E52 - Autonomous Magnetic Field Scanner

Final Report

May 2021

Table of Contents:

Abstract:
Collecting magnetic field data by hand is an arduous and time-consuming task that is also prone to human error while taking measurements. Drs. Ababei and Richie wanted a device that could autonomously record magnetic field data over a coil and then plot the data on a 3D scatter plot.
To accomplish this, the following was necessary: a CNC machine needed to be repurposed, a device needed to be 3D printed to hold a magnetic field sensing probe, a spectrum analyzer needed to interface with the probe, and then a Raspberry Pi needed to be able to control the motors and interface with the spectrum analyzer.

After discussion with our advisors, the final goal of this project includes one motor moving the sensor in the z-direction, using the spectrum analyzer to read the power output from the coil, and displaying the information on a 3D scatter plot. Removing hand measurements makes the process much less time-consuming, since it doesn't require anyone to be present while collecting data. Additionally, given the consistent motor movements and angle of the sensing probe, the measurements will be much more accurate.

The verification of all of these parts consisted of moving the motor with the Raspberry Pi, creating a GUI in python with a 3D plot, and using a python script to collect peak power data over a loop. All of these confirmed the possibility of using each specific library to create an autonomous field scanning device. The only remaining work was to interface every device from the Raspberry Pi. This is being worked on up until the day of the presentation.

Customer Needs:

For determining the customer needs, Drs.Ababei and Richie were consulted biweekly to assess their requirements for the electromagnetic sensor. Within the first few meetings, the primary objective was defined as repurposing a CNC machine that holds a magnetic field sensing probe using a 3-D printed part, detects the strength of the magnetic field, and plots the strength of the field on a 3-D scatter plot on an interactive GUI. Each measurement would be taken every half centimeter as specified by Drs. Ababei and Richie. It was necessary to automate this data collection process as that is the main characteristic which differentiates it from other electromagnetic sensors on the market. Additionally, Drs.Ababei and Richie decided that the sensing probe would have to measure the magnetic field strength over a 3.5” by 3.5” area. The CNC machine on hand has the range desired in the x and y directions, however, in the z-direction the range falls short (1.8” to the desired 3.5” =) [1]. With the laser engraver left unattached, the range increased significantly to 3”. The plan was to have the bottom part of the enclosure that holds the CNC laser engraver sawed off, while preserving the rails it revolves around to add more mobility. Potential modifications for the near future included a longer stainless steel threaded rod to move the sensor up and down through the proper range. Other less important needs included a short duration time for measuring the magnetic field and generating the graph, varying the sampling rate the probe is taking measurements at, and plotting the points of the magnetic field strength on a 3-D plot while the machine was running. The full list can be seen below in **Table 1.**

Table 1: Customer Needs with Importance

Need	**Description**	**Importance (*--*****)**
1. Automated	Hands free	*****
2. Scan b-field	Measure magnetic field-convert power from sensor and analyzer to magnetic field	*****
3. Plot data	Put data points into 3D plot	*****
4. Connect computer	USB Serial Connection	***
5. Real time plotting	Plot points as they are scanned	*
6. Vary sampling rate	Change how often the sensor takes a measurement	**
7. Use of spectrum analyzer	Connect sensor to analyzer	****
8. 3.5”x3.5”x3.5”	Dimensions	*****
9. Less than 10 min	Does not take too much time	**

The requirements for the customer needs changed during the second semester as time was limited and unexpected obstacles emerged while designing the product. These obstacles included frying components when attempting to make the motors move, altering the method used to move the motors, and obtaining access from Tektronix to download 64-bit API installation on a 32-bit linux Raspberry Pi to be able to test the code for the GUI. The way the method was altered for running the motors will be discussed in the next paragraph. Initially, Drs.Ababei and Richie desired that the machine moved in 3-directions and covered a 3.5” by 3.5” area for taking magnetic field strength measurements. This requirement changed to only having to move 2.5 cm in the z-direction. Our advisors determined that it was most important to measure the field strength in the z-axis, and it was not worth it to alter the original design of the CNC machine in order to measure another half inch. Additionally, Drs.Ababei and Richie came to the decision that it was not very important to have real time plotting for the CNC machine as long as the data collection process occurred in a timely manner.

In order to accomplish these new requirements, a strategy had to be discussed with Drs.Ababebi and Richie. It was decided that after the CNC machine was built, the CNC engraver motor would be removed. Therefore, a 3-D printed part could be created to fit there. This 3-D printed part would hold the magnetic field sensing probe in place with four stainless steel screws and nuts. Drs.Ababebi and Richie emphasized the fact that it was essential for the 3-D printed part to hold the probe firmly in place to prevent movement and to keep the probe straight while collecting data. Otherwise, the values obtained from the probe would be inaccurate. Also, the plan to run the motors was to initially use the default microcontroller the CNC machine came with. This would prove to be more difficult than expected, due to copyright issues, so new microcontrollers were considered to replace the default microcontroller. After debating the use of a Raspberry Pi or arduino, it was decided that a Raspberry Pi would be used with a python library for moving the motors. The initial plan was to use a RAMPSv1.4 board and ADS1115. However, certain components on the RAMPSv1.4 board stopped working, so the solution was to use a stepper motor driver, with the Raspberry Pi, to run the motors. The stepper motor driver selected was A4988, and the library that was picked was RpiMotorLib [2]. Lastly, a plot of the data collected by the moving sensor would be accomplished with an interactive GUI, that could upload past plots, save plots, and start the collection of data with the 3D field scanner. All of the GUI, CNC movements, and data collection were coded with python. The GUI and plot were accomplished using python’s PyQt5 library and matplotlib library respectively, while the CNC movements and data collection were performed by sending step commands from the RPiMotorLib.

Economic Analysis:

Due to Drs. Richie and Ababei already having a spectrum analyzer, DC power supply, and CNC machine, purchasing another device to collect magnetic-field data would be expensive and unnecessary. Additionally, there are no autonomous and moving magnetic-field data collection systems, so repurposing the CNC machine would still be needed. The proposed design cost is shown in **Table 2.** The cost does not include any of the supplies given to the project team by the advisors, because this design is for use in a Marquette Lab and not for commercial sale. The total price is roughly $ 90, and this includes a CNC kit where only the A4988 drivers ended up getting used in the final design.

