Tomás Cantón Cordeiro

Development of an IoT System for Impedance Analysis of Plasma Activated Liquids in Clinical Applications

Trabajo de Fin de Grado

Directed by Dr. Jan Mitrovics & Dra. Maria Dels Àngels Moncusí Mercadé

Grado en Ingeniería Informática

UNIVERSITAT ROVIRA i VIRGILI

Tübingen - Tarragona, 2024

Abstract:
This thesis aimed to develop an embedded system for measuring the conductivity of liquids, focusing on clinical applications. The project's motivation is to accurately monitor changes in conductivity that could indicate the presence of Reactive Oxygen and Nitrogen Species (RONS).

An IoT solution was implemented using the ESP32-S3, integrating a sensor with the AD5941 for impedance measurement. Communication protocols like SPI and MQTT were employed for data acquisition and transmission. The infrastructure included a Raspberry Pi as a server, enabling real-time data visualization with Docker, Node-RED, and InfluxDB.

The results demonstrated the system's ability to detect conductivity variations accurately, establishing the project's core and confirming the integration with the other EndoPAL project's components.

Resumen:
El objetivo de esta tesis fue desarrollar un sistema embebido para medir la conductividad de líquidos, enfocado en aplicaciones clínicas. La motivación radica en la necesidad de monitorear cambios en la conductividad que puedan indicar la presencia de RONS.

Se implementó una solución IoT con la ESP32-S3, integrando un sensor con el AD5941 para medir la impedancia. Se utilizaron protocolos de comunicación como SPI y MQTT para la adquisición y transmisión de datos. La infraestructura incluyó una Raspberry Pi como servidor, permitiendo la visualización de datos en tiempo real mediante Docker, Node-RED e InfluxDB.

Los resultados mostraron que el sistema puede detectar variaciones en la conductividad con precisión, estableciendo la base del proyecto y confirmando su integración con los demás componentes del proyecto EndoPAL.

Resum:
L'objectiu d'aquesta tesi va ser desenvolupar un sistema incrustat per mesurar la conductivitat de líquids en aplicacions clíniques. La motivació es basa en la necessitat de monitoritzar canvis en la conductivitat que puguin indicar la presència de RONS.

Es va implementar una solució IoT amb l'ESP32-S3, integrant un sensor amb l'AD5941 per a la mesura de la impedància. Es van utilitzar protocols com SPI i MQTT per a l'adquisició i transmissió de dades. La infraestructura inclou una Raspberry Pi com a servidor, facilitant la visualització de dades en temps real amb Docker, Node-RED i InfluxDB.

Els resultats van demostrar que el sistema és capaç de detectar variacions en la conductivitat amb precisió, establint la base del projecte i confirmant la seva integració amb els components del projecte EndoPAL.

Abbreviations

CAP	Cold Atmospheric Plasma
ADC	Analog-to-Digital Converter
AFE	Analog Front End
AIN	Analog Input
CE0	Counter Electrode 0
SE0	Sense Electrode 0
RE0	Reference Electrode 0
CPU	Central Processing Unit
EMI	Electromagnetic Interference
ESP	Espressif Systems Processor
ESP-IDF	ESPressif IoT Development Framework
GPIO	General Purpose Input/Output
IoT	Internet of Things
JLM	JLM Innovation GmbH
MQTT	Message Queuing Telemetry Transport
OEM	Original Equipment Manufacturer
PAL	Plasma-Activated Liquids
PAW	Plasma-Activated Water
PCB	Printed Circuit Board
PT1000	Platinum Resistance Temperature Detector 1000 Ohm
QoS	Quality of Service
RONS	Reactive Oxygen and Nitrogen Species
SOC	System on Chip
SPI	Serial Peripheral Interface
SSL/TLS	Secure Sockets Layer / Transport Layer Security
UART	Universal Asynchronous Receiver-Transmitter
UV	Ultraviolet
AC	Alternating Current
ECM	Extracellular Matrix
InfluxQL	Influx Query Language

List of Figures

List of Tables

1. Introduction

1.1 Company introduction

JLM Innovation GmbH is a technology company located in Tübingen, Germany. It specializes in providing services and products related to the development and validation of embedded electronic components. With over a decade of experience in the field, the company has established a strong reputation in the industry.

Dr. Jan Mitrovics is the founder and CEO of JLM, with more than 25 years of experience in developing sensor systems for various applications. Jan Mitrovics is a physicist and obtained his Ph.D. in Physical Chemistry with a focus on chemical sensor systems.

Since 2004, JLM Innovation has been producing sensor systems for research and development projects. Additionally, JLM Innovation develops sensor solutions and optimizations for the commercial market. The company offers a wide range of solutions for various sensor technologies, allowing the development of new products quickly and efficiently. JLM Innovation is a partner in several research projects, primarily in the fields of information technology, engineering, and biotechnology, where it conducts cutting-edge projects in collaboration with renowned global OEM companies and Tier-1 enterprises.

One of the main focuses of the research projects carried out by JLM Innovation is respiratory gas analysis systems for medical diagnosis. JLM Innovation has developed several sensor platforms that are used in clinical studies to evaluate new technologies. The company covers all development stages, from the sensor to the final system, and has capabilities for electronic and software development, board design, 3D printing, and manufacturing.

Figure 1.1: Logo of JLM Innovation GmbH.

1.2 Project context

The EndoPAL-i project is a collaborative initiative in the fields of biotechnology and computer engineering, aimed at developing an innovative system for the generation and application of PAL. The development of this project is carried out in collaboration with various companies and research institutions. The multidisciplinary team addresses everything from plasma generation to its clinical evaluation and applications in the industrial sector.

Clinical challenges: Impact of postoperative adhesions

The central clinical problem addressed by this project is the presence of postoperative adhesions in the abdominal area, which develop as a result of surgical procedures. These adhesions are connective tissue structures that limit the mobility of abdominal organs, potentially causing discomfort, significant complications, and various severe conditions. Worldwide, the number of abdominal surgical procedures continues to increase. In 2018 alone, approximately 2.6 million abdominal surgeries were performed in Germany, illustrating the magnitude of the problem.

The likelihood of developing postoperative adhesions, estimated at between 67% and 93%, means that in Germany, between 1.75 and 2.4 million new cases of adhesions occur annually. This not only has a significant impact on patients' health but also entails considerable costs for the healthcare system and society in general.

Postoperative adhesions are often associated with chronic pain symptoms that are intense and difficult to localize in affected patients. Furthermore, they are responsible for 15% to 20% of secondary infertility cases, as well as 50% to 70% of mechanical bowel obstructions, which often present with severe clinical symptoms. Diagnosing postoperative adhesions is usually challenging, as they are not typically detected through conventional imaging techniques and are often misdiagnosed for years, leading to a significant number of medical consultations and costly diagnostics. [7]

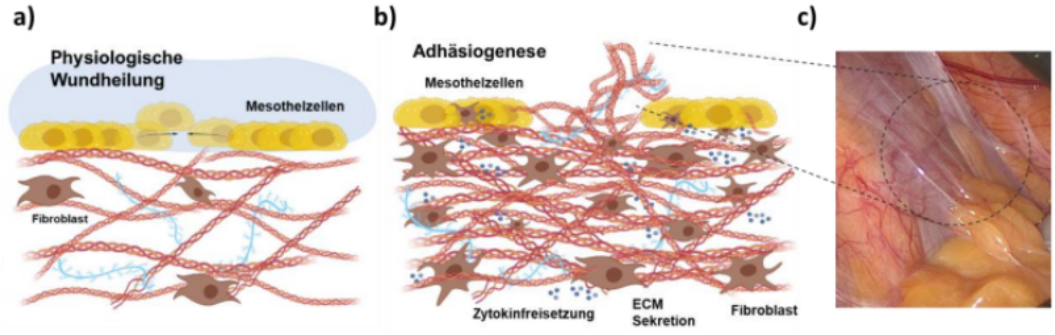

Figure 1.2: Illustration of postoperative adhesions in the abdominal area. (a) Physiological wound healing showing fibroblasts cells. (b) Development of adhesions, leading to formation of connective tissue that limit mobility. (c) Clinical image showing postoperative adhesions.

Industrial challenges - Innovations in water disinfection

In the industrial field, the main problem addressed by this project is related to water disinfection and the prevention of infections caused by Legionella through contaminated water aerosols. The proliferation of Legionella in showers, pools, air conditioning systems, and cooling towers remains a significant challenge in terms of hygiene. Currently, decontaminating water systems requires a considerable amount of energy and time, sometimes involving the use of polluting methods. This process poses a significant concern in terms of water safety and quality. Therefore, the EndoPAL-i project aims to provide an effective, safe, and sterile solution for water disinfection in industrial facilities, including homes and cooling systems. The need for effective and sustainable water disinfection is evident in various industrial and residential sectors.[7]

1.3 Introduction to EndoPAL-i

The EndoPAL-i project is an innovative initiative focused on developing medical and disinfection applications using a special type of plasma, Cold Atmospheric Plasma (CAP) [8].

Plasma is one of the four fundamental states of matter, characterized by containing electrically charged particles such as electrons and ions, instead of neutral particles. Plasma is a highly energetic state of matter typically found in high-temperature conditions, such as the interior of stars and nuclear fusion processes. In the EndoPAL-i project, CAP is used, which is generated at temperatures close to room temperature.

Unlike hot plasmas, CAP does not damage living tissues or surfaces with which it comes into contact, making it suitable for medical and disinfection applications. CAP is a mixture of charged particles and radicals that can have antimicrobial properties and other beneficial applications.

Among the particles formed in CAP are Reactive Oxygen and Nitrogen Species (RONS) such as hydrogen peroxide (H2O2), nitric oxide (NO), peroxynitrite (ONOO-), superoxide (O2-), hydroxyl radical (OH), nitrogen dioxide (NO2), nitrate (NO3-), among others. These RONS are highly reactive molecules that play a crucial role in medical and disinfection applications. H2O2, NO, and ONOO- are some of the most frequent and useful RONS in the EndoPAL-i project, as they can have antimicrobial effects and be beneficial for the prophylaxis of postoperative adhesions and water disinfection [8].

1.4 Objectives of the EndoPAL-i project

1.4.1 General collaborative objective

The objective of the EndoPAL-i project is to develop an innovative system for generating plasma-activated liquids with two fundamental applications. First, it aims to provide a routine application that addresses the problem of postoperative adhesions. Second, it seeks to contribute to water disinfection in industrial applications, addressing the problem of Legionella infection through aerosols in showers, pools, air conditioning systems, and cooling towers. This collaborative project involves various companies, among which the following stand out:

- **ElringKlinger GmbH:** Develops a system of interchangeable separation membrane cartridges with optimized properties.
- **Plasmatreat GmbH:** Designs a combined plasma reactor for the optimal generation of RONS.
- **JLM Innovation GmbH:** Works on developing sensors for process control in PAL generation and integrates these sensors into the overall systems.
- **BlueLab GmbH:** Develops systems for injecting PAL into drinking and industrial water systems and evaluates its disinfection properties.

1.4.2 JLM Innovations objective and personal contribution

The objective of JLM Innovation GmbH focuses on developing technologies for the detection of RONS in PAL, as well as creating test systems to evaluate the correct behavior of these processes. JLM contributes to the control and monitoring of the EndoPAL project.

My primary focus has been the development and implementation of an IoT system by combining an ESP32-S3 microcontroller and an AD5941 sensor to measure the impedance of activated liquids and calculate their conductivity. To achieve this, I have assembled the necessary hardware, developed the firmware to perform the measurements, and implemented a series of applications and protocols to ensure a cohesive, scalable, and integrable system within the EndoPAL project.

Additionally, I have developed a specialized chamber that allows the impedance sensor to come into direct contact with the PAL flow. This integration is essential for monitoring and calculating the conductivity of the liquid in real-time as it is activated, ensuring that RONS are being correctly generated for both clinical and industrial applications.

