Wireless Read-Out Circuit for an Inductively Coupled Pressure Sensor.

Daniel Scully 21431856

Daire Shankey 21443174

Electrical and Electronic Engineering

School of Engineering

University of Galway

Supervisors

Dr Adnan Elahi, Dr Riaz Rehman

Assessors

Dr Adnan Elahi, Prof John Breslin

In partial fulfilment of the requirements for the degree of

Electronic and Electrical/Computer Engineering

May 10, 2024

Declaration

I hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification, and that it is the result of my own independent work.

Daniel Scully, Daire Shankey

10/05/2024

Abstract

Advancements in medical technology continuously seek to improve patient care through innovative solutions. This project focuses on enhancing pressure management in medical treatments, such as those used for burn victims and patients requiring pressure garments. Existing methods, which often involve manual and imprecise pressure adjustments, fall short in providing optimal therapeutic outcomes, leading to potential complications like tissue damage or suboptimal healing.

Recent research has proposed wireless read-out circuits for inductively coupled pressure sensors as a promising alternative. These devices offer the advantage of real-time, accurate pressure monitoring without direct contact with the patient's body, which is not fully explored in current medical applications. However, there remains a significant knowledge gap in the integration and practical deployment of these technologies, particularly in achieving reliable and continuous pressure monitoring under varying clinical conditions.

This project aims to address these challenges by developing a non-invasive, accurate, and wireless system to measure pressure beneath pressure garments. The design incorporates a waveform generator (DDS) board for signal management, signal filtering to reduce noise, the Arduino platform for control, a phase detector for accurate phase measurement, and custom-designed inductive coils to ensure precise data acquisition. This system has been validated through rigorous testing protocols to ensure reliability across different operational conditions.

The results indicate successful functionality of the developed circuitry, demonstrating its capability to operate as intended. While direct pressure measurements were not obtained, the performance of the system underscores its potential to significantly enhance patient monitoring and treatment outcomes. The implications of these findings suggest a substantial shift towards more responsive and adaptive medical care practices, particularly in settings involving wound management and post-operative recovery.

Recognizing the limitations of the current prototype, including its dependency on specific hardware configurations and potential scalability issues, future work will focus on improving the design of its inductive coils as well as signal amplification to increase accuracy. This project underscores the significant potential of integrating advanced sensor technologies into medical practice, paving the way for more effective patient care and management.

Table of contents

List of figures

Abbreviation Table	
Abbreviation	**Full Name**
IC	Integrated Circuit
DDS	Direct Digital Synthesiser
LP	Low Pass filter
HP	High Pass Filter
RC	Resistor-Capacitor Filter
OP-AMP	Operational Amplifier
Raw Value	Arduino Analogue to digital Value
Known Phase	Controlled Phase Output Manually set
CSV	Comma Separated Values
RF	Radio Frequency
GUI	Graphical User Interface
SPI	Serial Peripheral Interface

Chapter 1: Introduction

1.1 Clinical Need

The Wireless Readout Circuit for an Inductively Coupled Pressure Sensor aims to help treat burn victims by monitoring pressure in their bandages or garments. Pressure is important for helping wounds heal properly. If there is too little pressure on the bandage/garment, the desired healing effect isn’t achieved whereas if there is too much pressure applied to the bandage/garment then there is a risk of cutting off the circulation and causing tissue damage. The device checks the pressure in real-time to make sure it's correct, making the recovery process smoother and more effective.

Current treatments are manually monitored pressure bandages/garments. While these treatments are somewhat effective, they don’t allow for precise, real-time pressure monitoring. If a change in bandage pressure occurs somewhere between manual pressure monitoring sessions, it won’t be noticed until that next session. The manual monitoring also makes it difficult to make exact adjustments to the pressure applied.

There are numerous benefits to developing the Wireless Readout Circuit for an Inductively Coupled Pressure Sensor. This device aims to optimise the healing of wounds which in turn, aims to reduce the risk of complications such as infections. Pressure monitoring in this case also has the opportunity to prevent complications in the healing process such as cutting off circulation to the wound area.

1.2 Proposed Solution

There are two different treatment technologies that address the clinical need, Smart Bandages, and Wireless Readout Circuits for Inductively Coupled Pressure Sensors.

Smart bandages allow for external pressure sensing of wounds. Smart bandages have the ability to monitor various different parameters such as moisture, pressure and temperature, sensors are integrated into the bandages. They provide real-time data of the wound environment. Due to the fact smart bandages are externally applied, they lack the ability to measure internal measurements. [1]

Wireless Readout Circuits for Inductively Coupled Pressure Sensors allow for non-invasive internal pressure sensing, providing accurate data such as pressure. These circuits facilitate real-time measurement from within the body, offering continuous tracking and precise measurement capabilities. [2]

In this case, Wireless Readout Circuits for Inductively Coupled Pressure Sensors are deemed the best-suited solution for the clinical need. They provide continuous tracking and are ideal for measuring pressure under bandages. The real-time tracking facilitated by these circuits ensures the maintenance of correct pressure levels, crucial for optimal healing.

Study 1: Focuses on inductively reading a passive resonance sensor in pressure measurement, particularly in pressure garment applications. They used a dual-layer pressure sensor and a portable measurement unit to get accurate readings. While demonstrating good outcomes in tests, they identified limitations in the pressure range and proposed enhancing the system for more effective monitoring of pressure garments. [2]

The study indicates that the sensor system for pressure garments has some limitations in the range it can measure. Improving this range to work effectively under different body conditions might be something to focus on.

1.3 Aims and Objectives

The overall aim of the project is to develop a non-invasive, accurate, and wireless system for measuring pressure under pressure garments.

The specific objectives of the project are:

1. To implement signal generation circuitry that generates an interrogation signal at the resonant frequency of the wireless sensor.
2. To design and implement a phase-shifting circuit that introduces a phase shift in the interrogation signal generated by the signal generator.
3. To implement a phase detector circuit that measures the phase difference in the interrogation signal due to the presence of a wireless sensor.
4. To design and build an inductive coil for the wireless reader system at the resonant frequency of the wireless sensor.
5. To validate the accuracy of the integrated wireless reader system and analyse its response.

1.4 Societal Impact

The potential impact of the proposed medical device on patient outcomes and quality of life is valuable. The optimised healing process aims to reduce the risk of complications during healing which in turn should improve the quality of life for the patients during the healing process.

From an economic perspective, the optimised healing process aims to reduce complications during the healing process which in turn reduces the need for medical interventions as a result of the complications which in turn results in shorter hospital stays which would reduce medical expenses for the patient. From a societal perspective, by preventing potential complications during the healing process, the device could potentially reduce the burden on the healthcare systems this in turn would allow healthcare professionals to focus on more critical patients.

There are a few potential ethical and regulatory issues associated with the device. On an ethical note, maintaining patient privacy and patient data security is an ethical concern. On a regulatory note, compliance with the appropriate standards would be of vital importance to ensure that the device meets the appropriate safety and performance standards before the devices reaches clinical use.

1.5 Conclusion

Chapter 1 explained why it's important to monitor pressure in treating burn wounds, leaning towards using Wireless Readout Circuits for Inductively Coupled Pressure Sensors found in Study 1.

Chapter 2: Background and Literature Review

2.1 Introduction

This chapter looks at what other researchers have found about our proposed solution.

2.2 Inductively Coupled Resonance Sensors in Pressure Garment Applications

Inductively coupled resonance sensors are sensors that utilise inductive coupling and resonance to detect changes in the surroundings. In this case they are used to detect changes in pressure garment applications. Inductively coupled resonance sensors allow for real-time pressure sensing. In the case of garment applications, they could check to see if the correct pressure required for optimal healing is achieved. [2]

2.2.1 Principles of Inductive Coupling

Inductive coupling occurs when the magnetic field generated by one electrical current influence and activates a reaction in another entity. For example, as electricity courses through one wire, it can induce an electrical current in a neighbouring wire, creating a connection through the magnetic field's pervasive influence. Inductive coupling can create resonance, which, in turn, generates desired electrical currents, used in radio transmission, reception, and non-contact charging devices [3]. This application of inductive coupling is what is used in the case of an inductively coupled resonance sensor.

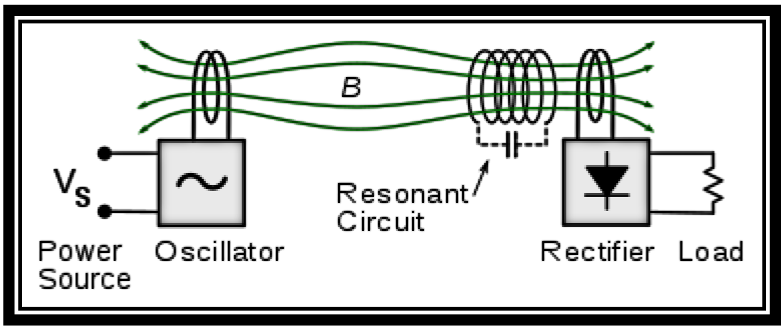


Figure 1 Resonant Inductive Coupling Source [4]

Resonant inductive coupling is a method that wirelessly transfers power between two coils. When electricity flows through one coil, it creates a magnetic field. If another coil, tuned to the same frequency, is nearby, it can receive this energy without needing physical connections [5].

2.2.2 Optimised Healing Process

In garment applications, the integration of Inductively Coupled Resonance Sensors has the potential to optimise the healing of wounds. By providing real-time pressure sensing, healthcare professionals can continuously monitor the pressure applied to the wound and can make any adjustments required to achieve optimal healing.

2.2.3 Real-Time Pressure Sensing

An advantage of inductively coupled resonance sensors is their ability to provide real-time pressure sensing. The advantage is that the pressure can be accurately and remotely monitored in real time, eliminating the need for manual observation and adjustment.

2.3 Phase Shifters

A phase shifter circuit is a type of circuit that is used to shift the phase of a signal without changing its amplitude. Phase shift is commonly used in numerous electronic applications such as oscillators, amplifiers, and modulation. [7]

2.3.1 Types of Phase Shift Circuits

There are two main types of phase shift circuits, active and passive. Active phase shifters use active components such as operational amplifiers and because they use amplifiers, they can also provide signal amplification aswell as a phase shift. Active phase shifters allow for variable shifts in comparison to the fixed shifts of passive phase shift circuits. Passive phase shifters use passive components such as resistors and capacitors. Passive phase shift circuits do not provide amplification and have fixed phase shifts. [6]

2.3.2 Applications

Phase shifting is crucial in RF applications, especially with Phase Shifters commonly used in signal cancellers, equalizers, and beam steering. These devices help direct antenna energy in desired directions, often used in radar and non-line-of-sight operations. Additionally, in dynamic gain equalizers, phase shifters adjust attenuation profiles to match desired responses, compensating for non-flat gain across communication bands. [8]

2.4 Inductor Coils

Inductor coils are passive components that stores energy in a magnetic field when current goes through it. Inductor coils are sometimes wound around a magnetic core like ferrite, when inductor coils are wound around a magnetic core, the inductance value increases. [9]

2.4.1 Types of Inductor Coils

There are lots of different types of inductor coils. Air gap inductor coils are coils that have no magnetic core. The air gap allows control of the saturation level. Planar coils which are printed on PCBs are compact coils. Multilayer coils are made of layers of magnetic core material with coils. All types of inductors have different characteristics and applications. [9] [10]

2.4.2 Design Considerations

When designing coils, there are several factors to take into account such as the size of the coils, the inductance value required for the application, what type of coil to use for the application, and what core material to use.

2.4.3 Applications

Inductor coils have many uses; they can be used in various applications such as power supplies, RF circuits for signal filtering and tuning, ignition coils in internal combustion engines, spark gap radio transmitters, and medical equipment like X-ray machines. [11]

Chapter 3: Design of Wireless Read-Out System

3.1 Introduction

This chapter examines the design of the circuitry specified in the research paper X by Y. The different parts of the circuit are examined individually and the reasons for selecting suitable components to be used is discussed.

3.2 Overall Schematic

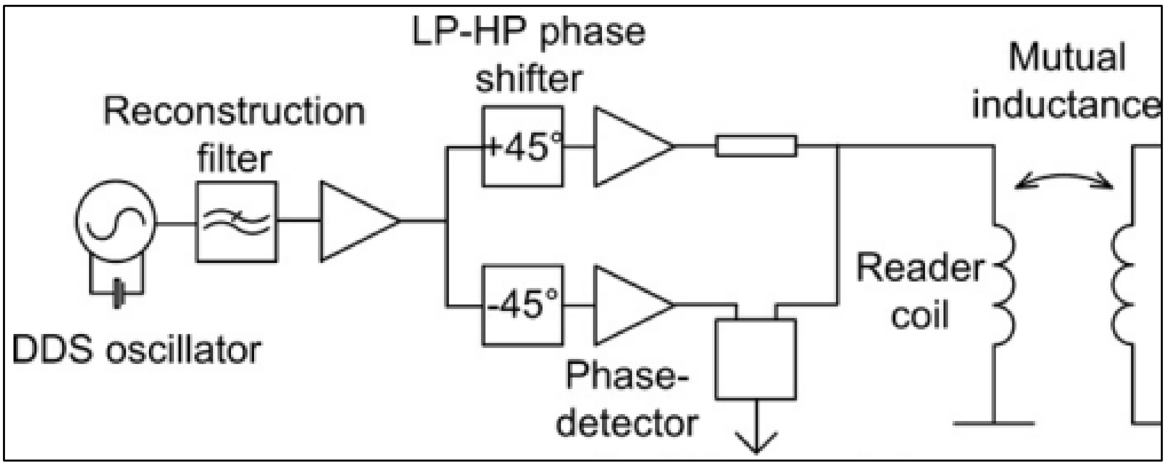


Figure 2 - RF Circuit from paper [2]

Using this circuit diagram as a starting point, the team began to investigate several different electronic components that could potentially be used. The team investigated the following:

1) Direct Digital Synthesiser
2) Reconstruction filter
3) Unity Gain Amplifiers
4) Low pass-High pass Phase Shifter
5) Phase Detector
6) Reader and Sensor Coils
7) Microcontroller

3.3 Direct Digital Synthesiser

The purpose of the DDS in the circuit is to generate a sine wave signal which is ultimately used to interrogate the reader coil. Due to this purpose, there are several factors that had to be considered when selecting a suitable signal generation board for the circuit.

- **Signal Type** – The board must be able to generate a sine signal.
- **Frequency** – The board must operate at a suitable frequency for the rest of the circuit.
- **Signal Quality -** The board must output a relatively smooth waveform as the phase data has to be extracted for analysis.
- **Controllability –** Ideally the board will be controllable from the micro controller and have flexible configurations for different testing setups.
- **Availability** – Is the board available and within the project budget?

3.3.1 Analog devices EVAL-AD5932EBZ

Figure 3 - Photo of EVAL-AD5932EBZ

The first board the team investigated was the Eval-AD5932EBZ. This board is made by Analog Devices and includes the AD5932 waveform generator IC.

The board can output a triangle or sine waveform which would suit the requirements laid out previously. It has a wide frequency range operating from low frequencies up to 40MHz. This frequency range would also be suitable for the circuit. The board comes with evaluation software programmed in LabView and available in a GUI. This is useful for initial testing for the board but for the end circuit the board should be controllable by the microcontroller written in C/C++. [12] This could be overcome by looking for open-source libraries on GitHub or else by coding a new library for the team's purpose.

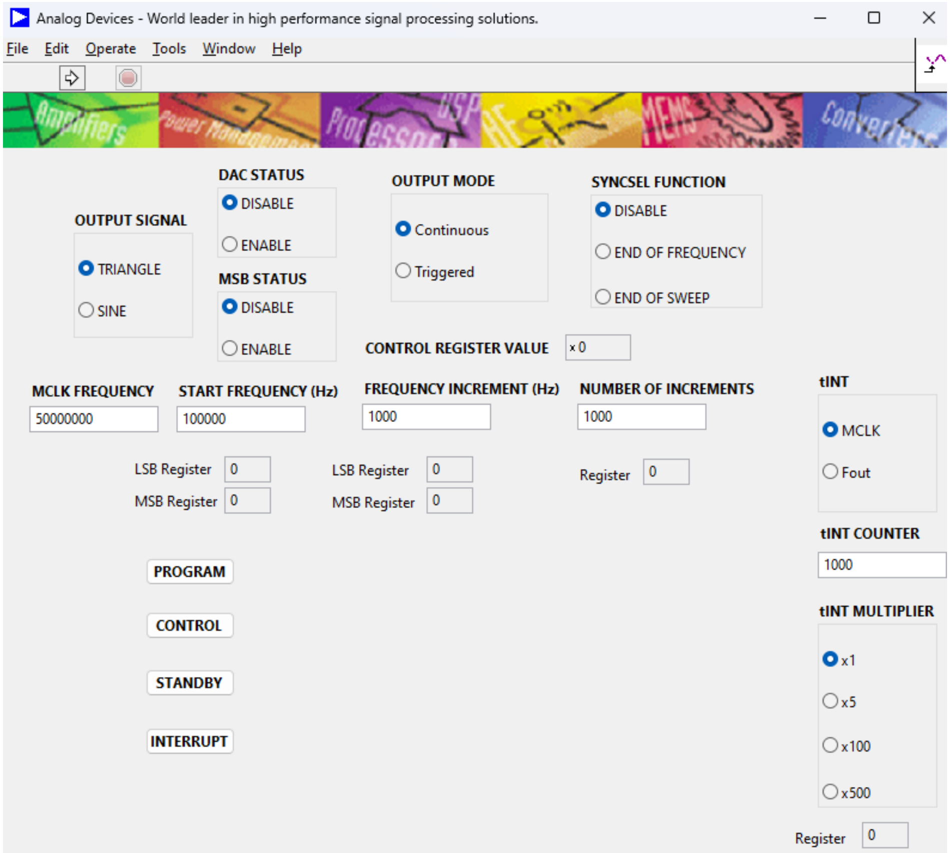


Figure 4 - LabView GUI for AD5932

After downloading and installing the LabView GUI for the AD5932, the team conducted a simple test to check the boards output. The team programmed the board to output several different frequencies within the boards range, however when connected to an oscilloscope there was no output. After checking SPI communication and all other connections, the team concluded that the board's integrated circuit was damaged during a previous project.

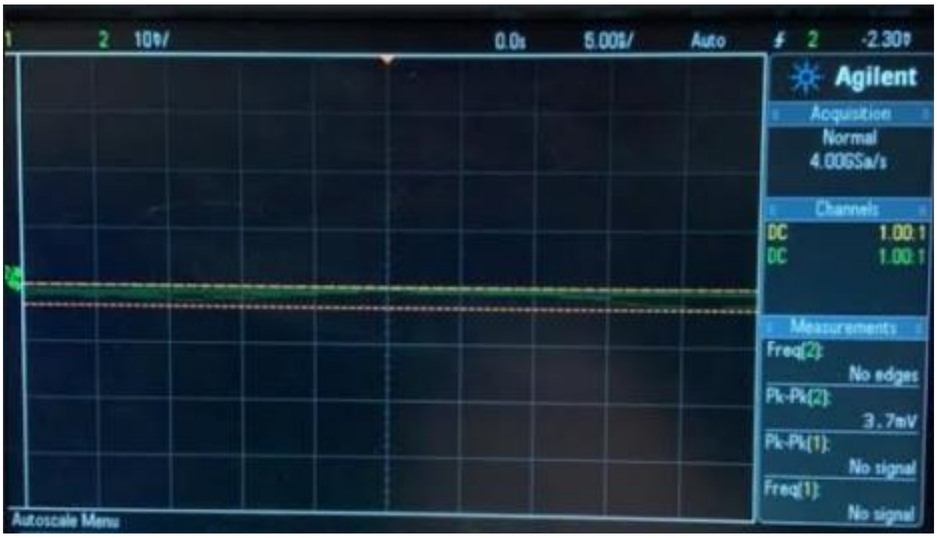


Figure 5 - Output of AD5932 on DSO-X 30121A scope

To conclude, the research on this board, initially it looked like a suitable option due to its availability, configurability, frequency range and signal type. However, after testing the board, it became clear it was unusable.

3.3.2 MIKROE Clock Gen Click

Figure 6- MIKROE Clock Gen Click [13]

After discounting the use of the AD5932, the use of the Clock Gen Click made by MIKROE was then investigated. This board is used for generating clock signals (square waveforms). While it hosts a wide range of frequencies, up to 200MHz [13], it is unsuitable as it can only produce a square output and not a sine waveform. The board is cheap and during the team's investigation of this board they came across a similar product by MIKROE, the Waveform Click.

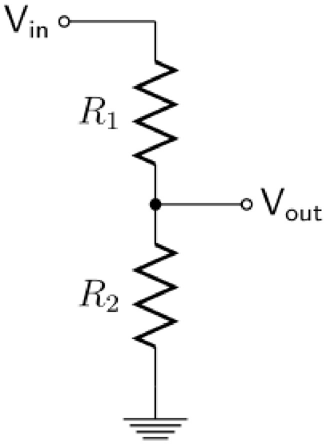
3.3.3 MIKROE Waveform Click

After researching the MIKROE range of signal generators the team came across the Waveform Click.

