Resonant Cavity Field Profiler

Team 31

Salaj Ganesh, Max Goin, Furkan Yazici

TA: Stasiu Chyczewski

Spring 2022

Abstract

This document highlights the key design features and components of our Resonant Cavity Field Profiler. This project was developed by the request of Starfire Industries for aiding in profilers their particle accelerators. All design requirements put forth by Starfire have been completed.

Table of Contents

1. Introduction

1.1 Background

Radio frequency quadrupole (RFQ) linear accelerators offer a compact solution for obtaining low velocity, 0.01-0.06 c, particle beams. RFQs have a distinct advantage as they offer very good transverse beam focusing, can accept a DC particle beam, and produce fine bunches. Good bunching and focusing is a requirement for input into many higher energy linear accelerators so RFQs are often used as a first stage in a series of accelerators. These properties also make them well suited for accelerating heavy ions. RFQs have four veins whose tips are machined such that the tip varies sinusoidally along the central axes of the accelerator with an increasing wavelength. This modulation of the tip geometry is the principal component that allows for a longitudinal electric-field component to accelerate ions. The veins are driven by a magnetic field coupled with RF excitation with altering phasing for each vein [3].

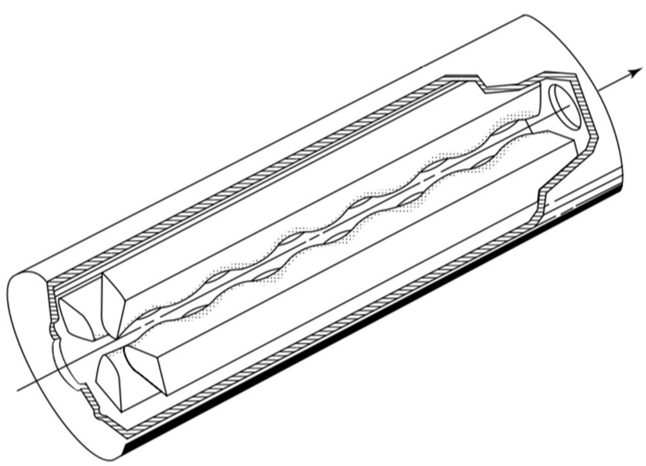

Figure 1: Cross-section of an RFQ accelerator. The modulation of the 4 veins tips can be seen clearly but the increasing wavelength along the particle path (arrow direction) is not represented.

1.2 Problem

This project proposal was submitted by Starfire for designing a device to tune Resonant Cavity Particle Accelerators.

The design process of Resonant Cavity Particle Accelerators requires fine characterization of their electric field for design verification and tuning. This is typically accomplished by displacing some of the internal volume of the cavity with a metal object such as a rod or a bead. This results in a small, but measurable change in the resonant frequency of the internal cavity as the resonant frequency is partially dependent on the cavity's volume. This frequency offset gives the operator an indication of the strength of the magnetic field displaced by the bead. This is then used to estimate the electric field strength and uniformity. This measurement also allows an engineer to determine the cavity's resonant mode. This volume displacement measuring method is typically done manually with a user making small changes to the position of the bead or rod and measuring the resulting frequency shift. This process can be very time-consuming and take a single user up to two days to accomplish.

1.3 Solution

A stepper motor will move the bead through the cavity, while a microcontroller will measure any resonant frequency offset and log the current position. This device will move the bead through all 4 cavities of the accelerator while simultaneously making measurements to estimate the current field conditions in response to the bead. The frequency offset will be controlled by the MCU, and the setting time of the control loop will be the limiting factor in the time it takes to perform a complete characterization. This will help technicians properly tune and characterize cavities to obtain optimum performance.

We are working with Tom Houlahan, the engineer responsible for the RF board design and our primary customer for our design. Starfire has provided a test cavity for the purposes of design verification and testing. Tom would like the characterization to take roughly the amount of time to get a cup of coffee or roughly five minutes at minimum.

