

## Wireless Readout Circuit for Inductively Coupled Pressure Sensors

---



# Summary

Front Matter.....	3
Acknowledgements.....	3
Abstract.....	3
Résumé (français).....	4
1. Introduction and Background.....	5
1.1 Introduction.....	5
1.2 Presentation of the University of Galway, the Lambe Institute, and the TMDLab.....	5
1.3 Project Context.....	6
1.4 Project Objectives.....	7
2. Methodology and Design Strategy.....	8
2.1 Methodology.....	8
2.2 Constraints.....	9
2.3 Projet Specifications.....	10
2.4 Component Selection.....	11
3. Hardware Subsystems.....	12
3.1 Phase Shifter Design and Testing.....	12
3.2 Phase Detection Using AD8302.....	19
3.3 Frequency Generation Using AD5932 (DDS).....	23
3.4 Signal Acquisition Using AD7177-2.....	25
3.5 Integration ADC into the Frequency Sweep System.....	26
4. System Integration and Testing.....	27
4.1 Experimental Validation of the Complete System.....	27
4.2 Final PCB Design Integrating All Modules.....	28
Conclusion.....	29
References.....	29
Appendix A — Zoomed view of the Qucs schematic (LPF + HPF).....	30
Appendix B — PCB after Corrosion and final.....	31
Appendix C — Comparison of phase shift curves for different inductance values.....	31
Appendix D — Zoom Qucs schematic of the second hybrid phase shifter.....	33
Appendix E —Phase Measurements of the Different Test Assemblies.....	34
Appendix F — Zoom Qucs schematic of the adjustable phase shifter.....	35
Appendix H — Block Diagramme AD8302.....	36
Appendix G — Phase detector PCB (AD8302) before corrosion and final.....	37
Appendix H — QucsStudio schematic and simulated FFT response.....	37
Appendix I — FFT measurements before and filtering.....	38
Appendix J — Arduino final code.....	41
Appendix K — Final Arduino wiring.....	50
Appendix R — Electrical schematic Part 1.....	52
Appendix S — Electrical schematic Part 2.....	53
Appendix N1 — Component Implantation.....	53
Appendix N2 — Left Top copper & Right Bottom copper.....	54
Appendix N3 — 3D View of PCB.....	55

# Front Matter

## Acknowledgements

I would like to express my sincere gratitude to all the people who contributed to making this internship such a valuable and memorable experience.

My deepest thanks go to Dr. Icaro Soares, my supervisor at the TMDLab, for his trust, guidance, and continuous support throughout the internship. His scientific expertise, availability, and constructive feedback were essential to the progress of the project and to my development, both technically and professionally.

I am also very grateful to Dr. Muhammad Farooq, whose help during the early stages of the internship was invaluable. His expertise and practical assistance in the fabrication of printed circuit boards, especially using the iron transfer method, gave me a strong technical foundation for the rest of the work.

I would like to thank Dr. Elahi for approving my internship at the TMDLab and for assigning me such an enriching project.

My thanks also go to Mr. Serge Bouter, professor at the IUT of Bordeaux, for his valuable help in finding this internship and for connecting me with the lab.

A big thank you to my colleagues — Eoghan, Farooq, James, Josh, Aksh, Mehedi, and Abel Aguilar for their support and positive spirit.

Finally, I would like to thank everyone else at the Lambe Institute who, directly or indirectly, helped make this internship such a rewarding and unforgettable experience.

## Abstract

This internship report outlines my 10-week professional experience, carried out from April 10 to June 14, 2025, at the Translational Medical Device Lab (TMDLab), part of the Lambe Institute for Translational Research in Galway, Ireland. This internship was part of the second year of my University Bachelor of Technology (BUT) in Electrical Engineering and Industrial Computing (GEII) at the IUT of Bordeaux, with a specialization in Electronics and Embedded Systems.

The main objective of the project was to reproduce a wireless measurement system based on an inductively coupled resonant sensor, as described in a scientific publication. This system is designed to detect variations in the resonant frequency, such as those observed in biomedical sensors to monitor pressure or changes in arterial diameter, and can be used in biomedical applications such as compression garments.

My work was based on the wireless readout system developed by **Salpavaara et al.**, published in *Sensors and Actuators A* (2011), which served as the foundation for the system I implemented. However, since only the block diagram of the circuit was available, I had to design the detailed architecture, identify the required functions, and then select and integrate components that matched

the functions defined in the block diagram. I used measurement tools such as an oscilloscope and a vector network analyzer (VNA) to analyze signals and verify the circuit's responses. To ensure a clear understanding and control of the system, I chose to develop each module independently (DDS generator, ADC converter, phase detector, etc.), testing them one by one before proceeding to full integration. At the same time, I recreated the circuit in QucsStudio, a simulation software, to better understand its behavior and improve its performance. I also made hardware adjustments, including testing different wiring configurations and ensuring stable connections within the circuit.

This internship gave me solid experience designing, testing, and simulating electronic circuits while working in an English-speaking research environment. I was able to strengthen my technical skills, improve my professional English, and gain insight into the requirements of working in an international research setting.

## Résumé (français)

Ce rapport de stage présente mon expérience professionnelle de 10 semaines, réalisée du 10 avril au 14 juin 2024 au Translational Medical Device Lab (TMDLab), intégré à l'Institut Lambe for Translational Research à Galway, en Irlande. Ce stage s'inscrivait dans le cadre de ma deuxième année de BUT Génie Électrique et Informatique Industrielle (GEII) à l'IUT de Bordeaux, avec une spécialisation en Électronique et Systèmes Embarqués.

L'objectif principal du projet était de reproduire un système de mesure sans contact, basé sur un capteur résonant à couplage inductif, tel que décrit dans une publication scientifique. Ce système est conçu pour détecter des variations de capacité, par exemple celles liées à la pression ou au diamètre d'une artère, et peut être utilisé dans des applications biomédicales telles que les vêtements de compression.

Mon travail a consisté à recréer ce système à partir des informations disponibles dans la publication. Toutefois, seul un schéma bloc du circuit était fourni. J'ai donc dû concevoir l'architecture détaillée du système, identifier les fonctions nécessaires, puis sélectionner et intégrer les composants capables d'assurer les rôles définis dans le schéma. J'ai utilisé des instruments de mesure tels qu'un oscilloscope et un analyseur de réseau vectoriel (VNA) pour analyser les signaux et vérifier le comportement du circuit.

Afin de bien comprendre le fonctionnement du système et de garder un contrôle clair sur son développement, j'ai choisi de développer chaque module indépendamment (générateur DDS, convertisseur ADC, détecteur de phase, etc.), en les testant un à un avant de procéder à l'intégration complète. En parallèle, j'ai reconstruit le circuit dans le logiciel de simulation uSimmics afin de mieux analyser son comportement et d'en optimiser les performances. J'ai également effectué divers ajustements matériels, notamment des tests de câblage et de stabilité des connexions.

Ce stage m'a permis d'acquérir une solide expérience en conception, test et simulation de circuits électroniques, tout en évoluant dans un environnement de recherche anglophone. Il m'a permis de renforcer mes compétences techniques, d'améliorer mon anglais professionnel et de mieux comprendre les exigences du travail en laboratoire de recherche à l'international.

# 1. Introduction and Background

## 1.1 Introduction

This report presents a rewarding internship experience that allowed me to apply my technical knowledge, explore the field of biomedical research, and immerse myself in a new cultural environment.

My name is Mathieu Chevalier, and I am currently in my second year of a University Bachelor of Technology (BUT) in Electrical Engineering and Industrial Computing (GEII) at the IUT of Bordeaux, specializing in Electronics and Embedded Systems.

As part of my academic program, I completed a two-month internship at the Translational Medical Device Lab (TMDLab) in Galway, Ireland — a research lab affiliated with the university hospital, focused on the development of innovative medical technologies.

I chose this internship for two main reasons. On the one hand, I wanted to deepen my knowledge of electronics by contributing to a real-world, medically relevant project. On the other hand, I was eager to improve my English, gain autonomy, and discover a new way of working by living abroad.

The project I worked on involved developing a non-invasive measurement system based on inductive coupling. The goal was to detect variations in the resonant frequency, for instance, changes caused by variations in the diameter of the aorta or pressure, using a resonant sensor circuit, supported by different functional blocks such as a signal generator (DDS), an analog-to-digital converter (ADC), and a phase detector.

The beginning of the internship came with challenges, especially adapting to the Irish accent and communicating in English. However, the immersive environment quickly helped me improve both my language skills and independence.

This report is divided into three main sections:

- The first part presents the lab and its structure,
- The second explains the project context and objectives,
- The third details the work I carried out, followed by a conclusion reflecting on both the professional and personal aspects of the experience.

## 1.2 Presentation of the University of Galway, the Lambe Institute, and the TMDLab

The University of Galway, formerly known as NUI Galway, is one of Ireland's most prestigious universities. Founded in 1845, it has grown into a dynamic institution with a strong focus on research, innovation, and societal impact. With more than 18,000 students and a wide range of undergraduate and postgraduate programs, it plays a central role in the academic and scientific landscape of the west of Ireland. The university is particularly renowned for its research in biomedical engineering, environmental science, and digital health.

One of the university's flagship research hubs is the Lambe Institute for Translational Research, located adjacent to Galway University Hospital. This strategic positioning fosters close collaboration between clinical practitioners and academic researchers. The institute's core mission is to translate laboratory discoveries into practical solutions for healthcare. It provides state-of-the-art infrastructure, including wet labs, electronics, and prototyping rooms, and access to advanced testing equipment. Research activities at the Lambe Institute are often multidisciplinary, combining electronics, computer science, materials engineering, and clinical sciences.

Within this vibrant environment, the Translational Medical Device Lab (TMDLab) stands out as a key player in medical device innovation. The TMDLab focuses on designing, developing, and validating novel technologies aimed at improving patient care, diagnostics, and therapeutic delivery. The lab's team of around 20 members includes principal investigators, postdoctoral researchers, PhD students, engineers, and interns from various scientific backgrounds.

The TMDLab engages in a wide range of projects, including the development of wearable health monitoring systems, wireless sensors, implantable devices, and point-of-care diagnostics. These projects often involve collaboration with hospital departments, industry partners, and international institutions, enabling a strong connection between engineering innovation and clinical relevance.

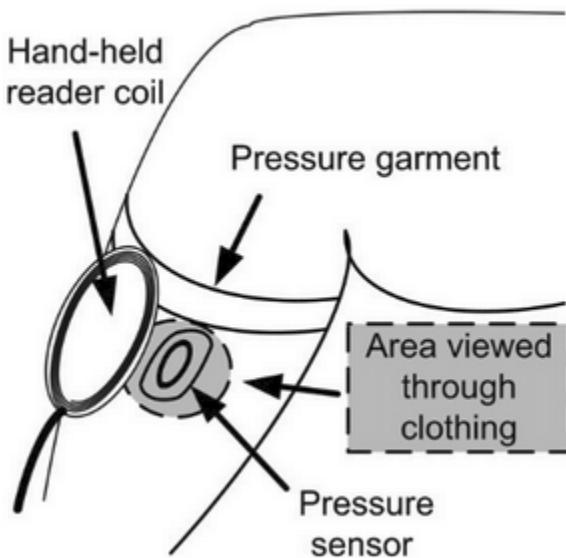
During my internship, I was fully immersed in this dynamic ecosystem. I was able to participate in cutting-edge research and work with talented individuals from diverse fields. The collaborative and supportive atmosphere within the TMDLab gave me insight into how multidisciplinary teams approach complex challenges in medical technology. It also helped me understand the importance of communication between engineers and clinicians in the development of devices that are not only functional but also aligned with real medical needs.

This experience greatly contributed to both my technical growth and my understanding of how academic research can lead to tangible innovations that impact healthcare at a global level.

## 1.3 Project Context

This project is inspired by a scientific study published by Salpavaara *et al.* in *Sensors and Actuators A (2011)*, which presents a method for non-invasive pressure monitoring using an inductively coupled (LC) sensor. The system is specifically designed for use in pressure garments, which are commonly prescribed to treat burn scars or leg swelling. These garments apply a therapeutic pressure in the range of 20 to 50 mmHg, and maintaining this pressure accurately is critical for effective treatment.

The principle of operation relies on placing a passive LC sensor beneath the garment, in contact with the skin. A hand-held reader coil, positioned outside the clothing, can wirelessly detect changes in the sensor's resonant frequency, which shift according to variations in applied pressure. Since capacitance in the LC circuit varies with mechanical deformation, monitoring the frequency allows the system to infer the pressure level without any physical connection between the reader and sensor.



**Application, Salpavaara et al., Sensors and Actuators A (2011)**

The major advantages of this approach are that it is completely wireless, non-invasive, and comfortable for the patient. The system can achieve an accuracy of approximately  $\pm 2$  mmHg, making it suitable for clinical use.

While the publication provides a conceptual overview and a functional block diagram, it lacks a detailed circuit design. Therefore, the main challenge of this project was to reproduce a functional prototype based on the described principle, by selecting and integrating appropriate components, designing the full signal chain, and validating the system through simulation and experimental testing.

This project lies at the intersection of electronics and biomedical engineering and demonstrates how inductive telemetry can be used for remote health monitoring. It aligns with the goals of translational medical research, aiming to develop practical tools that address real-world clinical needs.

## 1.4 Project Objectives

The main objective of this project is to design and implement a wireless readout system based on inductive coupling and resonance, for the purpose of tracking changes in a passive LC sensor.

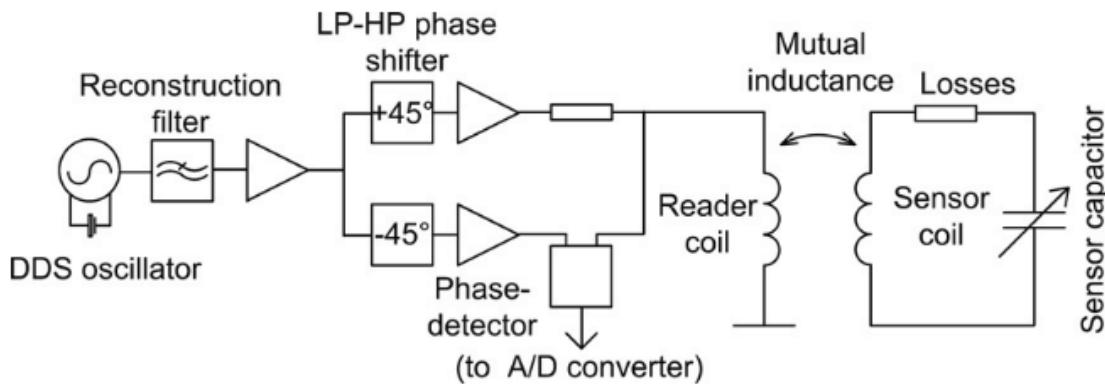
This work was inspired by the wireless readout architecture proposed by Salpavaara *et al.* in *Sensors and Actuators A (2011) A*. Figure X reproduces the original block diagram introduced in the paper, which served as the conceptual foundation for the system developed here.

However, the publication only described the system at a high level, with no details on the implementation. As a result, one of the first steps of this project was to analyze the initial structure, identify the required functionalities, and define how to implement each part with real components and tools available in the lab. Several additional elements were added to improve the system's performance and robustness, including filtering stages, signal buffering, and precise acquisition and control mechanisms.

To achieve the overall goal, the project was broken down into the following secondary objectives:

1. Understand the operation of a Direct Digital Synthesizer (DDS) and configure it to generate a controllable frequency sweep.
2. Design and fabricate a phase shifter with a linear response across the frequency range of interest.
3. Implement wireless inductive coupling using reader and sensor coils, and characterize resonance behavior.
4. Use a phase detector to measure the phase difference between transmitted and received signals.
5. Digitize the analog phase information using a high-resolution ADC.
6. Synchronize all modules via a microcontroller and develop acquisition routines.
7. Simulate and test each subsystem independently before full integration.
8. Build a complete prototype with proper signal integrity and EMI mitigation.

Each of these points was progressively developed and validated, forming the basis of the methodology described in the next sections.



**Figure 1 – Block diagram of the wireless readout system proposed by Salpavaara et al. (2011).**  
*This schematic served as the initial reference for the system architecture. Additional functional blocks, such as filters and signal conditioning circuits, were later integrated during the design process to improve accuracy and stability.*

## 2. Methodology and Design Strategy

### 2.1 Methodology

The project was structured in several logical phases, progressing from initial analysis to full system integration. Each step was designed to validate specific functional blocks and minimize the risk of error during final assembly.

#### 1. Concept and Design

The first step involved studying the scientific paper in detail to understand the operating principles of the inductively coupled resonant sensors. Based on this analysis, I defined the required functions and

compiled a list of components. The most complex parts, such as the phase detector (AD8302), the analog-to-digital converter (ADC), and the DDS signal generator, were tested individually using evaluation boards provided by the manufacturers in order to fully understand their behavior before integration. In parallel, I designed certain critical blocks myself—such as the phase shifter and various RF filters—since they required custom RF circuitry tailored to the specific needs of the application.