Figure 7 - MIKROE Waveform Click [14]

The waveform click is a precise sine/triangle/square waveform generator, capable of producing frequencies up to 12MHz. It utilizes a Direct Digital Synthesis (DDS) IC made by Analog Devices. It contains three different integrated circuits, the AD9833, AD5227 and the ADA4891.[14]

The AD9833 [16] is the DDS IC and can produce a clean and stable sine wave up to 5MHz. The board operates at 3.3-volt logic level, requiring a secondary microcontroller but still controllable from one laptop. The AD5227 is a 64-position digital potentiometer used for amplitude control of the waveform [18]. As this board seemed to be very suitable from initial research, the team requested access to it from the supervisor. After receiving the board, a simple test was setup using a voltage divider to interface it with the 5-volt Arduino UNO.



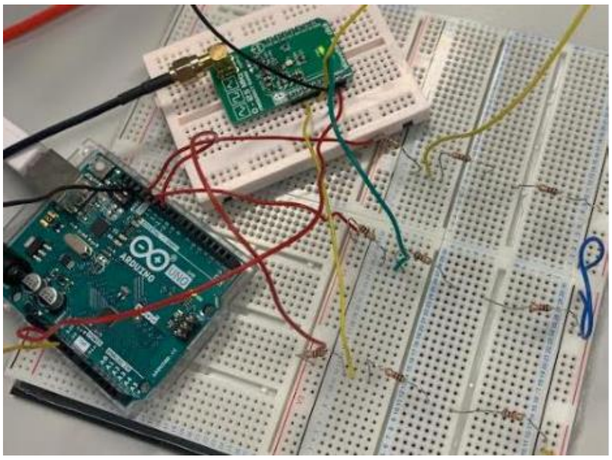


Figure 8- Voltage Divider Circuit [15] and Arduino UNO

The team created voltage divider circuits for each SPI pin to allow the waveform click to be controlled by the Arduino UNO.

The initial test showed some level of functionality available from the board which the team could then improve on from using filtering, correct logic level controller and by writing a new code library.

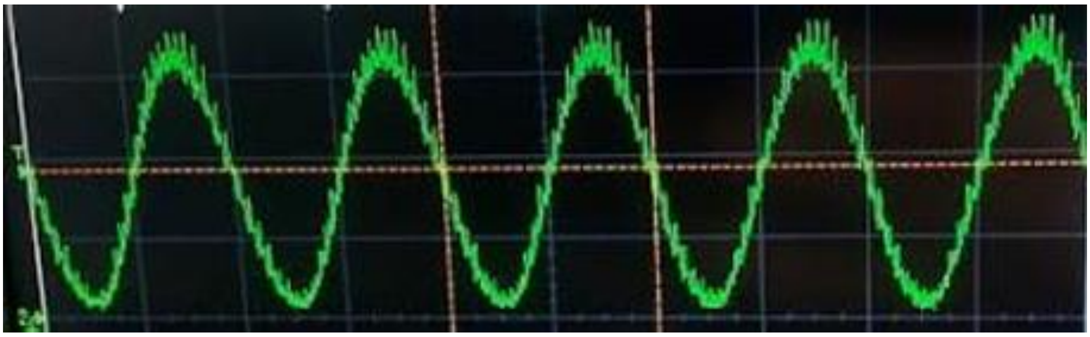


Figure 9 - Initial Output of DDS @1KHz observed on Ossiliscope

3.3.4 Conclusion on DDS Boards

Due to the design factors previously identified and upon reviewing each board it was decided to proceed with using the Waveform Click.

3.4 Reconstruction Filter

A reconstruction filter is commonly applied after a DDS generates a digital waveform, this waveform typically consists of a series of discrete steps, due to the digital-to-analogue conversion process. These steps introduce unwanted spectral components, primarily harmonics of the fundamental frequency of the waveform.

The primary purpose of using a reconstruction filter after a DDS is to smooth out these steps and remove the high-frequency components that are artifacts of the digital generation process. This filter is usually a low-pass filter that allows the desired frequency to pass while attenuating the higher-frequency harmonics. This is crucial for applications requiring clean, precise signals, such as for measuring the phase response in this system.

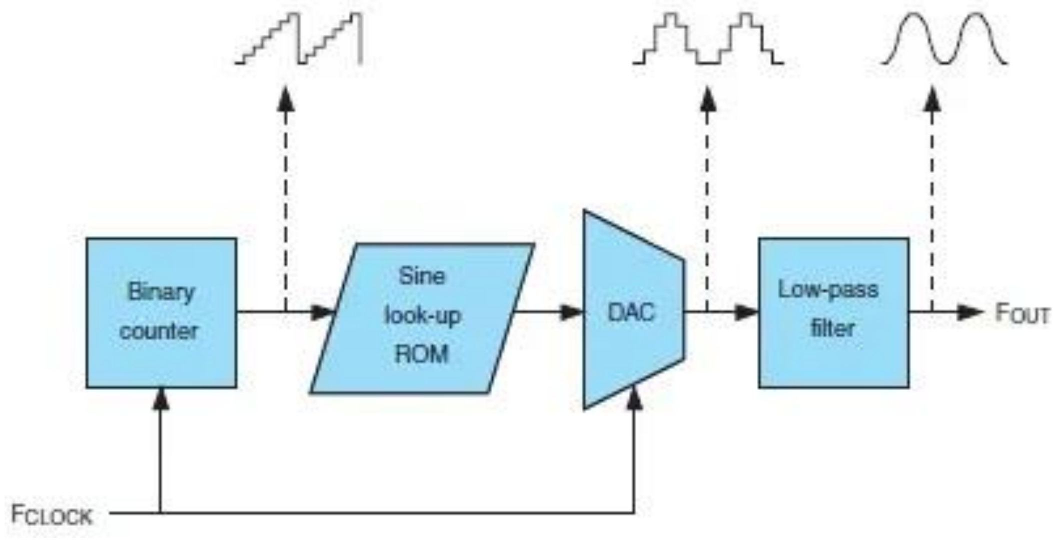


Figure 10 - Low Pass Filter after DAC Output [17]

3.4.1 Resistor Capacitor Filter

A Resistor-Capacitor (RC) filter is a simple and effective type of low-pass filter commonly used in electronic circuits. This filter configuration is particularly useful in applications requiring a basic method to reduce high-frequency components from a signal, such as in the smoothing process following a Direct Digital Synthesis (DDS) in digital-to-analogue conversion systems.

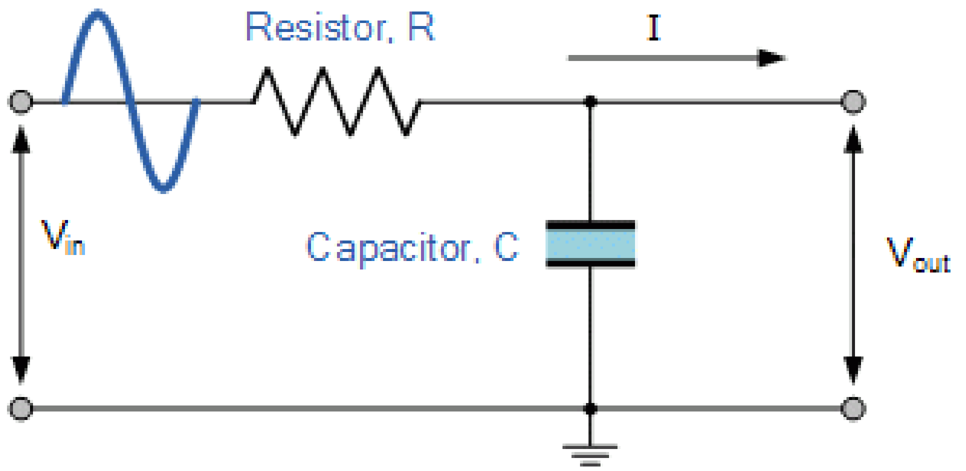


Figure 11- Resistor Capacitor Filter [19]

A resistor Capacitor filter is relatively simple to build and test, once a cut off frequency has been decided and a value for either a resistor or a capacitor has been decided, then the remaining component's value can be determined.

$$f_{cutoff} = \frac{1}{2\pi RC}$$

Figure 12- Formula for RC Filter

The following graph illustrates the expected behaviour of a low pass filter after performing calculations and selecting component values.

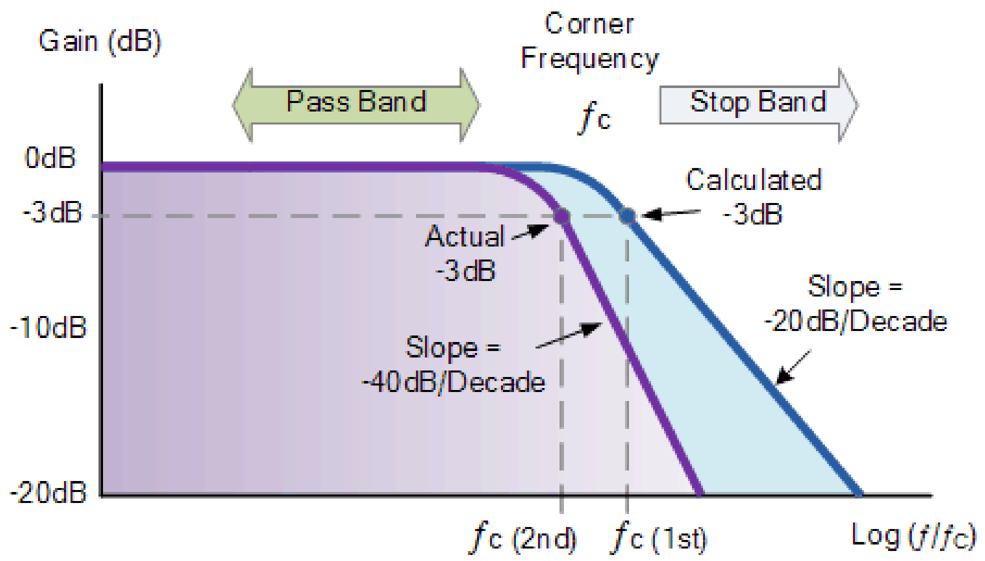


Figure 13- Graph of Low pass Filter Ouptut [20]

While RC filters may not offer the precision of more complex filters, their simplicity, cost-effectiveness, and ease of use make them an excellent choice for basic low pass filtering applications, such as in signal reconstruction following DDS.

3.5 Phase Shifting Circuit

In the circuit design, the phase shifter shifts the input signal by 90 degrees relative to the reference signal. The 90-degree phase shift facilitates the detection of any changes in phase.

The team explored two approaches for implementing a phase shifter: one utilising an operational amplifier and the other using a transistor. The team decided to proceed with the op amp-based phase shifter due to ease of construction and integration offered by the op amp design. One of the reasons the op amp phase shifter was chosen is because the frequency characteristics of the chosen op amp were well-suited to complement the frequencies of other components in the circuit, ensuring compatibility between components.

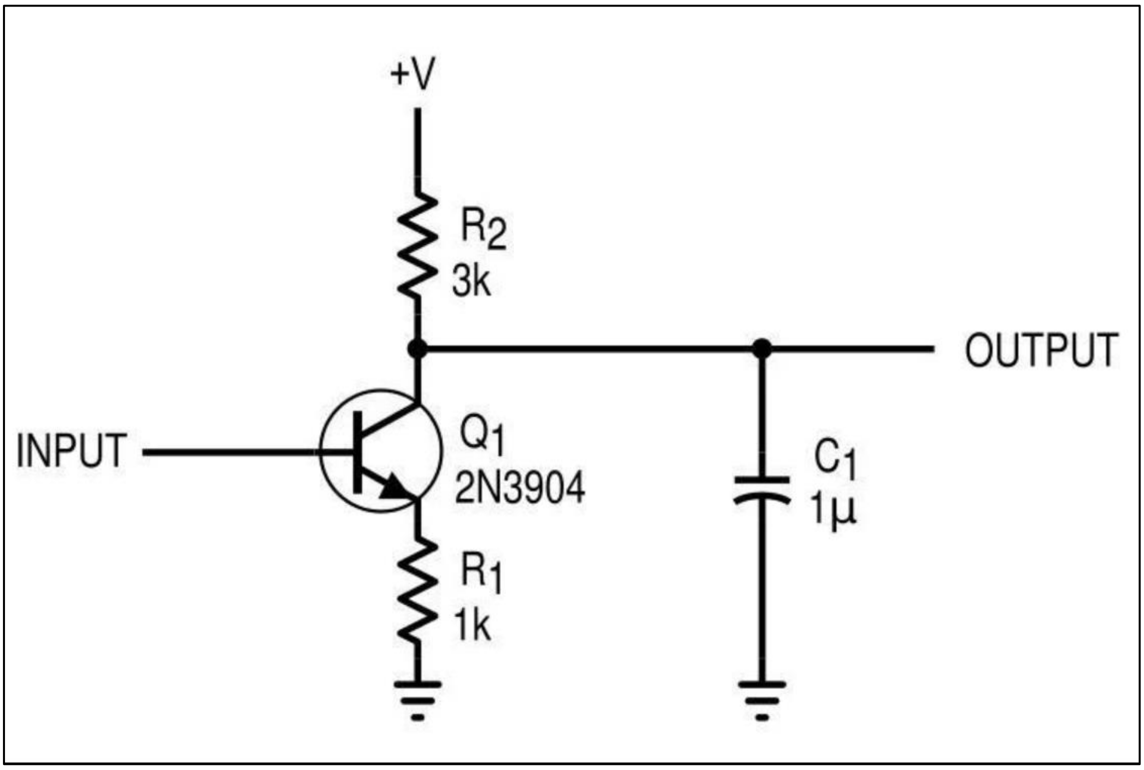


Figure 14- An example of a transistor based phase shifter [6]

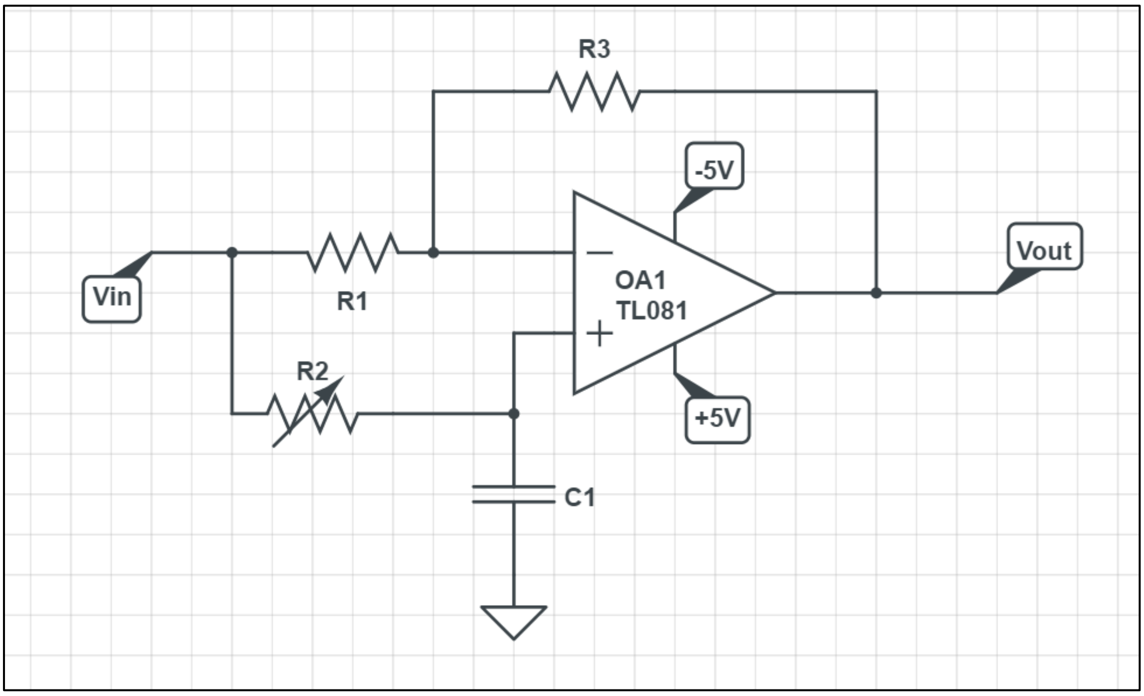


Figure 15- Op Amp based phase shifter

3.6 Phase Detector

The purpose of the phase detector is to detect and record any change in phase between two signals. The research paper mentions the use of the AD8302 integrated circuit as a phase detection system. Using this as a starting point, phase detection using AD8302 was investigated.

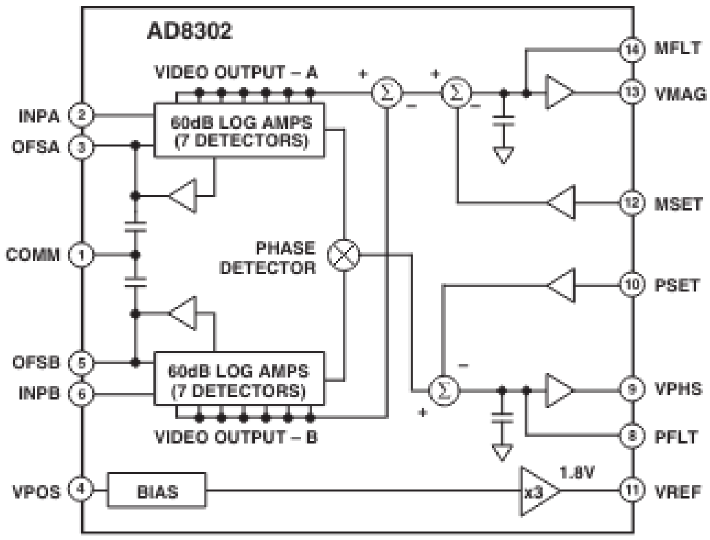


Figure 16- Block Diagram od AD8302 [21]

The IC takes two inputs INPA and INPB from pins 2 and 6. These signals are then passed through logarithmic amplifiers and the outputs of the amps are passed through an XOR.

The resulting phase information is then expressed as a voltage via the pin VPHS. The board also outputs a stable 1.8-volt reference voltage. The phase pin can then be analysed by the microcontroller and processed on a pc. [21]

As per the research paper, the device should be able to read a phase difference of 90 degrees to begin with and have a wide range around this for recording a change in phase.

PHASE OUTPUT	Pin VPHS		
Output Voltage Minimum	Phase Difference 180 Degrees	30	mV
Output Voltage Maximum	Phase Difference 0 Degrees	1.8	V

Figure 17 - Datasheet Specification of Voltage and Phase Output [21]

The datasheet stated the board can detect phase difference in the range of 0-180 degrees.

After initial research, it was decided to proceed with using an AD8302 to be used as a phase detector and after consultation with the supervisor the project was allocated an EVAL-AD8302-ARDZ.

3.6.1 EVAL-AD8302-ARDZ

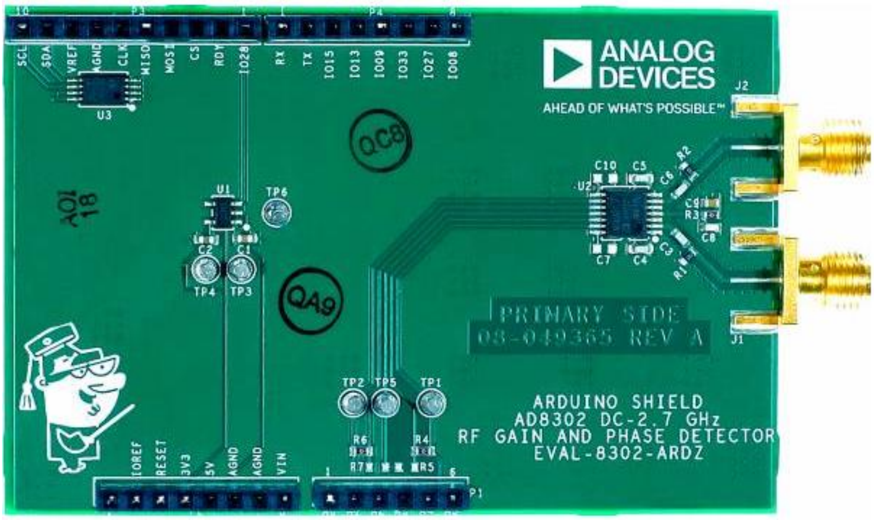


Figure 18- EVAL-AD8302-ARDZ [22]

This consists of the AD8302 IC integrated further into an Arduino Shield type of board. This is very useful for simple integration with the Arduino UNO microcontroller as it has the correlating pins and can simply slot in on top of an Arduino.