2. Background

2.1 Hardware components

2.1.1 ESP32

The ESP32 is a System on Chip (SoC) designed by EspressIF Systems [3], featuring a 32-bit microcontroller with integrated WiFi and Bluetooth connectivity. This microcontroller is widely used in IoT projects and facilitates the development of embedded electronic applications due to its processing power, low energy consumption, connectivity capabilities, and the ability to integrate various peripherals.

XIAO ESP32-S3

The XIAO ESP32-S3 is a specific variant of the ESP32 developed by Seeed Studio, a hardware and IoT solutions manufacturer. The ESP32-S3 includes a high-performance dual-core Xtensa LX7 microcontroller, providing powerful processing capabilities for complex applications. Additionally, the Seeed Studio SoC comes with 8MB of flash memory, which serves as non-volatile storage for application firmware.

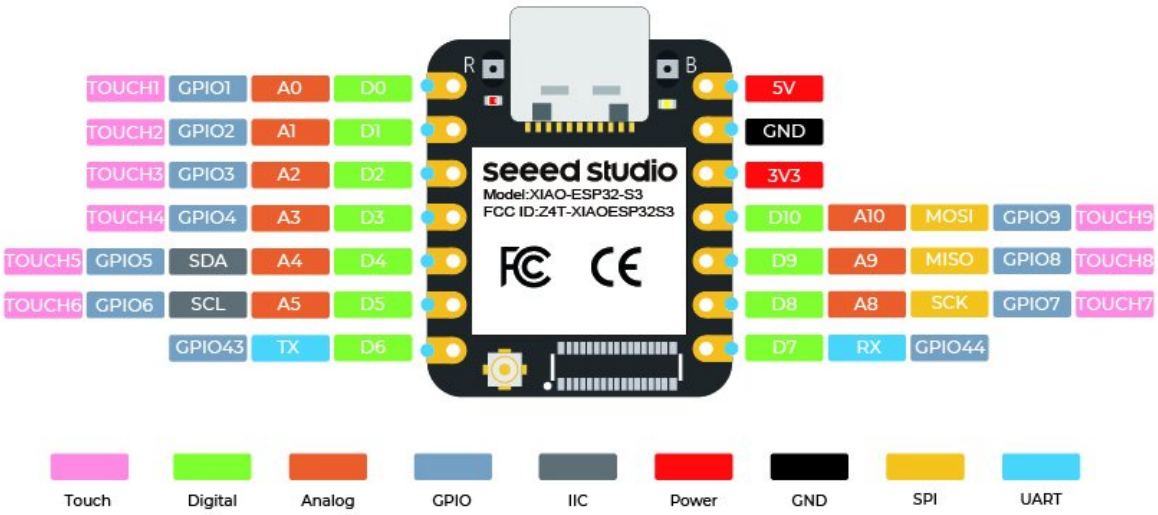

Figure 2.1: Pinout diagram of the SOC ESP32-S3.

This microcontroller offers a wide range of GPIO pins that can be configured to interact with sensors and other electronic components. The ESP32-S3 also features standard communication interfaces such as UART and SPI, which facilitate the connection and communication between devices. The UART protocol establishes a serial communication link between the computer and the ESP32 via a USB port, commonly used for application debugging and flashing the SoC's firmware without requiring an external programmer. [3]

The SPI protocol, which is employed to connect the ESP32-S3 with the AD5941 in this embedded system, operates on a master-slave architecture and allows synchronous binary data transfer between devices. In addition to SPI, the ESP32-S3 supports other interfaces like I2C and ADC, enabling it to connect with a wide variety of sensors, making it highly versatile for IoT applications. [3]

2.1.2 AD5941

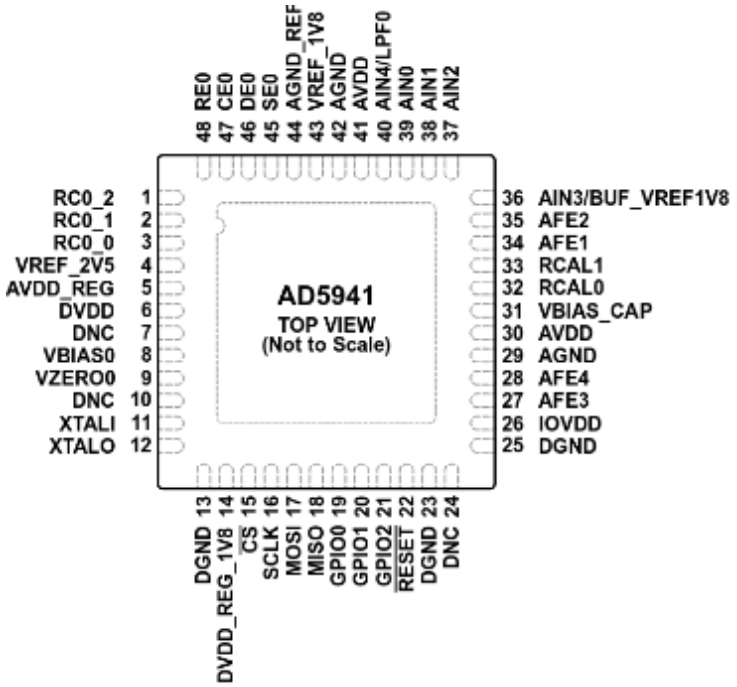

Figure 2.2: Pin configuration of the AD5941 AFE.

The AD5940/AD5941, manufactured by Analog Devices, is a low-power, high-precision electrochemical analog front end (AFE) designed for portable applications requiring highly accurate measurement techniques. These devices are capable of performing voltage, temperature, and impedance measurements, among others. The AFE consists of two high-precision excitation loops and a common measurement channel, allowing for a wide range of sensor measurements under various conditions.

The measurement channel of the AD5941 incorporates a 16-bit ADC. Its primary function is to convert analog signals from sensors and electrochemical devices into digital data that can be processed and monitored by a computer system. The ADC operates within a voltage range of 1.8V to 3.6V and communicates with an ESP32-S3 microcontroller via the SPI protocol. Additionally, it includes an input multiplexer in front of the ADC, allowing the selection of the input channel for measurement. [1]

2.1.3 Conductivity sensor

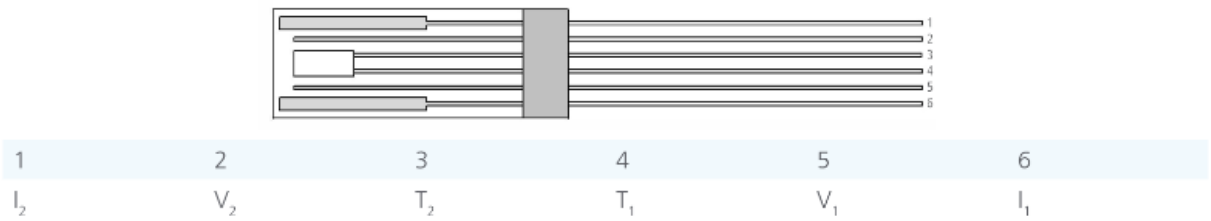

Figure 2.3: Pinout diagram of the conductivity sensor. Dimensions: 12.9 mm (L) x 5.5 mm (W) x 0.65 mm (H).

The LFS1K0.1305.6W.C.010-6 conductivity sensor [4] is designed to measure the electrical conductivity of liquids with high precision and stability. The conductivity of a liquid changes with the temperature and the ion concentration within the liquid. This sensor operates over a wide range of conductivity (from 100 µS/cm to 200 mS/cm) and temperatures (from -30°C to +100°C), making it suitable for diverse applications. It uses a four-electrode configuration, where an alternating current (AC) is applied between pins 1 and 6, and the resulting voltage is measured across pins 2 and 5, minimizing polarization effects and ensuring accurate conductivity readings. The integrated Pt1000 temperature sensor is connected to pins 3 and 4.

2.2 Software components

2.2.1 ESPressif IoT Development Framework

The ESP-IDF is a software development environment provided to assist developers using the ESP32 microcontroller in creating IoT applications. This environment includes the necessary tools to build and compile code, it also provides an API with software libraries [2] and source code for the SoC. The environment simplifies the processes of device configuration, code compilation, and flashing the ESP32-S3 memory as follows. [3]

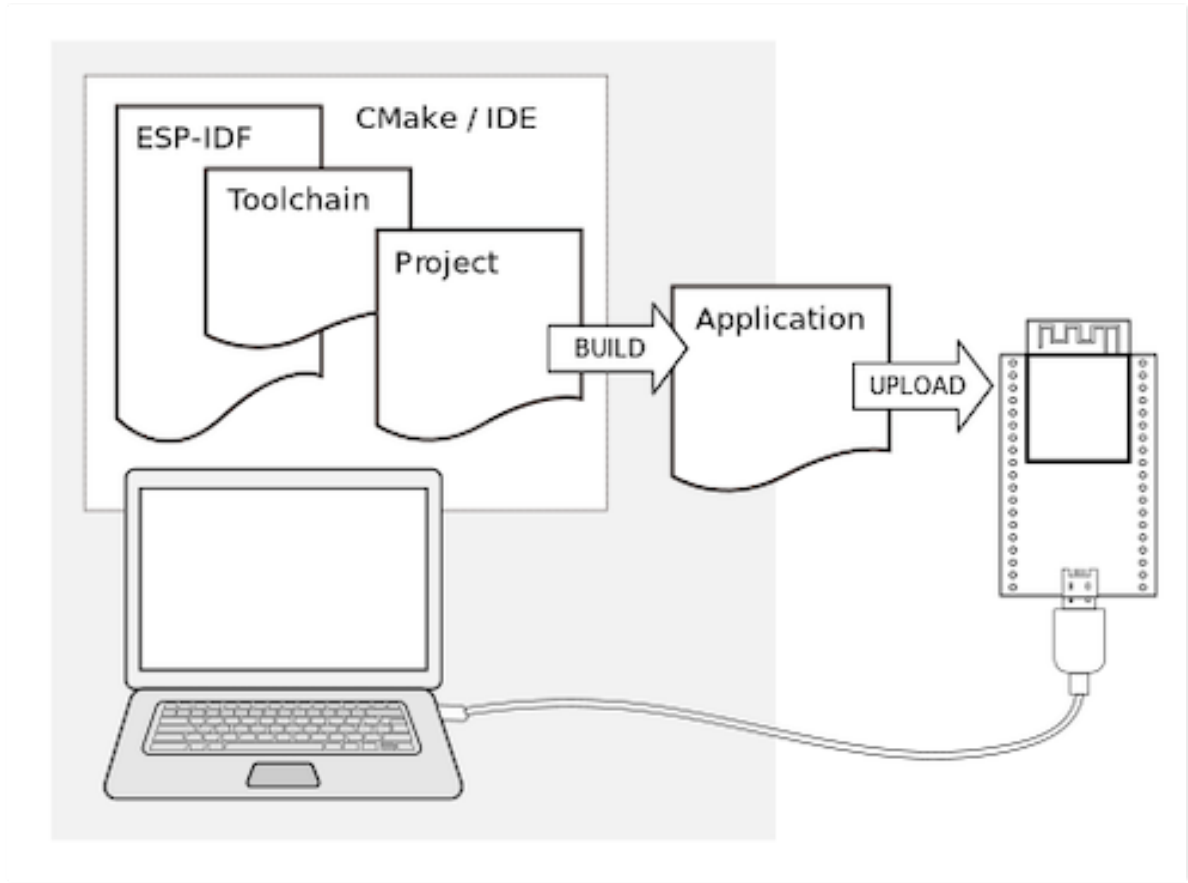

Figure 2.4: ESP-IDF Workflow Diagram.

2.2.2 IoT Stack

An IoT Stack is a collection of applications and services designed to simplify and accelerate the development of IoT projects. This stack is deployed via Docker containers to facilitate the implementation and management of different services. These applications enable efficient deployment and simplify the communication, processing, storage, and real-time visualization of PAL measurements.

