Scanning Array For EMI Detection

Worcester Polytechnic Institute

Department of Electrical and Computer Engineering

Major Qualifying Project

A Major Qualifying Project
Submitted to the Facility of
Worcester Polytechnic Institute
In partial fulfillment of the requirements for the
Degree of Bachelor of Science
Submitted 4/25/2024

Authors:
Sarah Chen
Michael Iberger

Advisors:
Dr. Gregory Noetscher
Dr. Sergey Makaroff

Table of Contents

Acknowledgements

We would like to express our gratitude to the engineers, Eric Craft and Greg Fitzgerald, from Teradyne for their aid and support throughout the project. We would also like to thank our professors and advisors Professor Noetscher, Professor Makaroff, and Doctor Serano for their excellent advising and guidance during this undertaking. Additionally, we would like to thank William Appleyard of ECE shop for helping us acquire materials and components for this project.

Abstract

Electromagnetic Interference (EMI) is a significant side effect of designing electronics. EMI scanners are used to measure the magnitude of the emitted radiation and ensure that these emissions do not negatively affect neighboring electronics. The current EMI scanner used by Teradyne is too small for efficient testing. To improve the efficiency and testing process of the large PCBs at Teradyne, we created a prototype scanner with larger H field loop probes. Future designs and iterations of this prototype work toward making an enlarged scanning unit that would increase efficiency of Printed Circuit Board (PCB) emission tests at Teradyne.

Executive Summary

Teradyne performs radiation emission tests on the large scale PCBs that they manufacture. However, due to the large size of the PCBs, scanning the emissions with the available equipment is tedious and inefficient. The current EMI scanner (EMScanneR by Y.I.C. Technologies) used to perform these tests has dimensions of 8.5” x 12.5”, however the engineers need to scan an area of 25” x 25”. As a result, the scanner needs to be moved six times to fully scan the entirety of the board’s surface emissions.

Teradyne hopes to develop a solution that would optimize the current scanning process and eliminate the constant supervision necessary from the test engineers. We brainstormed two possible solutions. The first option was to create a larger version of the current scanner. The design of this scanner would be the same as that of the Y.I.C. Scanner, however due to the enlarging of the current design, the frequency range of the probes would decrease with the increase in size. The second option consisted of creating a mechanical apparatus that would manually move the current scanner and replace the intervention from the test engineers. Comparing the two solutions, we determined that since our project is more ECE focused rather than robotics guided, we decided to take the first approach and design a larger version of the current scanner.

Our design is a prototype board made of 25 loop probes in a 5x5 array. We designed the board in HFSS, then we ran simulations of the model to validate the behavior of the design. Once we determined a working model, we imported the geometry into Altium and created a PCB design consisting of an eight layer board. We also created an arduino script that would circle through the loops to imitate the scanning mechanism of the scanner. We have not been able to test the board as of yet as the board has not been printed.

Currently our prototype design is still in the beginning stages of development and serves as proof of concept for creating a larger design of the current scanner used by Teradyne to measure EMI radiation from large PCBs under test.

Table of Figures

Table of Tables

1 Introduction

Teradyne is an automated test equipment design company. Their process for designing deliverables includes manufacturing and then validation of the products before launching them onto the market. One of the many validation tests performed at Teradyne includes Printed Circuit Board (PCB) emission tests. One of their main products consists of several large PCBs that are housed next to each other in close proximity. In order for the equipment to work, the PCBs must function simultaneously without emitting too much radiation, or absorbing too much of the emitted energy. Teradyne runs tests on their PCBs to measure the emission levels of each individual board to ensure that the radiation is within the acceptable level of standards.

Currently, Teradyne uses an Electromagnetic Interference (EMI) scanner from Y.I.C. Technologies called the EMScanneR to measure the magnitude and determine the location of the emissions. The scanner consists of 1,218 H field loop probes spaced every 7.5 mm on the board [1]. The dimensions of the board consist of a length of 12.44” and a width of 8.58”. The frequency of the scanner ranges from 150 kHz to 8 GHz. Below Figure 1 shows an image of the current scanner.

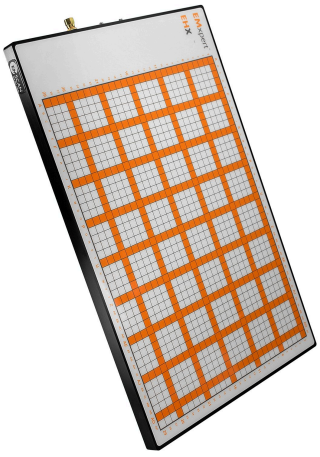

Figure 1. The EMScanneR from YIC Technologies

The scanner works in conjunction with the EMxpert adapter, EMxpert application software, and a spectrum analyzer to fully operate. The laptop with the application software connects to the adapter with a USB-A connector, and the laptop also connects to the Spectrum Analyzer with an Ethernet cable (Figure 2). Because the scanner is the main hardware interacting with the PCBs under test, this device is the primary focus of this project [2].

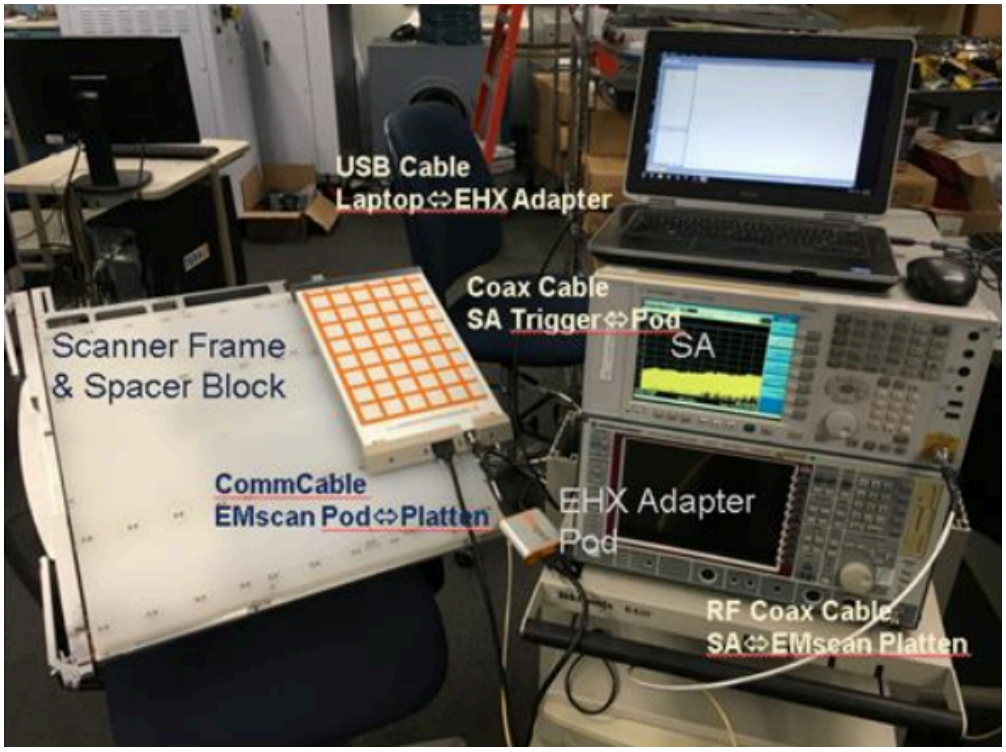

Figure 2. Current Scanning Array Setup at Teradyne

2 Background

2.1 Problem Statement

Electromagnetic radiation is emitted from almost every modern electronic device as Radio Frequency (RF) energy. RF energy shares a common trait even with some of the most dangerous electromagnetic waves; it is outside the spectrum of visible light that humans can see. Although it is not an inherently dangerous form of radiation, the unintended consequences from stray RF energy can easily be detrimental. Excess RF energy can couple with other components on a Printed Circuit Board (PCB) causing them to have a lower signal to noise ratio and may diminish the functionality of the device. In a more extreme case, unregulated, stray RF energy can interfere with an electronic implant like a pacemaker. For these reasons the Federal Communications Commission (FCC) has regulations in place to restrict the overabundance of radiated emissions and maintain the quality of the human environment [3]. For PCB manufacturers it is critical to have a scanner that can efficiently measure and pinpoint the source of these radiated emissions to ensure the product functions within regulation, properly, and safely.

The current EMI scanner used by Teradyne is able to capture a heatmap of radiated emissions from an adjacent PCB, however it is an inefficient means of doing so. Although the scanner provides consistency and reliability of the measurements throughout a wide frequency band (10 MHz-8 GHz), it has several downsides. The process is repetitive and requires manual intervention, making it very inefficient. The dimensions of the scanner are significantly smaller than that of the PCBs under test. Due to the size mismatch, the radiated emissions from the adjacent PCB cannot be fully measured in one round of measurements; the scanner must be

moved and dismounted repeatedly in order to fully measure the entire radiated energy of the device. The entire process to scan a PCB requires a minimum of six rounds of measurement: moving, mounting, and dismounting the current EMI scan unit to fully test one side of the PCB.

2.2 Current Solutions

Y.I.C. Technologies manufactures and sells a product called an EMScannerR that Teradyne uses for PCB emission testing. This scanner is produced in one size with dimensions of 12.44” x 8.58”. Due to the limited size of the current scanner, other approaches to measuring the PCB H field emissions were investigated. We considered three approaches. The first consisted of moving the existing scanner around with some sort of mechanical instrument, the second was making a larger version of the current scanner, and the third option was creating an entirely new approach to scan and detect PCB emissions. The first option of creating a mechanical robot to move the existing scanner was not an ideal approach. We determined this option would create several new challenges. One issue would be ensuring the scanner operated in unison with the moving mechanism. We foresaw the complications of coordinating the data measurements with the position of the scanner. Another issue we anticipated was mechanical wear. Introducing moving parts, the users of the device would experience reduced lifespan of the instrument due to mechanical failure. These challenges, time constraints, and the team's prior experience with Ansys HFSS suggested that making a larger version of the instrument was the most viable option. Rather than pursuing the option to use another scanner, it was determined through cost, benefit, and time analysis that the most efficient approach would be to work toward constructing a larger version of the current EMI scanning unit.

2.3 Our Solution

As Teradyne wishes to maintain the current scanner as the means of measuring the radiation emissions of the PCB boards, multiple solutions have been presented to improve the current status of measuring the radiated energy. Based on the current configuration of the scanner and the corresponding software, it was determined that the most efficient solution would be to create a larger EMI scanner with the same number of H field loop probes. The targeted dimensions of the new scanner would be 25”x 25”. The length would be about double in comparison to the original board and the width would be almost three times larger. Additionally, the new board would be square shaped rather than rectangular like the current scanner. A square shaped scanner was found to be the most efficient option because it would cover significantly more, if not all of the area of the PCB under test.

Since the proposed new design of the board consists of the same number of loop probes in an increased area, the size of the loop probes will increase. As a consequence, the resonance frequency will decrease and the frequency range of the loop will be reduced. This theory is illustrated in the antenna length formula. The relationship states that wavelength and frequency have an inverse relationship, thus as wavelength increases the frequency decreases and vice versa. This is shown in equation 1.

$$\lambda = \frac{v}{f}$$

Equation 1.

Teradyne’s requirements for the new scanner requested that the frequency range cover 10 kHz - 1 GHz, thus eliminating the L-band (1-2 GHz) , S-band (2-4 GHz), and C-band (4-8 GHz) frequencies. This requirement allowed the design of the new scanner to be simpler as the higher

end of the frequency range is in the lower end of Ultra High Frequency (UHF). Additionally, the new scanner would be large enough to cover the entire Device Under Test (DUT), thus satisfying the 25"x 25" requirement. Our design must also satisfy the RadEM requirements as seen below in Figure 3. The major objectives of this new scanner would be to limit manual intervention of the scanning procedure with the current EMI device, and also to reduce the time necessary to complete a full scan: from three days to one day.

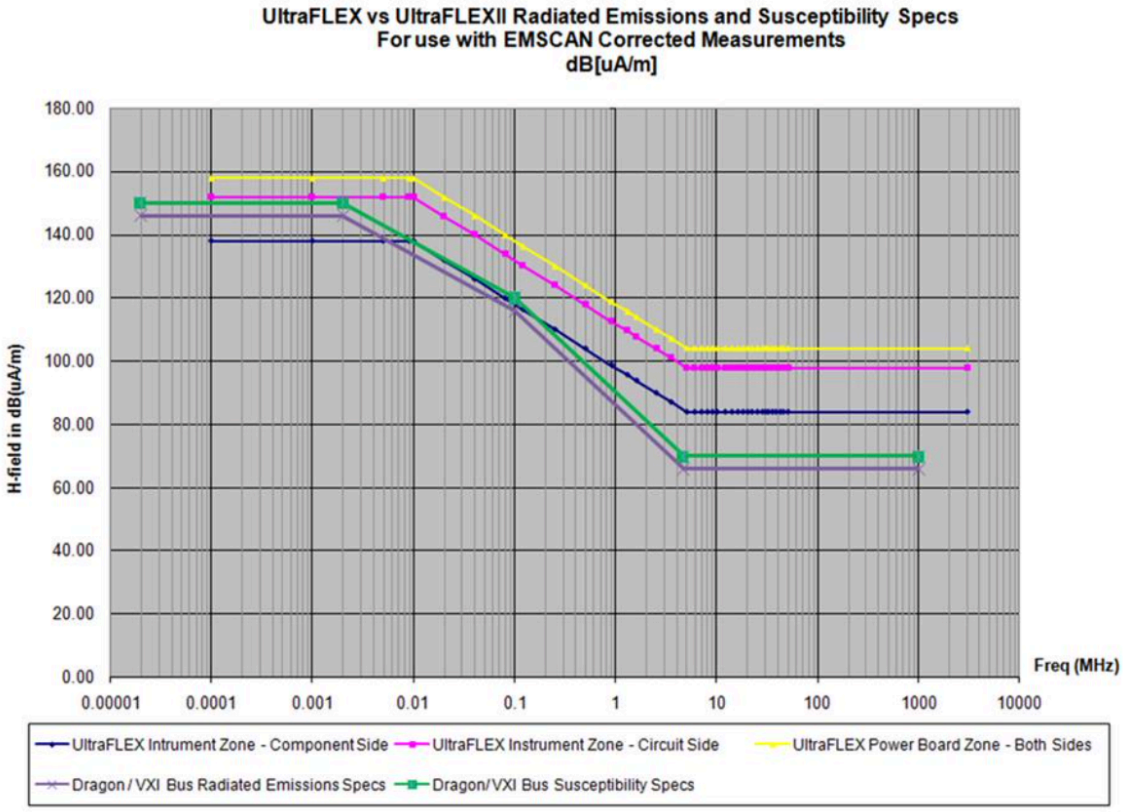

Figure 3. RadEM Requirements

Due to the project's time constraints, the final deliverable for the end of the 21 week period was downscaled to be a 5x5 loop array consisting of 25 loops. This 5x5 prototype of the scanner would still maintain the desired attributes of the 1,218 loop array but it would be on a

significantly smaller scale. Teradyne determined this to be a more feasible scope of the project and aspires to improve upon the design to implement all 1,218 loops in the future.

2.3.1 Scanner Design

2.3.1.1 Antenna Theory

We started researching loop probes to understand the functionality of EMI scanners. Our initial investigation included research into the difference between loop antennas versus dipole antennas. From our research, we determined that from the perspective of whistler modes, loop antennas are superior to dipole antennas. Whistler modes are low frequency electromagnetic waves which are often present in experiments with electromagnetic dipole antennas. Additionally, the antenna currents and field waves exceed those of dipole antennas by twice the magnitude; however dipole antennas have smaller radiation resistance[4]. Thus we determined that for our application, a loop probe would be the antenna type that we would use for our design. Magnetic loop probes measure the voltage drop across conductors and planes, and the current flowing in conductors and magnetic fields. Loop probes are commonly made from coaxial cable. Loop probes consist of a coil that is coupled with an emitting wire and shielded by a coax cable [5]. These loop probes have two primary geometric configurations: circular shaped and rectangular shaped. Figures 4 and 5 are examples of common circular and square shaped loop probes.

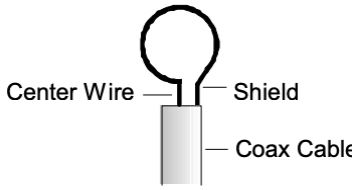

Figure 4. Example of Circular Loop Probe

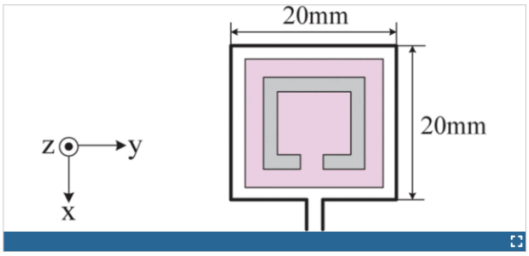

Figure 5. Example of Square-shaped Loop Probe

The round loop provides a uniform magnetic field without dead spots at corners or sharp angles. The rectangular loop is simpler to fabricate and it is more suitable for scanning flat structures such as PCBs and integrated circuits/packages [6]. However, due to the nature of the circular loop probe providing a more uniform H field, we decided to go with this design.

The current design of the EMI scanner consists of 1,218 probes. H field loop probes respond mainly to fast changes in current, and the measured value depends on the orientation of the probe's tip [7]. Rotation of the tip of the probe relative to the DUT will result in different results. For instance, a loop probe measuring with the probe vertical to the surface will produce different results to a horizontal orientation of the probe, or an orientation at 180 degrees. The maximum output measured by the probe occurs when the loop is aligned parallel (horizontally to the board) to the current carrying wire. Figure 6 demonstrates this principle.

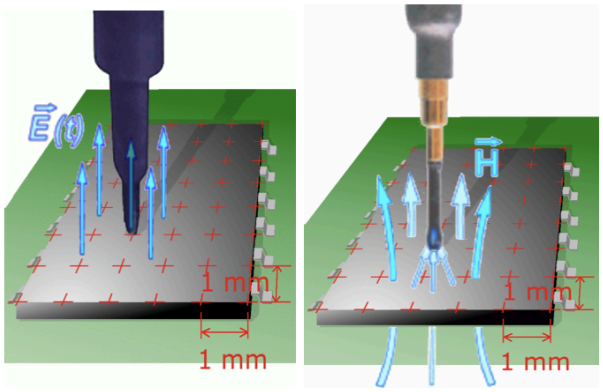

Figure 6. Probe's Horizontal Orientation to Measure Vertical Emissions

In our research, we investigated frequency responses for varying loop sizes. The diameter of the loop determines several critical factors when measuring: the frequency response, the sensitivity of the probe, and the ability to detect sources of emission. Table 1 shows the corresponding resonance frequency as the loop diameter decreases.