The same specs from the datasheet of the AD8302 apply to this board in terms of its phase response characteristics. The only drawback of using this board is that there is no software support for using it with Arduino. There is also a lack of documentation and diagrams for the full board. Although the board says on it “ARDUINO SHIELD” it was primarily designed to be used with the Analog Devices microcontroller. [22]

The EVAL-AD8302-ARDZ is designed to work as a shield for **EVAL-ADICUP3029** and **DC2026C** (also called **Linduino One**). For **EVAL-ADICUP3029**, PC software GUI and device development drivers are available. [22]

3.7 Reader and Sensor Coils

In the circuit design, the Sensor and Reader Coils facilitate wireless communication and data transmission. The Reader Coil operates as the primary transmitter and receiver within the inductive coupling system, responsible for initiating energy transfer to the Sensor Coil and capturing the signals it emits. Meanwhile, the Sensor Coil, in conjunction with a sensor capacitor, serves as a sensitive detector of environmental fluctuations, detecting changes such as variations in pressure. This capability provides essential data for the system's overall functionality.

The team explored two types of coils: planar coils and air gap coils. Planar coils, characterised by their flat, two-dimensional structure etched onto a PCB, offer advantages in compactness and integration onto circuit boards. Conversely, air gap coils, constructed by winding wire around a non-magnetic core, provide a three-dimensional structure with a gap between windings.

Figure 19- Plannar coil

Figure 20- Air gap coil

The team chose air gap coils due to the greater design flexibility and simpler manufacturing processes. By adjusting factors like diameter and spacing between windings, the inductance value could be easily modified, enabling rapid customisation to meet circuit requirements.

$$L = \frac{n^2 \times \mu_0 \times a}{l}$$

Figure 19- Formula for air gap inductor

3.8 Microcontrollers

For this project one of the aims from the start was to try and integrate the radio frequency system around the Arduino Platform.

The team had previous experience using Arduino UNO and it was also interfaceable with the other components of the circuit. The microcontroller is an important part of the circuit as it is essentially controlling different aspects of the circuit as well as being used to record and send data to the PC for analysis.

3.8.1 Arduino UNO

Arduino was chosen to be used for controlling the phase detector and for recording the phase data. This is an excellent choice for microcontroller as it will simply interface with the EVAL-AD8302-ARDZ in addition to interfacing with the PC via USB.

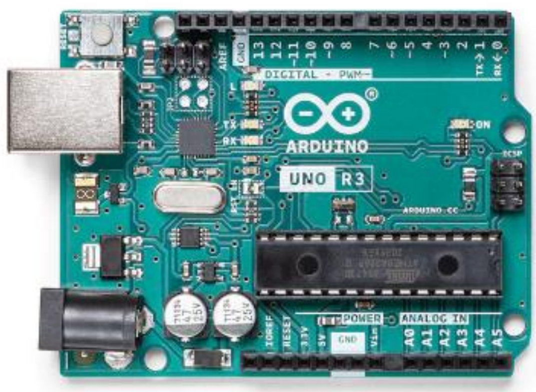


Figure 20- Arduino UNO [23]

The UNO operates at 5-volt logic level and has several different ports required for this project. The analogue in ports will be useful for performing Analogue reads on the phase pin as well as taking the EVAL-AD8302-ARDZ reference voltage as an input. The Arduino also comes with a dedicated IDE which software for boards can be developed in C and C++.

3.8.2 Arduino NANO 3.3 BLE

After deciding to use the waveform click as the DDS board, it then became necessary to find a 3.3-volt logic level microcontroller. The only such microcontroller the team has access to was the Arduino NANO 3.3 BLE. This is a smaller lightweight version of the UNO which retains most of its functionality but can operate at a lower logic level.

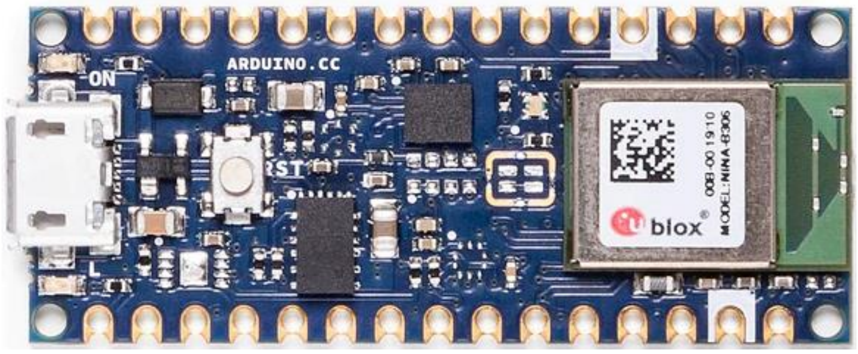


Figure 21- Arduino NANO 3.3 BLE [24]

The NANO is programmable just like the UNO and both can be controlled from the one PC. The NANO had the correct SPI pins needed to communicate with the waveform click as well as V-out of 3.3 Volts.

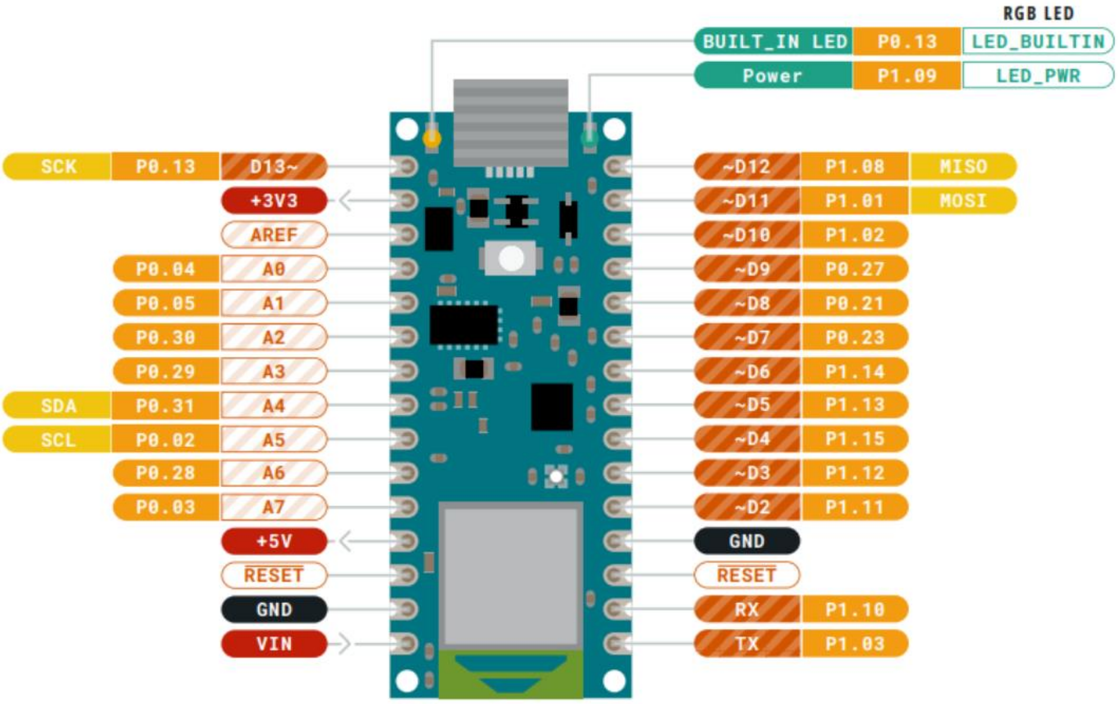


Figure 22 - Pinout Diagram of Arduino NANO 3.3 BLE [24]

As can be seen in the above diagram the correct clock pins and SPI data pins are present. Waveform Click SCLK pin to Arduino Nano D13, Waveform Click SDATA to Arduino Nano D11 and Waveform Click FSYNC to Arduino Nano D10.

3.9 Unity Gain Amplifiers

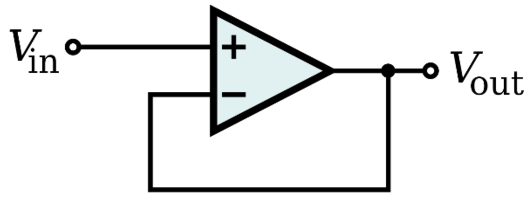


Figure 23- Unity Gain Op-Amp Circuit Diagram [25]

In the circuit diagram provided in the research paper, there are three such amplifiers used at different stages. They are used for the following reasons:

1. **Impedance Matching**: Incorporating a unity gain amplifier ensures that the output from the LP-HP phase shifter is not adversely loaded by the input of the phase detector, maintaining the original characteristics of the signal by providing effective impedance matching between stages.

2. **Signal Integrity**: The use of a unity gain amplifier preserves the amplitude and quality of the signal by preventing voltage drops caused by loading, ensuring that the signal transmitted to subsequent stages remains robust and undistorted.

3. **Drive Capability**: The buffer amplifier enhances the circuit's ability to drive subsequent stages without compromising signal integrity, providing sufficient current drive capacity, and preventing signal distortion.

4. **Stability and Noise Reduction**: A unity gain amplifier contributes to circuit stability by minimizing feedback between stages and reduces the susceptibility to noise, ensuring cleaner and more stable signal processing throughout the circuit.

Chapter 4: Hardware Prototyping of Reader System

4.1 Introduction

This chapter details the development of the prototype device, specifically looking at the exact implementation of each component into the full system.

4.2 Workflow of System

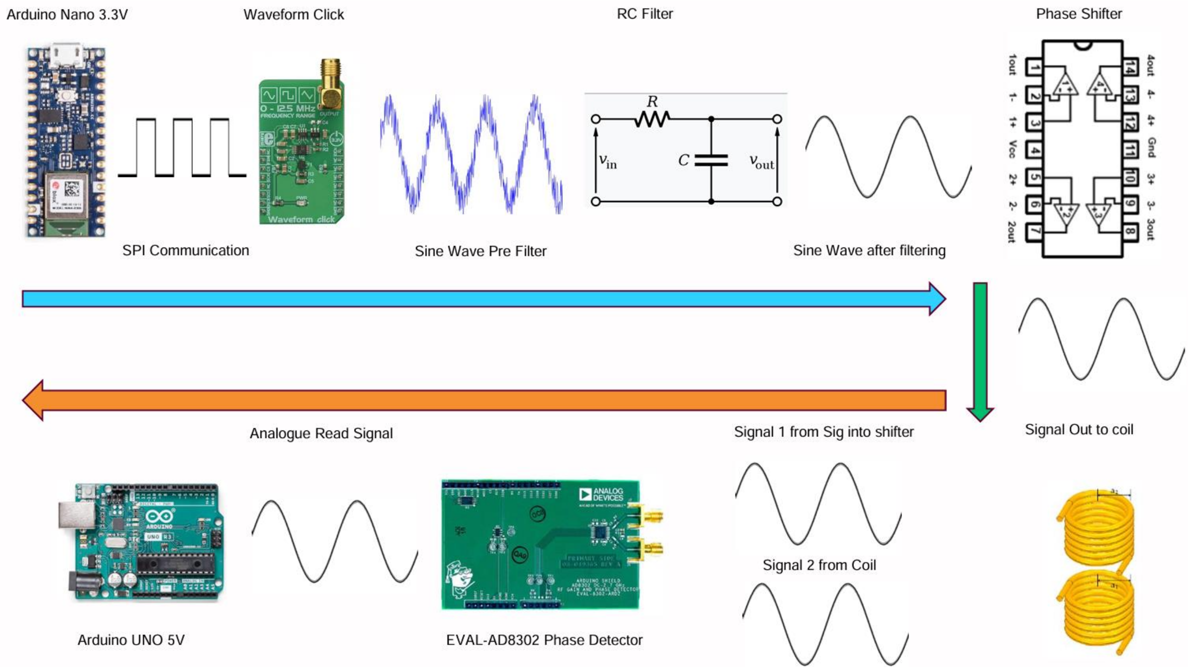


Figure 24 - Flow of Signals through Reader System

The Arduino nano communicates to the waveform click via SPI. The waveform click is programmed to output a certain frequency by having its register values written to by the Arduino nano. The pins 10,11 and 13 on the Arduino are connected to the SDI, SCK and FSN pins on the Waveform click. These pins act as serial data in, serial clock and chip select. The FSN pin is used to control when the waveform click will listen for SPI commands from the nano. When the waveform click has the correct settings initialised it will output a sine wave. The sine wave is then passed through the low pass resistor-capacitor filter. The filter then attenuates any frequencies above the cut off frequency.

The filtered sine wave is then sent to the phase shifting circuit. The op amps in the phase shifters integrated circuit cause a change in amplitude which subsequently causes a change in phase. The signal in of the phase shifter is also sent to port J1 of the EVAL-AD8302-ARDZ to establish a base line for phase difference with the signal out of phase shifter. Signal out of phase shifter is then sent to interrogate the reader coil and can be fine-tuned via variable resistor. The signal from the reader coil is then sent to port J2 of the EVAL-AD8302-ARDZ for phase comparison. While this is being done, the sensor coil is brought near the reader coil for an energy transfer to take place electromagnetically. The phase difference is calculated on the AD8302 integrated circuit and expressed as a voltage to its phase pin. The Arduino UNO then performs an "analogRead()" on the phase pin to extract its data. The Arduino then performs a conversion from its ADC value to voltage and then to phase. Next, the data is sent to the PC using the Visual Studio Code serial monitor for Arduino. The data is expressed in a comma separated value format which is then logged to a chosen directory on the host pc. A python script is then used to format the data and graph it, using pandas and Matplot.lib.

4.3 System Overview @3MHz

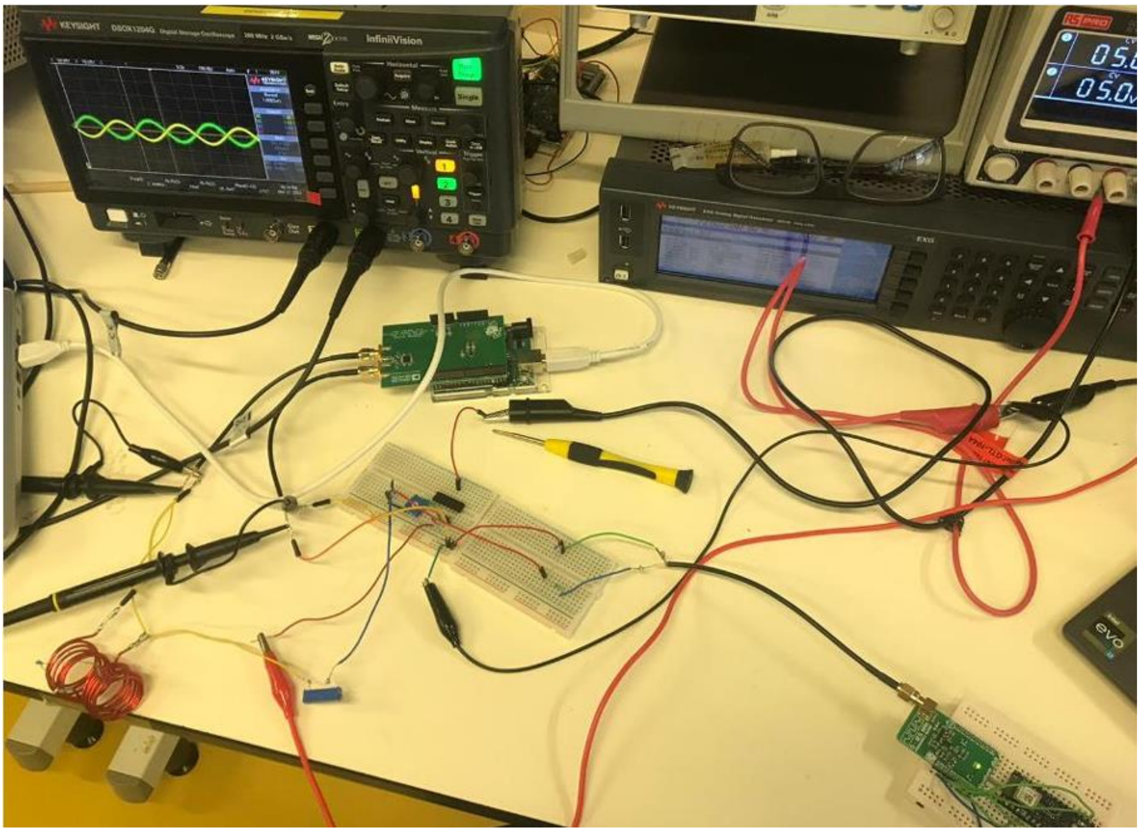


Figure 25- Full prototype Reader System

In the above diagram the system is being tested operating at 3MHz frequency with a starting amplitude of 1.4 volts peak to peak. The system operates as described in the system workflow. In this configuration of the system there is some calculations necessary to build the system.

4.3.1 Calculation and Assembly of Filter

To build a filter suitable for a signal of 3MHz, a calculation is performed to ascertain the correct values for the resistor and capacitor.

Using the formula:

$$f_{\text{cutoff}} = \frac{1}{2\pi RC}$$

(26)

It is rearranged to solve for C:

$$C = \frac{1}{2\pi R f_{\text{cutoff}}}$$

(27)

The cut off frequency for a 3MHz filter is selected at 4MHz to allow the 3MHz signal to pass through and a resistor value of 1000 ohms is selected as they are widely available in the lab.

Subbing this in gives:

$$C = \frac{1}{2\pi \times 1000 \times 4 \times 10^6}$$

(28)

Solving:

$$C = \frac{1}{2\pi \times 4000000} \approx 39.79 \times 10^{-12} \text{ Farads}$$

(29)

Using these values results in a capacitor of around 40 picofarads being chosen.

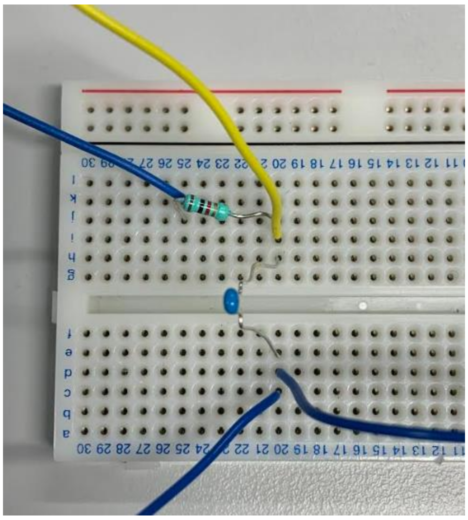


Figure 30 - Resistor Capacitor Filter on Breadboard

As can be seen in the above picture the resistor capacitor was correctly assembled.

4.3.2 Calculation of DDS Register values and Correct Register Addressing

From studying the data sheet of the AD9833 it was discovered that a calculation must be performed to ascertain the correct binary instructions for the registers of the device to correctly output a sine wave. [16]

$$FreqReg = \frac{f_{OUT} \times 2^{28}}{f_{MCLK}}$$

Thus, for this example, *Freq 0* = 400 Hz.

$$FreqReg = \frac{400\ Hz \times 2^{28}}{25\ MHz}$$

= 4295 decimal = 0x10C7 = 0001 0000 1100 0111

Figure 31- Formula and Method for Calculating Register values from Datasheet

Using this method for 3MHz:

$$FreqReg = \frac{3,000,000 \times 268,435,456}{25,000,000}$$

(32)

Calculating:

$$FreqReg = \frac{805,306,368,000,000}{25,000,000} = 32,212,254.72$$

(33)

- Rounded *FreqReg* = 32,212,255 (decimal)
- Convert to hexadecimal: 0x1E84800x1E8480.
- Convert 0x1E84800x1E8480 to binary = 000111101000010010000000000111101000010010000000

Now the correct register value has been calculated this can be added to the Arduino nano code.

4.3.3 Soldering Arduino Nano Pins

The Arduino NANO 3.3 BLE comes with no header pins as standard.

Figure 34 - Arduino NANO with no Header Pins

To overcome this, headers were ordered and soldered to the Arduino to allow it to interface with the waveform click.

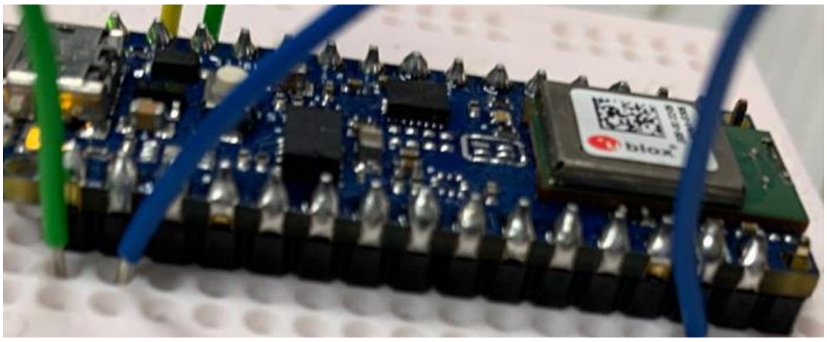


Figure 35- Arduino NANO with Header Pins Soldered

4.3.4 Implementation of Phase Shifter

Shortly after the team decided to construct the phase shifter using operational amplifiers, they received an OPA358AIDCKT which had a frequency of up to 80MHz, but due to its minuscule size it would prove to be difficult to implement. The team then looked at using a DIP (Dual Inline Package) due to their larger size, exploring the op amps available to them, they found a suitable one in the TL074N which had a suitable frequency (3MHz) for matching with the other components in the circuit.

