Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Measurement

Impedance-meter for quantification of ethanol content in liquid solutions

Bachelor Thesis

Arkady Vainshtein

B.Sc. program: Electrical Engineering and Computer Science
Supervisor: Doc. Ing. Mattia Butta Gonzales Ph.D.

May, 2023

Thesis Supervisor:
Doc. Ing. Mattia Butta Gonzales Ph.D.
Department of Measurement
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
160 00 Prague 6
Czech Republic

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Vainshtein Arkady** Personal ID number: **498660**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Measurement**

Study program: **Electrical Engineering and Computer Science**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Impedance-meter for quantification of ethanol content in liquid solutions

Bachelor’s thesis title in Czech:

Impedanční měřič pro kvantifikaci obsahu etanolu v kapalných roztocích

Guidelines:

The student will create a device which must be able to measure the ethanol content in a solution.
The first task is to study the dependance of the impedance vs. frequency for solutions containing different amounts of ethanol, using a bench-top RLC meter and custom made PCB probes for the solution. The student then will select the frequency where the impedance changes the most vs. the percentage of ethanol and produce a figure of merit of this dependance.
After studying the physics and behaviour of the impedance vs. amount of ethanol he will create a custom made multi-channel impedance meter based on AD5933 and a microcontroller which will allow to measure the impedance of the solution at different heights of the container of the solution in order to assess the distribution of the liquids. This will be done using a multiplexer controlled by the microcontroller and multi-point probes. The system will be equipped by several sensors of temperature to obtain a profile of temperature inside the container and correct the measurement of impedance.
The project includes the design and manufacturing of the PCB as well as the coding of the firmware.
The final task will be testing the system with different types of solutions.

Bibliography / sources:

[1] Northrop, R. B.: Introduction to Instrumentation and Measurements. CRC 2005, ISBN 0-8493-3773-9
[2] Ripka, P., Tipek, A. (Eds.): Modern Sensors Handbook. ISTE 2007, ISBN 978-1-905209-66-8
[3] Haasz, V., Sedláček, M.: Electrical Measurements. ČVUT in Prague, Praha 2006
[4] https://www.analog.com/en/products/ad5933.html#product-overview

Name and workplace of bachelor's thesis supervisor:

doc. Ing. Mattia Butta, Ph.D. Department of Measurement FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **16.02.2023** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until:
by the end of summer semester 2023/2024

doc. Ing. Mattia Butta, Ph.D. Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Declaration

I hereby declare that this Bachelor's thesis is the product of my own independent work and that I have clearly stated all information sources used in the thesis according to Methodological Instruction No.1/2009 – "On maintaining ethical principles when working on a university final project, CTU in Prague.

May, 2023

.....
Arkady Vainshtein

Abstract

This thesis presents the design, development, manufacturing, and implementation of a multi-channel impedance meter for quantifying ethanol content in liquid solutions. The system incorporates the AD5933 impedance converter IC, ESP32 microcontroller, temperature sensor, and a gradient probe to assess ethanol percentage and liquid distribution while compensating for temperature variations.
The thesis covers the hardware and firmware design, manufacturing of a custom PCB, and the implementation of the system. The final stage involves testing the system, performance validation, and assessment of its accuracy.

Keywords: Impedance meter, AD5933, ESP32.

List of Tables

List of Figures

List of Abbreviations

ADC Analog to Digital Converter. viii, 24–26, 29, 30, 35, 42, 43

CMOS Complementary Metal-Oxide Semiconductor. 31

DDS Direct Digital Synthesis. 25

DFT Discrete Fourier Transform. 25, 26

DSP Digital Signal Processor. 25, 26

GPIO General Purpose Input/Output. 29, 30, 37, 44, 45, 56

IC Integrated Circuit. vi, 24, 25, 27, 30, 32, 33, 37

I²C Inter-Integrated Circuit. 25, 29, 30, 35, 56

LCD Liquid Crystal Display. 35

LSB Least Significant Bit. 35, 41

MCU Micro-Controller Unit. ix, 20, 24, 29, 31, 36, 37, 48, 52, 53

MUX Multiplexer. 28, 29, 33, 44, 47, 57

OLED Organic Light-Emitting Diode. 24, 30, 35

PCB Printed Circuit Board. vi, viii, ix, 14, 24, 30, 35, 37–39, 41, 45, 47, 48, 64

SCL Serial Clock Line. 35

SDL Serial Data Line. 35

SMD Surface Mount Devices. 36, 37

SPI Serial Peripheral Interface. 29, 35

SRAM Static Random Access Memory. 29

THT Through Hole Technology. 38

UART Universal Asynchronous Receiver-Transmitter. 29

USB Universal Serial Bus. 24, 30

Contents

Chapter 1

Introduction

This thesis presents the design, implementation, and evaluation of a multi-channel impedance meter specifically designed for ethanol content analysis. The system integrates advanced components such as the AD5933 impedance converter, ESP32 microcontroller, temperature sensor, and gradient probe. By leveraging the principles of impedance measurement, the device enables the determination of ethanol percentages while compensating for temperature variations within the solution.

The primary objective of the thesis is to design and optimize the hardware and firmware components of the impedance meter system. This includes the development of a custom-made PCB. Furthermore, the firmware implementation on the ESP32 microcontroller enables efficient data acquisition, processing, and analysis.

The thesis also focuses on comprehensive testing, performance evaluation, and validation of the impedance meter system. Through a series of experiments conducted on different liquid solutions, the accuracy, repeatability, and stability of ethanol measurements are assessed. Additionally, measurements are carried out to approximate the distribution of ethanol within a given solution by measuring at multiple depths.

Chapter 2

Measurements

The primary objective of the measurements was to determine whether there is any correlation between impedance and frequency in different alcohol solutions and if so, determine the most sensitive frequency. To accomplish this, a custom probe connected to an LCR meter (HAMEG HM8118) was employed. The probe consisted of two slim PCBs with rectangular copper pads and traces coated with a clear lacquer to electrically insulate exposed traces, as depicted in Figure 2.1. The distance between the pads was 5mm.

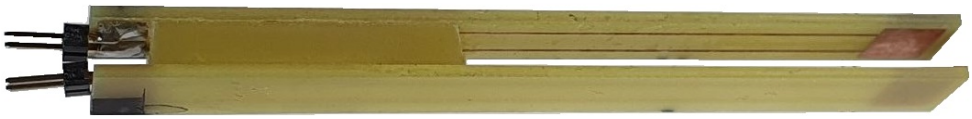


Figure 2.1: Test Probe

The alcohol solutions were composed of varying proportions of distilled water and 96% purity ethanol (by mass). To calculate the density, it was assumed that the remaining 4% of ethanol had a density equivalent to distilled water. Due to the different densities of the liquids, measuring the weight of each component and mixing them was insufficiently accurate. Instead, precise volumes were calculated according to the mass and density of each component before being mixed in a beaker with a capacity of approximately 150ml. The probe's pads were suspended at the center of the beaker.
All measurements were taken using an excitation voltage amplitude of 1V and a frequency range of 1 kHz to 100 kHz. The LCR meter provided output values of impedance magnitude Z in kΩ and phase Θ in degrees.

2.1 Preliminary Investigation

To observe the change in impedance magnitude caused by ethanol, I conducted measurements with a beaker filled with distilled water and then repeated the measurement with ethanol.

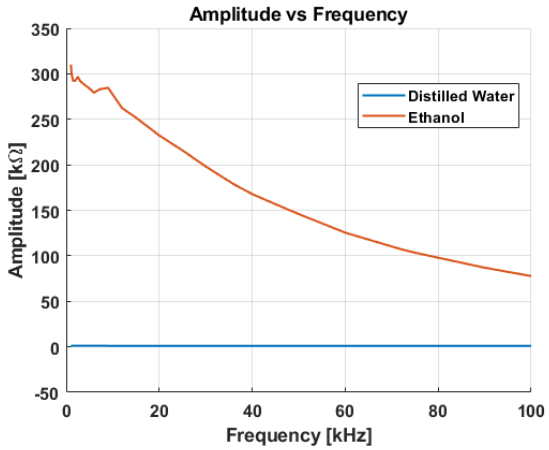


Figure 2.2: Amplitude vs Frequency

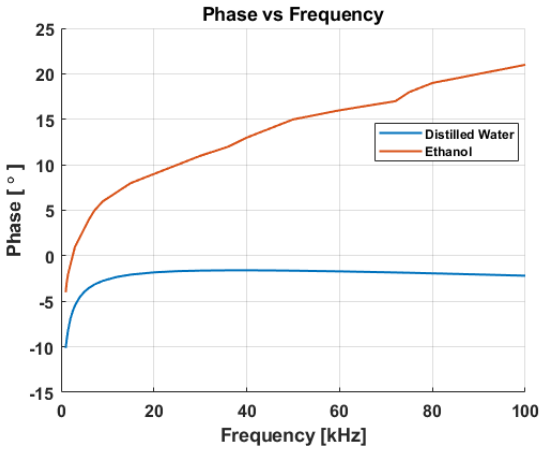


Figure 2.3: Phase vs Frequency

Figure 2.2 and 2.3 depict the impedance magnitudes and phases, respectively, of distilled water and ethanol at various frequencies. The impedance magnitudes in water ranged between 1.22 and 1.08 kΩ, while in ethanol they ranged between 310.2 and 77.7 kΩ. These results confirm a substantial and measurable difference in the impedance magnitudes between the two liquids. Additionally, as the frequency increased, the impedance magnitude in ethanol decreased at a higher rate than in water.
The phase changed in both water and ethanol. The phase angle started at -10° in water and increased by 8°, while in ethanol it started at -4° and increased by 25°. Therefore, ethanol increased the phase angle of the impedance and its rate of change.
These findings further support the presence of significant differences in the electrical properties of water and ethanol.

2.2 Impedance vs. Frequency

Once it was established that there is a measurable change in impedance magnitude and phase between a 0% and 100% ethanol solution, the aim was to identify the excitation frequency at which the rate of change is the highest. A higher rate of change indicates that for each percent of ethanol, the impedance difference will be greater, thereby enhancing the accuracy of the meter's quantification capabilities.
To accomplish this, I conducted a frequency sweep with various ethanol solutions. I began with 120 ml of distilled water and added ethanol to the same solution in the beaker prior

to each frequency sweep.
Table 2.1 lists information about the 7 solutions that I created. At Solution No.4, I reached the volume limit of the beaker. Thus, each time, I had to transfer the solution into a larger beaker and add ethanol. Solutions No. 5-7 were mixed in the large beaker and then poured into the measurement beaker with precisely 120ml. Although Table 2.1 displays the total volume of mixed solutions, measurements of solutions No. 5-7 were conducted using exactly 120ml.

Table 2.1: Ethanol solutions

No	Ethanol added [g]	Total Volume [ml]	Ethanol in volume[%]
1	0	120.7	0
2	5.450	128.022	5.394
3	8.697	139.698	12.832
4	10.269	153.484	20.157
5	13.179	171.177	27.829
6	7.641	181.436	31.592
7	15.836	202.697	38.178

Figures 2.4 and 2.5 depict the impedance magnitudes and phases, respectively, of the various ethanol solutions at various frequencies. It is evident from Figure 2.4 that the impedance magnitude increases with the percentage of ethanol in the solution. However, the magnitude of the impedance at 38% ethanol was much lower than expected. In the initial measurement with 100% ethanol, the magnitude started at 310kΩ, but with 38% ethanol, it started at only 5Ω. This discrepancy was causing challenges in the characterization of impedance dependence on the ethanol content. On the other hand, Figure 2.5 shows that the phase increases with the percentage of ethanol. However, pure distilled water had different phase angles, starting from -19° instead of the previously measured -10°. Furthermore, there are two different trends in the phase data. Solutions No.5-7 follow one trend, suggesting that their constant volume of 120 ml might be a factor.
Initially, I hypothesized that volume affects the measurements somehow, but this still didn't explain the difference in measurements of pure water at the same 120 ml volume. After inspecting the probe, I noticed that the coating around the submerged parts dissolved (probably during the initial pure ethanol solution measurement), exposing the traces and creating a larger conductive area that is exposed to the solution. This, in turn, decreases the impedance. This meant that those measurements are not accurate.

I decided to conduct a second round of measurements using freshly mixed solutions of approximately 120 ml in volume for each frequency sweep. By keeping the amount of copper submerged in the solution constant, it was anticipated that the results obtained would be more consistent. Subsequently, I conducted measurements with seven different

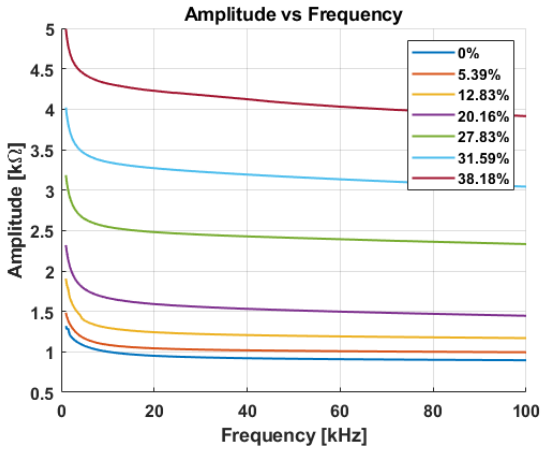


Figure 2.4: Amplitude vs Frequency at Different Ethanol Percentages

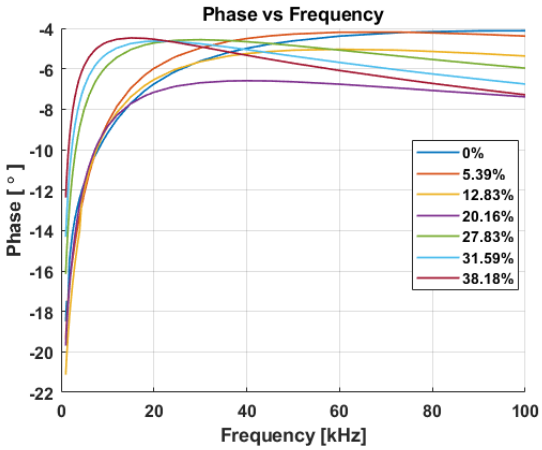


Figure 2.5: Phase vs Frequency at Different Ethanol Percentages

solutions with varying ethanol concentrations, as outlined in Table 2.2.

Table 2.2: Ethanol solutions

No	Ethanol added [g]	Total Volume [ml]	Ethanol in volume[%]
1	0	120.057	0
2	5.450	121.207	5.412
3	8.697	121.301	10.096
4	10.269	120.951	14.524
5	13.179	122.370	20.074
6	7.641	122.924	24.997
7	15.836	124.0267	34.708

The results of the new measurements were found to be more uniform. Specifically, the impedance magnitudes depicted in Figure 2.6 followed a similar trend as the previous measurement but with smaller magnitudes due to differences in ethanol percentages. We can see how the amplitude of the impedance decade as we increase the frequency. This is a common behavior for all solutions. For all frequencies, we can observe a shift in the amplitude of the impedance which depends on the ethanol percentage. Figure 2.7 revealed that the frequency sweeps for all ethanol concentrations exhibited a common trend. The impedance phases of the solutions differed most significantly at around 10kHz and 100kHz, implying that measurements obtained at these frequencies would be the most sensitive to variations among the solutions.

Through this experience, I gained an appreciation for the significance of meticulous consideration beyond the confines of graphical data and the criticality of scrutinizing measurement techniques even when they appear seemingly valid. Moreover, it highlighted the remarkable sensitivity of every millimeter of exposed copper trace and the indispensable

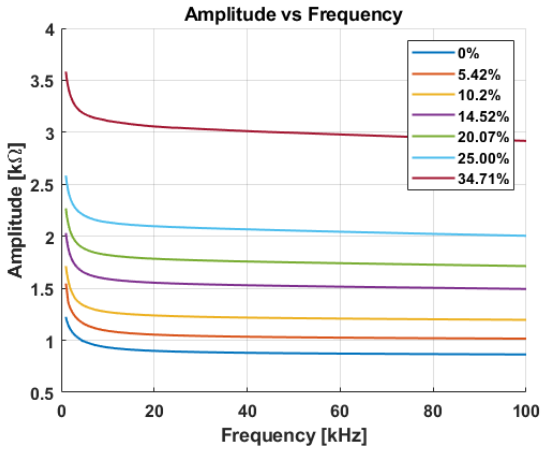


Figure 2.6: Amplitude vs Frequency at Different Ethanol Percentage

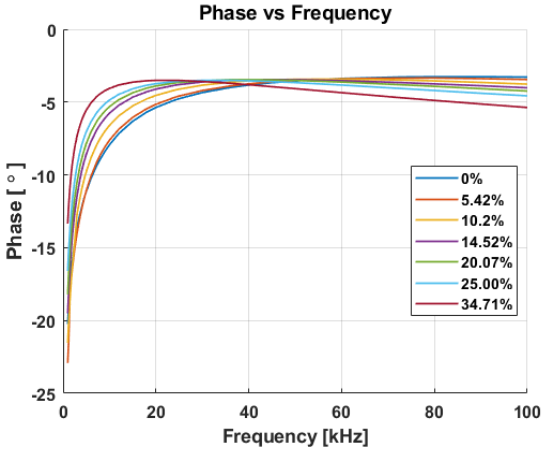


Figure 2.7: Phase vs Frequency at Different Ethanol Percentage

role of effectively insulating the final probe.