	Primary Sensor Type	E/H or H/E Rejection	Upper Resonant Frequency
901 6-cm loop	H-Field	41 dB	790 MHz
902 3-cm loop	H-Field	29 dB	1.5 GHz
903 1-cm loop	H-Field	11 dB	2.3 GHz
904 3.6-cm ball	E-Field	30 dB	>1 GHz
905 6-mm stub tip	E-Field	30 dB	>3 GHz

Table 1. Loop Diameters and Corresponding Resonance Frequencies

Source: Adapted from [8]

The larger the diameter of the loop, the more magnetic field flux lines the loop will cross and the better the inductive coupling will be with the offending wire (wire that introduces unwanted electrical signals). This results in higher sensitivity of the larger loop. Figure 7 shows how the frequency response changes with different loop diameter sizes. As the loop diameter decreases, the resonance frequency of the loop increases. The first resonance occurs at the frequency where the circumference of the loop is one half wavelength. Probes with a smaller loop diameter have a reduced non-ideality. From our research, it was determined that the loop diameter of interest would be small enough to have a resonance frequency that occurs higher than the highest frequency in the frequency band of interest: 1 GHz.

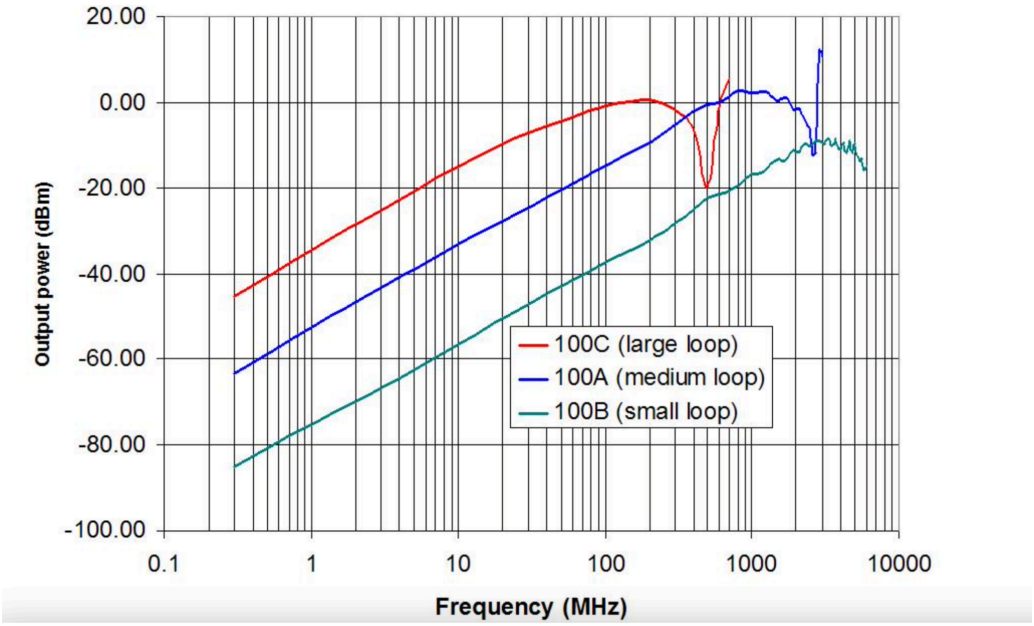

Figure 7. Frequency Response of Different Loop Diameters

2.3.1.2 Lumped Element Model

During our research, we started with a lumped element model of an H field loop probe. We discovered how the equivalent circuit for calculating the loop impedance can be represented as a series Resistor, Inductor, and Capacitor (RLC) circuit as shown in Figure 8 and Equation 2 From these approximations of the lumped element's transfer function, we determined a model to follow in order to replicate the frequency response of the probe for our design [9].

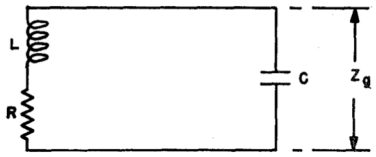

Figure 8. Lumped Element Model of Loop Probe

Source: Adapted from [10]

The equation for loop impedance is

$$Z_g = \frac{R + jw\left[L(1 - w^2 LC) - CR^2\right]}{(1 - w^2 LC)^2 + w^2 C^2 R^2},$$

where

Z_g = loop impedance;
L = inductance of the loop, henries;
C = capacity of the loop, farads;
R = resistance of the loop, ohms; and
w = $2\pi f$ where f = frequency, cycles.

Equation 2.

2.3.1.3 Signal Routing and Multiplexers

We investigated several approaches to integrate and route all of the loops together into a singular output. Below is a rough schematic that shows how the antenna elements are connected on the EMScanneR board that Teradyne uses (Figure 9).

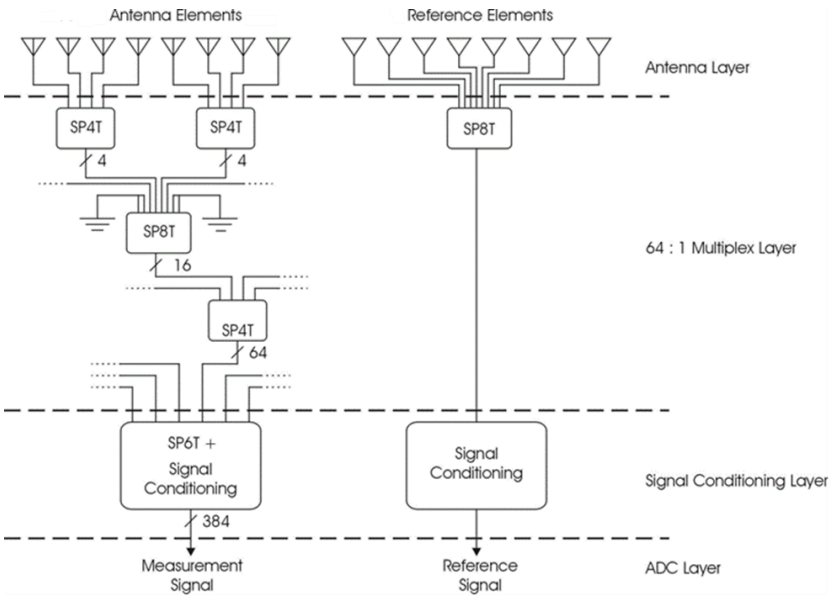

Figure 9. Routing Structure for Antenna Elements in EMScanneR

Our original understanding of the scanner was that there was an array of antenna elements being activated in a sequence to generate a heatmap of the radiated emissions in each cell. The routing structure above provided us with the groundwork for how we would move forward with the project and how we could use multiplexers to switch between the desired antenna elements. The current board uses a combination of SP4T (4:1) and SP8T (8:1) multiplexers to route all of the antennas to the measurement signal output.

Multiplexers are integrated circuits which have an extensive amount of practical applications. Multiplexers function by taking in several inputs, in our case they are RF inputs, and routing them to a singular output pin. Multiplexers are controlled by a digital signal control structure written in binary; these digital control signals are routed to the control input pins on the multiplexer. The high and low voltages which are sent to the control input pins represent “1” and “0” respectively. The values received at each control input pin correspond to a decimal number that tells the multiplexer which of the desired antennas should be routed to the output pin. For

example if an SP8T multiplexer has three control inputs there are 2^3=8 possible combinations of “0” and “1”; each of the combinations is unique and allows the multiplexer to identify one of the eight inputs that must be routed to the output.

Figure 10 below shows how the RF analyzer, laptop, controller, and scan plate work in conjunction with one another. The computer is essentially the brain of the system by operating everything. Running a scan from the computer simultaneously sends commands to the controller and the RF analyzer. The controller sends binary signals from the computer to the scan plate telling it to activate the first of the selected cells; at the same time the RF analyzer at the top of the diagram performs the specified frequency sweep at that cell. The data is sent back to the computer, readings are saved for that cell, and the process repeats for the next cell in the sequence.

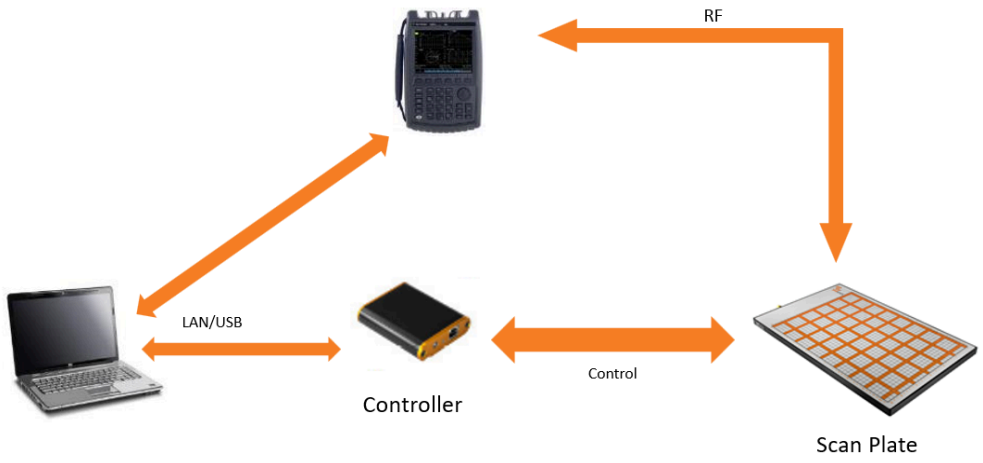

Figure 10. Control and Signal Output Paths for EMScanneR

2.3.1.4 RF Impedance Matching

RF impedance matching is a fundamental part of RF PCB design. Impedance matching involves using the dielectric constant (ε_r) and the dielectric thickness to create transmission lines that are the correct width to have a 50 ohm impedance.

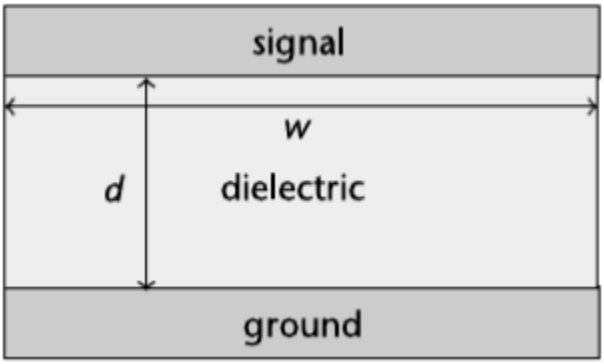

Figure 11. Simplified Transmission Line Cross-Section

Source: Adapted from [11]

It is such an important part of designing RF PCBs because traces that are the wrong impedance will create a mismatch and diminish power at the load and cause reflections. A mismatch will happen when you terminate a transmission line at a much lower or much higher impedance than its characteristic impedance [11]. This can introduce reflections and severely impact the signal being sent to the load. These reflections are essentially unabsorbed energy that has the ability to interact with the original signal by adding or subtracting energy from it [12]. In our case, recording accurate radiated emissions is critical to the success of the project and a sizable impedance mismatch would render the prototype useless.

2.3.1.5 Different RF PCB Traces

Although an impedance matched line is probably the most important factor for maintaining signal integrity, choosing the correct topology for a transmission line ensures that the signal will remain on its intended path free from external interference. Two types of transmission line structures that we researched at the start of the project were microstrip lines and coplanar waveguides (Figure 12).

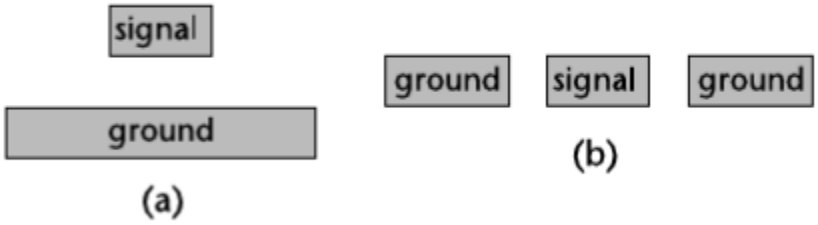

Figure 12. Cross Sections of a Microstrip Line (a) and a Coplanar Waveguide (b)

Source: Adapted from [11]

We determined that these topologies wouldn't be ideal for our design because they do not provide much shielding around the inner conductor. Radiated emissions from the environment and other parts of the circuit can easily couple to the microstrip line from the sides and the top. Similarly, the coplanar waveguide only provides shielding on the sides leaving room for energy to propagate on the top and bottom. For these reasons, we decided to move forward with a third type of topology called the grounded coplanar waveguide (Figure 13).

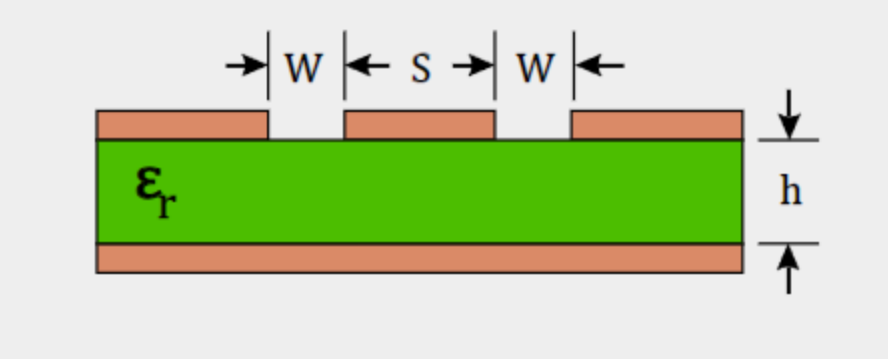

Figure 13. Cross Section of a Grounded Coplanar Waveguide

Source: Adapted from [13]

We believe the grounded coplanar waveguide topology is superior to the other two because it provides shielding on the bottom and both sides of the signal trace. Some grounded coplanar waveguides provide shielding on the top of the signal trace as well, however they must be embedded fully in the board and that may introduce new layout challenges. These waveguides in our design run on the RF layer of our PCB and are used to route all of the multiplexer inputs and outputs to their proper location.

3 Methodology

3.1 Basic Theory Behind the Design

To fundamentally understand the concepts behind the frequency response of the loop probe, we first made a lumped element modeling consisting of a first order system modeling a capacitor. With every increase in the number of energy storage components, this corresponds to an increase in the number order of the system: represented in (jω) terms. Thus to keep the system relatively simple, we kept the model with only one capacitor and one inductor. We then created a second order system model in MATLAB with capacitor and inductor circuit components to replicate the transfer function that we found in our research by altering the capacitor and inductor values. Through changing the resistor, inductor, and capacitor values, we were able to observe how the frequency response changed by changing the component values. Figure 14 shown below shows the results of the transfer function. As we discovered in our research, the frequency response of the resonance circuit we made in MATLAB demonstrated the same behavior as shown in Figure 7. As the size of the loop decreased, the resonance frequency increased, thus proving our hypothesis that the frequency response could also be modeled as a second order RLC circuit.

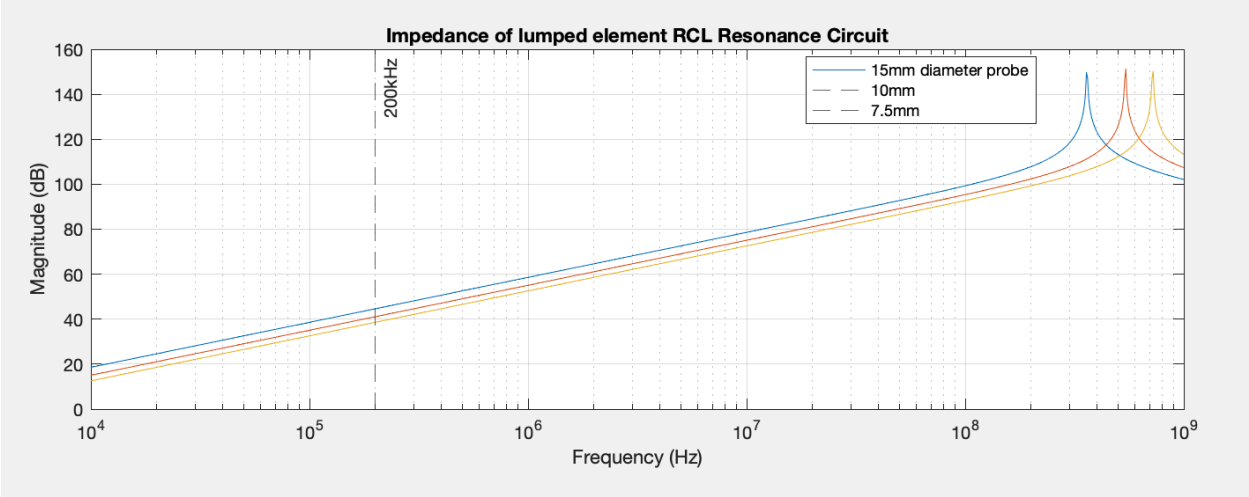

Figure 14. Frequency Response of the RLC Resonance Circuit

From the resonance circuit we created in MATLAB, we determined that the larger the diameter of the loop probe, the lower the resonance frequency. To transfer the concept behind this modeled RLC circuit in MATLAB to a physical design to model in HFSS, we then researched H field loop probe geometries. As a geometric model to reference in the HFSS design, we found in our research a model providing the dimensions and geometry for an H field loop probe. Through the analytical equations for the parallel plate capacitor, solenoid inductor, and resistor, we were able to compute approximations for the RLC values of the loop. We used basic solenoid inductor and resistor equations to find the inductor value for L and the resistor value for R. The capacitor value was more complicated to compute due to a lack of explicit parallel plates in the loop schematic. However, the loop has self capacitance which we were able to use to compute the capacitance. The equation for the self inductance of a solenoid of cross sectional area A, number of turns in the coil N, and length l is shown in Equation 3 The equation of capacitance between two conducting plates that we used to find the capacitance of the loop in terms of permittivity of the medium ε , the area of the plates A, and the distance between the

plates d is shown in Equation 4. Equation 5 demonstrates the resistance of a cylindrical segment of a conductor that we used to calculate resistance in terms of the resistivity of the material ρ, the length L, and the area A.

$$L = \frac{\mu N^2 A}{l}$$

Equation 3. Self Inductance of a Solenoid

$$C = \frac{\varepsilon A}{d}$$

Equation 4. Equation for Capacitance Between Two Conducting Plates

$$R = \rho \frac{L}{A}$$

Equation 5. Equation for the Resistance of Cylindrical Segment of a Conductor

By representing the loop probe geometry in terms of the lumped element model that we simulated in MATLAB, we were able to see how the RLC circuit corresponds to a physical model of a loop probe. However, to get a more realistic model of the probe with distributed inductances and capacitances, we needed to move the model into HFSS where this software could compute the real world distributed values of the loop probe. To begin this process of transferring the model to HFSS, we researched loop probe geometries and found a probe with a 15 mm diameter loop. With this geometry shown in Figure 15, we then began to imitate the model of this H field loop probe in HFSS [14].