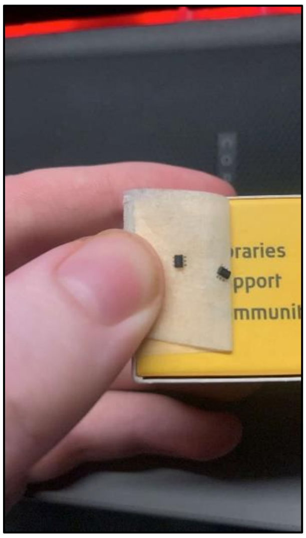


Figure 36- Two OPA358AIDCKT op amps

The team then constructed the phase shifter circuit using two fixed resistors of 10KΩ and a variable resistor of 10KΩ and a capacitor of value 5pF. Based on calculations, these resistor and capacitor values would provide a phase shift of 90 degrees.

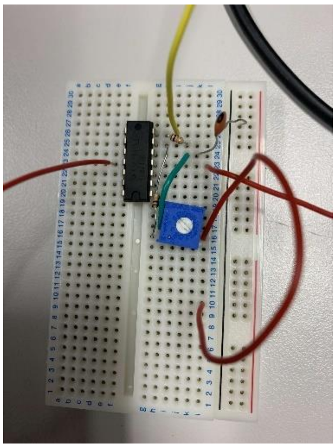


Figure 37- Phase shifter using TL074N

4.3.5 Implementation of Coils

Once the team decided to use air gap coils for the sensor and reader coils, the next step was to carry out calculations to design two inductor coils of the same inductance value that could operate at the target frequency of 3MHz. Due to the coils being handmade, it was expected that some inaccuracies could possibly occur during the manufacturing process. On the third attempt, the coils were close enough to use for the overall system prototype. When the sensor coil was paired with an 80pF capacitor, a resonance frequency of 3.79MHz was observed on an ENA Series Network Analyzer.

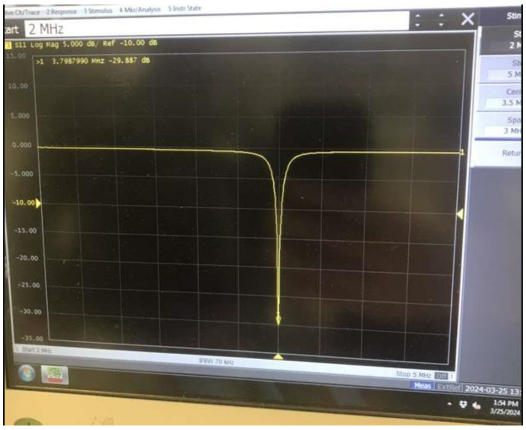


Figure 38- Resonance Frequency of coils on ENA

Chapter 5: Software Development for Integrated Circuit boards to Interface with Arduino and Python Automation.

5.1 Introduction

This chapter discusses some of the challenges that had to be overcome from a software point of view. Additionally, it highlights the functionality achieved by implementing new software libraries for two of the boards and automation setup for rapid data analysis.

5.2 Challenges with EVAL-AD8302-ARDZ

One of the main challenges of this project was the lack of documentation and support for the EVAL-AD8302-ARDZ phase detection shield. The board itself is described as an "Arduino Shield" by Analog Devices, however after going to great lengths to find software support and documentation for the board, the team decided to contact analog devices support. To which they clarified the board is only supported for Linduino and EVAL-ADICUP3029. Due to this misconception about the board being an Arduino Shield the team had decided to use it for the project. After examining the functional block diagram of the shield and the data sheet of the AD8302 IC, the development of a library began using git for source control.

5.2.1 Software Library for EVAL-AD8302-ARDZ

The design of a library was drawn up and through several trial-and-error iterations using git as a valuable tool, the library was successfully created.

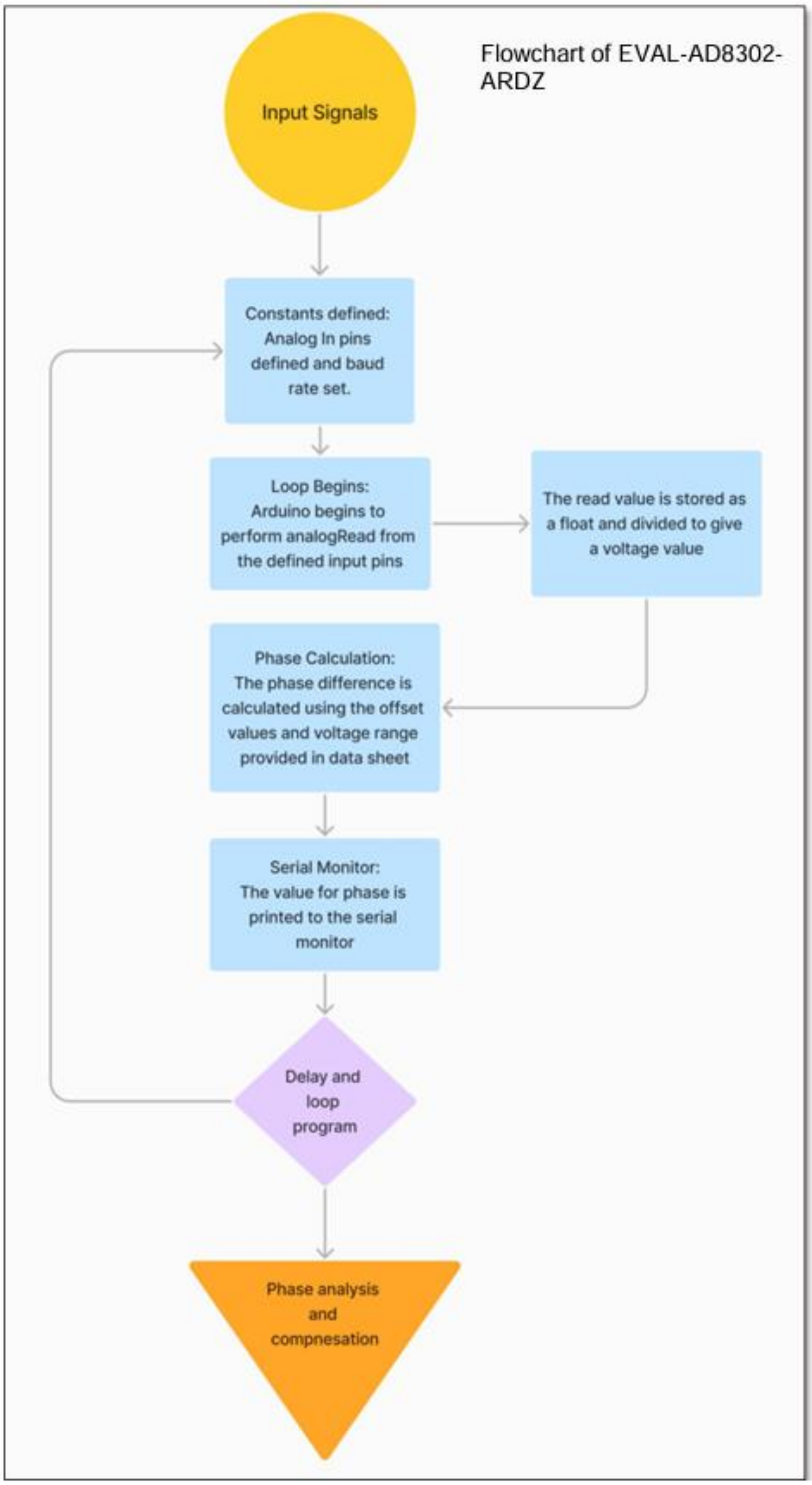


Figure 39 - Flowchart of Phase Detection Code

5.3 Challenges with Waveform Click

The waveform click is made by MIKROE and designed to interface with other MIKROE products such as microcontrollers and is not typically meant to be used with Arduino. The logic level issue was overcome by sourcing a 3.3-volt Arduino NANO 3.3 BLE. However, all the software development kits and libraries provided for the waveform click are not suitable for Arduino. There are some standard libraries for one of its ICs the AD9833, but they do not consider the other two integrated circuits onboard the waveform click and produce a very noisy sine waveform.

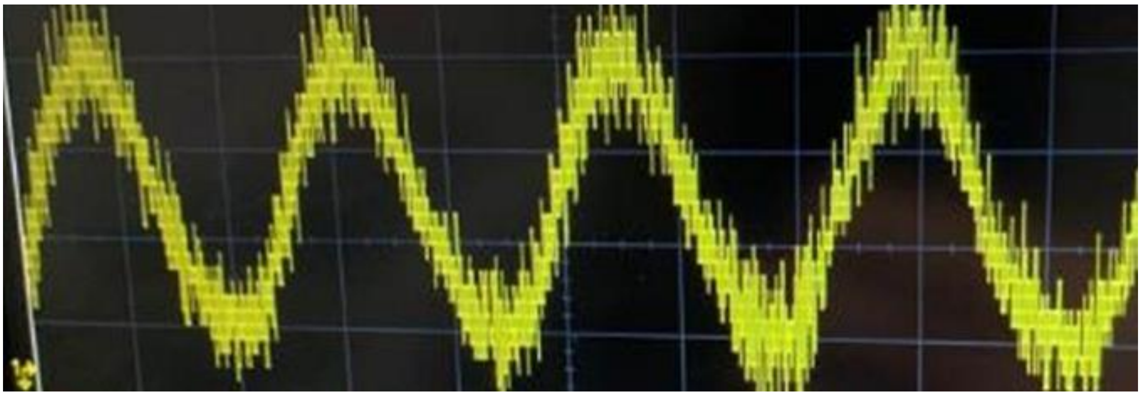


Figure 40- Noisy Output of Waveform Click using provided library

As the new library for the waveform click was being developed there was major issues with maintaining a stable output when testing the system, through debugging the SPI protocol and a process of elimination it was determined that correct register addressing was the issue causing the lack of consistent outputs.

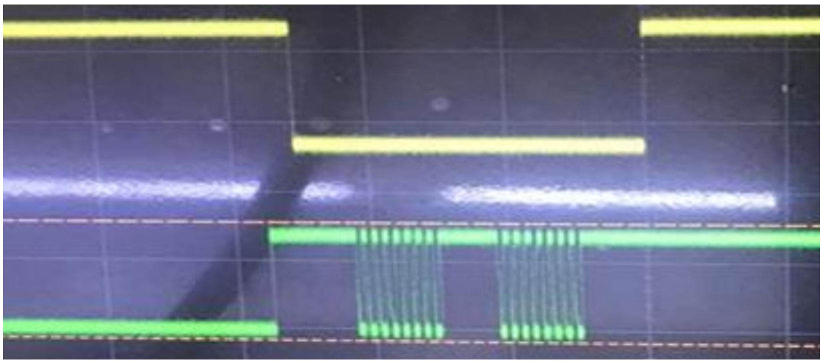


Figure 41 - SPI debugging on Oscilloscope

After a lengthy debugging process the library for controlling the waveform click was successfully developed allowing control over its two programmable integrated circuits.

5.3.1 3MHz Library for Waveform Click

Using an object orientated approach, a new library was developed and tested. This library's main function is to write the correct register values using correct register addressing on the waveform click to create a sine wave object. It is also possible to control the setting of the 63-position digital potentiometer of the AD5227 using this code also.

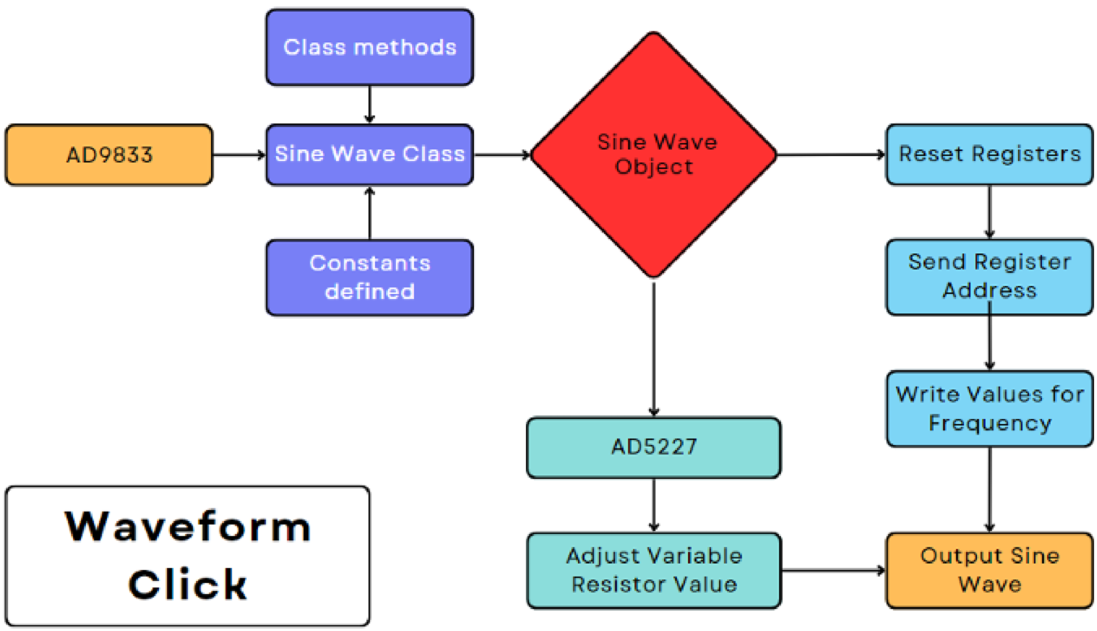


Figure 42 - Flowchart of Waveform Click Code

5.4 Automation

After conducting several tests using the phase detector system to validate its output, data analysis had to be conducted using graphs and comparison to validate the results and interpret the systems readings.

Originally this was done by opening the serial monitor on Arduino and copying its output into notepad. This is very slow as Arduino serial monitor only allows what is visible on screen to be copied. The data is then manually formatted in notepad and saved as csv format to be opened in Microsoft excel. The data is then sorted into appropriate columns and rows. Finally using the scatter function on excel, graphs are created. To make high quality graphs and analyse the system could take hours at a time using this approach. To streamline this process, the team investigated automating the process. The first thing that was done was to use Putty as a serial monitor allowing a full serial session to be saved to a file. While speeding up the process there was still much room for improvement. This is when everything on the Arduino was ported over to Visual Studio Code.

5.4.1 Visual Studio Code for Automation and Python Analysis

The Arduino Extension for VS code was downloaded and installed along with the Serial monitor extension for VS code. This allowed the Arduino code to be edited, uploaded and the serial port listened to. The main advantage of the VS code approach was having all this in one place as well as the ability to save serial monitor sessions to log files in a chosen directory on the host pc.

Then using Pandas, Matplot.lib and NumPy, python analysis scripts were made for rapidly analysing the output of the system.

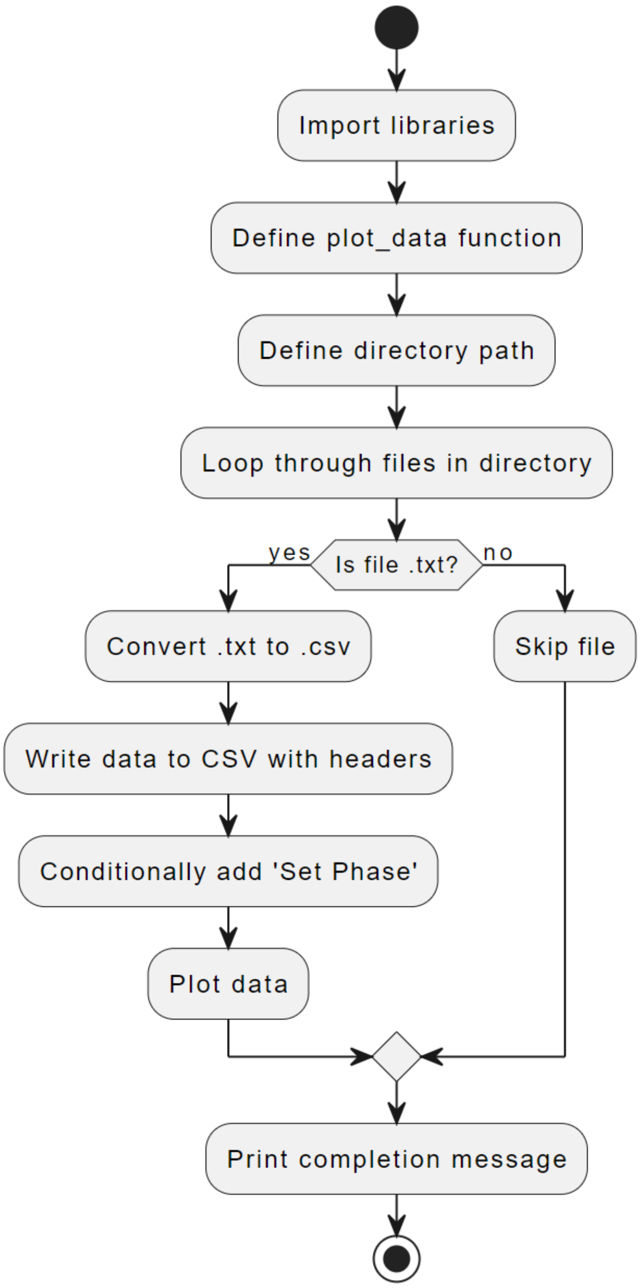


Figure 43 - Flowchart of Python Analysis Automation Script

Chapter 6: Testing and Validation of Results

6.1 Testing the Waveform Click DDS

The waveform click is responsible for generating the sine wave signal into the rest of the circuit and because of this is, it is a key component. Due to the nature of the phase analysis being the core purpose of the circuit it is very important to generate and maintain a very smooth sine wave for it to be properly analysed. If there is too much noise on the wave it will not be fit for purpose. To be able to isolate problems in the circuit it is important that the starting point for the wave is kept precise and accurate.

The waveform click's data sheet states that it can output a smooth sine wave up to frequencies of about 5MHz. This is somewhat of a limitation that will be discussed later in the report. It is also important that the wave be greater than 1MHz due to findings that the phase detector board does not work well with low frequency in practice.

The op amp used in the circuit should not exceed 4 MHz, so it was decided to generate a signal of 3MHz for the system to functional optimally. 3MHz is a high enough frequency that the phase detector board can accurately measure the phase angle and is low enough that it doesn't near the noisy threshold of 5MHz on the waveform click.

6.1.1 Waveform Click with default library and Voltage Divider

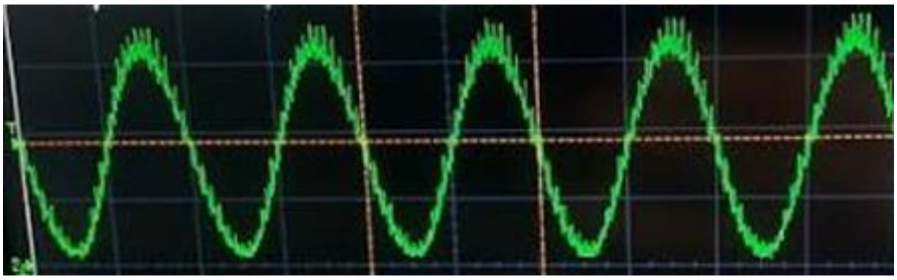


Figure 44 - DDS ouput @1KHz on Ossiliscope

This was the first test conducted using the DDS with a voltage divider configuration mentioned previously. The waveform is somewhat noisy, but this is at very low frequency and will likely deteriorate as the frequency increases.

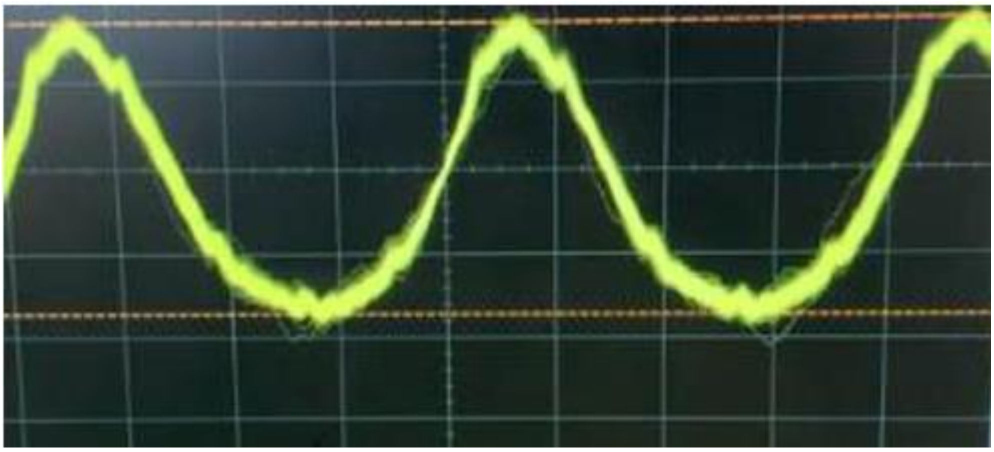


Figure 45 -DDS outut @1MHz on Ossiliscope

The above picture confirms the noise and lack of precision using a voltage divider setup to generate a smooth sine signal.