2.3 Optimal Frequency

In order to determine the optimal frequency for the meter, a sensitivity analysis was conducted for both 10kHz and 100kHz. The sensitivity was defined as the rate of change of impedance with respect to ethanol percentage, i.e., the slope at that frequency. To obtain the best slope, the magnitude was separated into its real and imaginary parts, and one of them was fitted to the data points using linear regression. The real and imaginary parts of the impedance were calculated using the phase angle such that the Real part $= Z\cos(\Theta)$ and the imaginary part $= Z\sin(\Theta)$. Figures 2.8 and 2.9 depict the real and imaginary parts vs. frequency, respectively.

To determine the optimal frequency, the imaginary parts of 10kHz and 100kHz were iso-

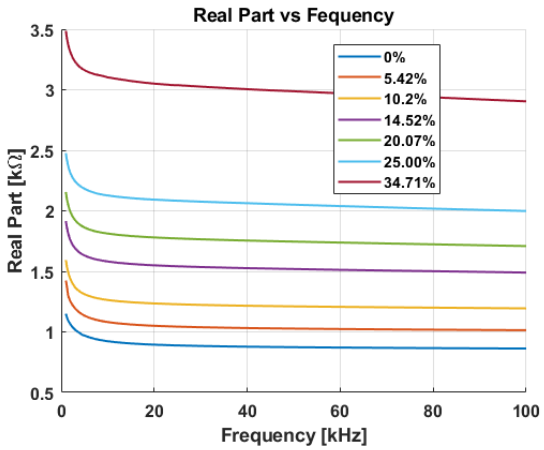


Figure 2.8: Real Part vs Frequency at Different Ethanol Percentage

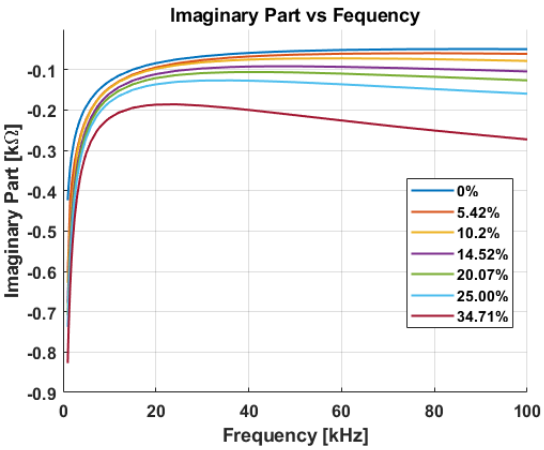


Figure 2.9: Imaginary Part vs Frequency at Different Ethanol Percentage

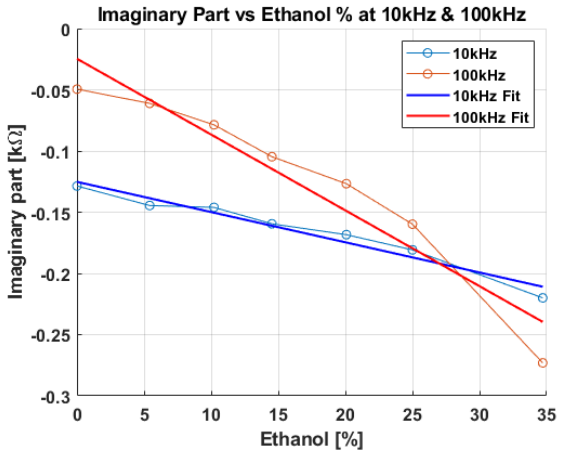


Figure 2.10: Imaginary Part vs Ethanol % at 10kHz and 100kHz

lated with respect to ethanol percentage, linearly fitted, and plotted as shown in Figure 2.10.

The results clearly indicate that the slope at 100kHz is steeper, indicating higher sensitivity. However, part of the information could be on the real and other on the imaginary axis, we can maximize the response obtained by the meter by shifting the phase of the impedance by a constant value that condensates the information on either the real or imaginary part. therefore I looked for the optimal phase bias.
To confirm this, a script was developed in Matlab to calculate the sensitivity at different phase biases. The imaginary part of the magnitude at 10kHz and 100kHz with 0° phase bias was used as the starting point. At each iteration, the script linearly fitted the data points, calculated the slope, and incremented the phase bias by 0.5° until it reached 120°. The highest absolute value of the slopes at 10kHz and 100kHz was then calculated to determine the best slope between them.

Figure 2.11 depicts the sensitivity of the imaginary parts at different phase biases. The results show that the sensitivity at 100kHz appears better at 0° phase bias, but if a phase bias is added, the sensitivity at 10kHz becomes better. The best sensitivity of the imaginary part at 100kHz is 0.057860 $\frac{k\Omega}{\%}$ with 96° phase bias, while the best sensitivity of the imaginary part at 10kHz is 0.061440 $\frac{k\Omega}{\%}$ with 92.5° phase bias, which is equivalent to a real part with only 2.5° phase bias which is shown in Figure 2.12.

These results clearly indicate that the optimal frequency for the meter is 10kHz, as it has the best sensitivity around the real part of the impedance magnitude. Overall, the findings underscore the importance of conducting a sensitivity analysis and considering phase biases to determine the optimal frequency. With this, the measurements are finalized and the data will suffice for now. However, as the development of the sensor

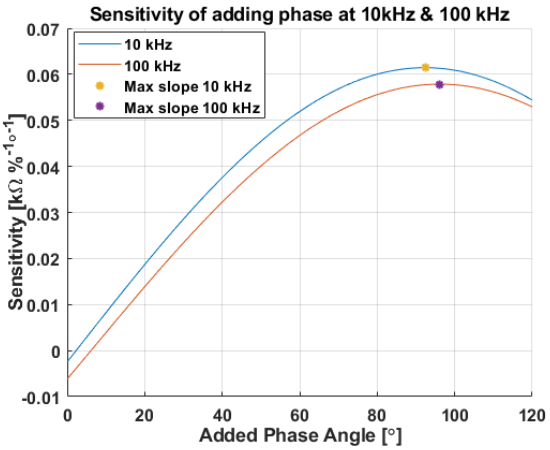


Figure 2.11: Sensitivity of Imaginary part vs Phase Bias at 10kHz and 100kHz

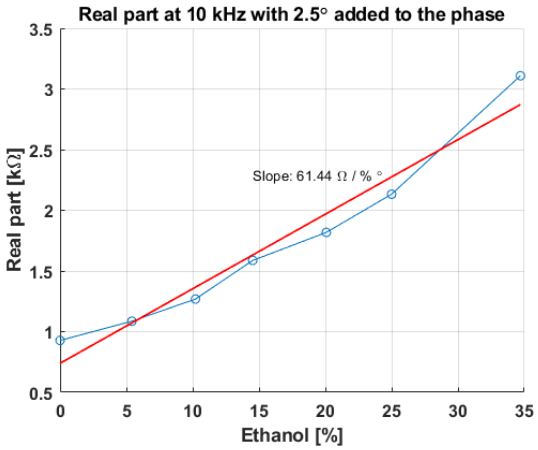


Figure 2.12: Linear fit of Real Part at 10kHz and 2.5° Bias

continues, more measurements will be made with the final isolated probe to further refine and optimize its performance.

In conclusion, the sensitivity analysis conducted in this chapter has allowed us to determine the optimal frequency for the ethanol meter. By analyzing the real and imaginary parts of the impedance and experimenting with phase biases, we have established that a frequency of 10kHz provides the best sensitivity for the measurement of ethanol concentration.

2.4 AD5933 Evaluation Board

The impedance meter will be based on the AD5933[1] 12-bit impedance converter. A detailed description of the AD5933 chip and its role in the impedance meter will be provided in Section 3.2.1. Analog Devices, the manufacturer, offers an evaluation board[2] for the AD5933 that includes a MCU and a graphical user interface software with frequency sweep capability. To understand the operational principles of the AD5933, perform measurements, and compare them to previous measurements, I utilized the evaluation board. The primary objective of this was to make new measurements at 10kHz using solutions that closely matched those in Section 2.2 and compare the results with the previous data. This comparison aimed to determine the suitability of the AD5933 for such measurements and obtain an overview of its features.

In order for the AD5933 evaluation board to operate properly, it requires a feedback resistor. It's important to choose a resistor that closely matches the impedance values to be measured, as this will ensure that the board operates within the linear region of the sensor. For my purposes, I selected a simple metal film resistor with a resistance of 2kΩ, which is near the center of the impedance range I previously measured.

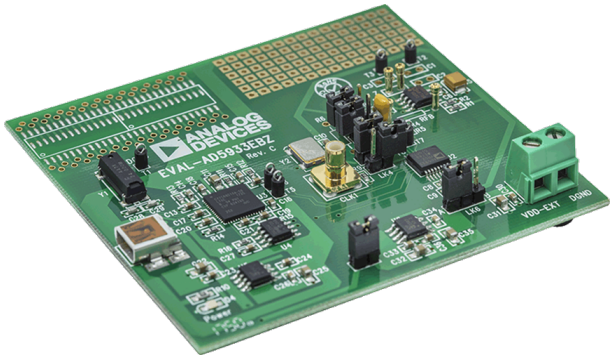


Figure 2.13: EVAL-AD5933EBZ [2]

Before proceeding with any measurements, it is necessary to calibrate the AD5933 chip. This involves performing a frequency sweep across a known resistance, which, in this case, is a resistor that closely matches the feedback resistor[3]. I calibrated the chip using the provided GUI software and measured different resistors to check if the sensor measured correctly. I then proceeded with the measurements, knowing that the results would be accurate and reliable.

I performed measurements on the new solutions using the AD5933, measuring the impedance of each solution at 10kHz, and compared the results to the LCR measurements. Figures 2.14 and 2.15 show the AD5933 measurements of amplitude vs ethanol and phase vs amplitude, respectively.

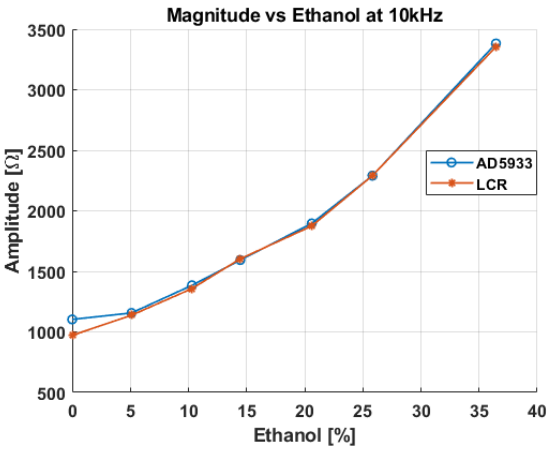


Figure 2.14: Amplitude vs Ethanol at 10kHz.

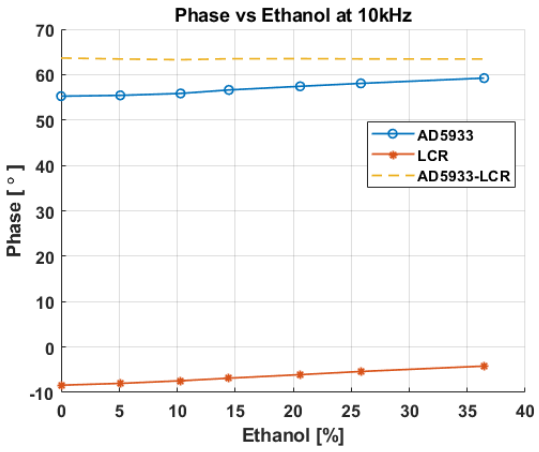


Figure 2.15: Amplitude vs Ethanol at 10kHz.

Figure 2.14 shows that the amplitude measurements from both the AD5933 and LCR meter are in agreement, indicating that the AD5933 accurately measured the impedance. However, as shown in Figure 2.15, the phase angles measured by the AD5933 and LCR meter were different, although they followed the same trend of increasing angles with

increasing ethanol percentages. Nonetheless, this discrepancy did not pose a significant issue, as I was able to introduce a phase shift, similar to what I did for the LCR measurements, to increase the sensitivity of the real part of the impedance. To determine the optimal frequency for this phase shift, I used the algorithm described in section 2.3, which calculated a phase shift of 119° as yielding the highest sensitivity for the real part of the impedance, with a value of 61.93 $\frac{\Omega}{\%}$. Figure 2.16 shows the fitting of the real part with a 119° phase shift. The slope matches the slope of the LCR measurement in Figure 2.12, confirming that the AD5933 can accurately measure the ethanol concentration. Later it has been found that the real behavior is not linear but almost exponential.

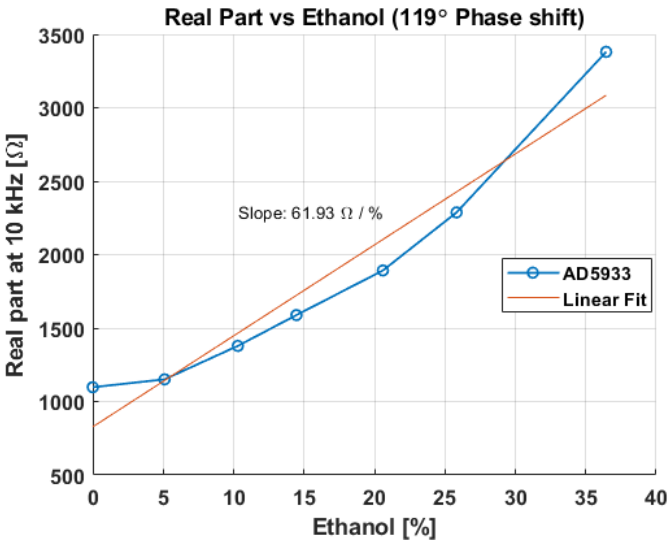


Figure 2.16: Linear Fitting of Real part with 119° Phase Bias

Chapter 3

Unit Concept and Design

This chapter describes the concept and the design of the Impedance Measuring device that I manufactured. First, we will get a general understanding of the core principles of the system and later each part of the device and its implementation will be described.

3.1 System Concept

This section will describe the block structure.

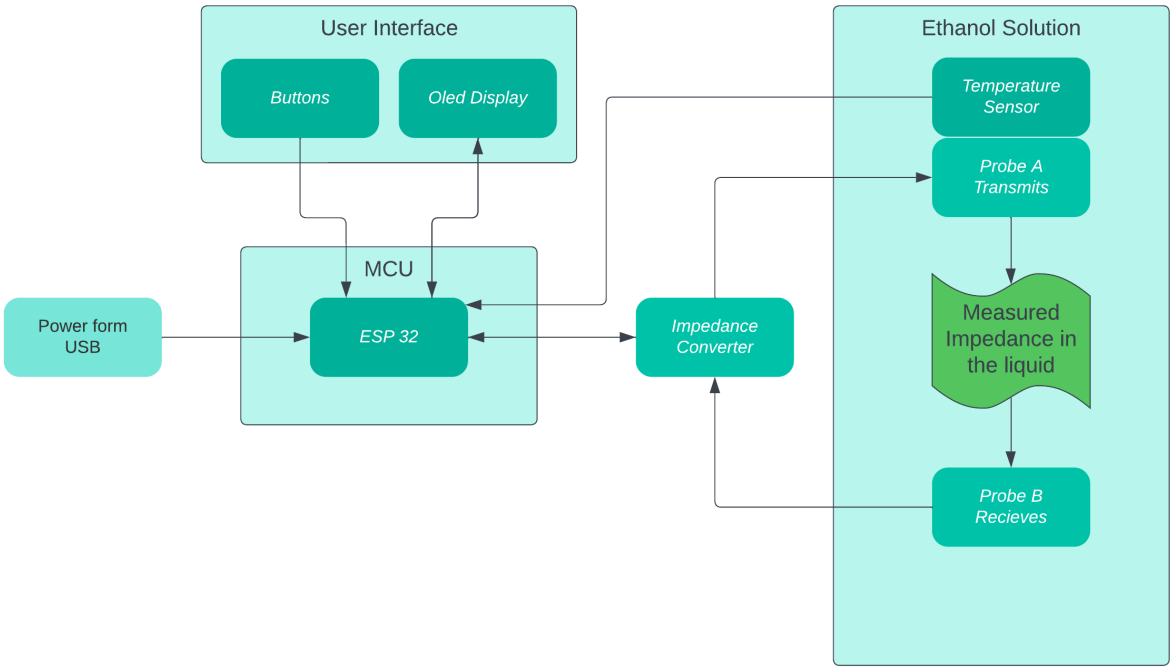


Figure 3.1: System Concept Block Diagram.