## 2. Component Testing

After validating the critical components, I proceeded to develop and test each functional module independently. Each block (signal generation, analog-to-digital conversion, phase detection, etc.) was assembled and evaluated separately to confirm its functionality and simplify troubleshooting if necessary. The circuits were first evaluated through computational simulations using Qucs, and then validated experimentally using measurement tools such as oscilloscopes, impedance analyzers, and a vector network analyzer (VNA).

## 3. Modular Prototyping

Once the individual modules were validated, I progressively assembled them to build the complete system. This modular approach allowed me to test interactions between blocks without needing a finalized PCB and made it easier to apply design corrections when needed.

## 4. Integration on PCB

After confirming the overall operation of the system, I moved on to integrating all modules into a single printed circuit board (PCB). This stage required special attention to signal routing, power supply management, and ensuring system stability in a practical environment. In addition, I had to generate the standard manufacturing files (such as Gerber files and the Bill of Materials) and prepare them for production through a third-party PCB manufacturer.

## 2.2 Constraints

The progress of this project was shaped by a series of technical, organizational, and time-related constraints, each of which required constant adaptation and strategic decision-making.

One of the first major challenges was the discovery of a completely new technical environment. Instruments such as the impedance analyzer and the vector network analyzer (VNA) available in the lab were significantly different from the ones I had previously used during my studies. In addition, the PCB fabrication techniques, including the thermal transfer (iron-based) method, were unfamiliar to me. This steep learning curve demanded time and effort to overcome. Fortunately, the support and guidance of my colleagues enabled me to quickly become proficient with these tools.

Working in a medical research setting also involved strict safety and regulatory requirements. Before I was allowed to handle any equipment, I completed a mandatory safety training session, reviewed detailed risk assessments, and signed several prevention-related documents. This aspect of the internship helped me develop a stronger sense of professional responsibility and attention to protocol, especially in a high-stakes healthcare context.

From a logistical standpoint, I was fortunate to work in a well-equipped laboratory where most of the required components were already available. When additional parts were needed, they could be ordered without significant budgetary or administrative delays, ensuring smooth project continuity.

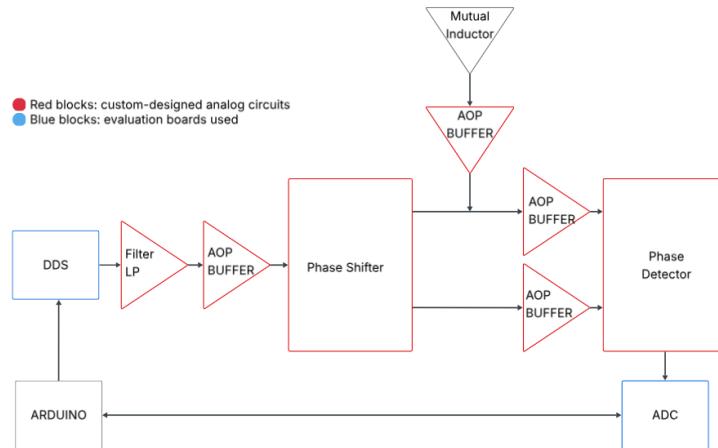
However, the most limiting factor was undoubtedly time. Given the complexity and scope of the project, it quickly became clear that it could not be fully completed within the two-month internship. As a result, I had to establish clear priorities, focusing my efforts on the most essential and technically challenging parts of the system.

Although I worked independently on the development, I was not isolated. Regular discussions and spontaneous help from team members were invaluable in troubleshooting and refining my approach. This form of guided autonomy helped me grow in confidence, discipline, and technical maturity. That said, the absence of a dedicated teammate sometimes made problem-solving slower and more demanding, pushing me to develop stronger decision-making skills and the ability to work effectively under pressure.

## 2.3 Project Specifications

### 1. System Overview

While the original block diagram proposed in the reference paper served as a foundation, it was improved and adapted to better suit the specific constraints of the target application.



**Figure 2 – Functional block diagram of the complete system**

Red blocks represent analog circuits specifically designed for this application (filters, buffers, phase shifters). Blue blocks correspond to the evaluation boards used. The system measures the phase shift introduced by a passive LC sensor inductively coupled to the reader coil. It relies on a DDS generator controlled by an Arduino, a phase detector (AD8302), and high-resolution data acquisition via the AD7177-2.

The updated system includes the following functional modules:

1. Frequency Generation – DDS (AD5932)  
 A programmable sine wave generator controlled by a microcontroller via Serial Peripheral Interface (SPI). A frequency sweep is performed from 1 MHz to 25 MHz to detect the sensor's resonant frequency. This frequency range was chosen to cover the Industrial, Scientific, and Medical (ISM) band of 13.56 MHz.
2. Low-Pass Filter for DDS harmonics  
 A passive LC low-pass filter was added to suppress harmonics generated by the DDS, improving the quality of the excitation signal.
3. Phase Shifting  
 A third-order passive Bessel filter was designed and inserted to establish a known reference phase, ensuring precise phase comparison over the entire sweep range.
4. Phase Detection – AD8302  
 The AD8302 chip measures the phase shift between the transmitted and received signals and outputs a corresponding voltage, indicating how the resonance shifts.
5. Data Acquisition – ADC (AD7177-2)  
 A high-resolution 24-bit ADC digitizes the phase voltage, synchronized with the frequency sweep. Measurements are averaged and logged to increase accuracy.
6. System Control – Arduino  
 The microcontroller orchestrates the sweep, manages the SPI communication, and performs data acquisition. All data is logged through the serial interface for post-processing.
7. Inductive Coupling  
 A reader coil transmits the signal, which is received by a passive LC sensor. The resonant frequency of this sensor varies with physical changes (e.g., pressure or deformation).

## 2.4 Component Selection

### **Objective:**

In collaboration with my supervisor, I have selected the most essential components required for reproducing the wireless measurement system. These components form the backbone of the architecture. As the project advanced, additional elements were identified and integrated based on functional needs and testing feedback.

### **Main Components and Their Roles:**

- **AD8302 – Phase Detection:**  
 This vector detector is used to convert the phase difference between two RF signals into a voltage output. Its precision and wide frequency range make it ideal for accurately capturing phase variations in our application.
- **AD5932 – Signal Generation (DDS):**  
 The Direct Digital Synthesis (DDS) chip generates the excitation signal for the sensing system. It supports programmable frequency sweeps, which are crucial for probing the resonant frequency of the inductively coupled LC sensor.
- **AD7177 – Signal Processing (ADC):**  
 This high-resolution analog-to-digital converter digitizes the analog output from the phase

detector (or amplitude signal, if used). Its 24-bit resolution allows for precise acquisition of low-level signals in a noisy environment.

- **Passive Filters – Phase Shifting:**

A combination of low-pass and high-pass filters was implemented using resistors, capacitors, and inductors. These filters contribute to conditioning the RF signals, controlling bandwidth, and ensuring proper phase delay where necessary.

## 3. Hardware Subsystems

### 3.1 Phase Shifter Design and Testing

The objective of this part of the project was to design a passive phase shifter capable of introducing a fixed  $90^\circ$  phase shift between two RF signals in the 1–25 MHz frequency range, with possible extension up to 50 MHz. The main challenges included maintaining phase linearity, ensuring moderate attenuation, and keeping the design feasible with the components and tools available in the lab.

#### 1. Preliminary Study: Limitations of an Analytical Approach

An initial method found on a specialized [RF website](#), based on analytical equations, was simulated in QucsStudio. While promising in theory, the simulations only produced coherent results at very high frequencies (hundreds of MHz), making the method unsuitable for the target 1–25 MHz band.

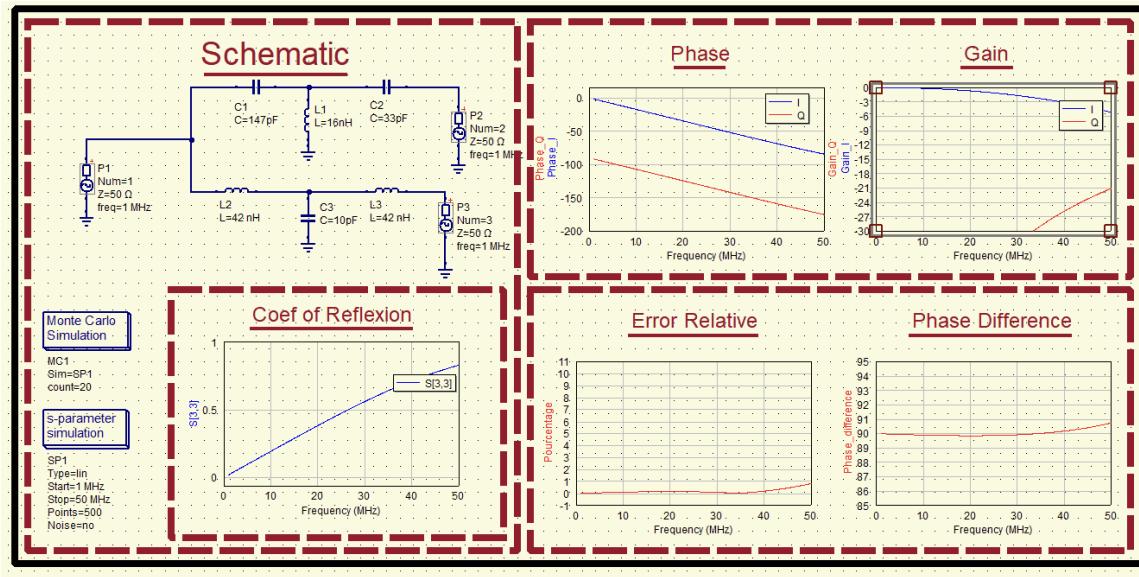
#### 2. Empirical Design Using Bessel Filters

A more empirical approach was then adopted using QucsStudio's filter design tool. The phase shifter was built from two third-order passive filters:

- a high-pass filter (HPF),
- a low-pass filter (LPF)

Both are designed with a Bessel response to ensure maximum phase linearity. The capacitor values were chosen based on available lab stock, and the inductances were adjusted accordingly.

**Figure 3** shows that the phase shifter provides an overall linear phase response, with a phase difference close to  $90^\circ$  across a wide frequency range. The relative error remains below 2%, confirming the stability and accuracy of the design. For exact component values and simulation settings, see [Appendix A](#).



**Figure 3 –** Qucs schematic of the initial phase shifter (LPF + HPF)

#### [Appendix A – Zoomed view of the Qucs schematic \(LPF + HPF\)](#)

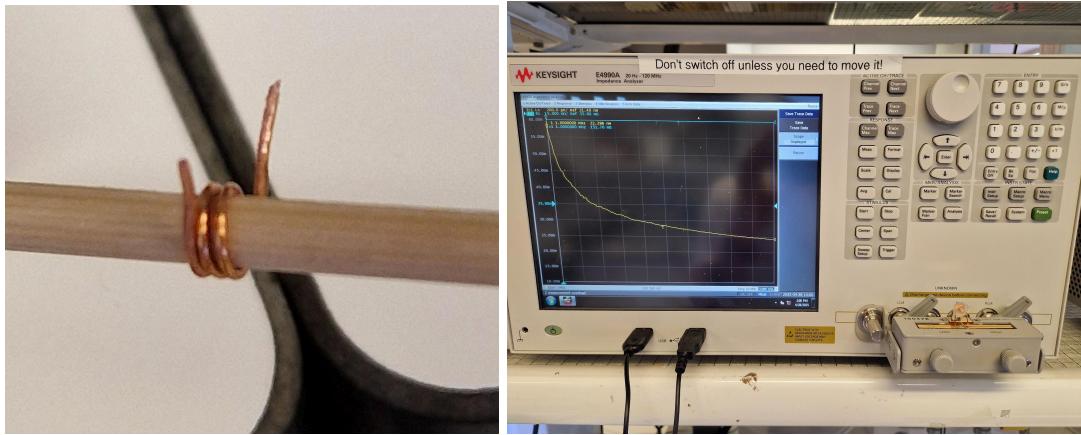
### 3. Manual Fabrication of Inductors

As the required capacitors were available, I focused on manually winding the inductors. Using **Coil64**, I entered:

- the mandrel diameter (3.2 mm wooden stick),
- the copper wire diameter,
- the target inductance value.

Once the coils were wound, I manually sanded the ends of each copper wire. This step was essential to remove the insulating protective coating, which is typically applied to prevent corrosion or short circuits when wires are in contact. Without sanding, electrical continuity cannot be ensured, and the inductors would not function properly when soldered into the circuit.

The number of turns was calculated automatically by the Coil64 software based on the target inductance. Once the coils were wound, they were measured using a Keysight E4990A impedance analyzer to verify that their inductance values matched the expected results. Before proceeding with the measurements, the instrument was calibrated using fixture compensation to ensure accuracy and repeatability. This calibration step followed the official procedure provided by Keysight, which includes open, short, and load compensation steps as outlined in their documentation [E4990A Fixture Compensation Guide](#). The fabrication process and the measurement setup are shown in **Figure 4**.



**Figure 4** – Left: coil winding process; Right: inductor measurement using the E4990A

Once validated, the inductors were fixed with hot glue to prevent movement that could alter their value over time. An example is shown in **Figure 5**.

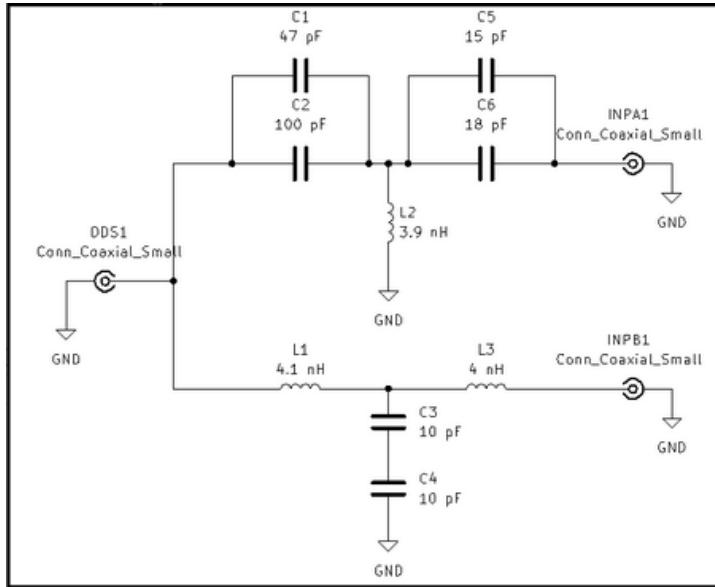


**Figure 5** – Fixed inductor after validation

#### 4. First PCB Fabrication and Testing

A first prototype PCB was designed using KiCad, respecting several basic RF layout rules:

- symmetrical signal paths,
- continuous ground planes,
- equal path lengths for both branches.



**Figure 6 – Initial PCB schematic in KiCad (with the last value)**

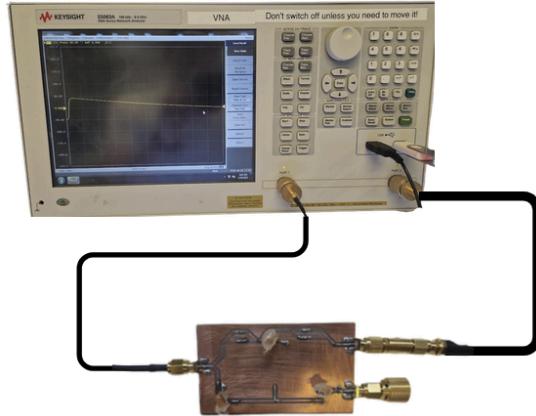
#### Appendix B – PCB after Corrosion and final

**Initial testing** was performed using an oscilloscope to verify the general behavior of the phase shifter. For more accurate characterization, a **Keysight E5063A Vector Network Analyzer (VNA)** was used. Since the VNA had no differential port, the following workaround was implemented to evaluate the phase difference:

1. **VNA Calibration:** A full Open-Short-Load-Thru (OSLT) calibration was performed prior to measurements, using the appropriate calibration kit. This step was essential to correct systematic errors due to cable losses, mismatch, and connector variability, ensuring reliable and repeatable results. Calibration was conducted at the SMA connectors at the end of the test cables.
2. **Single-ended Phase Measurement Procedure:**
  - First, the phase response of input 1 was measured while input 2 remained disconnected (open with an SMA adaptor).
  - Then, the reverse setup was applied: input 2 was measured while input 1 remained open.
  - This approach allowed both branches of the phase shifter to be characterized independently.
3. **Phase Comparison:** The phase data from both measurements were exported in **.csv** format and post-processed in Excel. The phase difference between the two paths was computed to determine the actual performance of the phase shifter across the frequency sweep.

This method, while indirect, provided a reliable estimate of the differential phase behavior in the absence of a true differential measurement port.

The measurement configuration is illustrated in **Figure 7**.