This design was compared to competitors that could record magnetic field data. These competitors included BMP with their magnet field position sensor and Infineon with the XENSIV (TLE 493D). The Infineon XENSIVE comes with a board containing the sensor and an Arduino library, and it can sense a magnetic field up to 160 mT. The product cost $ 45. The BMP sensor can measure the magnetic field within a 6.3 inch range, has a 12-bit resolution, and has the following dimensions: 17.5 mm by 9.6mm by 170mm. The BMP product costs $ 430 [4]. However, neither of the competitors had an automated process with consistent movement and a probe held in place, so repurposing the CNC machine would still be needed for both products. The repurposed CNC machine is attempting to automate the process, so the cost from repurposing the CNC machine was added onto both. Both competitors would not be able to fit the design specs, due to the inability to manipulate data, without hacking the copyrighted systems. This data would be necessary for graphing in the GUI. The competitors were compared to the proposed design in **Table 3.** The comparison of these options solidified moving along with the proposed design, because it was less expensive than both competitors and gives more versatility by not fighting copyrighted devices and hacking software.

Table 2: Cost of Proposed Design

Item	Price
Raspberry Pi	35
Screws/Bolts	17.34
Board/Drivers	40 (board not used)
Total	$ 92.34

Table 3: Comparison Proposed Design/Competitors

	Proposed Design	BMP magnet field position sensor [6]	Infineon XENSIV
Price	$ 92.34	$ 487.54	$ 102.34
Can Satisfy Requirements?	Yes	no	no

Design Risk Analysis:
Table 4: Risk Analysis with Mitigation Strategies

Risk:	**How to Mitigate the Risk:**
Human contact to live circuitry and exposed wires.	Do not touch circuitry while power is supplied/Turn off power supply before touching circuit
Overheating the Raspberry Pi/circuit	Keep current supplied below 1A/power circuit with only 5-6V
Magnetic field sensing probe moving during data collection and not sitting straight within the 3-D printed probe holder.	Tighten the stainless steel nuts after getting probe situated into proper orientation every time before use
Metal threaded rod does not rotate with the motors running	Use an allen key and tighten the mini screws attaching the rod to the motors before use

Table 4, shows the potential risks that can jeopardize the ability for the project to work and harm users. Whenever there is live circuitry, there is a risk due to the current running through the conductors from the power supply. A mitigation tactic is to keep the current below 1 Amp. Generally, it is also advised to not touch the circuitry while the power supply is turned on. Other risks concern the functionality of the machine running. The Raspberry Pi and circuit board can overheat; therefore, the voltage and current should be kept below 6 Voltz and 1 amp respectively. Additionally, the motor runs smoother and makes less noise at a lower voltage. The last two risks deal with mechanical components on the machine. The magnetic field sensing probe must be kept straight and cannot move during use. Everytime someone uses the machine, the probe should be placed in the correct orientation, and the stainless steel nuts should be tightened. Also, the metal threaded rods should be tightened down with an allen key before use. If not, the user runs the risk of the metal threaded rod not rotating. This will cause the machine to be not able to move in that respective direction.

In regards to applicable standards, regulatory requirements, and legal issues, there are none. This project is for research purposes and uses open source code.

Design Documentation:

As described in the customer needs section, instead of treating the CNC machine as one whole system, the motors were treated as separate entities. With running G-code in python being problematic and only needing to move the magnetic field sensing probe in the z-direction, using only the step commands was perfect. The setup is described below, in addition to a wiring diagram and driver setup. The setup for the Raspberry Pi is below as well with all of the steps to download the dependencies. The Spectrum Analyzer interfacing with the Raspberry Pi does not have exact install instructions, just set-up suggestions because this was not completed by the group at the time of the write up. Finally, where to find the CNC machine build instructions and the description of the probe holder is below with figures of the final design.

Motor Wiring:

Before using the A4988 stepper motor driver, it is important to set the reference voltage of the motor driver to ensure it is operating within the proper parameters. From the A4988 datasheet, the reference voltage needs to be about 0.96V [7]. The reference voltage can be set by giving power to the stepper motor driver from the power supply and then using a multimeter to turn the screw on the driver to the desired value. This can be done by attaching the positive lead of the multimeter to a screwdriver to rotate the screw, while the negative lead is attached to ground. As the screwdriver turns, the value should change depending on the orientation. Once the stepper motor is set to the correct reference voltage, it will be safe for use in the circuit.

For wiring the motor to the Raspberry Pi, the main components needed are the Raspberry Pi, A4988 stepper motor driver, stepper motor, and a capacitor value of 47pF or higher. The A4988 motor driver will be connected at the MS1, MS2 and MS3 pins to the Raspberry Pi pins GPIO 14, 15 and 18, respectively. The VDD and GND pins of the A4988 will also be connected to the Raspberry Pi to prevent a floating ground. The STEP and DIRECTION pins will be connected to GPIO pins 21 and 20, respectively, to allow the commands for the motor directions and number of steps. A DC power supply will also be connected to the A4988 at the VMOT and GND pins to power the stepper motor driver. The capacitor will be connected between the power supply and the motor driver unit to help protect from power surges. The sleep and reset pins will be connected together as well. The pins connecting to the 4-pin motor are the 2B, 2A, 1B, and 1A pins, with 2B and 2A grouped together for one inductor within the motor, and 1A and 1B grouped together for the other inductor. The Raspberry Pi's GPIO pinout is displayed in **Figure 1**, with the wiring schematic given in **Figure 2**. For the value of the DC power supply, 6V is recommended, with 5V working well and making the motors quieter. When the circuit is connected, the current at 6V may approach 1A. During this time, lowering the power supply to 5V gives a lower current with the motor still operating.