Figure 3.1 shows the block diagram that represents a simplified core principle of the device. In the actual design, the device and probe are designed to measure the impedance

at different depths within the solution. A complex probe equipped with pads at various heights allows for the measurement of impedance at each specific depth. In addition, a multiplexer and demultiplexer are integrated into the design to selectively connect the pads at each depth.
At the core of the device, is the ESP32 DevKit C v4 which serves as the MCU that will control the entire system. The ESP32 is powered by a standard 5V USB connection and is equipped with an onboard voltage regulator, eliminating the need for additional power supply circuits on the device.
An impedance converter is controlled by the MCU and generates an AC voltage which is sent to one side of the probe. When the probe is submerged in the ethanol solution, the AC voltage will pass to the other side of the probe and back to the impedance converter where it will be digitally converted. Additionally, the probe is equipped with a temperature sensor. The MCU reads the temperature from this sensor and it enables an algorithmic adjustment of the ethanol concentration estimation based on temperature.
The user interface includes 3 tactile switches to allow users to navigate menus, select options, and initiate specific operations. A 1.3" OLED display provides visual feedback, presenting menu options and measurement results.
By integrating all the components into a single custom PCB and utilizing the ESP32 and display as shields housed on top, the device can achieve a compact design.

3.2 System Design

This section presents the electrical design of the system. First, the functionality and key features of the AD5933 IC will be presented to provide a comprehensive understanding of its operation. Following that, a simplified circuit (without power or signal wiring) of impedance measurement will be presented to provide a broad overview of the most robust part of the device. Subsequently, the specific components and their connections will be detailed to provide a comprehensive understanding of the system. The full schematics are included in Appendix A.
The chosen software for the schematic design was KiCAD 7.0. It offers schematics and PCB editors as well as 3D and Gerber file viewers while being free and open source.

3.2.1 Impedance Converter

The most crucial component of the impedance meter is the impedance converter. The AD5933 is a highly accurate impedance converter IC packaged in a 16-lead SSOP. The AD5933 integrates an onboard frequency generator with a 12-bit, 1 MSPS Analog to Digital Converter (ADC). It can generate excitation voltages up to 100kHz and measures up

to 10MΩ with an accuracy of 0.5% [1]. In addition, the IC offers an internal oscillator clocking at 16.776 MHz and communication via I^2C bus.

The frequency generator allows an external complex impedance to be excited with a known frequency. The response signal from the impedance is sampled by the on-board ADC and a Discrete Fourier Transform (DFT) is processed by an on-board Digital Signal Processor (DSP) engine. The DFT algorithm returns real and imaginary data at each output frequency. Figure 3.2 depicts the internal operation of the AD5933.

Throughout the measurement process, the AD5933 plays a crucial role in generating the excitation signal, acquiring the impedance response, and performing necessary signal processing to obtain impedance data. Its high precision, versatile frequency range, and integrated DSP capabilities make it an ideal choice for impedance measurement applications.

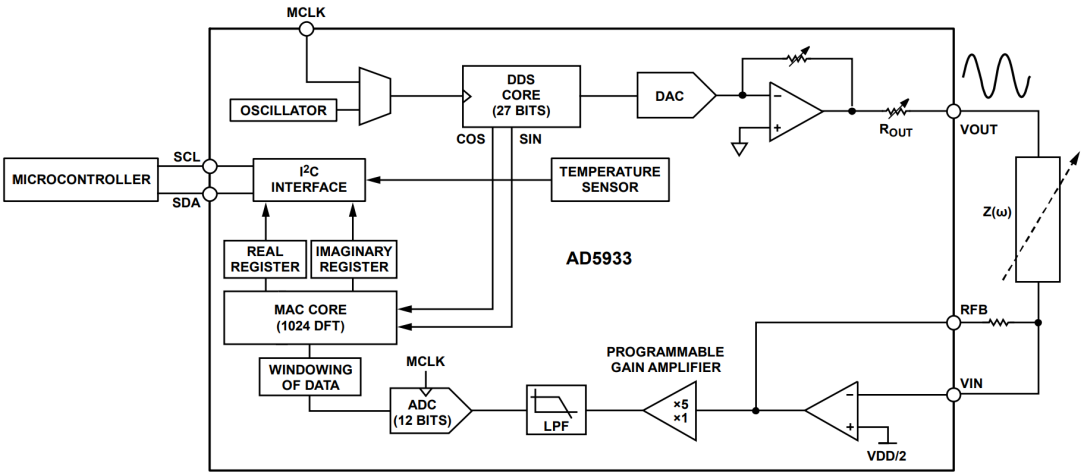


Figure 3.2: AD5933 Function Diagram[1]

The operation of the AD5933 can be split into two major stages.

Transmit stage:

A 27-bit phase accumulator Direct Digital Synthesis (DDS) core provides the output excitation signal at a particular frequency. The input to the phase accumulator is taken from the contents of the start frequency register. The frequency sweep is fully described by the programming of three parameters: the start frequency, the frequency increment, and the number of increments. Once these three parameter values have been programmed, the sweep is initiated by issuing a start frequency sweep command to the control register. The excitation voltage amplitude can be set to four different ranges as described in Table 3.1.

Receive stage:

The receiving stage comprises a current-to-voltage amplifier, followed by a programmable gain amplifier (PGA), an antialiasing filter, and ADC. The unknown impedance is con-

nected between the V_{OUT} and V_{IN} pins as depicted in Figure 3.2. The current-to-voltage amplifier configuration means that a voltage present at the V_{IN} pin is a virtual ground with a dc value set at VDD/2. The signal current that is developed across the unknown impedance flows into the V_{IN} pin and develops a voltage signal at the output of the current-to-voltage converter. The gain of the current-to-voltage amplifier is determined by the feedback resistor connected between Pin RFB and Pin V_{IN}. It is important to choose a feedback resistance value that maintains the signal within the linear range of the ADC (in this case 0V-3.3v). The digital data from the ADC is passed directly to the DSP core, which performs a DFT on the sampled data. The result is stored in two, 16-bit registers representing the real and imaginary components of the result.

The impedance calculation is a two-step process. First, the magnitude is calculated using the real and imaginary components as follows:

$$Magnitude = \sqrt{R^2 + I^2} \tag{3.1}$$

Then it is necessary to multiply the magnitude by a scaling factor called gain factor [1] as follows:

$$Impedance = \frac{1}{Gainfactor \times Magnitude} \tag{3.2}$$

The gain factor is calculated during the calibration of the system with a known impedance connected between the V_{OUT} and V_{IN} pins. By measuring a known resistance we can calculate the Magnitude and then calculate the gain factor using Equation 3.3:

$$Gain\ factor = \frac{1}{Impedance \times Magnitude} \tag{3.3}$$

Once the gain factor has been calculated, it can be used in the calculation of any unknown impedance between V_{OUT} and V_{IN}.

To calculate the phase of the impedance, the arc-tangent function can be applied to the ratio of the imaginary and real numbers measured by the AD5933 as follows:

$$Measured\ phase = arctan{-}1(\frac{Imaginary}{Real}) \tag{3.4}$$

However, the entire system introduces an internal phase shift or lag. In order to determine the actual phase of the unknown impedance, it is necessary to calculate the system phase. This calculation is performed during the calibration sequence. When a known resistor is connected between V_{OUT} and V_{IN} no additional phase lead or lag is introduced to the system. Therefore, at each measurement point during calibration, the

measured phase corresponds to the system phase.

After the calibration process is completed, the actual phase of the unknown impedance can be determined using Equation 3.5:

$$Phase = Measured\ phase - System\ phase \tag{3.5}$$

where the measured phase refers to the phase obtained from the impedance measurement, and the system phase represents the phase introduced by the internal components and circuitry of the system during calibration.

Table 3.1: Output Excitation Voltage Ranges, DC Bias, Series Resistance

Range	Output Excitation Amplitude [Vp-p]	Output DC Bias Level [V]	Output Resistance R_{Out} [Ω]
1	1.98	1.48	200
2	0.94	0.76	2400
3	0.383	0.31	1000
4	0.198	0.173	600

3.2.2 Simplified Circuit

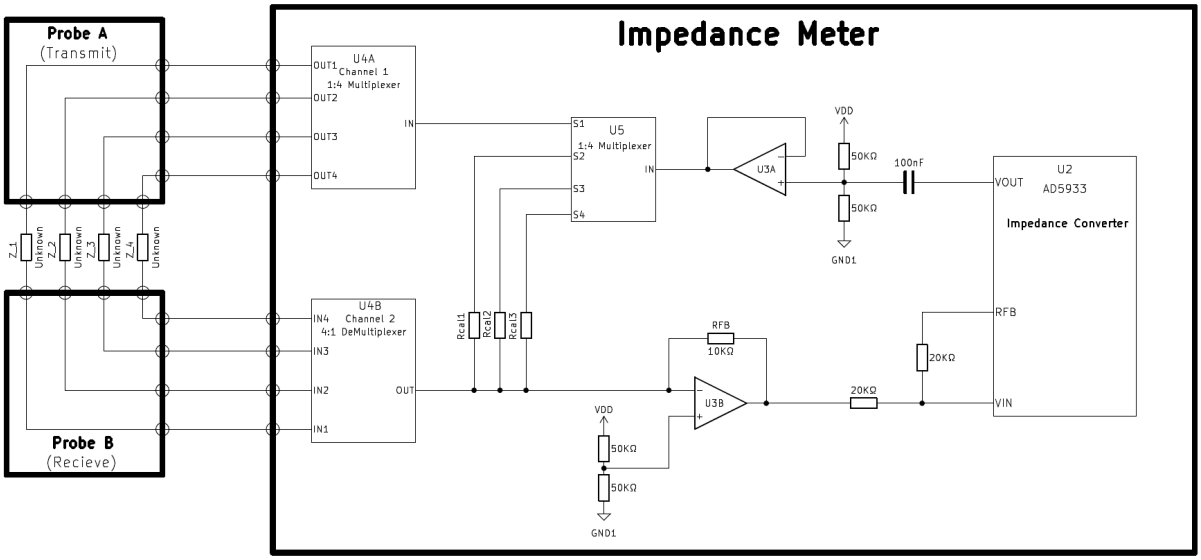


Figure 3.3: Simplified Circuit of Impedance Measurement.

Figure 3.3 depicts a simplified version of the impedance measurement circuit i.e. without power and signal connections.

There are several important features inherited from the AD5933 IC design that needed to be addressed. Table 3.1 shows that each excitation voltage range generated by the

AD5933 has a unique DC bias and output resistance R_{Out}. It is crucial to those variables into account when designing the impedance meter as they can greatly affect the measurements. For example, if the device has to measure impedance at the low kΩ range, R_{Out} of 2.4kΩ will render those measurements completely useless.
To address this issue, it is necessary to incorporate a voltage follower in the design using a high-quality operational amplifier with low output impedance and unity gain stability. This will ensure that the measured impedance reflects the true impedance of the solution being tested.
Furthermore, Figure 3.2 shows that the positive input of the internal receiving amplifier of the AD5933 is biased to VDD/2 (1.65V in this case), while the excitation voltages are biased at different DC values. This voltage difference between the transmitting and receiving sections of the circuit can lead to polarization of the impedance under test, including the probe pads and the liquid between them. Such polarization can introduce additional inaccuracies in the measurements.
The solution is to match the DC bias of the excitation voltage to VDD/2. A simple solution is to add a high pass filter before the voltage follower to eliminate the DC component of the excitation voltage. Additionally, biasing the voltage follower to VDD/2 will ensure that the excitation voltage passing through the liquid is at the same DC bias as the receiving section, while maintaining minimal output resistance. The chosen capacitor for the high-pass filter was 100nF with a cutoff frequency f_{Cutoff} of 31.8 Hz which ensures that any practical AC signal the AD5933 sends, passes. f_{Cutoff} given by:

$$f_{\text{Cutoff}} = \frac{1}{2\pi RC} \tag{3.6}$$

In addition to that, the internal I-V amplifier of the AD5933 can also add inaccuracies. The conversion at this stage is sensitive to the bias current and offset voltage of the internal amplifier. Therefore providing an external high-performance amplifier those inaccuracies can be mitigated. The RFB resistor is moved to the external amplifier and the I-V conversion is completed externally. Then by adding two identical resistors the internal amplifier of the AD5933 serves as a simple inverting amplifier with gain 1.

In order to facilitate calibrations across different impedance ranges, a 4-way Multiplexer (MUX) is utilized in the system. The MUX receives the signal from the voltage follower and selectively routes it accordingly. In the measure mode, the signal is passed further for impedance measurement. However, in the calibration mode, the signal is directed to a chosen calibration resistor and then forwarded to the I-V amplifier for calibration purposes. More about the calibration MUX in Section 3.2.5.

Lastly, to enable impedance measurements at various depths using the gradient probe, two 4-channel multiplexers are incorporated into the system. One of the multiplexers functions as a signal selector, receiving the signal from the calibration Multiplexer (MUX) and directing it to the chosen pad on the transmitting part of the probe. The other multiplexer operates as a demultiplexer, receiving the signal from the corresponding pad on the receiving part of the probe and subsequently routing it to the I-V amplifier. More about the probe MUX in Section 3.2.6.
The simplified circuit does not include the temperature measurement component. Instead, the temperature sensing is directly performed by the Micro-Controller Unit (MCU). more on that in Section 3.2.7.

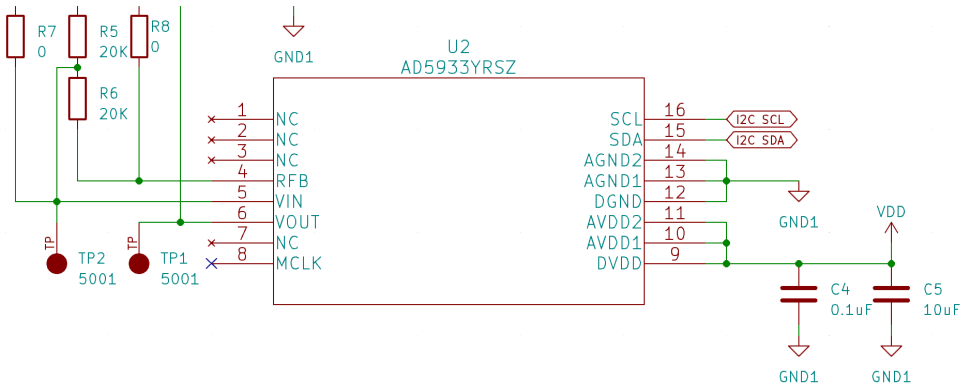


Figure 3.4: AD5933 Schematic Connections

Figure 3.4 shows the schematics of the AD5933. The power supply is bypassed with 10 µF and 0.1 µF capacitors to ensure stable operation [1]. Additionally, two test points are included to facilitate monitoring of the output and input voltages.
Notably, the schematics feature two 0Ω resistors R7 and R8. These resistors are intentionally left unconnected and not soldered, serving as pads that can be easily shorted if needed. They provide a convenient way to bypass the external I-V amplifier for troubleshooting purposes if needed.

3.2.3 ESP32 Devkit C V4

The ESP32 DevKit C v4 is a small-sized ESP32-based development board developed by Espressif company. However, the board utilized in the impedance meter is produced by AZ-Delivery company and corresponds 100% to the layout of Espressif. This board employs dual-core 32-bit Xtensa processors and has 520KB of SRAM, 36 GPIOs and supports various communication buses such as UART, SPI, and I^2C[4]. In addition, it has onboard 12-bit ADC and voltage regulator which simplifies the design. Figure 3.5 shows

the ESP32 Devkit.

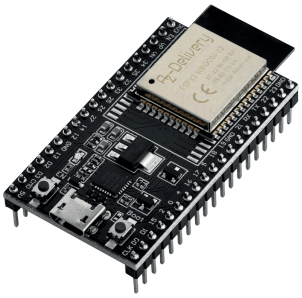


Figure 3.5: ESP32 Devkit C V4 [5]

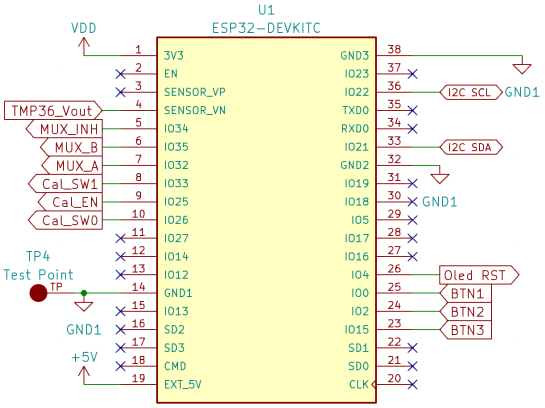


Figure 3.6: ESP32 Devkit C V4 Schematic Connections

The devkit board itself will be housed like a shield on the custom PCB. The ESP32 operates at 3.3V, this will be the supply voltage VDD for all the ICs on the custom PCB. The ground of the ESP32 will be the whole system's ground. A test point will be connected to the ground.
The onboard 12-bit ADC will be utilized for reading temperature from the TMP36 sensor, as detailed in section 3.2.7. The dedicated I^2C bus GPIOs will be used to control and communicate with both the AD5933 impedance converter and the OLED display. Some GPIOs will be used as digital outputs to control the multiplexers while others as digital inputs to read the buttons. Figure 3.6 shows the ESP32 schematic connections.
ESP32 WROOM-32 CPU doesn't support USB communication natively. However, the devkit has a CP2102 chip on board to handle the communication. The ESP32 is programmed via USB, more on that in section 5.1.