The IoT Stack allows developers to focus on application logic and generating value from the collected data. The following services are used to build the IoT application that has been developed.

Mosquitto

Mosquitto is an open-source MQTT (Message Queuing Telemetry Transport) broker that acts as an efficient and secure communication channel between the ESP32-S3 SoC and the IoT server.

Node-RED

Node-RED is a visual development environment that allows the creation of custom workflows to process, transform, and automate the data communicated from the IoT server to the InfluxDB database.

InfluxDB

InfluxDB is a time-series database that stores the measurements performed by the SoC based on time. It uses the InfluxQL language, which allows queries to retrieve the data. It is also noted for its high-speed continuous data flow control, enabling real-time tracking and historical analysis of measurements.

Portainer

Portainer is a tool that facilitates the administration, management, and maintenance of Docker containers used in the execution of the implemented IoT system, ensuring stable and efficient operation of all components.

2.3 Project phases and timeline

The Gantt diagram presented in Figure 2.5 illustrates the timeline and sequence of tasks involved in the development of the IoT system for impedance analysis of PALs. The diagram serves as a visual representation of the project's workflow, highlighting key phases and the duration of each task.

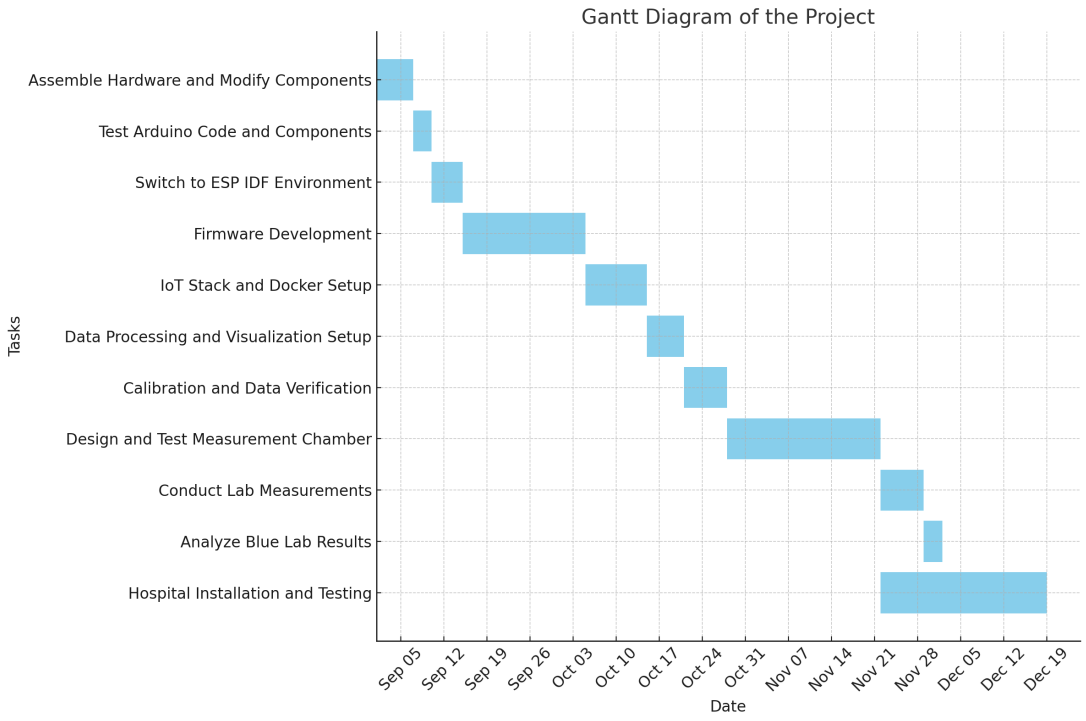

Figure 2.5: Gantt Diagram of the Project

The development phase that required the most time was the firmware development. This phase extended beyond the initial estimated duration due to the complexity involved in ensuring that the firmware could accurately perform all the required measurements.

The design and testing of the measurement chamber also took longer than expected. The initial design of the chamber encountered leakage, which led to the introduction of air bubbles in the system that compromised the accuracy of the measurements, needing a redesign.

The installation of the IoT system at the hospital was another phase that faced delays. The clinical environment presented challenges due to EMI from other medical devices, causing intermittent loss of data transmission and periods where data was not received. This issue was resolved by incorporating a Faraday cage, which successfully isolated the system from external interferences, ensuring reliable acquisition and transmission of the data.

3. System integration and configuration

3.1 Hardware assembly and modifications

During the assembly process of the embedded system, the main components are integrated: the ESP32-S3, the AD5941, and the WiFi antenna (Figure 3.1).

The ESP32-S3 acts as the CPU and is soldered onto the PCB, ensuring a solid connection for the necessary pins. To enable wireless connectivity, a WiFi antenna is connected, which is coupled to the ESP32-S3 using a specific connector.

The AD5941 is a high-precision ADC that is assembled on the PCB and connected to the ESP32-S3 via the SPI protocol, allowing for the acquisition and communication of analog signals.

To optimize the functionality of the PCB in this project, several key modifications were made. A shortcut was added between C22 (a capacitor on the PCB) and CLPF (a low-pass filter) to deliver the reference voltage of 1.82 V to pin AIN4 in a more accessible and precise manner for the circuit design. Additionally, another shortcut was implemented between the RE0 (Reference Electrode) and CE0 (Counter Electrode) pins to facilitate impedance measurement in liquids. Finally, a 1000 Ohm resistor was added between GND and AIN0 to be used as the known resistance (R known) for temperature measurement, ensuring accurate readings (Figure 3.2).

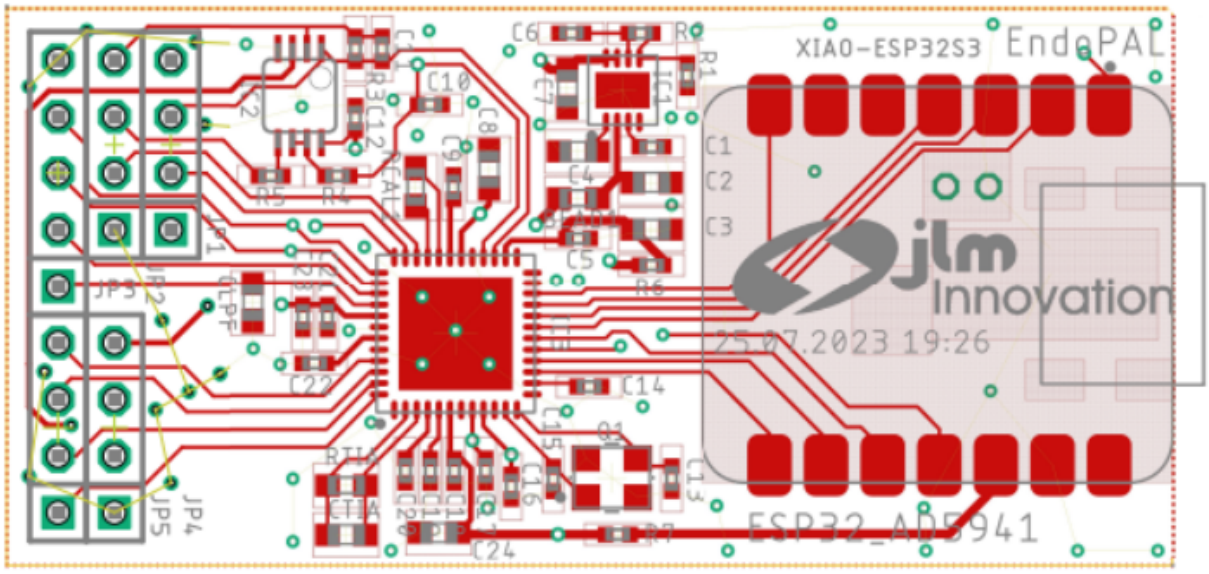

Figure 3.1: PCB with ESP32-S3 and AD5941 Populated. The ESP32-S3 is located on the right side, while the AD5941 is positioned centrally.

3.2 Firmware configuration and communication

After the hardware assembly, the next step involved setting up communication between the ESP32-S3 and the AD5941 using the SPI (Serial Peripheral Interface) protocol. In this setup, the ESP32-S3 operates as the master, managing data transfer, while the AD5941 functions as the slave, supplying sensor measurements.

Four GPIO pins were configured through the firmware to handle the SPI:

- **MOSI (GPIO 9)**: Sends data from the master (ESP32-S3) to the slave (AD5941).
- **MISO (GPIO 8)**: Receives data from the slave to the master.
- **SCK (GPIO 7)**: Synchronizes data transfer on the falling edge.
- **CS (GPIO 6)**: A dummy GPIO used for chip selection, managed by the AD5941 library. It is defined to fulfill the required structure but does not perform any function.

Following this, the ESP32-S3 firmware was developed using the Espressif IoT Development Framework. The firmware coded in C was designed to iteratively perform impedance measurements across different frequencies, along with temperature and voltage readings. The UART protocol served as the bridge for transferring the compiled firmware via USB from the computer to the ESP32-S3's flash memory.

With the firmware loaded onto the ESP32-S3, the sensor connections were established on the PCB to facilitate the sensor data acquisition through the AD5941.

- **Temperature Measurement**: Used the AIN4 and AIN0 pins, connected to the T1 and T2 pins of the sensor, to measure resistance changes that correspond to temperature variations.
- **Voltage Measurement**: Employed the AIN1 and AIN3 pins, linked to the V1 and V2 pins of the sensor, to capture voltage variations.
- **Impedance Measurement**: The RE0 and CE0 pins were connected to the I1 pin, while SE0 was connected to the I2 pin of the sensor to establish the impedance measurement circuit.

These configurations ensured that the ESP32-S3 could effectively receive and process the data from the AD5941.

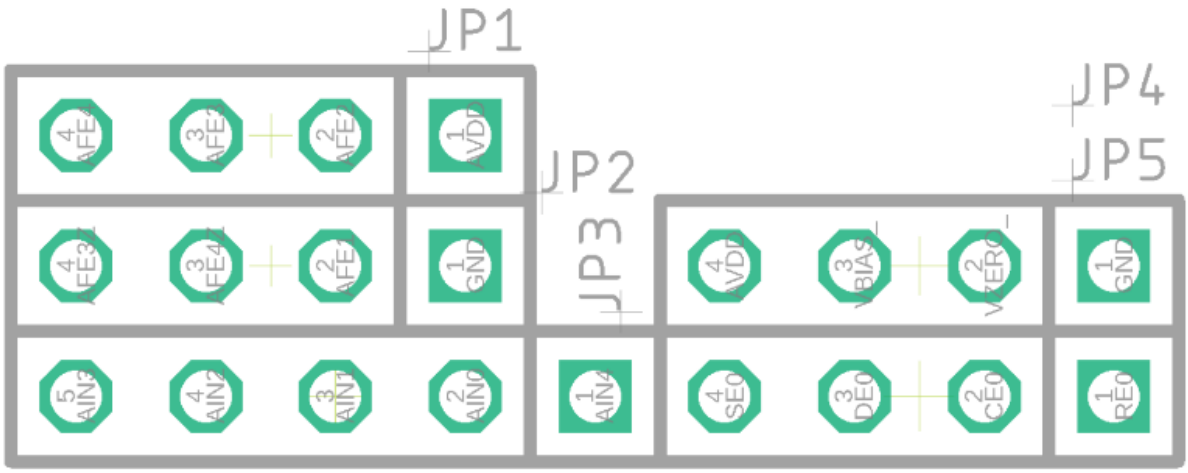

Figure 3.2: PCB Sensor Connection Pins. The figure illustrates the pinholes on the PCB where the sensor connections are made. ADC (AIN1 and AIN3), temperature (AIN4 and AIN0), and impedance (RE0, CE0, SE0).