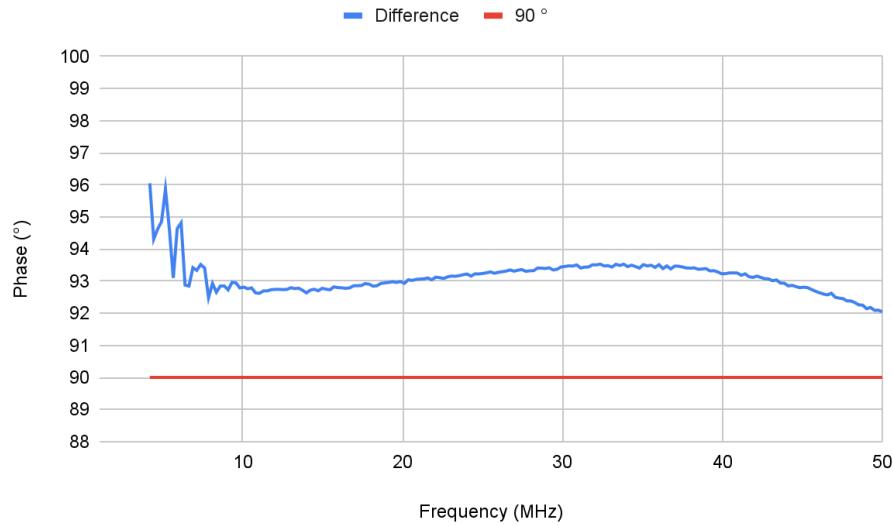


**Figure 6** – VNA measurement setup

## 5. Phase Shift Adjustment and Optimization

The first prototype did not produce the expected  $90^\circ$  phase shift. Curve analysis suggested that the inductance values were too high. I recalculated and hand-fabricated several new inductors with reduced values and re-tested them.

The updated results, shown in **Figure 7**, indicate a significant improvement in phase behavior. Additional comparisons of different configurations are presented in [Appendix C](#).



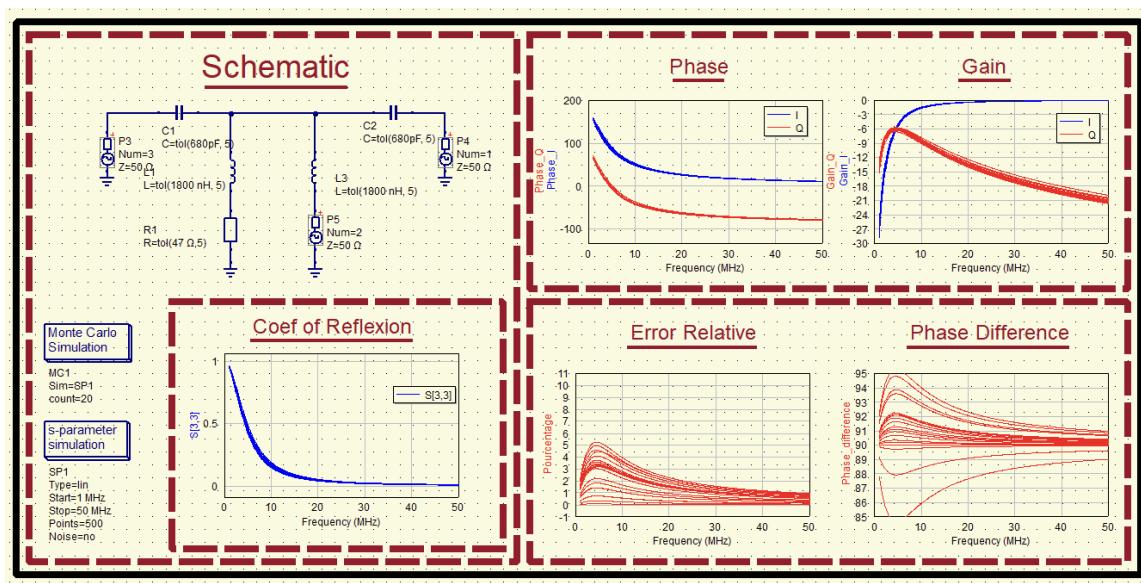
**Figure 7** – Final phase shift curve after adjustment

[Appendix C](#) – Comparison of phase shift curves for different inductance values

After several iterations, a stable phase shift close to  $90^\circ$  was achieved within the 8–50 MHz range. Beyond 8 MHz, the response became unstable. This does not represent a problem in the targeted application since most of LC sensors in medical devices operate at 13.56 MHz.

## 6. Alternative Design on Breadboard

Further investigation revealed a gain imbalance between the two branches. I then built a second version using a hybrid filter structure on a breadboard, seen on the Qucs schematic in **Figure 8**. At this time, selecting inductors from available lab stock (in  $\mu\text{H}$ ), which were easier to find with precise values.



**Figure 8** – Qucs schematic of the second hybrid phase shifter

### [Appendix D – Zoom Qucs schematic of the second hybrid phase shifter](#)

Even in the worst-case scenario with component tolerances, the phase shift error remained below 5%. By characterizing the response beforehand, any imperfections could be directly corrected in software during signal processing.

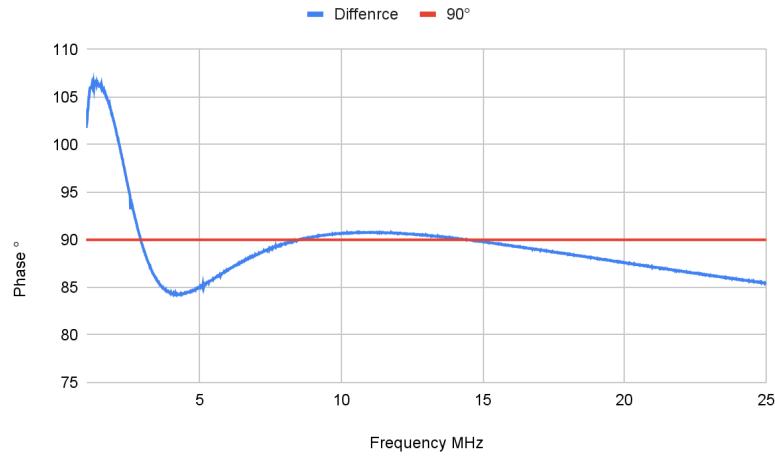
## 7. Limitations of the Final Version

Despite the progress made, this version had two significant limitations :

- the use of  $\pm 10\%$  tolerance inductors, resulting in frequency detuning,
- the difficulty of manually producing precise inductance values, particularly as low as 1800 nH.

These imperfections caused a parasitic resonance phenomenon, visible as a sharp dip in the phase response over a narrow frequency band, as well as a pronounced peak at the very beginning of the sweep. These artifacts, clearly observed in **Figure 9**, mainly occur at frequencies below the target

band. In contrast, within the 10–15 MHz range, the phase response remains stable and close to  $90^\circ$ , confirming the filter's effectiveness in the intended operating range. Additional phase measurements from various test configurations are provided in [Appendix E](#).



**Figure 9** – Effect of parasitic resonance on the phase shift curve

#### [Appendix E – The different phase measurements of the test assemblies](#)

These issues indicate that industrial wound inductors with tighter tolerances would be better suited for future high-precision versions.

## 8. Final Exploratory Concept: Adjustable Phase Shifter (Unfinished)

Toward the end of the project, I began designing an adjustable phase shifter using Qucs. The schematic, shown in [Appendix F – Zoomed view Qucs schematic of the adjustable phase shifter](#), includes tunable components that enable dynamic adjustment of the phase between the two branches of the circuit. Specifically:

- Variable damping resistors are placed after the inductors to reduce unwanted resonances observed in earlier versions. Their values can be adjusted to achieve a more stable response;
- Adjustable pull-down resistors help stabilize the gain and prevent floating voltages on certain branches;
- A variable shunt capacitor, placed only on one branch of the phase shifter, introduces a controlled imbalance that allows real-time tuning of the phase difference between the two paths.

This theoretical design offers the possibility of fine phase tuning without modifying the circuit structure, simply by adjusting the selected component values.

### Conclusion

This phase shifter project allowed me to explore several design approaches for generating a fixed phase shift in the lower RF range. After discarding an analytical method unsuitable for our frequency

band, I developed several working prototypes using Bessel filters, which were tested through simulation and hardware implementation.

The main challenges involved inductor precision, gain imbalance between signal paths, and phase shift instability at lower frequencies, particularly below 15 MHz. Despite these constraints, a functional and sufficiently stable version was achieved from 10 MHz upward, with the possibility to compensate for remaining deviations through software post-processing.

This project helped me strengthen my skills in RF circuit design, manual fabrication of passive components, QucsStudio simulation, and instrument-based testing using both the Keysight E4990A impedance analyzer and the Keysight E5063A VNA. It also taught me to adapt technical decisions to practical constraints and to iterate effectively based on experimental results.

## 3.2 Phase Detection Using AD8302

The phase detection block precisely measures the phase difference between two RF signals and converts this difference into an analog DC voltage, which can then be utilized by the rest of the system. This voltage output allows us to quantify the variations in the monitored parameter. To determine if the phase detector (AD8302) was suitable for our setup, I first needed to understand its operating principle. Then, I designed an evaluation board to test its performance under real conditions.

### 1. Characteristics and Operating Principle of AD8302

The AD8302 is a dual-function analog signal processor capable of simultaneously measuring two key parameters of RF signals:

- the phase difference between two input signals,
- and the amplitude ratio (or gain) between those signals.

In this project, only the phase output (VPHASE) was used. The AD8302 generates an analog voltage that is linearly proportional to the phase difference between its two differential inputs (INPA and INPB) over a range from  $0^\circ$  to  $180^\circ$ :

- A  $0^\circ$  phase shift produces a voltage of approximately 1.8 V,
- A  $\pm 90^\circ$  shift corresponds to  $\sim 0.9$  V,
- And a  $\pm 180^\circ$  shift results in an output around 0 V.

The internal architecture ensures high linearity, especially near  $\pm 90^\circ$ , which corresponds to the most sensitive region of the phase output curve. With a bandwidth extending up to 2.7 GHz, the AD8302 is particularly well-suited for RF applications such as this one.

For more insight into the internal operation, a **block diagram of the AD8302** is provided in **Appendix X**, along with a link to the official Analog Devices datasheet.

For further technical details, refer to the official Analog Devices datasheet:

[AD8302 Datasheet - Analog Devices](#)

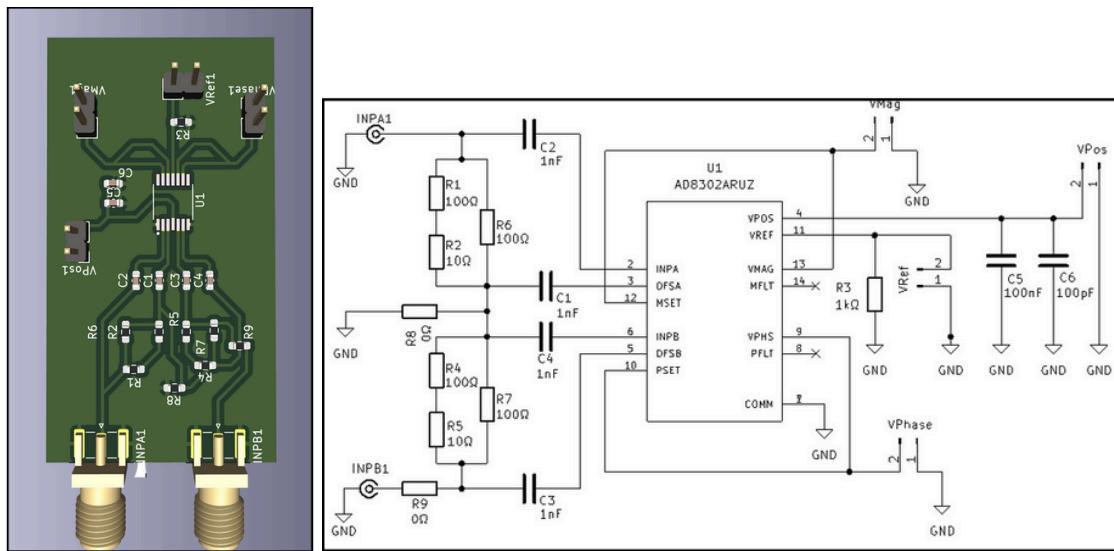
## 2. Custom Evaluation Board Design

To test the performance of this component under realistic conditions, I designed and produced a custom evaluation board directly inspired by the reference schematic provided in the Analog Devices datasheet:

- Conducted a thorough analysis of the datasheet to identify critical components and comply with recommended guidelines (filtering, decoupling, and input impedance matching).
- Adapted the Bill of Materials (BOM) to suit available laboratory resources.
- Addressed component availability issues: the required  $52.3\ \Omega$  resistor was unavailable, thus I chose to achieve this value through a parallel resistor combination:

$$Req = (100 + 10) // 100 = \frac{110 * 100}{110 + 100} = 52.38$$

The PCB layout was created using KiCad, applying essential RF design rules: short and symmetric signal traces, continuous ground planes, and tight decoupling capacitors near power pins. The schematic and layout are shown in **Figure 11**.



**Figure 11 – AD8302 Phase Detector: Custom PCB Layout and Corresponding Schematic Diagram**

### Appendix G — Phase detector PCB (AD8302) before corrosion and final

- PCB layout was designed in KiCad, adhering strictly to RF design best practices (short traces, symmetric inputs, continuous ground planes, precise decoupling).
- PCB fabrication was done using the thermal transfer method (iron transfer technique).

## 3. Correction of an Initial Design Flaw

After assembling the first PCB prototype, initial testing revealed an unsuccessful operation. Upon careful investigation, I identified a schematic error inherited from the original reference: two pins had been accidentally swapped.

I corrected the schematic, produced a revised PCB, fabricated a new board, and reassembled the circuit. nnn

#### 4. Experimental Validation Methodology

To rigorously validate the operation of the phase detector, I established an experimental procedure based on signal propagation delays using two coaxial cables of different lengths:

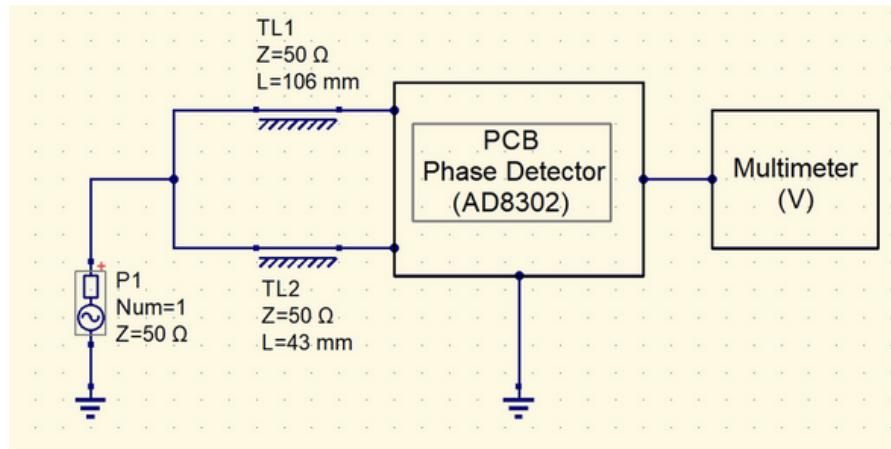
- **Cable TL1:** 106 mm
- **Cable TL2:** 43 mm

An RF signal was generated and split into two identical paths. Each path passed through one of these coaxial cables. This setup created a known phase difference between the two signals. The theoretical phase difference introduced by the cable length difference was calculated using the formula:

$$\Delta\phi = \frac{2\pi \cdot \Delta L}{\lambda}$$

- $\Delta L$  represents the length difference between the two cables.
- $\lambda$  is the RF signal's wavelength within the cable.

The two coaxial cables were connected to the inputs of the AD8302 and the output voltage was measured using a precise multimeter. The physical setup of this test is shown in **Figure 12**, which illustrates the signal generator, cable connections, and measurement tools used during the validation process.



**Figure 12** — Test setup for VNA measurement

#### 5. Measurement Conditions and Results

- **Injected RF signal:**

For this experimental validation, I varied the RF signal frequency from 1 MHz up to 166 MHz, recording multiple measurement points across this frequency range. The frequency of 166 MHz was particularly critical, as at this frequency, the selected cable lengths theoretically corresponded to a phase difference close to **180°**.