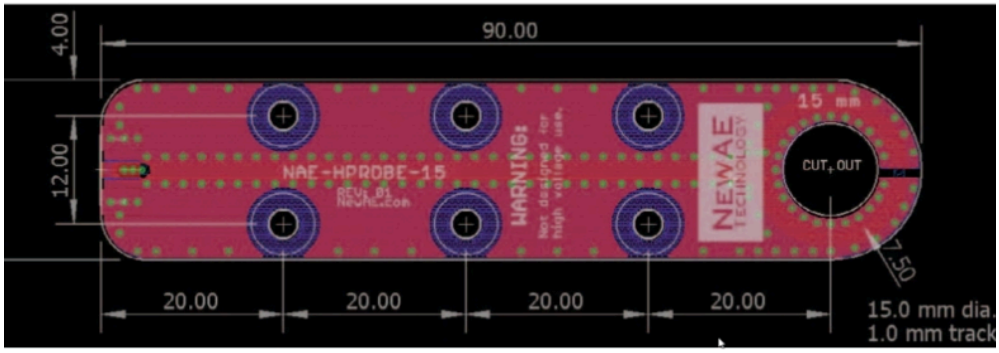

Figure 15. Geometry and Dimensions of H field Probe

3.2 HFSS Model

We chose to use the HFSS software to model and design the structure of the scanner array due to the advanced computational and simulation abilities as a frequency domain solver. Due to the nature of the board as a scanner consisting of H field probes, we could design and model an array in HFSS and simulate the magnetic field emissions of the scanner as an essential part of the design process.

3.2.1 Initial Loop Probe Design

The H field loop probe is the elementary building block of our design. Thus we began our construction of the scanner by first constructing the basic geometry of a loop probe. We first made a single layer loop probe design in HFSS modeled after the geometry of the 15 mm loop probe that we found in our research. This initial design consisted of a loop and feed embedded in a substrate. Figure 16 shown below shows the initial stages of our design with the loop probe and the long connecting loop feed.

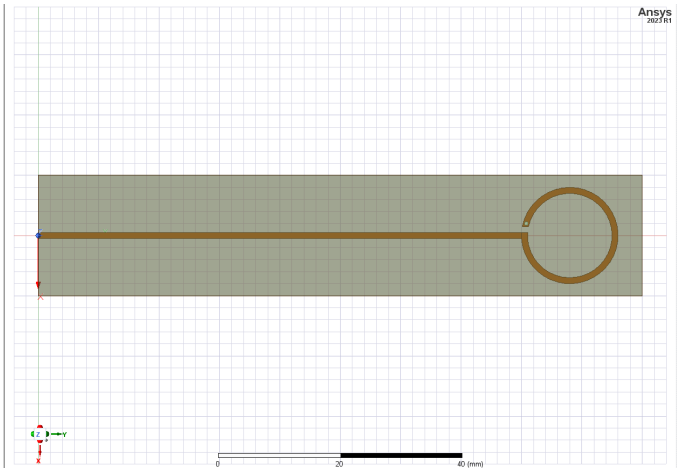

Figure 16. First Design of a Single Loop with Coax Feed

In order to test the functionality of our preliminary design in an array, we modeled the geometry of the single loop and created an array structure consisting of four loops in a 2x2 array as shown in Figure 17 This tested the parameterization of the loops when placed adjacently in an array structure.

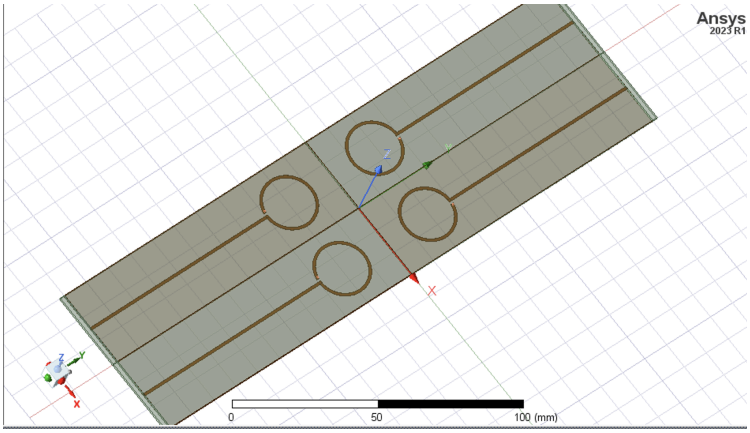

Figure 17. Array of Four Loop Probes

While creating an animation of this array, we found that the loop diameter and feed length did not animate in harmony and quickly identified issues in the initial parameterization. We ran into difficulties with the rectangular shape of the loop shape and substrate. This caused irregularities in the loop design when animating varying loop sizes of the array. We then had to determine which specific variables we would need to parameterize in the model for the array design. They were determined to be the number of loops and the dimensions of the components comprising the array.

3.2.2 Reduced Feed Length Design

To prevent the issues that were appearing with parameterizing different parts of the loop, we decided to decrease the feed length and create a square substrate in which the loop and feed could reside. This design would ameliorate the parameterization difficulties and could be modeled in the layout shown in Figure 18.

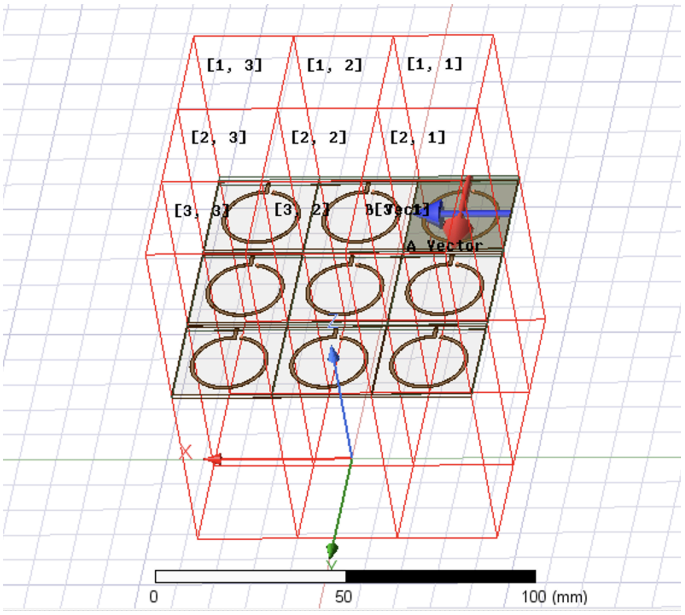

Figure 18. Proposed New Design of the Loop Array

The square shaped substrate in which the loop and feed resided created a more symmetrical architecture to animate and parameterize the variables of interest. We shortened the length of the loop feed significantly and validated the design through creating an animation of the loop sweeping the diameter through a range of values. Figure 19 is a screenshot taken from the animation of varying loop diameters. The success in the integrated geometry of this design was validated through the animation of the varying loop diameters.

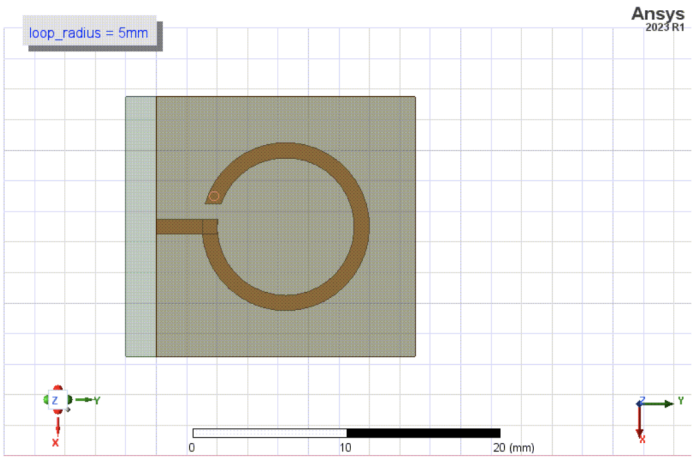

Figure 19. Screenshot of Animating Loop Radius with New Design

With the improvements in this updated design, we began to add more details to the loop architecture. The NAE loop probe, from which we based our design is a four layer PCB containing shielding vias and a main via that carries the signal from the power source to the loop. Our next process was to add layers and vias. Figure 20 shows our next iteration of the loop design. It contained three layers: a top and bottom loop shield, and the center loop probe. We also added shielding vias around the loop and feed as modeled in the NAE loop probe. For placing the vias, we pulled discoveries from our research when considering the layers in which the vias would travel in addition to the symmetry of via placement in enclosing the feed lines

[15]. We also made a cutout to include the loop gap at the top of the loop. We parameterized this model to maintain the geometry with the vias and feed length in a set delta from the substrate edge as the loop size increased.

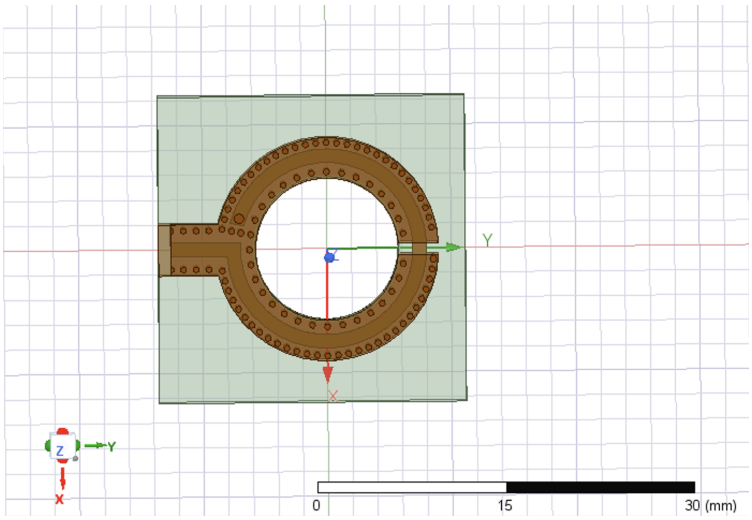

Figure 20. Loop Iteration with Shielding Vias

We then created an array with this updated design and mapped 16 loops into a 4x4 array to simulate the behavior of the probes. Figure 21 shows the results of the H field when all the loops were excited in the array.

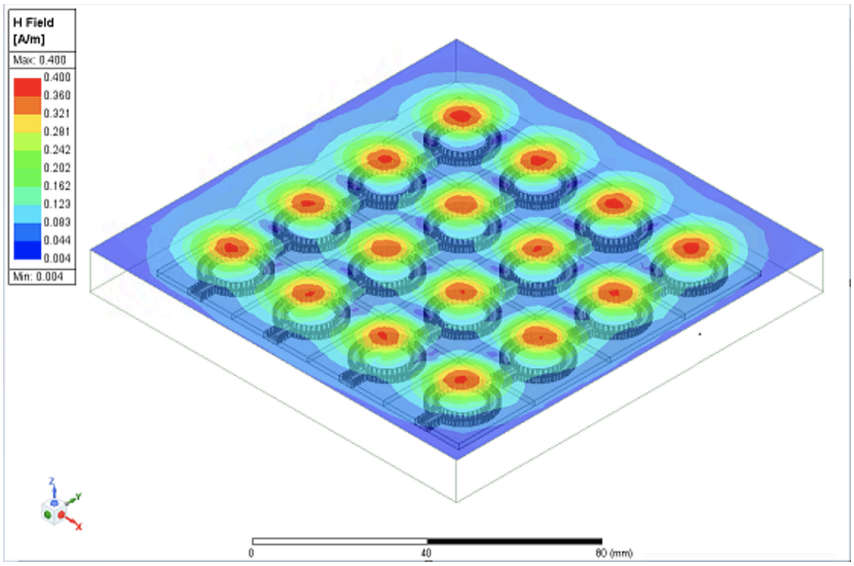

Figure 21. Plotted H Field for Array of Updated Design with Reduced Feed Length

3.2.3 Modified Design with Coax Feed Under the Loop

When we simulated the H field of the array when exciting all of the loops, we observed noticeable dead spots between the loops as shown as the blue regions in Figure 21. We thus determined that these dead spots between the loops were due to signal isolation between the loops. In order to reduce these areas, we thus decided to reduce the amount of substrate surrounding the loop. We also realized that having the feed extend beyond the loop and next to the adjacent loops would cause potential issues with signal interference of the wiring and powering the loop. Our concern was that these signal lines could cause leakage into the surrounding loops. As a result, we decided to change the structure and make the loop feed a coax feed that would extend on the underside of the loop. This would add another layer to the loop in the design, and it would ensure that the signal to power the loop would travel up from the bottom side of the loop instead of from the same level as the loop as in the NAE loop probe design. This updated geometry is shown in Figure 22.

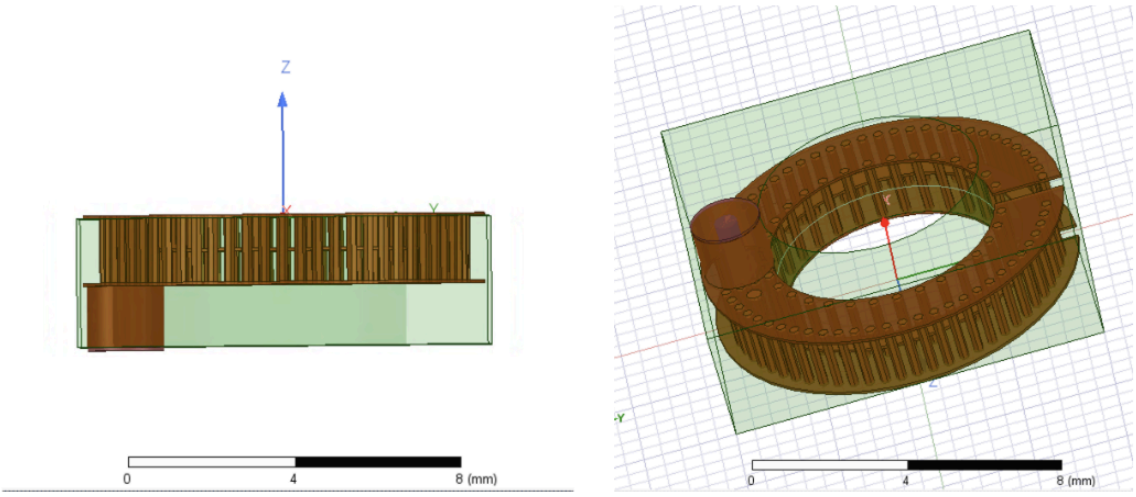

Figure 22. Preliminary Loop Model with Coax Feed Under the Loop

With this modified design of the loop probe, we parameterized the loop diameter and simulated the impedances at different diameter values. Figure A shows the plot of resonance frequencies of the loop at different diameter sizes. Since our frequency range of interest spans 10 kHz-1 GHz, we needed to choose a loop diameter that had a resonance frequency above 1 GHz.

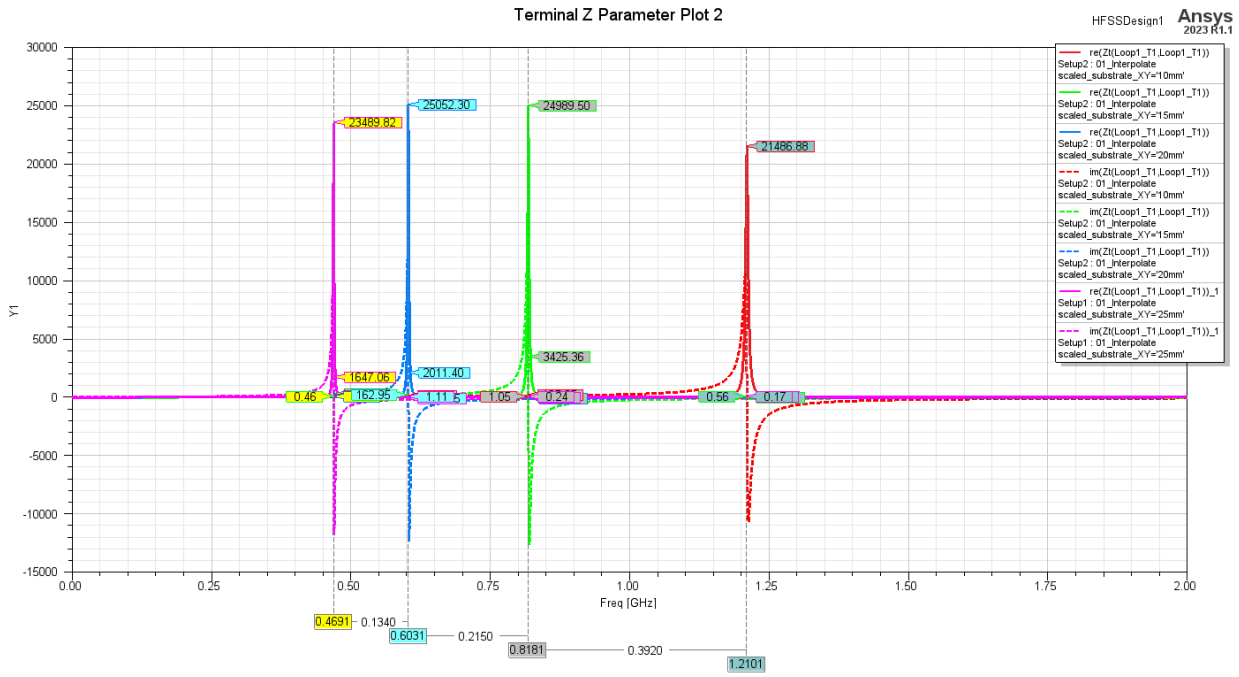

Figure 23. Z Parameter Plot of Different Loop Diameters

In this parameter sweep, we chose diameter values from 5-25 mm with a step of 5. Thus we simulated resonance frequencies at 5 mm, 10 mm, 15 mm, 20 mm, and 25 mm. The red graph shows the resonance frequency of a 10 mm loop, the neon green is 15 mm and the magenta and blue are large diameter sizes but less relevant as they resonate significantly lower than 1 GHz. From these results, we were able to determine that our diameter of interest would be between 10-15 mm. Thus we simulated the Z-parameters again but changed the diameters from 10-15 mm with increments of 1 mm.

We also simulated the H field of the loop from 0-1 GHz to observe the radiation pattern of the loop as the frequency varied through our range of interest. Figure 24 shows a screenshot of the simulation at 0.2 GHz where the radiation pattern is relatively uniform throughout the loop. The image also shows how the signal travels from the large via in the loop through the inner loop and seems to stop before the gap in the loop.

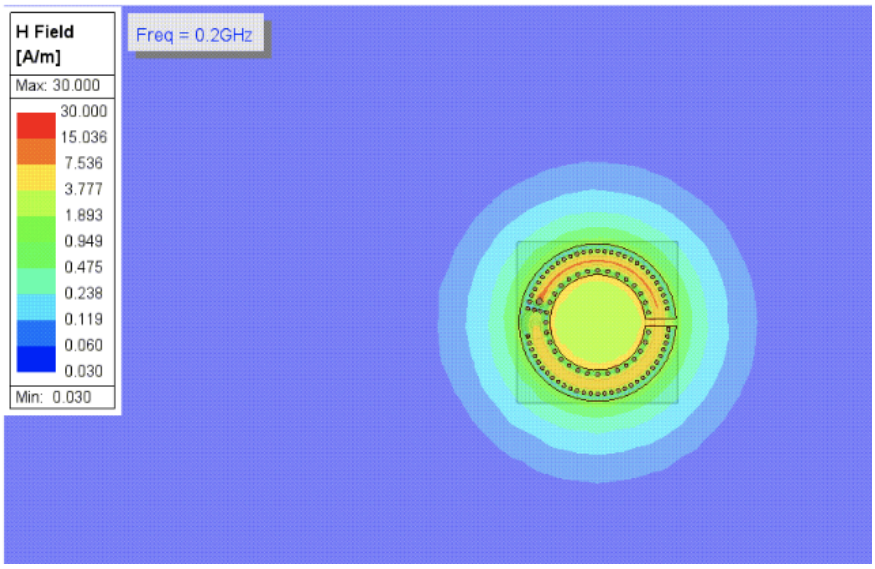

Figure 24. Simulated H field of Single Loop From 0 -1 GHz

3.2.4 Adding a Ground Plane to the Loop

In our design process, we recognized that we were missing a ground plane to isolate the muxes from the loops and to eliminate crosstalk. Figures 25 and 26 show the different options with which experimented to determine the ideal location for the ground plane. Figure 25 shows the ground plane directly under the loop as an added sheet of copper. This variation, we quickly found, shorted the signal entering the loop by sending all the signal to ground before it could even enter and go up into the loop. We thus determined that the ground plane should then be placed under the coax feed so as to avoid short circuiting the loop.