Up to this point, the theoretical aspects and configurations of the sensor and communication protocols have been discussed. In terms of firmware execution, the main function begins with the initialization of the WiFi, the configuration of the GPIOs for SPI communication, and the initialization of the MQTT client for data transmission (see Appendix A.2). Once these steps are completed, the main routine (`AD5940_Main`), designed to manage impedance, temperature, and voltage measurements, is invoked (see Appendix A.3).

Within this routine, the AD5940 platform is configured to perform the necessary measurements. Interrupts are managed using flags that determine when to perform impedance or temperature/voltage measurements, adjusting the AD5940's configuration accordingly. As measurements are taken, the results are processed and transmitted in real-time via MQTT messages for further analysis (see Appendix A.4).

After successfully collecting data from the sensors, the next phase involved transmitting this information to a computer for analysis and visualization. This was achieved using the MQTT protocol, which operates on a publisher/subscriber model. In this configuration, the ESP32-S3 functions as the publisher, sending data to the MQTT broker hosted on a Raspberry Pi.

The Raspberry Pi serves as the central server, hosting a Docker container that includes the essential project applications: the MQTT broker Mosquitto, Node-RED, and InfluxDB. The ESP32-S3 publishes sensor data over the Wi-Fi network using the Raspberry Pi's IP address. Meanwhile, the computer connects to the same IP address as a subscriber, accessing the data managed by Mosquitto for further analysis and visualization.

During the project's development, MQTT 5 was chosen over MQTT 3 due to its enhanced scalability, which allows for a greater number of connected devices—an advantage that could be crucial in future project expansions. Additionally, QoS 0 (Quality of Service) was selected for

data transmission, prioritizing speed and efficiency. Given the continuous data transmission, the occasional loss of a message does not significantly impact the overall information flow, making QoS 0 the most suitable option. MQTT also offers security features, including support for TLS (Transport Layer Security) for encrypted communication, and authentication mechanisms, ensuring that data transmission remains secure.

3.3 Data processing and visualization

Once the data is transmitted to the MQTT broker, Node-RED is used to process and control the flow of the information before storing it in InfluxDB. Node-RED acts as a visual intermediary, facilitating control over the workflow and ensuring the data is formatted correctly for storage.(Figure 3.3)

The workflow consists of several key steps:

1. **Data Reception**: MQTT sends a message for each topic—Impedance, Temperature, and ADC—accompanied by the corresponding measurement data.
2. **Data Formatting**: The message is received by an input node in Node-RED and passes through a ”format measurement” node where it is structured into topics, variables, and values.
3. **Data Conversion**: The formatted data is then converted into JSON format.
4. **Database Insertion**: Finally, the JSON data is sent to the JLM InfluxDB database for storage.

This workflow ensures that the data received is well-structured before it is stored in the database.

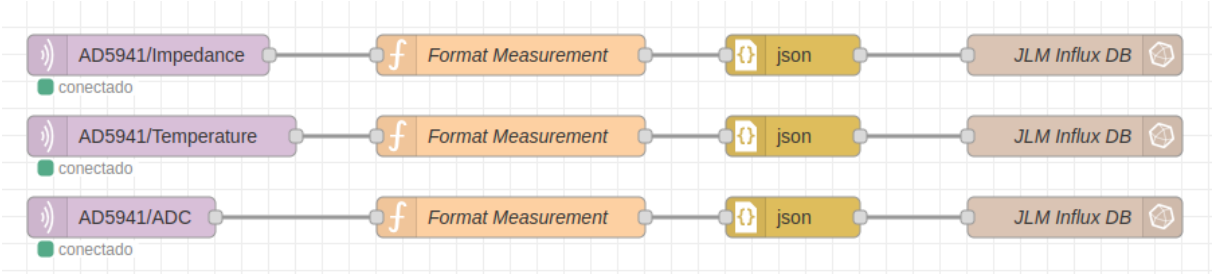

Figure 3.3: NodeRed Workflow Diagram.

InfluxDB was chosen for data storage due to its optimization for time-series data management, which is ideal for the continuous data streams generated in this project. The use of InfluxDB allows for real-time data storage and visualization, facilitating immediate analysis and error detection.

InfluxDB 2.0 was selected over version 1.x because the newer version integrates database functionality with built-in visualization tools. This enables the creation of dashboards, alerts, and data analysis within a single environment, eliminating the need for additional tools like Grafana.

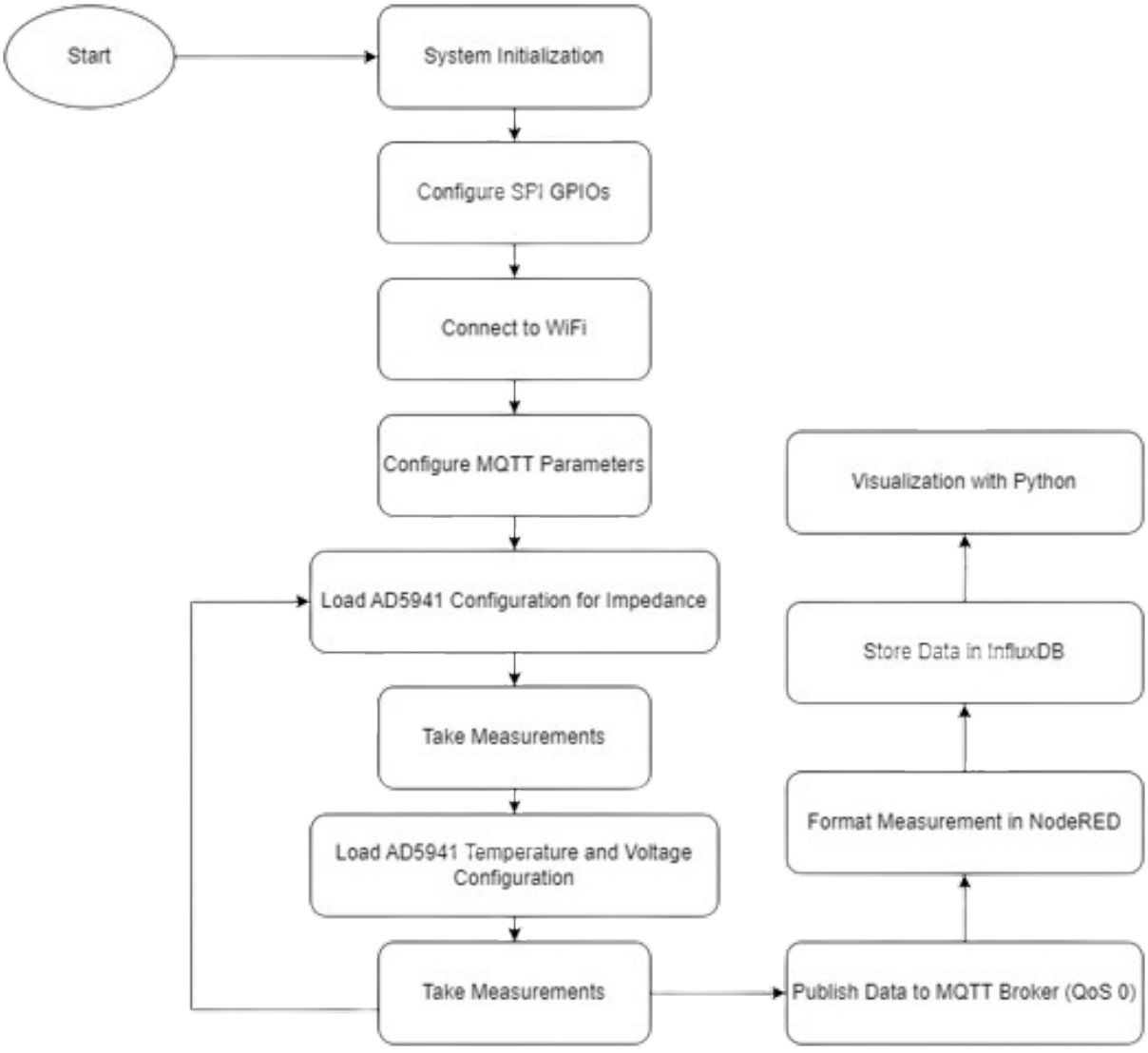

Figure 3.4: Workflow for the IoT system detailing the sequential process of initialization, measurement, and data management for PAL.

The workflow depicted in Figure 3.4 outlines the sequential process involved in the operation of the IoT system developed for measurements of PAL. It highlights the continuous cycle of data acquisition, processing, and storage.

3.4 Chamber design

Design of the first chamber

A flow chamber was developed for the measurement of conductivity in Plasma Activated Liquids (PAL). The chamber was constructed using teflon due to its resistance, high temperature tolerance, and low friction with liquids. The design was crafted in Autodesk Fusion360, a software that allows modeling of complex components. The chamber has in and out holes through which the liquid flows, as well as a top aperture designed to locate a 3D-printed component that holds the sensor in place, ensuring the direct contact with the liquid. The production was executed using a CNC (Computer Numerical Control) machine, which automated the manufacturing process and accurately translated the digital design into the teflon block.

Despite these considerations, this version of the chamber had leakage issues due to the material properties, particularly the wear and tear of the teflon over time. This wear prevented the system from achieve a proper seal, which allowed air bubbles to enter, compromising the accuracy of the measurements and the functionality of the system. After testing the chamber, the results were inconsistent, with significant variations in the data. By inserting a piece of paper between the teflon chamber and the screws, the presence of leaks was identified, which eventually turned into a drip due to the material's degradation. These challenges led to the development of an optimized chamber that ensures a more secure and isolated environment.

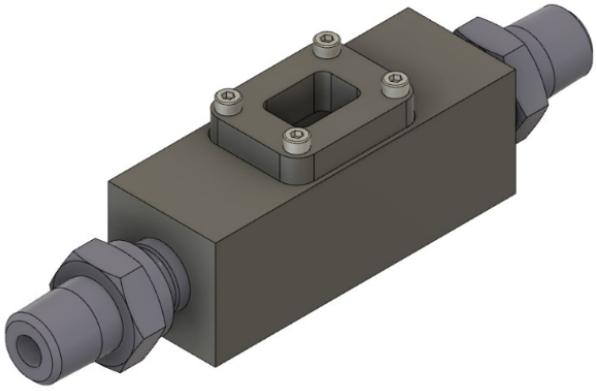

Figure 3.5: Design of the flow chamber for PAL conductivity measurement. Dimensions: 60 mm (x), 22 mm (y), 26 mm (z).

Final design of the chamber

To address the issues encountered in the initial design, a final version of the flow chamber was developed using 3D printing with resin, resulting in a more compact and robust structure. This improved design focuses on enhancing the chamber's durability and sealing capabilities. The chamber now includes a sealing mechanism with adjustable rings that, when tightened, create a hermetic seal, effectively preventing liquid leaks and ensuring the integrity of the system.

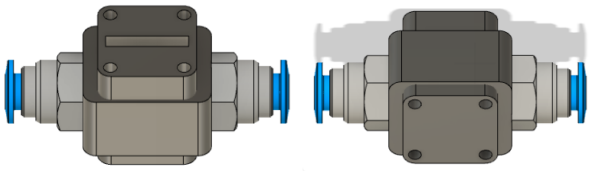

Figure 3.6: Design of the final chamber for PAL conductivity measurement. Left: chamber top view; Right: bottom view. Dimensions: 24 mm (x), 20 mm (y), 24 mm (z).

4. System testing

4.1 ADC polling validation

The ADC polling [2] involves the microcontroller periodically checking the ADC value to convert an analog signal into a digital one. In the context of this project, the ADC is crucial for continuously monitoring changes in the properties of the liquids, such as conductivity, temperature and, in the future, pH. A potentiometer is used to manually adjust the input voltage to the analog inputs, AIN1 and AIN3 of the PCB.