3.2.4 AD8606

As discussed in section 3.2.2, the design of the impedance meter required two operational amplifiers: one for voltage follower functionality to re-bias the excitation voltage, and the other for the external I-V amplifier.
For this, I found the AD8606ARMZ more than suitable due to its exceptional characteristics and performance. The AD8606ARMZ is a precision dual operational amplifier with rail-to-rail input and output capabilities. It is packaged in an MSOP-8 package.
It offers a low input bias current (1 pA maximum), and low offset voltage (65 µV maximum), reducing the impact on the measurement accuracy. More importantly, the AD8606 offers very low output impedance at frequencies between 1kHZ to 100kHz of 1Ω at gain

10 and below 20Ω at gain 100 [6]. Allowing for low-impedance measurements.

Figure 3.7 depicts the schematic connections of the AD8606. The power supply is bypassed with 10 µF and 0.1 µF capacitors to ensure stable operation, as recommended in the datasheet [6]. The lowest voltage level is 0V, connected to the V- (negative supply) terminal. The highest voltage level is 3.3V, connected to the V+ (positive supply) terminal. A test point is added in order to monitor the rebiased excitation voltage. Furthermore, resistors R7 and R8, described in Section 3.2.2, serve as pads that can be easily shorted if necessary, offering a convenient means for bypassing the external I-V amplifier during troubleshooting.

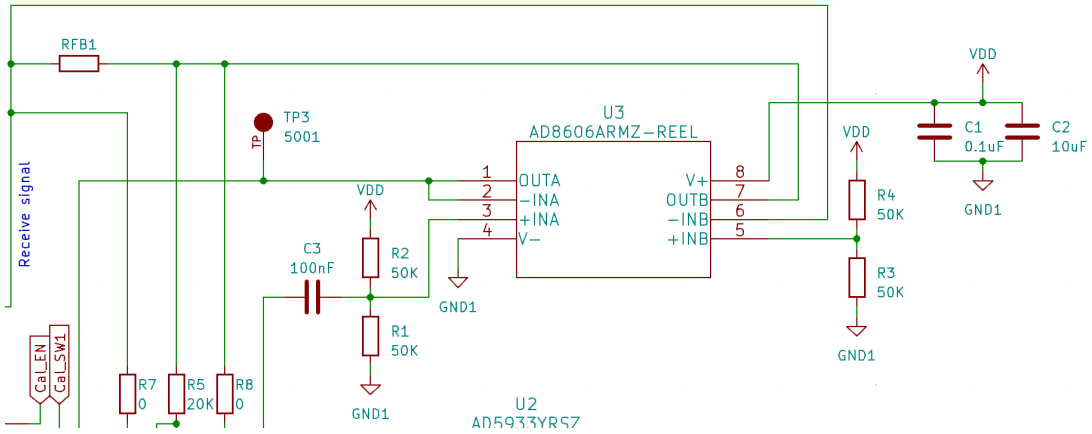


Figure 3.7: AD8606ARMZ Schematic Connections

3.2.5 Calibration Multiplexer

In order to calibrate the AD5933 onboard I had to find a suitable multiplexer that will selectively pass the excitation voltage to several calibration resistors, which will be connected to the receiving amplifier(U3B). I found the ADG704 suitable for the task as it has very low R_{ON} (8Ω max On-resistance at 3V supply) and will barely affect the calibration. The ADG704 is a 4-channel CMOS multiplexer and it comes in a compact 10-lead µSOIC package. The typical power requirement is 0.001 μA with a maximum draw of 1μA [7].

The ADG704 features one common input D and four outputs S1-S4. The output selection is determined by the 3-bit binary address logic lines A0, A1, and EN. When EN is set to logic 0, the device is disabled, and when it is set to logic 1, the device is enabled. The specific output that is switched on is determined by the logic levels of A0 and A1, as seen in table 3.2. The signal lines will be connected to the MCU as shown in Figure 3.6.

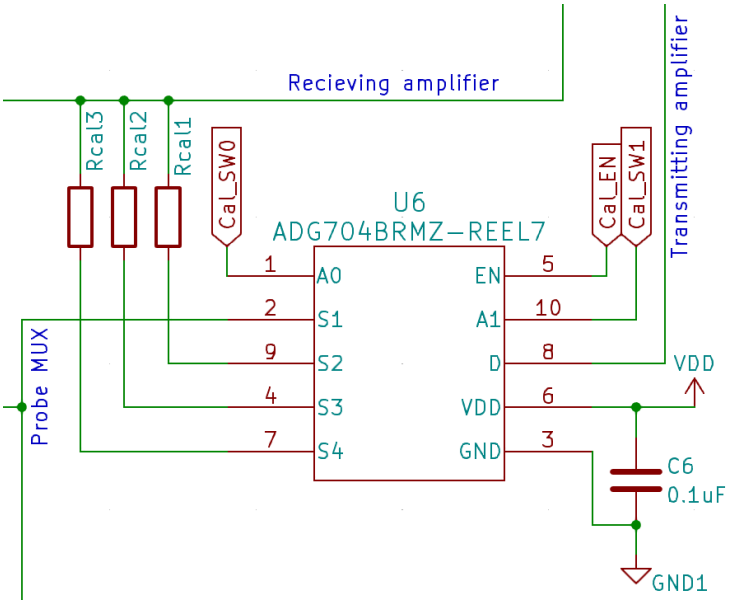


Figure 3.8: ADG704 Schematic Connections

Table 3.2: ADG704 Truth Table [7]

A1	A0	EN	ON Switch
X	X	0	NONE
0	0	1	1
0	1	1	2
1	0	1	3
1	1	1	4

Figure 3.8 shows the schematic connections of ADG704. Pin D is connected to the transmitting amplifier which passes the re-biased AC voltage from AD5933. Output pin S1 forwards the signal to the input of channel 1 of the probe multiplexer, while pins S2-S4 direct the signal through calibration resistors R_{CAL1}-R_{CAL3} and back to the receiving amplifier, respectively. In addition, a 0.1 µF bypass capacitor was included in the circuit. The specific values of the calibration resistors are left to be determined during the implementation stage of the project. This approach allows for flexibility and adjustment based on any changes or fine-tuning required during the actual implementation process.

3.2.6 Probe Multiplexer

The probe has 8 pads in total, with 4 pads on the transmitting side and 4 pads on the receiving side, forming 4 pairs of pads at each depth. To selectively route the excitation voltage to a specific pair of pads at a desired depth, a multiplexer and demultiplexer solution were required. However, instead of using separate multiplexers, a convenient solution was to use a dual 4-channel multiplexer. This solution is less complex and cost-efficient. The SN74LV4052APW is a dual 4-channel multiplexer/demultiplexer IC available in a

TSSOP(16) package. It has 2 inputs and 8 outputs such that half of them pass signal from one input and the other half pass signal from the second input. The IC operates on supply voltages between 2V to 5.5V, has relatively low R_{ON} resistance (typically around 63Ω) [8]. These characteristics make the SN74LV4052APW an ideal choice for the Impedance meter.

A very convenient feature of the SN74LV4052APW is that both channels are controlled by the same 3-bit binary address logic lines: INH, A, and B. This means that when the MUX is instructed to pass a signal from a specific input to a certain output, the corresponding output on the second channel will be activated for the second input as well. This synchronization simplifies the control logic and ensures that both probe pads on a certain depth will be active during excitation. Table 3.3 represents the 3-bit signal truth table of the SN74LV4052APW.

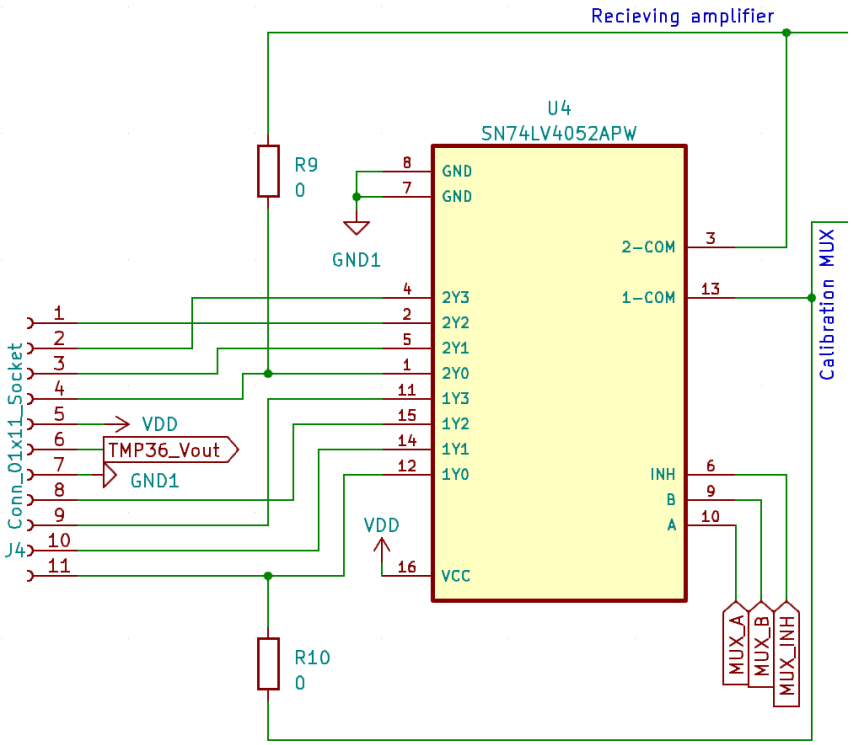


Figure 3.9: SN74LV4052APW Schematic Connections

Figure 3.9 shows the schematic connections of SN74LV4052APW. Pin 1-COM is connected to the calibration MUX ADG704 which passes the excitation voltage. When the SN74LV4052APW is instructed to pass a signal to a certain output, the excitation voltage passes to the corresponding pad on the transmitting part of the probe while simultaneously allowing the voltage to pass back through the pad at the same depth on the receiving part of the probe. The voltage then passes through pin 2-COM to the receiving amplifier. Additionally, the schematics show two 0Ω resistors R9 and R10. These resistors

Table 3.3: SN74LV4052APW Truth Table[8]

INPUTS			ON CHANNELS
INH	B	A	
L	L	L	1Y0, 2Y0
L	L	H	1Y1, 2Y1
L	H	L	1Y2, 2Y2
L	H	H	1Y3, 2Y3
H	X	X	None

are intentionally left unconnected and not soldered, serving as pads that can be easily shorted if needed. They provide a convenient way to bypass the SN74LV4052APW for troubleshooting purposes, allowing the signal to be directly routed to the probe pads at the lowest depth.
The schematics show the wiring of the temperature sensor which goes to the probe.

3.2.7 Temperature Sensor

To measure the temperature of the liquid solution I had to find a suitable temperature sensor. I didn't want to use an external temperature or just submerge an insulated "through hole" sensor. The most obvious choice was to attach a small temperature sensor to the probe itself. My search led me to the TMP36.
The TMP36 is a low-voltage, precision centigrade temperature sensor produced by Analog Devices. It provides a voltage output that is linearly proportional to the Celsius temperature. It comes in different packages and I chose Lead Small Outline Transistor Package (SOT-23 since it's easier to solder.

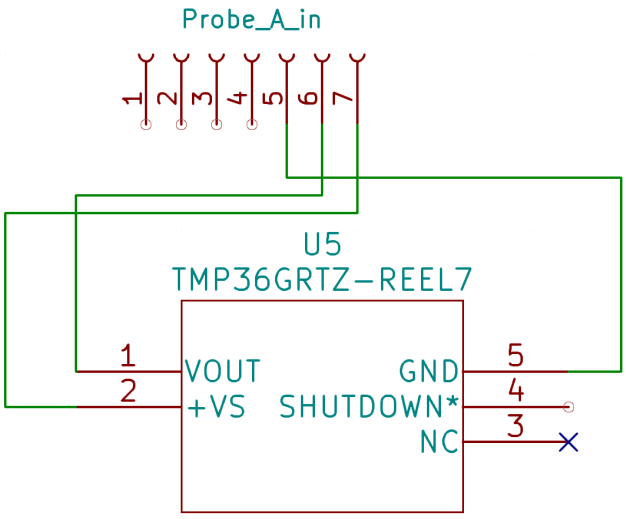


Figure 3.10: TMP36 Schematic Connections

The sensor can operate with a supply of 2.7V - 5.5V which suits the 3.3V VDD supplied by ESP32. The current supply runs well below 50μA[9]. The TMP36 is capable of measuring temperatures from -40°C to 125°C [9], which is great for the purpose of measuring cold liquids. 10°C is the typical minimal temperature similar sensors can achieve. The output scale factor of the TMP36 is 10mV/°C. However, it has an offset voltage of 0.5V, and therefore at 25°C, the output is 750mV instead of the expected 250mV. Additionally, the TMP36 does not require any calibrations [9]. I had to verify that the onboard ADC has enough resolution to measure the temperature effectively. The onboard 12-bit ADC operates at 0V - 3.3V meaning the Vrange is 3.3V. The Least Significant Bit (LSB) is equal to the voltage range divided by the total number of codes ($\frac{V_{\text{range}}}{2^n}$). In this case, the LSB is 0.8 mV which is enough for temperature measurements at 0.1°C resolution.
The sensor will be positioned on one of the probe's PCBs facing outward and all the connections will be routed with the probe's connections. The schematic connection is shown in Figure 3.10.

3.2.8 OLED Display

Displays are crucial for providing visual feedback, presenting menu options, and measuring results. There are various types of displays available ranging from alphanumeric Liquid Crystal Display (LCD)s to graphic displays. For the impedance meter, I selected a 1.3" Monochrome Organic Light-Emitting Diode (OLED) display breakout by Adafruit shown in Figure 3.11.
I chose this display for several reasons. Monochrome OLED displays are more power efficient since they do not require a light source like LCD's. The 1.3" size fits perfectly within the impedance meter device while still offering a good readability. The breakout board includes all the necessary drivers and circuitry. It features the popular SSD1306 driver chip produced by Solomon Systech, which provides seamless integration and easy control of the display. In addition, the Adafruit SSD1306 [10] and Adafruit GFX [11] libraries provide robust support for driving this display, ensuring convenient and efficient development. In terms of communication interfaces, the Adafruit Organic Light-Emitting Diode (OLED) breakout is capable of both SPI and I²C protocols. However, considering that the AD5933 is already connected to the I²C bus, I chose the I²C interface for the OLED display.

The display required only power, SDL, SCL, and Reset connections. Figure 3.12 shows the schematic connections of the display. The software implementation of the display is described in chapter 5.

Figure 3.11: Adafruit 1.3" OLED Breakout [12]

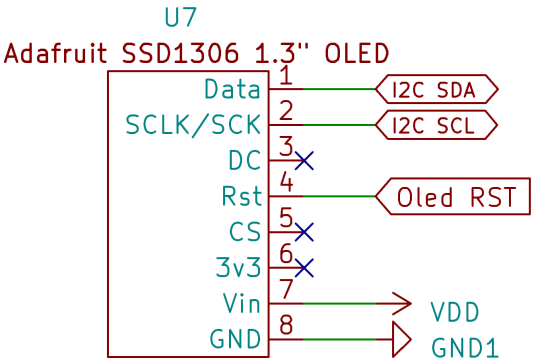


Figure 3.12: Display Schematic Connections

3.2.9 Tactile switches

To control the device, navigate the menu, and perform operations, a setup with three SMD tactile switches by TE Connectivity was implemented. The first switch functions as the "Select" button, the second switch is used for scrolling, and the third switch acts as the "Back" button. The tactile switch consists of four pads. Two pads within each switch are internally connected, while the other two pads are also connected, but the two pairs are isolated from each other. When a switch is pressed, it bridges the two pairs of pads, establishing a connection. This connection allows the desired functionality to be triggered and enables the corresponding action to be performed.

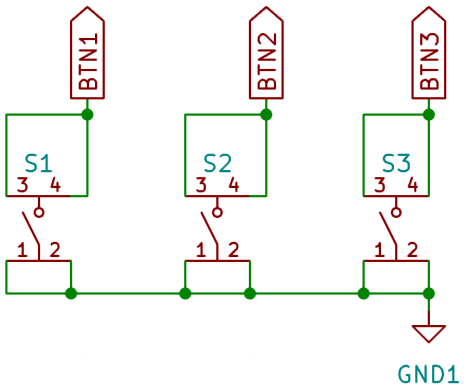


Figure 3.13: Tactile Switcher Schematic Connections

To integrate tactile switches with the MCU, it is important to consider the issue of floating and its unpredictable behavior. When a switch is not pressed, the signal line can be left floating, resulting in an undefined state. To ensure reliable input detection and prevent this unpredictability, a pull-up or pull-down resistor circuit can be used as illustrated in Figure 3.14. These resistors provide a defined voltage level when the switch is not pressed while limiting the current, effectively eliminating the floating condition.