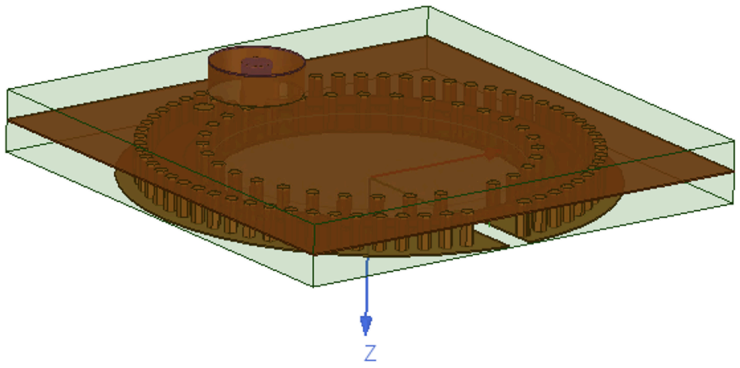

Figure 25. Single Loop with Ground Plane Between Loop and Coax Feed

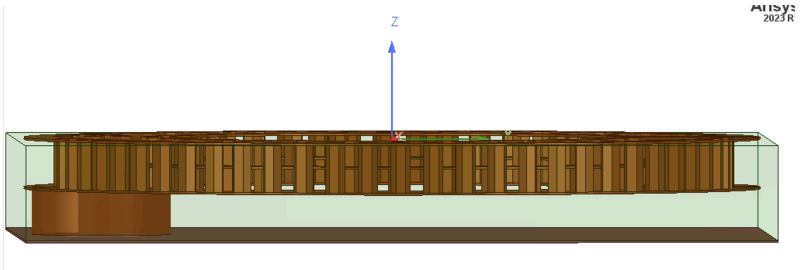

Figure 26. Iteration 2 of Ground Plane Under Coax Feed

Once we determined the location for the ground plane in the design, we simulated the radiation pattern in a plane oriented vertically with the loop. We observed a relatively uniform pattern emerging from the loop as shown in Figure 27. This confirmed our choice in placing the ground below the coax feed instead of directly beneath the loop.

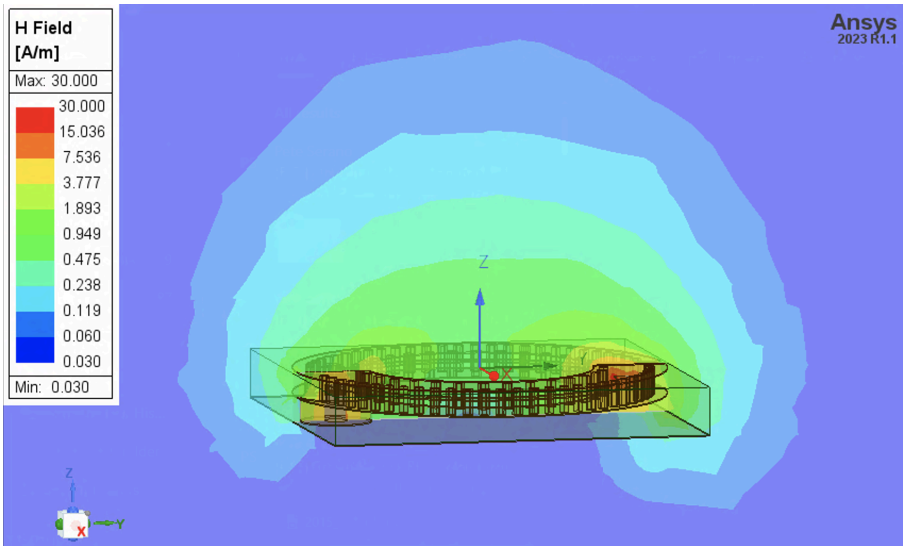

Figure 27. H field Visualization of Loop with Ground Layer Under Coax Feed

Once we made this alteration of adding the ground plane to the loop, we then made a 16 loop array of the H field probes. This final design is shown below in Figure 28.

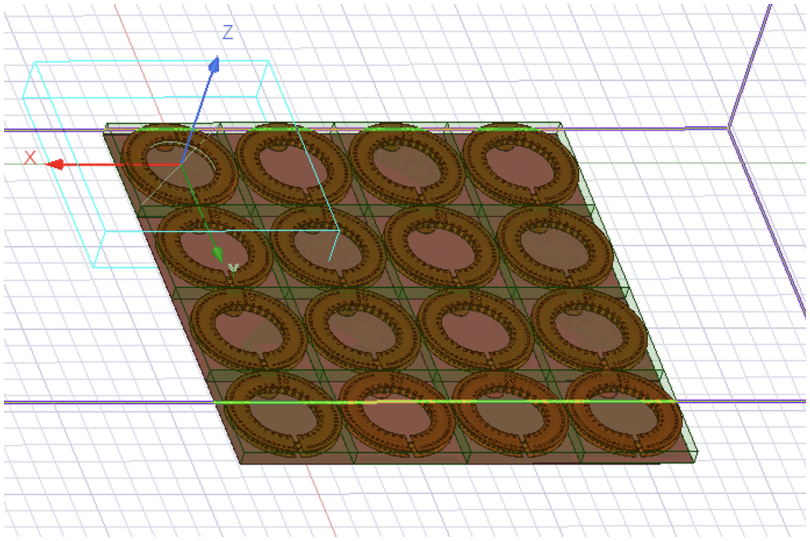

Figure 28. 4x4 Array of Grounded Loops

3.3 Control Structure of the EMScanneR

When we received the scanner and equipment setup from Teradyne we quickly realized the device had a unique control structure with an unknown pinout. Although we couldn't attain that information from Y.I.C., because it was confidential intellectual property, we discovered a work around. We began by drawing the connector on paper and used a digital multimeter to check the voltages at each pin while the device was running a scan (Figure 29).

Figure 29. Micro D-subminiature 25 (DB-25) Connector

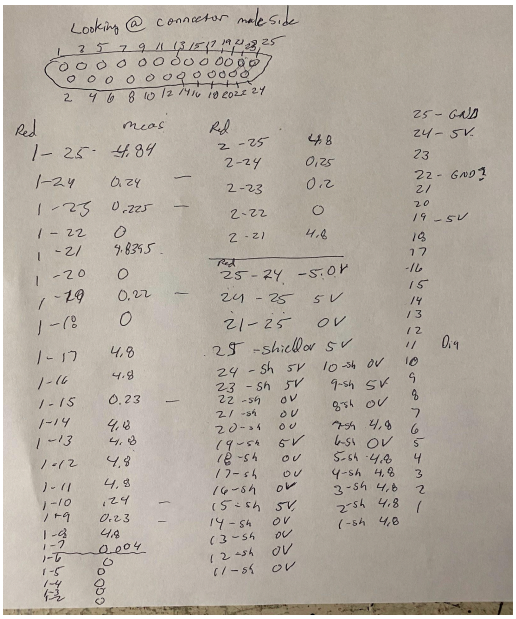

Figure 30. Measured Voltages on DB-25 Pins

Throughout this process we meticulously measured the voltages at every pin when the scanner was exciting a specific cell. This process was repeated for numerous areas on the board.

We recorded data from all of the pins as we selected different cells and found 12 bits that would consistently change. We determined that these were probably the control pins for the scanner and continued experimenting to see if this was true.

The method that we came up with to try to understand the control structure involved creating a pin “sniffer” (Figure 31).

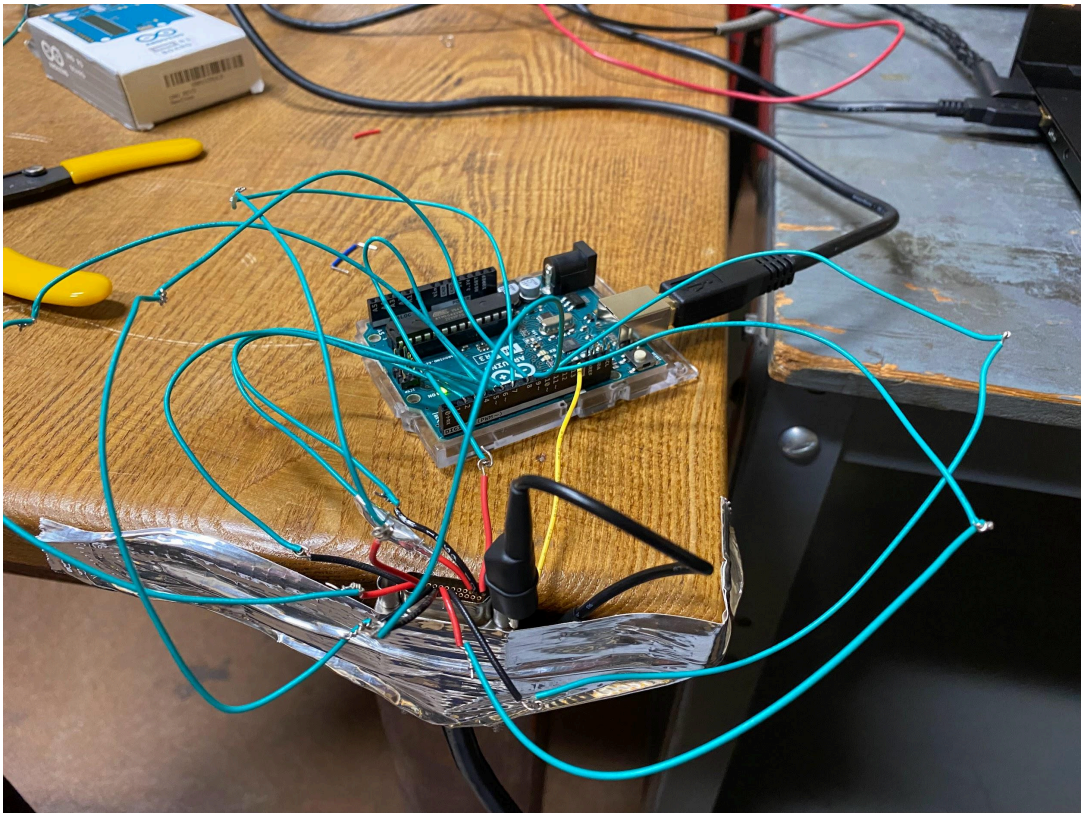

Figure 31. Pin Sniffing Circuit

We used our Arduino Uno R3 and wrote an Arduino script that would allow the board to take in the digital signal values at the anticipated pins. The Arduino would write the data to the serial monitor and a program called CoolTerm to generate a text file with all of the results. We believed that we would be able to unscramble the bits and determine what was the most

significant bit and least significant bit, however we were unsuccessful. For the sake of time and with the hope that Teradyne could acquire the intellectual property from Y.I.C. we decided to move on without determining a pattern.

3.4 Altium Schematic of 5x5 Scanner

3.4.1 Multiplexer Analysis

Teradyne provided presentation slides from a meeting with Y.I.C. which revealed existing SP8T multiplexers in the design structure of the EMI scanner. Our analysis of different components in excel also indicated that SP8T, specifically the HMC253AQS24E produced by Analog Devices, would work best for this application [16]. We know there are 1,218 antennae in the total array and they must be routed to a singular output. Dividing 1,218 by 8 inputs tells us 153 multiplexers are needed in the outermost level. Now the next step is to route each of the 153 outputs to an input on the next group of multiplexers. The process to determine the cascaded stages is the same: 153/8 = 20 multiplexers in the second level and 20/8 = 3 multiplexers in the third level. Now we have reached a point where we have all 1,218 loops routed to three outputs. Each multiplexer has eight inputs so there will only be one component necessary in the fourth level. We added together our previous calculations and found that 177 SP8T multiplexers would allow us to control the 25” x 25” array. In total, 177 of the HMC253AQS24E would cost around $2,812.53 total to purchase on Mouser.

Figure 32. HMC253AQS24E SP8T Multiplexer Produced by Analog Devices Inc

We performed a cost benefit analysis of various types of multiplexers comparing: cost, number of input pins, and the total cost after cascading all of the multiplexers. We calculated that we would only need five multiplexers to route all 25 loops, however we added extra multiplexing stages to see how it would increase output loss in an upscaled design. In the end, we determined that we would use seven muxes for our 5x5 small scale prototype.

3.4.2 Multiplexer Schematics

For our 5x5 scanner prototype we decided to use seven multiplexers and cascaded them into four different levels. Below is our Altium schematic for our completed design (Figure 33).

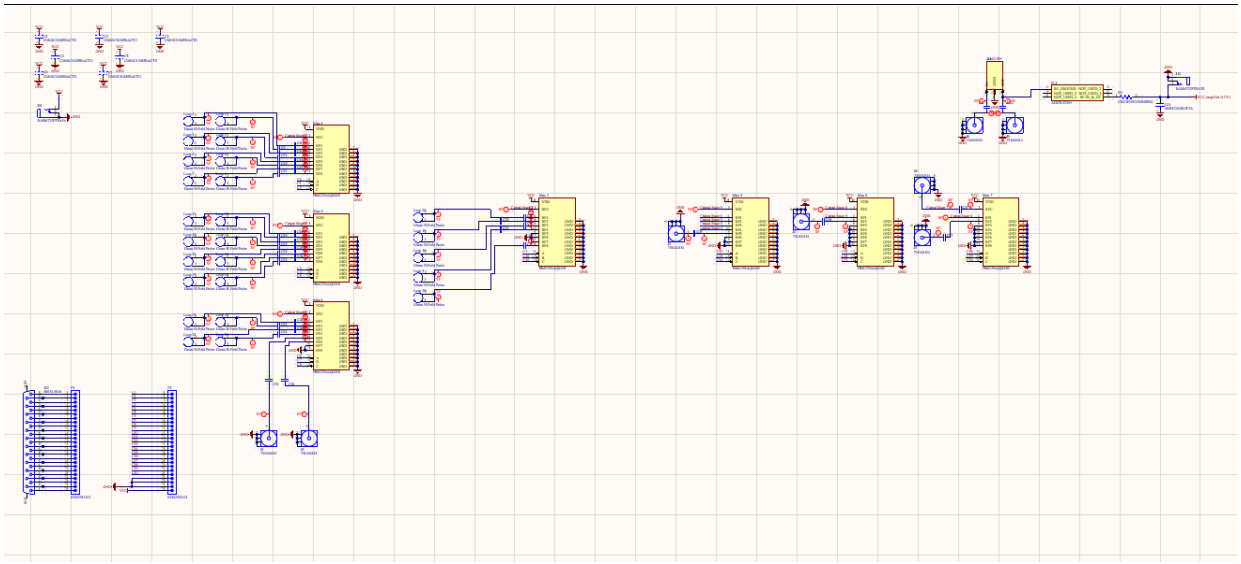

Figure 33. Prototype 5x5 Array Complete Schematic

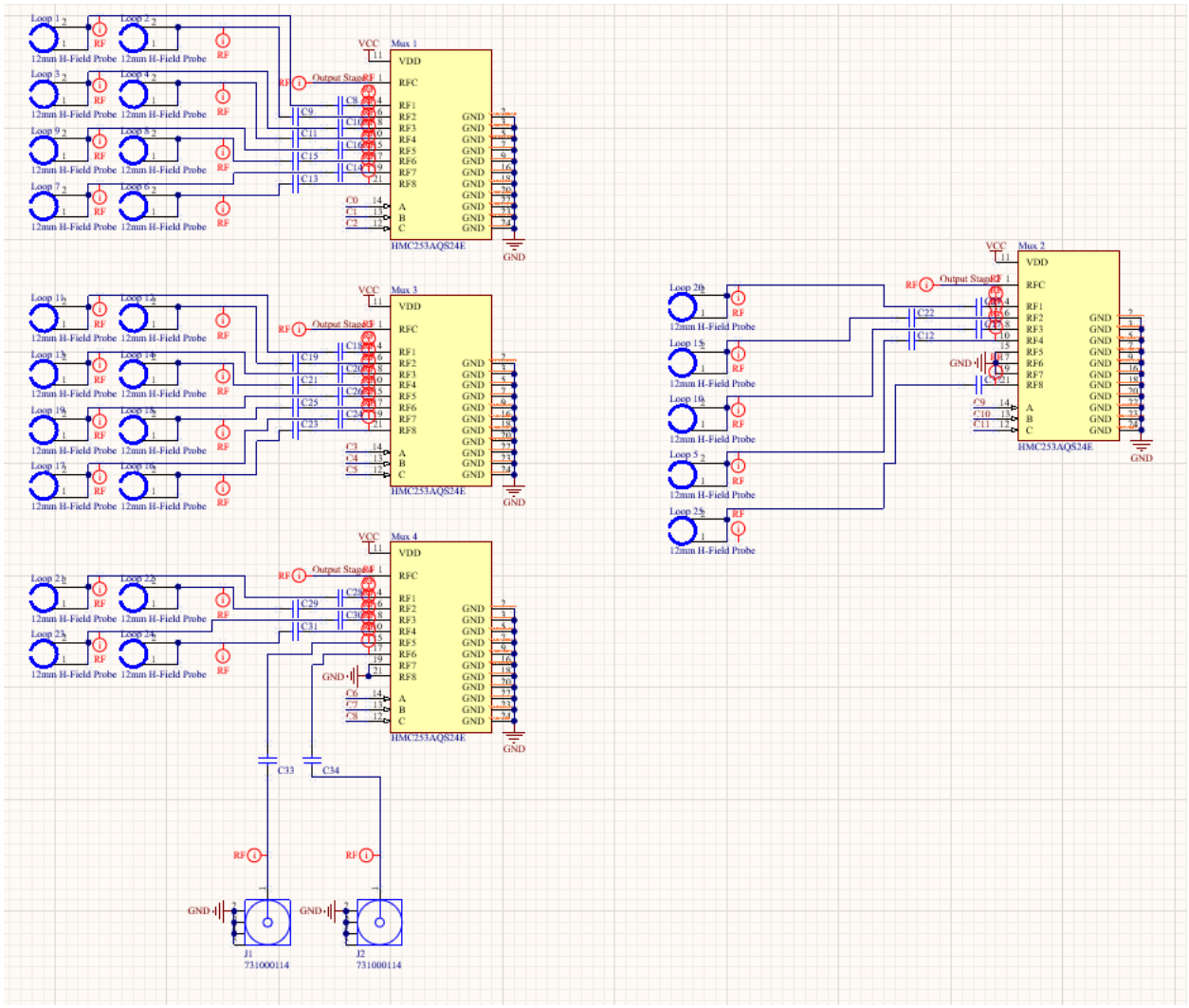

Figure 34. Prototype 5x5 Array Level 1

These are the first four multiplexers in the circuit that make up the outermost level which we will refer to as Level 1 (Figure 34). These multiplexers will share the same three control input signals. All 25 loops are connected to an RF input on one of these multiplexers. Two SubMiniature version A (SMA) connectors (J1 and J2) have also been added on “Mux 4” so we can connect external probes.