The ADC Code is the digital value generated by the ADC conversion, represented in bits. The maximum value for a 16-bit resolution is 65535, the closer the value is, the closer the input voltage is to the V_{ref}. However, in a differential ADC setup, where the voltage is measured as the difference between two inputs (AIN+ and AIN-), the midpoint of the range corresponds to 0 V. Values above represent positive voltages, while values below represent negative values. 4.1.

$$\text{ADC Voltage} = \left(\frac{\text{ADC Code}}{2^{n-1}} - 1 \right) \times V_{\text{ref}} \tag{4.1}$$

where $n = 16$ is the ADC resolution, and $V_{\text{ref}} = 1.82\,\text{V}$ is the reference voltage.

Table 4.1: ADC Code and Voltage comparison.

AIN3 - AIN0	Voltage (V)	AIN0 - AIN3	Voltage (V)
65268	1.82	759	-1.7925
62328	1.6554	1001	-1.7789
60492	1.5525	4360	-1.5908
57988	1.4123	7452	-1.4177
56939	1.3536	10553	-1.244
53417	1.1563	13704	-1.0676
50039	0.9672	15608	-0.961
49283	0.9248	18205	-0.8155
46486	0.7682	19513	-0.7423
44708	0.6686	20278	-0.6994
41972	0.5154	22245	-0.5893
40514	0.4338	24154	-0.4824
34931	0.1211	29505	-0.1827
32759	-0.0005	30536	-0.125

These results show how the ADC Code varies with the input voltage adjusted by the potentiometer. As the input voltage changes, the ADC Code rises and decreases accordingly to the reference voltage, confirming the correct ADC behavior.

4.2 Temperature testing and equations

The PT1000 temperature sensor is widely used due to its precise response to temperature changes. It operates by increasing its resistance as the temperature rises, making it a reliable component for accurate measurements [5].

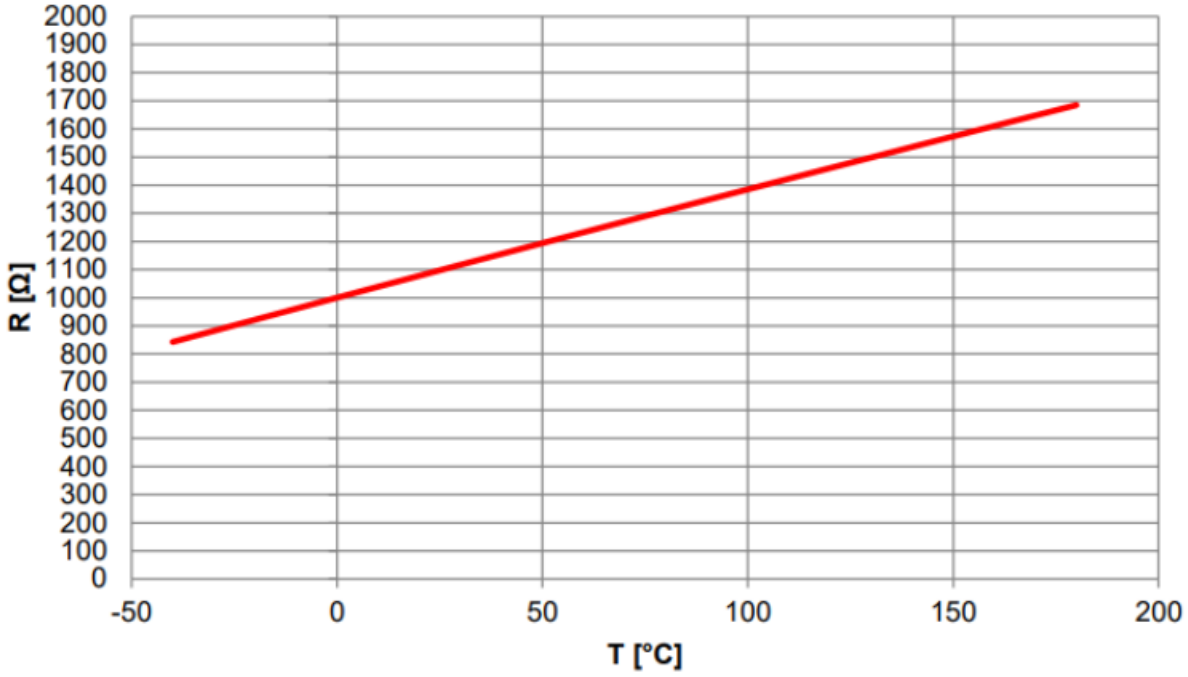

Figure 4.1: Relationship between Resistance (R) in Ohms and Temperature (T) in °C. The graph shows a nearly linear increase in resistance with temperature, where 0°C corresponds to 1000 Ω.

The graph 4.1 illustrates the relationship between resistance and temperature for the PT1000 sensor; it can be described by the Callendar-Van Dusen equation (4.2), which is generally used for precise temperature measurements. [5]

$$\mathrm{R}(t) = R_0 \times (1 + A \times t + B \times t^2 + C \times (t - 100) \times t^3) \tag{4.2}$$

where R_0 is the resistance at 0°C, t is the temperature in °C, and A, B, and C are constants:

$$A = 3.9083 \times 10^{-3}\,^{\circ}\mathrm{C}^{-1}, \quad B = -5.775 \times 10^{-7}\,^{\circ}\mathrm{C}^{-2}, \quad C = -4.183 \times 10^{-12}\,^{\circ}\mathrm{C}^{-4}$$

For temperatures above 0°C, the C term can be neglected, simplifying the equation to:

$$\mathrm{R}(t) = R_0 \times (1 + A \times t + B \times t^2) \tag{4.3}$$

Furthermore, the relationship between resistance and temperature can be approximated as nearly linear, which simplifies the equation to:

$$\begin{aligned} R(T) &= 3.9083 \times T + R_k \times 1000, \\ T &= \frac{R(T) - 1000}{3.9083} \end{aligned} \tag{4.4}$$

This linear approximation is enough for the purposes of this project, as the goal is to verify that temperature increases with conductivity, confirming the expected system behavior.

Resistance calculation in the circuit

To measure the resistance of the PT1000 sensor in the circuit, a voltage divider configuration is used. The resistance R_1 can be calculated using the formula:

$$R_1 = \frac{V_{\text{ref}} - V_2}{\frac{V_2}{R_2}} \tag{4.5}$$

This formula is derived from Ohm's Law and the principles of voltage division. It allows the calculation of the unknown resistance R_1 based on the reference voltage V_{ref}, the measured voltage V_2, and the known resistance R_2. The temperature T corresponding to a given resistance $R(T)$ can be calculated from the linear relationship shown in equation 4.4.

During the temperature measurement process, the following variables are captured:

- **ADCCode**: The raw digital output from the ADC in the AD5941 after sampling the voltage across the PT1000 sensor.
- **ADCVolt**: The measured voltage across the PT1000 sensor, derived from the ADCCode.
- **Unknown Resistance**: The resistance of the PT1000 sensor, calculated using the voltage divider formula.
- **Temperature (T)**: The temperature is calculated using the relationship referenced above, where $R(T)$ is the measured resistance, 1000 Ω is the resistance at 0°C, and 3.9083 Ω/C is the temperature coefficient (A).

Table 4.2: Relationship between resistance and temperature for different measured values. Ambient temperature measured at 22.1°C.

Resistance R_1 (Ohms)	**Measured R_1 (Ohms)**	**Temperature (°C)**
1000	1005.00	1.28
1100	1090.00	23.03
1222	1226.93	58.64
1440	1428.43	110.7
1550	1537.26	138.82
1660	1647.42	167.34
2000	1983.81	254.21
Ambient Temp.	1082.78	21.49

The equations (4.4, 4.5) simplifies the calculation of temperature from the measured resistance. Given the nearly linear behavior of the sensor, these equations provide a direct method for converting resistance readings into temperature values, since the linear approximation is enough for our purposes.

The table 4.2 shows the relationship between resistance and temperature using the PT1000 sensor. The error in the data could be due to the sensor's natural tolerances, slight imprecisions in the used resistors, and the application of the simplified linear equation. However, this level of error is acceptable as the linear approximation simplifies the calculations and the temperature measurement is for verification purposes; when PALs are generated, the temperature is expected to rise. As long as this behavior is observed, we can deduce that the system is functioning correctly.

4.3 Impedance measurements for conductivity analysis

Impedance is a complex quantity that represents the total opposition a circuit offers to the flow of alternating current (AC). It extends the concept of resistance to AC circuits and includes both resistance (the real part) and reactance (the imaginary part), which accounts for the effects of capacitors and inductors. Impedance is crucial in circuits where the phase shift between voltage and current is significant, such as in the analysis of PAL. In the setup, impedance is measured by using a voltage divider circuit, where a known resistor (R_1) is placed in series with the unknown impedance (Z_x) of our electrode.

$$Z_x = \left(\frac{V_{in} - V_{out}}{V_{out}} \right) \times R_1 \tag{4.6}$$

Where V_{in} is the input voltage and V_{out} is the output voltage across the unknown impedance Z_x.

This method allowed us to determine the impedance across a range of frequencies, providing insights into the conductive properties of the liquid. The AD5941 chip facilitates these impedance measurements, enabling the capture of both; the magnitude and phase of the impedance across different AC frequencies.
During the impedance measurement, three key variables are captured:

- **Freq (Frequency)**: Represents the frequency at which the impedance is measured. In this project, the frequency spectrum used ranges approximately from 200 Hz to 200,000 Hz, but in this test, the spectrum was extended from 100 Hz to 1,000,000 Hz to gather as much information as possible.
- **RzMag (Magnitude of Impedance)**: The absolute value of the impedance. This is the key parameter of interest as it is directly related to the conductivity of the liquid. By measuring $RzMag$, we can calculate how easily a material allows the flow of electric current (conductivity) by doing the inverse of the magnitude.
- **RzPhase (Phase of Impedance)**: Indicates the phase difference between the applied voltage and the resulting current. It helps in understanding the reactive components of the impedance, such as capacitance and inductance.

Table 4.3: Impedance measured at different frequencies for known resistances.

Theoretical Resistance (Ω)	100.0	1000.0	2000.0	10000.0
Freq (Hz)	**Multimeter - 100.5** Ω	**Multimeter - 1030.0** Ω	**Multimeter - 1993.5** Ω	**Multimeter - 9920.0** Ω
100	233.87	878.70	1484.16	7347.64
229.09	233.74	754.24	1485.62	7360.11
524.81	233.23	759.51	1491.19	7388.36
1318.26	230.29	777.22	1529.30	7578.62
3019.95	219.81	869.19	1710.60	8476.30
6918.31	178.92	933.60	1837.18	9105.34
15848.93	223.97	883.62	1740.10	8617.65
36307.81	231.94	983.38	1933.55	9592.02
75857.76	142.47	1019.38	2005.60	9946.11
190546.08	143.30	1019.01	2004.59	9943.25
436515.84	120.74	1019.90	2010.71	9963.87
1000000	120.73	1019.97	2010.28	9965.31

The results presented in Table 4.3 indicate that higher frequencies tend to yield impedance measurements closer to the theoretical resistance values. Particularly, frequencies above 100,000 Hz demonstrate increased accuracy and stability in the measurements. Despite this observation, the company opted to focus on lower frequency ranges between 200 Hz and 10,000 Hz for initial testing, considering factors such as the specific electrical properties of the liquids under study. Later, the frequency range was extended up to 200,000 Hz to explore any additional information from using a wider spectrum.

5. Results

5.1 Plasma Activated Water conductivity

In the initial testing phase, the goal was to verify the sensor's capability to detect changes in conductivity as gas was pumped into a water container, simulating PAW. The water was continuously circulating through the chamber, coupled with the sensor, using a pump operating at 2000 pulses per second. This setup ensured a constant flow and consistent contact between the water and the sensor throughout the testing.

