.3nF capacitor.

The impedance Z of this network can be calculated using the following formula:

$$Z = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2 + \frac{1}{j\omega C}}}$$

where R_1 is the 10k ohm resistor, R_2 is the 2k ohm resistor, C is the 3.3nF capacitor, and ω is the angular frequency.

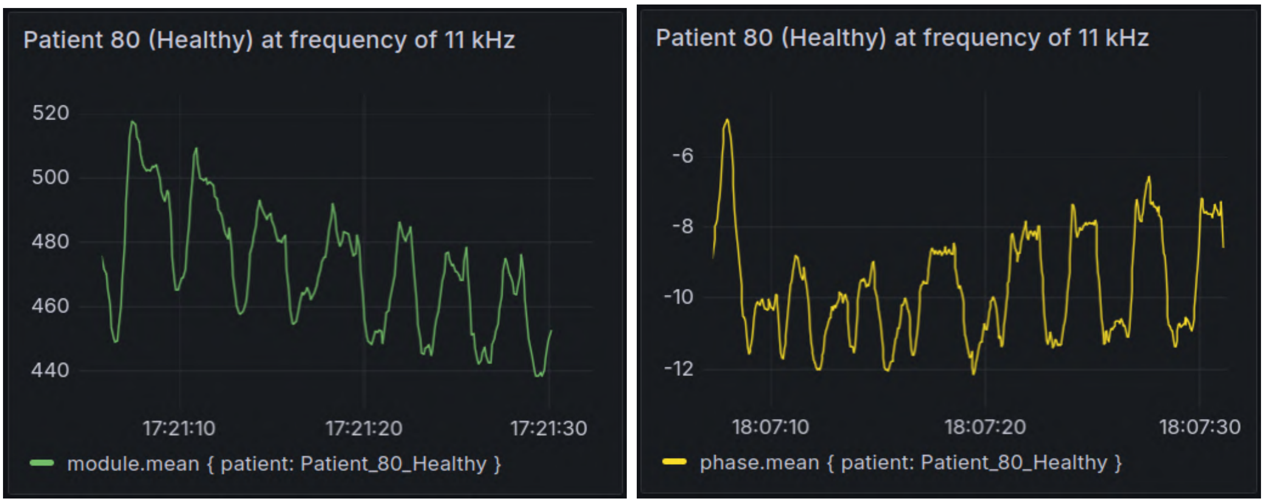
5.4.3 Electrode Measurements

We also used electrodes on the hands to monitor changes due to breathing. This setup allowed us to observe the impedance variations caused by physiological changes in the body.

5.4.4 Anonymous Measurement Files

We have collected anonymous measurement files along with their explanations. Below are the observations for different tissue types.

- **Healthy Tissue:** The impedance magnitude is high, and the phase is also relatively high.
- **Tumor Tissue:** The impedance magnitude drops to approximately one-fourth of the healthy tissue value, and the phase is lower than in the case of the healthy tissue.
- **Bronchial Tissue:** The impedance magnitude is low, and there is almost no phase shift, revealing the resistive behavior of the bronchial tissue.

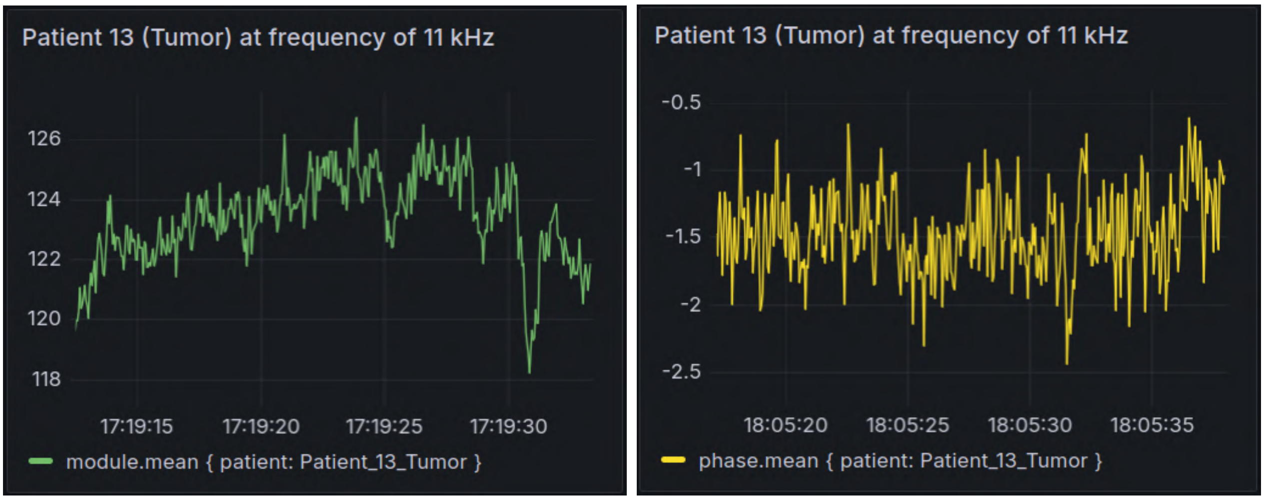


(a) Modulus

(b) Phase

Figure 5.7: Impedance measurement of healthy tissue: High magnitude and relatively high phase. **Effect of ventilation is visible**