6.1.2 Default library with Arduino NANO

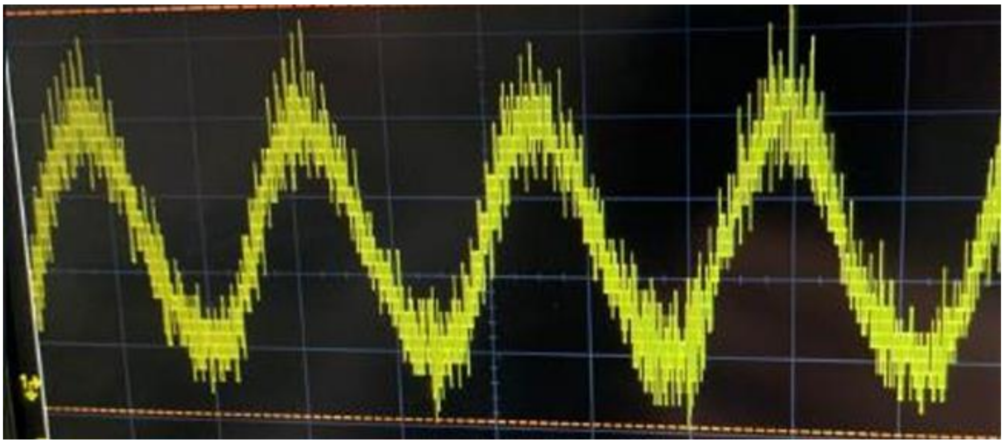


Figure 46 - DDS output @5MHz using Nano and Default Library

This output is extremely noisy and would not be suitable for use in the circuit at all. After debugging the wiring and hardware setup the waveform observed was still very noisy.

6.1.3 New Software Library with Arduino Nano

Using the code mentioned previously in the software development chapter, a much-improved result was observed using the waveform click.

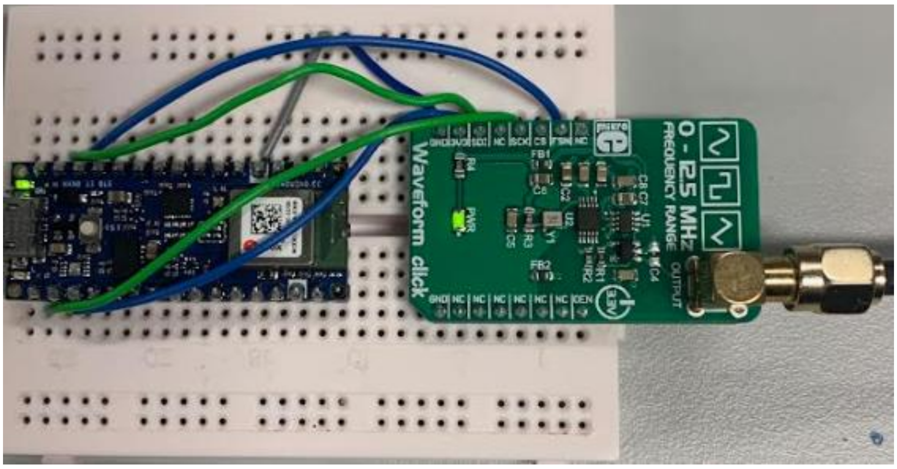


Figure 47 -Waveform Click and Nano

To keep debugging simple, the hardware setup was improved as much as possible by keeping the wire lengths as short as possible which is important in SPI applications.

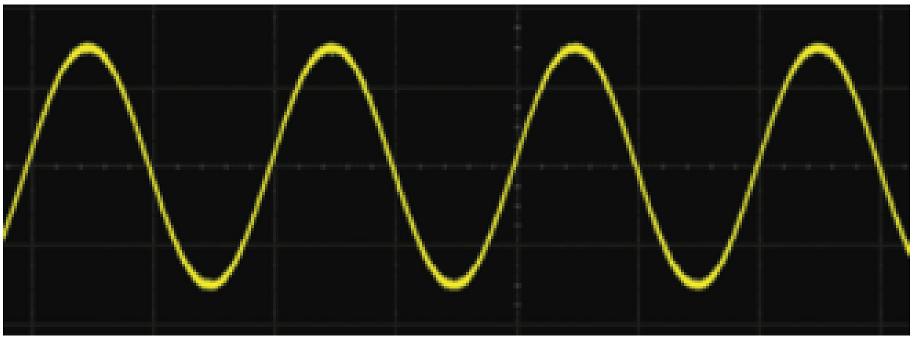


Figure 48 - Smooth Sine Ouput @1MHz

As can be seen in the above figure, the new library and hardware setup are a significant improvement on previous iterations. However, upon repeating tests several times a huge discrepancy was noticed on the oscilloscope.

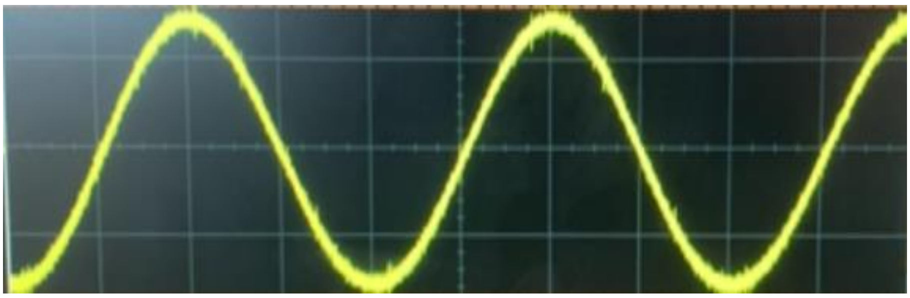


Figure 49- Oscilloscope 1MHz

The above figure shows a different quality of waveform using the exact same software and hardware setup as used previously. Several examples like this were observed at different frequencies.

6.1.4 Correct Register Addressing DDS

The lack of consistency in the DDS output was leading to issues during testing as the board could not be relied upon to have a useable output. Following this, the data sheet was studied further and register addressing was proposed as a potential issue.

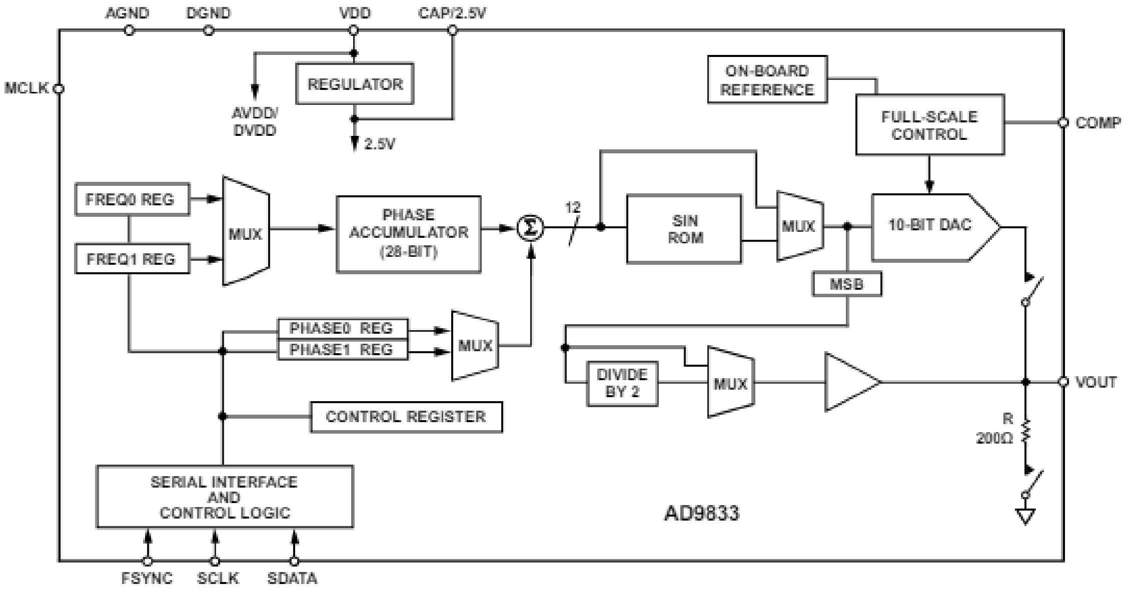


Figure 50 - Block Diagram of AD9833

As can be seen in the block diagram there are several different registers that may be addressed on the AD9833. Additionally, there are other integrated circuits on the waveform click which further emphasises the importance of correct addressing in the SPI communication.

The following sequence was found in the data sheet and incorporated to the code:

<u>Command Sequence Explained</u>

0x2100—Control Register

- DB13 is set to 1. This allows a complete word to be loaded into a frequency register in two consecutive writes. The first write contains 14 LSBs. The second write contains 14 MSBs.
- RESET bit DB8 is set to 1. This resets internal registers to 0, which corresponds to an analog output of midscale.

0x50C7—Frequency Register 0 LSB

- DB15 and DB14 are set to 0 and 1, respectively, which is 4the Frequency Register 0 address.

- The remaining 14 bits are the 14 LSBs of data: 0x10C7 = 01 0000 1100 0111

0x4000—Frequency Register 0 MSB

- DB15 and DB14 are set to 0 and 1, respectively, which is the Frequency Register 0 address.
- The remaining 14 bits are the 14 MSBs of data and are all 0s in this case.

0xC000—Phase Register 0

- DB15, DB14, and DB13 are set to 110, with DB12 set to don’t care (X), respectively, which is the address for Phase Register 0.
- The remaining 12 bits are the data bits and are all 0s in this case.

0x2000—Exit Reset

- A signal appears at the output of the DAC seven MCLK cycles after RESET is set to 0.

After implementing this change to the code, the same frequency and smoothness was observed every time when outputting a waveform.

6.1.5 Amplitude Control DDS

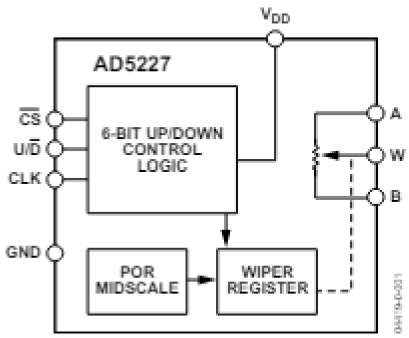


Figure 51 - Block Diagram of AD5227

The above figure shows the block diagram of the AD5227, this is used as a variable resistor on the waveform click. From studying the data sheet, code was added to the git of the waveform click and using delays, the amplitude control is engaged after the wave is initialised.

To conclude the testing of the DDS, the board can now accurately and precisely output a waveform at frequencies of 1MHz, 2MHz and 3MHz with their amplitude being controlled on a software level.

6.2 Testing the Resistor Capacitor Filter

The resistor capacitor filter was tested at varied stages throughout the project. It was most notably needed in the earlier stages before the DDS code had significantly improved.

Before the correct register addressing was added the filter was very much needed.

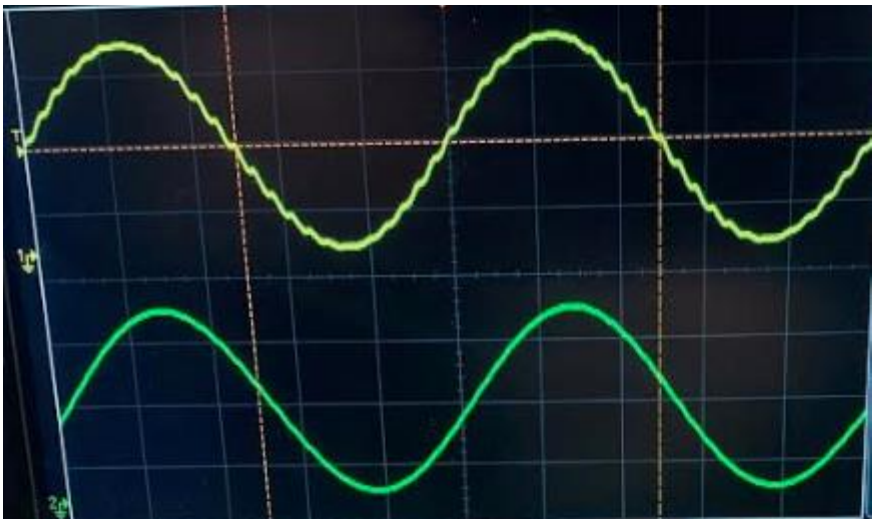


Figure 52 - DDS Output Before and After Filtering

As can be seen in the figure above, the DDS output at 3MHz was not very smooth (yellow) and after passing the signal through the filter, the observed waveform was much smoother (green). This shows that the filter was operating sufficiently.

The filter was tested in the Vector Network Analyser also which displayed that the filter was performing a bit less than its ideal values but still sufficiently for the purpose it served.

After the DDS code was updated and improved to its latest iteration, the effect of the filter became less noticeable.

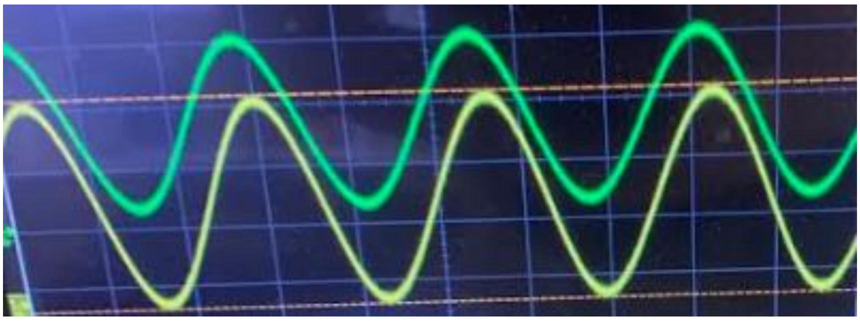


Figure 53 - DDS Output @3MHz

For example, in the above picture the green wave is the unfiltered and the yellow Is filtered, both are much the same.

6.2.1 Conclusion on Testing Waveform Click and Filter

To conclude the testing of the signal generator and the filter, both are operating sufficiently and are ready to be introduced to the full circuit to serve their purpose.

6.3 Testing the Phase Detector EVAL-AD8302-ARDZ

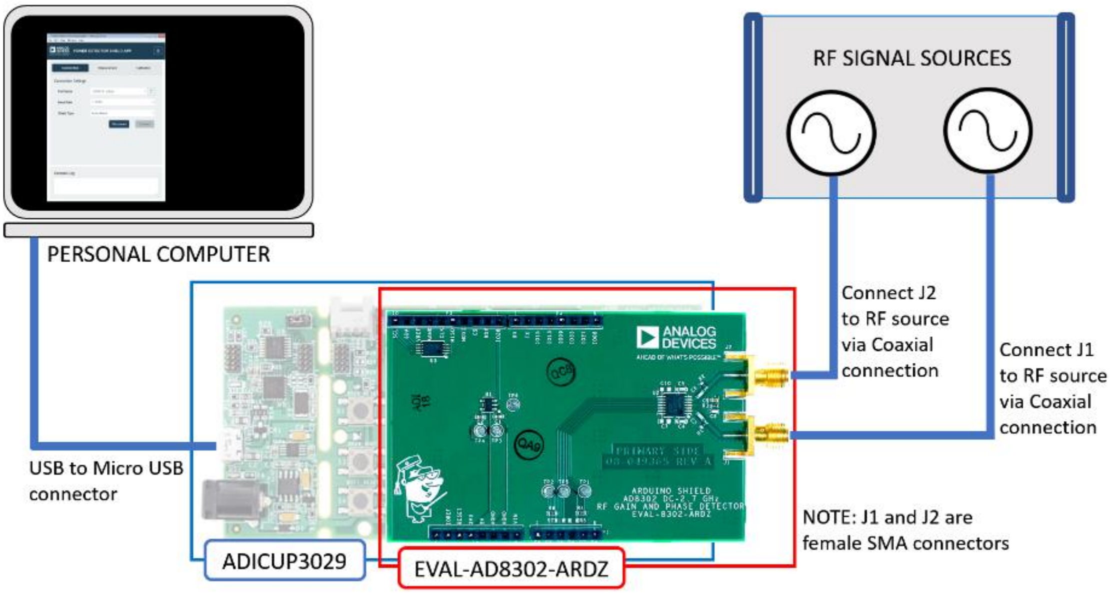


Figure 54 - Recommended Setup for Testing from Analog Devices [22]

To begin testing the phase detector, the procedure outlined on the Analog Devices wiki was followed. After installing the GUI for the phase detector, it became apparent that the recommended setup for testing was only meant for using the ADICUP3029 microcontroller. This meant that software had to be developed for the Arduino to allow it to interpret the phase data of the integrated circuit AD8302.

From a hardware point of view, the phase shield must be inserted into the Arduino UNO ensuring the pins are aligned correctly. Then two independent signals should be applied to ports J1 and J2 of the phase detector.

The first thing that was done was to map the Arduino readings of the phase pin to the correct mapping specified in the data sheet. With 1.8 volts corresponding to 0 degrees and 30 mV corresponding to 0 degrees phase difference.

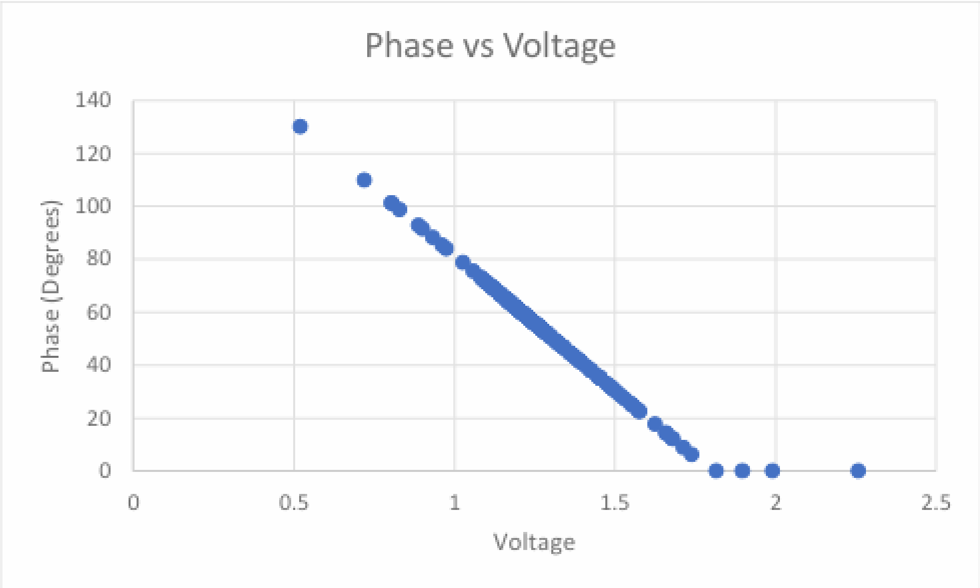


Figure 55 - Graph of Phase Values Ranging from 0 - 130 Degrees

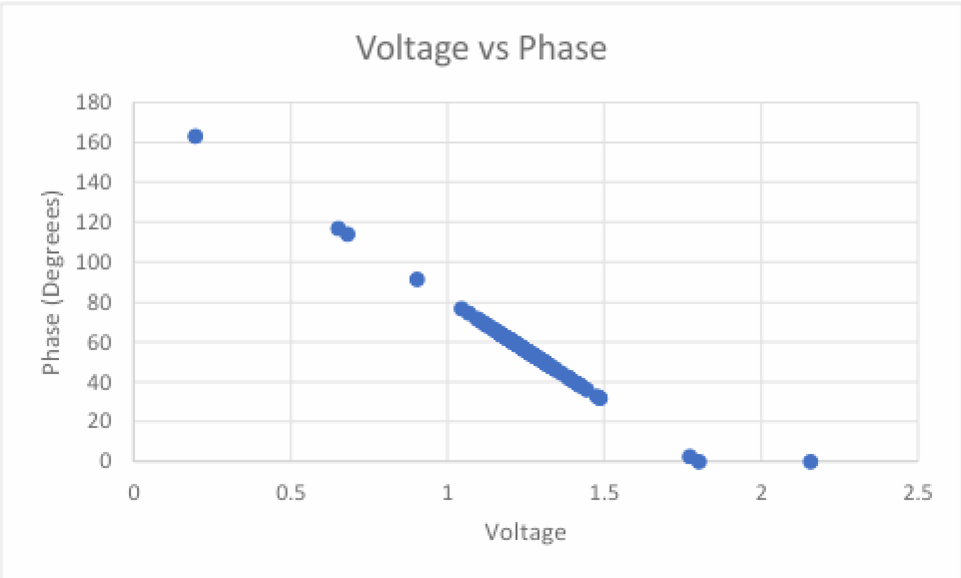


Figure 56 - Graph of phase Values Ranging from 0 - 160 Degrees

As can be seen in the above graphs the Arduino can interpret the phase pins voltage correctly and map it to a phase value between 0 and 180. This was done using two out of synch sine waves and by changing their phase offset.

6.3.1 Synching the Signal Generator to Validate Output

To validate the readings from the phase detector the tests had to be heavily controlled using the signal generators built in phase detector and synching mechanism to verify that the same result can be observed from the Arduino as on the signal generator.

Below is a guide to synch the signal generators for this purpose.