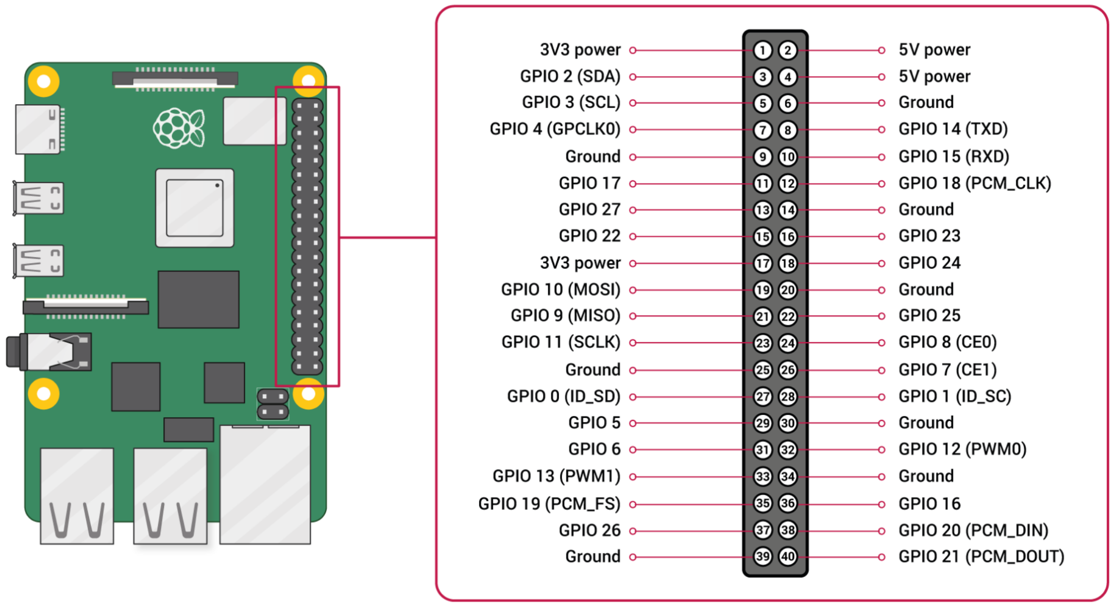

Figure 1: Raspberry Pi GPIO Pinout [8]

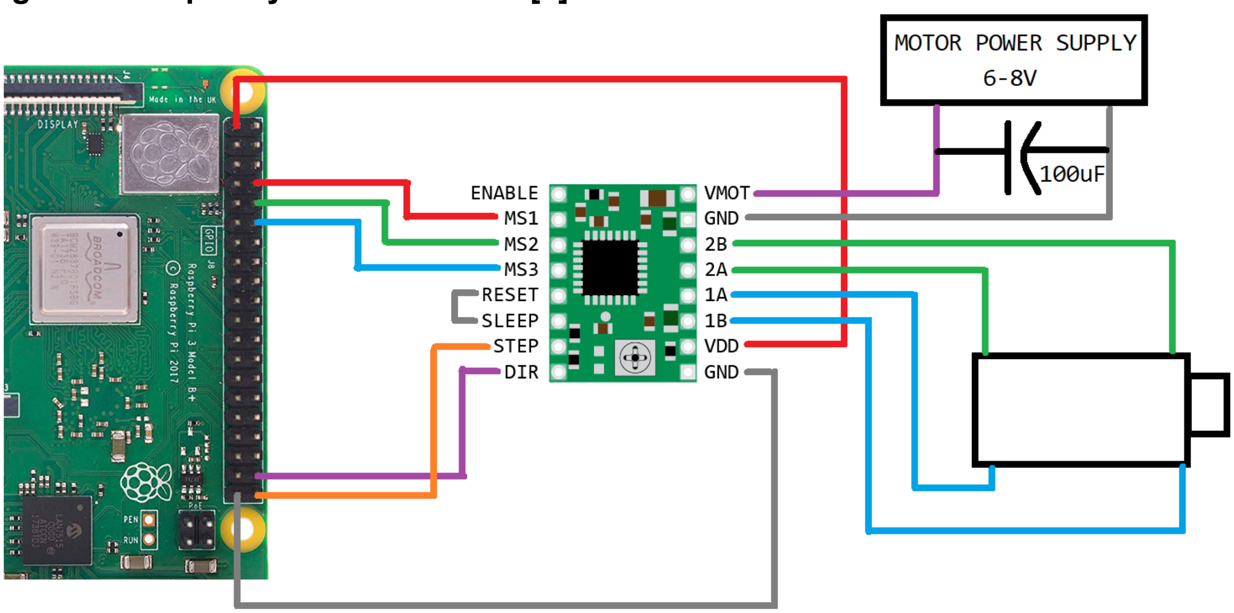

Figure 2: Motor wiring schematic [9]

Raspberry Pi:
*Open a terminal through the GUI or by using keyboard command "CTRL+ALT+T"
*All italicized phrasing is a command to execute in the Raspberry Pi's linux terminal.
Go to github.com/ajbetancourt/E52-Senior-Design and download the zip file containing all the code and extract to the desired location.

<u>Setting up virtual environment:</u>

- Use command:
    - *python --version*
- **If python 3.X.X is installed skip to next section: *Downloads***
    - The GUI library is dependent on python 3 being installed so if it is already installed there is no need for a virtual environment
- If python 2.X.X is installed a virtual environment with Python 3 is required, because there
- First download Python 3:
    - *sudo apt-get update*
    - *sudo apt-get upgrade*
    - *sudo apt-get install python3*
- Check installation success
    - *python3 --version*
    - Should output python 3.X.X
- Create virtual environment called virtEnv for python3
    - *sudo apt-get install python3-venv -y*   ## install a virtual environment
    - *python3 -m venv ~/virtEnv   ## create a location for virtual environment*
- **Activate virtual environment (use this command first any time you want to run the GUI)**
    - *$ source ~/virtEnv/bin/activate*
- All downloads should be downloaded into this virtual environment to ensure that the libraries are all added to Python 3, and not into python 2. Note: there are currently two versions of python on the Raspberry Pi.

<u>Downloads:</u>

- Make sure you are in the Python 3 virtual environment in the Raspberry Pi terminal, it should look like this:
  "(virtEnv) pi@raspberrypi:~/currentFile "
- Install pip
    - *sudo apt install python3-pip*
- numpy download:
    - *pip3 install numpy*
- matplotlib download:
    - *python -m install -U pip*   ##make sure the pip is updated
    - *python -m pip install -U matplotlib*

- pyqt5 download [10]:
    - *python -m pip install pyqt5*
    - There is a bug with the pip install of this library, you will need to manually install PyQt5 and SIP from the source provided by pypi.org and riverbankcomputing.com, respectively. Then manually install these files. Instructions for manual installation can be found on the PyQt5 manual [10]
    - There are a few ways errors have been solved that have been posted online, but the manual installation
- RPiMotorLib:
    - *sudo pip3 install rpimotorlib*
- RPi.GPio
    - *pip3 install RPi.GPIO*

Spectrum analyzer API setup windows:

- Dr. Richie can provide the api installation .exe file
- When downloaded open the .exe file and follow the wizard
- The api's download will allow for interfacing between your pc and the spectrum analyzer

Spectrum Analyzer setup linux:

- Need at least cyusb.conf downloaded
- Possible way one: If you can afford to spare the ram -> a 64-bit OS is recommended to load onto the Raspberry Pi (ideally Ubuntu)
- Data can be transferred serially as well with the USB connected from the RSA to a PC, then using the Raspberry Pi to call a function, in the pc, then grab the data