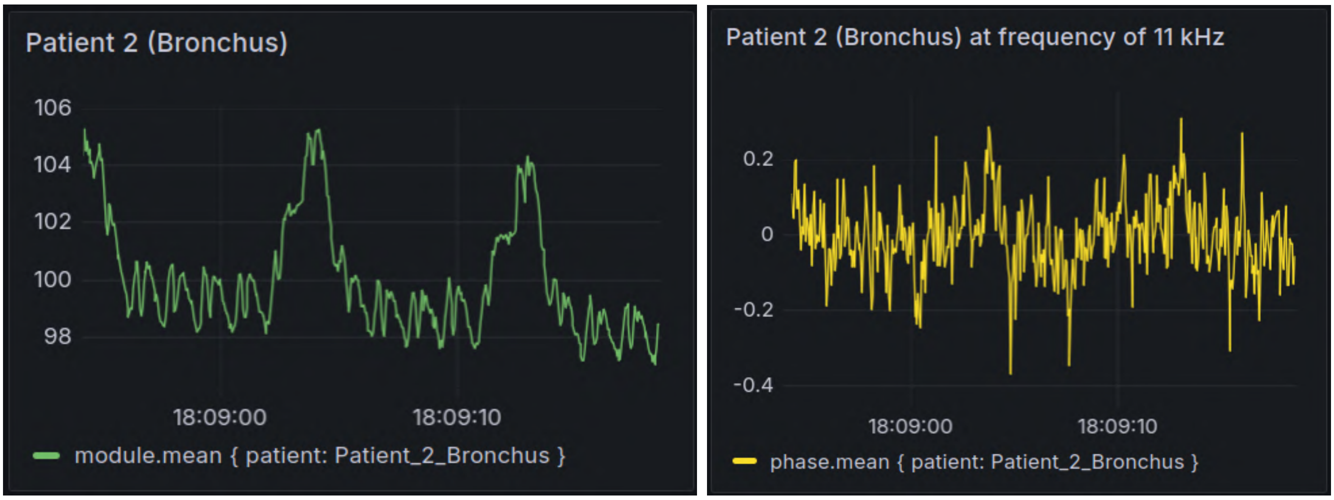
Source: Own



(a) Modulus (b) Phase

Figure 5.8: Impedance measurement of tumor tissue: Reduced magnitude and lower phase.

Source: Own



(a) Modulus (b) Phase

Figure 5.9: Impedance measurement of bronchial tissue: Low magnitude and negligible phase shift.

Source: Own

Chapter 6

Conclusions and Future Lines

Electrical impedance measurement is a promising technique in biomedical engineering, which can be used to analyze the properties of biological tissues. By applying an alternating current through a set of electrodes and measuring the resultant voltage, one can determine the impedance, which provides insights into the physiological and pathological states of tissues.

6.1 Introduction

Coming from a very basic Python program that provided only static images for data visualization, we identified a significant need to develop a tool capable of handling real-time data as well as post-processing capabilities. This led to the development of an advanced dashboard integrated with Grafana, allowing comprehensive management and analysis of patient data and measurements.

6.2 Development of the Advanced Dashboard

6.2.1 Integration with Grafana

The new tool integrates seamlessly with Grafana, providing a robust platform for real-time data visualization. This dashboard enables us to register patient data and associate it with the corresponding measurements. Key functionalities include:

- **Real-Time Visualization:** The dashboard displays real-time graphs of impedance magnitude and phase with a refresh time of 50 ms.
- **Magnitude and phase angle diagrams**: time sequences at selected frequencies, and frequency graphs.
- **Data Registration:** Patients' data and related measurements can be systematically recorded for future reference and analysis.

6.2.2 Challenges in Real-Time Streaming

One of the major challenges was achieving real-time streaming due to the high volume of data. As data is received, it must be simultaneously stored in a buffer and uploaded to

the InfluxDB database. Additionally, Grafana's default refresh time is 5 seconds, which was inadequate for our needs.
To address this, we modified the path files in Grafana to achieve a refresh time of 50 ms. We also integrated a Plotly plugin in Grafana to handle frequency plots, utilizing tags in InfluxDB to store frequency data.