- *Front-Panel Operation*: Press [Utility] and select the **Sync** softkey again to toggle between "off" and "on".
- *Remote Interface Operation*:

`OUTPut:SYNC {OFF|ON}` *Setting is stored in non-volatile memory*

Figure 57 - Instructions from User Manual of Signal Generator

Figure 58 - Connecting Two Signal Generators using BNC - BNC Cable

Figure 59 - Rear of Signal Generator Triggering connected

The signal generators are synched together by using the burst function and controlling the clocking mechanism to ensure the sine waves are synched using a trigger from one generator to the other. One generators clock becomes slave to the other generators master clock.

6.3.2 Testing Synched Outputs @1KHz

The signal generators were then tested against the phase detector board at several intervals of phase.

Figure 60 - Two Sine Signals with 0 degree phase difference

As can be seen in the above oscilloscope output there is 0 degrees phase difference between the two signals.

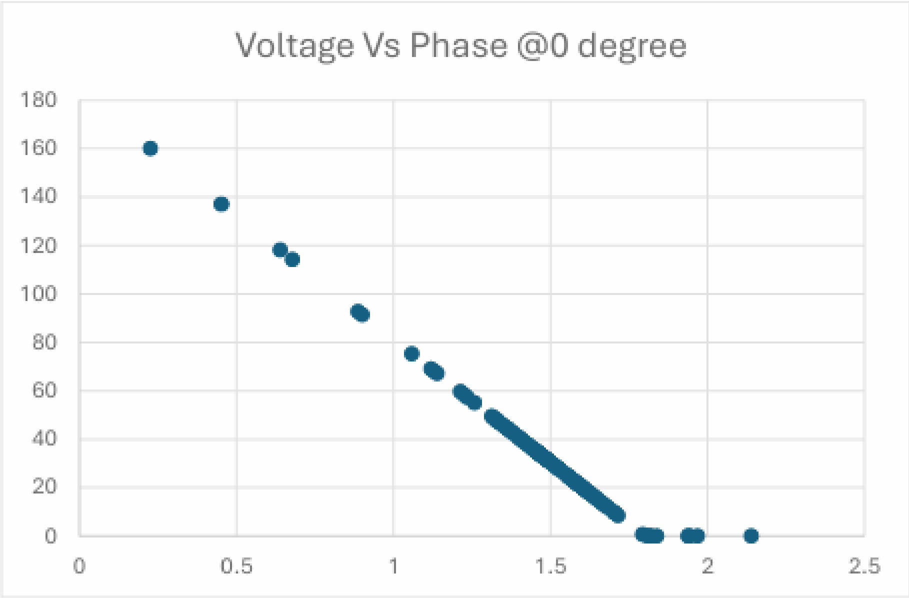


Figure 61- EVAL-AD8302-ARDZ Output @0 Degrees phase difference @2KHz

From the studying the graph and comparing it to the oscilloscope output, the phase detector board is not working at all for 0 degree-controlled phase at 2KHz frequency.

Figure 62 - Two Sine signals with 45 degrees phase difference

The signal generators were then set to be 45 degrees out of phase.

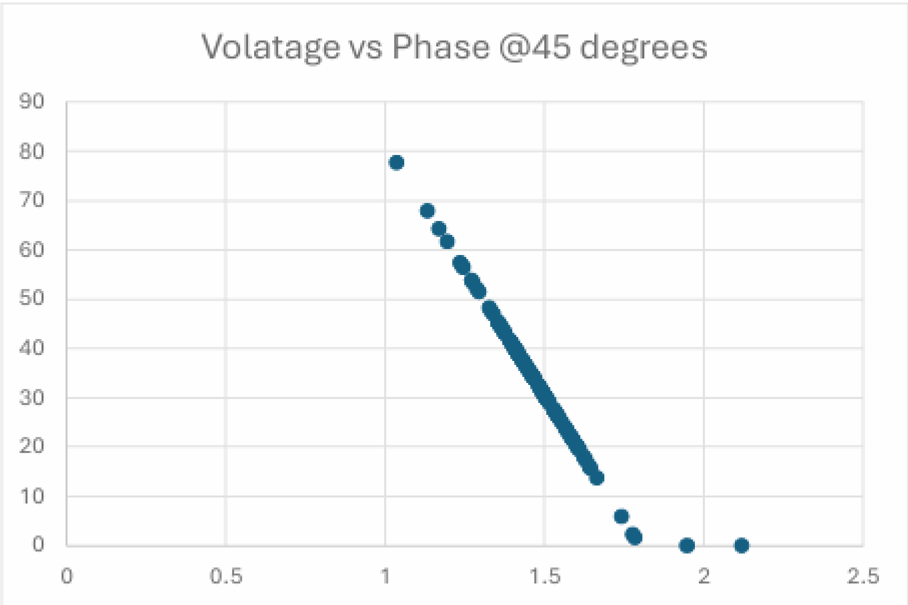


Figure 63- EVAL-AD8302-ARDZ Output @45 Degrees phase difference @2KHz

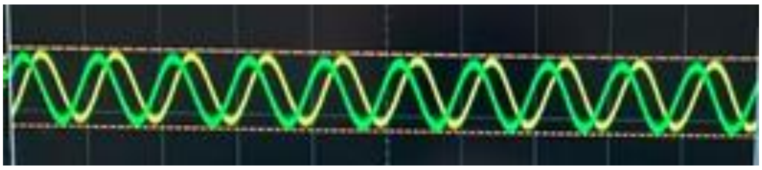


Figure 64 – Two Sine signals with 90 Degrees phase difference @2KHz

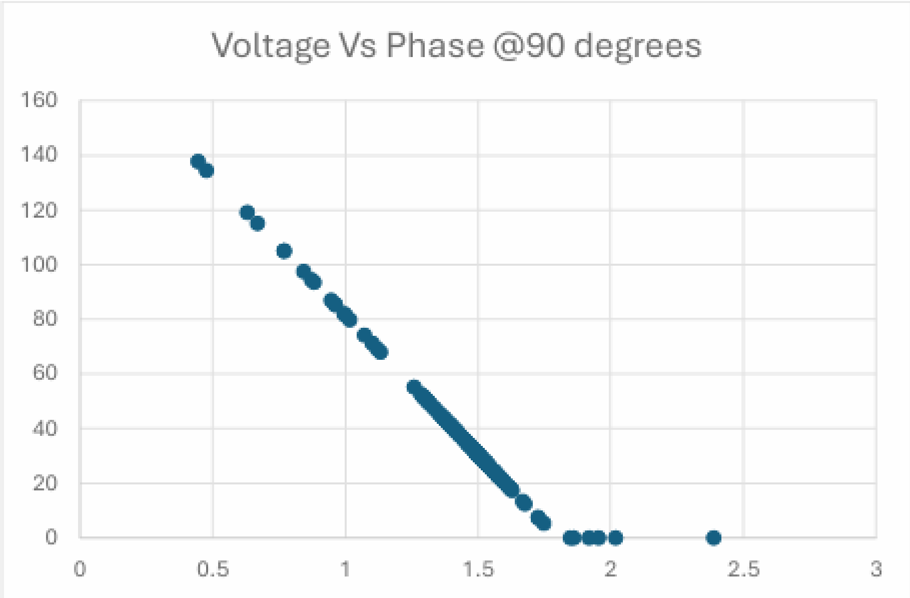


Figure 65- EVAL-AD8302-ARDZ Output @90 Degrees phase difference @2KHz

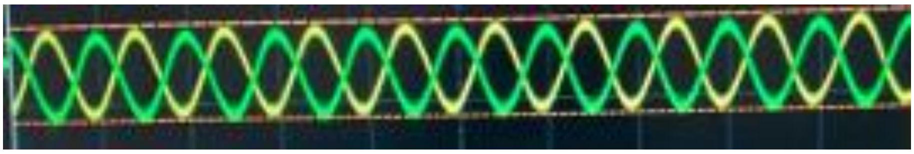


Figure 66- Two sine signals set to 160 degrees phase difference

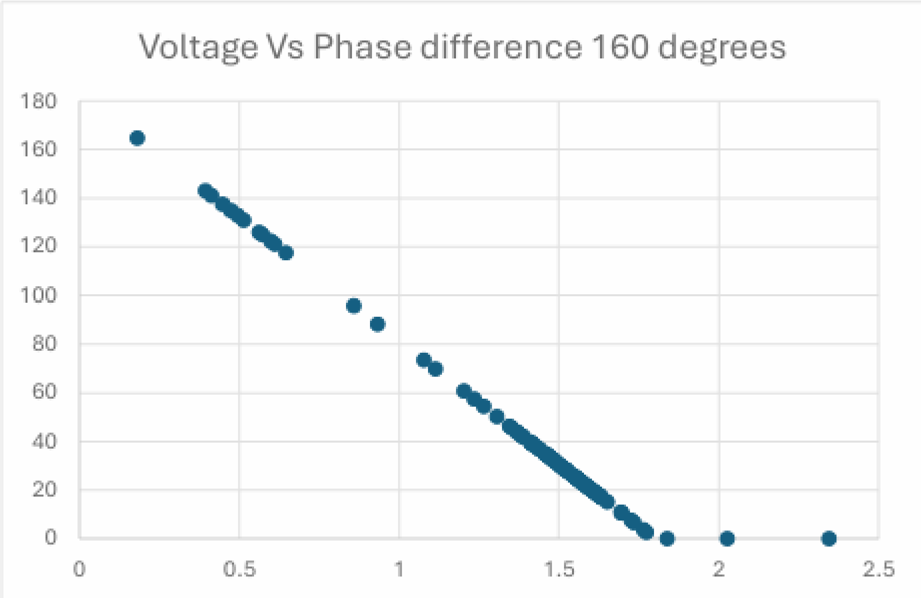


Figure 67- EVAL-AD8302-ARDZ Output @160 Degrees phase difference @2KHz

As can be seen by comparing each graph to its corresponding oscilloscope output the board is not functioning correctly at 2KHz despite the data sheet specifying it works at low frequency.

To paint a clear picture of how the board is behaving, all the graphs are combined and compared against a calculated ideal value.

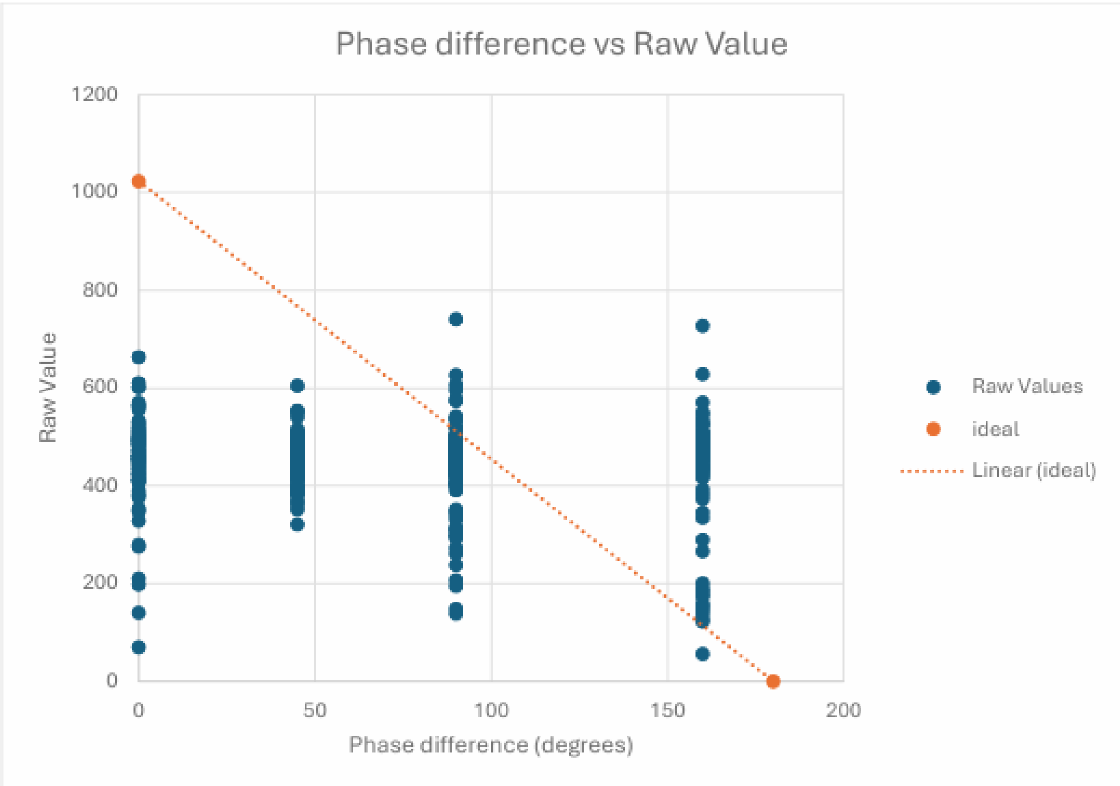


Figure 68 - Phase Difference vs Raw Value for each Phase interval

As can be seen in the graph most values do not fall on the orange ideal line. This means that the board cannot consistently or accurately detect phase difference at 2KHz frequency.

6.3.3 Testing Synched Outputs at 3MHz

After witnessing the poor performance of the phase detector at low frequency, it was decided to test the board in the MHz range to see if it would yield any better results. A shorter test at 3 intervals for 3MHz was conducted.

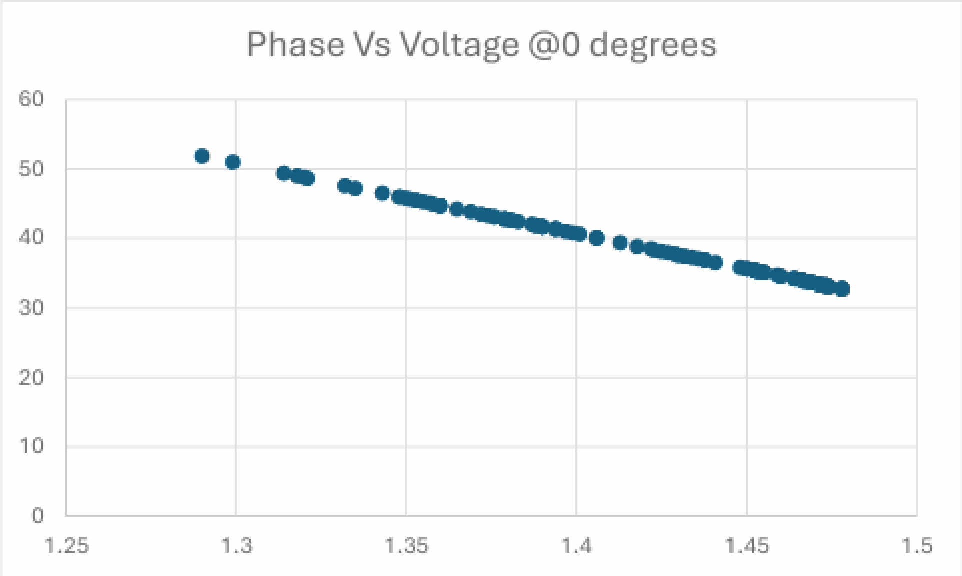


Figure 69 - EVAL-AD8302-ARDZ Output @0 Degrees phase difference @3MHz

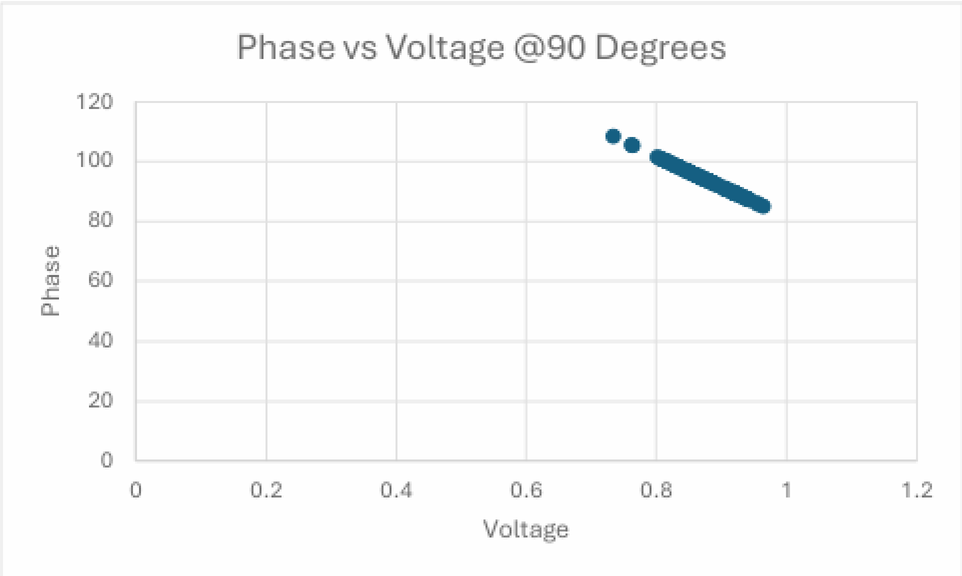


Figure 70 - EVAL-AD8302-ARDZ Output @90 Degrees phase difference @3MHz

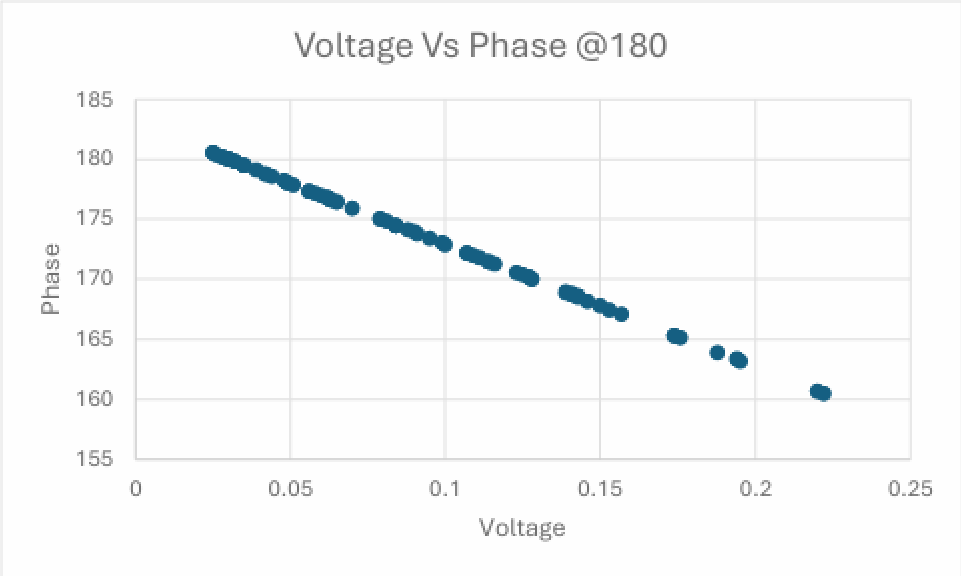


Figure 71 - EVAL-AD8302-ARDZ Output @180 Degrees phase difference @3MHz

For each test the results were compared against the oscilloscope's output and against the ideal value.

Combining each graph again to compare it to the ideal value paints a clearer picture.

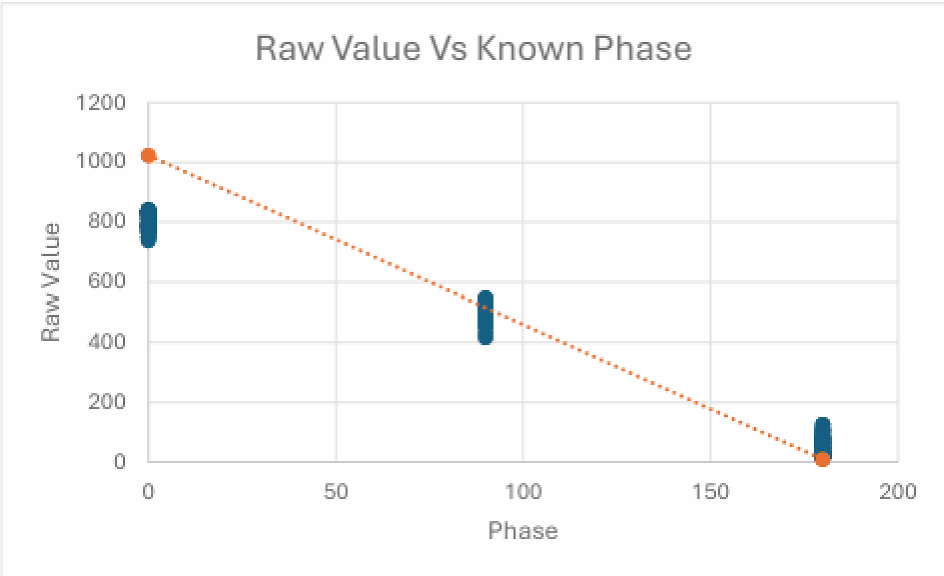


Figure 72 - Raw Value Vs Known phase @3MHz

As can be seen in the graph the board is much more accurate at 90-degree phase difference and at 180-degree phase difference when compared to the 2KHz test.

There is an offset at 0 but this is as expected according to AD8302 datasheet.