The ESP32 has built-in pull-up resistors and can be enabled in software which simplifies the design further. I chose a pull-up configuration since the ground will be more easily accessible on the PCB. This meant that the logic level at the GPIO will be 1 when the switch is not pressed and 0 when it's pressed. The schematic connection of the tactile switches is seen in Figure 3.13.

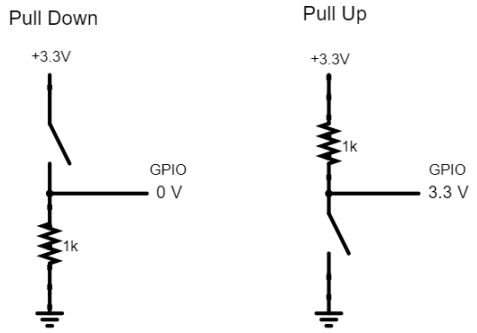


Figure 3.14: Pullup/Pulldown Circuits

3.3 PCB Design

One of the main objectives in developing the impedance meter unit was to create a custom PCB that could accommodate all the necessary components while maintaining a compact form factor and a visually appealing appearance. A 2-layered PCB was enough. The overall size of the PCB was determined based on the dimensions of the MCU, display, and tactile switches, as all three components would be located on the top of the PCB and other parts. Considering that the assembly would be done manually, I chose 0805 to be the size of the Surface Mount Devices (SMD) resistors and capacitors. This size strikes a balance between being relatively small and still manageable for hand soldering.
In the PCB design I utilized traces with different thicknesses: 0.3mm for the main wires and 0.2mm for the logic signal traces. All bypass capacitors were placed as close as possible to their designated pads. Whenever it was possible GND and VDD were connected via multiple connections. The final dimensions of the PCB are 100mm x 38mm. All the PCBs full schematics can be seen in Appendix A.

3.3.1 Device PCB

All the ICs, resistors, capacitors, and tactile switches are placed on the top layer of the PCB. The EPS32 and display are connected via pin headers and are placed above the other components. Figure 3.15 illustrates the placement of each major component.

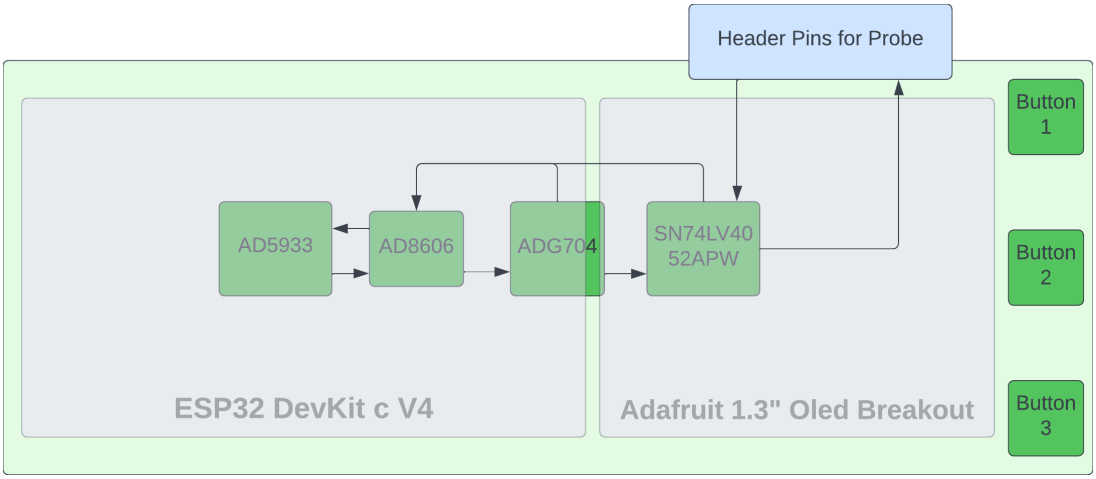


Figure 3.15: PCB's Block Diagram (ESP32 and Display are above the other components)

Both layers on the PCB are based on ground planes to ensure proper grounding. To minimize the occurrence of non-grounded islands, vias were placed to connect these islands to the opposite grounded layer. Efforts were made to minimize trace lengths, but due to the small size of the PCB and the close proximity of the components, it wasn't a major concern.

The ESP32 and the display feature 2.54mm pitch male pin header connectors. To accommodate these connectors, matching 2.54mm pitch THT female headers were incorporated into the design.

As described in Section 3.3.2, the probe features 1.27mm pitch male headers, therefore I decided to incorporate a right-angled 1.27mm pith THT male pin headers on the main board as well (11 in total). This choice not only reduces bulk from the connector but also ensures that the same type of connector is used on both sides, thereby maintaining continuity throughout the design. There is a silkscreen marking of the headers, 'A' for transmitting the probe and 'B' for receiving the probe.

The wiring between the probe and the main board will follow this configuration: The main board houses all 11 pins, which will connect to a matching female header. From this female header, 7 wires will be routed to the transmitting side of the probe, carrying the 4 signal wires to the pads, power supply to the temperature sensor, and Vout of the sensor. The remaining 4 wires will carry the signal from the receiving side of the probe. Figure 3.16 and Figure 3.17 illustrate the top and bottom PCB layers, respectfully.

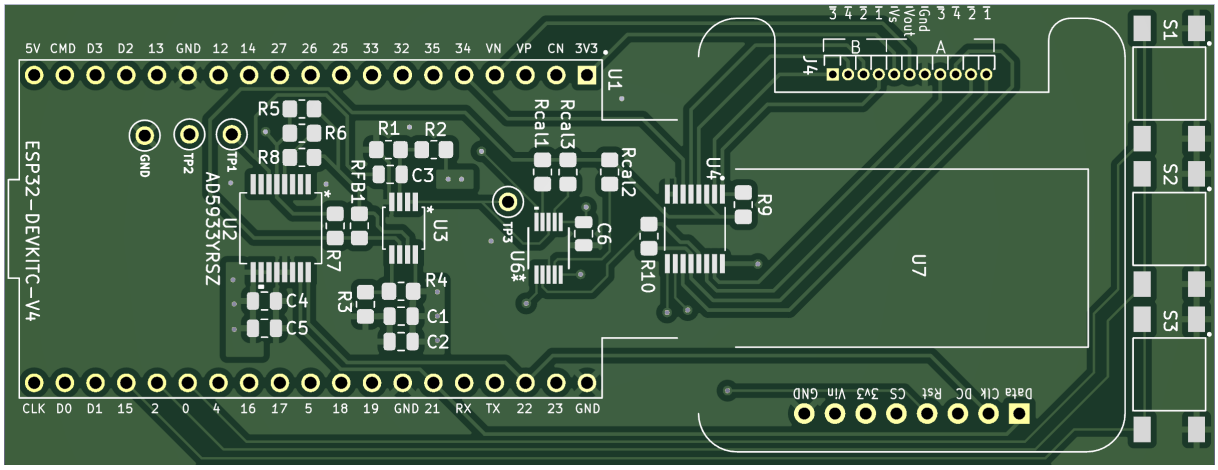


Figure 3.16: Render of PCB's top layer

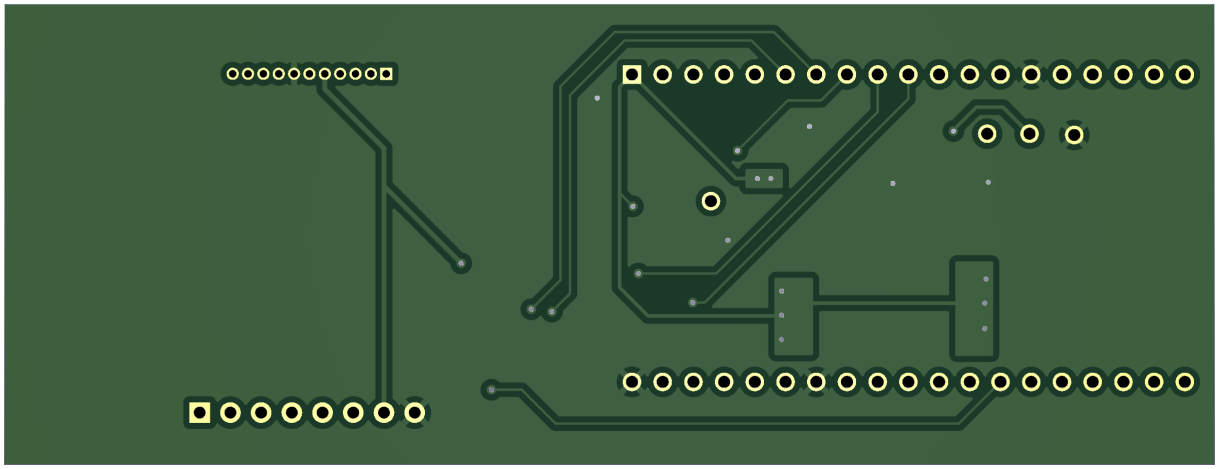


Figure 3.17: Render of PCB's bottom layer

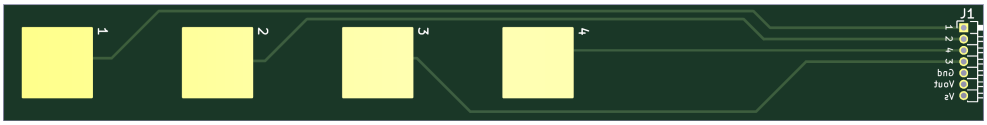
3.3.2 Probe PCB

The probe consists of two identical PCBs that are positioned with their top layers facing each other. The PCB schematic is rather simple.

On the top layer of the probe PCB illustrated in Figure 3.18a, there are four 0.8mm x 0.8mm pads, with each pad connected to its respective pin header. These pads serve as contact points with the liquid.

On the bottom layer of the probe PCB illustrated in Figure 3.18b, there is a designated area for the temperature sensor which is directly connected to the pin headers, allowing for temperature measurements to be incorporated into the impedance measurements. In order to protect the sensor it will be coated with epoxy.

The temperature sensor is integrated only on one of the probe's PCB which is considered the transmitting side of the probe.



(a) Top layer

(b) Bottom layer

Figure 3.18: Render of Probe's PCB

Chapter 4

Implementation

4.1 Physical Implementation

This Section delves into the practical implementation of the designed impedance meter. The implementation strategy followed a systematic approach, focusing on one component at a time. Each component was individually implemented, including soldering, microscopic evaluation of the soldering, code development, testing, and troubleshooting. All the software-related aspects are discussed in detail in Chapter 5.

4.1.1 Temperature sensor

This temperature sensor was chosen as the starting point due to its relative simplicity and straightforward implementation. The sensor was soldered onto its designated pads on one of the probe's PCBs. Additionally, header pins were soldered onto the probe, to ensure the needed connections with ESP32. The power supply connections for the sensor (VDD and GND), were routed directly from the ESP32. The Vout pin of the TMP36 was connected to pin 39 (labeled as VN) on the ESP32.
A short script was developed for reading the analog signal from the TMP36 and converting it to voltage using the calculated LSB value, as outlined in Section 3.2.7. However, upon initial testing, the obtained readings did not correspond to the expected values. A meticulous review of the TMP36 datasheet[9] was conducted during the troubleshooting process. It was during this analysis that a critical detail came to light. The TMP36 has a shutdown function, which allows a reduction of the supply current. While the shutdown feature was considered optional during the initial design phase, a closer examination of the datasheet revealed a crucial recommendation: "If the shutdown feature is not used, it is recommended that the SHUTDOWN pin be connected to +Vs." Addressing this revela-

tion, a straightforward solution was implemented. A thin enamel-coated copper wire was delicately soldered to connect the +Vs and Shutdown pins of the TMP36. This modification ensured the proper operation of the temperature sensor. Subsequent readings were consistent, with voltage measurements aligning with the expected values. Notably, the measured voltages of around 700mV indicated an ambient temperature of approximately 20°C. When the sensor was blasted with hot air the voltage readings rose up as expected. This experience taught me the importance of thoroughly examining datasheets when designing and implementing a system. It highlighted the fact that even small details, such as the shutdown function of a sensor, can significantly impact the performance and reliability of the system.

Despite achieving successful and consistent temperature readings, the signal periodically spiked. To investigate this further, I decided to verify the readings using a digital multimeter. By connecting the multimeter to the Vout of the temperature sensor and GND, I obtained readings that consistently were slightly different than those obtained by the microcontroller unit (MCU). Notably, the multimeter readings did not exhibit any spikes. This discrepancy raised concerns regarding the MCU or its associated GPIO (General Purpose Input/Output) configuration. To investigate this, I devised a simple experiment. I connected the GPIO pin to the positive terminal of a stable power supply, with the negative terminal connected to the ESP32's ground. The objective was to supply a known and stable voltage to the GPIO pin and observe the resulting readings. Surprisingly, the converted voltage readings were incorrect.

To troubleshoot this issue further, I modified the script to display the raw analog readings instead of the converted voltages. The ESP32 is equipped with a 12-bit ADC, capable of converting voltages in the range of 0V to 3.3V. Therefore, the expected range for the analog readings should fall between 0 and 4095. the ADC was reaching saturation with voltages lower than the expected 3.3V. This discovery suggested that the ADC's linear region was not accurately aligned with the 0V to 3.3V voltage range.

To address the issue, I conducted a measurement procedure. Starting from 0V, I gradually incremented the supplied voltage while recording the raw analog readings. By analyzing the data I was able to identify the actual linear region of the ADC. I linearly fitted the data as shown in Figure 4.1 to derive the Equation 4.1 of the relationship between the raw analog readings and the corresponding voltage values. That meant that I can recalculate the voltage based on the analog read given by Equation 4.2.

$$Analog = 1232.8 \times V - 170.19 \tag{4.1}$$

$$V = \frac{Analog - 170.19}{1232.8} \tag{4.2}$$

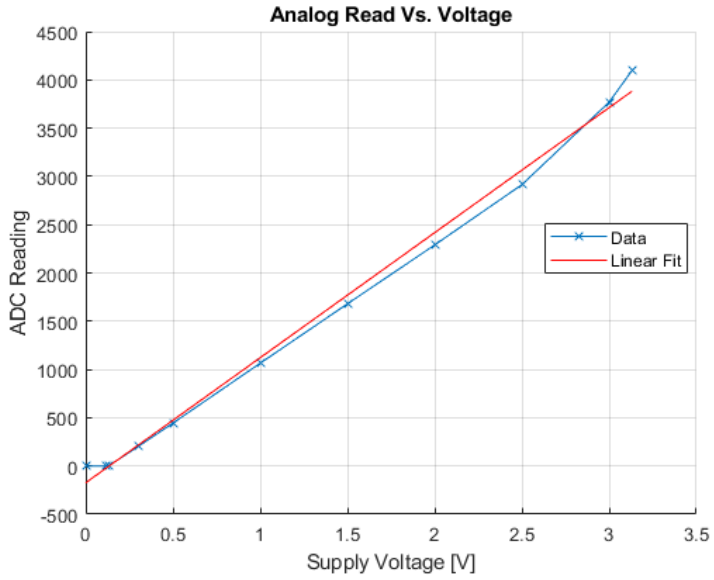


Figure 4.1: ADC's linear region

After updating the correct equation in the script on the ESP32, the readings were consistent and matched the multimeter readings. However, there was still an issue with voltage spikes. Upon further investigation, it was discovered that the spikes were caused by a faulty USB cable used to supply power to the ESP32. By replacing the USB cable with a new one, the voltage spikes were eliminated.
After eliminating the voltage spikes, I updated the script to convert the voltage readings to temperature using Equation 4.3, taking into account the voltage bias of the sensor:

$$Temperature = (V - 0.5) \times 100 \tag{4.3}$$

In order to address the minor noise present in the temperature readings, I recorded 1000 readings and calculated the standard deviation, which was found to be 0.12° C. This standard deviation represents the level of variation or noise in the raw temperature data. To mitigate this noise and improve the stability of the temperature measurements, I decided to implement a running average technique. By applying a 5-sample window for the running average, I observed a significant reduction in the standard deviation, bringing it down to 0.05° C, indicating that the running average effectively smoothed out the temperature readings and reduced the impact of the noise.
After addressing various challenges and implementing necessary modifications, the temperature sensor was finally operational.

4.1.2 Probe Multiplexer

In order to proceed with the implementation of other components, the next step was soldering the female pin headers for the ESP32 board. Additionally, the male headers were soldered for the transmission of the excitation signal to the probe.

Once the headers were in place, I proceeded to carefully solder the SN74LV4052APW multiplexer using a hot air gun. it was crucial to conduct testing to verify their functionality. the plan was to apply a known voltage to the input of the transmitting multiplexer. By selectively controlling the output channels using the ESP32, the voltage on the probe's pads could be measured to assess whether the multiplexer successfully directed the voltage to the desired output channel. Similarly, the receiving demultiplexer underwent a similar testing procedure, where the voltage was sent through the probe's pads, and the resulting output voltage was measured.