Challenge	**Solution**
High data volume	Buffering and InfluxDB integration
Grafana refresh time	Path file modification to 50 ms
Frequency visualization	Plotly plugin integration

Table 6.1: Challenges and Solutions in Real-Time Data Streaming

6.3 Future Directions

6.3.1 Machine Learning Application

A significant future direction for this project is the exploration of machine learning algorithms to enhance the dashboard's capabilities. Potential algorithms include Support Vector Machines (SVM), Naïve Bayes, and decision trees. The goal is to develop a system capable of classifying tissues state based on impedance data using artificial intelligence.

- **Machine Learning Algorithms:** SVM, Naïve Bayes, and decision trees will be explored for signal classification.

- **Automatic Tissue State Estimation:** Define estimators that can automatically determine the state of the tissue based on provided frames and frequency data to **allow performing real-time electronic biopsy**.

Chapter 7

Sustainability Analysis and Ethical Implications

7.1 Introduction

According to Article 4.2 of *Real Decreto* 822/2021, all university study plans must reference respect for gender equality, the principle of non-discrimination for any reason, and consideration of sustainability and climate change.
This is a sustainability analysis and ethical implications report for the BT (Bachelor's Thesis), assessing the environmental, social, and economic impacts, as well as the potential ethical implications.

7.2 Sustainability Matrix

The sustainability matrix allows us to organize different concepts related to the BT. It includes three parts: the development of the BT, the execution of the project, and the risks and limitations inherent in both the academic work and the project. Each part is analyzed from three perspectives: environmental, economic, and social, representing the three dimensions of sustainability.

7.2.1 Environmental Impact

Development of BT

The environmental impact during the preparation of the BT includes energy consumption, material usage, and waste generation. The main environmental considerations are as follows:
Energy Consumption: The BT involves significant use of computing resources. The total power consumption for the equipment used (servers, Red Pitaya, frontend systems) is calculated as follows:

$$\text{Total Power Consumption} = \sum_{i=1}^{n} P_i \times T_i \tag{7.1}$$

where P_i is the power consumption of device i and T_i is the total time the device is used.

Assuming:

- **Server:** 500W, 5 hours/day, 48 days
- **Red Pitaya:** 10W, 5 hours/day, 48 days
- **Frontend System:** 15W, 5 hours/day, 48 days

$$\text{Total Energy (kWh)} = (500 \times 5 \times 48 + 10 \times 5 \times 48 + 15 \times 5 \times 48)/1000 = 126\ kWh \tag{7.2}$$

CO_2 **Emissions:** Using the carbon intensity of electricity (e.g., 0.24 kg CO2/kWh for Spain), the total CO_2 emissions are:

$$\text{Total } CO_2 \text{ Emissions (kg)} = 126 \times 0.24 = 30.24 \text{ kg } CO_2 \tag{7.3}$$

Material Usage: The BT uses electronic components and computing resources, all of which have an environmental footprint. The materials used include semiconductors, metals, and plastics, primarily in the Red Pitaya and computing systems.

Project Execution

The environmental impact of the project during its entire lifespan is considered, including energy usage, waste generation, and potential recycling.
Energy Usage: The project will continue to use similar equipment. Annual energy usage is estimated for continuous operation.
Waste Generation and Recycling: The end-of-life disposal of electronic components must follow proper e-waste recycling protocols. This includes the potential for recycling and reusing parts to minimize environmental impact.

Risks and Limitations

Potential environmental risks include scenarios that could increase the project's footprint, such as unexpected increases in energy usage or challenges in recycling electronic waste.
Environmental Risks: These include higher than expected energy consumption or difficulties in recycling due to changes in technology or regulations.
Limitations: The main limitation is the estimation accuracy of energy consumption and material usage, which could be refined with more precise data and longer observation periods.

7.3 Economic Impact

Development of BT

The economic impact includes resource consumption and the cost of these resources.
Human and Material Costs: The BT involves significant time investment and material costs.
Human Costs: The project required 450 hours of work from the project engineer, calculated as follows:

- 5 hours/day for 48 days, totaling 450 hours.
- Hourly rate of 15 euros/hour.