1.4 Setup

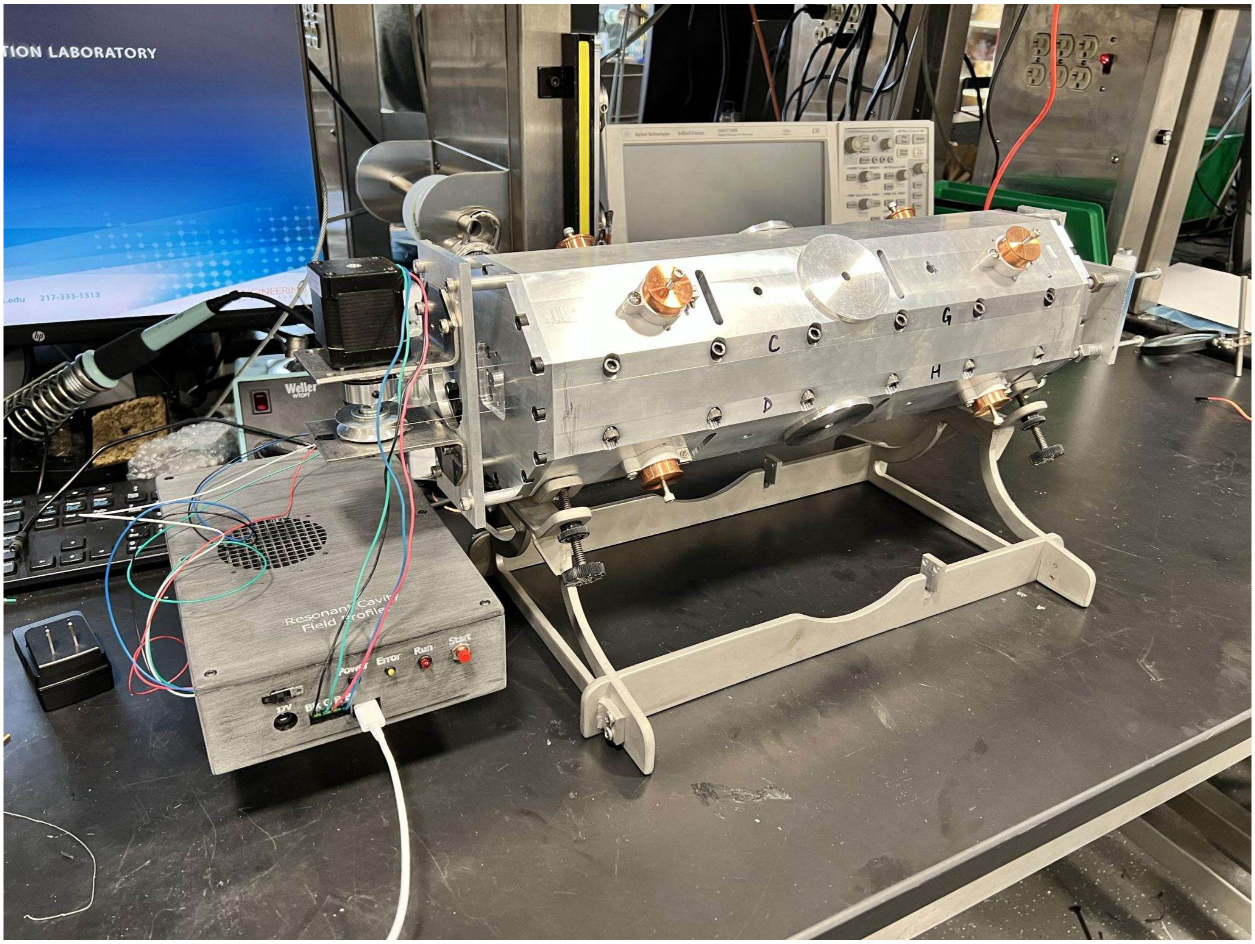

Figure 2: Basic setup with the test cavity. Our board is within the box to the left. The RF antennas cannot be seen but are placed into port holes along the back of the cavity.

1.5 High-level requirements list

High level requirements were decided upon after discussions with our client, Starfire Industries. Careful considerations were made to ensure usefulness of the product and ensure it would meet some quality standards. To achieve this, the cavity profiler must perform the following:

1. Pull a bead through the cavity and record the responding resonating frequency. It will continue and calculate the bead position for an optimal resonating frequency. These measurements should occur at every 1 mm interval of bead movement.

2. Characterization will complete in a matter of minutes (five minutes was requested). The entire characterization process will not require any input from a user. Once complete, it would notify the user of its completion using notification LEDs.

3. Data will be logged on a PC for later use so users can refer to it later. The data should include the position of the bead and its corresponding field characterization.

2. Design

2.1 Block Diagram

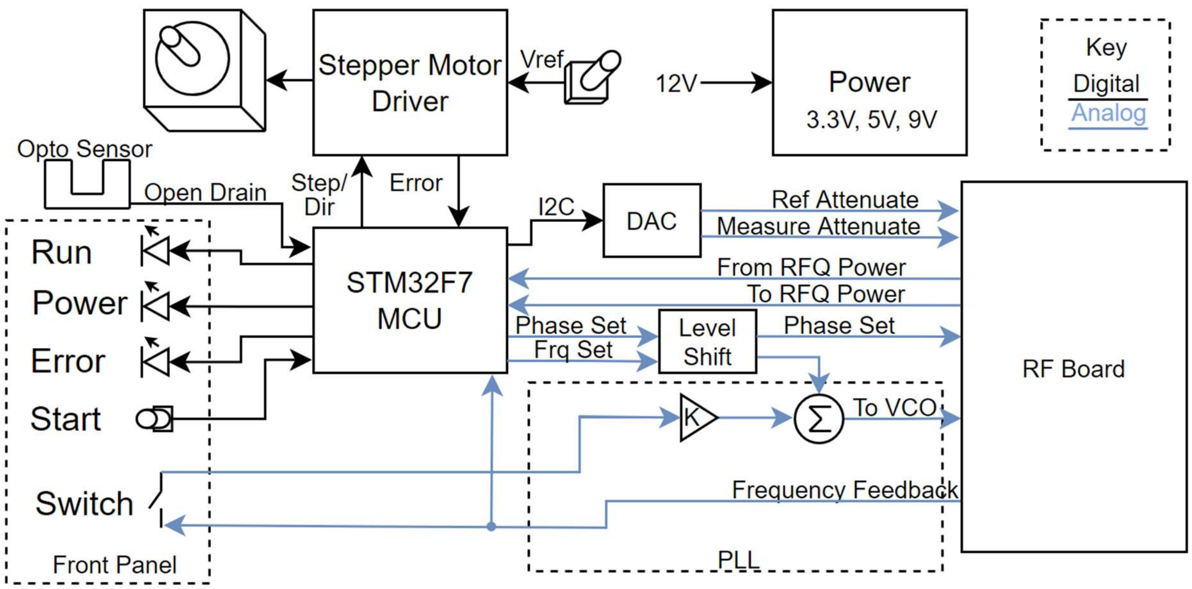

Figure 3: Detailed Block Diagram

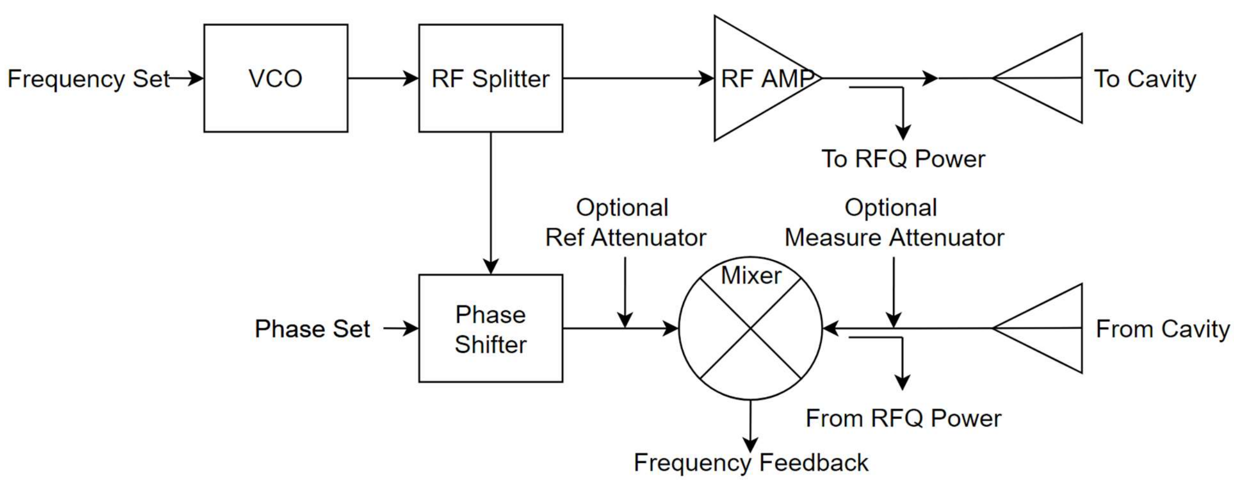

Figure 4: RF Board Block Diagram

2.2 Design Overview

Our design consists of a microcontroller, stepper motor driver, power supplies, and an analog circuit. The microcontroller records the analog signals using its onboard ADCs and supplies the signals for phase and frequency adjustments on the RF board using its DACs. It also controls the stepper motor driver and I2C DAC as well as displays its status to the user using a few LEDs. The stepper motor and optical sensor are used to drive a metal bead on a nylon string through the RF cavity while having the capability of returning the metal bead to a set location. This bead will be used to displace the EM field in the cavity and cause a resonant frequency perturbation. The power supplies generate the required voltages for all the subsystems. In total four voltage rails are generated. These include 9 V for the phase set level shifter, 5 V for the stepper motor, a separate 5 V rail for the analog circuitry, and 3.3 V for the microcontroller and LEDs. The analog circuits supply the RF board with control signals in the appropriate voltage ranges and maintains resonance of the chamber using a PLL.