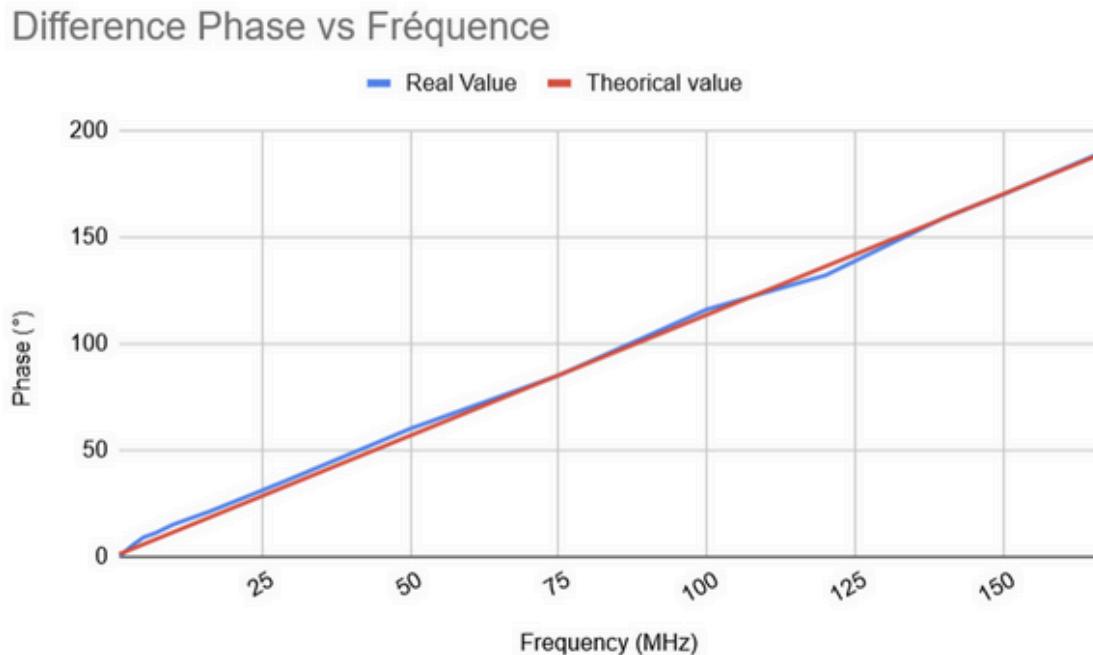
- **Measurement procedure:**
  - Powered the AD8302 circuit board with a stabilized laboratory benchtop power supply.
  - Generated the RF test signals using an RF signal generator.
  - Measured the output voltage (V<sub>PHASE</sub>) of the AD8302 using a precise multimeter.
  - Recorded multiple voltage measurements at various frequencies, especially around 166 MHz, to thoroughly characterize the device's response.
- **Data logging:**  
All measured voltages were systematically logged in an Excel spreadsheet, allowing direct comparison with theoretical predictions.

## 6. Results Analysis and Validation

The measured voltages closely matched the theoretical expectations, validating the performance of the phase detector.

As shown in **Figure 13**, the blue curve represents the calculated phase difference between the two coaxial paths, while the red curve shows the corresponding output voltage measured at the AD8302 output. A critical point appears near 166 MHz, where a phase shift of approximately 180° was expected. At this frequency, the output voltage was around 0 V, in accordance with the datasheet specifications.

This strong agreement between theoretical and experimental data confirms the accuracy and reliability of the AD8302 phase detector when integrated into the system.



**Figure 13** —Measured Phase Difference (blue) and Corresponding Output Voltage (red) as a Function of Frequency

## Intermediate Conclusion

This comprehensive validation process—covering design, debugging, and rigorous experimental verification—enabled me to definitively confirm the AD8302 as a robust and precise central component for phase measurement in my project. This critical step was indispensable before proceeding with full system integration.

## 3.3 Frequency Generation Using AD5932 (DDS)

Before implementing any measurement or control system, it was essential to verify that the **AD5932** could generate a clean and stable sinusoidal signal across the desired frequency range. This step was critical to ensure that the LC sensor could be properly excited in the 1–25 MHz band.

### 1. Initial Validation Using Analog Devices Software

For this validation, I used the AD5932EBZ evaluation board, which was connected directly to a computer and configured using the official Analog Devices software. This graphical interface allowed easy adjustment of:

- the start frequency, the frequency increment, and the number of steps.

No microcontroller was needed at this stage, making it convenient to quickly test the core functionality of the DDS.

However, when observing the output signal on an oscilloscope, I noticed that the waveform was far from ideal. Although the frequency was correct, the signal showed significant harmonic distortion, which is typical of DDS architectures due to their internal digital nature.

To improve the spectral purity, I decided to design and add a passive LC low-pass filter at the DDS output, aiming to suppress unwanted high-frequency components.

The FFT spectra captured using the oscilloscope, both before and after filtering, are presented in [Appendix I](#).

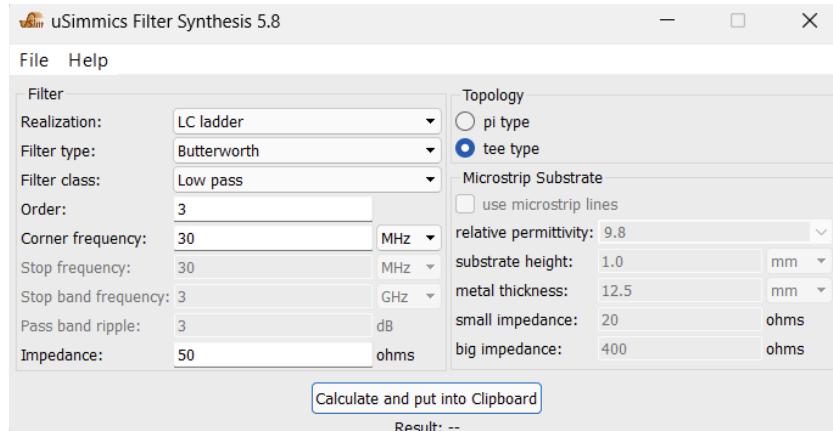
### 2. LC Filter Design and Simulation

The filter values were initially calculated using QucsStudio's built-in **Filter Synthesis** tool (*Tools → Filter Synthesis*), with the parameters shown in **Figure X**, you can see the schematic in the [Appendix H](#). A third-order passive LC low-pass filter was synthesized using the Butterworth response, targeting a 30 MHz cutoff frequency and  $50\ \Omega$  impedance.

The chosen component values, later slightly adjusted through simulation, were:

- $L_1$  and  $L_2 = 330\ nH$
- $C = 330\ pF$

The goal was to preserve signal integrity up to 25 MHz while effectively attenuating higher-frequency harmonics.



**Figure X – Filter Synthesis settings in QucsStudio used to generate the initial LC low-pass filter**

#### Appendix H- QucsStudio schematic Low Pas Filter

After physical implementation of the filter, I re-analyzed the DDS output using a Rohde & Schwarz oscilloscope with FFT functionality. The measurements showed:

- a clear reduction in harmonics,
- a more consistent amplitude,
- and an overall improvement in signal quality across the sweep range.

#### Appendix I – FFT measurements before and after filtering

### 3. Automated Frequency Sweep with Arduino

Once the output signal was verified and cleaned, I replaced the PC-based software with an autonomous Arduino-based control system. The goal was to embed the DDS in the complete acquisition chain.

The Arduino Uno was connected to the AD5932EBZ board via:

- the **SPI interface** (FSYNC, SDATA, SCLK),
- the **RESET** pin,
- and the **CTRL** pin, which triggers each frequency increment.

The Arduino sketch was organized into several key functions:

- **writeDDS(uint16\_t w)**  
Sends a 16-bit configuration word to the DDS over SPI.
- **initDDS()**  
Sends the full initialization sequence to the DDS, configuring:
  - the start frequency (1 MHz),

- the step size (100 kHz),
- and the total number of steps (240).

These values were precomputed and stored in a lookup table.

- **pulseCTRL()**

Sends a short high pulse on the CTRL pin to advance to the next frequency step. A local variable tracks the current frequency (`curFreq`), which increments after each pulse.

In the main loop, the Arduino sent a CTRL pulse every 0.5 seconds, resulting in a smooth sweep from 1 MHz to 25 MHz. The current frequency was printed to the Serial Monitor for tracking purposes.

#### [Appendix J – Arduino final code](#)

This setup provided a stable, spectrally clean, and precisely controlled excitation signal that could be reliably used to drive the LC sensor.

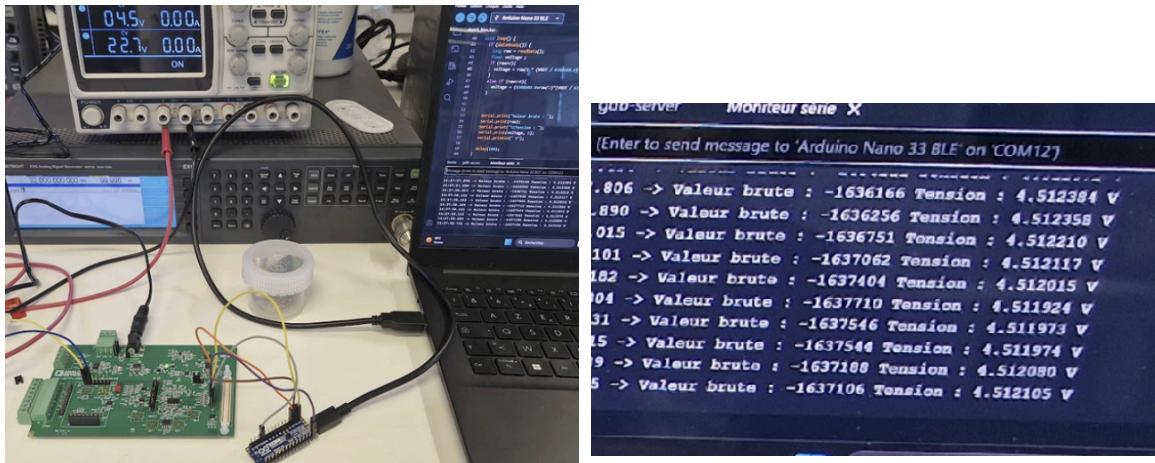
### 3.4 Signal Acquisition Using AD7177-2

Before integrating the ADC into the complete system, a standalone test was carried out to validate its proper operation. A constant 4.5 V voltage was applied to its input using a laboratory power supply. The signal was read directly by the Arduino via the SPI interface, and the measured values were very close to the expected voltage, confirming that the converter was functioning correctly.

The test setup and the measured result are shown in **Figure 14**. It displays the wiring between the AD7177-2 evaluation board and the Arduino, as well as the voltage values displayed in the serial terminal.

The ADC is initialized using the functions `resetADC()` (register reset) and `writeRegister()` (conversion parameter configuration). The `readData()` function reads a raw 24-bit sample, which is then converted into voltage.

For detailed information, refer to [Appendix K](#) for the final Arduino wiring, [Appendix J](#) for the complete Arduino code, and **Figure 15** for the corresponding functional block diagram.



**Figure 14** – Standalone test of the AD7177-2 ADC: top – connection of the evaluation board with the input set to 4.5 V; bottom – photo of the serial terminal showing the measured voltage.

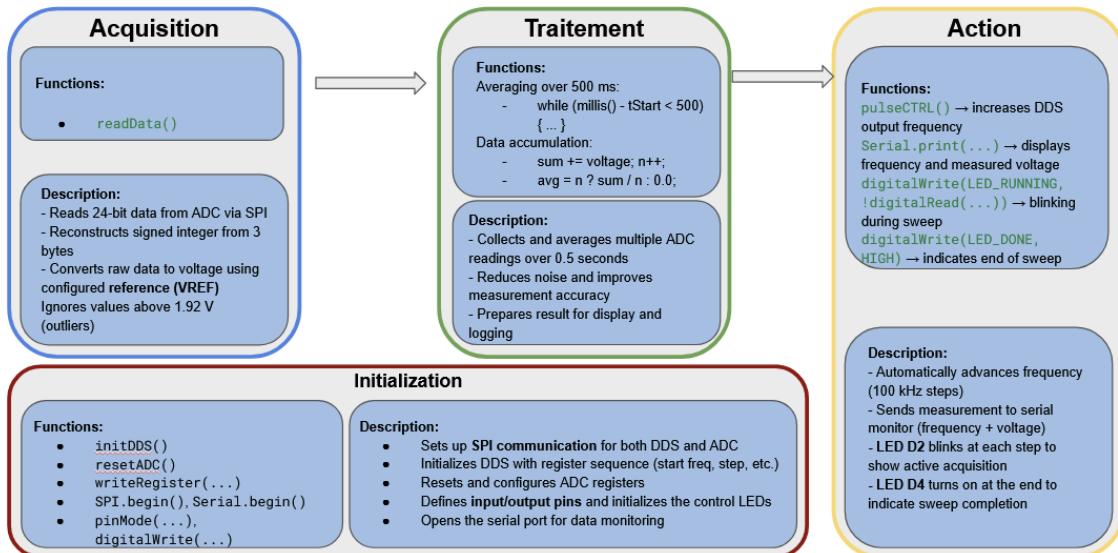
### 3.5 Integration ADC into the Frequency Sweep System

After validating the ADC functionality, it was integrated into an automated acquisition loop alongside the DDS-based frequency sweep. At each frequency step (from 1 MHz to 25 MHz, with 100 kHz increments), the system collects voltage measurements over a 500 ms period. The average of these values is then printed to the serial terminal along with the corresponding frequency.

The Arduino program handles all stages of the process:

- initializing the peripherals (DDS, ADC),
- triggering frequency increments via the DDS CTRL pin,
- performing SPI readings from the ADC,
- computing the average voltage,
- displaying the result,
- and managing two status LEDs:
  - one LED (D2) toggles during each measurement cycle,
  - the other (D4) lights up at the end of the full sweep.

This structure is summarized in the functional diagram shown in **Figure 11**. The full source code is available in **Appendix J**, and the Arduino wiring details are provided in **Appendix K**.



**Figure 15 – Functional block diagram of the Arduino program**

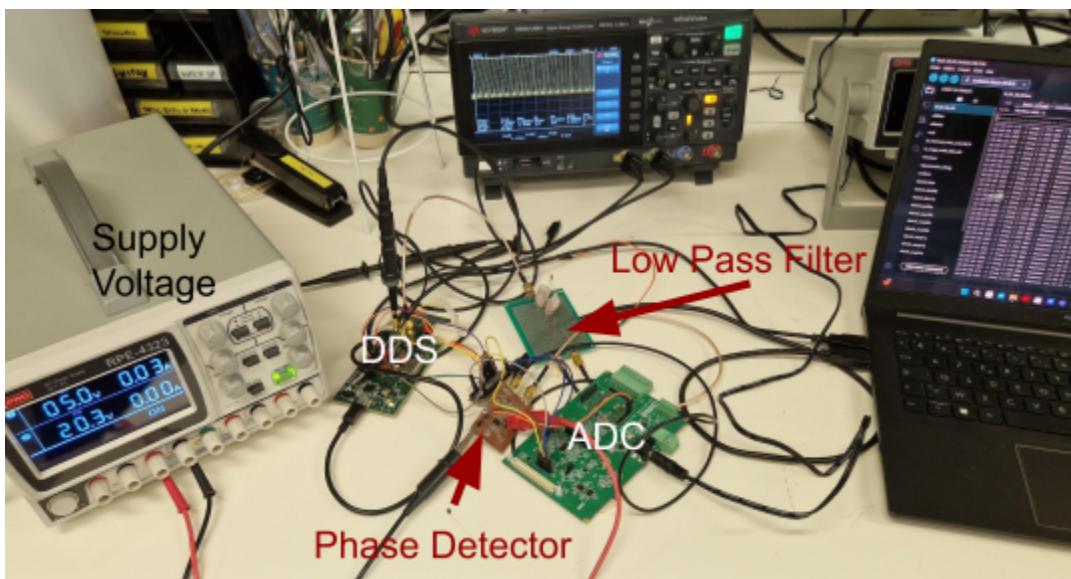
The diagram presents the logic flow of the code, from peripheral initialization to measurement and result display. It shows how each function contributes to the automated frequency sweep and acquisition, including visual status indicators.

# 4. System Integration and Testing

## 4.1 Experimental Validation of the Complete System

**Figure 16** below shows the complete setup used to validate the full system. The sinusoidal signal generated by the DDS passes through an LC low-pass filter, then into two coaxial cables with a length difference of 63 cm, creating a frequency-dependent phase shift. These two signals are connected to the AD8302 phase detector, and its output is read by the AD7177-2 ADC via an Arduino.

During the test, one phase value is acquired every 100 kHz from 1 MHz to 25 MHz, with each point being the average of 500 ms of data.

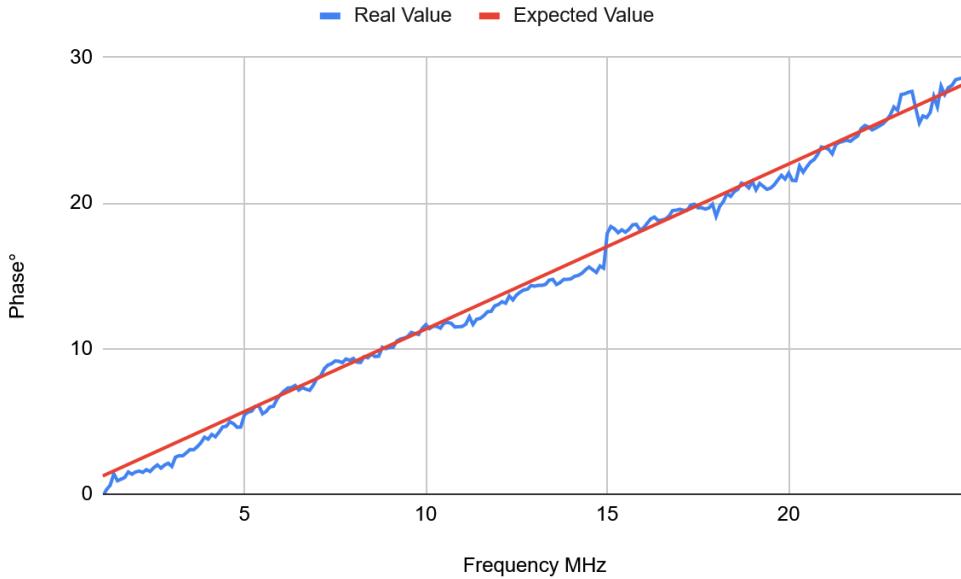


**Figure 16** – Final measurement setup

Overview of the test setup : DDS → LC filter → two coaxial cables → AD8302 → ADC → Arduino.