Total Human Cost $= 450 \text{ hours} \times 15$ €/hour $= 6750$ €
Material Costs: Although the project did not require the acquisition of additional electronic materials as they were available from the IEB Research Group, the potential costs for these materials are listed below for reference:

Item	**Cost (€)**
Computer	850
Screen	150
Red Pitaya	490
Frontend System	300
Total Potential Cost	1790

Table 7.1: Potential Material Costs if Acquired

Energy Costs: The project also incurred costs associated with the operation of various equipment. The energy consumption was calculated based on the following assumptions:

- **Server:** 500W, 5 hours/day, 48 days
- **Red Pitaya:** 10W, 5 hours/day, 48 days
- **Frontend System:** 15W, 5 hours/day, 48 days

$$\text{Total Energy (kWh)} = (500 \times 5 \times 48 + 10 \times 5 \times 48 + 15 \times 5 \times 48)/1000 = 126 \text{ kWh} \quad (7.4)$$

Resource	**Units (kWh)**	**Unit Cost (€/kWh)**	**Total Cost (€)**
Server Operation	120	0.19	22.80
Red Pitaya Operation	2.4	0.19	0.46
Frontend Operation	3.6	0.19	0.68
Total	126		23.94

Table 7.2: Energy Cost Analysis of BT Development

Total Project Cost: Summing up the human and energy costs, the total cost of the project is:
Total Project Cost = Human Cost + Energy Cost = 6750 €+ 23.94 €= 6773.94 €

Project Execution

The viability analysis of the project includes the cost of operation, maintenance, and potential updates.
Lifecycle Cost: The total cost over the project's lifecycle includes initial setup, operational costs, and maintenance.

$$\text{Lifecycle Cost} = \text{Initial Cost} + \text{Operational Cost} + \text{Maintenance Cost} \quad (7.5)$$

Risks and Limitations

Economic risks could affect the project's viability, such as unexpected costs or market changes. The limitations include the accuracy of cost estimations and unforeseen expenses.
Economic Risks: These include potential increases in energy prices or hardware costs, which could affect the project's overall budget.
Limitations: The main limitations are the assumptions made in cost calculations and the potential for unforeseen expenses during the project's lifespan.

7.4 Social Impact

Development of BT

The social impact includes the effects on the people involved in the BT, such as the student and advisor.
Personal and Professional Growth: The BT provides significant learning and professional development opportunities.
Ethical Considerations: The BT adheres to ethical standards, including the use of inclusive and non-sexist language.

Project Execution

The project's implementation can impact various social groups, including researchers, healthcare professionals, and patients.
Beneficiaries: The primary beneficiaries are patients and healthcare providers who will use the system for diagnosing and monitoring pulmonary and cardiac health.
Potential Negative Impacts: There could be negative impacts, such as dependence on technology or potential biases in the system.

Risks and Limitations

Social risks include scenarios where the project could negatively impact specific groups, and the limitations in estimating these impacts.
Social Risks: These include potential biases in data interpretation or over-reliance on technological solutions.
Limitations: The limitations include the challenges in predicting social impacts accurately and the need for ongoing monitoring and assessment.

7.5 Ethical Implications

Ethical considerations are crucial in ensuring that the project adheres to professional and societal standards.

Needs Addressed

The project addresses the need for accurate and real-time monitoring of pulmonary and cardiac health, defined by healthcare professionals and researchers.

Anticipated Consequences

Beyond the intended outcomes, the project could lead to advancements in biomedical engineering and better healthcare outcomes. However, it is essential to consider unintended consequences, such as data privacy concerns or misuse of the technology.

Adherence to Ethical Standards

The project adheres to the UPC Code of Ethics and the UPC Code of Research Integrity. Ethical considerations include data protection, informed consent, and ensuring the technology is used for the intended purposes only.

As student, I had **no access to patient information**. Only **anonymized files** were provided. The measurements were taken in the **research project RTI 2018-098116-B-C21 under the approval of the Ethics Committee CEIC-73/2010** of Hospital de Sant Pau (HSP).

7.6 Relation to Sustainable Development Goals

This project contributes to several Sustainable Development Goals (SDGs), particularly in healthcare and technology.

Good Health and Well-Being (SDG 3)

The project aims to improve health outcomes by providing accurate and real-time monitoring of pulmonary and cardiac conditions.

Industry, Innovation, and Infrastructure (SDG 9)

The project contributes to technological innovation in biomedical engineering, promoting industry growth and infrastructure development.

Responsible Consumption and Production (SDG 12)

By considering the environmental impact of the project's lifecycle, the project aligns with responsible consumption and production goals.

7.7 Conclusions

This sustainability analysis of the Bachelor's Thesis (BT) and the associated biomedical setup reveals several key conclusions across environmental, economic, social, and ethical dimensions.

7.7.1 Environmental Impact

The BT and project execution have measurable environmental impacts primarily through energy consumption and material usage. The total energy consumption for the BT development is 126 kWh, resulting in CO_2 emissions of 30.24 kg. The ongoing project will continue to use similar equipment, necessitating efficient energy use and proper e-waste recycling protocols to mitigate its environmental footprint. Risks include potential increases in energy consumption and challenges in recycling electronic waste due to technological or regulatory changes.