The RF board generates RF at around 600 MHz using a voltage-controlled oscillator (VCO). Its output is split down two paths. One path is sent to a power amplifier which is used to drive the RF cavity. The second path is sent into a set of voltage-controlled phase shifters which are used to supply the mixer with a phase shifted version of the RF launched into the cavity. There are also power meter taps to measure the RF power after the power amplifier, after RF returns from the cavity, and at the inputs to the mixer. The mixer supplies us with a DC signal that represents the phase difference between the RF and the returned RF. This signal can then be used to determine the frequency shift of the resonant cavity using a little math.

2.3 Subsystem Overview

2.3.1 Microcontroller

For this project we chose a STM32 microcontroller. This microcontroller (MCU) will supply drive signals to a stepper motor to move the metal bead through the four quadrants of the RF cavity. It will also control a couple of LEDs on the front panel to indicate the current state of the system. One of the main reasons we chose this microcontroller over others is its ability to easily communicate with external devices through its COM port. Communication to an external computer will allow the user to set operating conditions and to log position and frequency correction data for further analysis. It can also sample feedback signals from the RF front end and issue corrections to the PLL circuitry through the DACs. This microcontroller is highly capable and notably has two 12-bit DACs, three 12-bit ADCs, four UARTs, 320 kB onboard SRAM, and USB 2.0 FS support. These will fully satisfy our requirements for a microcontroller. The main schematic for the MCU can be seen in Figure 21 in Appendix C. the accompanying schematic for the USB interface can be found in Figure 22 in Appendix C as well.

2.3.2 Analog Circuitry

The analog circuitry consists of a PLL feedback circuit and two analog level-shifters. The PLL feedback circuit is responsible for maintaining a drive frequency that is equal to the resonant frequency. A buffer and a summing amplifier are used to form a control loop from output signals from the RF tuning circuit provided by Starfire.

The tuning circuitry outputs a signal that is proportional to the cosine of the phase difference between the reference and measurement arms. A metal bead in the cavity causes a frequency shift on the order of 30 kHz, which yields a phase shift of approximately 10 mRad [12]. The phase and resonant frequency vary linearly with one another over these small ranges, so we're looking at ~0.33 mRad/kHz. This translates to an output of ~(750 mV)$*$cos(90º−10 mRad) = 7.5mV. Since these perturbations are very small, this is in the vicinity of the zero crossing. Therefore, we assume linearity when calculating the feedback response of the tuning circuit.

$$\text{DC feedback gain } (\beta) = (7.5\text{ mV}) * (30\text{ kHz}) = 0.25\text{ mV/kHz}$$

The tuning sensitivity of the VCO is ~12 MHz/V around our resonance operating range which is ~600 MHz. With the VCO's tuning sensitivity (α) of ~12 MHz/V, as well as RF-circuit response (0.25 mV/kHz), we calculate the gain required for stability [13].

Below is the block diagram of the feedback loop. fr and Δf_r are, respectively, the unperturbed resonant frequency of the cavity and the perturbation caused by the metal bead. V0 is set at the start of the process to set f0 = fr, which causes the output of the cavity tuning circuit to collapse down to simply a difference between Δf and Δf^r. This linearization and mapping from phase to frequency works because we are looking at very small perturbations and operating in the vicinity of the zero crossing. The feedback circuit is responsible for amplifying that output signal, summing it with V0, and passing that summed signal to the VCO. With these assumptions, we determine requirements for amplifier gain on the RF-circuit's output that lead to a locked frequency (that is: Δf0 = Δfr, when resonance changes, the drive frequency changes to approximately match it).

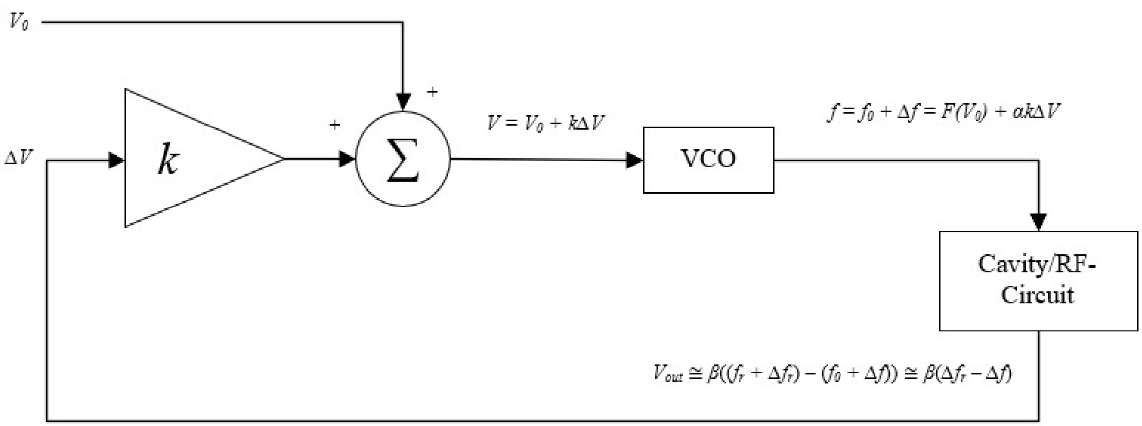

Figure 5: VCO Control Loop Block Diagram

Using the diagram above, we write: $\Delta f = \alpha * k * \Delta V = \alpha * k * \beta * (\Delta f_r - \Delta f)$. Here, k is our feedback gain, we calculate the conditions under which Δf approaches Δfr. We shoot for an error of 0.1% as it is desired by Starfire.

$$\text{Need } \frac{\Delta fr - \Delta f}{\Delta fr} = 0.1\%$$

$$\frac{\Delta f}{\Delta fr} = 1.001$$

Using this ratio, we calculate k,

$$k = \frac{\Delta f}{\alpha * \beta * (\Delta fr - \Delta f)} = \frac{\Delta f}{\alpha * \beta * (\Delta fr - \Delta f)} = \frac{\Delta f / \Delta fr}{\alpha * \beta * (1 - \Delta f / \Delta fr)} = 333.7$$

The designed amplifier circuit to achieve this gain is included in VCO control loop circuit in Figure 24 in Appendix C.