6.3.4 Testing Synched Outputs @4MHz

After witnessing better results testing at 3MHz, it was decided to do a more comprehensive test at 4MHz using several intervals of controlled phase.

- Signal Generators Synced @4MHz via Burst mode.
- 22 Degree intervals of controlled phase from 0 -176 degrees
- 1180 data points of Phase vs Raw Value
- Operates correctly when compared to data sheet for AD8302.
- Some fluctuations in the "controlled phase" due to lag on triggering and noise at 4MHz.

Figure 73 – Phase set to 176 degrees @4MHz

Figure 74- Phase set to 90 degrees @4MHz

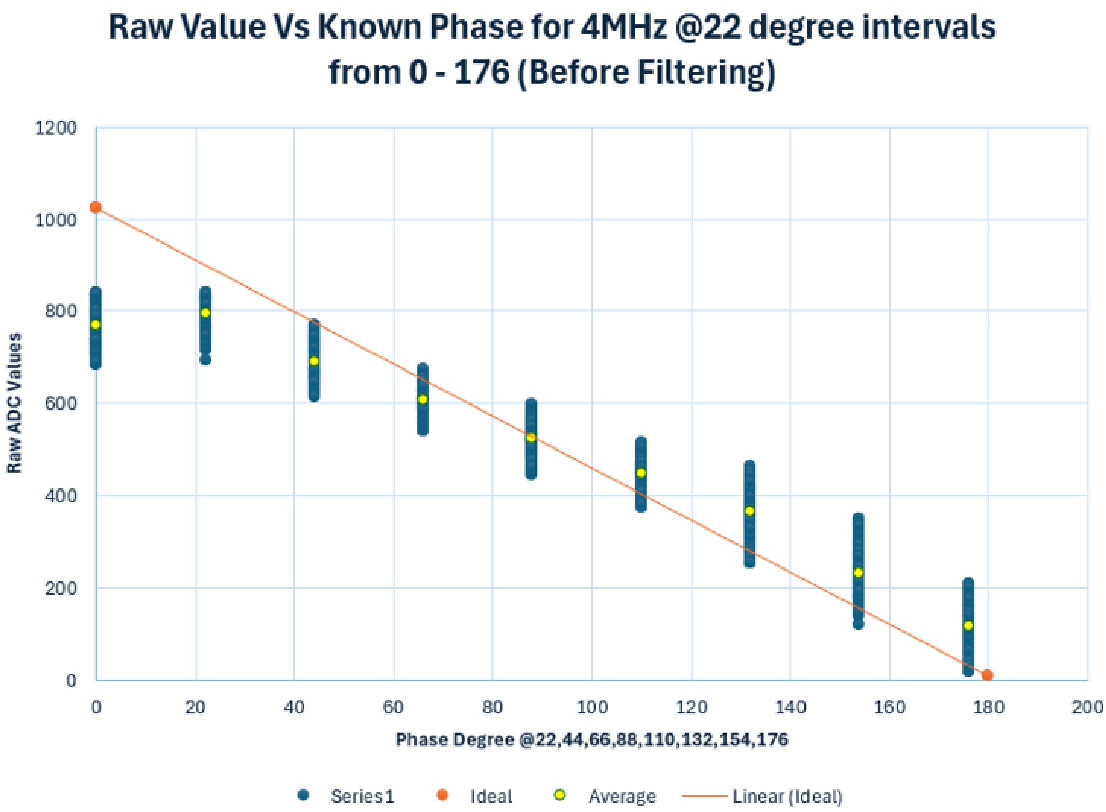


Figure 75 - Raw Value Vs Known Phase @4MHz

As can be seen, the phase detector board operates sufficiently at 4MHz and operates particularly well at 90 degrees phase difference.

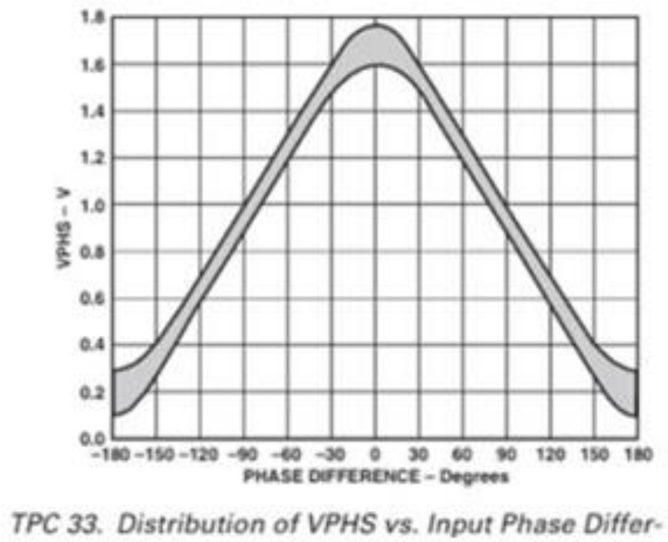


Figure 76 - Phase Behaviur outlined in DataSheet

The AD8302 datasheet indicated the board is operating near to its optimal performance. The above graph shows the difference in input and output results which heavily aligns with what has been seen during 3MHz and 4MHz testing. The board is mostly accurate but varies as the values approach 0.

To conclude the testing of the EVAL-AD8302-ARDZ, the board is operating sufficiently at the 3-4MHz range and is ready to be introduced to the full circuit.

6.4 Testing the Phase Shifter Circuit

After constructing the phase shifter circuit with the previously calculated resistor and capacitor values, the team tested the phase shifter to validate its functionality.

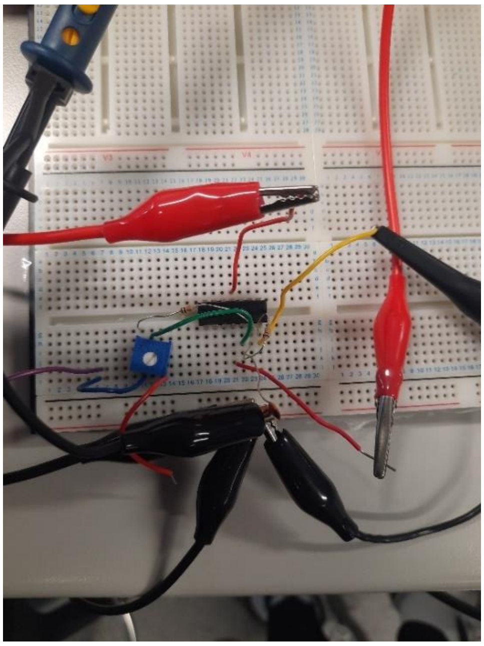


Figure 77- Test setup of phase shifter

As the variable resistor was adjusted during testing, the team observed the input signal gradually shifting towards the desired 90-degree phase shift. With each adjustment, the phase shift became more pronounced, closely approaching the target. Upon reaching the resistor's capacity, a complete 90-degree phase shift was achieved.

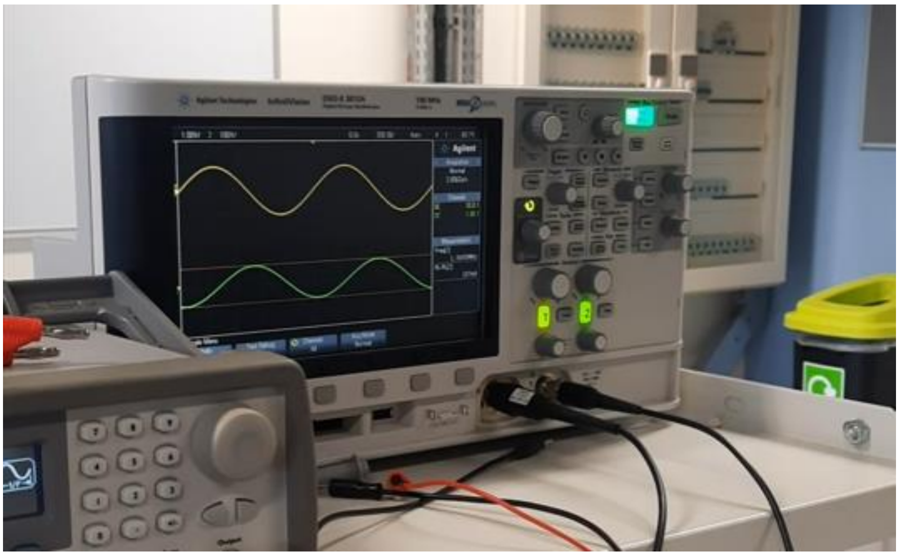


Figure 78- Input signal shifted 90 degrees

In conclusion, the phase shifter performed as intended, demonstrating its capability to achieve a precise 90-degree phase shift during testing.

6.5 Testing the Coils

After the coils were correctly constructed and their frequencies verified on an ENA, the team began testing their behaviour. A notable shift was observed when the sensor coil was introduced to the reader coil. It was also observed that as the spacing between the coils was reduced, the inductance value changed which induced a greater overall phase shift. The biggest shift was observed by squeezing the reader coil and sensor coil together, effectively acting as a makeshift pressure sensor.

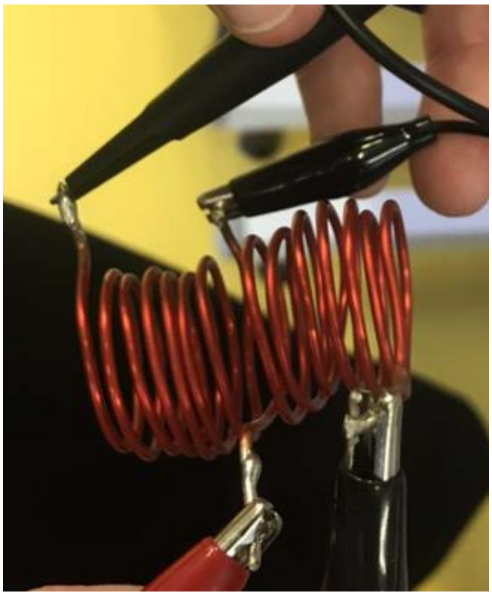


Figure 79- Sensor and Reader coil test setup

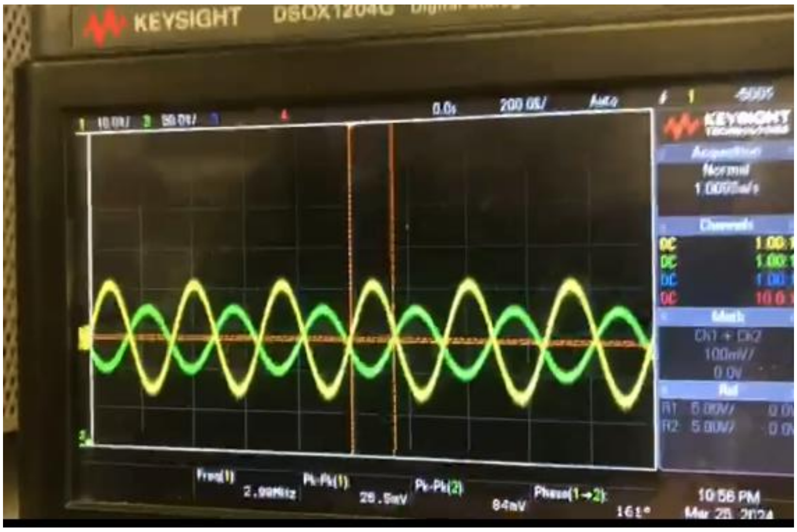


Figure 80- Observed interaction between Sensor and Reader coil

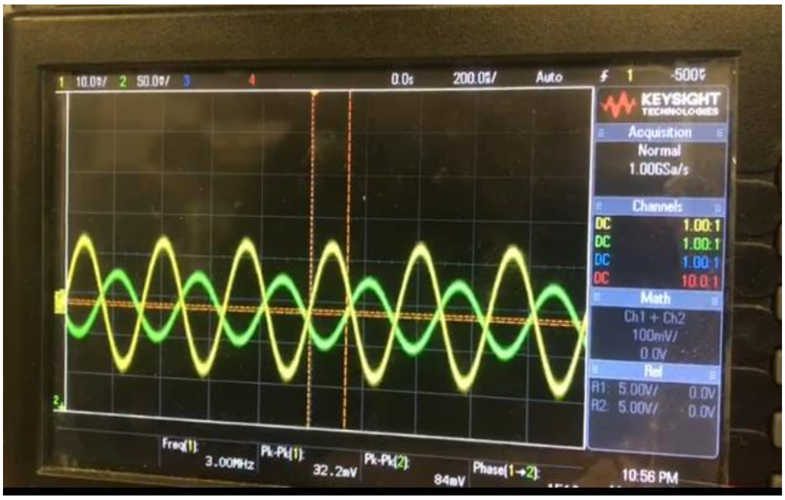


Figure 81- Observed interaction between Sensor and Reader coil squeezed together

To conclude, coil testing provided valuable insights. Changes in inductance and phase shift were observed during interaction, with squeezing them together yielding the most significant shift, similar to a pressure sensor.

6.6 Full System Test

After all the separate components were tested in an isolated and controlled manner to verify their operation, the full system was able to be tested. All the components were introduced together apart from the DDS at first. This was to verify the systems operation using a controlled bench top signal generator.

After verifying the systems operation, the DDS was introduced instead of the signal generator for testing. Some of the circuit was improved by shortening connections and soldering some of the permanent connections.

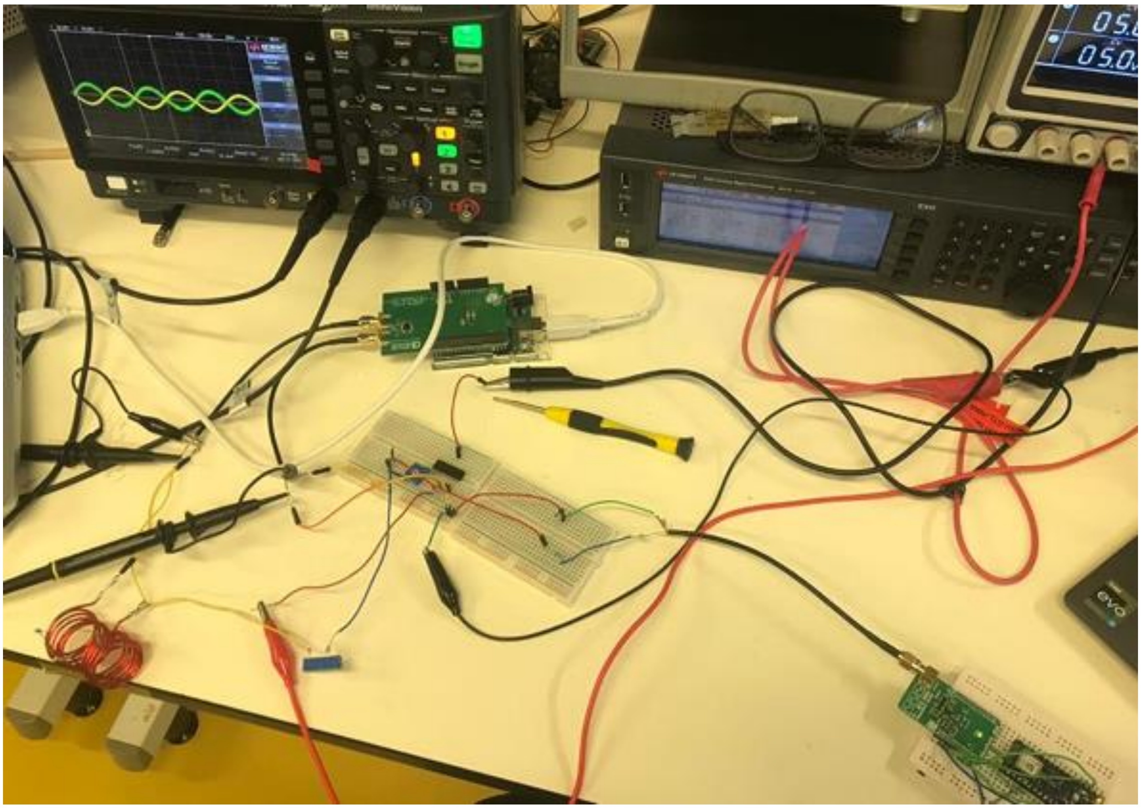
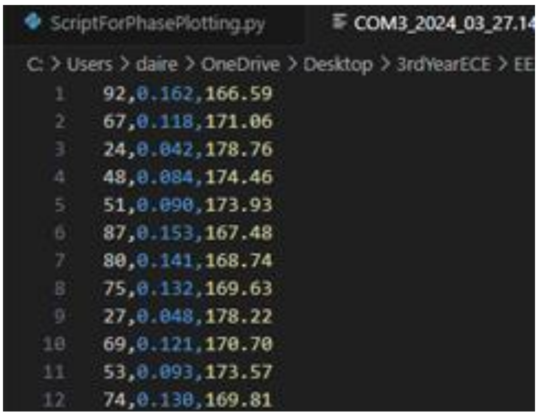


Figure 82 - Full System Test

To conduct a full system, test the circuit is connected and the power supply to the phase shifter is switched on to deliver ± 5 volts. The DDS and phase detector then have their code uploaded and are initialised. The circuit is probed at different points to ensure it is operating correctly. The signals into ports J1 and J2 were constantly monitored on the oscilloscope to verify the readings on the Arduino Serial monitor. VS Code is opened on the pc and the serial port begins listening. Then the sensor coil is introduced to the reader coil. The data is then analysed and presented in a graph using the python automation method previously mentioned.



ScriptForPhasePlotting.py COM3_2024_03_27.14

C: > Users > daire > OneDrive > Desktop > 3rdYearECE > EE

```
92,0.162,166.59
67,0.118,171.06
24,0.042,178.76
48,0.084,174.46
51,0.090,173.93
87,0.153,167.48
80,0.141,168.74
75,0.132,169.63
27,0.048,178.22
69,0.121,170.70
53,0.093,173.57
74,0.130,169.81
```

Figure 83 - Serial Readings in CSV format on VS code log file of COM3

The data is collected and can be analysed rapidly by running the python scripts on the log files.

In this example case the waves were initially set to be 176 degrees out of phase to establish a baseline for reading.

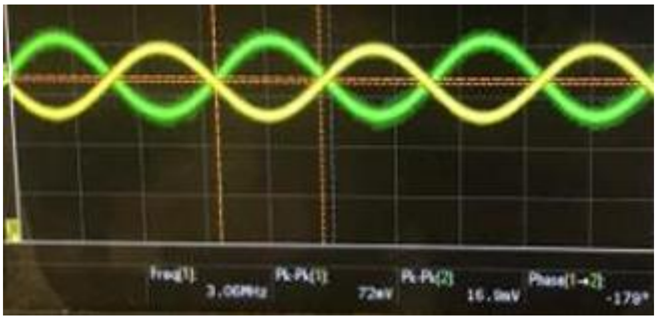


Figure 84 - J1 and J2 signals set to 176 phase difference before Sensor Coil introduced

The sensor coil is then introduced to the circuit and the change in signal is phase is both observed on the scope as well as analysed by the circuit and post processing.

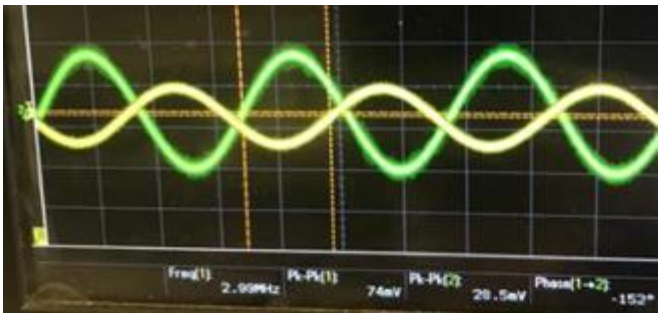


Figure 85 - Oscilloscope after Sensor coil is introduced

As can be seen by comparing the before and after of the oscilloscope readings, the phase difference changes from 176 degrees to 152 degrees. This is a change in phase difference of 24 degrees due to an energy transfer taking place between the coils causing a change in their capacitance and subsequently causing a change in amplitude and phase. When the python scripts are run on the results a clear graph is produced.