7.7.2 Economic Impact

The economic impact analysis indicates that the development of the BT incurs a total cost of 23.94 euros, with the majority attributed to server operation. The lifecycle cost of the project includes initial setup, operational, and maintenance costs. Economic risks include fluctuations in energy prices and hardware costs, which could affect the project's budget. Limitations of this analysis include the accuracy of cost estimations and the potential for unforeseen expenses.

7.7.3 Social Impact

The social impact of the BT encompasses personal and professional growth for the student and advisor, adhering to ethical standards such as inclusive and non-sexist language. The project benefits various social groups, particularly healthcare providers and patients, by improving diagnostic and monitoring capabilities for pulmonary and cardiac health. Potential negative impacts include reliance on technology and biases in the system. Social risks involve data interpretation biases and over-reliance on technological solutions, with limitations in accurately predicting social impacts.

7.7.4 Ethical Implications

The ethical implications highlight the necessity of addressing data privacy, informed consent, and ensuring the technology is used appropriately. The project aligns with the UPC Code of Ethics and the UPC Code of Research Integrity, addressing the need for accurate health monitoring while considering unintended consequences such as data privacy concerns or misuse of technology.

7.7.5 Relation to Sustainable Development Goals

The project contributes to several Sustainable Development Goals (SDGs):

- **Good Health and Well-Being (SDG 3):** By providing accurate and real-time monitoring of pulmonary and cardiac conditions, the project aims to improve health outcomes.
- **Industry, Innovation, and Infrastructure (SDG 9):** The project promotes technological innovation in biomedical engineering, fostering industry growth and infrastructure development.

- **Responsible Consumption and Production (SDG 12):** Through consideration of the environmental impact across the project's lifecycle, it aligns with responsible consumption and production goals.

In conclusion, the sustainability analysis underscores the importance of a balanced approach, considering environmental, economic, social, and ethical factors. The BT and its associated biomedical setup contribute positively to healthcare innovation while highlighting areas for improvement in sustainability practices.

Bibliography

[1] Wikipedia. *Cell (biology). U*pdated source as of July, 2020, and cited on June 05, 2024. *A*vailable at `https://en.wikipedia.org/wiki/Cell_(biology)`.

[2] Ivorra. *U*pdated source as of july, 2003, and cited on may 25, 2024. *A*vailable at `https://intellbio.com/wp-content/uploads/2019/09/bio-impedance-monitoring-for-physicians-1.pdf`.

[3] J. R. Bourne, *Bioelectrical impedance techniques in medicine*, in Crit. Rev. Biomed. Eng. vol. 24, 1996, p. 680. *U*pdated source as of July, 2020, and cited on June 05, 2024. *A*vailable at `https://upc.edu`.

[4] Georgina Company Se, Lexa Nescolarde, Virginia Pajares, Alfons Torrego, Pere I. Riu, Javier Rosell, and Ramon Bragos. Effect of calibration for tissue differentiation between healthy and neoplasm lung using minimally invasive electrical impedance spectroscopy. *IEEE Access*, 10:107423–107434, 2022. (accessed: 05.05.2024).

[5] Benjamin Sanchez, Johan Schoukens, Ramon Bragos, and Gerd Vandersteen. *Novel Estimation of the Electrical Bioimpedance Using the Local Polynomial Method. Application to In Vivo Real-Time Myocardium Tissue Impedance Characterization During the Cardiac Cycle. IEEE Transactions on Biomedical Engineering*, 58(12):3376–3385, 2011. (accessed: 14.06.2024).

[6] I. Montsech, *Design and Development of an FPGA-based Medical EIS System*, Master Thesis submitted to the Faculty of the Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, Universitat Politècnica de Catalunya, October 2022.

[7] *Abbot - Cardiovascular*, FlexAbility SE Ablation Catheter. *U*pdated source as of july, 2020, and cited on june 05, 2024. *A*vailable at `https://www.cardiovascular.abbott/us/en/hcp/products/electrophysiology/ablation-technology/flexability-se-ablation-catheter/about.html`.

[8] Representation of a, b and g dispersions in biological tissues, 2018. (accessed: 19.06.2024).

[9] *Python Software Foundation.* Python. *A*vailable on `https://www.python.org/`.

[10] O. Casas, R. Bragos, P. J. Riu, J. Rosell, M. Tresanchez, M. Warren, A. Rodriguez-Sinovas, A. Carreño, and J. Cinca. In vivo and in situ ischemic tissue characterization using electrical impedance spectroscopy. *Journal of Instrumentation and Bioengineering*, 10(3):123–134, 2024. (accessed: 01.06.2024).