The two analog level-shifter circuits are responsible for mapping the two 0-3.3 V DAC outputs (FrequencySet and PhaseShift signals) of the MCU to 0-5 V and 0-9 V outputs respectively. This is accomplished by using a non-inverting op-amp configuration as follows.

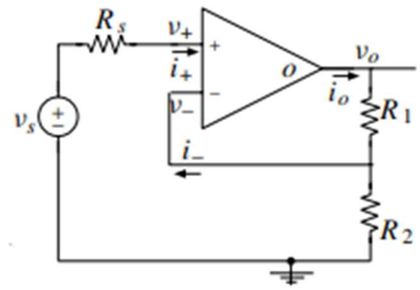

Figure 6: Non-Inverting Op-Amp Configuration

The resistor values were picked to maintain a maximum output current $I_{o,max}$= 1 mA of our LMV116MF amplifiers. We determined the values of R_1 and R_2, using the Eq. (1.1) and Eq. (1.2).

$$R_1 + R_2 = \frac{V_{o,max}}{I_{o,max}} \quad (1.1)$$

$$G = 1 + \frac{R_1}{R_2} \qquad (1.2)$$

2.3.3 Stepper Driver

We will be using a Trinamic TMC2100 as the stepper motor driver IC. The TMC2100 can supply up to 1.2 A RMS and up to 256 micro steps offering high positional performance and smooth operation. This IC can be easily interfaced with a couple of GPIO pins on the MCU. Over the GPIO pins, the microcontroller can issue speed and distance commands and read back the current positional data. A small optical sensor will be used to “home” the bead to ensure the same starting location before each profiling pass. The schematic for the stepper driver can be found in Figure 25 in Appendix C. Note that the TMC2100 uses a couple of hardware configuration pins to enable and disable certain features. By leaving pin CFG1 open and tying CFG2 to high we set the stepper motor to four micro steps with smooth motion. All other configuration pins were set to the optimal recommendation by the datasheet [7].

2.4 Software Overview

To tie all the subsystems together, some software design was needed. C code was written for the microcontroller and the computer side of the software was written in Python. Various features of our project are defined as functions within the code on the microcontroller.

One of the main functions of our project is the ability to move a bead with a stepper motor while measuring the feedback and sending the data back to the computer. An overview of this system can be seen below in Figure 7.

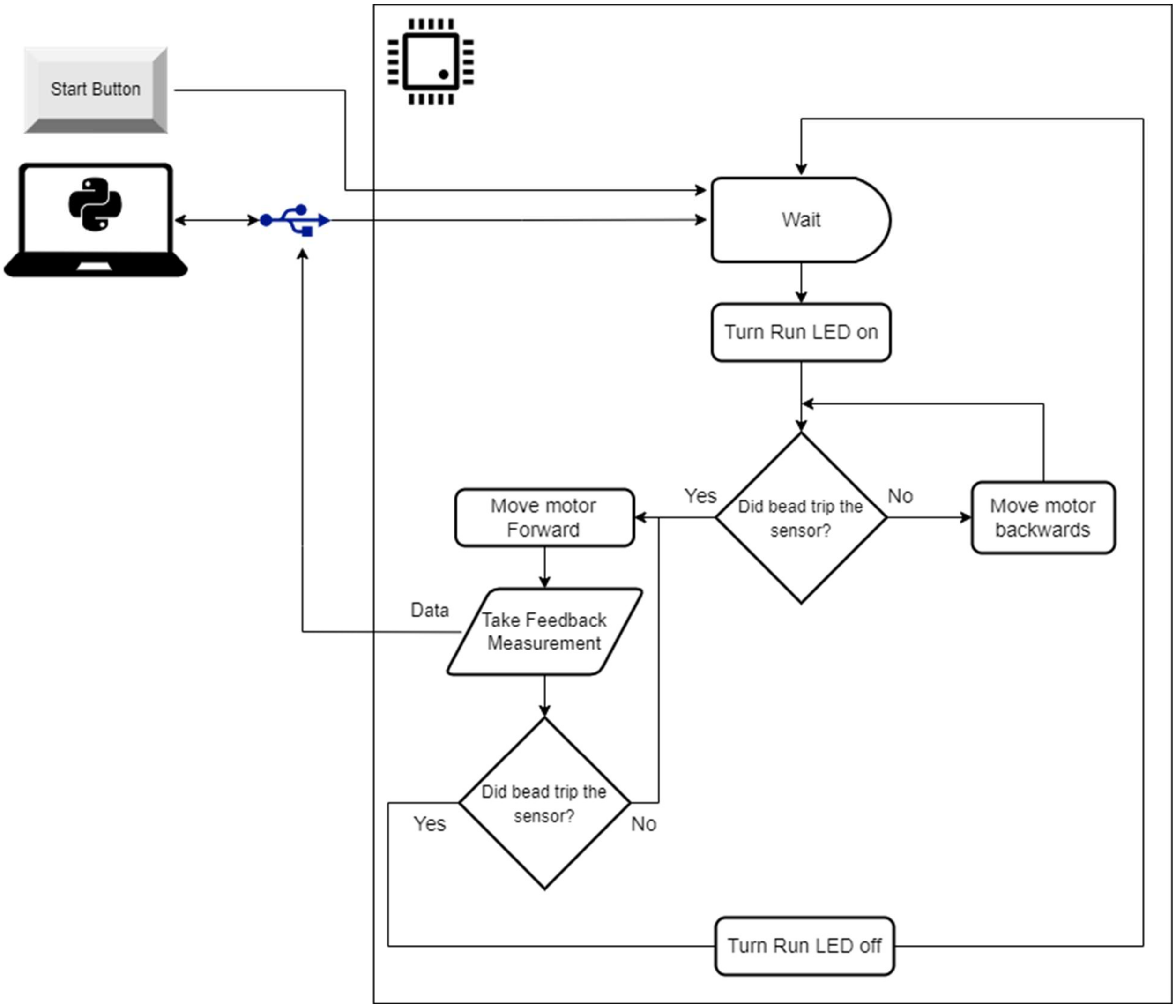

Figure 7: Software flowchart of running procedure

First the microcontroller waits for any input from the computer and checks if the button has been pressed. If either of those conditions occur, it will then proceed with the procedure. Whenever a procedure is run, an LED on the front panel of the box is lit, letting the user know a run is in progress. The first step of the procedure is to reset the bead to its original position. This involves moving the stepper motor in the opposite direction until an optical sensor mounted on the side of the cavity is tripped. This optical sensor consists of a small laser and acts like a switch whenever the laser is obstructed. The MCU can read the state of the optical sensor and waits until the bead is back at the start. Once the bead has been reset, the microcontroller makes a small step forward. It then takes a reading of the feedback through one of its ADCs and sends the data to the host computer through USB. It then repeats this process of moving the bead one step and taking measurements. The microcontroller will know when the bead has done a full cycle when it retriggers the optical sensor. The microcontroller then turns off the run LED and waits for future input.