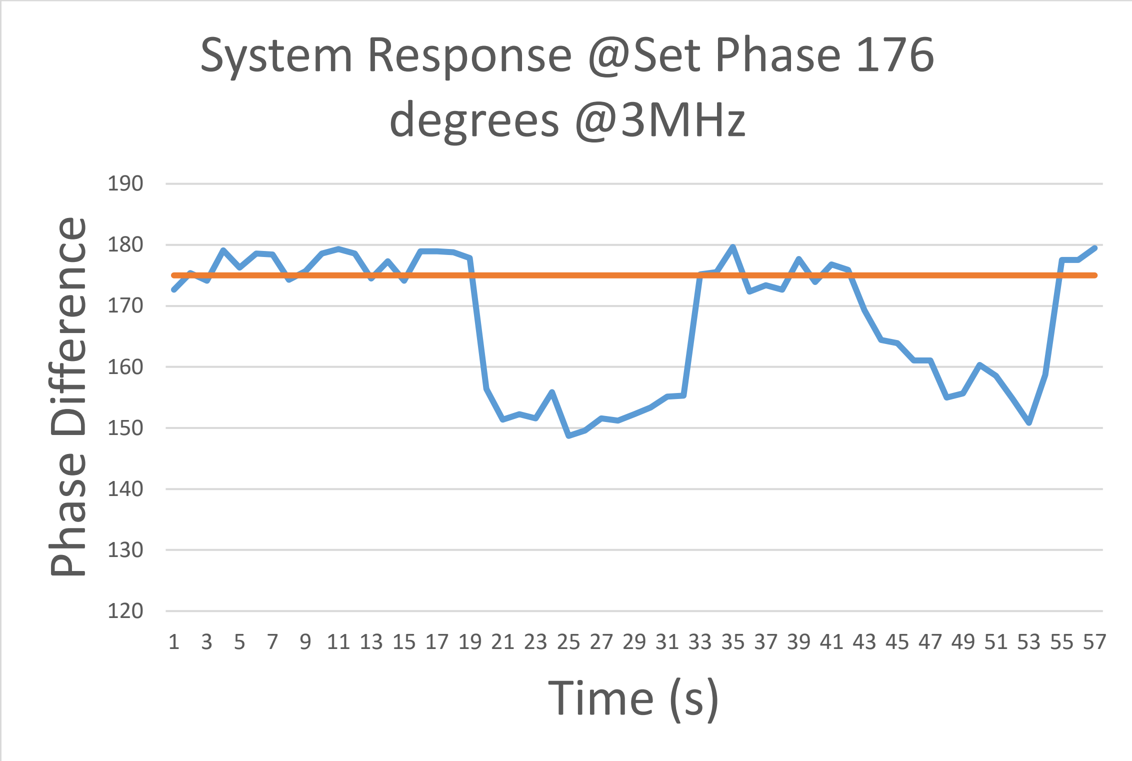


Figure 86 - System Response at 3MHz

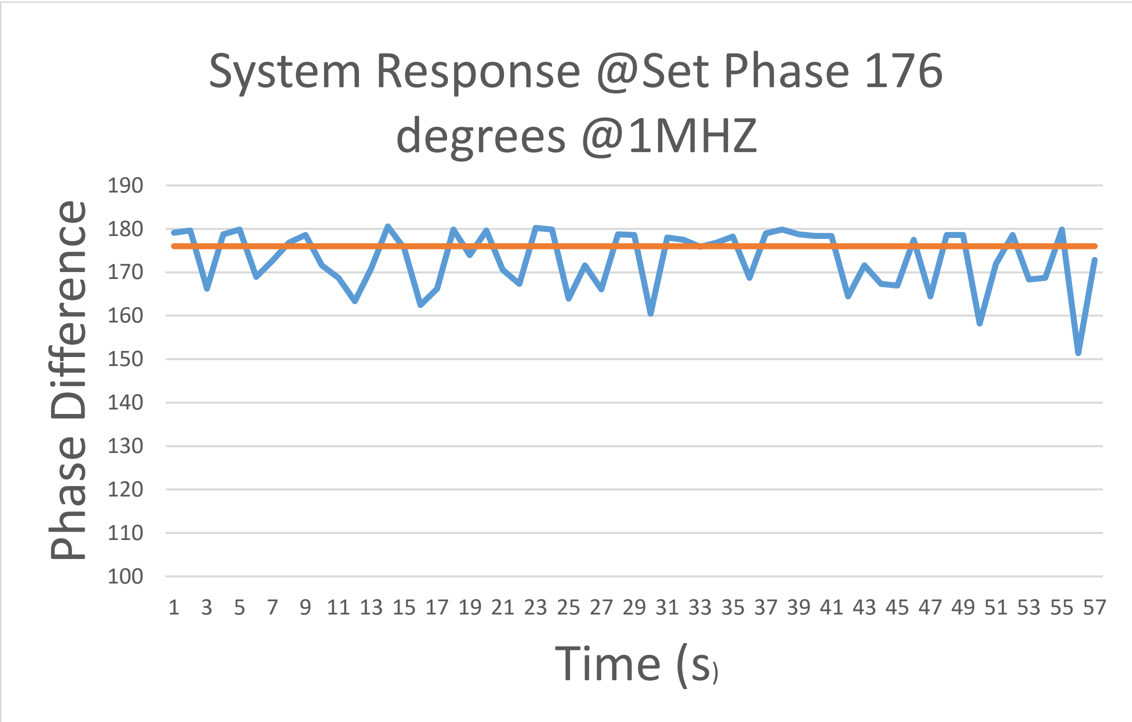


Figure 87 - System Response @1MHz

Chapter 7: Conclusions and Future Work

7.1 Introduction

This chapter discusses some of the key finding and results of the read-out system which was developed as well as giving future recommendations for improving the system.

7.2 Key Findings and Results

From studying figure 77 and figure 78 conclusions can be drawn on how the system performed. Figure 77, which details the system response operating at 3MHz and set to a baseline phase indicates that a significant shift in phase occurs when the two coils interface together. The phase changes from176 degrees to as low as 149 degrees. This shift confirms the concept outlined previously which states that the using analogue signal processing the change in pressure data can be interpreted and analysed.

Contrasting this with the system performance at 1MHz (figure 78), indicates that the systems performance deteriorates as the control sine signal is moved away from the resonant frequency of the coils. As the 1MHz signal is used to probe the reader coil the change in phase is much less apparent. This highlights the need for the systems signal used to be aligned with the resonant frequency of the coil.

7.3 Future Work

Despite the success of the prototype system, there are areas for improvement that future work could address to enhance the system's performance and utility further:

1. **Component Standardization:** Future iterations could focus on standardizing components to reduce variability in performance, especially at different frequencies. This would help in maintaining consistency across different units.
2. **Coil Design Optimization:** Redesigning the inductive coils to maximize energy transfer and efficiency could further enhance the system's sensitivity and accuracy.

3. **Advanced Signal Processing:** Implementing more sophisticated filtering and signal processing algorithms could help in reducing noise and improving the signal quality, especially in complex clinical environments.
4. **Scalability and Integration:** Efforts could be made to miniaturize the system and integrate it with other medical devices and systems, broadening its application scope across various medical monitoring tasks.
5. **User Interface Development:** Developing a more intuitive user interface for the system could facilitate easier interaction for engineers to test the system efficiently.

By addressing these aspects, the next phases of this project can refine the technology to ensure that it can be deployed effectively in clinical settings. This project lays a robust foundation for the future exploration and integration of wireless read out technologies in sensor technologies, with the potential to revolutionize how patient care is administered in various medical fields.

7.4 Reflection

7.4.1 Daniel Scully

In my role for the project, I focused on the background research, design, construction, and testing of both the phase shifter and the reader and sensor coils. I gained valuable insight from troubleshooting the circuit, troubleshooting helped me learn from mistakes I made and learn new skills as well. I thought that working in the Lambe Institute for the duration of the project was a great experience. Overall, I enjoyed the project and learnt from my experiences throughout its course.

7.4.2 Daire Shankey

During the project I primarily focused on investigating, developing, and implementing the DDS waveform generator, Low pass filter, Phase detector and connecting the full circuit. I also created software libraries and documentation for the boards we used which is available on my personal GitHub page as well as creating an automated process for collecting data and processing it from the system.

Most of my time was spent troubleshooting and debugging due to a lack of documentation and the challenges surrounding interfacing different components. This project was a huge learning experience for me as it involved a lot of up skilling for much of the work which had to be completed. I really feel having come through this project that I am much better equipped now to approach engineering problems and tackle them using my experience from the project. It was also a great opportunity to be able to work in the lab at the Lambe Institute where it was possible to learn from very experienced post-doctoral researchers.

References

[1]	M. S. Z. M. A. C. M. M. M. H. N. H. M. A. I. M. H. M. Md. Imran Hossain, “Science Direct,” 24 12 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211715623005313. [Accessed 6 1 2024].
[2]	J. V. P. K. J. L. Timo Salpavaara, “sciencedirect,” 5 3 2011. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0924424711001245#preview-section-abstract. [Accessed 24 9 2023].
[3]	G. Poulos, “www.allthescience.org,” 13 12 2023. [Online]. Available: https://www.allthescience.org/what-is-inductive-coupling.htm?utm_content=cmp-true. [Accessed 15 10 2023].
[4]	“Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Resonant_inductive_coupling#:~:text=The%20resonant%20inductive%20coupling%20is,frequency%20as%20the%20driving%20frequency..
[5]	K. &. C. J.-S. &. Y. K. Lum, “Wireless power transfer framework for minirobot based on resonant inductive coupling and impedance matching,” *International Journal of Power Electronics and Drive System (IJPEDS)* , vol. 11, no. 1, pp. 317-325, 2020.
[6]	S. Gallagher, “allaboutcircuits.com,” 3 May 2019. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/understanding-phase-shift-in-analog-circuits/.
[7]	ee-diary, “What is phase shifter circuit and how to build it?,” ee-diary, 23 March 2023. [Online]. Available: https://www.ee-diary.com/2023/03/what-is-phase-shifter-circuit-and-how.html. [Accessed 22 February 2024].
[8]	M. I. J. W. M. R. S. M. I. Mackenzie Cook, “A Highly Compact 2.4GHz Passive 6-bit Phase Shifter with Ambidextrous Quadrant Selector,” 2015. [Online].

	Available:https://www.skyworksinc.com/-/media/SkyWorks/Documents/Articles/IEEE_042016.pdf. [Accessed 7 May 2024].
[9]	Matsusada Precision, “Matsusada Precision,” 10 November 2021. [Online]. Available: https://www.matsusada.com/column/words_inductor.html. [Accessed 8 May 2024].
[10]	George Brown College, “etcourse.com,” 15 October 2020. [Online]. Available: https://www.etcourse.com/news-blog/air-gap-in-magnetic-circuits#:~:text=The%20amount%20of%20air%20or,expand%20outside%20the%20magnetic%20circuit.. [Accessed 8 May 2024].
[11]	HBR Industries, “www.hbrindustries.com,” [Online]. Available: https://www.hbrindustries.com/what-are-the-best-applications-of-an-inductor-coil/. [Accessed 8 May 2024].
[12]	Analog Devices. (n.d.) *EVAL-AD5932EBZ User Guide (UG-796).* [Online] Available at: https://www.analog.com/en/resources/evaluation-hardware-and-software/evaluation-boards-kits/eval-ad5932.html#eb-documentation (Accessed: 10 May 2024).
[13]	MikroElektronika. (n.d.) *Clock Gen Click.* [Online] Available at: https://www.mikroe.com/clock-gen-click (Accessed: 10 May 2024).
[14]	MikroElektronika. (n.d.) *Waveform Click.* [Online] Available at: https://www.mikroe.com/waveform-click (Accessed: 10 May 2024).
[15]	Seams, J. (2009) 'Resistive voltage dividers', *Electronic Products.* [Online] Available at: https://www.electronicproducts.com/resistive-voltage-dividers/ (Accessed: 10 May 2024).
[16]	Analog Devices. (n.d.) *AD9833: Low Power, 12.65 mW, 2.3 V to 5.5 V, Programmable Waveform Generator* [Data sheet]. Available at: https://www.analog.com/media/en/technical-documentation/data-sheets/ad9833.pdf (Accessed: 10 May 2024).
[17]	Circuit Cellar Staff. (2014) 'DDS Basics (EE Tip #122)', *Circuit Cellar.* [Online] Available at: https://www.circuitcellar.com/resources/ee-tips/dds-basics-ee-tip-122/ (Accessed: 10 May 2024).
[18]	Analog Devices. (n.d.) *AD5227: 64-Position Up/Down Control Digital Potentiometer* [Data sheet]. Available at:

	https://www.analog.com/media/en/technical-documentation/data-sheets/AD5227.pdf (Accessed: 10 May 2024).
[19]	Back, C. (2017) 'Design Calculations and Testing of Basic Filters', *RS DesignSpark.* [Online] Available at: https://www.rs-online.com/designspark/design-calculations-and-testing-of-basic-filters (Accessed: 10 May 2024).
[20]	U.S. Army Aberdeen Test Center. (2020) *Test Operations Procedure (TOP) 02-2-632 Data Acquisition System (DAS).* Aberdeen Proving Ground, MD: Survivability Lethality Directorate. Available at: https://apps.dtic.mil/sti/trecms/pdf/AD1118068.pdf (Accessed: 10 May 2024).
[21]	Analog Devices. (n.d.) *ADB302: LF–2.7 GHz RF/IF Gain and Phase Detector* [Data sheet]. Available at: https://www.analog.com/media/en/technical-documentation/data-sheets/ad8302.pdf (Accessed: 10 May 2024).
[22]	Analog Devices, approved by Getz, R. (2021) *User Guide for the EVAL-AD8302 Evaluation Board.* [Online] Available at: https://wiki.analog.com/resources/eval/user-guides/eval-ad8302-ardz (Accessed: 10 May 2024).
[23]	Arduino. (n.d.) *Arduino Uno Rev3.* [Online] Available at: https://store.arduino.cc/products/arduino-uno-rev3 (Accessed: 10 May 2024).
[24]	Arduino. (n.d.) *Arduino Nano 33 BLE.* [Online] Available at: https://store.arduino.cc/products/arduino-nano-33-ble (Accessed: 10 May 2024).
[25]	Cian, D. (2021) 'How does a unity gain buffer work?', *Electrical Engineering Stack Exchange.* [Online] Available at: https://electronics.stackexchange.com/questions/594794/how-does-a-unity-gain-buffer-work (Accessed: 10 May 2024).

Appendix A: Software Library for EVAL-AD8302-ARDZ

```
const int VPHASE_PIN = A1;
const float referenceVoltage = 1.8;
const unsigned long duration = 5000;
const int VMAG_PIN = A2;

void setup() {
  pinMode(VPHASE_PIN, INPUT);
  pinMode(VMAG_PIN, INPUT);
  Serial.begin(115200);
}

void loop() {
  int rawValue = analogRead(VPHASE_PIN);
  float voltage = rawValue * (referenceVoltage / 1023.0);

  // Normalize the phase difference between 0 and 180 degrees
  //float phaseDifference = map(voltage, 0, 1.8, 180, 0);  // Adjusted for cyclical behavior
  float phaseDifference = 180.0 - (voltage - 0.03) / (1.8 - 0.03) * 180.0;

  // Handle cyclical behavior for phase difference
  if (voltage > 1.8) {
    phaseDifference = 0.0000000;

  }

  //Serial.print("Raw Value: ");
  //Serial.print(rawValue);
  //Serial.print(" | Voltage: ");
  //Serial.print(voltage, 3);

  //Serial.print(" V | Phase Difference: ");
  //Serial.print(phaseDifference, 2);
  //Serial.println(" degrees");

Serial.print(rawValue);
Serial.print(",");
Serial.print(voltage, 3);
Serial.print(",");
Serial.print(phaseDifference, 2);
Serial.println();



  delay(1000);}
```

Appendix B: Software Library for Waveform Click 3MHz output with Amplitude Control

```
#include <SPI.h>//spi communication

// Define pins and constants
const int FSYNC_PIN = 10; // AD9833 FSYNC defined as pin 10 on Arduino
const int CS_PIN_AD5227 = 2; // AD5227 Chip Select defined as pin 2 on Arduino
const double MASTER_CLK = 25e6; // Master clock frequency for AD9833

class AD9833SineWave {
public:
   explicit AD9833SineWave(int fsyncPin) : fsyncPin(fsyncPin) { //constructor for sine wave class
      pinMode(fsyncPin, OUTPUT);
      digitalWrite(fsyncPin, HIGH); // De-select the AD9833 initially
   }

   void begin() {
      SPI.begin();
      SPI.beginTransaction(SPISettings(2000000, MSBFIRST, SPI_MODE2)); // Lower SPI speed for stability
      writeRegister(0x2100); // Reset the AD9833
      delay(10); // Wait for the reset to take effect
      SPI.endTransaction();
   }

   void setFrequency(double frequency) {
      unsigned int freqValue = static_cast<unsigned int>((frequency * (1ULL << 28)) / MASTER_CLK);
      unsigned int LSB = freqValue & 0x3FFF; // Lower 14 bits
      unsigned int MSB = (freqValue >> 14) & 0x3FFF; // Upper 14 bits

      SPI.beginTransaction(SPISettings(2000000, MSBFIRST, SPI_MODE2));
      writeRegister(0x2100); // Reset command to ensure clean start
      writeRegister(0x4000 | LSB); // Set the lower part of the frequency
      writeRegister(0x4000 | MSB); // Set the upper part of the frequency
      writeRegister(0xC000); // Default phase register (optional, for phase adjustment if needed)
      writeRegister(0x2000); // Exit from Reset
      SPI.endTransaction();
   }

private:
   int fsyncPin;

   void writeRegister(unsigned int data) {
      digitalWrite(fsyncPin, LOW); // Enable communication with AD9833
      SPI.transfer(highByte(data)); // Send higher 8 bits
```

```
    SPI.transfer(lowByte(data));  // Send lower 8 bits
    digitalWrite(fsyncPin, HIGH); // Finish communication
  }
};

AD9833SineWave sineWave(FSYNC_PIN); // Create an AD9833 control object

void setup() {
  Serial.begin(9600); // Start serial for debugging
  pinMode(CS_PIN_AD5227, OUTPUT);
  digitalWrite(CS_PIN_AD5227, HIGH); // Ensure AD5227 is deselected

  sineWave.begin(); // Initialize the AD9833 module
  sineWave.setFrequency(3000000); // Set output frequency to 3 MHz

  // Set AD5227 to min resistance to achieve maximum amplitude
  setAD5227Position(63); // Set to 0 for minimum resistance and for max amplitude
}

void loop() {
  // The waveform is continuously output at the set frequency and amplitude.
  delay(1000); // Placeholder delay
}

void setAD5227Position(int position) {
  position = constrain(position, 0, 63); // Ensure the position is within the valid range
  SPI.beginTransaction(SPISettings(2000000, MSBFIRST, SPI_MODE2));
  digitalWrite(CS_PIN_AD5227, LOW); // Select the AD5227
   SPI.transfer(position << 2); // Send the position with alignment to the MSBs for
AD5227
  digitalWrite(CS_PIN_AD5227, HIGH); // Deselect AD5227
  SPI.endTransaction();
}
```

Appendix C: Automation Python Scripts for Analysis

```
import csv
import os
import pandas as pd
import matplotlib.pyplot as plt

# Function to plot the data
def plot_data(csv_file_path):
   # Load data
   df = pd.read_csv(csv_file_path)

   # Check if the required columns exist in the DataFrame
    if 'Time(s)' in df.columns and 'Phase Difference' in df.columns and 'Set Phase' in
df.columns:
       time = df['Time(s)']
       phase_difference = df['Phase Difference']
       set_phase = df['Set Phase']

       # Creating the plot with two series
       plt.figure(figsize=(10, 5))  # Adjust size as needed
       plt.plot(time, phase_difference, color='magenta', label='Phase Difference')
           plt.plot(time, [set_phase.iloc[0]] * len(time), color='cyan', label='Set Phase',
linestyle='--')  # Assuming 'Set Phase' is constant

       # Set titles and labels
       plt.title('Phase Response of System')
       plt.xlabel('Time (s)')
       plt.ylabel('Phase Difference (Degrees)')

       # Show legend
       plt.legend()

       # Show grid
       plt.grid(True)

       # Display the plot
       plt.show()
   else:
       print(f"Required columns are not present in {csv_file_path}")

# The directory containing the .txt files
directory_path = r"C:\Users\daire\OneDrive\Desktop\3rdYearECE\EE3126 3rd Year
Project Semester (1 and 2)\SystemResultslog"

# Loop through all files in the directory
for filename in os.listdir(directory_path):
   #comment
   # Check if the file is a .txt file
   if filename.endswith('.txt'):
```

```
        txt_file_path = os.path.join(directory_path, filename)
        # Create a .csv filename based on the .txt filename
        csv_file_path = os.path.join(directory_path, f"{os.path.splitext(filename)[0]}.csv")

        # Open the .txt file and create a .csv file
        with open(txt_file_path, 'r') as infile, open(csv_file_path, 'w', newline='') as outfile:
            csv_writer = csv.writer(outfile)
            # Write header with the specified headings
            csv_writer.writerow(['A', 'Time(s)', 'Phase Difference', 'Set Phase'])
            # Initialize a counter for the time values
            time_counter = 1
            # Initialize a counter for the rows to add 'Set Phase' value
            row_count = 0
            for line in infile:
                row_count += 1
                # Split the line by comma and strip whitespace
                row_data = line.strip().split(',')
                # Replace the second value with the time counter value
                row_data[1] = str(time_counter)
                # If there are not enough columns, add the 'Set Phase' column
                if len(row_data) < 4:
                    row_data.append('176' if row_count <= 200 else '')  # Add 'Set Phase' only
for the first 200 rows
                # Increment the time counter
                time_counter += 1
                csv_writer.writerow(row_data)

        # After writing the CSV file, call the plot function
        plot_data(csv_file_path)

print("Conversion, modification, and plotting complete!")
```