Python Commands for Certain Tasks:

- Whole project
    - *python main_gui.py*
- Motor movement
    - *python test.py* ## this is the most basic motor movement file you can build with
- Motor movement 2
    - *python zaxistest.py*
- GUI on its own
    - *python main_gui_backup.py*
- Spectrum analyzer (if api is installed)
    - *python peak_power_detector.py*
- Spectrum analyzer with plot of span
    - *python peak_power_with_plot.py*

Spectrum Analyzer Settings in Python Script in the main function [2]

- In the peak_power_detector.py or peak_power_with_plot.py

- Center frequency: 4.2MHz
    - *rsa.CONFIG_SetCenterFreq(cf)*
- Resonant Bandwidth = 30kHz
    - *specSet.rbw = c_double(30e3)*
- Video Bandwidth = 30kHz
    - *specSet.vbw = c_double(30e3)*
- Trace Length = 801
    - *specSet.traceLength = c_double(801)*
- Span = 1 MHz
    - *specSet.span = c_double(1e6)*

Probe Holding Design:

The 3-D printed part that holds the magnetic field sensing probe was designed in Solidworks. There are two parts in the design that act together to hold the probe. There are four holes to secure the top piece onto the bottom piece.These holes are meant for ¼" inch, 2-inch long stainless steel bolts and can be secured using ¼" inch stainless steel nuts. The design is based on the diameter at three points on the probe. The center ring on the grip part of the probe was measured to be .752". The other two measurements were taken .86" away from the center ring in both directions and were found to be .654" and .752". The spline command on Solidworks was used to connect these points to mimic the shape of the curve on the probe. Also, the 3-D printed part was made to fit where the CNC engraver motor was meant to be, using a cylindrical design.The inner diameter of the plastic piece meant to hold the CNC engraver was measured to be 1.72". A height of 1.5" was selected for the vertical distance of the cylindrical portion to ensure the 3-D printed part would not slip out. Additionally, the thickness between the inner and outer diameter of the plastic piece was measured as well to ensure it would not interfere with the screw holes. It was measured to be .14", so the design includes a .25" distance from the edge of the cylindrical portion. The dimensions for the design can be seen below in **Figures 3** and **4.**

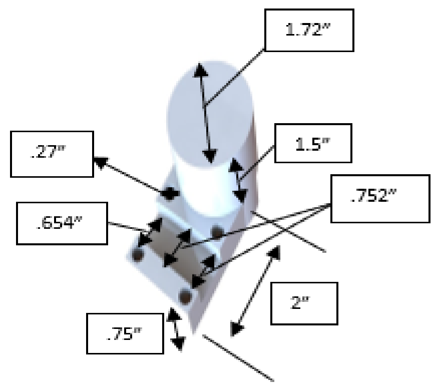

Figure 3: Bottom Component of 3-D printed probe holder

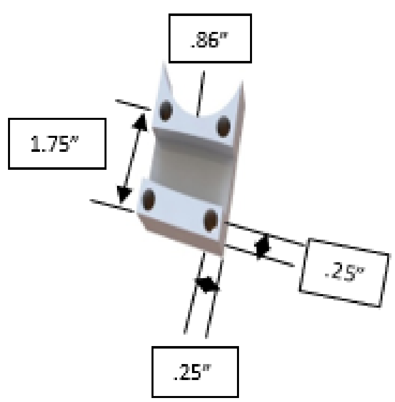

Figure 4: Bottom Component of 3-D printed probe holder

CNC machine:

The CNC machine seen built in **Figure 5** was assembled by using the 3017-Pro User manual [11]. The only major changes in this design were removing the original microcontroller to use a raspberry pi and replacing the engraving system with the 3D printed probe holder. Otherwise, the design follows this manual up to Step 10, Control Board Installation.

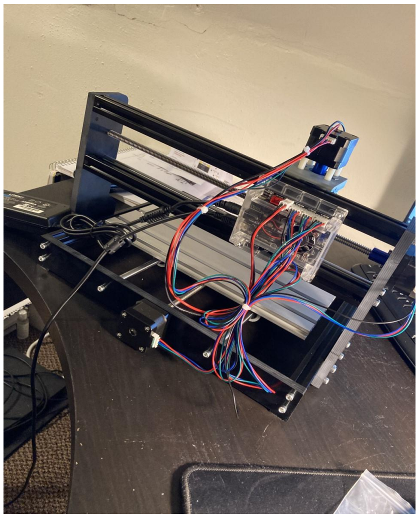

Figure 5: CNC machine assembly

Technical Justification:____

Motors:

______The first part of testing was for the NEMA17 stepper motor. The motor was used with a Raspberry Pi and A4988 stepper motor driver to control the movement of the motor in the z-direction. The library used to control the motors is the RPiMotorLib[2]. The first step in motor testing found the relationship between the number of steps inputted into the code and the distance the probe traveled. A guess and check methodology was used to find the appropriate step number, as Drs. Ababei and Richie specified that the CNC machine should travel in half centimeter increments. The ability for the motor to travel up and down the z-axis consistently needed to be tested as well. This was controlled for each step by entering a"true" or "false" in the code, which caused the motor to rotate clockwise or counter-clockwise respectively. The steps tested started at 470 steps in both the positive and negative directions of the z-axis and varied based on the results, in order to obtain a half centimeter displacement. The test setup can be seen in **Figure 2**. A ruler stood parallel to the probe on the CNC machine with tape holding it in place. The initial position was recorded, so therefore the distance traveled was found by recording the position that the probe traveled to after a certain amount of steps. If the change in position was more than a half centimeter, the number of steps was decreased, and if the change in position was less than a half centimeter, the number of steps was increased. This process continued until acquiring a half centimeter displacement.