All of the outputs from Level 1 are routed to “Mux 5”, which we will refer to as Level 2 (Figure 35).

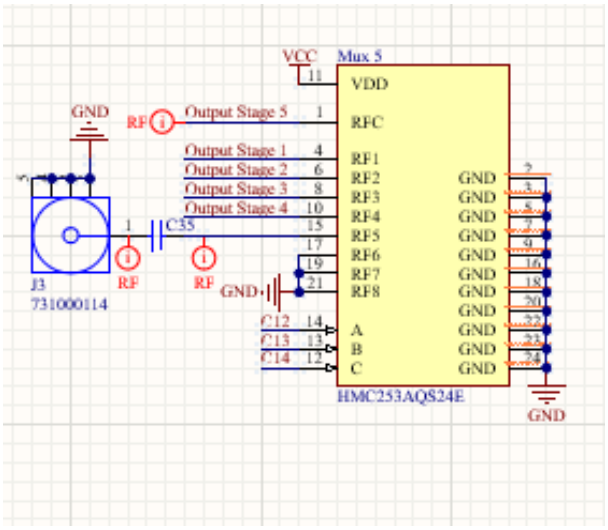

Figure 35. Prototype 5x5 Array Level 2

In this section the four outputs from the multiplexers in Level 1 are routed to the first four pins of "Mux 5". This multiplexer has its own unique set of control bits to choose between the four previous multiplexers and the SMA connector (J3).

The output of Level 2 is now going to be routed as an input for Mux 6 (Level 3) as seen below in Figure 36.

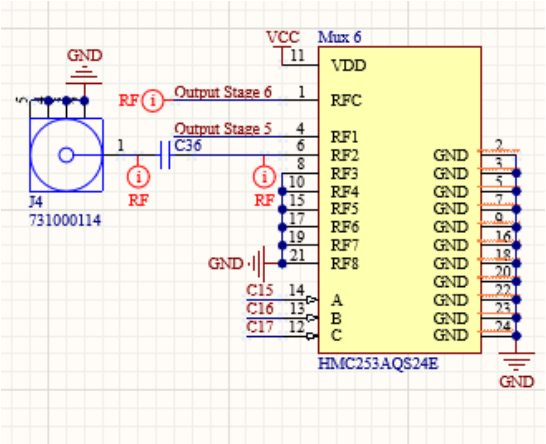

Figure 36. Prototype 5x5 Array Level 3

This multiplexer simply adds another layer of loss and also has its own unique set of control bits. The output of Level 3 is finally cascaded into Level 4 as an input pin (Figure 37).

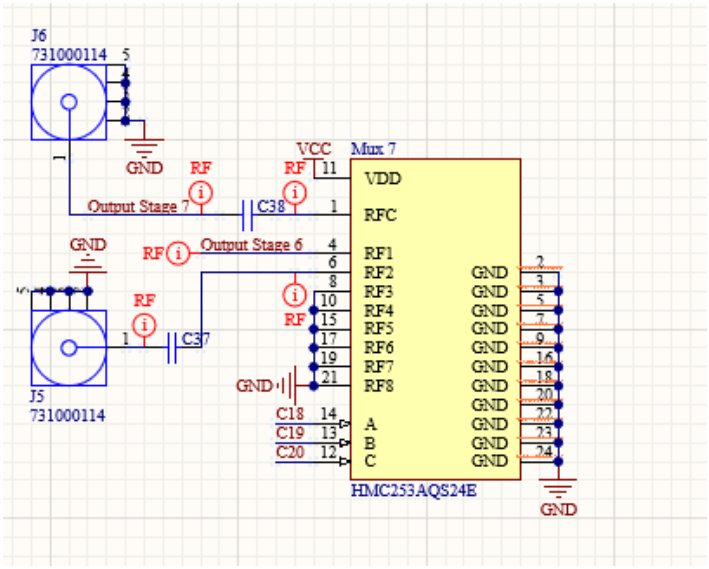

Figure 37. Prototype 5x5 Array Level 4

This multiplexer adds a fourth layer of loss for the routed signals and has its own unique set of control bits. This level has its output tied to an SMA connector so the signal can be connected to and recorded by the RF Analyzer. It is worth noting there is at least one SMA connector at every level. This allows us to compare the frequency response of an ideal antenna to the custom loops in the PCB and will give us an idea of how signal losses increase when we cascade multiplexers. Essentially, the Arduino Uno R3 will operate a switching structure for the board using 12 unique control signals (three for each layer).

Every multiplexer will be powered by a 5 V source that is fed into a barrel jack connector. They will also be tied to a decoupling capacitor, located near each input voltage pin, to short any stray RF energy to ground (Figure 38). All multiplexing circuits used the

HMC253AQS24E multiplexer development board as reference for routing and component selection (Figure 39).

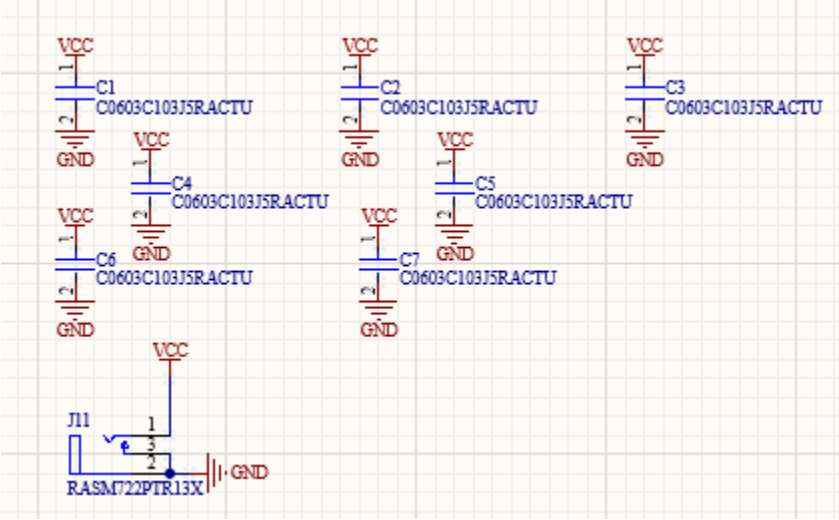

Figure 38. Multiplexer Power Source Circuit

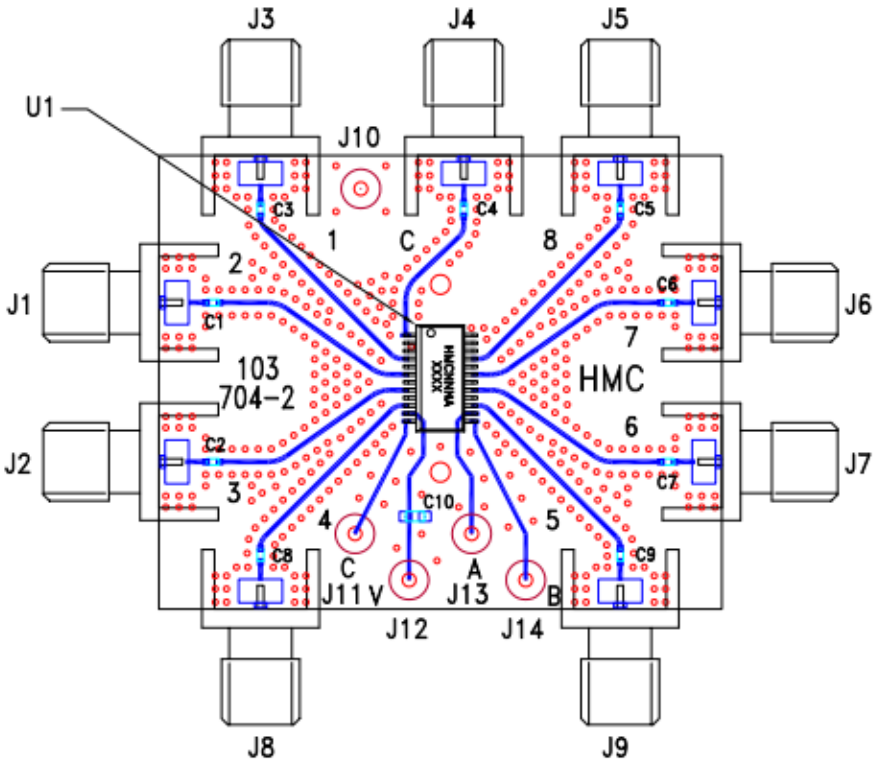

Figure 39. Evaluation Board EV1HMC253AQS24 for HMC253AQS24E Multiplexer

Source: Adapted from [17]

3.4.3 Amplifier Circuit Schematic

In the schematic, we also included a separate amplifying circuit that can be externally connected to the final output (Figure 40). This circuit used inspiration from the GALI-52+ developer board for component selection and routing (Figure 41). The output from the final multiplexer can be connected externally via an SMA cable to the input of the amplifier. The output of the amplifier will be connected to the RF Analyzer instead in this case. This circuit will be powered by a separate 12 V source from the second barrel jack connector.

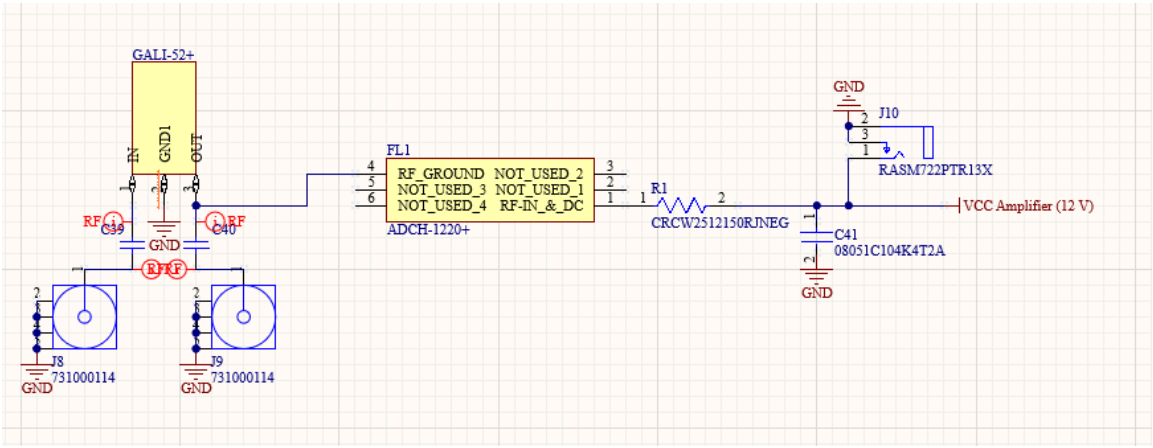

Figure 40. Prototype 5x5 Array Amplifier Circuit

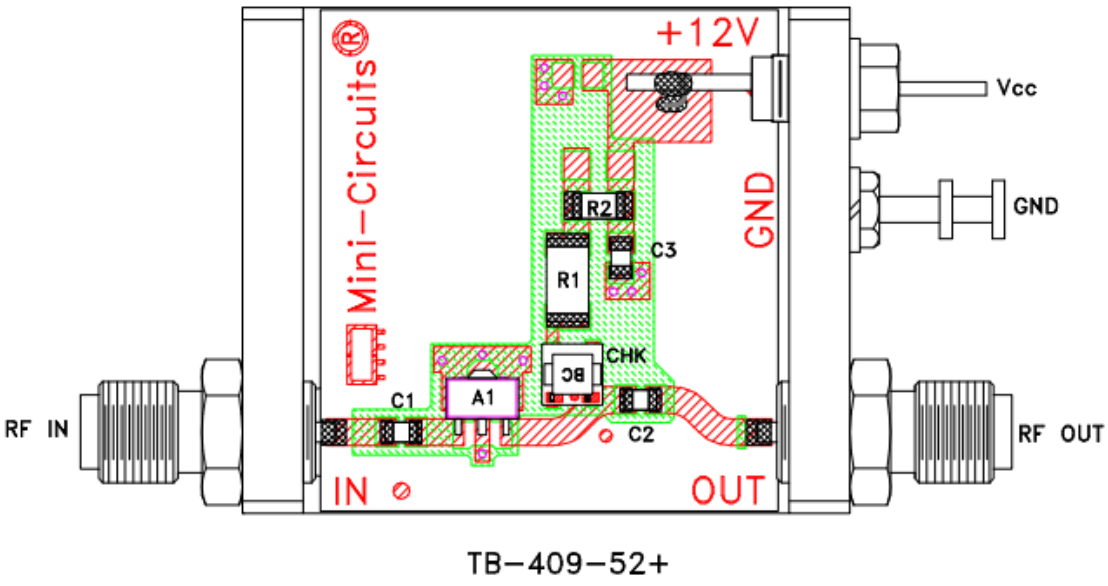

Figure 41. Evaluation Board TB-409-52+ for GALI-52+ Amplifier

Source: Adapted from [18]

3.4.4 Micro D-subminiature 25 and Header Schematic

We wanted to keep the DB-25 connector as a feature in our design in the event that Teradyne acquires the control structure of the EMScanneR from Y.I.C. technologies. Theoretically, this would allow them to use our board with their existing setup. Figure 42 below shows the DB-25 connector and two adjacent rows of 25 through hole header pins.

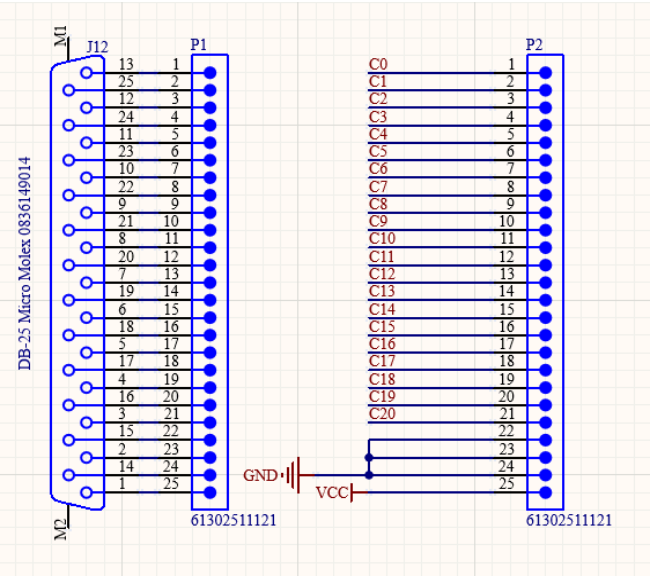

Figure 42. Micro D-subminiature 25 Connector and Header Pins

The first row of header pins (P1) is directly tied to the DB-25 connector. The second row of header pins (P2) is tied to each of the 21 control pins. This configuration allows P2 to be connected manually using jumper wires to P1. Essentially, the control bits can be wired in any order to the DB-25 connector. This enables the design to work with the current setup at Teradyne and our Arduino test setup.

3.5 Creating Our Own Control Software

We decided to develop code that will be able to run our 5x5 scanning array in the event Teradyne is unable to get the intellectual property from Y.I.C. We also learned how to write code in Microsoft Visual Basic that would work in conjunction with the code from the Arduino. The code in visual basic is what allows us to sweep the frequency in the RF Analyzer at the same time the control signals are being sent to the multiplexers.

3.7 Altium Component Library

3.7.1 Creating the Loop Probe Component in Altium

Before we could start laying out the PCB we had to transfer our antenna design from Ansys HFSS into Altium (Figure 43).

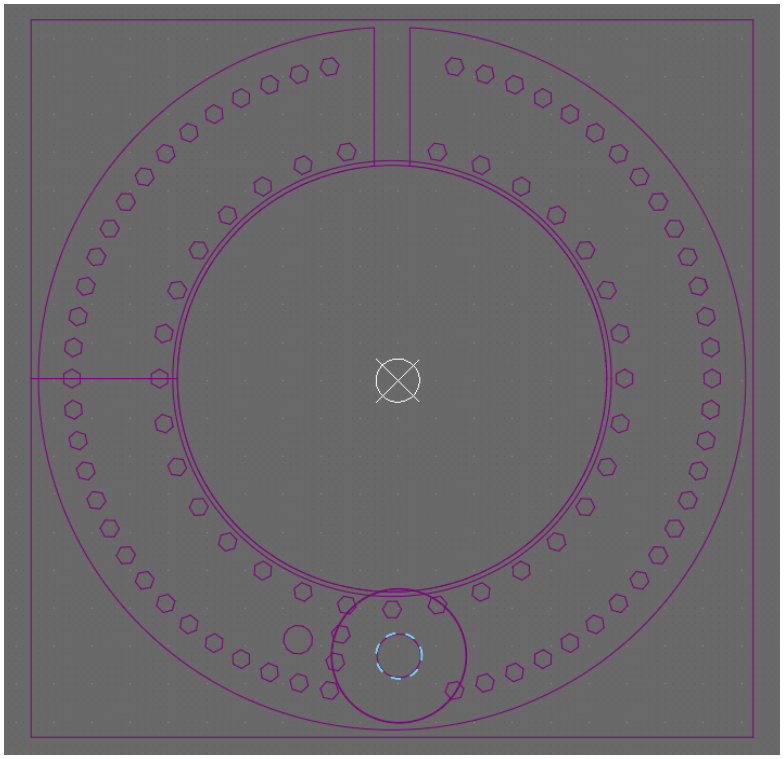

Figure 43. Drawing Exchange Format File of Antenna Loop Probe

This process introduced a steep learning curve since it involved creating a multilayer embedded component. First, we exported the design as a Drawing Exchange Format (DXF) file. Altium generated a 2D outline of the component, however the copper had to be manually drawn on each layer to create a 3D component. Everything that spans multiple layers in Altium must have a predefined layer stack. In our case, we looked at our HFSS design for dimensions and matched our layer thickness and spacing based on that (Table 2).