In addition to running the main procedure, our design boasts a couple of other software procedures as well. Calibration is done by sweeping the VCO until a maximum return power is found. It also needs to tune the phase shift until a correct zero edge is obtained. This function is also defined in the code and callable through the USB interface. The code also includes USB callable functions such as setting certain parameters such as step size and speed. The user can also call functions to reset the bead anytime they need to and manually set the DAC values.

3. Verification

3.1 Microcontroller R/V

Our microcontroller is a central part of our design and handles many different tasks. It is responsible for data acquisition from the frequency circuitry, issuing stepper motor control signals and communicating with a host computer. Luckily, while waiting for our intended microcontroller to ship, we were able to test many of these core functions on a similar STM32 microcontroller. This test MCU was not identical to the intended one, but satisfied many of our needs, and provided an identical software library. On this MCU we were able to test out ADC/DAC conversion code and the USB interface.

Creating a skeleton of the code on the test MCU gave us a solid foundation when it became time to start testing code on our intended microcontroller. After directly porting the necessary code over from the test project, we were able to test and verify our microcontroller was able to satisfy all our requirements set forth by Table 1 in Appendix A. Using a Python script running on a host computer, we were able to receive and transmit data to the microcontroller. The MCU was also capable of reading signals from the frequency circuitry through its ADCs and DACs. Figure 8 below shows an excerpt of the csv file of the data we were able to receive form the MCU.

From RFQArm Measure	LPArm Measurenbsp	Frequency Feedback	Ref Arm Power	To RFQArm Measure
132	1896	551	24	1779
112	1896	553	29	1775
174	1896	552	18	1780
167	1895	553	44	1775
116	1894	552	13	1779
149	1894	552	48	1781
108	1894	553	30	1779
119	1894	552	50	1778
123	1894	552	22	1782
161	1893	552	43	1775
124	1893	552	20	1781
136	1892	552	12	1777
177	1892	552	16	1775
188	1892	553	13	1779
144	1892	552	19	1778
172	1892	552	53	1773
172	1892	550	19	1773
129	1891	549	10	1779
183	1891	552	24	1778
170	1888	553	47	1783

Figure 8: Excerpt from CSV file of Feedback Data

The I2C DACs were verified successfully by writing I2C messages as fast as possible to sweep the outputs of both channels in opposite directions as seen in Figure 9. The sweep passes though the full output range of the DAC and took roughly 500 ms to complete. This means the update period was every $\frac{500}{2^8}$ ms which is roughly 512 Hz. This is reasonable for an I2C controlled device, and we could double the rate if only writing to one channel. We can tell 12C was the limiting factor in update rate as the messages are so tightly packed in Figure 10 that any space between I2C start and stop conditions is not easily identifiable.

Figure 9: Dual channel DAC sweep

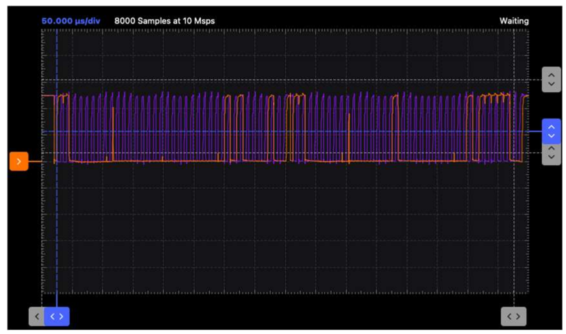

Figure 10: DAC I2C messages, SCL in purple

3.2 Frequency Circuitry R/V

We needed to test three circuit topology, one PLL feedback loop circuit and two level shifters. Since the feedback loop is composed of two different amplifier subcircuits, we tested each separately. Buffer amplifier and summing amplifier were tested by supplying different DC voltage levels and comparing the output voltage to the expected gain results. We were able to get the correct gain and summation values.

To test the level shifters, we swept the DAC outputs for their full range (0-3.3 V) and observed the output of the shifters. We were able to observe the full range sweep for both the 0-5 V and 0-9 V level shifters.

Since the main point of this circuit is to drive the RF/Cavity tuning circuitry provided by Starfire, we had to make sure that the DC feedback readings were meaningful. The results were verified by the responsible engineer at Starfire during our demo run.

3.3 Stepper Driver R/V

The rotor driving the nylon wire attached to the metal bead is 28mm in diameter. We have the stepper motor driver configured to four micro steps per step and our motor is a 200-step stepper motor. This means a full rotation of the motor is 800 steps and each step is $\frac{28mm \times \pi}{800\ steps} = 0.11\ \frac{mm}{step}$. For this reason, we take nine steps between each field measurement to result in 1mm spacing of such measurements.

3.4 Power Circuitry R/V

A high-quality power circuitry is needed for the optimal operation of many of our other subsystems. It is important for this circuit to provide the precise voltage that is needed without risking damaging these components. Since we have sensitive analog components in our design, it was important to validate the power circuitry to a higher level of precision. For our design we imposed a couple of requirements to ensure this optimal operation (Table 4 in Appendix A)

Our final design (refer to Figure 23 in Appendix C) included a 24 W AC to DC power supply supplying 12V to our PCB through a standard DC barrel plug. Then a series of linear regulators step down from 12V to the various voltages our components need. To verify this power circuitry satisfied our needs, we loaded our board onto the test cavity and ran some test procedures. With the maximum load possible we probed the circuit with an oscilloscope to monitor the output voltages. As we can see from Figures 11-14 below, our power supply behaved within specifications, and we did not see any deterioration in performance. During this heavy load we also tested all other voltages and they behaved normally as well.