As illustrated in Figure 5.1, the sensor successfully detected a progressive increase in conductivity as more gas was introduced into the container. The graph clearly shows a positive trend, confirming that the sensor can accurately track the expected rise in conductivity over time as the gas, which increases the water's ion concentration, is added. This test served as a preliminary validation of the sensor's functionality, confirming its ability to respond to changes in conductivity in real-time and setting the stage for more detailed analyses in subsequent experiments.

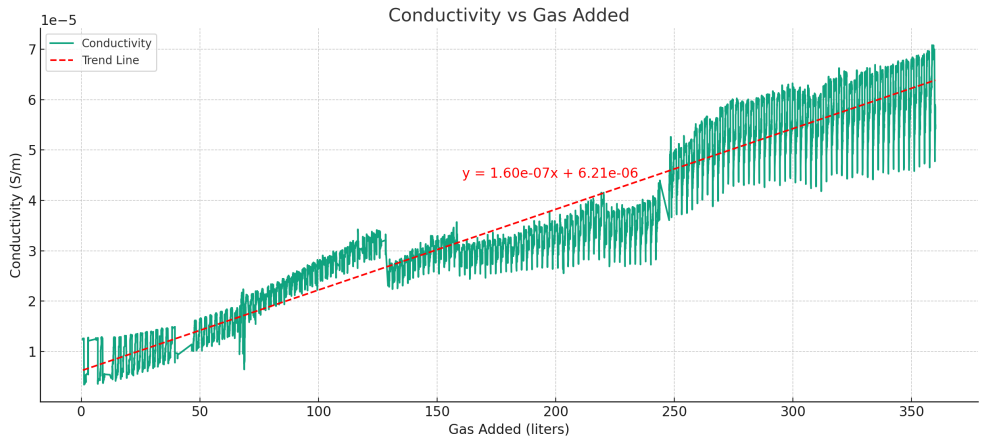

Figure 5.1: Conductivity vs. Gas: Demonstrating the sensor's ability to detect increases in water conductivity as gas is progressively added.

5.2 Accuracy validation with BlueLab liquids

In this experiment, we were presented with four liquids of unknown conductivity, in addition to a control sample of tap water. Subsequently, BlueLab provided us with the conductivity values obtained using their specific conductivity sensor. These values were used to contrast the accuracy of our sensor. The average conductivity results for each frequency, along with the relative error compared to BlueLab's values, are shown in Table 5.1.

The graph in Figure 5.2 illustrates the conductivity measurements over time for the control and the four unknown liquids. The dips observed between each measurement are the result of the liquid changes and should not be considered in the analysis. The behavior stabilizes once the liquid is being properly analyzed. The graph aligns with expectations as conductivity increases according to the liquid being tested. BlueLab provided the liquids in ascending order of conductivity to prevent contamination or error carry-over from a higher to a lower conductivity measurement.

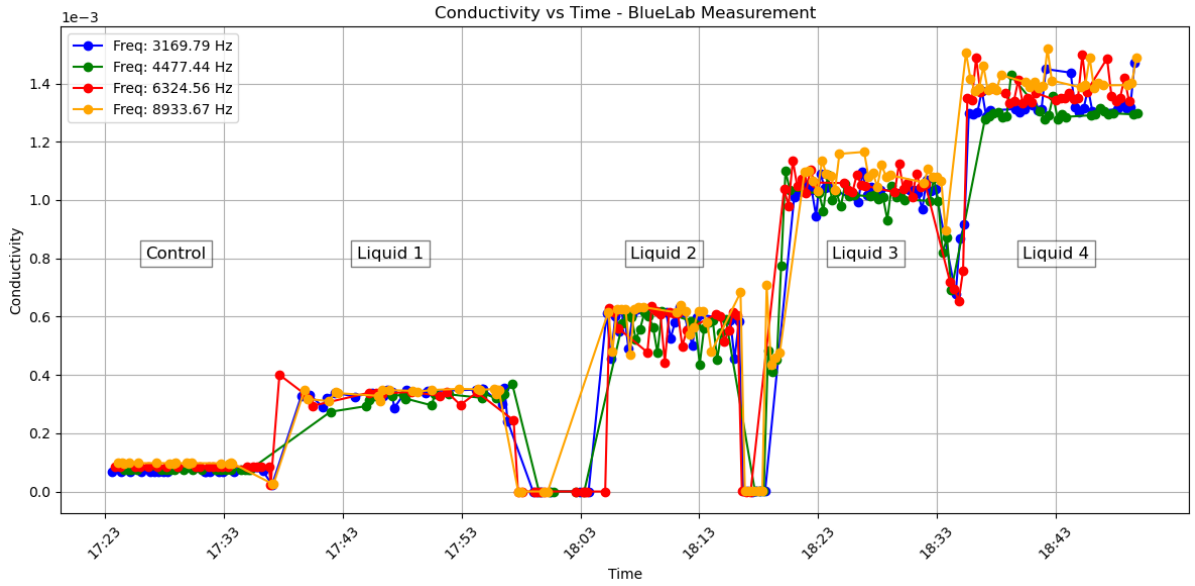

Figure 5.2: Conductivity vs Time for the different liquids tested.

Table 5.1: Validation of Conductivity Measurements with BlueLab

Measurement	Frequency (Hz)	Mean Conductivity (mS/cm)	BlueLab Conductivity (mS/cm)	Relative Error (%)
Control	3169.79	0.6886	0.50	37.73%
Control	4477.44	0.7516	0.50	50.32%
Control	6324.56	0.9334	0.50	86.67%
Control	8933.67	0.9726	0.50	94.52%
Liquid 1	3169.79	3.3613	3.03	10.92%
Liquid 1	4477.44	3.2213	3.03	6.30%
Liquid 1	6324.56	3.2939	3.03	8.71%
Liquid 1	8933.67	3.3607	3.03	10.93%
Liquid 2	3169.79	5.7829	5.46	5.92%
Liquid 2	4477.44	5.6994	5.46	4.39%
Liquid 2	6324.56	5.8383	5.46	6.92%
Liquid 2	8933.67	5.9227	5.46	8.47%
Liquid 3	3169.79	10.3620	10.54	1.69%
Liquid 3	4477.44	10.1420	10.54	3.78%
Liquid 3	6324.56	10.5210	10.54	0.18%
Liquid 3	8933.67	10.8510	10.54	2.94%
Liquid 4	3169.79	13.1350	14.30	8.15%
Liquid 4	4477.44	12.9290	14.30	9.59%
Liquid 4	6324.56	13.4880	14.30	5.68%
Liquid 4	8933.67	13.9580	14.30	2.40%

The results obtained indicate that, for the higher conductivity liquids (Liquids 1 to 4), our sensor provides measurements that are very close to those obtained by BlueLab, with errors below 10%. This suggests that our sensor is accurate for liquids with high conductivity and that the frequencies used were suitable for these measurement ranges. However, for the control (tap water), a higher error is observed, which may be due to the use of different sources of tap water as controls. Since BlueLab only provided the four test liquids, it's possible that the water control from different sources led to discrepancies in conductivity measurements. Overall, the results confirm that the measurements made by our sensor are quite accurate, particularly in higher conductivity liquids, where performance closely matched the standard provided by BlueLab.

5.3 Impedance measurements at the UKT

Lastly, we present the results obtained during the measurements conducted in the Universitätsklinikum Tübingen (UKT). Unlike previous experiments, which used only four frequencies, this time a full spectrum of 16 frequencies from 200 to 200000 Hz was employed. This decision was made in collaboration with the hospital and JLM, aiming to evaluate the information that each frequency could provide and to draw more detailed conclusions about the system's behavior in a clinical environment.

The figure 5.3 shows the comparison of impedance (rather than conductivity) across different frequencies over various activation times. In each measurement, a clear trend is observed: as the activation time increases, the impedance decreases. This result confirms that our system has been correctly integrated with the other parts of the EndoPAL project.

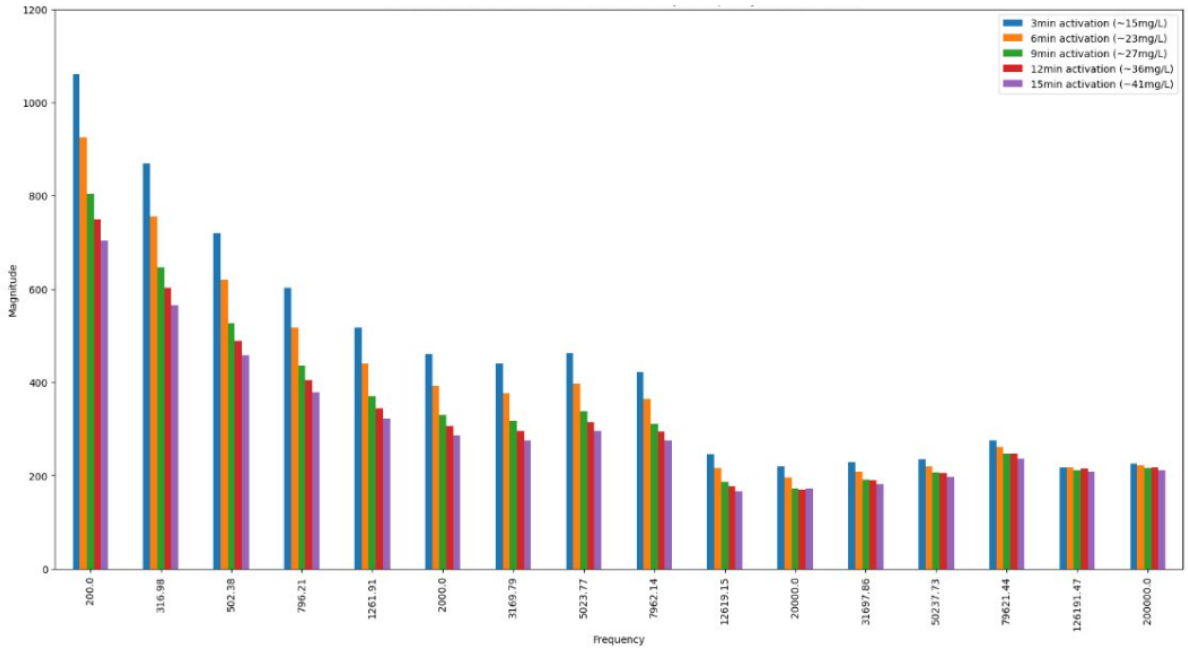

Figure 5.3: Impedance comparison across different frequencies at various activation times.

6. Conclusions

This project aimed to develop and implement an embedded system capable of measuring the conductivity of liquids in a clinical environment, focusing on detecting reactive oxygen and nitrogen species (RONS). The IoT system used an impedance sensor to measure the conductivity of plasma-activated liquids (PAL). Several challenges were faced during development, including the assembly and configuration of electronic components and the programming of the ESP32-S3 firmware, alongside building a reliable infrastructure for real-time data collection and visualization.

Within the EndoPAL-i project, the objectives were successfully achieved. This system has two primary applications: addressing postoperative adhesions and contributing to water protection in industrial environments to prevent Legionella infections. My contribution focused on the development of the IoT system, whose primary goal was to measure the impedance of plasma-activated liquids in order to calculate their conductivity and detect the presence of RONS.

One of the key accomplishments of this project was the successful integration of a Raspberry Pi as the server for the IoT system, utilizing technologies such as Docker, Node-RED, and InfluxDB for data management and visualization. This implementation allowed for the creation of a scalable and visually accessible system, representing a significant step toward its future integration into clinical or industrial environments. During the process, several initial tests were conducted, including voltage and temperature measurements, to ensure that the system components responded correctly and that the readings were consistent. These initial tests were essential to confirm that the system was calibrated and ready to perform conductivity measurements obtained through the inverse of impedance.