	Top Overlay		Overlay				
	Top Solder	Solder Resist	Solder Mask	0.4mil	3.5		
1	GND 1	CF-004	Signal	2.756mil		2oz	
	Dielectric 1	FR-4	Prepreg	29.506mil	4.2		0.02
2	Loop Probe	CF-004	Signal	2.756mil		2oz	
	Dielectric 2	FR-4	Dielectric	29.506mil	4.2		
3	GND 2	CF-004	Signal	2.756mil		2oz	
	Dielectric 3	FR-4	Prepreg	29.506mil	4.2		0.02
4	Bottom Port	CF-004	Signal	2.756mil		2oz	
	Bottom Solder	Solder Resist	Solder Mask	0.4mil	3.5		
	Bottom Overlay		Overlay				

Table 2. Layer Stack for Antenna Loop Probe in Altium

The dielectric constant is the number written on the gray boxes in the fifth column of the layer stack; 4.2 was determined to be within range of our intended PCB manufacturer, PCBWay [19]. We chose two ounce copper because our simulated antenna already had a copper layer thickness of 2.8 mil. Two ounce copper (2 Oz) is the number written in the sixth column and it indicates that the layer is 2.756 mil in thickness.

We generated all of the copper in the component using the fill tool once every layer was fully defined. This left us with the top most shield and the loop probe trace seen below in Figure 44.

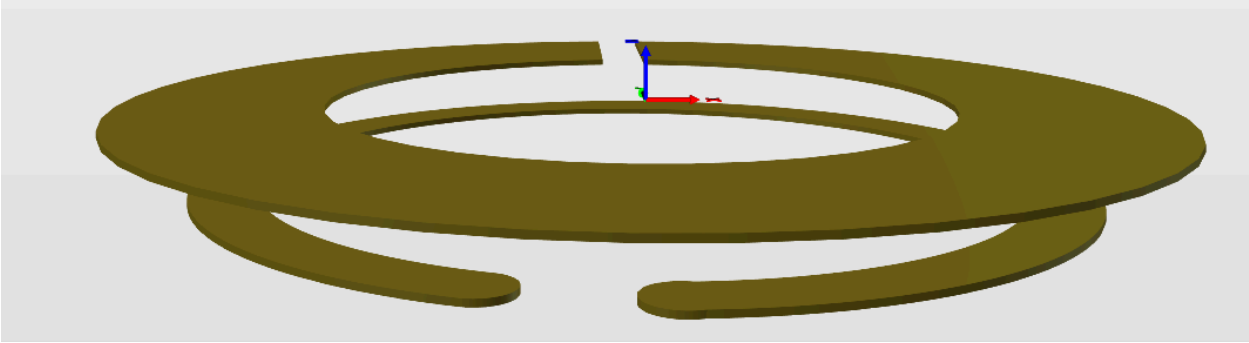

Figure 44. 3D View of Loop Probe PCB Library File

It is worth noting that the bottom most shield was not included in the PCB library file. This was an intentional decision because that layer needed to be a “polygon pour” instead of a “fill”. A polygon pour will create a cutout around vias that extend through a layer if they are not of the same net while a fill or a solid region will not. In earlier iterations the via that connected the loop probe trace to the multiplexing layer was being shorted directly to ground when it passed through the loop shield. When using polygon pour, Altium can recognize that the via and the layer are two distinct nets and resolve that issue in our design.

We linked this PCB library file to the corresponding schematic drawing that already existed in the design (Figure 45).

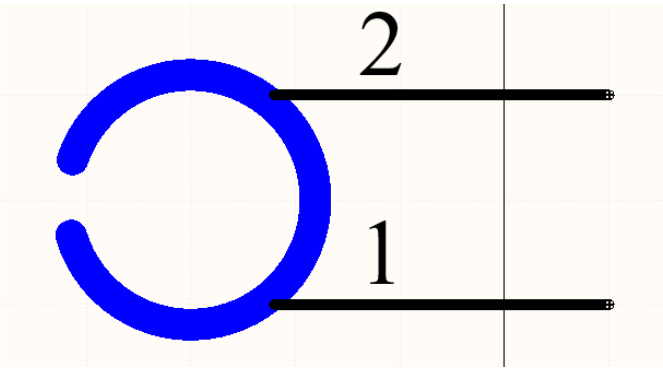

Figure 45. Loop Probe Schematic Library File

3.7.2 Importing Components into Altium

Although the loop probes were the only components that needed to be manually created in Altium, they were not the only imported components. Other components that were not available in the Altium built in library that needed to be imported include the Molex 83614-9014 Micro D Connector, the GALI-52+ RF Amplifier, and the ADCH-1220+ RF Choke [20]. This process was fairly straightforward because many suppliers like Digikey and Mouser provide free 3D library files for components that are not available directly on Altium.

4 Results and Modifications

The loop array architecture underwent innumerable modifications throughout the design process. The entire design of the loop probe used as the basis building block of the scanner, underwent a complete reconstruction from the initial design made based off of the NAE H field 15 mm probe. The loop feed was entirely reconstructed and moved to a layer beneath that of the loop itself, attaching vertically to the bottom of the loop and going down into the structure of the board. Additionally the muxing structure was restructured repeatedly as the PCB design and wiring changed with evolution of the design.

4.1 HFSS Simulation Results

There were many iterations of the loop and array design which we improved and refined upon in designing the scanner. Once we established the basic geometry of the probe, we continued to add further details to the loop. The major components of the loop in the later more final iterations of our design included: the top shield, the bottom shield, the internal loop, the shielding vias, and the via loop. Figure 46 shows the radiation pattern of a row of excited loops in the final design of the H field probe. The radiation pattern shows a relatively uniform distribution of the near field when exciting the loops.

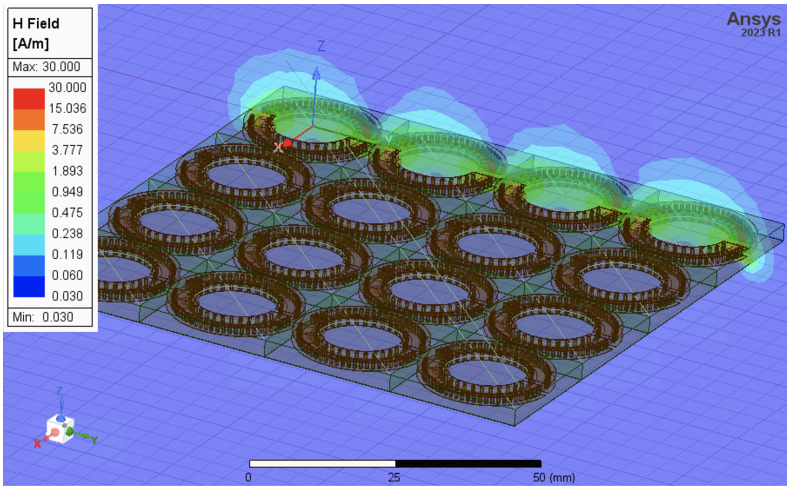

Figure 46. Complex Magnitude of H Field at 1 GHz

Once we determined a functional geometry for the loop where the loop was properly parameterized and had a ground plane to provide shielding from the muxes, we then simulated the Z parameters to determine the resonance frequencies for a range of loop diameters. Since we were trying to maximize the loop size while ensuring that the resonance frequency was beyond the 1 GHz upper limit, we found the range of diameters of interest 10-15 mm as discussed in 3.2.3 in the methodology. From there, we targeted these diameter values with increments of 1 mm and simulated the Z parameters. The results of the simulation are shown in Figure 47. In this plot, the 12 mm diameter is the ideal size since it is the largest diameter before the resonance frequency encroaches in the frequency band of operation. The 12 mm diameter resonates just beyond 1 GHz while the loops of increasing size resonate under the 1 GHz frequency.

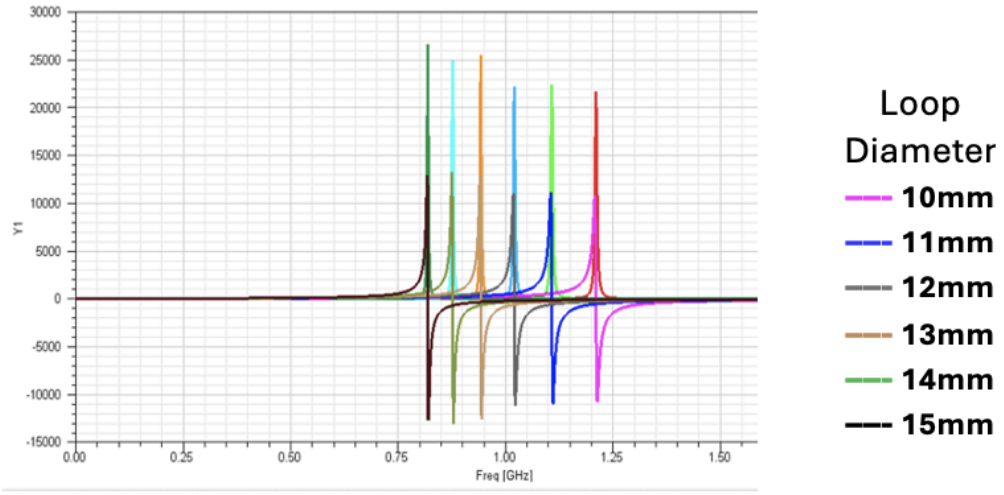

Figure 47. Resonance Frequencies of the Single Loop

We also simulated the Z parameters for the entire 4x4 array. The results of this simulation are shown in figure 48. Despite the added noise accounting for all 16 loops in the array simulation, the results are still the same and illustrate that the 12 mm loop is the diameter of interest for our application. Although the resonances may have shifted higher slightly when accounting for more loops in the simulation, the next largest size 14 mm still resonates under 1 GHz.

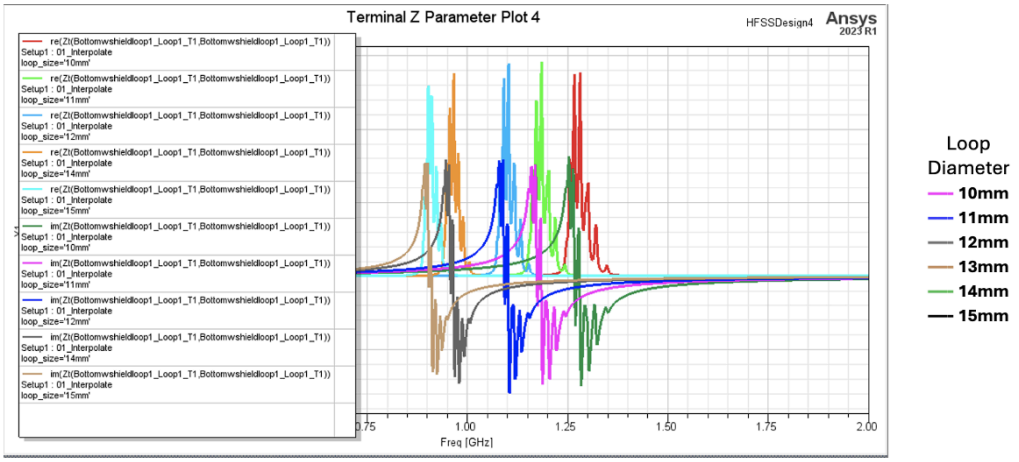

Figure 48. Resonance Frequencies of 4 x 4 Array

4.2 Design Modifications

4.2.1 Early Altium Designs

There were several improvements that we made to our design as we started laying out the components. Our weekly meetings with Teradyne provided us with helpful insight on how we could add extra features to our design. The modifications we made as we progressed to this point included: extra SMA connectors for external antennae and amplification, increasing the total number of board layers from six to eight, rerouting the multiplexer RF input pins to the nearest antenna loop rather than in numerical order, and adding two header pin rows. We were able to present Teradyne with our first iteration of the PCB after these changes were made.

This is the first iteration of our Altium PCB design that was sent to Teradyne (Figure 49). Teradyne reviewed this design in Altium as well as an exported Open Database (ODB++) file.

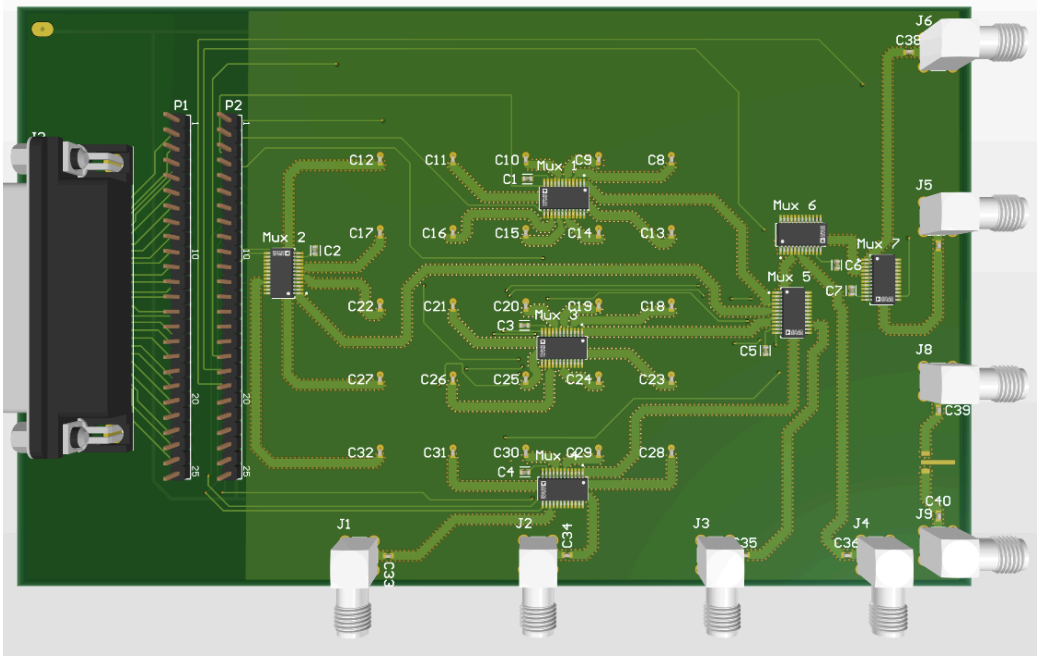

Figure 49. Version 1 of Altium PCB

Teradyne first mentioned in their findings that the amplifier was not powered properly (Figure 50). Upon further examination it was a pretty obvious mistake, however it was an easy problem to resolve in future designs by adding the missing components.

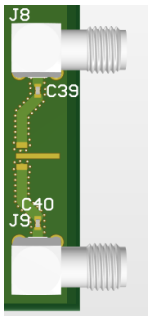

Figure 50. Incomplete Amplifier Circuit

Trace width and surrounding shielding vias were also noted to be problematic. The traces were so large and close to the vias that the minimum air gap spacing may have been violated during fabrication. Ground via connections were also pointed out as missing under some multiplexers; in the completed prototype these pins would act as an inaccurate reference for measuring DC voltages. The last concern mentioned was that the RF traces narrowed down near the pins inconsistently potentially creating a mismatch (Figure 51).

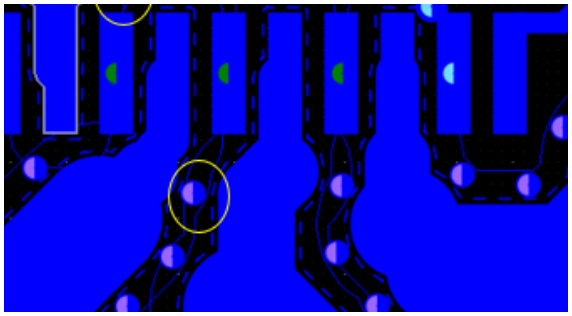

Figure 51. RF Traces Creating Impedance Mismatch at Multiplexer Input Pins

At this point we had already decided an eight layer board would provide us with ample routing space. Our layer stack matched our PCB manufacturer's standard eight layer stack for two ounce copper, however the dielectric thickness didn't. We were forced by Altium to use the original dielectric thickness of 29.506 mil from the imported HFSS model of the loop in many of the dielectric layers to maintain board symmetry (Table 3).

	Top Overlay		Overlay				
	Top Solder	Solder Resist	Solder Mask	0.4mil	3.5		
1	Loop Shield 1	CF-004	Signal	2.756mil		2oz	
	Dielectric 7	PP-006	Prepreg	29.506mil	4.2		0.02
2	Loop Trace	CF-004	Signal	2.756mil		2oz	
	Dielectric 1	FR-4	Core	29.506mil	4.2		0.02
3	Loop Shield 2	CF-004	Signal	2.756mil		2oz	
	Dielectric 2	FR-4	Prepreg	29.506mil	4.2		0.02
4	GND 1	CF-004	Plane	2.756mil		2oz	
	Dielectric 3	FR-4	Core	29.506mil	4.2		0.02
5	GND 2	CF-004	Plane	2.756mil		2oz	
	Dielectric 4	FR-4	Prepreg	29.506mil	4.2		0.02
6	DC Signal Layer	CF-004	Signal	2.756mil		2oz	
	Dielectric 5	FR-4	Core	29.506mil	4.2		0.02
7	GND 3 (RF)	CF-004	Signal	2.756mil		2oz	
	Dielectric 6	PP-006	Prepreg	29.506mil	4.2		0.02
8	RF	CF-004	Signal	2.756mil		2oz	
	Bottom Solder	Solder Resist	Solder Mask	0.4mil	3.5		
	Bottom Overlay		Overlay				

Table 3. Layer Stack for Version 1 of Altium PCB

This wouldn't be problematic if this board exclusively routed DC signals. RF signals on the other hand are a different story. RF signal lines require a 50 ohm impedance matched trace. Making the traces match that value depends on the dielectric constant, dielectric thickness, and copper width. Our extremely thick PCB required us to have a 56.259 mil trace on the RF multiplexing layer if we wanted to maintain 50 ohms. This was the main cause of the previously mentioned size constraints and routing issues.

Before implementing their feedback we excited the RF lines in HFSS to visualize how RF energy may propagate throughout the board and create unwanted effects. Below are some results from our HFSS simulation of the PCB (Figure 52 and 53). Some major issues that we found during this analysis include leakage of RF energy into adjacent DC lines and into other loops. One reason this happened is because the DC lines were on the same layer as the RF traces and there was nothing preventing them from coupling together. The second reason is because the vias that bring RF energy from the loops to the multiplexing layer did not have any ground return vias running parallel to them. The ground return vias would help to act as a Faraday cage that absorbs stray energy.