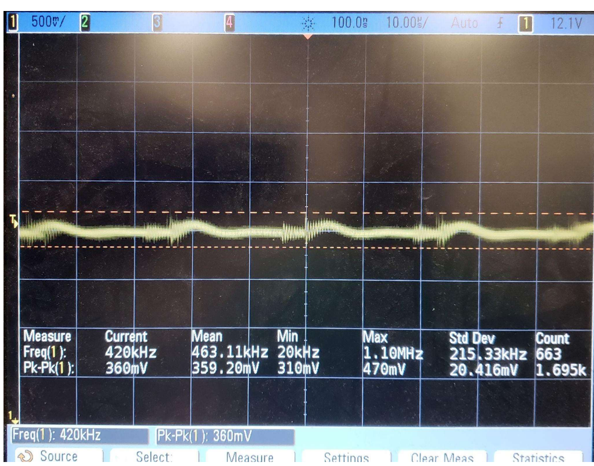

Figure 11: 12V Output

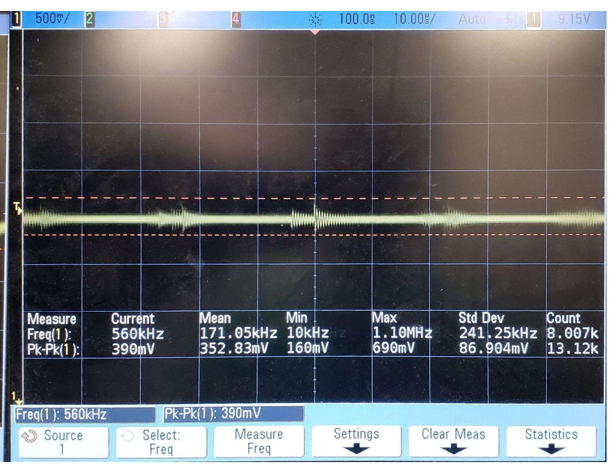

Figure 12: 9V Output

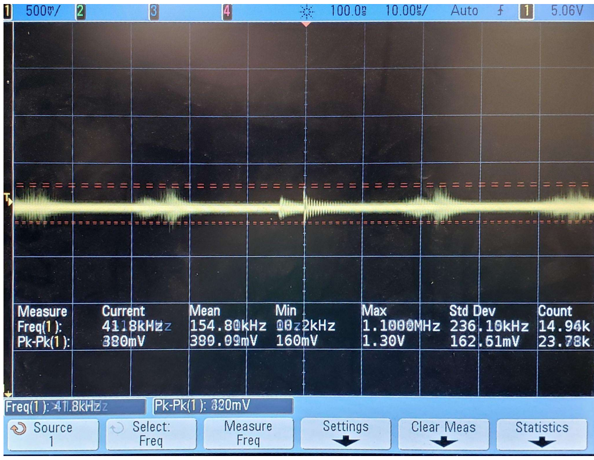

Figure 13: 5V Output

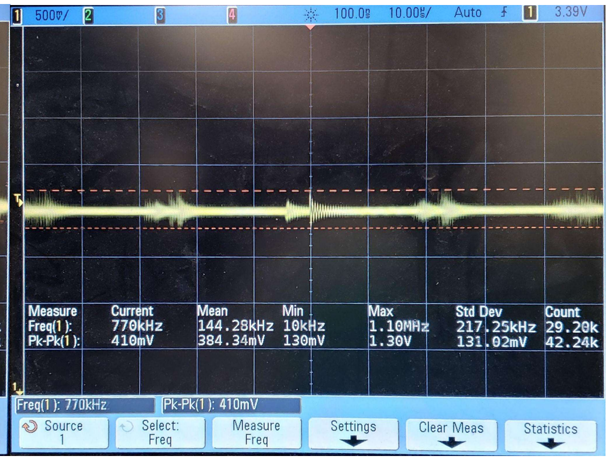

Figure 14: 3.3V Output

Also, as seen in Figure 15 below, the analog portion of the power circuitry behaved as we intended as well.

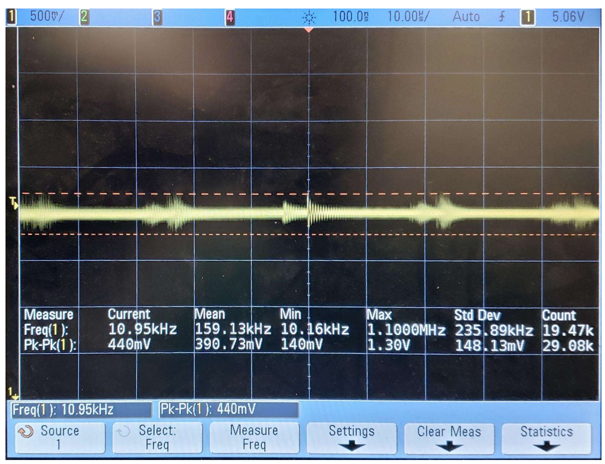

Figure 15: 5V Analog Output

4. Cost and Schedule

4.1 Cost of Labor

The average salary for ECE graduates at the University of Illinois at Urbana Champaign is around $100,000 per year or around $50 an hour. Assuming each member of our team works for 10 hours per week, for the remaining 12 weeks of the semester, we can estimate the total labor cost this project will require.

$$\frac{\$50}{hour} \times \frac{10\ hours}{week} \times 12\ weeks\ \times 3\ people\ = \$18{,}000$$

4.2 Cost of Parts

Refer to Table 5 in Appendix B for a breakdown of the individual components.

4.3 Total Cost

In the best-case scenario, the total price of the project comes out to $18,168.85. Thankfully, we did not need to replace any of the components due to malfunctions.

5. Conclusion

5.1 Accomplishments

Thankfully, after hours of debugging and a few minor modifications, we were able to accomplish all our main design goals. Paired with a test cavity, our circuit was able to read various frequency signals and send data to an external computer all while moving a small bead through. Currently all our testing was performed on a test cavity provided by Starfire. In the future we would like to see the project running on a real cavity. We presented our design to our engineering contact at Starfire, and he was very pleased with our work. In Figure 16 below we can see an example of the data our project would collect.

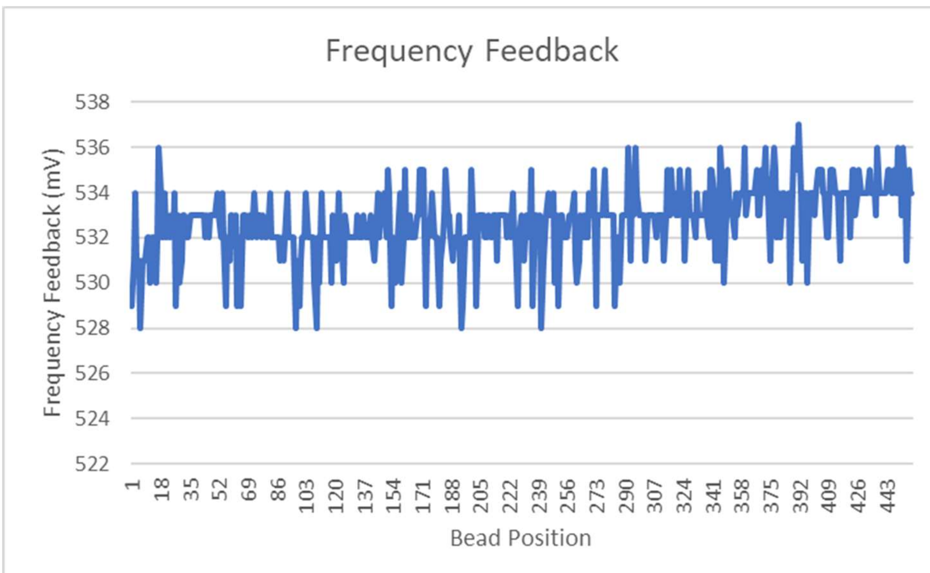

Figure 16: The feedback from moving the bead through the cavity

5.2 Design Changes

The design required a few changes to become functional. The largest architectural change occurred in the analog circuit. We realized after finalizing the board that we needed to be able to disconnect the feedback signal from the PLL circuit for the initial calibration of the resonant frequency of the cavity. We don't want the PLL to be attempting to alter the frequency we request from the VCO. This change required the removal of R18 (seen in Figure 17 with a red X) and replacing it with a switch in series with the original SMD part. We also needed to be able to sample the PLL feedback signal, so we added a jumper wire off one of R14's solder pads to one of the ADC input pins on the MCU (seen in Figure 18 with a blue circle).