However, several issues were encountered during the development, particularly with the design of the measurement chamber. The first design, made of Teflon, exhibited leakage issues that compromised the accuracy of the measurements. This problem was resolved by redesigning the chamber using 3D-printed components, significantly improving its performance. Another significant challenge arose during the system installation at the hospital, where electromagnetic interference (EMI) from other medical devices was detected, affecting data transmission.

Overall, the impedance and conductivity measurements provided a solid first step toward detecting RONS in PAL. While the data collected so far is preliminary, it confirms that the system is capable of detecting increases in conductivity, ensuring that the project is moving in the right direction.

7. Challenges and Upgrades

The results obtained in the laboratory were highly satisfactory, confirming that the system functions correctly and can accurately measure the properties of PAL. However, during the installation at the hospital UKT, interferences were detected. These interferences, caused by EMI from other medical equipment, compromising the system's results. To mitigate this, enclosing the system with a Faraday cage proved to be an effective solution, blocking external electromagnetic fields and enhancing the reliability of the system.

One of the most promising upgrades for the system is the integration of UV light and a spectrometer. This addition would allow for advanced measurements by analyzing the absorbance of the liquid, providing a deeper understanding of the specific reactive species formed in PAL and their proportions. For instance, the ability to detect and quantify individual RONS species could lead to more precise therapeutic interventions in clinical settings or improved water disinfection processes in industrial environments [6].

In terms of system security, the current implementation relies on self-signed certificates generated by the author, providing only a basic level of protection. While this security approach may be enough for initial development and testing, it is insufficient for production. To enhance security, it is suggested to adopt industry-standard protocols such as TLS with certificates issued by trusted Certificate Authorities (CAs). Additionally, implementing Multi-Factor Authentication (MFA) for each service would ensure that only authorized users have access to sensitive system components and data. By incorporating these advanced security measures, the system will be protecting the data integrity and user safety.

This project has been a significant learning experience, allowing me to understand how disciplines like electronics, computer science, and mathematics integrate within biotechnology to solve complex problems. Developing an IoT system from scratch has provided me with valuable insight into designing and implementing real-time data collection, processing, and analysis systems. The skills I developed during my studies, particularly in problem-solving, and adapting to new technologies, were crucial in overcoming challenges such as the chamber design or EMI, and building a reliable, scalable IoT infrastructure.

Through this project, I have strengthened my technical abilities and gained confidence in applying engineering principles to real-world problems. It has taught me the importance of combining knowledge from different fields and has inspired me to explore further intersections between computer science and biotechnology. Additionally, this experience has enhanced my ability to adapt to unfamiliar situations and plan and execute projects effectively, which will be really valuable for future challenges in my career.

Bibliography

[1] Analog Devices Inc. (5941a). *AD5940-5941 datasheet*. Retrieved from https://www.analog.com/media/en/technical-documentation/data-sheets/AD5940-5941.pdf

[2] Analog Devices Inc. (5941b). *AD5940 examples* [Repository]. GitHub. Retrieved from https://github.com/analogdevicesinc/ad5940-examples

[3] Espressif Systems. (ESP32). *ESP32-S3 datasheet*. Retrieved from https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf

[4] Innovative Sensor Technology IST. (2023). *LFS1K0.1305.6W.B.010-6 Conductivity Sensor Datasheet*. Retrieved from https://www.farnell.com/datasheets/3167908.pdf

[5] Microtherm Sentronic GmbH. (2022). *PT1000 Temperature Sensor Datasheet*. Retrieved from https://www.microtherm.de/files/microtherm/downloads/MIC_PT_EN_rz_db_20220321_web.pdf

[6] Liu, Z., Zhou, C., Liu, D., He, T., Guo, L., Xu, D., & Kong, M. G. (2019). *Quantifying the concentration and penetration depth of long-lived RONS in plasma-activated water by UV absorption spectroscopy*. AIP Advances, 9(1), 015014. https://doi.org/10.1063/1.5037660

[7] Zhou, R., Zhou, R., Wang, P., Xian, Y., Mai-Prochnow, A., Lu, X., Cullen, P. J., Ostrikov, K., & Bazaka, K. (2020). *Plasma-activated water: generation, origin of reactive species and biological applications*. Journal of Physics D: Applied Physics, 53(30), 303001. https://doi.org/10.1088/1361-6463/ab81cf

[8] Yan, D., Sherman, J. H., & Keidar, M. (2017). *Cold atmospheric plasma, a novel promising anti-cancer treatment modality*. Oncotarget, 8(9), 15977-15995. Retrieved from http://www.impactjournals.com/oncotarget/

A. Appendices

A.1 Figures

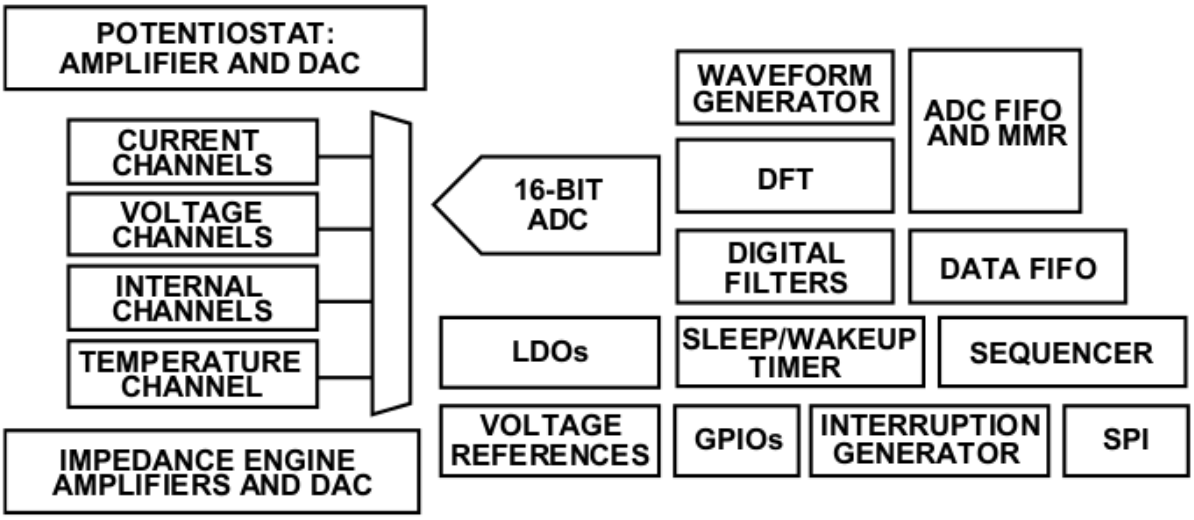

Figure A.1: Simplified block diagram of the AD5941, showing key components involved in impedance measurement and data acquisition.

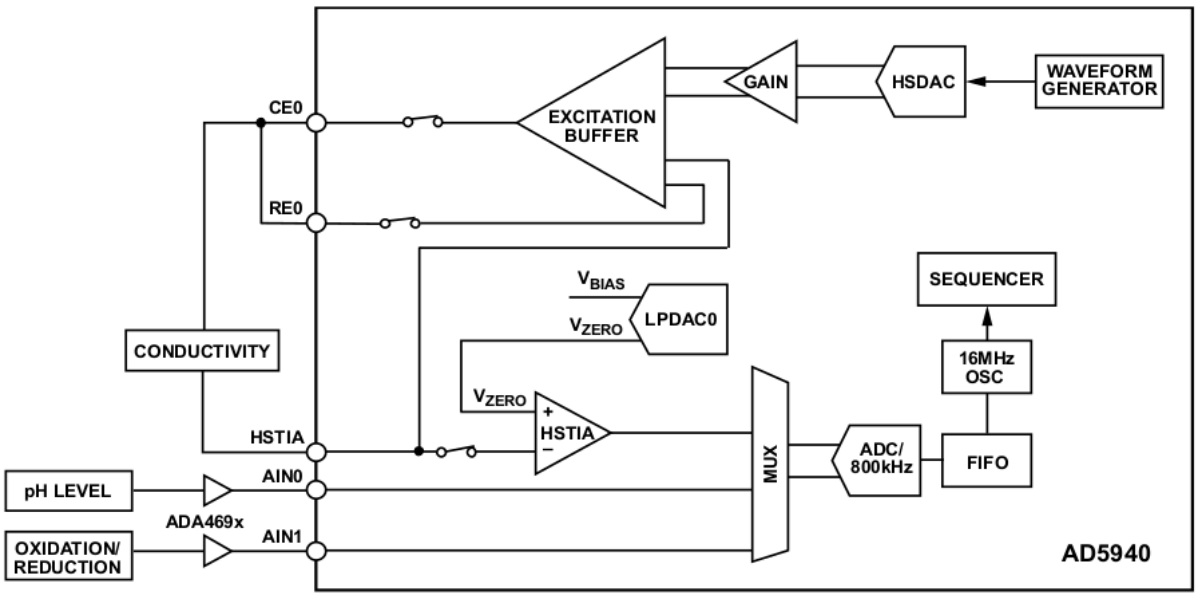

Figure A.2: Typical water analysis application using the AD5940/AD5941, illustrating the impedance measurement process.

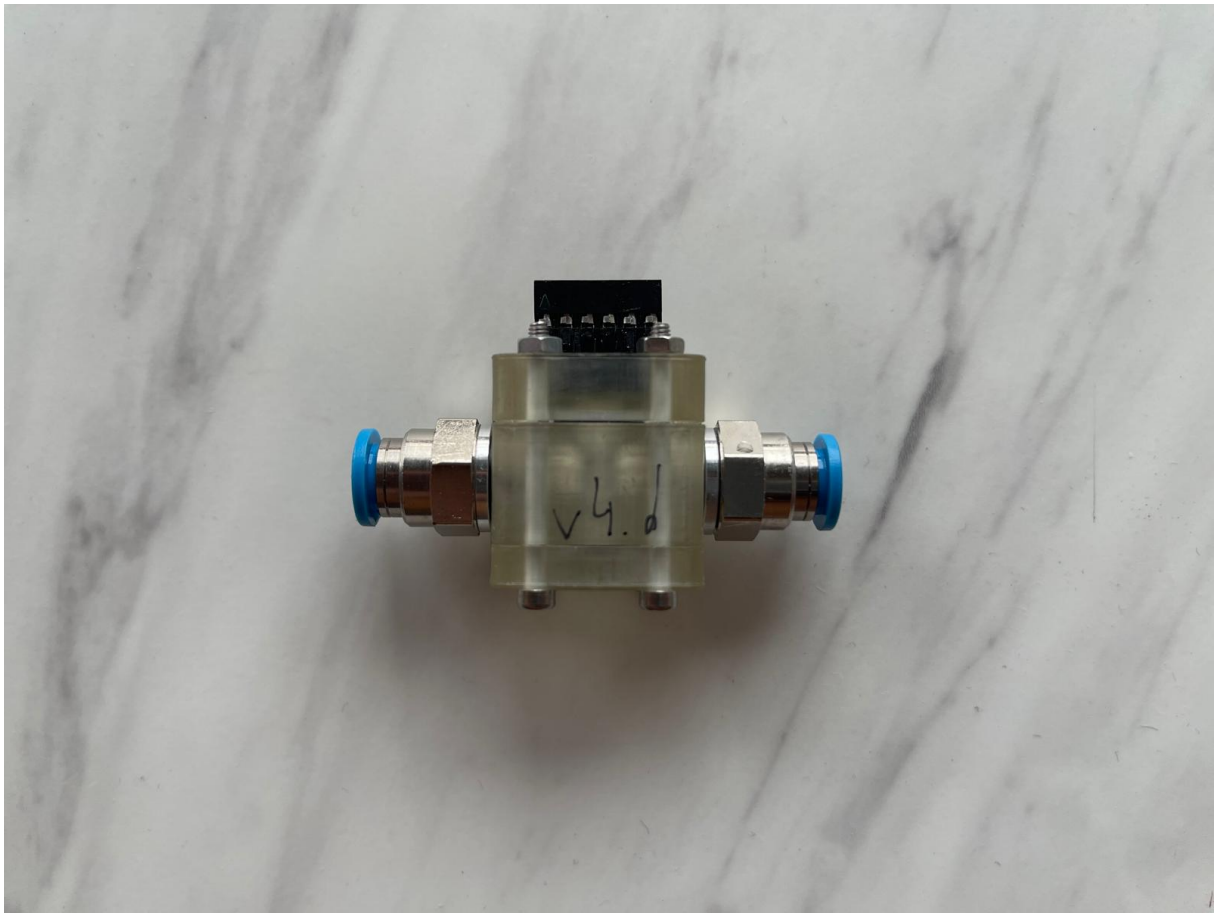

Figure A.3: Complete assembly of the flow chamber with the conductivity sensor inserted, v4d.