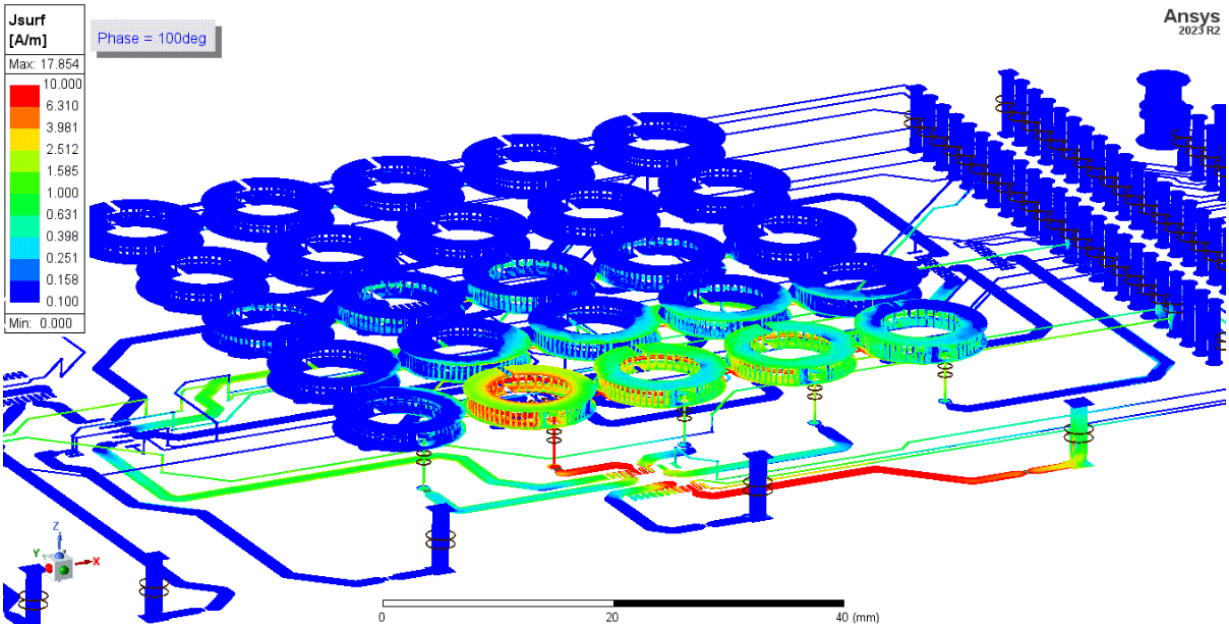

Figure 52. Excitation of RF Lines in HFSS

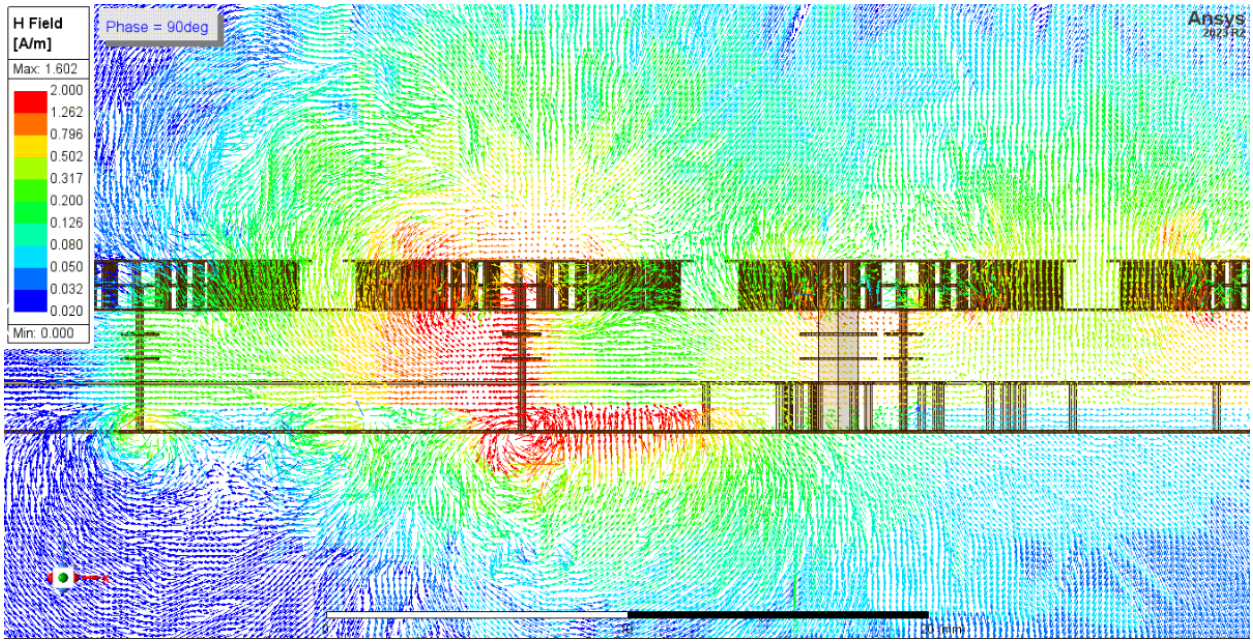

Figure 53. Visualization of RF Energy Propagating Through the Unshielded Feed Line

We implemented all of the feedback that we received since Version 1 to improve our next design. We will refer to the next version of our PCB Design as Version 2 (Figure 54 and 55). In this version we reconciled many issues. We completed the amplifier circuit, updated dielectric thickness to match the PCB manufacturer standard, reduced the RF trace width, rerouted the DC signal lines, added ground return vias around the unshielded feed lines, added "GND" net polygon pours to every signal layer, added stitching vias on the RF multiplexing layer, and added two barrel jacks to conveniently supply power to both circuits.

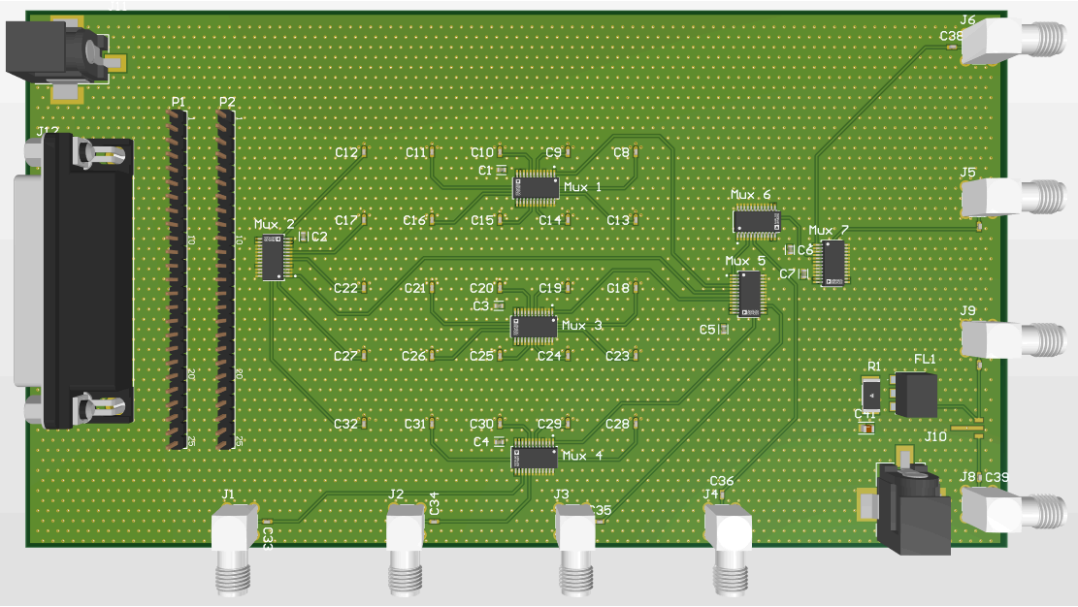

Figure 54. Version 2 of Altium PCB Multiplexing Side

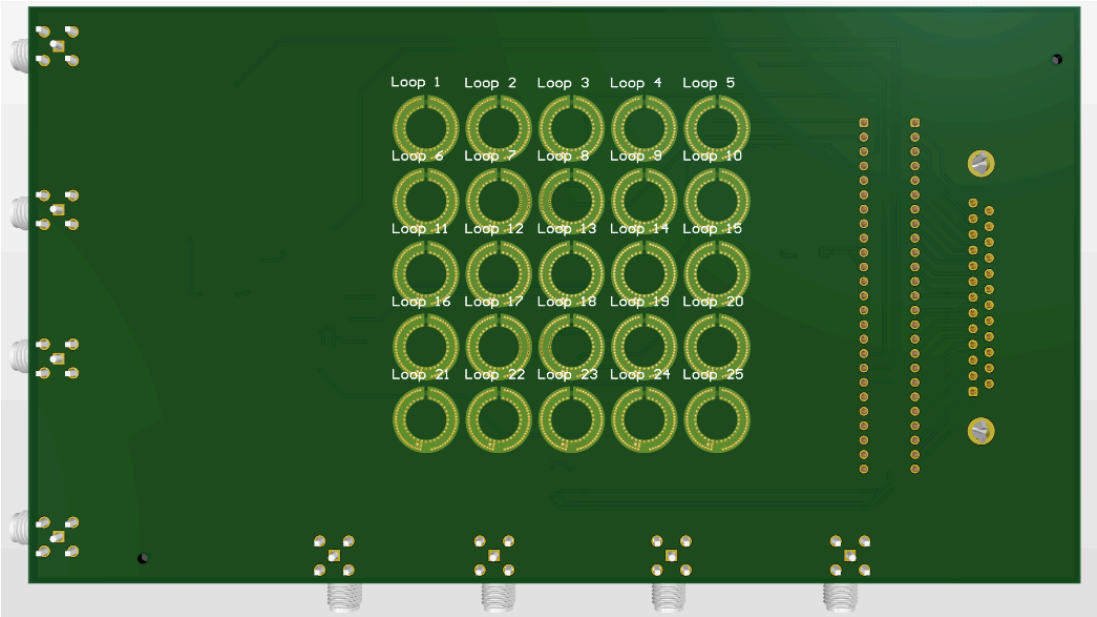

Figure 55. Version 2 of Altium PCB Loop Side

In Version 2 the new dielectric thickness based on the PCBWay's eight layer two ounce stackup made it possible to create 50 ohm RF traces at 11.811 mil (Table 4). This made it significantly easier to route all of the RF lines again because they are now comparable to the width of the multiplexer pins.

#	Name	Material	Type	Thickness	Dk	Weight	Df
	Top Overlay		Overlay				
	Top Solder	Solder Resist	Solder Mask	0.4mil	3.5		
1	Loop Shield 1	CF-004	Signal	2.756mil		2oz	
	Dielectric 1	PP-006	Prepreg	7.087mil	4.2		0.02
2	Loop Trace	CF-004	Signal	2.756mil		2oz	
	Dielectric 2	FR-4	Core	7.874mil	4.2		0.02
3	Loop Shield 2	CF-004	Signal	2.756mil		2oz	
	Dielectric 3	FR-4	Prepreg	7.087mil	4.2		0.02
4	DC Signal 1	CF-004	Signal	2.756mil		2oz	
	Dielectric 4	FR-4	Core	7.874mil	4.2		0.02
5	DC Signal 2	CF-004	Signal	2.756mil		2oz	
	Dielectric 5	FR-4	Prepreg	7.087mil	4.2		0.02
6	GND 1	CF-004	Signal	2.756mil		2oz	
	Dielectric 6	FR-4	Core	7.874mil	4.2		0.02
7	GND 2 (RF)	CF-004	Signal	2.756mil		2oz	
	Dielectric 7	PP-006	Prepreg	7.087mil	4.2		0.02
8	RF	CF-004	Signal	2.756mil		2oz	
	Bottom Solder	Solder Resist	Solder Mask	0.4mil	3.5		
	Bottom Overlay		Overlay				

Table 4. Layer Stack for Version 2 of Altium PCB

Source: Adapted from [21]

Moving the DC signal lines down and increasing the number of shielding vias and ground planes throughout the design appears to have had a positive effect as well. After simulating Version 2 in HFSS we were able to reduce our isolation from -40 dB to -20 dB (Figure 56).

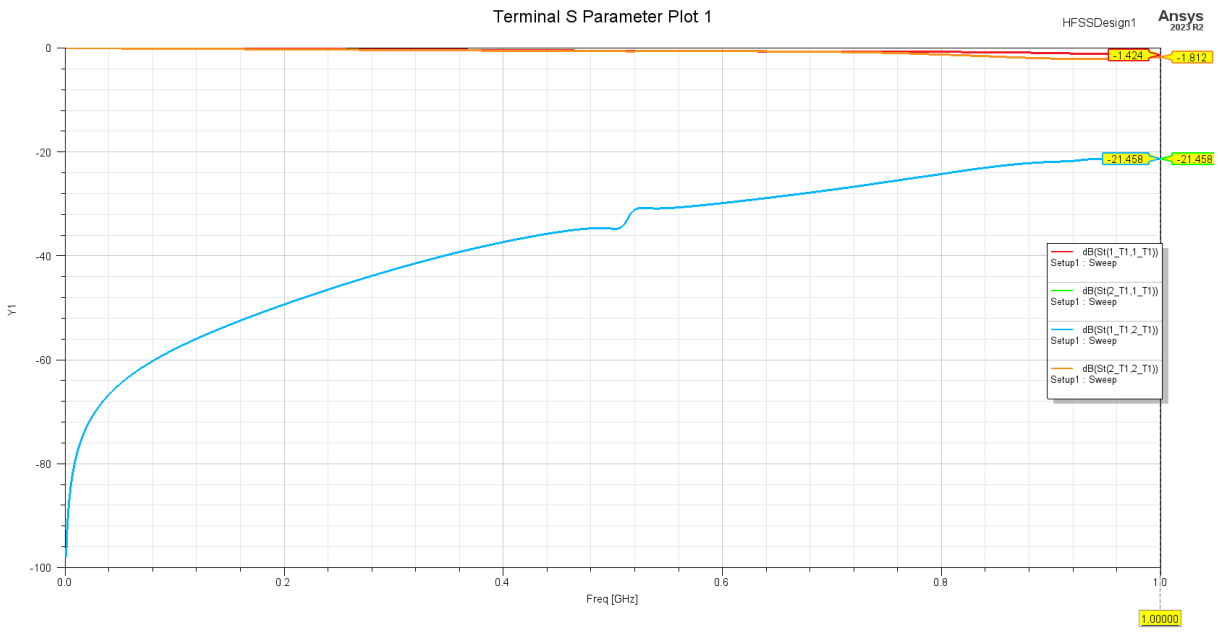

Figure 56. S-Parameter Plot After Updates

4.3 Final Design

4.3.1 Final PCB Layout

After receiving more feedback from Teradyne and reviewing our PCB manufacturer's website we were able to finalize our board design [22]. Version 3 is the latest iteration of the 5x5 prototype scanning array shown in Figures 57 and 58 below.

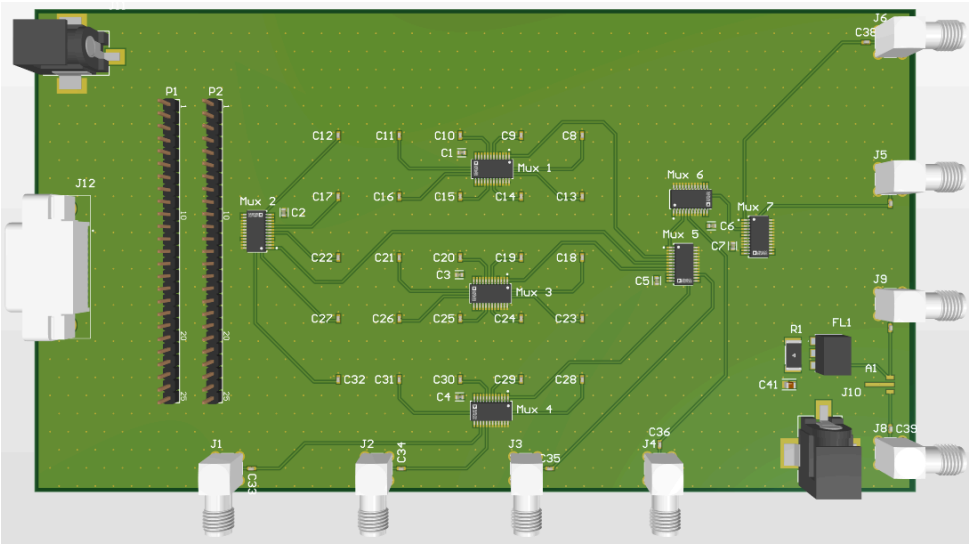

Figure 57. Version 3 of Altium PCB Multiplexing Side

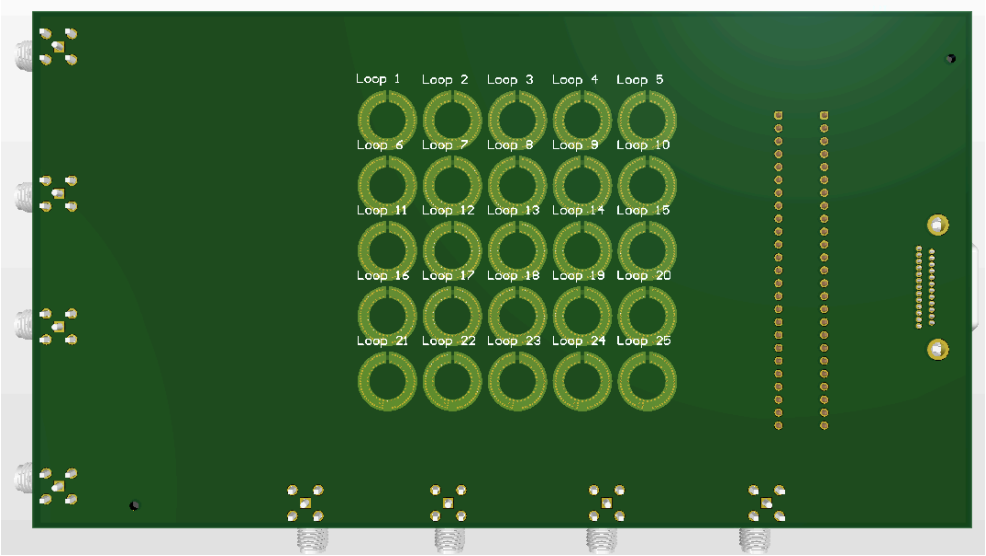

Figure 58. Version 3 of Altium PCB Loop Side

Developing Version 2 into 3 required substantially less revisions compared to the progression of Version 1 into 2. This time Teradyne only had one suggestion for how we could improve our second design and prepare it to be manufactured. One issue that they noted in the previous design involved a floating copper plane on layer “DC Signal 2”. Generally leaving a

floating copper plane is bad design practice because it will act as a diving board that causes the surrounding signals to fluctuate by +/- several mV. We fixed this issue in Version 3 by tying two vias that span to the nearest ground layer on each end of the floating copper (Figure 59).

Figure 59. Floating Copper Plane After Grounding Via Fix

Two issues in Version 2 that we corrected in this design were incorrect blind/buried via size and an incorrect D-Sub 25 connector component. PCBWay specifies on their website that they follow the IPC standard for blind and buried vias. The standard suggests that the diameter of blind and buried vias should be 6 mils or less [23]. In Version 2 almost all of the blind and buried vias did not meet this criteria. After learning this, all blind and buried vias were changed to meet the requirement in the final design. Finally, when we checked the bill of materials file we noticed that the Micro D-Sub 25 connector had the correct name, but the component footprint was actually a regular D-Sub 25. The problem with this is that it is a much larger component and would not work with Teradyne's current setup. In Version 3 this component was replaced with a Micro D-Sub 25 to retain the circuit's intended functionality.