The results were processed to convert the measured output voltages from the AD8302 into phase angles (in degrees). These are plotted in **Figure 16**, which shows the measured phase difference as a function of frequency.

The curve clearly demonstrates that the phase shift increases smoothly and predictably with frequency, reaching close to  $180^\circ$  at 25 MHz, as expected from the theoretical model. This validates the proper functioning of the full system, including the DDS, filtering stage, phase detection, and ADC acquisition.



**Figure 16 – Measured phase shift vs frequency**

Measured phase difference (in degrees) between the two signals, after voltage-to-angle conversion. The phase shift evolves as expected across the 1–25 MHz range.

## 4.2 Final PCB Design Integrating All Modules

To make the system more compact, reliable, and easier to operate, a printed circuit board (PCB) was designed to combine all the modules previously validated individually. This board integrates:

- an LC filter to attenuate DDS harmonics,
- a hybrid phase-shifting filter to produce a stable phase difference between two paths,
- a phase detector (AD8302),
- buffer op-amps to isolate each functional block,
- an Arduino interface for frequency sweep control and SPI data acquisition,
- a DDS input to connect the sinusoidal source.

This PCB-based integration reduces interference from jumper wires, improves signal stability, and results in a more robust system for future testing.

The schematic was split into two parts for better readability. It is presented in **Appendix L** (Part 1: Phase Shift System, Filter Low Pass, Phase Detector and Arduino) and **Appendix N** (Part 2: Leds Process, Signal Coil Input, Snuging Resistor and Selector Reader ).

- **Appendix L – Electrical schematic Part 1**
- **Appendix N – Electrical schematic Part 2**

The final PCB routing, designed using KiCad, is also included in :

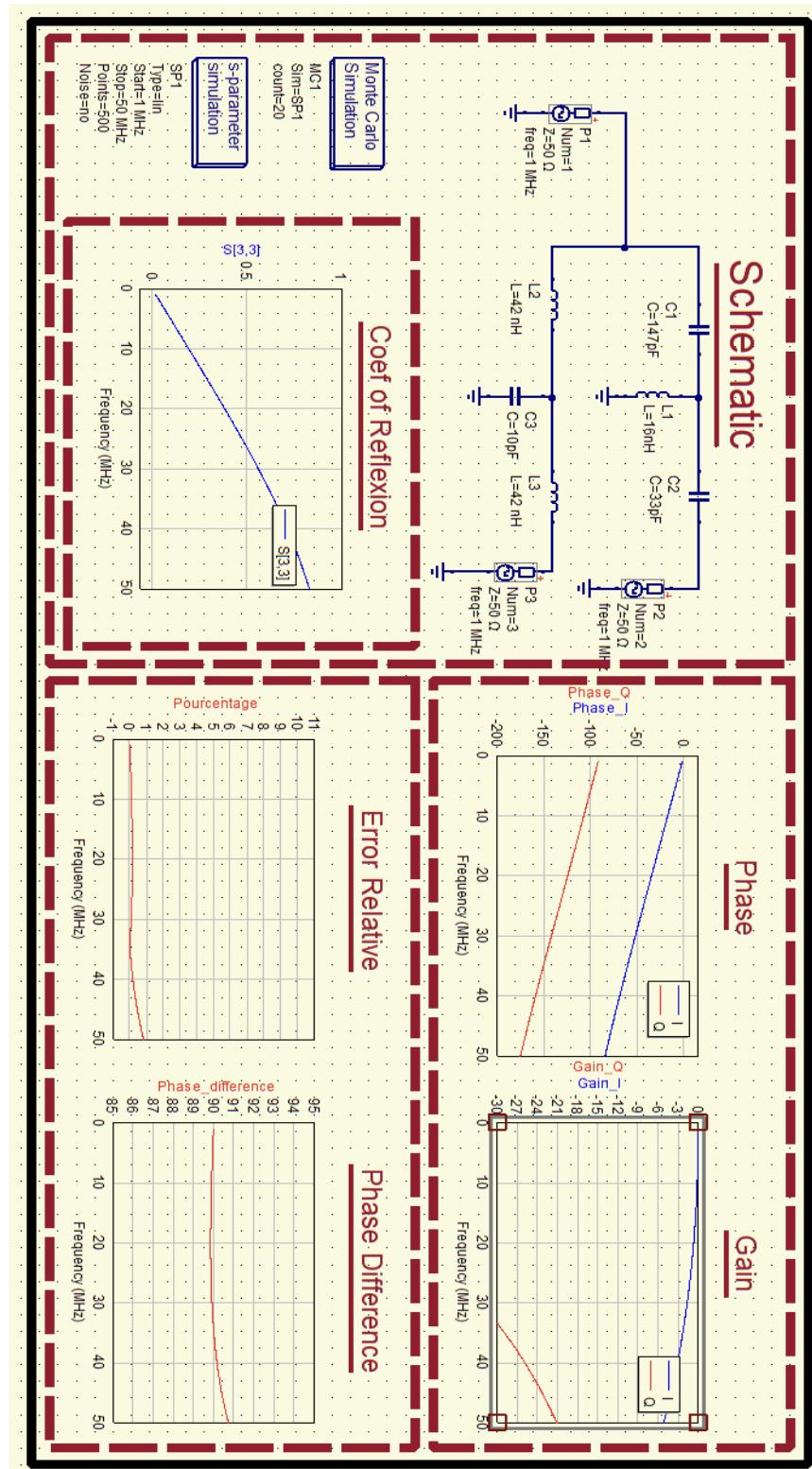
- **Appendix N1— Implantation**
- **Appendix N2 — Left Top copper & Right Bottom copper**
- **Appendix N3 — View 3D of PCB**

## Conclusion

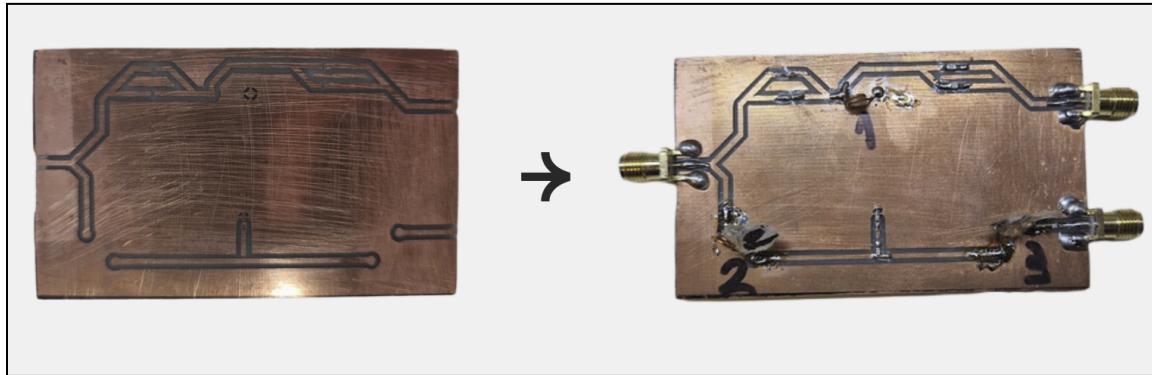
The conclusion will summarize the validated components, remaining tasks, and overall progress made during the internship. It will reflect on the project outcomes and lessons learned.

## References

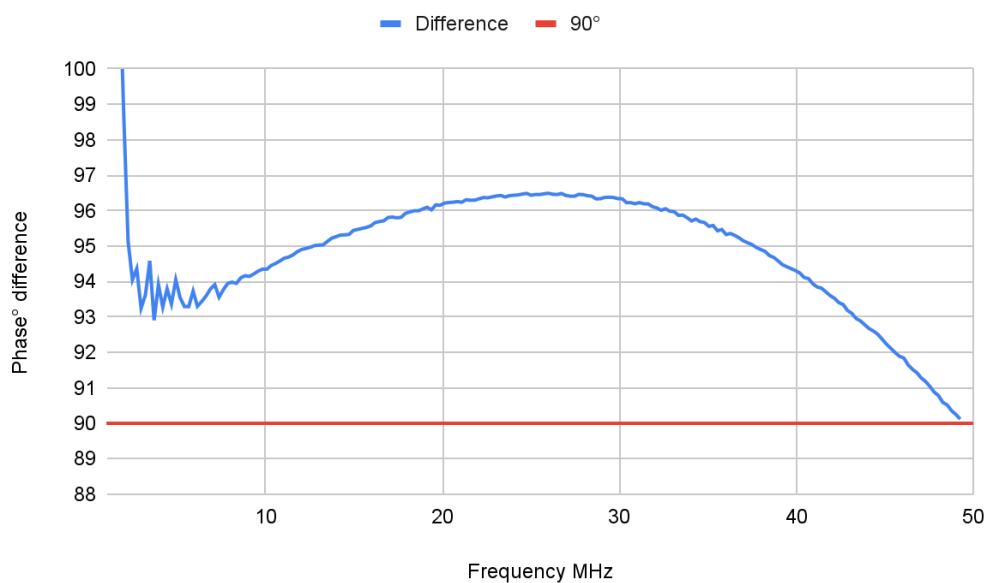
# Appendix A — Zoomed view of the Qucs schematic (LPF + HPF)



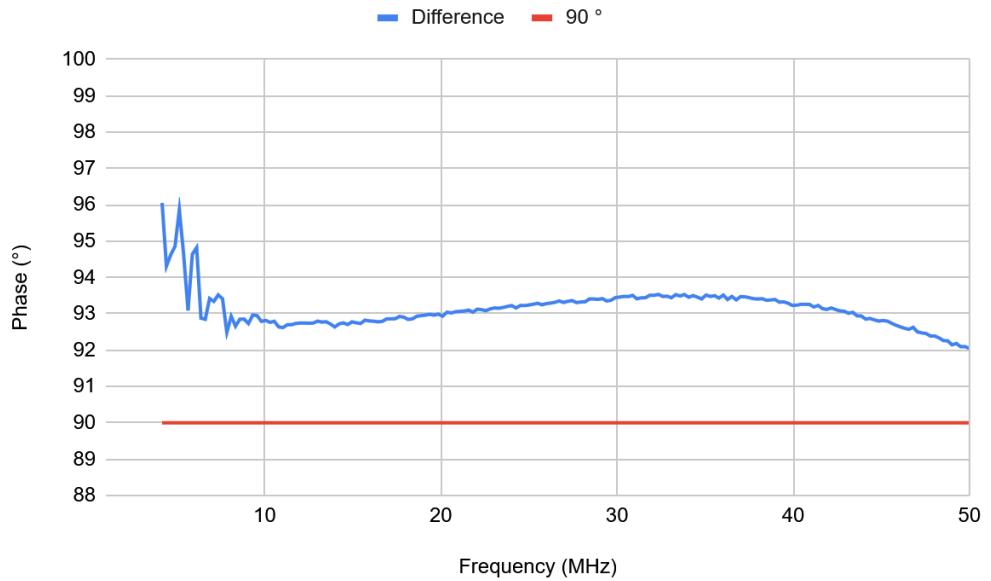
## Appendix B — PCB after Corrosion and final



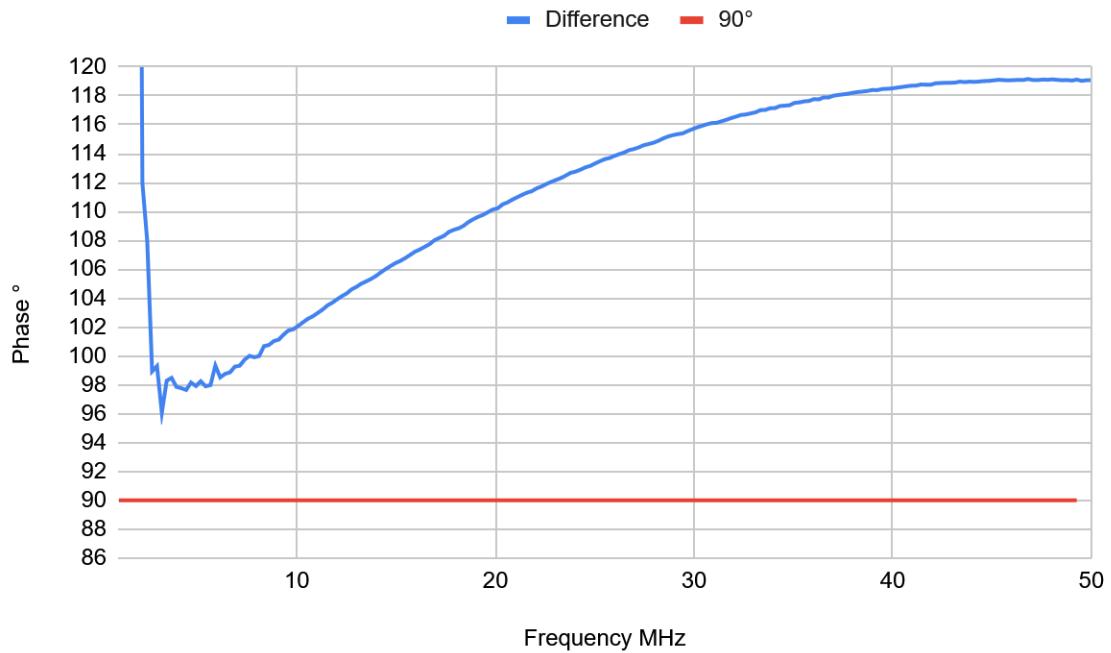
## Appendix C – Comparison of phase shift curves for different inductance values