I wrote a script to control the SN74LV4052APW multiplexer and performed tests to verify its functionality. The script allowed me to activate and deactivate specific output channels of the multiplexer using the ESP32.

However, I encountered a significant issue with the multiplexer as no signal was passing through regardless of the configuration. I decided to inspect the circuitry more closely. I realized that the ground pins of the SN74LV4052APW were not properly connected to the ground.

Despite the presence of a copper island visually resembling a ground connection, further examination in KiCad revealed that it was not actually connected to the ground. The green highlighted copper in Figure 4.2 shows that for some reason KiCad considered it as ground connected and filled the island with copper, regarding the fact that there were no vias placed there.

This misleading representation during the development process led me to believe that the ground connection was intact, when in fact it was not.

Luckily, the solution to the disconnected ground issue was straightforward. I carefully scraped away the solder mask layer above a nearby ground plane on the PCB and soldered a thin enameled copper wire between that ground plane and the ground pin of the MUX. This ensured a proper electrical connection and restored the functionality of the MUX.

After addressing the ground connection issue, I expected the MUX to function properly. However, I encountered another setback when the MUX still didn't work as expected. To troubleshoot the issue, I decided to examine the GPIOs of the signal lines connected to the MUX (pins 32,34,35). Using a multimeter, I checked the logic levels at the corre-

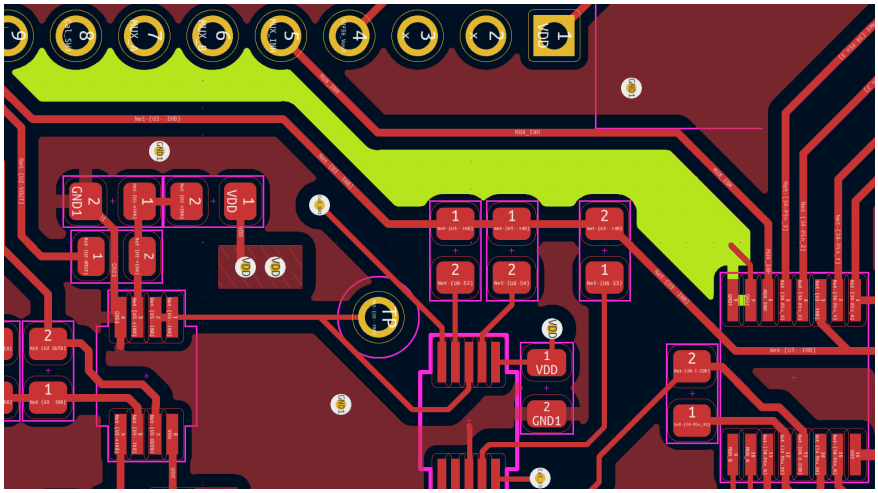


Figure 4.2: SN74LV4052APW False Connection

sponding pins and discovered that only one (pin 32) out of the three pins behaved as it was supposed to.

To gain further insights, I decided to consult a pinout schematic of the ESP32 DevKit from a different source. To my surprise, I found that pins 35, 34, 39, and 36 were designated as input-only pins in the schematic. This revelation contradicted the information provided in AZ-Delivery's datasheet, which depicted these pins as General Purpose Input/Output (GPIO), suggesting their capability for both input and output operations.

The issue was easily resolved by soldering and rewiring the connections of pins 34 and 35 to different pins that were functional. Specifically, pin 34 was successfully rerouted to pin 27, while pin 35 was connected to pin 14. These new pin assignments were then enabled in the software, while pins 34 and 35 were disabled. Following these adjustments, the SN74LV4052APW multiplexer operated flawlessly, achieving the intended functionality.

4.1.3 Calibration Multiplexer

For the integration of the ADG704 multiplexer, I followed a similar approach and testing methodology as I did with the SN74LV4052APW as well as soldering of the bypass capacitor C6. This time, the implementation proceeded smoothly without any issues. The ADG704 multiplexer performed as expected and proved to be fully operational. The calibration resistors were soldered as well, their values were 68kΩ, 100kΩ, and 150kΩ. These resistors were changed during the testing phase to optimize the device for more accurate measurements.

In addition, I decided to integrate the tactile switches at this stage. I soldered them to the PCB and made the necessary updates to the script to enable their functionality. The tactile switches successfully controlled the multiplexers, and the system operated as

intended.

4.1.4 Amplifiers

The next step in the implementation was to solder the AD8606 dual op-amp onto its designated pads on the PCB, as well as soldering the biasing resistors R1-R4 and bypass capacitors C1-C2. To verify the functionality of the amplifiers, the output voltage of both amplifiers was measured and confirmed to be at VDD/2, which in this case was 1.65V. This indicated that the amplifiers were properly biased and operating within the desired voltage range.
However, in order to fully assess the performance of the amplifiers, it was necessary to integrate the AD5933 and observe how the amplifiers behaved with a proper input signal.

4.1.5 AD5933

With all the necessary components in place, I proceeded to implement the AD5933. I soldered it onto the PCB, along with the bypass capacitors C4-C5, high-pass capacitor C3, resistors R5 and R6, and the initial feedback resistor RFB, which was initially chosen to be 100kΩ but later during optimization was changed to 10kΩ. To enable comfortable oscilloscope measurements, I also added the test points on the backside of the PCB for easy connection to an oscilloscope. Taking extra care, I inspected the soldering under a microscope to ensure a reliable and proper connection.
To enable testing of the AD5933 and AD8606, I developed a script using the AD5933 library [13]. This library provided essential functions for performing basic operations with the AD5933. With the script in place, the AD5933 transmitted an excitation voltage of range 1 (2 Vp-p) at a constant frequency of 10kHz. To monitor the signals, I connected the AD5933's Vout (Testpoint 2) to CH1 of an oscilloscope, and AD8606's Transmitting Vout (Testpoint 3) to CH2 to observe the rebiased signal.
The initial test was successful. The voltage at Vout measured 2Vp-p and was biased at 1.48V, while the rebias voltage showed a bias of 1.65V and a voltage of 2Vp-p. These results indicated that the excitation signal and the biasing were functioning as expected. I proceeded to conduct a full-frequency sweep at an excitation range of 3 (0.4 Vp-p). This test also yielded positive results, as observed on the oscilloscope. The sine wave contracted as the frequency increased, demonstrating the proper functioning of the frequency sweep. Additionally, there was a noticeable DC bias difference, with a voltage of 0.31V at Vout and 1.65V after rebiasing. These outcomes validated the successful operation of the AD5933 and its ability to generate the desired frequency sweep.

It was time to calibrate the device and measure some reference resistances to ensure its capability of measuring impedance accurately. To calibrate the device, I followed the standard procedure described in detail in Section 3.2.1. However, now, I had the flexibility to choose any desired calibration resistor Rcal using the calibration MUX. After calibrating the device, I successfully measured the calibration resistors themselves by routing the excitation signal directly to each of them. This allowed me to verify the code and calculations work correctly. I proceeded to measure external resistors. The measured impedance values of the external resistors consistently showed a slight offset of around 100Ω higher than their expected values. This offset was expected as the multiplexers introduced some serial resistance. However, this offset was accounted for and compensated during further tests and impedance measurements.
This confirmed the successful operation of the AD5933 and AD8606 components in the system. They performed as expected during the calibration process and subsequent impedance measurements, providing reliable results.

4.1.6 Display

Up until this point, the data from the device was communicated through the Serial interface, providing textual feedback. However, it was now time to enhance the visual feedback by implementing a display.
The implementation of the display proved to be straightforward. After soldering the female header pins to the PCB and connecting the display, I proceeded to run a simple script using Adafruit's SSD1306 [10] and GFX [11] libraries to control the display. Much to my satisfaction, the display operated flawlessly, displaying the desired information as intended.

4.1.7 Complete PCB

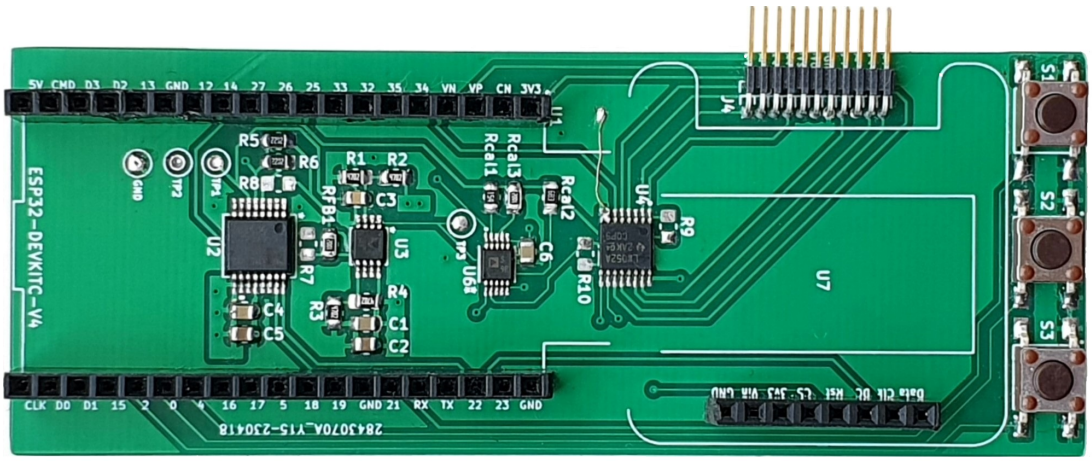


Figure 4.3: Complete PCB

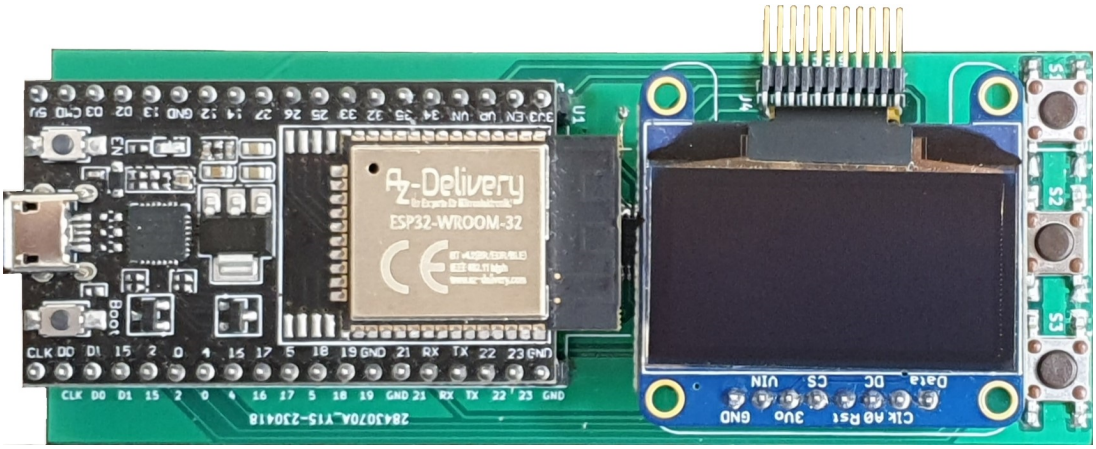


Figure 4.4: Complete PCB with MCU and display

4.1.8 Gradient Probe

The implementation of the probe was half complete during the implementation of the temperature sensor. all that was left is to add the second probe's PCB and bind them together. to achieve that I used a 2 tiny PCB of width 5mm (same as the testing probe) and glued everything together using epoxy glue.+

During the implementation of the temperature sensor, the probe was only partially completed. The final step in completing the probe assembly was to add the probe's second PCB (receiving side) and securely bind all the components together. Two small PCBs with a width of 5mm, matching that of the testing probe, were used. All PCBs were carefully attached together using epoxy glue, creating a unified structure for the complete probe assembly. This ensured the proper alignment and connection of all the components,

allowing for accurate impedance measurements. To provide additional protection to the epoxy I taped the upper part with insulating tape.

4.2 Testing and Optimization

This section delves into the actual testing and optimization of the device. During this phase, extensive testing was conducted to ensure the device's functionality and performance. The goal was to identify any potential issues or areas for improvement and implement necessary optimizations.

With the fully operational device, I conducted tests to measure impedance in ethanol and distilled water solutions. The objective was to replicate the measurements conducted in the Preliminary Investigation (Chapter 2), but this time including measurements from both the test probe and the gradient probe. This allowed for a proper comparison as the test probe had bare copper pads, while the gradient probe was coated with a protective layer.

Upon creating solutions of 0%, 15%, and 36% concentration, identical to the reference solutions (same water and ethanol), I encountered unexpected results during the measurements. The measured impedance values were significantly higher, almost two orders of magnitude, compared to the reference measurements. To ensure the accuracy of the measurements, I verified them using an LCR meter, which confirmed the high impedance values. To further investigate the issue, I conducted measurements using different distilled water samples. Although the impedance values were closer to the reference measurements, they still remained consistently higher. At this point, I began to suspect that the coating on the gradient probe might be a contributing factor. To test this hypothesis, I carefully sanded off the coating on the deepest pads of the probe. Although the measurements exhibited a change after scraping off the coating, the impedance values still remained higher than anticipated. Notably, there was a consistent trend of increasing impedance with increasing ethanol concentration. This observation suggests that there may be additional factors influencing the measurements, including the base impedance with pure water.

To address the discrepancies observed in the impedance measurements, a decision was made to acquire completely new distilled water from a hardware store. Three bottles, each containing 1 liter of distilled water, were purchased for the subsequent measurements. Additionally, a new and sealed bottle of ethanol was used for this particular set

of measurements.
During the first measurement of pure water using the new distilled water, the impedance was found to be approximately 13 kΩ, which was significantly higher than the reference value of 900Ω. Subsequent measurements with 15% and 36% ethanol solutions showed further increases in impedance, but the baseline impedance remained around 13 kΩ.
Given these findings and the consistent trend observed across different water samples, it was decided to discard the previous reference and rely solely on the new sealed distilled water as the basis for future measurements. This approach would ensure greater accuracy and consistency in the impedance measurements moving forward. The plan was to create solutions with increments of 5% from 0% to around 70% ethanol. Instead of measuring at 10kHz like before, full frequency sweeps (1kHz to 100kHz) were performed to find the optimal frequency for impedance measurements in these solutions.
The impedance measurements were progressing smoothly and consistently, with the impedance increasing at a steady rate for each 5% increment in ethanol concentration. However, during the middle of the measurements, the first bottle of distilled water ran out and a new bottle was opened. Upon opening another bottle of distilled water, a 35% solution was prepared. However, to my surprise, the impedance measurement was twice as high as the previous measurements. In order to verify if the issue was due to contamination on my part, I prepared a 30% ethanol solution and measured it. Unfortunately, the measurements were significantly higher as well. To confirm my suspicions, I measured the impedance of pure water from the new bottle, and it was around 37 kΩ, 24kΩ higher than the previous bottle.
To further investigate the issue, I decided to open the third bottle of distilled water. The impedance measurements obtained from this bottle were around 70 kΩ, indicating a significant increase compared to the previous measurements. I re-measured the second bottle of water, and to my surprise, it showed a similar impedance value of 37 kΩ once again. However, when I touched the water with my finger, the impedance dropped. This confirmed that the distilled water samples had varying conductivity levels, which were volatile and unpredictable. With such inconsistencies, it became evident that accurate impedance measurements could not be obtained using these de-mineralized water samples.

At this point, an unconventional decision was made to try and perform the impedance measurements using tap water. To ensure accuracy, I adopted a meticulous approach. I measured the tap water multiple times, both separately and in between ethanol solutions, to check for consistency.
I proceeded to make solutions ranging from 0% to 90% concentration of tap water. Up until 40%, the results exhibited a steady linear dependence, similar to the previous mea-

surements. Moreover, unlike previous measurements, the imaginary part of the impedance grew at a much higher rate than the real part. However, when I went beyond 40% concentration, the impedance followed an exponential curve, although it did not fit perfectly. Nevertheless, when I plotted the imaginary part, it showed an almost perfect exponential fit as depicted in Figure 4.5. This was a very promising finding.
Notably, when measuring the tap water, the impedance was around 1500Ω with around 0Ω imaginary part. This provided a stable baseline for comparison and analysis.
To validate the stability of the measurements, I repeated some of the measurements with a few solutions on the following day. The results matched the previous measurements, confirming the consistency and stability of the impedance measurements using tap water.