Table 5: Position of the motor at different step values

step+direct	Trial 1 start	Trial 1 finish	Abs val movement
470 up	4.2 cm	3.2 cm	1 cm
470 down	3.2 cm	4.2 cm	1 cm
400 up	4.2 cm	3.4 cm	0.8 cm
400 down	3.4 cm	4.2 cm	0.8 cm
300 up	4.2 cm	3.55 cm	0.65 cm
300 down	3.55 cm	4.2 cm	0.65 cm
255 up	4.2 cm	3.7 cm	0.5 cm
255 down	3.7 cm	4.2 cm	0.5 cm

Spectrum Analyzer Sigma Vu:
______The second part of testing was for the spectrum analyzer. Tektronix RSA 607a, a spectrum analyzer, and probes were tested with a PC using both Signal-VU and a python script. Signal-Vu is a software by Tektronix that displays the peak frequency and power of a probe in real time. The Signal-Vu was given values for center frequency, video bandwidth, and resolution bandwidth. The video bandwidth (VBW) and resolution bandwidth (RBW) were varied for three trials: The first trial had a RBW and VBW of 25 kHz, the second trial had a RBW and VBW of 30 kHz, and the third trial had a RBW of 25 kHz and a VBW of 0 kHz. The RBW and VBW for each trial can be seen in **Table 6**.The constants included a center frequency of 4.2 MHz and a span of 7.5 MHz, starting from 450 kHz ranging to 7.50 MHz. After entering the RBW, VBW, and the constants, the script searched for peak power and the frequency it occurred at. Then it graphed the power in a bode plot over a specified frequency span. The goal was to find the settings that gave the cleanest graph results, because if the signal from the magnetic field was weak, noise would give false readings. The setup can be seen in **Figures 6-8**.

Table 6: Values of RBW AND VBW for Spectrum Analyzer Testing

Trial #	Resolution Bandwidth (RBW)	Video Bandwidth (VBW)
1	25 kHz	25 kHz
2	30 kHz	30 kHz
3	25 kHz	0 kHz