**Figure 17** — Tow 4 nH and one 16 nJH

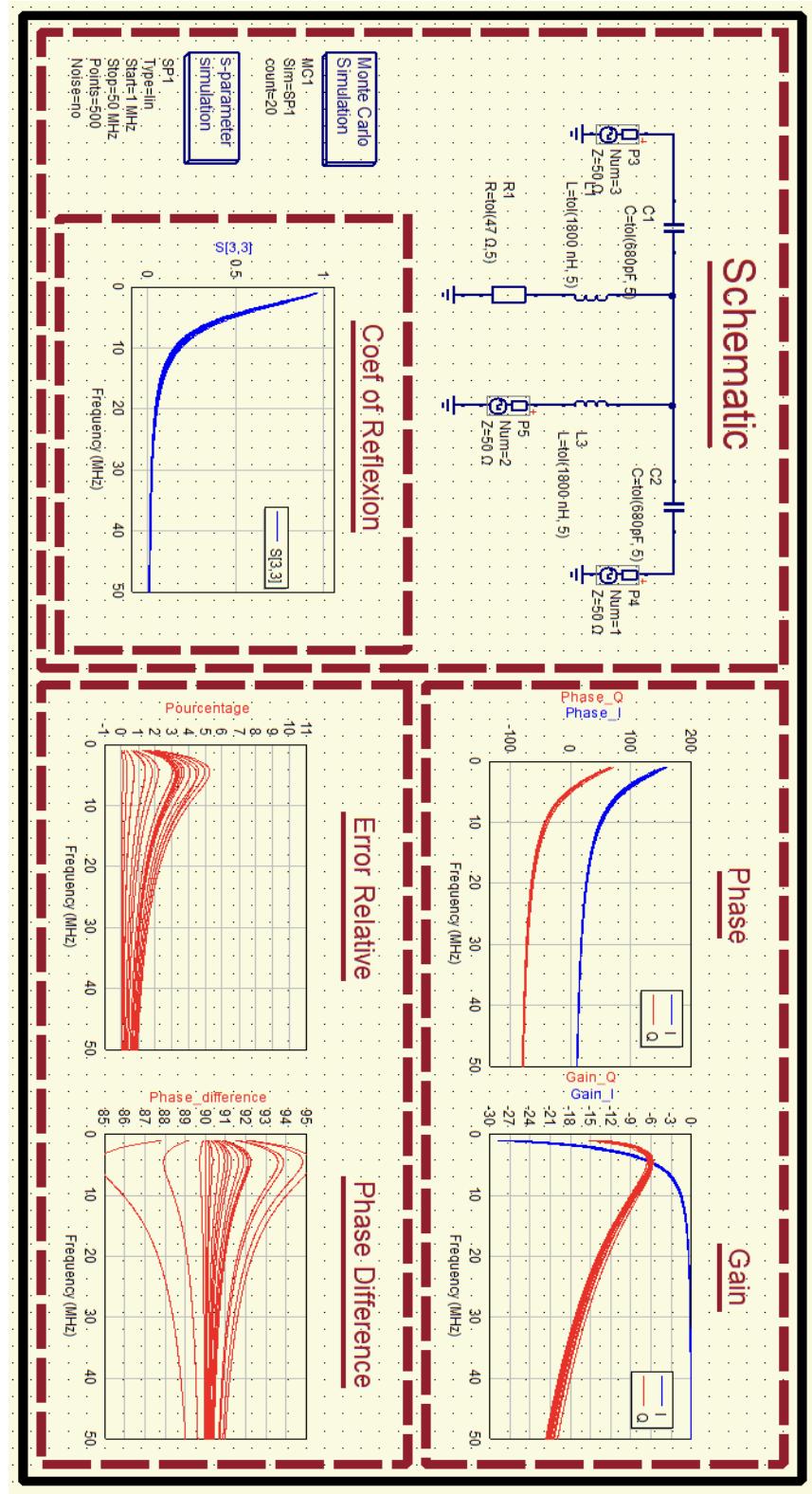


**Figure 18 — Three 4 nH**

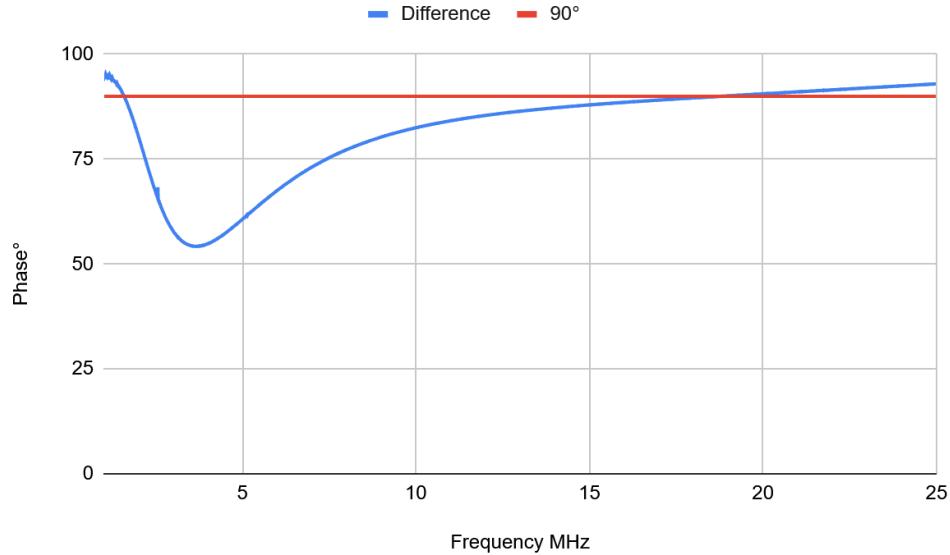


**Figure 19 — First Schematic with 16 nH, 41 nH, 62 nH**

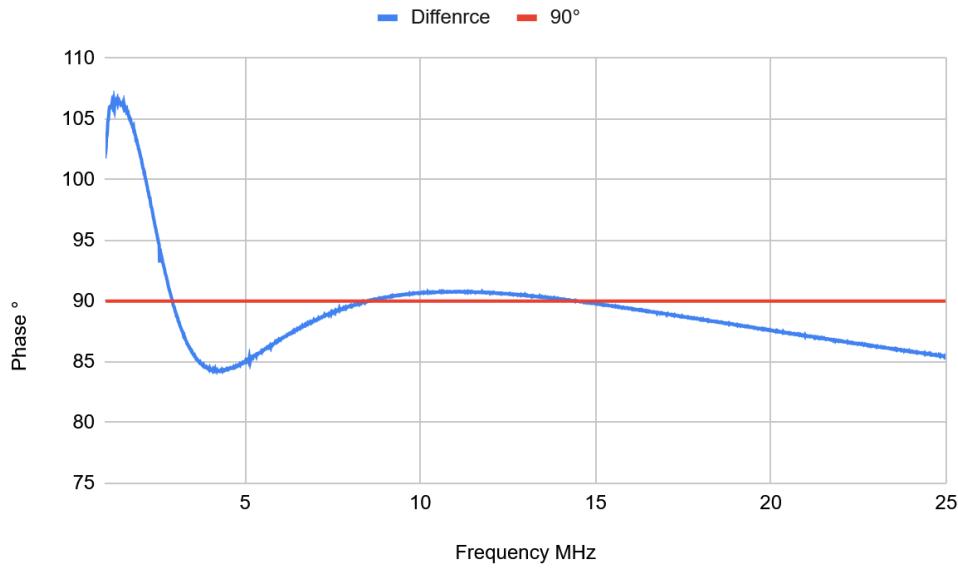
# Appendix D - Zoom Qucs schematic of the second hybrid phase shifter



## Appendix E —Phase Measurements of the Different Test Assemblies

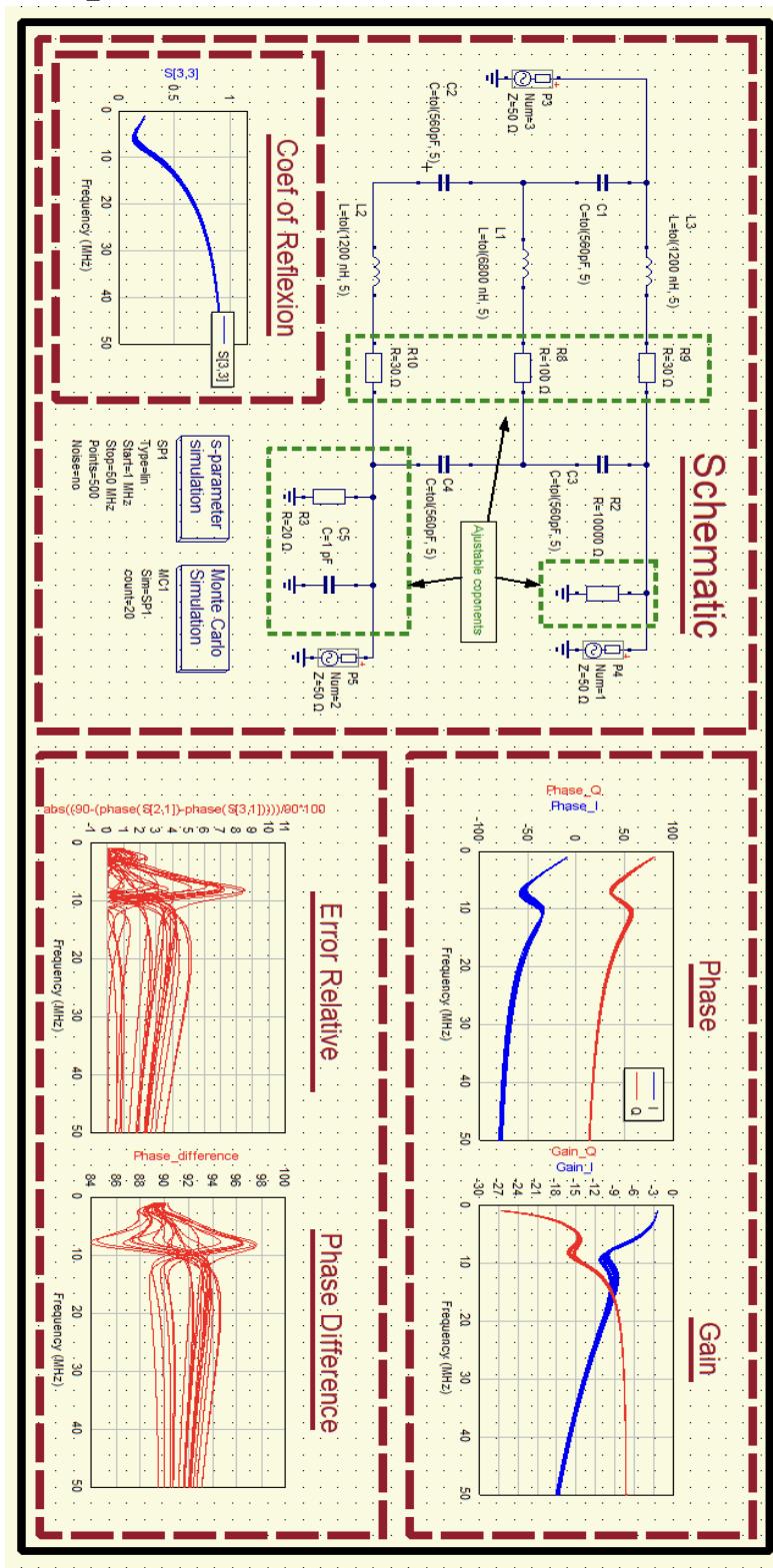


**Figure 20** — With two inductors 850 nH with resistor of  $47 \Omega$

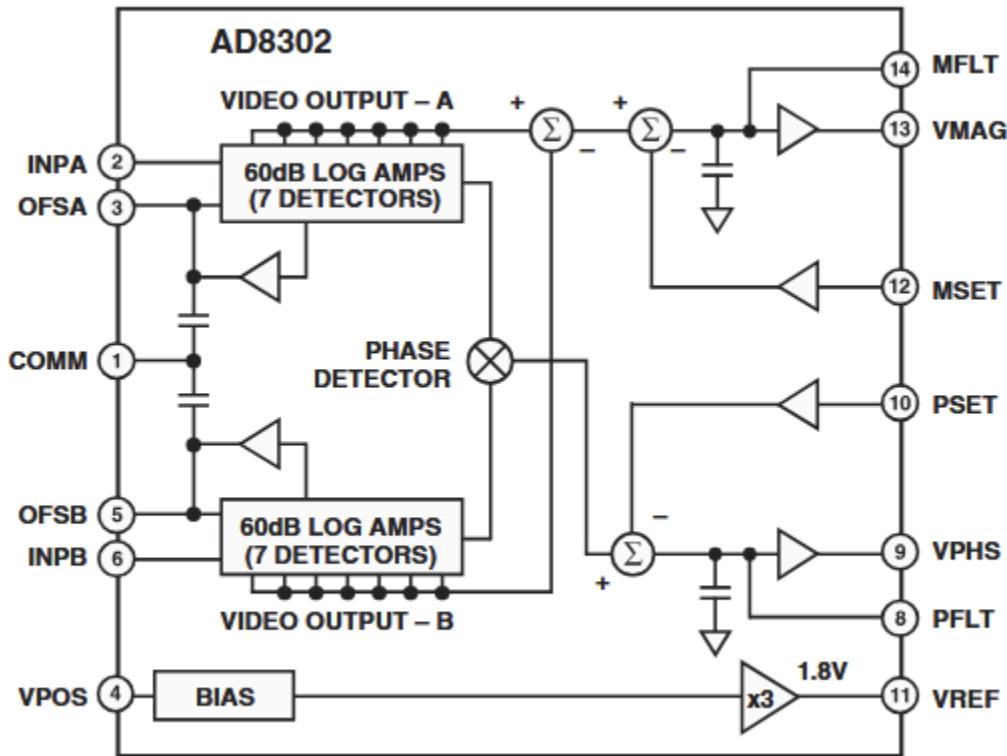


**Figure 21** — With two inductor 1650 nH

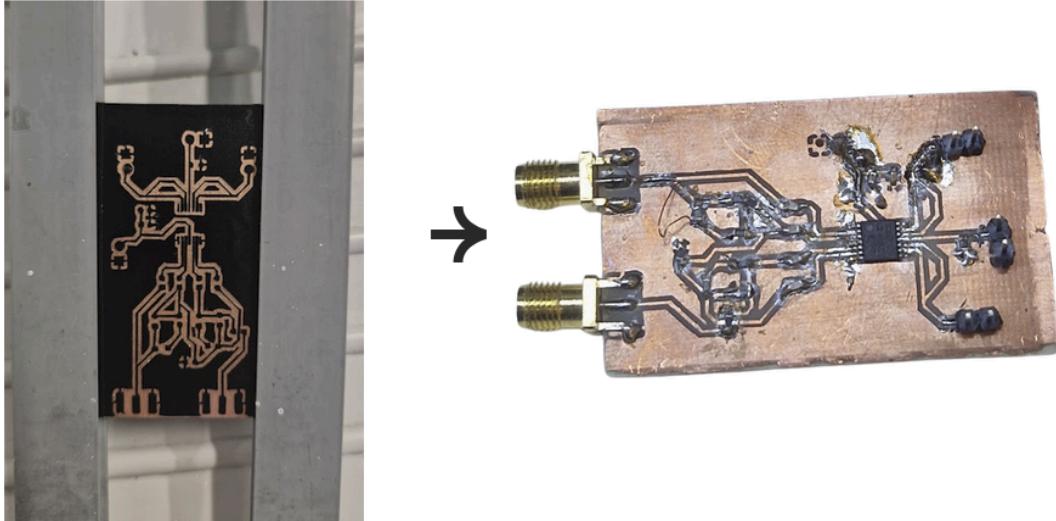
## Appendix F — Zoom Qucs schematic of the adjustable phase shifter



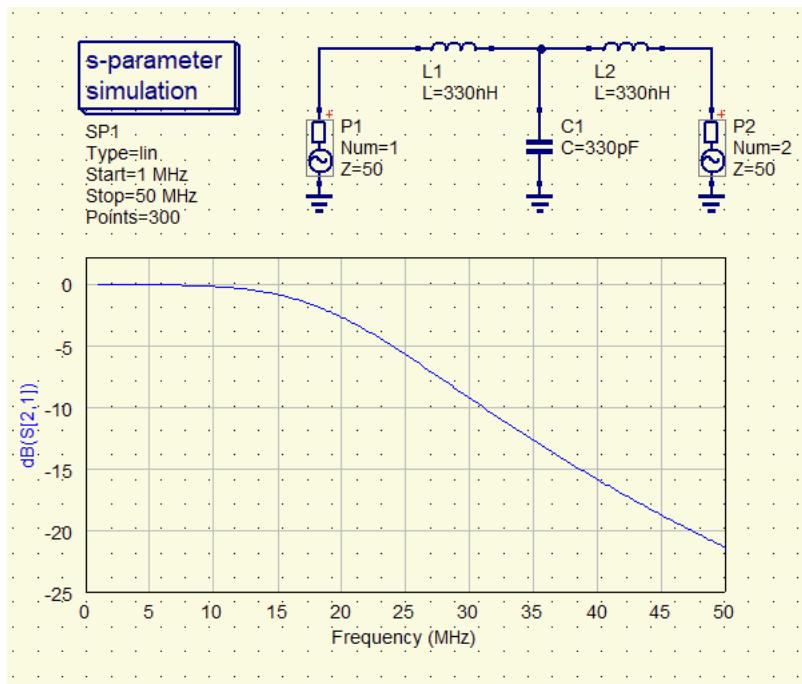
## Appendix G — Block Diagramme AD8302



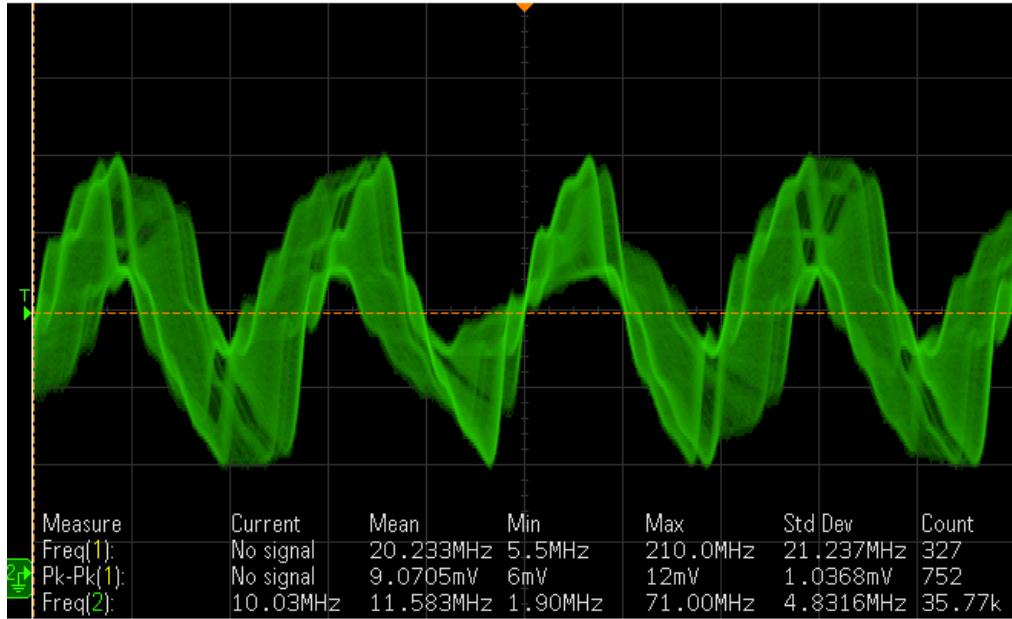
## Appendix H — Phase detector PCB (AD8302) before corrosion and final