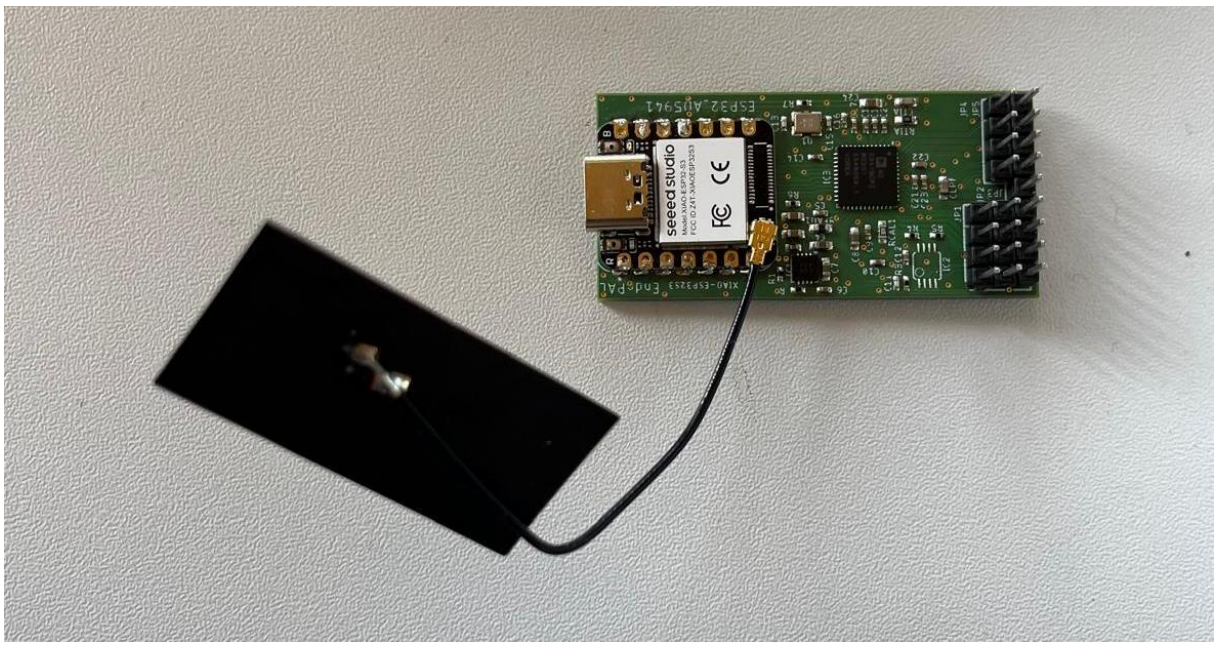

Figure A.4: PCB populated with the ESP32-S3 Xiao, its antenna, and the AD5941 chip.

A.2 Firmware Initialization Code

Listing A.1: Initialization of WiFi, GPIOs for SPI, and MQTT client

```
void app_main(void)
{
        // Initialize Watchdog Timer (WDT)
        esp_task_wdt_deinit(); // Disable WDT for easier testing

        // Initialize Logger
        initLog();

        // Initialize WiFi
        wifi_init_sta();

        // Initialize MQTT client
        mqtt5_app_start();

        // Initialize GPIOs for SPI communication
        initGPIO();

        // Start main AD5940 measurement routine
        AD5940_Main();
}
```

A.3 Main Routine for Measurements

Listing A.2: Main routine for managing measurements

```
void AD5940_Main(void)
{
        // Initial platform configuration (Impedance measurement setup)
        AD5940PlatformCfg(false);

        // Initialize impedance measurement
        AppIMPInit(AppBuff, APPBUFF_SIZE);
        AppIMPCtrl(IMPCTRL_START, 0);

        uint32_t adc_count = 0;
        uint32_t temp_count = 0;
        uint32_t impedance_temp;
        uint32_t impedance_iterations = 0;
        bool imp_flag = false; // Flag to indicate impedance measurement
        bool temp_flag = false; // Flag to indicate temperature measurement

        while (1)
        {
                // Check if neither impedance nor temperature measurement is active
                if(!imp_flag && !temp_flag)
                {
                        if (AD5940_INTCTestFlag(AFEINTC_1, AFEINTSRC_SINC2RDY))
                        {
                                AD5940_INTCClrFlag(AFEINTSRC_SINC2RDY);
                                uint32_t adc_rd = AD5940_ReadAfeResult(AFERESULT_SINC2);
                                adc_count++;

                                if (adc_count == 150) // Trigger impedance measurement after acquiring 150 ADC samples
                                {
                                        adc_count = 0;
                                        float adc_diff_volt = AD5940_ADCCode2Volt(adc_rd, ADCPGA_GAIN_SEL, 1.82);
                                        ADCShowResult(adc_rd, adc_diff_volt);

                                        imp_flag = true; // Set flag to start impedance measurement
                                        AppIMPInit(AppBuff, APPBUFF_SIZE);
                                        AppIMPCtrl(IMPCTRL_START, 0);
                                }
                        }
```

```
            }

            // Perform impedance measurement if flag is set
            if (imp_flag)
            {
                if (AD5940_GetMCUIntFlag())
                {
                    AD5940_ClrMCUIntFlag();
                    impedance_temp = APPBUFF_SIZE;
                    AppIMPISR(AppBuff, &impedance_temp);
                    ImpedanceShowResult(AppBuff, impedance_temp);
                    impedance_iterations++; // Count impedance measurements

                    if (impedance_iterations == AppIMPCfg.SweepCfg.SweepPoints) // Switch to temperature measurement after rea
                    {
                        imp_flag = false;
                        temp_flag = true;
                        impedance_iterations = 0;
                        AD5940PlatformCfg(true); // Reconfigure for temperature measurement
                    }
                }
            }

            // Perform temperature measurement if flag is set
            if(!imp_flag && temp_flag)
            {
                if (AD5940_INTCTestFlag(AFEINTC_1, AFEINTSRC_SINC2RDY))
                {
                    AD5940_INTCClrFlag(AFEINTSRC_SINC2RDY);
                    uint32_t temp_rd = AD5940_ReadAfeResult(AFERESULT_SINC2);
                    temp_count++;

                    if (temp_count == 150) // Trigger temperature measurement after acquiring 150 ADC samples
                    {
                        temp_count = 0;
                        float temp_diff_volt = AD5940_ADCCode2Volt(temp_rd, ADCPGA_GAIN_SEL, 1.82);
                        TemperatureShowResult(temp_rd, temp_diff_volt);
                        AD5940PlatformCfg(false); // Reconfigure for next impedance measurement
                        temp_flag = false;
                    }
                }
            }

            vTaskDelay(1); // Small delay to avoid excessive CPU consumption
        }
}
```

A.4 Firmware Data Processing and Transmission

Listing A.3: Impedance Data Processing and Transmission

```
int32_t ImpedanceShowResult(uint32_t *pData, uint32_t DataCount)
{
    float freq;

    fImpPol_Type *pImp = (fImpPol_Type *)pData;
    AppIMPCtrl(IMPCTRL_GETFREQ, &freq); // Get the current frequency
    /* Process data */
    for (int i = 0; i < DataCount; i++)
    {
        char sendString[64];
        snprintf(sendString, sizeof(sendString),
        "{\"Freq\":%8.2f,\"RzMag\":%8.2f,\"RzPhase\":%4.6f}",
        freq, pImp[i].Magnitude, pImp[i].Phase * 180 / MATH_PI);
        printf("%s\n", sendString);
        sendMQTT("AD5941/Impedance", sendString); // Send data via MQTT
    }
    return 0;
}
```

Listing A.4: ADC Voltage Data Processing and Transmission

```
void ADCShowResult(int adcCode, float adcVoltage)
{
        char sendString[64];
        snprintf(sendString, sizeof(sendString),
        "{\"ADCCode\":%d, \"ADCVolt\":%.4f}",
        adcCode, adcVoltage);
        printf("%s\n", sendString);
        sendMQTT("AD5941/ADC", sendString); // Send ADC data via MQTT
}
```

Listing A.5: Temperature Data Processing and Transmission

```
void TemperatureShowResult(int adcCode, float adcVoltage)
{
        char sendString[128];
        float knownR = 1000;
        float unknownR;
        float temperature;
        unknownR = adcVoltage / ((1.82-adcVoltage)/knownR); // Calculate unknown resistance
        temperature = (unknownR - 1000) / 3.87; // Convert resistance to temperature
        snprintf(sendString, sizeof(sendString),
        "{\"ADCCode\":%d, \"ADCVolt\":%.4f, \"Unk. R\":%f, \"Temp\":%f}",
        adcCode, adcVoltage, unknownR, temperature);
        printf("%s\n", sendString);
        sendMQTT("AD5941/Temperature", sendString); // Send temperature data via MQTT
}
```

A.5 Docker Compose Configuration

Listing A.6: Docker Compose Configuration for IoT System

```
networks:
default:
driver: bridge
ipam:
driver: default
nextcloud:
driver: bridge
internal: true
ipam:
driver: default

services:
portainer-ce:
container_name: portainer-ce
image: portainer/portainer-ce
restart: unless-stopped
ports:
- "9000:9000"
volumes:
- /var/run/docker.sock:/var/run/docker.sock
- ./volumes/portainer-ce/data:/data

mosquitto:
container_name: mosquitto
build:
context: ./.templates/mosquitto/.
args:
- MOSQUITTO_BASE=eclipse-mosquitto:latest
restart: unless-stopped
environment:
- TZ=$("-TZ:-Etc/UTC")
ports:
- "1883:1883"
volumes:
- ./volumes/mosquitto/config:/mosquitto/config
- ./volumes/mosquitto/data:/mosquitto/data
```

```
- ./volumes/mosquitto/log:/mosquitto/log
- ./volumes/mosquitto/pwfile:/mosquitto/pwfile

nodered:
container_name: nodered
build:
context: ./services/nodered/.
args:
- DOCKERHUB_TAG=latest
- EXTRA_PACKAGES=
restart: unless-stopped
user: "0"
environment:
- TZ=$"-TZ:-Etc/UTC"
ports:
- "1880:1880"
volumes:
- ./volumes/nodered/data:/data
- ./volumes/nodered/ssh:/root/.ssh

influxdb2:
container_name: influxdb2
image: "influxdb:latest"
restart: unless-stopped
environment:
- TZ=Etc/UTC
- DOCKER_INFLUXDB_INIT_USERNAME=
- DOCKER_INFLUXDB_INIT_PASSWORD=
- DOCKER_INFLUXDB_INIT_ORG=
- DOCKER_INFLUXDB_INIT_BUCKET=
- DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=
- DOCKER_INFLUXDB_INIT_MODE=setup
ports:
- "8087:8086"
volumes:
- ./volumes/influxdb2/data:/var/lib/influxdb2
- ./volumes/influxdb2/config:/etc/influxdb2
- ./volumes/influxdb2/backup:/var/lib/backup
healthcheck:
test: ["CMD", "influx", "ping"]
timeout: 10s
retries: 3
start_period: 30s
```