[11] Georgina Company-Se, Lexa Nescolarde, Virginia Pajares, Alfons Torrego, Pere J. Riu, Javier Rosell, and Ramon Bragós. *Differentiation using minimally-invasive bioimpedance measurements of healthy and pathological lung tissue through bronchoscopy. Frontiers in Medicine*, 2023. (accessed: 05.05.2024).

[12] Benjamin Sanchez, Gerd Vandersteen, Irene Martin, Diego Castillo, Alfons Torrego, Pere J. Riu, Johan Schoukens, and Ramon Bragos. *In vivo electrical bioimpedance characterization of human lung tissue during the bronchoscopy procedure. A feasibility study. Medical Engineering & Physics*, 35:949–957, 2013. (accessed: 10.03.2024).

[13] Gerard Amorós-Figueras, Esther Jorge, Tomás García-Sánchez, Ramón Bragós, Javier Rosell-Ferrer, and Juan Cinca. *Recognition of Fibrotic Infarct Density by the Pattern of Local Systolic-Diastolic Myocardial Electrical Impedance. Frontiers in Physiology*, 2017. (accessed: 20.04.2024).

[14] B. Sanchez, X. Fernandez, S. Reig, and R. Bragos. *An FPGA-based frequency response analyzer for multisine and stepped sine measurements on stationary and time-varying impedance. . Measurement Science and Technology*, 25(1):015501 (12pp), 2014. (accessed: 19.02.2024).

[15] Yitaek Hwang. *How to increase Grafana refresh rate frequency. A*vailable at `https://questdb.io/blog/increase-grafana-refresh-rate-frequency/`.",, 2024. (accessed: 19.05.2024).

Appendix A

Python code for real-time data uploading to InfluxDB database

The Red Pitaya processes high-frequency data and transmits it to a computer for further analysis. By using a buffer to manage the data packets and performing complex mathematical operations, the system efficiently handles large volumes of data. The integration with InfluxDB allows for real-time visualization and post-processing, providing valuable insights into the impedance characteristics of tissues.

```
import time
import socket
import multiprocessing
import numpy as np
from influxdb import InfluxDBClient
from datetime import datetime

### Object class to receive the data in a new process, to avoid
   freezing the main GUI
class Receiver(multiprocessing.Process):
    def __init__(self, host, port, file, size, stop, queue,
   frequencies_val):
        super().__init__()
        self.host = host
        self.port = port
        self.file = file
        self.size = size
        self.stop = stop
        self.queue = queue
        self.frequencies_val = frequencies_val

    def run(self):

        self.client = InfluxDBClient(host='localhost', port=8086,
   username='root', password='root', database='bioimpedance')

        data_length = 0 # To count the length of the received data
```

```python
        data_frame = bytearray() # To save the first frame of the received packets

        # Create the socket object
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            # Connect to the server
            s.connect((self.host, self.port))
            # Save packet data to a file
            with open(self.file, 'wb') as f:
                # Set a timeout
                timeout = time.time() + 1*60
                sample_counter = 0
                while not self.stop.is_set():
                    # Receive a data packet
                    data = s.recv(self.size)
                    print(f"self.size = {self.size}")
                    f.write(data)
                    # Add the received bytes length to data_length
                    data_length = data_length + len(data)

                    packet_size = 16384
                    num_frames = len(data) // (52 * 8)
                    print(f"Numero de frames: {num_frames}")

                    data_frame.extend(data)
                    #data_frame.extend(data[(self.size+len(data_frame)-data_length):])

                    len_data_frame=len(data_frame)
                    print(f"La longitud del data frame es = {len_data_frame}")
                    if len(data_frame) > 52*8:
                        for frame_start in range(0, num_frames * (52 * 8), 52 * 8):
                            if sample_counter % 1 == 0:
                                # Read the a spectrum frame from the data packet (all 52 frequencies)
                                # The data format for each spectrum frame point (64 bits) is:
                                # [63:48] = 0; [47:32] = xk_index; [31:16] = data_im; [15:0] = data_re
                                data_frame = data[frame_start:frame_start + (52 * 8)]
                                read_frame = np.frombuffer(data_frame, dtype='<i2', count=52*4)

                                xk_index = read_frame[2::4]
                                frame_re = read_frame[0::4]
                                frame_im = read_frame[1::4]
```

```
                              frame = frame_re + 1j*frame_im

                              sort_samples = np.argsort(xk_index)

                              xk_index_sorted = xk_index[
sort_samples]
                              frame_sorted = frame[sort_samples]

                              frame_CH1 = [(np.conj(frame_sorted[-i
-1]) + frame_sorted[i])/2 for i in range(26)]
                              frame_CH2 = [1j*(np.conj(frame_sorted
[-i-1]) - frame_sorted[i])/2 for i in range(26)]

                              # Impedance calculation
                              z = np.divide(np.array(frame_CH2), 1e
-3*np.array(frame_CH1))

                              # The phase must be normalized
between (0, 360) degrees
                              z_mag = np.abs(z)
                              # z_phase = np.arctan2(np.imag(z), np
.real(z))
                              z_phase = np.mod(np.arctan2(np.imag(z
), np.real(z))*180/np.pi + 360, 360)

                              # Donar format i enviar dades a
InfluxDB en Influx Line Protocol
                              for i in range(len(self.
frequencies_val)):  # Asumiendo que self.frequencies_val es
accesible y contiene las frecuencias
                                  # Create point in Line Protocol
format with timestamp
                                  module_line = f"module,frequency=
{self.frequencies_val[i]} value={float(z_mag[i])}"
                                  phase_line = f"phase,frequency={
self.frequencies_val[i]} value={float(z_phase[i])}"

                                  # Send data to InfluxDB client
                                  self.client.write([module_line,
phase_line], {'db': 'bioimpedance'}, 204, 'line')
                                  print("Datos enviados a InfluxDB
para la frecuencia en cuesti n.")

                              self.queue.put([z_mag, z_phase])

                              # Reset the byte array
                              data_frame = bytearray()