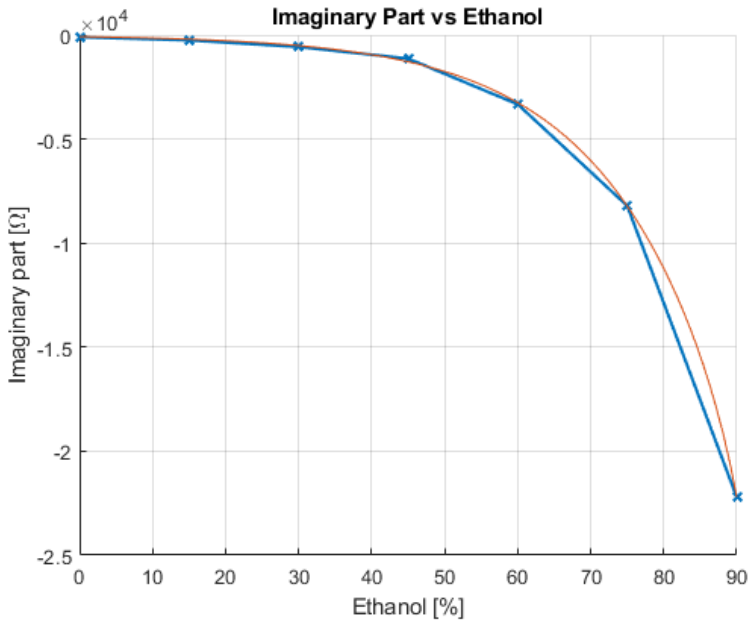


Figure 4.5: Imaginary part vs Ethanol

Confident in the accuracy of the measurements using tap water, I proceeded to the next step of the tests, which involved investigating the dependence of impedance on temperature. The approach was to heat each solution on a hot plate and measure the impedance as the temperature gradually decreased. For this experiment, I prepared solutions with concentrations of 0%, 5%, 10%, 15%, 22.5%, 30%, 45%, 60%, 75%, and 90%. Each solution was measured at 3 different depths to account for variations in the positioning and contact of the measurement pads.
The collected data was analyzed, revealing a linear dependence of impedance on the temperature within the range of 18 °C - 29°C. However, it was observed that the slope of these temperature dependencies varied with the ethanol percentage in the solution. At lower ethanol concentrations, the slope was smaller, while it increased with higher ethanol

concentrations. This indicated that the temperature sensitivity of the impedance was influenced by the ethanol content, with higher concentrations leading to a more pronounced change in impedance with temperature.

By calculating the temperature slopes for each ethanol concentration, I was able to determine the rate of change in impedance with temperature for each solution. With this information, I could accurately estimate the impedance of each solution at any given temperature. This allowed for precise characterization of the imaginary part vs ethanol percentage relationship for different temperatures and provided a valuable tool for temperature compensation. For visual purposes, I inverted the imaginary parts. Figure 4.6 depicts the relationship between the imaginary part of the impedance and ethanol concentration at different temperatures. The plot clearly demonstrates an exponential increase in the magnitude of the imaginary part as the ethanol percentage increases. Additionally, as the temperature rises, the magnitude of the imaginary part decreases.

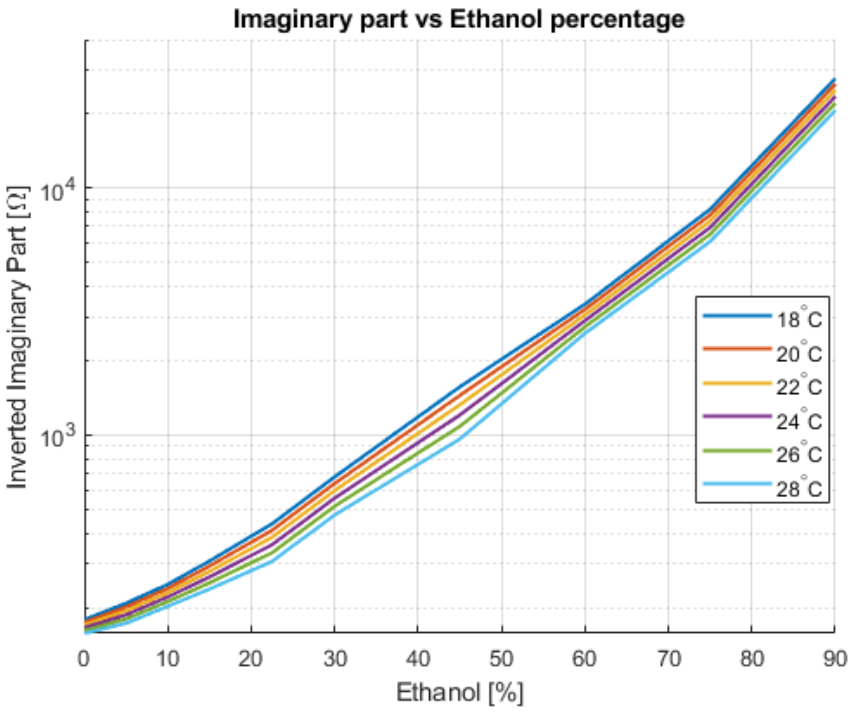


Figure 4.6: Imaginary part vs Ethanol at Different Temperatures

Although the data couldn't perfectly fit into a double exponential form, a 7th-degree polynomial was sufficient to get a really close fit of each curve. Using that, I could develop an algorithm for the MCU to calculate the ethanol percentage while compensating for the temperature.

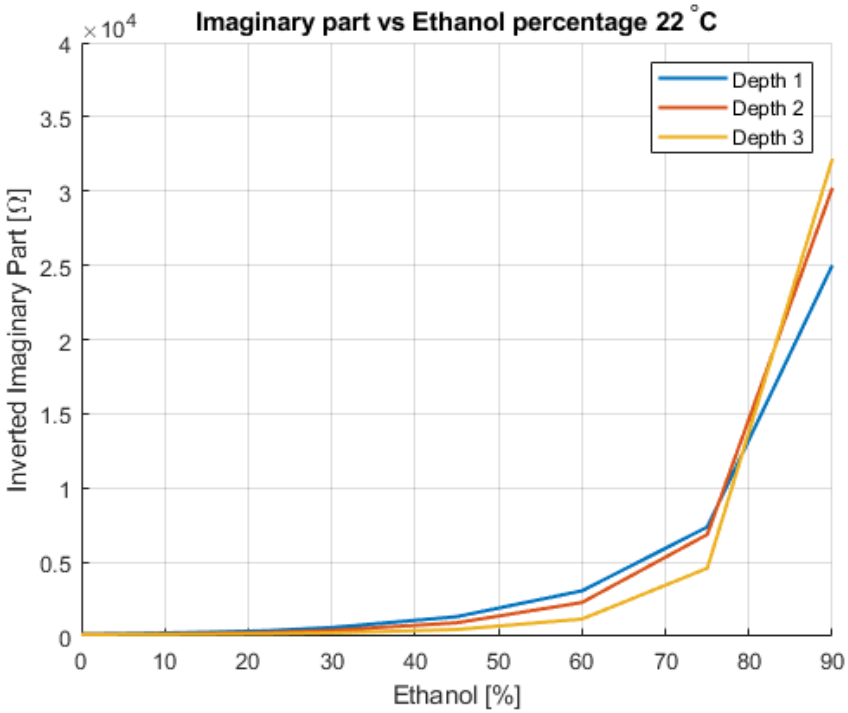


Figure 4.7: Imaginary part vs Ethanol at 22 °C

Figure 4.7 shows how the imaginary part is changing for each set of pads on the probe at 22 °C. Apart from the fact that at depth 1 the pads are sanded down to bare copper, factors such as proximity to container walls, the overall depth, and volume can affect the impedance and impedance phase of every pad on the probe.

By calculating all the coefficients at different temperatures I can know the precise equation of each temperature curve. each curve has 8 coefficients. After calculating the coefficients for each temperature curve, ranging from 15 degrees to 29 with a step size of 0.5 degrees Celsius for each of the 3 pairs of probe pads, I proceeded to upload these coefficients to the ESP32 as three 2D arrays. Each 2D array represents a specific measuring depth, and each row within the array contains 8 coefficients corresponding to different temperature step sizes. By organizing the coefficients in this manner, the ESP32 can easily access the appropriate coefficients based on the measured depth and temperature.

During a measurement, the MCU first measures the temperature of the liquid, rounding it to the nearest half-degree. It then measures the impedance and calculates the imaginary part using the calculated phase. Subsequently, a bisection method function is called. This function is responsible for looking up the corresponding set of coefficients based on the depth and rounded temperature values. Using the calculated imaginary part and the retrieved coefficients, the bisection method function performs the bisection

algorithm on a 7th-degree polynomial.
The bisection method is applied iteratively, refining the range of possible ethanol percentages until the desired precision is achieved. By dividing the range in half and evaluating the polynomial at the midpoint, the function determines whether the desired ethanol percentage lies in the left or right half of the range. This process continues until the desired precision is met, providing an accurate estimation of the ethanol percentage.

With this capability, the device can effectively measure the ethanol concentration at different depths within the liquid. This provides the capability to approximate the distribution of ethanol within the liquid. For instance, if the ethanol concentration is higher towards the upper side of the cup, the device should detect varying ethanol percentages at different levels. Conversely, if the ethanol is evenly distributed, all measurement pads should yield the same percentage of ethanol concentration.

Chapter 5

Firmware

5.1 ESP32 Programming

The chosen framework for programming the ESP32 was Arduino IDE, which is compatible with the DevKit C V4 board after downloading the ESP32 packages. Arduino IDE also provided a convenient Serial Monitor tool, facilitating communication and debugging during the implementation stage.

5.2 Libraries

In addition to the standard libraries, several open-source libraries were incorporated into the firmware:

- Adafuit's SSD1306 library [10]: This library provides functions for controlling and displaying content on the SSD1306 OLED display module.
- Adafuit's GFX library [11]: The GFX library is a companion library to the SSD1306 library. It provides a set of graphics functions, such as drawing lines, text, and different fonts as well as bitmap drawing (for the logo) which enhance the visual capabilities of the OLED display and enable the creation of a more intuitive user interface.
- AD5933 library [13]: This library is specific for the AD5933. It includes the most basic functions for initializing the device, setting frequency parameters, performing sweeps, and reading impedance values. However, it is somewhat incomplete and I had to make some changes to tailor it for the impedance meter.

The integration of these open-source libraries enhances the functionality and usability of the firmware, allowing for efficient control of the OLED display and seamless communication with the AD5933.

5.3 Firmware Structure

After booting, the necessary libraries and the I²C bus are initialized. Next, the AD5933 impedance converter is configured with predefined frequency settings. Following that, the relevant GPIO pins are enabled and configured accordingly. Additionally, the display module is initialized and powered on, showcasing a custom logo. This concludes the initialization phase of the firmware. Subsequently, the main program execution begins, carrying out the desired functionalities of the device.

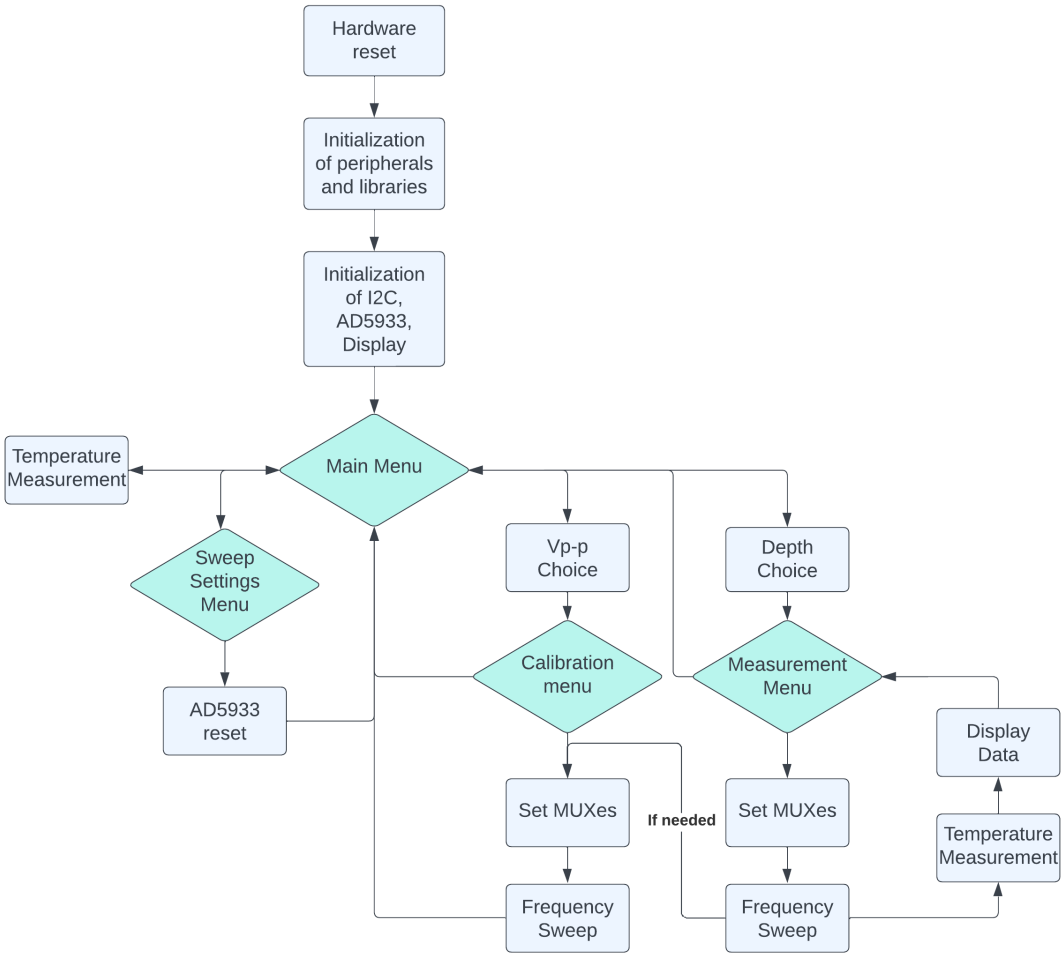


Figure 5.1: Firmware Block Diagram

5.3.1 Main Program

The firmware is designed as a menu-based system, providing a user-friendly interface to perform various operations. The main menu serves as the initial state, offering different options to the user. A visual representation of the firmware's functionality is depicted in Figure 5.1. The main menu offers the user several options, including the ability to measure the temperature of the probe.

Prior to conducting any measurements, it is crucial to calibrate the device. To initiate the calibration process, the user selects the calibration option from the main menu. The device prompts the user to choose the desired excitation voltage. Next, the user selects a calibration resistor for the calibration sequence or has the option to return to the main menu. During the sequence, the program configures the calibration MUX to allow the excitation voltage to pass through the selected resistor. It then instructs the AD5933 to perform a sweep while simultaneously reading the imaginary and real registers, and calculating the necessary gain factor and system phase. Once the calibration is complete, the device provides visual feedback and returns to the main menu.

Once in the measurements menu, the user can perform meaningful measurements. The program prompts the user to select the desired depth for measurement. This selection determines the number of pads on the probe that will be iterated during the measurement process. For example, if the user chooses depth 1, only the bottom pads will be measured. As the depth increases, such as depth 4, the entire line of pads will be measured. This depth selection allows for targeted measurements at specific positions within the liquid.
During the measurement sequence, at each depth iteration, the program sends a signal to the probe MUX to pass the signal to the corresponding pads. The AD5933 then performs a sweep and the program calculates the impedance based on the real and imaginary registers. If the impedance falls below a certain threshold, this indicates that the ADC of AD5933 is saturated. An automatic iterative re-calibration is triggered. This re-calibration process adjusts the excitation voltage to a lower range while using the same calibration resistor (Rcal). Once the measurements are within the linear region of the ADC, the ethanol approximation algorithm described in Section 4.2 is applied to estimate the ethanol content.
Upon completion of the measurement sequence, the data is presented to the user, allowing them to swipe between each measured depth's data using the tactile switches. The data contains the ethanol approximation, impedance magnitude, impedance phase, and real and imaginary magnitudes. This user interface feature provides easy navigation and access to the obtained measurement results.

In addition to the core functionalities, a sweep settings menu was implemented, allowing the user to select different frequency sweep ranges. The available options include 10kHz, 100kHz, or a full frequency sweep between 1kHz and 100kHz. This feature provides the device with added flexibility, enabling impedance measurements on any liquid (without ethanol approximation).

Chapter 6

Results

6.1 Single Depth Results

These measurements are focused on assessing the device's capability to accurately measure impedance in various solutions. By conducting measurements at a single depth, this test aimed to determine the device's accuracy with some temperature variations. Six precisely calculated solutions of various ethanol concentrations were made and measured using the device.

Table 6.1: Single Depth Approximations

Real Percentage [%]	9.5	13.7	21.58	38.12	46.73	83.66
Approximated Percentage [%]	10.84	14.572	21.196	39.804	47.09	82.5
Error [%]	1.34	0.872	-0.384	1.684	0.36	-1.16

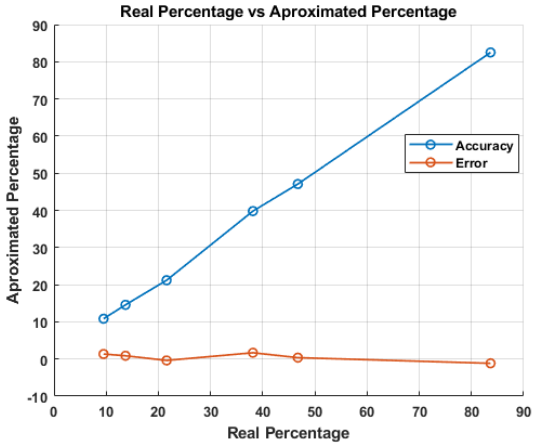


Figure 6.1: Accuracy and Error of Approximations

These results clearly indicate that the device is capable of successfully approximating the ethanol content in a solution with relatively good accuracy.