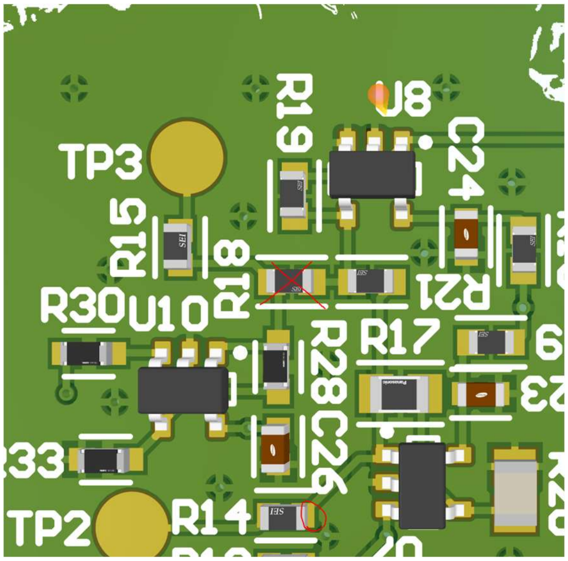

Figure 17: Analog circuit modifications

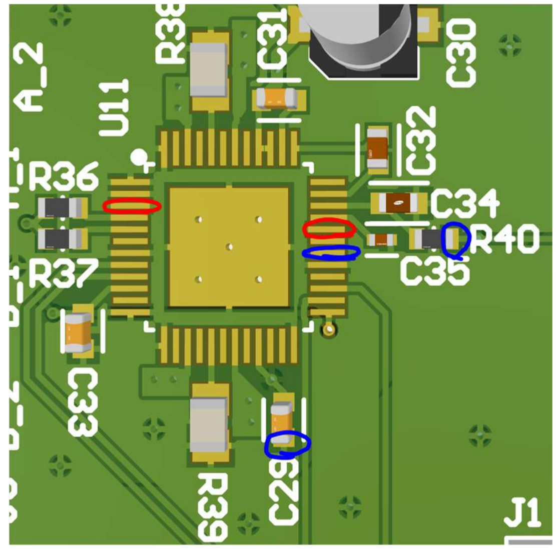

Figure 18: TMC2100 circuit changes

The next major changes were with the stepper motor driver. Pins 3 and 29 needed to be disconnected from ground and left floating (seen circled in red in Figure 18). These pins are specifically used for configuring the IC. Pin 29 is for enabling the motor outputs. Leaving this pin floating enables the motor driver permanently and configures a lower holding current while the motor has not moved for more than three seconds. We found that this lower idle current setting greatly reduced motor noise. Pin 3 configures how the voltage reference pin, pin 30, is used. Setting pin 3 floating configures the IC to use the voltage on pin 30 to scale and set the output current based on the voltage across the current shunt resistors. This allows us to tune the output phase currents as needed. We initially calculated the correct shunt resistance using the TMC2100 data sheet to achieve 1 A phase current on the motor as this is what the motor's data sheet called for. That design would constantly assume an overcurrent was occurring so was not useful. After installing a 200K Ohm voltage divider across 5V. This was done with a 100K Ohm resistor attached to R40's 5 V pad (seen circled in blue on Figure 18) to a 100K Ohm potentiometer whose center pin is soldered to pin 30 (seen circled in blue on Figure 18) of the TMC2100 and other pin soldered to a ground pad. C29 has a convenient ground pad that worked well for this modification (seen circled in blue on Figure 18). The potentiometer will supply pin 30 with 0 V to 2.5 V and allows for easy tuning for less audible motor noise.

The final modification was to fix a minor design error where the pinout for the DC jack was incorrect. This required cutting the small thermal relief traces around the top 2 pads (seen circled in red in Figure 19) and soldering a jumper wire from the rightmost pad to the lower pad of C14 (seen circled in blue in Figure 19).

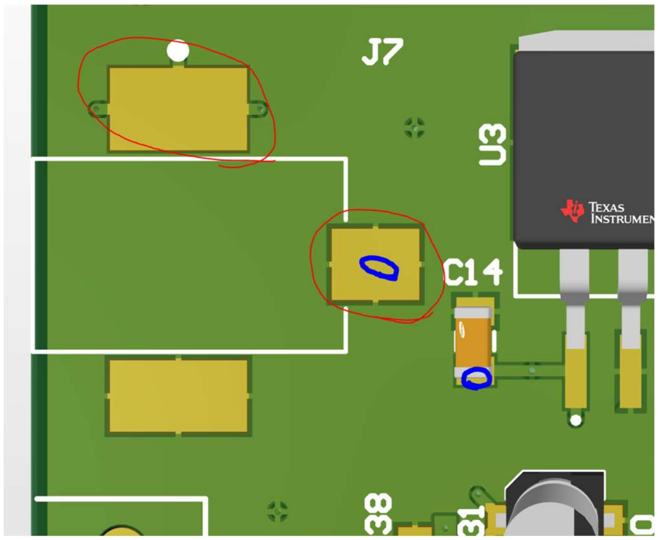

Figure 19: DC Jack pinout correction

5.3 Future Work

Despite our completed design satisfied all the needs specified by Starfire, there were a few areas of the design we would like to improve upon. Currently, our design relies on a barebones Python script running on a host computer to receive commands and send data. We could improve this by creating a GUI program to interface with the device. Having a GUI would make the system more user friendly and easier to operate. This would allow less technically inclined personnel to operate the device and not require a manual to reference device commands. Adding another USB port for storage devices would also allow the device to store any data without the need for an external computer.

Another aspect of the design we would like to improve upon is the stepper motor operation. Currently the stepper motor and driver are always active, creating a high-pitched whining noise. During calibration procedures, stepper motor operation is not needed so it and the driver can safely be turned off. This feature is supported by our chosen stepper driver IC through GPIO control over pin 3 of the TMC2100. A revision of this board could easily activate this feature.