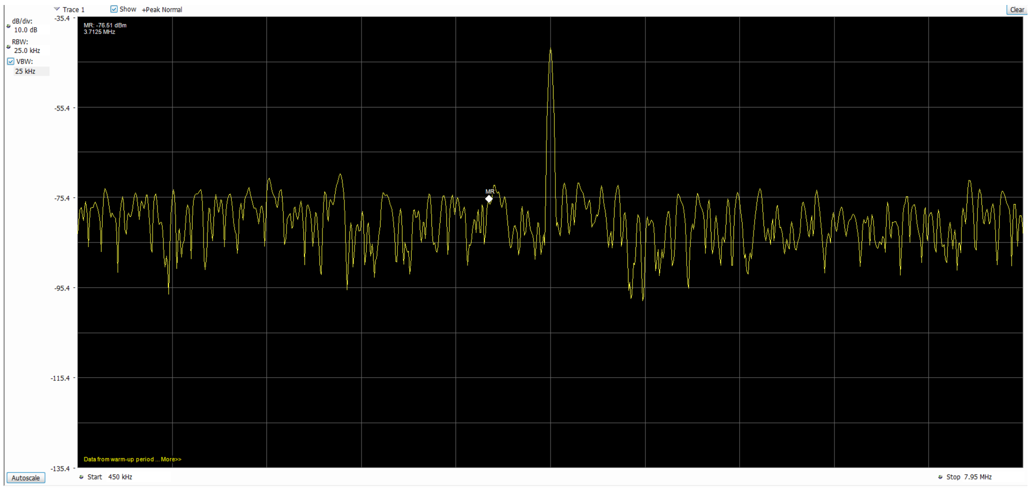

Figure 6: Spectrum Analyzer Sigma-Vu reading 1

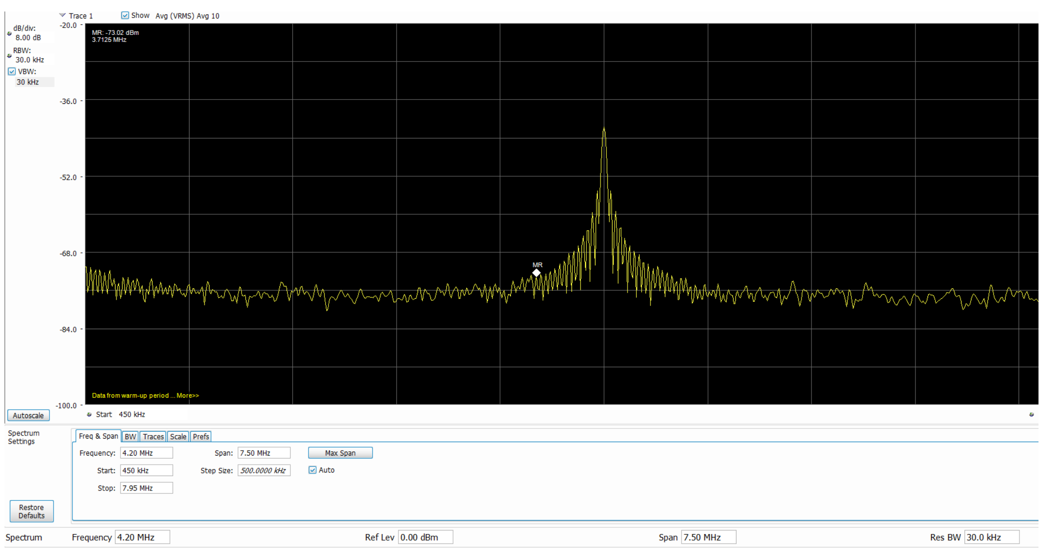

Figure 7: Spectrum Analyzer Sigma-Vu reading 2

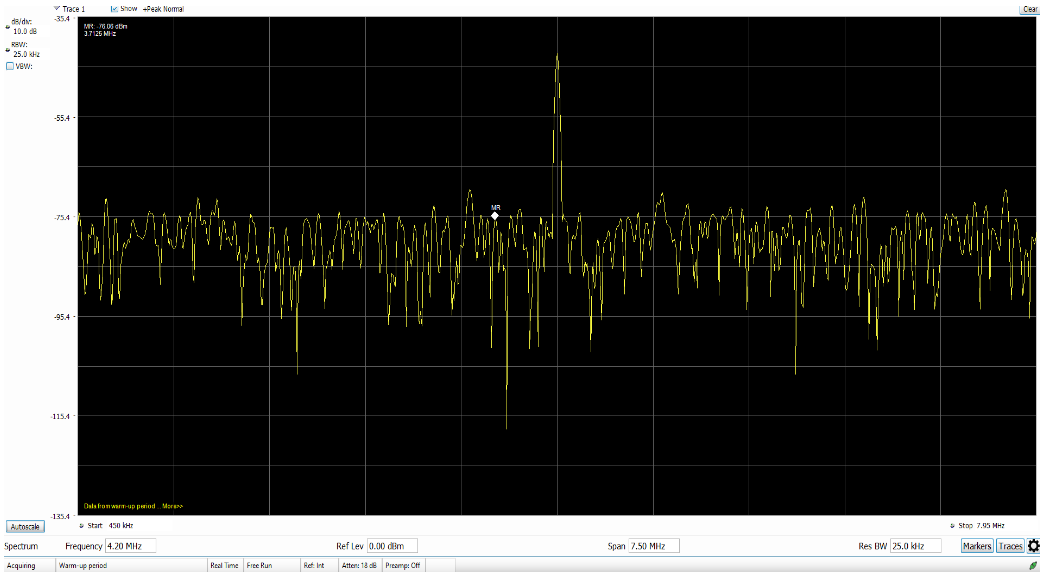

Figure 8: Spectrum Analyzer Sigma-Vu reading 3

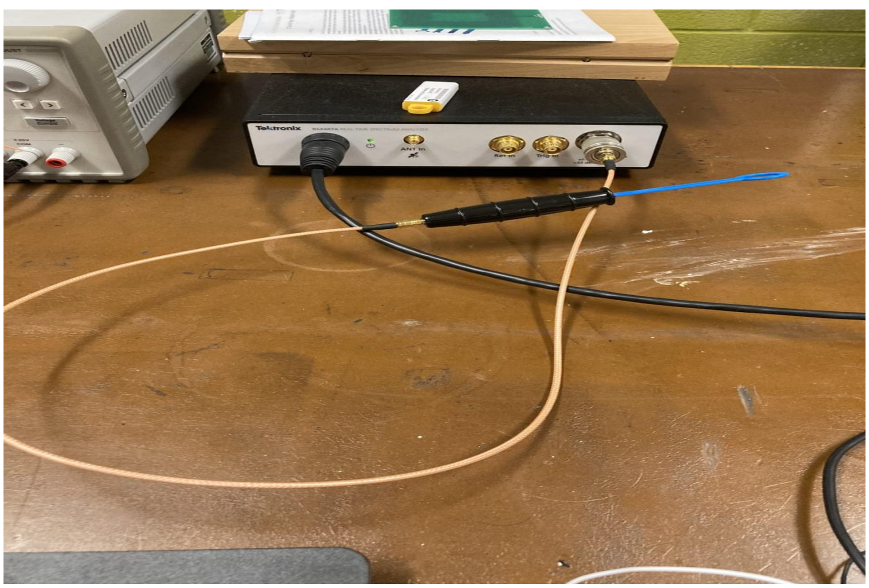

Figure 9: Spectrum Analyzer with magnetic field sensing probe attached

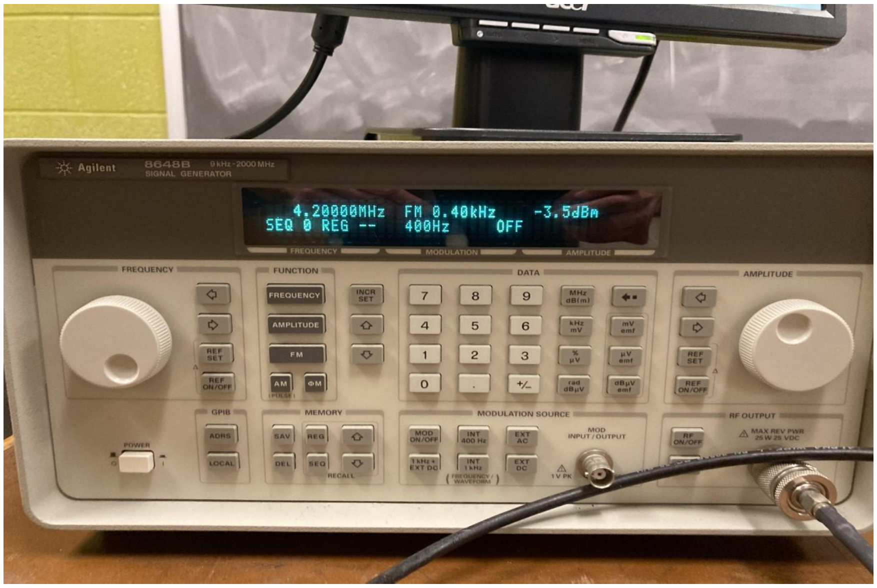

Figure 10: Agilent 8648B Signal Generator at 4.2MHz

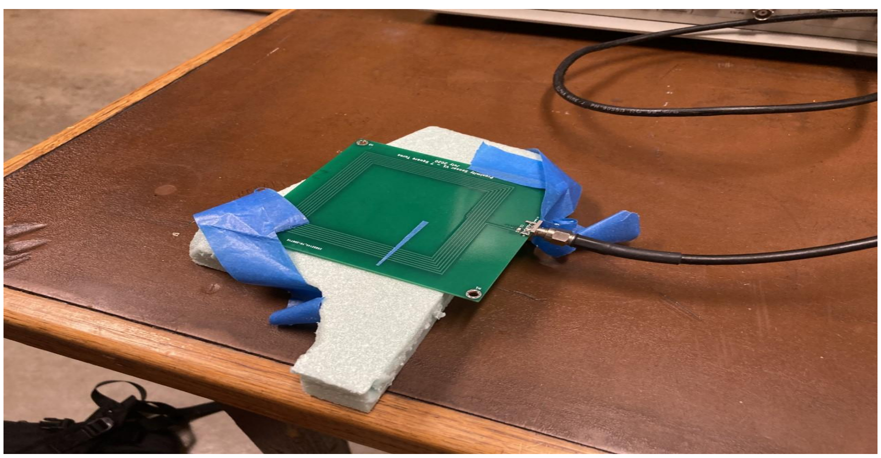

Figure 11: Magnetic-field producing coil

Spectrum Analyzer Python Code:

For the third part of testing, python scripts were used to obtain the peak power on a bode plot using matplotlib. The plot displayed the peak power on a bode plot. It also had an integer value in the file named peak power that could be returned in a function from that file, which was then sent to an array in the main file and graphed in the GUI's scatter plot [2]. There were 5 trials for testing. The resolution bandwidth and video bandwidth were varied each trial starting from 15 kHz going to 35 kHz. The RBW and VBW for each trial can be seen in **Table 7**. The center frequency was kept constant at 4.2 MHz to keep consistent with testing from the spectrum analyzer. There were a few other trials to see the effect of lowering the RBW and VBW on the results.