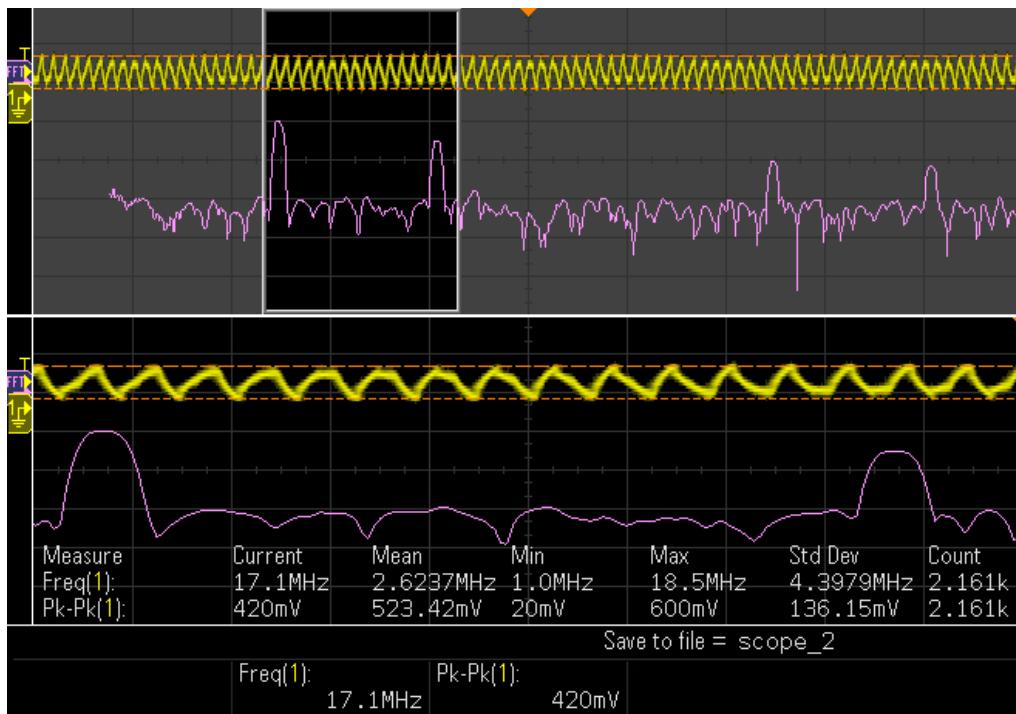
## Appendix I - QucsStudio schematic and simulated FFT response



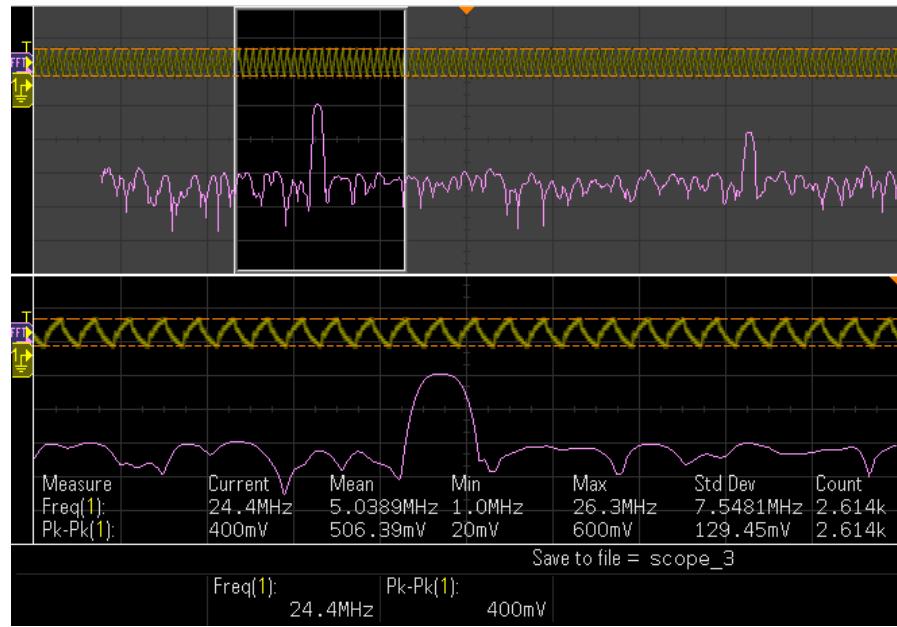
## Appendix J — FFT measurements before and filtering



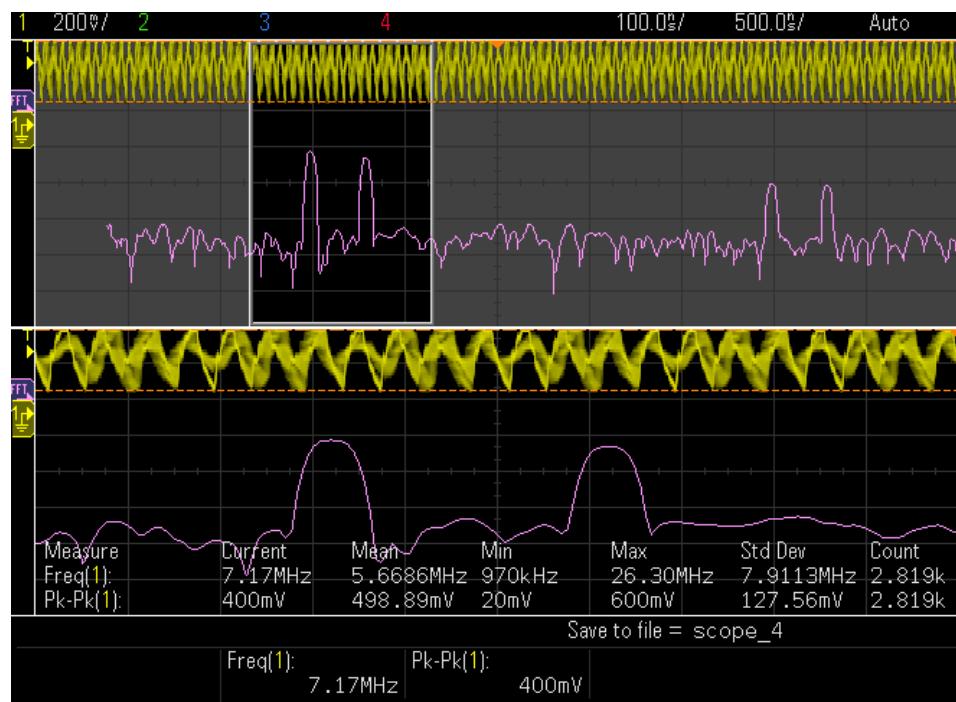
**Figure 22** — 25 MHz without filter



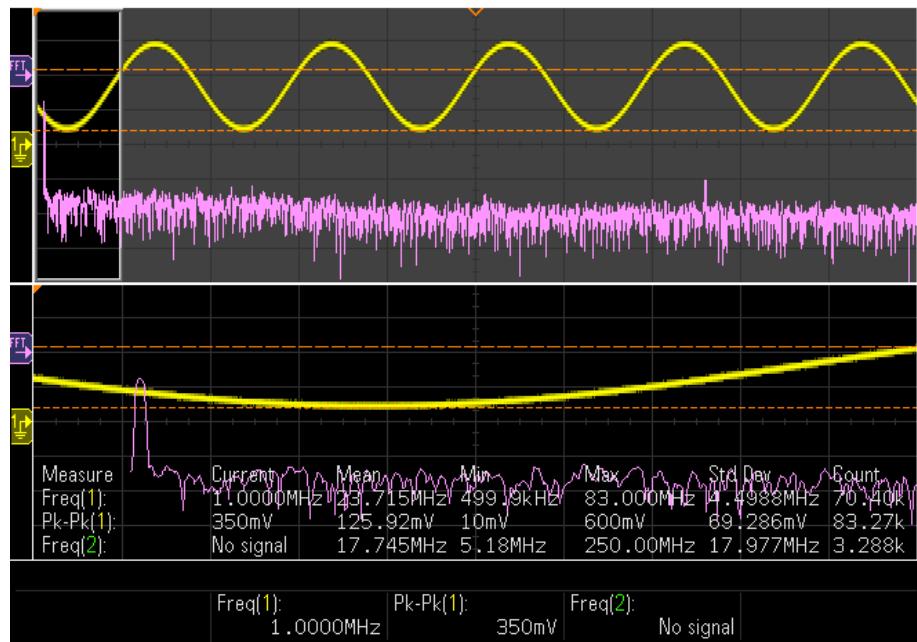
**Figure 23** — 17.1 MHz without filter



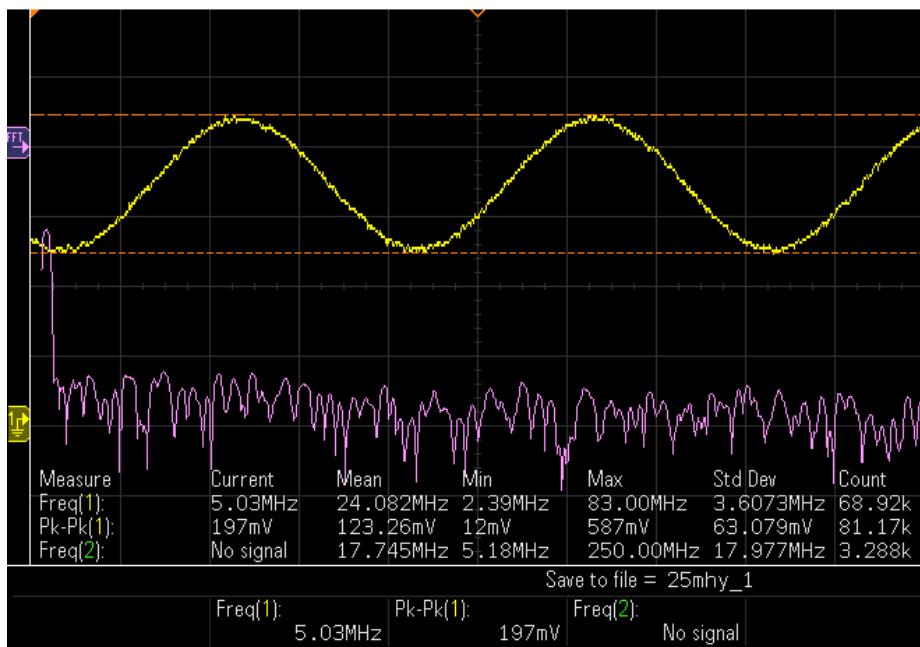
**Figure 24 — 25 MHz without filter**



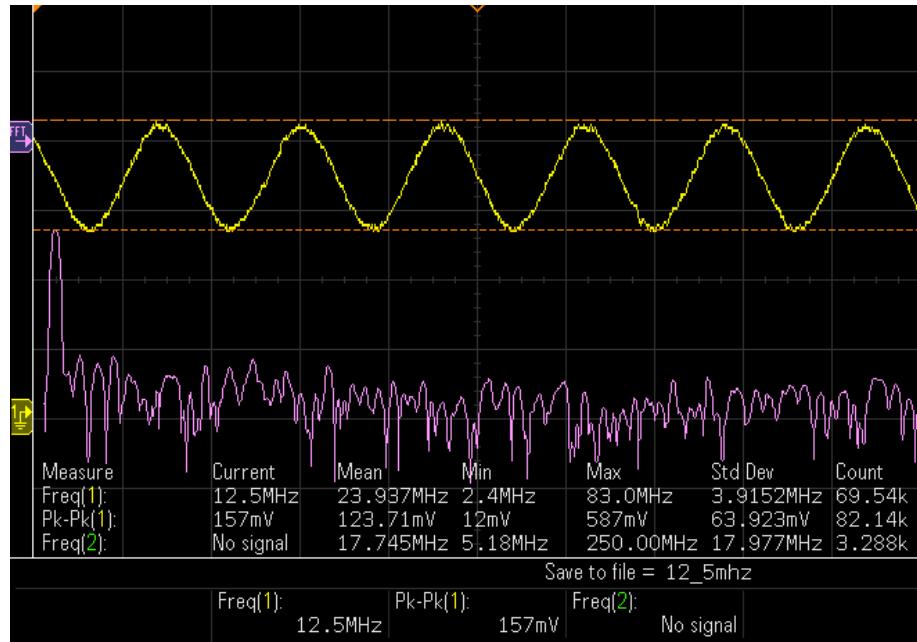
**Figure 25 — 7.17 MHz without filter**



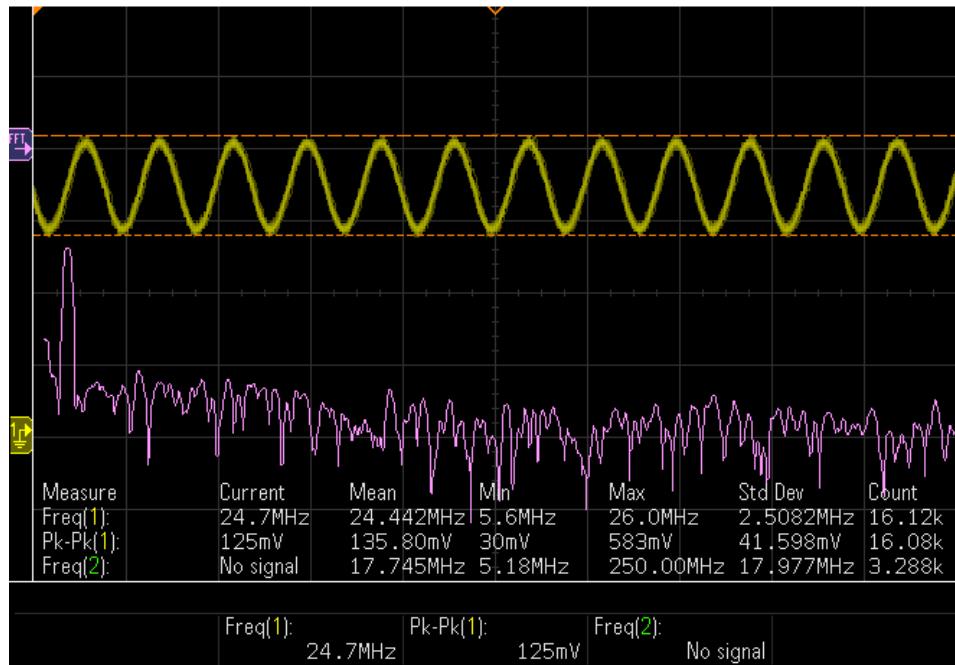
**Figure 26 — 1 MHz with filter**



**Figure 27 — 5 MHz with filter**



**Figure 28 — 12,5 MHz with filter**



**Figure 29 — 25 MHz with filter**

## Appendix K — Arduino final code

```
/**
```

```

* ===== DDS Initialization =====
* Configures SPI communication with the DDS generator (AD5932).
* Defines pins, SPI settings, initializes the DDS sequence,
* and prepares the starting frequency and sweep increments.
*/
#include <SPI.h>

const uint8_t PIN_DDS_CS = 8; // CS pin for DDS
const uint8_t PIN_DDS_CTRL = 7; // Control pin for DDS
SPISettings ddsSPI(4000000, MSBFIRST, SPI_MODE2); // SPI settings for DDS

// DDS configuration sequence (instructions to send)
const uint16_t progDDS[] PROGMEM = {
    0x0FFE, 0xCEB8, 0xD051, 0x2312, 0x3008, 0x13E8
};

const uint32_t FSTART = 1'000'000UL; // Start frequency: 1 MHz
const uint32_t dF = 100'000UL; // Frequency step: 100 kHz
uint16_t stepCount = 0; // Number of steps taken
uint32_t curFreq = FSTART; // Current frequency

const uint8_t LED_RUNNING = 2; // Blinking LED during acquisition
const uint8_t LED_DONE = 4; // Static LED indicating end of sweep

// Sends a DDS command via SPI
void writeDDS(uint16_t w) {

```

```

    SPI.beginTransaction(ddsSPI);

    digitalWrite(PIN_DDS_CS, LOW);

    SPI.transfer16(w);

    digitalWrite(PIN_DDS_CS, HIGH);

    SPI.endTransaction();

}

// Initializes the DDS with the predefined sequence

void initDDS() {

    for (uint8_t i = 0; i < sizeof(progDDS) / 2; ++i)

        writeDDS(pgm_read_word(&progDDS[i]));

    stepCount = 0;

    curFreq = FSTART;

}

// Sends a pulse on DDS control pin and updates the frequency

inline void pulseCTRL() {

    digitalWrite(PIN_DDS_CTRL, HIGH);

    delayMicroseconds(2);

    digitalWrite(PIN_DDS_CTRL, LOW);

    if (stepCount < 240) {

        stepCount++;

        curFreq += dF;

    }

}

```

```

/**
 * ===== ADC Acquisition =====
 * Initializes and communicates with ADC (AD717x).
 * Performs 24-bit reads and reconstructs signed integer values.
 */

#define CS_ADC 10
#define LED_OK LED_BUILTIN
bool useInternalRef = false;
float VREF = useInternalRef ? 2.5 : 5.0; // Reference voltage selection
SPISettings adcSPI(1000000, MSBFIRST, SPI_MODE3);

// Writes a value to an ADC register
void writeRegister(byte reg, byte val) {
    SPI.beginTransaction(adcSPI);
    digitalWrite(CS_ADC, LOW);
    SPI.transfer(0x00 | (reg & 0x3F));
    SPI.transfer(val);
    digitalWrite(CS_ADC, HIGH);
    SPI.endTransaction();
}

// Reads a value from an ADC register
byte readRegister(byte reg) {
    SPI.beginTransaction(adcSPI);