```

```
                                # The new length must take into
   account the bytes over self.size
                                data_length = np.mod(data_length,
   self.size)

                            sample_counter += 1

def main():
    stop = multiprocessing.Event()
    queue = multiprocessing.Queue()

    host = "10.42.0.165"
    port = 1001
    file_name = datetime.now().strftime("%Y%m%d%H%M") + ".bin"
    pkt_size = 16384

    frequencies_val_idx = [1, 3, 5, 7, 11, 15, 19, 25, 33, 41, 51,
   63, 77, 95, 117, 143, 173, 209, 253, 307, 371, 447, 539, 649, 781,
    939]
    frequencies_val = [(125e6/8)/16384*1e-3*val for val in
   frequencies_val_idx]

    receiver = Receiver(host, port, file_name, pkt_size*8, stop,
   queue, frequencies_val)
    receiver.start()

    try:
        while True:
            time.sleep(1)
    except KeyboardInterrupt:
        stop.set()
        receiver.join()

if __name__ == '__main__':
    main()
```

Listing A.1: Python code for real-time data uploading to InfluxDB database

Appendix B

Python code for data uploading to InfluxDB old binary files for research purposes

```
#!/usr/bin/env python3

import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
from influxdb import InfluxDBClient

#Configurem InfluxDB
client = InfluxDBClient(host='localhost', port=8086, username='root',
    password='root', database='bioimpedance')

filename = "/home/josepmencion/Documents/TFG/Python/test/test121.dat"

xk_index = data[2::4] # coeficients (frequencies) de la FFT

data_re = data[0::4] # Re FFT complexa
data_im = data[1::4] # Im FFT complexa

data_cplx = data_re + 1j*data_im

total_data_frames = len(data_cplx) // 52

print("Total de mostres recollides: ")

print(total_data_frames)

# 26 frequencies
frequencies = [
    0.95367431640625, 2.86102294921875, 4.76837158203125, 6.
  67572021484375,
```

```
    10.49041748046875, 14.30511474609375, 18.11981201171875, 23.84185791015625,
    31.47125244140625, 39.10064697265625, 48.63739013671875, 60.08148193359375,
    73.43292236328125, 90.59906005859375, 111.57989501953125, 136.37542724609375,
    164.98565673828125, 199.31793212890625, 241.27960205078125, 292.77801513671875,
    353.81317138671875, 426.29241943359375, 514.0304565429688, 618.9346313476562,
    744.8196411132812, 895.5001831054688
]



# data_cplx conte FFT carregada anteriorment
for i in range(total_data_frames):

    xk_index_frame = xk_index[i*52:(i+1)*52]
    data_frame = data_cplx[i*52:(i+1)*52]

    sort_frame = np.argsort(xk_index_frame) # ordenem el frame seguint els coeficients

    xk_index_sorted = xk_index_frame[sort_frame]
    data_frame_sorted = data_frame[sort_frame]

    data_frame_re = [(np.conj(data_frame_sorted[-i-1]) + data_frame_sorted[i])/2 for i in range(26)]
    data_frame_im = [1j*(np.conj(data_frame_sorted[-i-1]) - data_frame_sorted[i])/2 for i in range(26)]

    Z = np.divide(np.array(data_frame_im), 1e-3*np.array(data_frame_re))



    P = np.mod(np.arctan2(np.imag(Z), np.real(Z))*180/np.pi + 360, 360)
    plt.figure(4); plt.plot(np.abs(P)); plt.grid();




    # data_frame actual
    data_frame = data_cplx[i*52:(i+1)*52]

    # modul i fase data_frame actual
```

```
    modulo = np.abs(np.divide(np.array(data_frame_im), 1e-3*np.array(data_frame_re)))
    fase = np.abs(np.mod(np.arctan2(np.imag(Z), np.real(Z))*180/np.pi + 360, 360))  # Fase en graus

    # Imprimim el modul i fase pel data_frame actual
    print(f"Data frame {i+1}:")
    for j in range(len(modulo)):
        # Creem punt en Line Protocol
        module_line = f"module,frequency={frequencies[j]} value={float(modulo[j])}"
        phase_line = f"phase,frequency={frequencies[j]} value={float(fase[j])}"
        client.write([module_line, phase_line], {'db': 'bioimpedance'}, 204, 'line')
        print(f"  Frecuencia {j+1}: M dulo = {modulo[j]}, Fase = {fase[j]}  ")
    print("\n---\n")



plt.show()
```