Table 7: Values of RBW AND VBW for Matplotlib Testing

Trial #	Resolution Bandwidth (RBW)	Video Bandwidth (VBW)
1	15 kHz	15 kHz
2	20 kHz	20 kHz
3	25 kHz	25 kHz
4	30 kHz	30 kHz
5	35 kHz	35 kHz

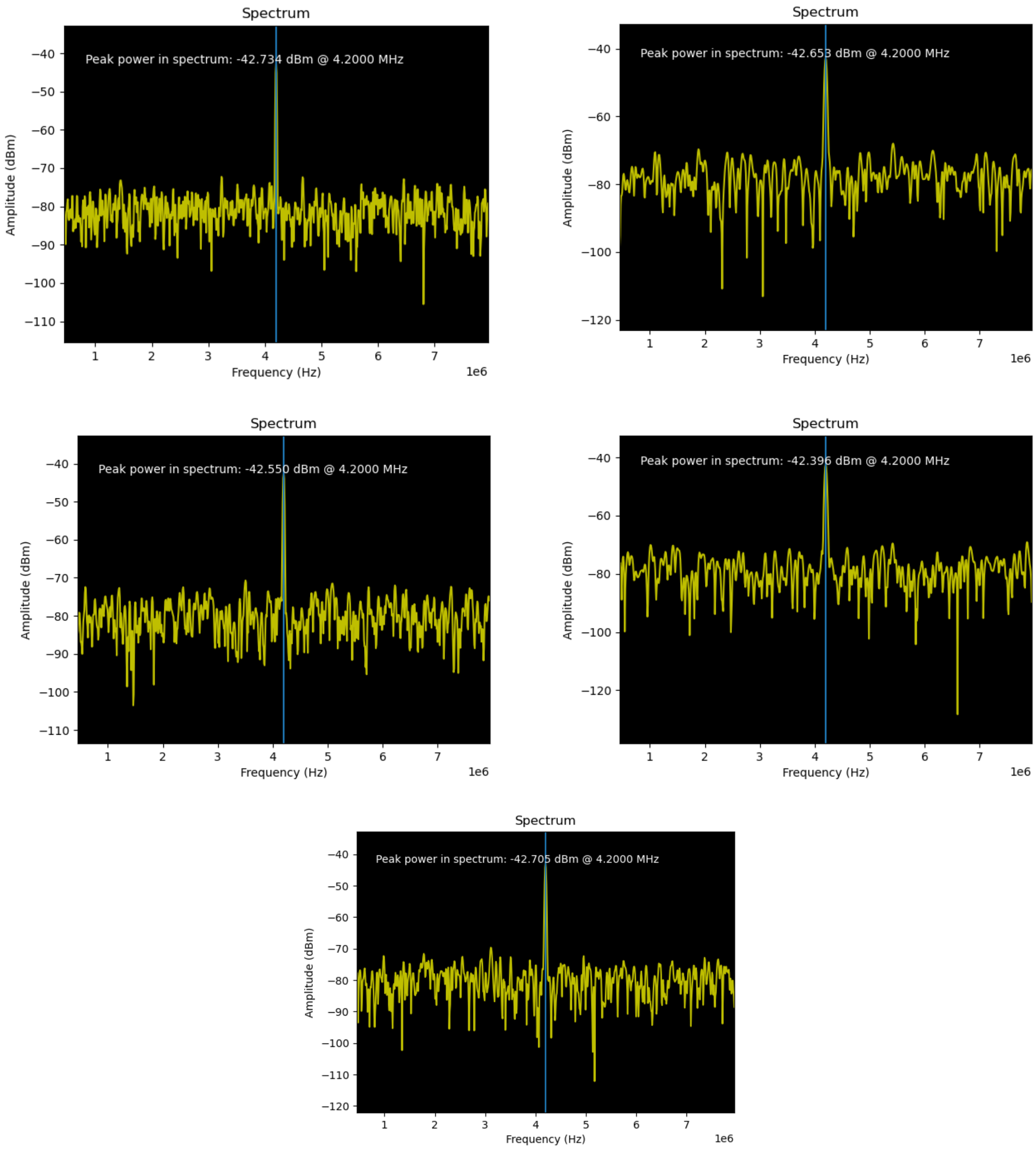

Figure 12: Python Code matplotlib max power

GUI:

The GUI consists of a 3D plot, a button to run the machine, motor calibration buttons, move motor up or down, and given the libraries used has the ability to add save, and upload features as well. To test the gui a randomly generated plot was inputted, and a few buttons were added with no added functionality, to show there is room for additional function.

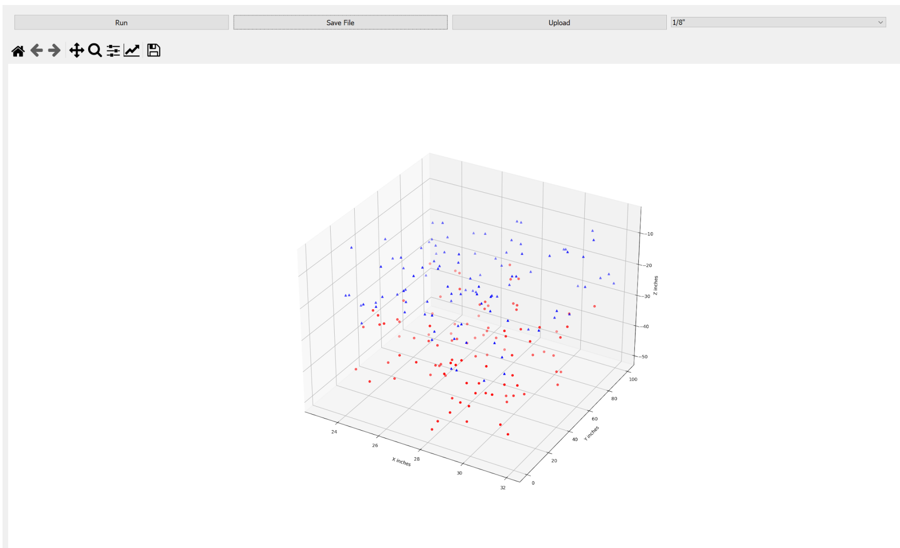

Figure 13: GUI with random plot, 3 buttons, and increment selection

<u>Testing Summary:</u>

The requirements our advisors wanted included moving the magnetic field sensing probe in the z-direction in half centimeter increments in both the positive and negative direction, and the ability to accurately calculate the magnetic field when using the spectrum analyzer. After testing for the correct amount of steps, the motor was able to move the probe a half centimeter when inputting 255 steps, and the probe moved both up and down after inputting “true” for up and “false” for down in the code. Additionally, the results obtained from testing with the spectrum analyzer, seen in **Figures 6** and **7**, show that trials 1 and 2 had appropriate RBW and VBW values for testing as noise was mitigated well along with the spike occurring near the center frequency of 4.2 MHz. The third trial, seen in **Figure 8**, would not be good for testing as some points other than the center spike varied by over 40 dBm from steady state. The python script results, seen in **Figure 12**, are similar to the results from sigma-vu and prove that the script is a viable way of getting peak power. The spike occurred near the center frequency of 4.2 MHz as well and noise was minimal for the 25 and 30 kHz trials.The python script results also showed that the RBW and VBW could even be lowered to 15 kHz and still obtain similar results.