```

```

digitalWrite(CS_ADC, LOW);

SPI.transfer(0x40 | (reg & 0x3F));

byte v = SPI.transfer(0x00);

digitalWrite(CS_ADC, HIGH);

SPI.endTransaction();

return v;

}

```

// Resets ADC by sending 8x 0xFF bytes

```

void resetADC() {

SPI.beginTransaction(adcSPI);

digitalWrite(CS_ADC, LOW);

for (int i = 0; i < 8; i++) SPI.transfer(0xFF);

digitalWrite(CS_ADC, HIGH);

SPI.endTransaction();

}

```

// Reads 24-bit signed data from ADC and formats as signed long

```

long readData() {

uint8_t b[3];

SPI.beginTransaction(adcSPI);

digitalWrite(CS_ADC, LOW);

SPI.transfer(0x44); // Read command

b[0] = SPI.transfer(0);

b[1] = SPI.transfer(0);

```

```

b[2] = SPI.transfer(0);

digitalWrite(CS_ADC, HIGH);

SPI.endTransaction();

long v = ((long)b[0] << 16) | (b[1] << 8) | b[2];

if (v & 0x800000) v |= 0xFF000000; // Sign extension for 24-bit

return v;

}

/***
 * ===== Setup =====
 * Configures pins, initializes SPI communication,
 * initializes DDS and ADC, and sets up the registers.
 */

void setup() {

    // Configure control pins

    pinMode(PIN_DDS_CS, OUTPUT);

    digitalWrite(PIN_DDS_CS, HIGH);

    pinMode(PIN_DDS_CTRL, OUTPUT);

    digitalWrite(PIN_DDS_CTRL, LOW);

    pinMode(CS_ADC, OUTPUT);

    digitalWrite(CS_ADC, HIGH);

    // Status LEDs

    pinMode(LED_OK, OUTPUT);

    pinMode(LED_RUNNING, OUTPUT);
}

```

```

pinMode(LED_DONE, OUTPUT);
digitalWrite(LED_RUNNING, LOW);
digitalWrite(LED_DONE, LOW);

// Initialize serial and SPI
Serial.begin(115200);
SPI.begin();

// Initialize DDS
initDDS();
Serial.println("DDS ready. 'n' = +100kHz + measure (0.5 s), 'r' reset to 1 MHz");

// Initialize ADC with register configuration
resetADC();
delay(20);
writeRegister(0x01, 0x00); // Interface mode
writeRegister(0x02, 0x00); // Read mode
writeRegister(0x10, 0x18); // Input channel
byte setup0 = useInternalRef ? 0xF0 : 0xF2;
writeRegister(0x20, setup0); // Reference
writeRegister(0x28, 0x00); // Gain
digitalWrite(LED_OK, HIGH); // System ready
}

/**
```

```

* ===== Processing & Action =====
* 'n' command: Increments frequency, measures for 500 ms,
* averages values, prints voltage, and controls LEDs.
* 'r' command: Resets DDS frequency to 1 MHz.

*/
void loop() {
    if (Serial.available()) {
        char c = Serial.read();
        if (c == 'n') {
            digitalWrite(LED_DONE, LOW);
            while (stepCount < 240) {
                pulseCTRL(); // Pulse and update frequency
                // Blink LED_RUNNING each measurement
                digitalWrite(LED_RUNNING, !digitalRead(LED_RUNNING));
            }
            const uint32_t tStart = millis();
            double sum = 0;
            uint16_t n = 0;
            // Averaging over 500 ms
            while (millis() - tStart < 500) {
                long raw = readData();
                float voltage = (raw > 0) ? (raw / 2.0) * (VREF / 8388608.0) : 0;
                if (voltage < 1.92) { // Filter outliers

```

```

        sum += voltage;

        n++;

    }

}

// Compute average

float avg = n ? sum / n : 0.0;

// Display frequency and measured voltage

Serial.print("Freq ");

Serial.print(curFreq / 1e6, 3);

Serial.print(" MHz = ");

Serial.println(n ? avg : ---" V");

}

// Sweep done: turn on DONE LED

digitalWrite(LED_RUNNING, LOW);

digitalWrite(LED_DONE, HIGH);

} else if (c == 'r') {

    writeDDS(0x0FF); // Reset DDS

    stepCount = 0;

    curFreq = FSTART;

    digitalWrite(LED_DONE, LOW);

    Serial.println("Reset DDS 1 MHz");

```

```

    }
}

}

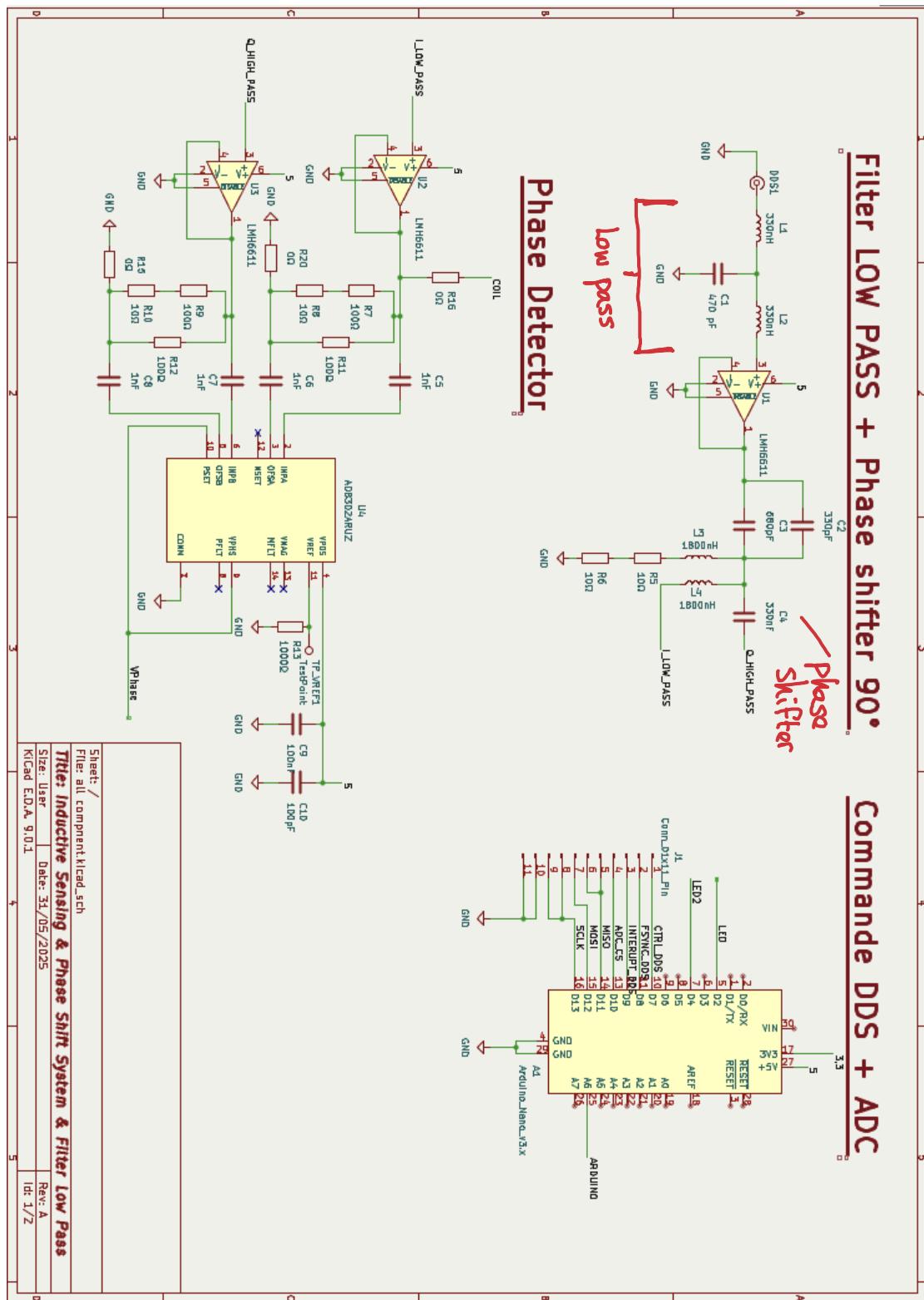
```

## Appendix K — Final Arduino wiring

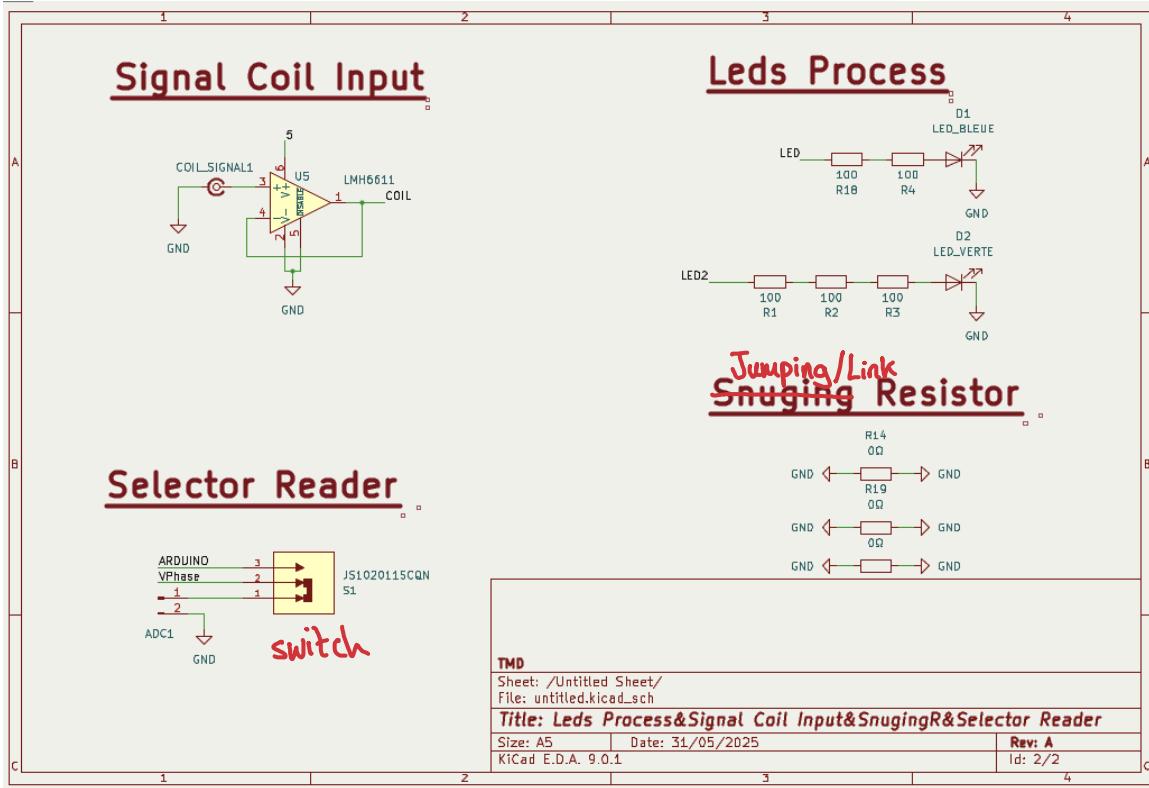
Function	Arduino Pin	Target Module	Module Pin Name
SPI – Data Out (MOSI)	11	DDS (AD5932)	SDATA
SPI – Clock (SCK)	13	DDS / ADC	SCLK
SPI – Data In (MISO)	12	ADC (AD7177-2)	DOUT
Chip Select – DDS	8	DDS (AD5932)	FSYNC
CTRL (Frequency increment)	7	DDS (AD5932)	CTRL
Chip Select – ADC	10	ADC (AD7177-2)	/CS
Status LED (optional)	13 or BUILTIN	ADC (debug purpose)	-
Power Supply	5V / GND	DDS / ADC	VCC / GND



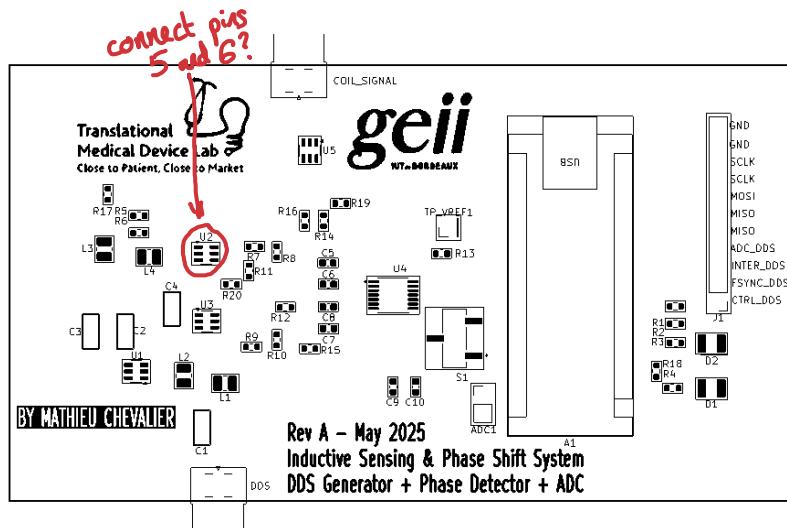
# Appendix R - Electrical schematic Part 1



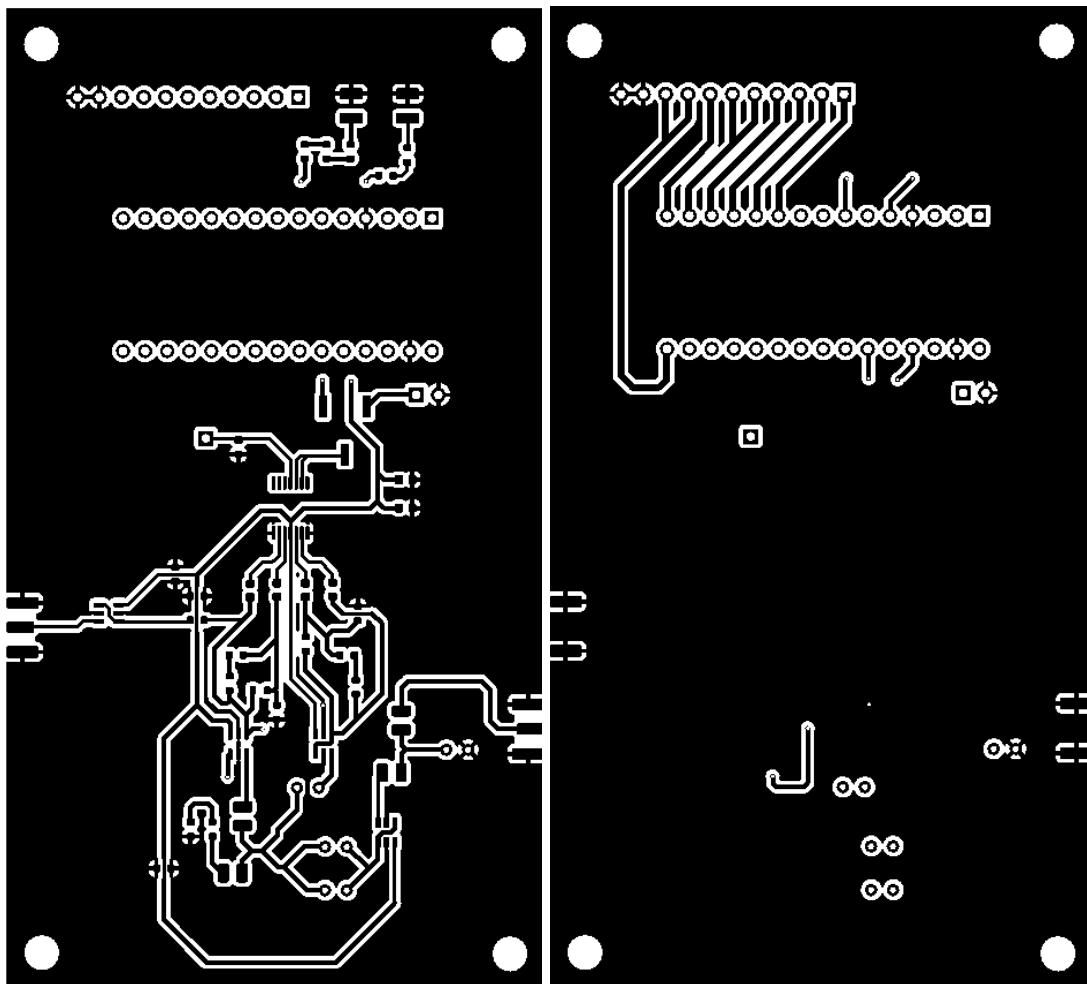
## Appendix S – Electrical schematic Part 2



## Appendix N1 — Component Implantation



## Appendix N2 — Left Top copper & Right Bottom copper



## Appendix N3 — 3D View of PCB