6.2 Multi Depth Results

These measurements are focused on the device's capability to assess the distribution of ethanol within a given liquid solution by approximating the ethanol content at different depths. Depth 1 is considered the deepest.
First, two solutions were prepared by mixing ethanol with water, with the purpose of demonstrating the effectiveness of measuring at three different depths to approximate the ethanol percentage throughout the solution.
Then, a separate solution consisting of 40% ethanol was prepared using a different approach. However, rather than mixing the ethanol and water, this solution was made by carefully layering the ethanol on top of the water, forming distinct layers. After some time, the solution was slightly mixed to observe any changes in the ethanol distribution. Subsequently, the solution was thoroughly mixed.

Table 6.2: Multi Depth Approximations (Mixed Solution)

Real Percentage [%]	20.3	40
Depth 3 [%]	19.73	40.23
Depth 2 [%]	21.03	40.5
Depth 1 [%]	20.21	40.11

Table 6.3: Multi Depth Approximations (Segregated Solution)

	Segregated	Slightly Mixed	Completely Mixed
Real Percentage [%]	40	40	40
Depth 3 [%]	100	72.024	42.608
Depth 2 [%]	7.825	28.564	40.36
Depth 1 [%]	1.235	17.24	39.244

These results demonstrate that the device is capable of accurately approximating the ethanol concentration at different depths within the solution. Furthermore, the measurements demonstrate the device's ability to assess the distribution of ethanol throughout the solution.

Chapter 7

Conclusion

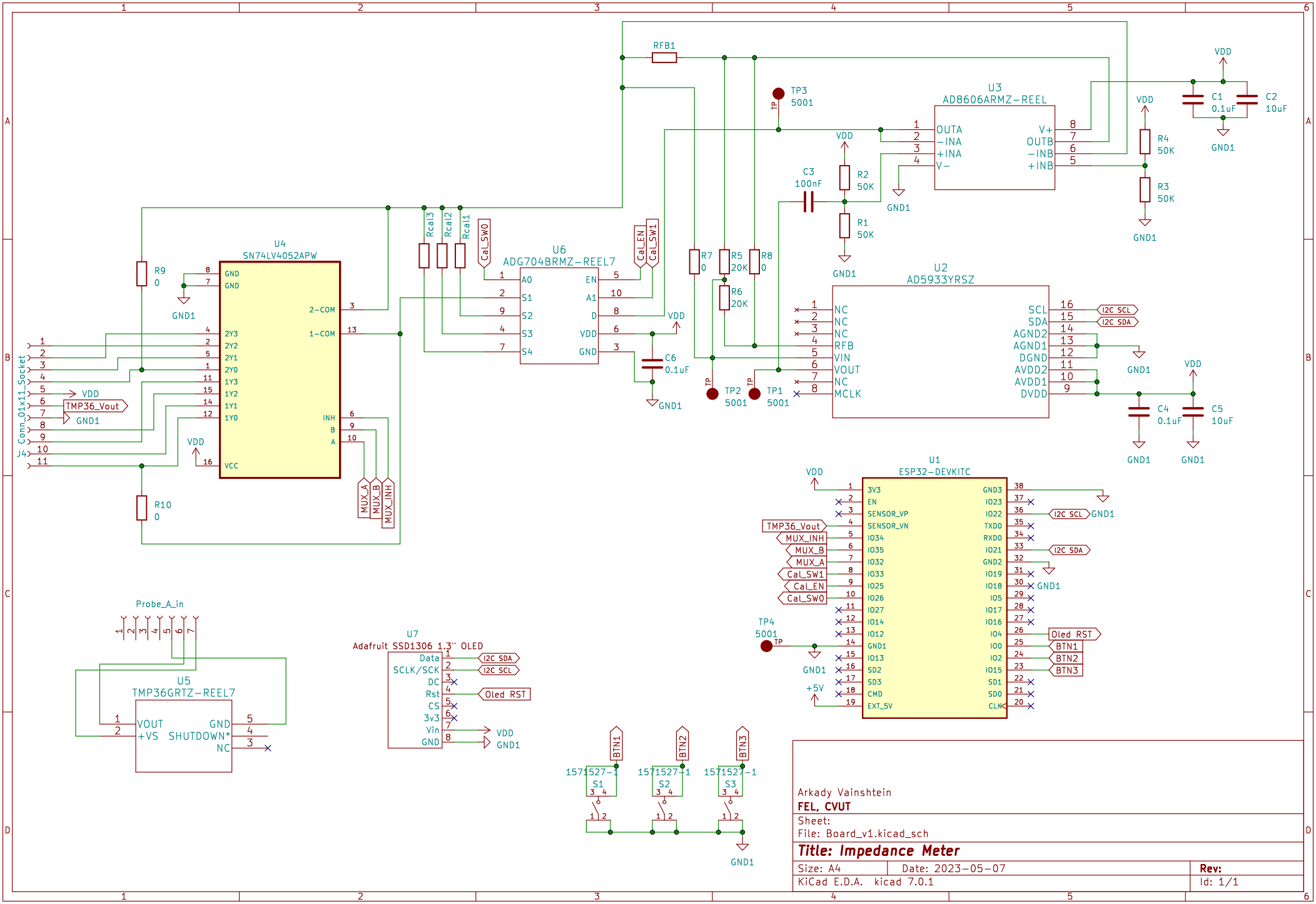
In this thesis, I have successfully developed a multi-channel impedance meter capable of approximating ethanol concentration in temperature-variable liquid solutions. Throughout this thesis, I have taken on the tasks of designing, manufacturing, implementing, and developing the firmware for the device.
The device has demonstrated its full operational capability, effectively measuring impedance and estimating ethanol percentages at different depths while compensating for temperature. The goals set forth at the beginning of this thesis have been accomplished.

The process of working on this project has been educational for me. As someone with no prior experience in PCB design and manufacturing, I have gained valuable insights and practical skills through the successful execution of these tasks. Additionally, I successfully overcame various challenges that emerged during the development of the device like in the implementation phase or the characterization of the impedance relationship with ethanol. By addressing these challenges, I ensured the completion of the thesis and the achievement of the desired goals.

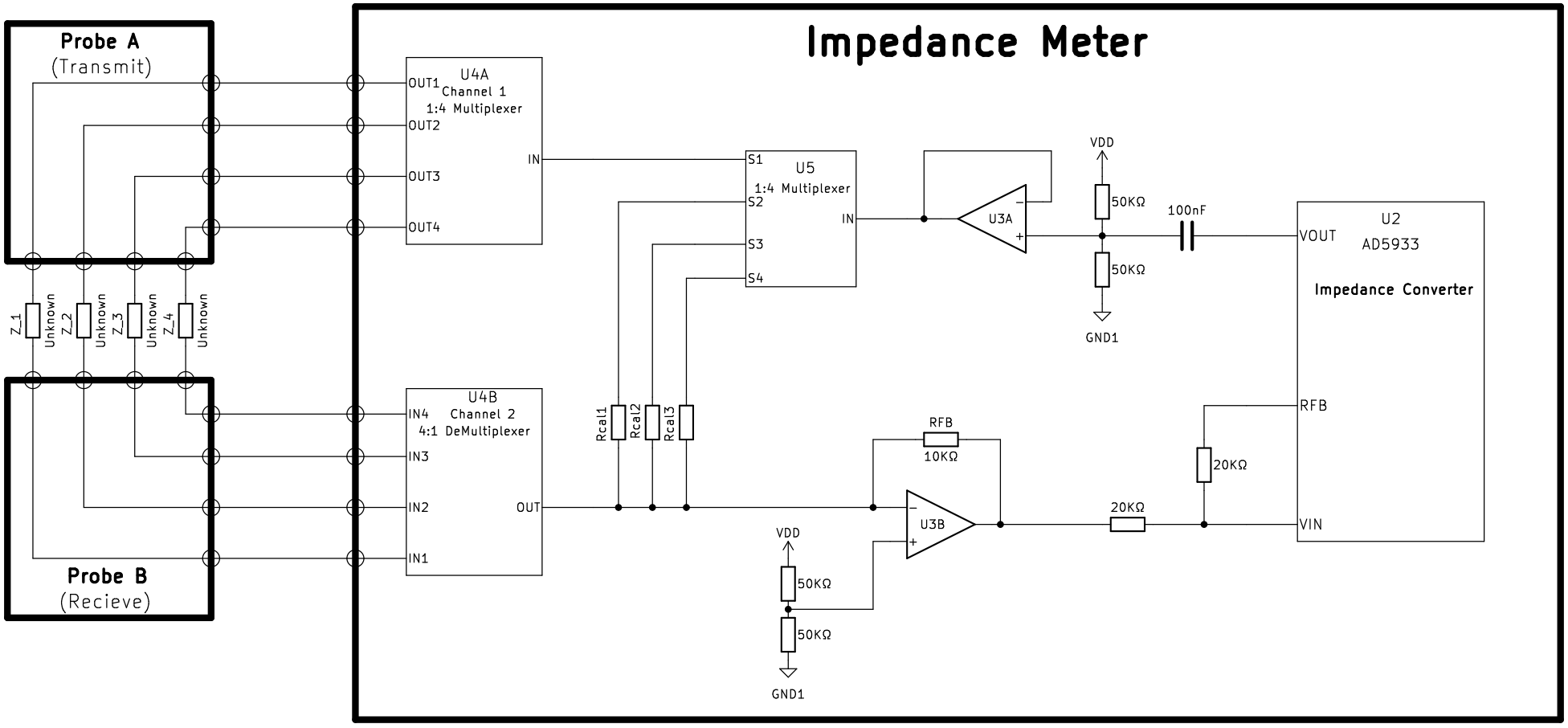
Future improvements and development can focus on researching a wider range of solution types and investigating the effects of container shapes and volumes on measurements. Further research can also be conducted to overcome impedance instability in water.

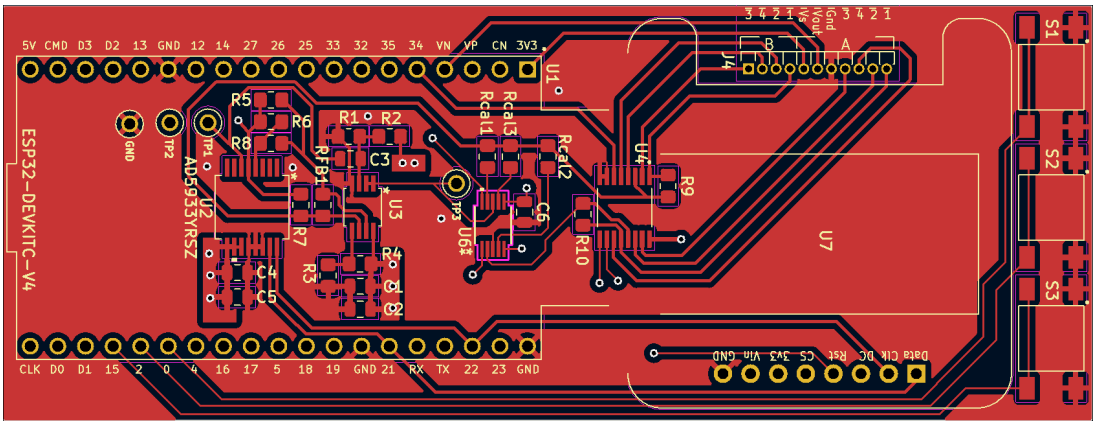
Appendix A

Schematics

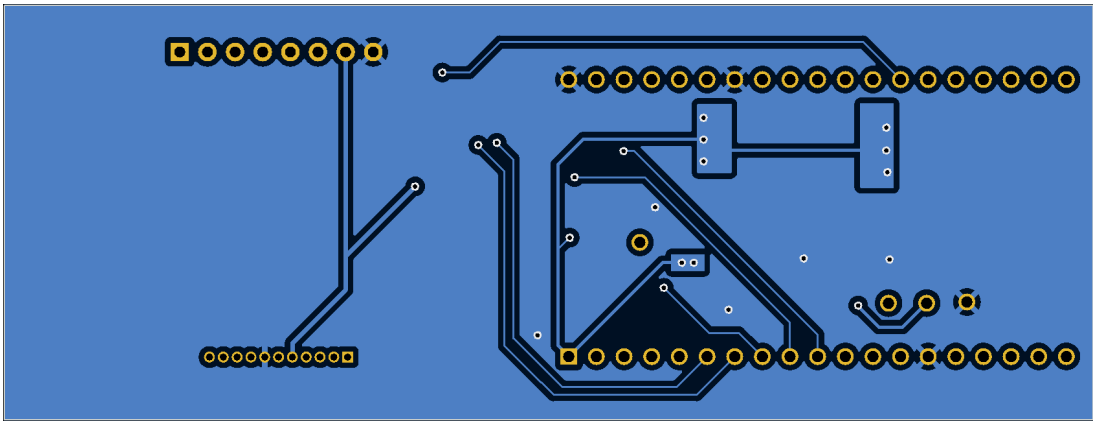


Arkady Vainshtein		
FEL, CVUT		
Sheet:		
File: Board_v1.kicad_sch		
Title: Impedance Meter		
Size: A4	Date: 2023-05-07	**Rev:**
KiCad E.D.A. kicad 7.0.1		Id: 1/1


Impedance Meter
Probe A
(Transmit)
Probe B
(Recieve)
Z_1
Unknown
Z_2
Unknown
Z_3
Unknown
Z_4
Unknown
U4A
Channel 1
1:4 Multiplexer
OUT1
OUT2
OUT3
OUT4
IN
U4B
Channel 2
4:1 DeMultiplexer
IN4
IN3
IN2
IN1
OUT
U5
1:4 Multiplexer
S1
S2
S3
S4
IN
Rcal1
Rcal2
Rcal3
U3A
U3B
VDD
50KΩ
50KΩ
GND1
100nF
RFB
10KΩ
20KΩ
20KΩ
VDD
50KΩ
50KΩ
GND1
U2
AD5933
Impedance Converter
VOUT
RFB
VIN



(a) Top Layer Schematics



(b) Bottom Layer Schematics

(c) Probe's Top Layer Schematics

(d) Probe's Bottom Layer Schematics

Figure A.1: PCB's Schematics

Bibliography

[1] Analog Devices, *AD5933 Datasheet*. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD5933.pdf [Visited: 15.04.2023].

[2] Analog Devices, *AD5933 Evaluation Board*. [Online]. Available: https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/EVAL-AD5933.html [Visited: 15.04.2023].

[3] Analog Devices, *AD5933 Evaluation Board Datasheet*. [Online]. Available: https://www.analog.com/media/en/technical-documentation/user-guides/UG-364.pdf [Visited: 15.04.2023].

[4] AZ-Delivery, *AC-Delivery ESP32 DevKit C V4 Datasheet*. [Online]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/ESP-32_DevKit_C_V4_Datenblatt_AZ-Delivery_Vertriebs_GmbH_24ec770f-c65e-4bd3-92c9-cd64b4d070b8.pdf?v=1615364587 [Visited: 15.04.2023].

[5] AZ-Delivery, *AC-Delivery ESP32 DevKit C V4 product page*. [Online]. Available: https://www.az-delivery.de/en/products/esp-32-dev-kit-c-v4 [Visited: 21.05.2023].

[6] Analog Devices, *AD8606 Datasheet*. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8605_8606_8608.pdf [Visited: 15.04.2023].

[7] Analog Devices, *ADG704 Datasheet*. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/adg704.pdf [Visited: 15.04.2023].

[8] Texas Instruments, *SN74LV4052A Datasheet*. [Online]. Available: https://www.ti.com/lit/ds/symlink/sn74lv4052a.pdf?ts=1682339260343&ref_url=https%253A%252F%252Fwww.google.com%252F [Visited: 15.04.2023].

[9] Analog Devices, *TMP36 Datasheet*. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf [Visited: 15.04.2023].

[10] Adafruit, *Adafruit's SSD1306 Library*. [Online]. Available: https://github.com/adafruit/Adafruit_SSD1306 [Visited: 15.05.2023].

[11] Adafruit, *Adafruit's GFX Library*. [Online]. Available: https://github.com/adafruit/Adafruit-GFX-Library.git [Visited: 15.05.2023].

[12] Adafruit, *Adafruit 1.3" Monochrome OLED Breakout Datasheet*. [Online]. Available: https://cz.mouser.com/datasheet/2/737/monochrome_oled_breakouts-2489926.pdf [Visited: 15.04.2023].

[13] Michael Meli, *AD5933 Library*. [Online]. Available: `https://github.com/mjmeli/arduino-ad5933` [Visited: 15.04.2023].