4.3.2 Bill of Materials

Altium automatically generates and updates a bill of materials document that is tied to the overall design (Table 5). This file records all of the information about each component; some details included are the name, description, designator, quantity, manufacturer, and manufacturer

part number. Most components that can be directly downloaded through the built in library in Altium also have the ability to update the unit price from available suppliers. This feature allows us to determine exactly how many components we need to purchase. Although it does not always apply to components that have been imported, we made sure that every component we used in the design was available on either Mouser or Digikey. In our final design we have determined that we will need to populate each prototype board with:

- Mini-Circuits ADCH-1220+ **RF Choke** (1x)
- Mini-Circuits GALI-52+ **RF Amplifier** (1x)
- Kyocera AVX 08051C104K4T2A **Ceramic Capacito**r (1x) [24]
- Molex 73100-0114 **Right Angle SMA Connector** (8x) [25]
- Wurth Electronics 61302511121 **Vertical 25 Pin Header** (2x) [26]
- Molex 836149014 **D-Sub 25 Micro-D Connector** (1x) [27]
- KEMET C0603C103J5RACTU **Ceramic Capacitor** (7x) [28]
- Vishay CRCW2512150RJNEG **SMD Chip Resistor** (1x) [29]
- Murata GRM1555C1H101JA01D **Ceramic Capacitor** (33x) [30]
- Analog Devices HMC253AQS24ETR **RF SP8T Multiplexer** (7x)
- Switchcraft RASM722PTR13X **Barrel Jack Connector** (2x) [31]

There will also be 25 12 mm H field Probes, however they are not included in this list because they are physically embedded into the board and will be made by the PCB manufacturer.

	Line #	Name	Description
1		12mm H-Field Probe	
2		ADCH-1220+	Filter
3		GALI-52+	RF Amplifier SMT Low Noise Amplifier, DC - 2000 MHz, 50
4		08051C104K4T2A	Automotive Ceramic Capacitor, 0805, 100nF, 10%, X7R, 15%, 100V
5		731000114	SMA Right-Angle PCB Jack Receptacle, Brass Body, Gold Plated, 50 Ohm, 5-Pin THD, RoHS, Tray
6		83614-9014	Connector
7		61302511121	
8		C0603C103J5RACTU	Capacitor; Ceramic; Cap; .010uF; Tol 5%; SMT; Vol-Rtg 50V; X7R; Tape and Reel
9		CRCW2512150RJNEG	RES Thick Film, 150Ω, 5%, 1W, 200ppm/°C, 2512
10		GRM1555C1H101JA01D	
11		HMC253AQS24ETR	IC RF SWITCH SP8T 2.5GHZ 24QSOP
12		RASM722PTR13X	CONN PWR JACK 2.1X5.5MM SOLDER

Table 5. Altium Design Bill of Materials

5 Conclusion

We designed a 25 H field loop probe array as a smaller scale prototype for a 1,218 loop probe array with dimensions of 25” x 25”. Our design functions within the frequency range of 10 kHz - 1 GHz and uses 12 mm probes to receive radiated energy. The board design consists of an eight layer PCB with two copper grounding layers, DC traces for control signals and component power, and RF traces that act as waveguides for radiated near field energy. We also wrote an Arduino script as a means to circle through the loops as part of the control structure. Due to the time constraints, we were unable to test the board and compare our results with the behavior of the Y.I.C scanner. Regardless, our design has laid the groundwork for a functional prototype of a larger scale scanner for EMI testing.

6 Recommendations

We established earlier in the paper that the original goal of this project was to create a larger physical prototype for Teradyne (25” x 25”). Our future recommendations involve testing the functionality of the 5x5 PCB prototype array, improving on any design flaws, and developing a unique non-proprietary control structure for the device. Once these goals have been reached the design can be expanded upon and interfaced with existing Y.I.C. control software. As the design progresses it should aim to capture the same spatial heatmap that the original EMScanneR produces.

References

[1] EMxpert, “EMxpert EHX User Manual.”

[2] G. Fitzgerald, “Proposal for a 1-shot RadEM Measurement System”, Nov. 2023, Internal Teradyne Corp Presentation -600 Riverpark Dr. No. Reading, MA 01864, slides 2-4.

[3] “Equipment Authorization – RF device,” Federal Communications Commission, https://www.fcc.gov/oet/ea/rfdevice (accessed Apr. 24, 2024).

[4] R. L. Stenzel, J. M. Urrutia, “Comparison of electric dipole and magnetic loop antennas for exciting whistler modes” in *Phys. Plasmas* 1 August 2016; 23 (8): 082120.

[5] V. Kraz, “Near-Field Methods of Locating EMI Sources” in *Compliance Engineering Magazine May/June 1995*, Soquel, CA, USA: Credence Technologies Inc, 1995, pp. 1-7.

[6] M. Zhang, “The Magic Loop – Techniques of using magnetic field loops” in Machone.

[7] B. Deutschmann, H. Pitsch, G. Langer, “Near Field Measurements to Predict the Electromagnetic Emission of Integrated Circuits”, Nov. 29, 2005.

[8] ETS Lindgren, “Model 7405 Near-Field Probe Set User Manual.”

[9] J. Bang, Y. Park, K. Jung and J. Choi, "A Compact Low-Cost Wideband Shielded-Loop Probe With Enhanced Performance for Magnetic Near-Field Measurements," in IEEE Transactions on Electromagnetic Compatibility, vol. 62, no. 5, pp. 1921-1928, Oct. 2020, doi: 10.1109/TEMC.2019.2946714.

[10] F. M. Greene, “Development of Electric and Magnetic Near Field Probes” National Bureau of Standards, Jan. 1975.

[11] M. Sahrling, “Layout Techniques for Integrated Circuit Designers” Artech House, 2022.

[12] “Controlled Impedance Routing,” Controlled Impedance Routing | Altium Designer 24 Technical Documentation,

https://www.altium.com/documentation/altium-designer/interactively-routing-controlled-impedance-pcb (accessed Apr. 24, 2024).

[13] Chimney Electronics, “Coplanar Waveguide With Ground Characteristic Impedance Calculator,” Apr. 2023.

[14] NewAE Technology Inc, “H Field Probe,” 2020.

[15] Z. Yan, W. Liu, J. Wang, D. Su, X. Yan and J. Fan, "Noncontact Wideband Current Probes With High Sensitivity and Spatial Resolution for Noise Location on PCB," in *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 12, pp. 2881-2891, Dec. 2018, doi: 10.1109/TIM.2018.2830859.

[16] HMC253AQS24E Analog Devices | Mouser, https://www.mouser.com/ProductDetail/Analog-Devices/HMC253AQS24E?qs=he1Eu8VmQT3P1fyM0szJtQ%3D%3D

[17] “EV1HMC253AQS24,” DigiKey, https://www.digikey.com/en/products/detail/analog-devices-inc/EV1HMC253AQS24/5403563

[18] “TB-409-52+,” Mini-Circuits, https://www.minicircuits.com/pcb/WTB-409-52+_P02.pdf.

[19] “Impedance Control,” PCBWay, https://www.pcbway.com/pcb_prototype/Impedance_Control.html (accessed Apr. 25, 2024).

[20] “RF CHOKE / SURF MOUNT / RoHS,” Mini-Circuits, https://www.minicircuits.com/WebStore/dashboard.html?model=ADCH-1220%2B

[21] “Stackup for 4, 6, 8,..., 18 layers multi-layer laminated structure,” Custom PCB Prototype Manufacturer, https://www.pcbway.com/multi-layer-laminated-structure.html

[22] “PCBWAY Manufacturing Capabilities,” Custom PCB Prototype Manufacturer, https://www.pcbway.com/capabilities.html

[23] “Blind and Buried Vias,” Sierra Circuits - Proto Express, https://www.protoexpress.com/kb/blind-and-buried-vias/#:~:text=These%20vias%20are%20meant%20for,mils%20(150%CE%BCm)%20or%20less. (accessed Apr. 25, 2024).

[24] “08051C104K4T2A KYOCERA AVX,” Mouser, https://www.mouser.com/ProductDetail/KYOCERA-AVX/08051C104K4T2A?qs=46Fswce4YXm7yQcAJaRRKg%3D%3D.

[25] “0731000114 Molex,” DigiKey, https://www.digikey.com/en/products/detail/molex/0731000114/1465147.

[26] “830066431 Wurth Elektronik,” Mouser, https://www.mouser.com/ProductDetail/Wurth-Elektronik/830066431?qs=2WXlatMagcFo8QBpzWAtGw%3D%3D

[27] “83614-9014 Molex,” Mouser, https://www.mouser.com/ProductDetail/Molex/83614-9014?qs=Qn3jS7NaQuCqKRx50FSkyw%3D%3D

[28] “C0603C103J5RACTU KEMET,” Mouser, https://www.mouser.com/ProductDetail/KEMET/C0603C103J5RACTU?qs=BimOss5pjlFJEfKhlb7g1g==

[29] “CRCW2512150RJNEG,” DigiKey, https://www.digikey.com/en/products/detail/vishay-dale/CRCW2512150RJNEG/1173421.

[30] “GRM1555C1H101JA01D,” DigiKey, https://www.digikey.com/en/products/detail/murata-electronics/GRM1555C1H101JA01D/3693829.

[31] “RASM722PTR13X SWITCHCRAFT,” Mouser, https://eu.mouser.com/ProductDetail/Switchcraft/RASM722PTR13X?qs=NW1tGwZwExFzXYAsdXkb2Q%3D%3D (accessed Apr. 25, 2024).

Appendix A

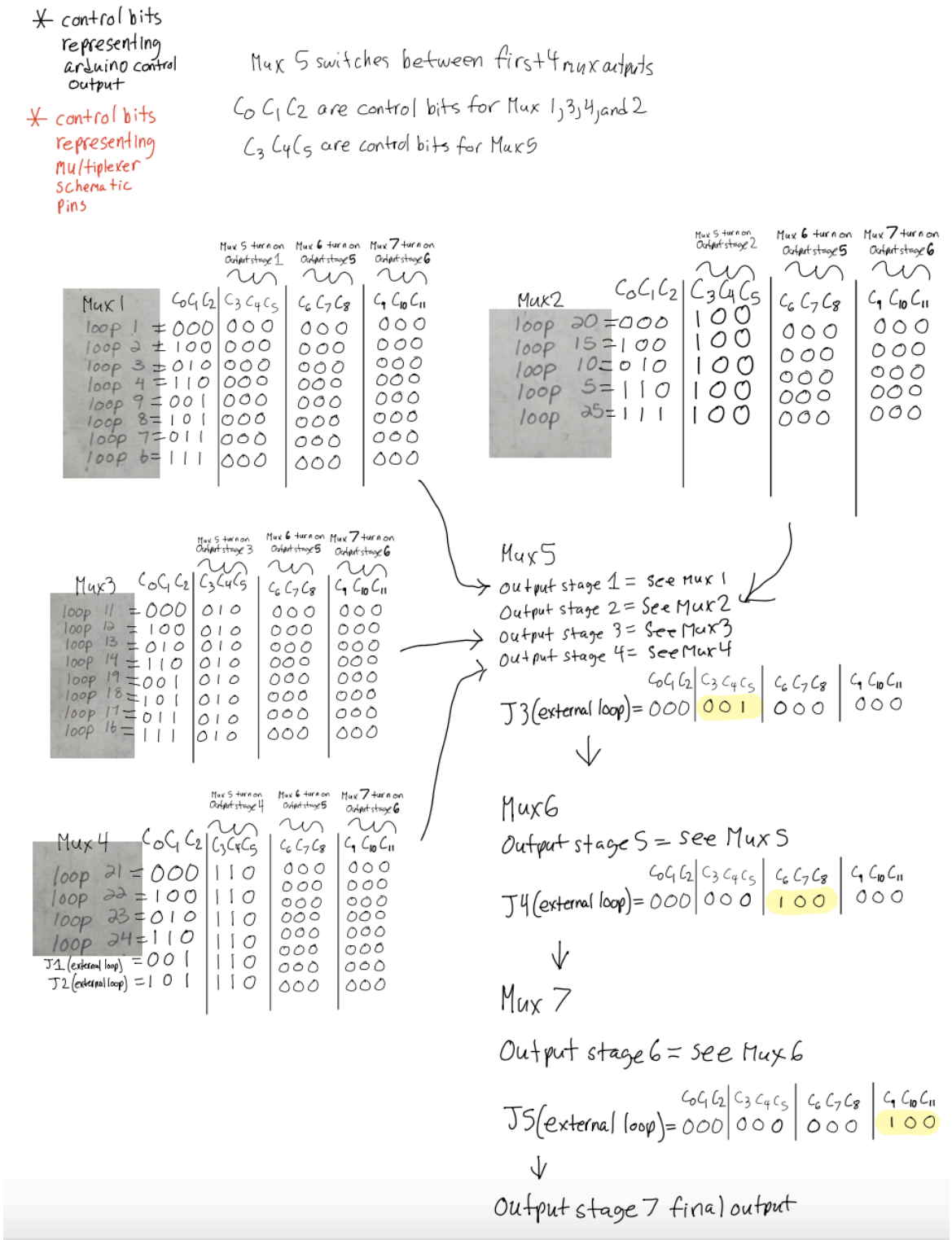

Figure X. Corresponding Binary Values to Mux Control Pins

Appendix B Arduino Code for Circling Through the Loops

```
// Define the number of pins you want to control
const int numPins = 12;

// Array to hold the pin numbers
const int pins[numPins] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13};

void setup() {
  // Set all pins as OUTPUT
  for (int i = 0; i < numPins; i++) {
    pinMode(pins[i], OUTPUT);
  }
}

void loop() {
  // Exciting Loop 1
  unsigned int binaryValue = 0b000000000000; // Binary value of loop 1

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 2
  unsigned int binaryValue = 0b000000000001; // Binary value of loop 2

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
```

```
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 3
  unsigned int binaryValue = 0b000000000010; // Binary value of loop 3

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 4
  unsigned int binaryValue = 0b000000000011; // Binary value of loop 4

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 5
  unsigned int binaryValue = 0b000000001011; // Binary value of loop 5
```

```
  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 6
  unsigned int binaryValue = 0b000000000111; // Binary value of loop 6

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 7
  unsigned int binaryValue = 0b000000000110; // Binary value of loop 7

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
```

```
}


void loop() {
  // Exciting Loop 8
  unsigned int binaryValue = 0b000000000101; // Binary value of loop 8

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 9
  unsigned int binaryValue = 0b000000000100; // Binary value of loop 9

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 10
  unsigned int binaryValue = 0b000000001010; // Binary value of loop 10

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
```

```
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 11
  unsigned int binaryValue = 0b000000010000; // Binary value of loop 11

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 12
  unsigned int binaryValue = 0b000000010001; // Binary value of loop 12

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}
```

```
void loop() {
  // Exciting Loop 13
  unsigned int binaryValue = 0b000000010010; // Binary value of loop 13

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 14
  unsigned int binaryValue = 0b000000010011; // Binary value of loop 14

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 15
  unsigned int binaryValue = 0b000000001001; // Binary value of loop 15

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
```

```
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 16
  unsigned int binaryValue = 0b000000010111; // Binary value of loop 16

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 17
  unsigned int binaryValue = 0b000000010110; // Binary value of loop 17

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 18
  unsigned int binaryValue = 0b000000010101; // Binary value of loop 18
```

```
  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 19
  unsigned int binaryValue = 0b000000010100; // Binary value of loop 19

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 20
  unsigned int binaryValue = 0b000000001000; // Binary value of loop 20

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }
```

```
  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 21
  unsigned int binaryValue = 0b000000011000; // Binary value of loop 21

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 22
  unsigned int binaryValue = 0b000000011001; // Binary value of loop 22

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 23
  unsigned int binaryValue = 0b000000011010; // Binary value of loop 23

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
```

```
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 24
  unsigned int binaryValue = 0b000000011011; // Binary value of loop 24

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Exciting Loop 25
  unsigned int binaryValue = 0b000000001111; // Binary value of loop 25

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}
```

```
void loop() {
  // External Loop on Mux 4
  unsigned int binaryValue = 0b000000011100; // Binary value of loop

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // Second external Loop on Mux 4
  unsigned int binaryValue = 0b000000011101; // Binary value of loop

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // External Loop on Mux 5
  unsigned int binaryValue = 0b000000100000; // Binary value of loop

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;
```

```
    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // External Loop on Mux 6
  unsigned int binaryValue = 0b000001000000; // Binary value of loop

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}


void loop() {
  // External Loop on Mux 7
  unsigned int binaryValue = 0b001000000000; // Binary value of loop

  // Iterate through each pin and set its state based on the corresponding bit in the binary value
  for (int i = 0; i < numPins; i++) {
    // Check if the i-th bit of binaryValue is set
    bool isBitSet = (binaryValue >> i) & 1;

    // Set the pin state accordingly
    digitalWrite(pins[i], isBitSet ? HIGH : LOW);
  }

  delay(10); // Delay for 0.01 second before updating again, adjust as needed
}
```

Appendix C

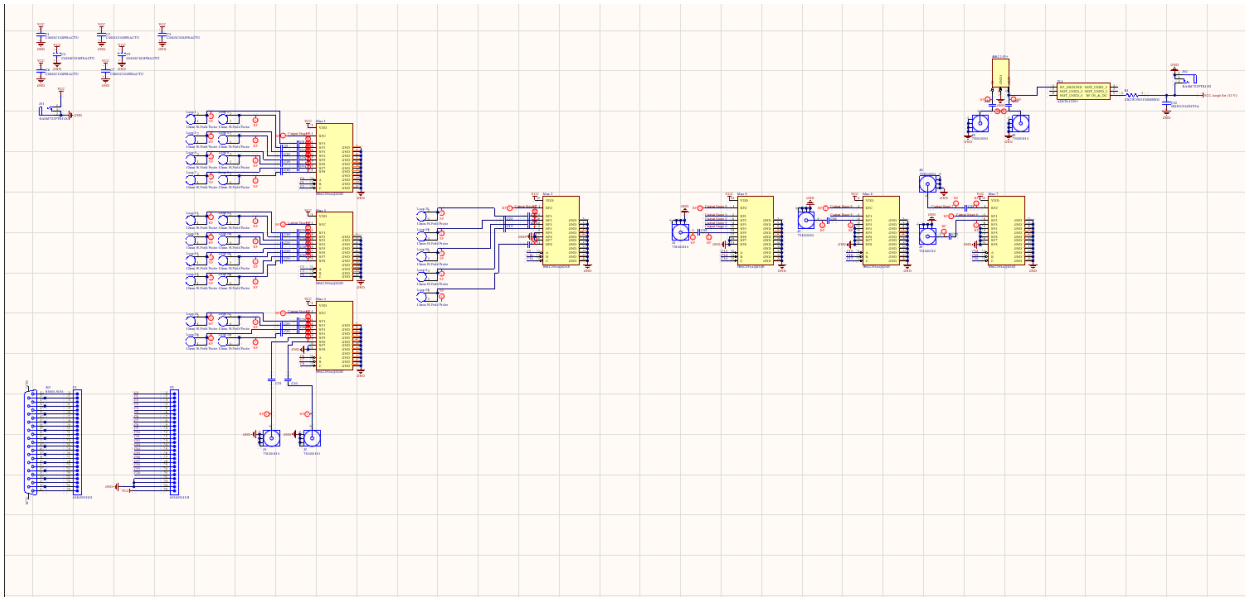

Figure 60. Prototype 5x5 Array Complete Schematic