Project Legacy:

This project consisted of using python to interface with the following: a spectrum analyzer to read power induced in a loop by a magnetic field, a Raspberry Pi to create a GUI with a 3D plot, and the same Raspberry Pi to interface with the motors via the onboard GPIO pins. After experimenting with quite a few libraries for both the motor control and total CNC movement, the team learned that for our application, treating the motors as individual entities worked best for our timeframe and allowed for experimentation in one direction with the RPiMotorLib. This library's control of the movements was validated, and the team then experimented with step sizes for our incrementation value, ½ cm. In regards to the GUI, research was done on different GUI libraries, and the PyQt5 library was found to work best for our application, as it was capable of fulfilling all of the customer needs without being too complex or focused on aesthetics.

Since all the initial requirements were not expected to be met, Drs.Ababei and Richie decided to extend this project another year. Next year's group will be tasked with making the data collection 3-dimensional, including modifying the CNC machine to be able to move 3.5 inches in the z-direction. This can be done the same way as the z-direction motor, but it will require more nested-for- loops in the "run.py" file. Additional drivers and GPIO pins on the Raspberry Pi will also be needed. The CNC machine can be altered to move 3.5 inches in the z-direction by cutting off the bottom plastic part on the CNC laser motor holder and adding longer stainless steel rods. Additionally, the next group will have to finish interfacing the spectrum analyzer and Raspberry Pi, include the ability to start the entire data collection process with a click of a button, and make it possible for data collection to be displayed on the 3-D plot during the process.

To interface with the spectrum analyzer, there will either need to be an additional PC used for measurement or a 64-bit OS uploaded on the Raspberry Pi (Kernel or full upload). A Raspberry Pi 4 with at least 4GB of ram is suggested. An installation of the Linux API of the spectrum analyzer will need to be downloaded with the instructions inside the README.txt file. This file is located inside the downloaded tar file from the flashdrive Dr. Richie has, or from Tektronix website. After this is done, the remaining task is to change the location of the API in the "peak_power_detector.py" file to the location of the API installation. Then this data can be collected and plotted in real time after the "run.py" file updates the "plot_points()" function in the "main_gui.py" file, since the points are currently hard coded in.

It is then up to the students to add additional features to the project. Some possibilities that our project team discussed are varying the increments with a text box or drop-down menu, changing color with magnitude of the magnetic field, saving the plots as .csv or text files, and having the ability to upload files as well. The work remaining in this project is interfacing, programming, along with minor wiring and legacy group has 3 members, all of the ECE students.

References:

[1] “GP,” *Amazon*, 2011. [Online]. Available: https://www.amazon.com/gp/product/B07RNZS3YD/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8. [Accessed: 10-April-2021].

[2] “Gavinlyonsrepo/RpiMotorLib”, *GitHub*, 2018. [Online]. Available: https://GitHub - gavinlyonsrepo/RpiMotorLib: RpiMotorLib : Raspberry Pi Motor Library : A python 3 library for various motors and servo s to connect to a raspberry pi [Accessed: 4-April-2021].

[3] ”RSA600A Series Laboratory Spectrum Analyzer Datasheet”, *Tektronix*. [Online]. Available: https://www.tek.com/datasheet/spectrum-analyzer-1 [Accessed: 7-April-2021].

[4] I. T. AG, “S2GO_3D_TLI493DW2BW-A0,” *Infineon Technologies*. [Online]. Available: https://www.infineon.com/cms/en/product/evaluation-boards/s2go_3d_tli493dw2bw-a0/. [Accessed: 14-Oct-2020].

[5] “Linear Hall Effect IC Magnetic Sensors,” *GMW Associates*, 27-Apr-2020. [Online]. Available: https://gmw.com/product/linear-hall-effect-ics/. [Accessed: 14-Oct-2020].

[6] “Product,” *Balluff*. [Online]. Available: https://www.balluff.com/local/us/productfinder/product/?key=BMP0009. [Accessed: 14-Oct-2020].

[7] ”How to control a stepper motor with A4988 driver and Arduino,” *Makerguides*. [Online]. Available:https://www.makerguides.com/a4988-stepper-motor-driver-arduino-tutorial/ . [Accessed 5-May-2021].

[8] ”Helpful GPIO Pinout resources that you can reference while connecting sensors to your Raspberry Pi,” *Techcoil*. [Online]. Available: https://www.techcoil.com/blog/helpful-gpio-pinout-resources-that-you-can-reference-while-connecting-sensors-to-your-raspberry-pi-3/. [Accessed 5-May-2021].

[9] ”GPIO,” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.org/documentation/usage/gpio/. [Accessed: 5-May-2021].

[10] “Installing PyQt5,” *PyQt5 Reference Guide.* [Online]. Available: https://doc.bccnsoft.com/docs/PyQt5/installation.html [Accessed 4-March-2021].

[11] “ CNC Router 3018-PRO,” *Genmitsu.* [Online]. Available: https://s3.amazonaws.com/s3.image.smart/download/101-60-280PRO/Genmitsu_3018_PRO_User_Manual_V1.1%20_%20202009.pdf [Accessed 20-April-2021].

Appendix:

Github:
https://github.com/ajbetancourt/E52-Senior-Design

Testing/Prototype Videos:
Probe moving downward in the z-axis:
https://drive.google.com/file/d/1262nzeXioDAADg_n8nc5mPg0vBBMI1qK/view?usp=sharing
Probe moving upward in the z-axis:
https://drive.google.com/file/d/12ogyOK78ERDfPj_I3HjgeLZtojnyOS83/view?usp=sharing
Run Button/motor movement:
https://drive.google.com/file/d/1ZZUZ1Oiug-SBjjQAEknosuckf32Mv1A4/view?usp=sharing
Calibration of motors through GUI:
https://drive.google.com/file/d/1yAanfbXpCQv_TkZ-IwDd2Dwng9wcWPQq/view?usp=sharing
GUI Only
https://drive.google.com/file/d/1QE5ZUTHfDGornhhvaHZEOmmtiuoIpU7L/view?usp=sharing
Spectrum Analyzer Script test:
https://drive.google.com/file/d/1UW-RNFvmJsHomSZnBPtC4bdRAs_4Igug/view?usp=sharing