Listing B.1: Python code for data uploading to InfluxDB old binary files for research purposes

B.1 Data Reception and Buffer Management

The Red Pitaya collects data at a high sampling rate, which must be managed efficiently to avoid data loss and ensure accurate measurements.

B.1.1 Data Reception Process

The data reception process involves establishing a TCP connection to the Red Pitaya and receiving data packets of a specified size. Below is the relevant code snippet:

B.1.2 Buffer Management

The buffer management involves storing incoming data packets in a byte array and processing them once enough data has been accumulated. The buffer size and packet structure are crucial for efficient data handling.

- **Buffer Size:** The buffer size is defined to accommodate multiple frames of data. Each frame consists of 52 frequency components, with each component represented by 8 bytes (64 bits).

- **Packet Size:** Each packet received from the Red Pitaya is 16384 bytes, corresponding to the data size handled per transmission.

B.2 InfluxDB Integration and Data Storage

The processed data is formatted and sent to the InfluxDB database using the Influx Line Protocol.

B.2.1 Data Storage Format

The impedance magnitude and phase for each frequency are stored with corresponding timestamps. The data lines are formatted as follows:

```python
module_line = f"module,frequency={self.frequencies_val[i]} value={float(z_mag[i])}"
phase_line = f"phase,frequency={self.frequencies_val[i]} value={float(z_phase[i])}"
```

Listing B.2: InfluxDB Integration and Data Storage

These lines are then written to the InfluxDB client:

```python
self.client.write([module_line, phase_line], {'db': 'bioimpedance'}, 204, 'line')
```

Listing B.3: InfluxDB client data uploading

Appendix C

Frequencies for Red Pitaya Measurements

The Red Pitaya uses a decimation process to reduce its effective sampling frequency, enabling the measurement of specific frequencies of interest. By using the provided indices and applying the calculation formula, we derive the true frequencies in kHz.

C.1 Red Pitaya Sampling and Decimation

The Red Pitaya operates with a base sampling frequency of 125 MHz. To achieve different frequencies of interest for our measurements, we use a decimation process. Decimation is the process of reducing the sampling rate of a signal, effectively lowering the frequency resolution. For our measurements, we use a decimation rate of 8. This means that the effective sampling frequency after decimation is:

$$f_{\text{sample}} = \frac{125 \times 10^6}{8} = 15.625 \times 10^6\,\text{Hz}$$

C.2 Frequency Calculation Method

The frequencies of interest are determined by specific indices that correspond to the frequencies at which measurements are taken. The formula used to calculate the true frequencies from these indices is as follows:

$$f_{\text{true}} = \left(\frac{f_{\text{sample}}}{16384}\right) \times \text{index} \times 10^{-3}\,\text{kHz}$$

Where: - f_{sample} is the effective sampling frequency (15.625 MHz after decimation). - 16384 is the length of the Fast Fourier Transform (FFT) used in the processing. - The index values are pre-determined and correspond to specific frequencies of interest.

C.2.1 Indices and Corresponding Frequencies

Using the indices which can be seen on the table below, the true frequencies can be calculated as follows:

$$f_{\text{true}} = \left(\frac{15.625 \times 10^6}{16384} \right) \times \text{index} \times 10^{-3}$$

For each index, the frequency in kHz is computed. Below is a table showing the indices and their corresponding true frequencies:

Index	**Frequency (kHz)**
1	0.954
3	2.861
5	4.768
7	6.676
11	10.490
15	14.304
19	18.117
25	23.904
33	31.572
41	39.239
51	48.768
63	60.159
77	73.412
95	90.463
117	111.315
143	136.968
173	165.351
209	199.464
253	241.280
307	292.768
371	353.927
447	426.659
539	514.024
649	619.366
781	745.695
939	897.854

Table C.1: Indices and Corresponding Frequencies