Finally, due to an error in our design, a mechanical switch had to be added to the front of the box to reconnect the feedback loop after calibration. In a future revision of the board, this switch could instead be implemented as an analog switch controlled by the MCU's GPIO. Having the MCU control the connection of the feedback loop will ensure the switch would be in the correct position for calibration and procedure and limit any user error. In the current design we warn the user to move the switch to the correct position before calibration, but if that is not done calibration is not valid. This would be a major improvement for users of this device.

5.4 Ethics and Safety

This project will be working with potentially high-power RF circuits operating at around 600Mhz. Leakage from the cavity could represent a serious source of interference to communications in the UHF band. This is mitigated by the design of the RF cavity and is outside the scope of our work. This still should be considered, and the test cavity should not be operated with the tuning port holes open. A fully operational particle accelerator of this class is capable of accelerating particles to energies capable of damaging tissues. This will not be an issue as the cavity will not be under vacuum and therefore not capable of accelerating particles.

6. References

[1] IEEE Code of Ethics." IEEE, https://www.ieee.org/about/corporate/governance/p7-8.html.

[2] Powers, Tom. Theory and Practice of Cavity RF Test Systems. United States: N. p., 2006. Web.

[3] T. P. Wangler, "1 Introduction," in *RF linear accelerators*, 2nd ed., Weinheim: Wiley-VCH Verlag, 2017.

[4] *Narrow Band Phase Shifter JSPHS-661+ - Mini-Circuits*. https://www.minicircuits.com/pdfs/JSPHS-661+.pdf.

[5] STMicroelectornics, "Arm® Cortex®-M7 32b MCU+FPU, 462DMIPS, up to 512KB Flash 256+16+4KB RAM, USB OTG HS/FS, 18 TIMs, 3 ADCs, 21 com IF," STM32F722xx, STM32F723xx Datasheet, February 2022 Revision 8

[6] STMicroelectornics, "STM32F72xxx and STM32F73xxx advanced Arm®-based 32-bit MCUs," RM0431 Reference manual, June 2018 Revision 3

[7] Trinamic Motion Control, "Standalone intelligent Step/Direction driver for two-phase bipolar stepper motor," TMC2100 Datsheet, October 2014 [revised June 2020]

[8] IEEE, "How to Cite References: IEEE Documentation Style," Available: https://ieee-dataport.org

[9] Microchip Technology, "8-/10-/12-Bit Single/Dual Voltage Output Nonvolatile Digital-to-Analog Converters with I²C™ Interface," MCP47FEBXX Datasheet, 2015

[10] T. Powers, "Theory and Practice of Cavity RF Test Systems," Thomas Jefferson National Accelerator Facility

[11] Thomas P. Wangler, *RF Linear Accelerators. 2nd Edition*, Wiley-VCH Verlag GmbH & Co. 2008

[12] STMicroelectornics, "Getting started with STM32F7 Series MCU hardware development," AN4661 Application note, February 2017 Revision 5

[13] Mini-Circuits, "Voltage Controlled Oscillator ROS-615R+," EDR-10412/2 datasheet.

Appendix A

Requirements and Verification Tables

Table 1: Microcontroller Requirements and Verification

Requirements	Verification	Status
Communicate with a PC over USB for data logging	Connect MCU to multiple different computers and operating systems and ensure it appears for data interactions.	Verified
Utilize the ADC on microcontroller to convert analog signals from Frequency-Lock Circuitry	Connect a potentiometer to the MCU ADC and have it output in the same range as the Frequency-Lock Circuitry	Verified

Table 2: Analog Front End Requirements and Verification

Requirements	Verification	Status
Amplify an analog feedback signal by a multiplier of 333x	Connect a function generator and have it output a signal in the mV range and ensure the signal gets amplified by correct multiplier	Verified
Sum the amplified feedback signal with a frequency set signal	Input two analog signals and ensure the output is as desired	Verified

Table 3: Stepper Driver Requirements and Verification

Requirements	Verification	Status
Precisely control a Stepper motor (provided by Starfire) to move a bead through the RF cavity with 1mm movements.	Connect a test bead to the stepper motor and move it slowly while measuring the displacement of every tick. Each tick should displace the bead around 1±0.2 mm.	Verified
Communicate with MCU on the current position of the bead and reset it to a given position if needed.	Write a test program on the MCU to randomly move a test bead to a certain location and reset it multiple times. Should always reset to the same position	Verified

Table 4: Power Circuitry Requirements and Verification

Requirements	Verification	Status
Output the required 15V, 9V, 5V, and 3.3V (± 5%) needed for subsystem operation	Supply AC and measure outputted voltages using a voltmeter	Verified
Properly filter the analog output to prevent power supply noise interfering with sensitive measurements	Ensure the analog power lines remain constant (± 1%) with a large load on other power lines (stepper motor).	Verified

Appendix B

Parts and Cost Table

Table 5: Bill of Materials

Part Number	Description	Supplier	Quantity	Net Price
STM32F722RET6	32-Bit Microcontroller	STM	1	$13.13
MCP47FEB0202	Dual Channel DAC	Microchip	1	$1.34
629105136821	Micro USB 2.0 Receptacle	Wurth	1	$1.27
LMV116MF/NOPBCT-ND	Output Amplifiers	TI	4	$7.00
TMC2100-TA	Bipolar Motor Driver	Trinamic	1	$5.98
TL3305AF260QG	Tactile Switch	E-Switch	1	$0.21
MPM-10-15	10W AC Power Supply	Mean Well	1	$13.44
L7809ABD2T-TR	9V 1.5A Voltage Regulator	STM	1	$1.15
LDL1117S33R	3.3V 1.2A Voltage Regulator	STM	1	$0.50
LM2931ADT50R	5V 100mA Voltage Regulator	STM	1	$1.40
TL780-05CKTTR	5V 1.5A Voltage Regulator	TI	1	$1.74
MCP47FEB02A0-E/ST	IC DAC 8-bit	Microchip	1	$1.74
OPB903W55Z	Optical Sensor	TT Electronics	1	$6.39
PM5RD/GD/YD	RGB LEDs	Bivar Inc.	3	$2.32
N/A	Miscellaneous Resistors, Capacitors, Inductors	N/A	N/A	$38.50
N/A	4 layer PCB	PCBway	1	$30.00
N/A	Solder Stencil	PCBway	1	$10.00
N/A	PLA Filament, Project Box	N/A	807 g	$24.23
N/A	Fan and Buttons	Amazon	1	$18.51
Total Price				$168.85

Appendix C

Board Schematics

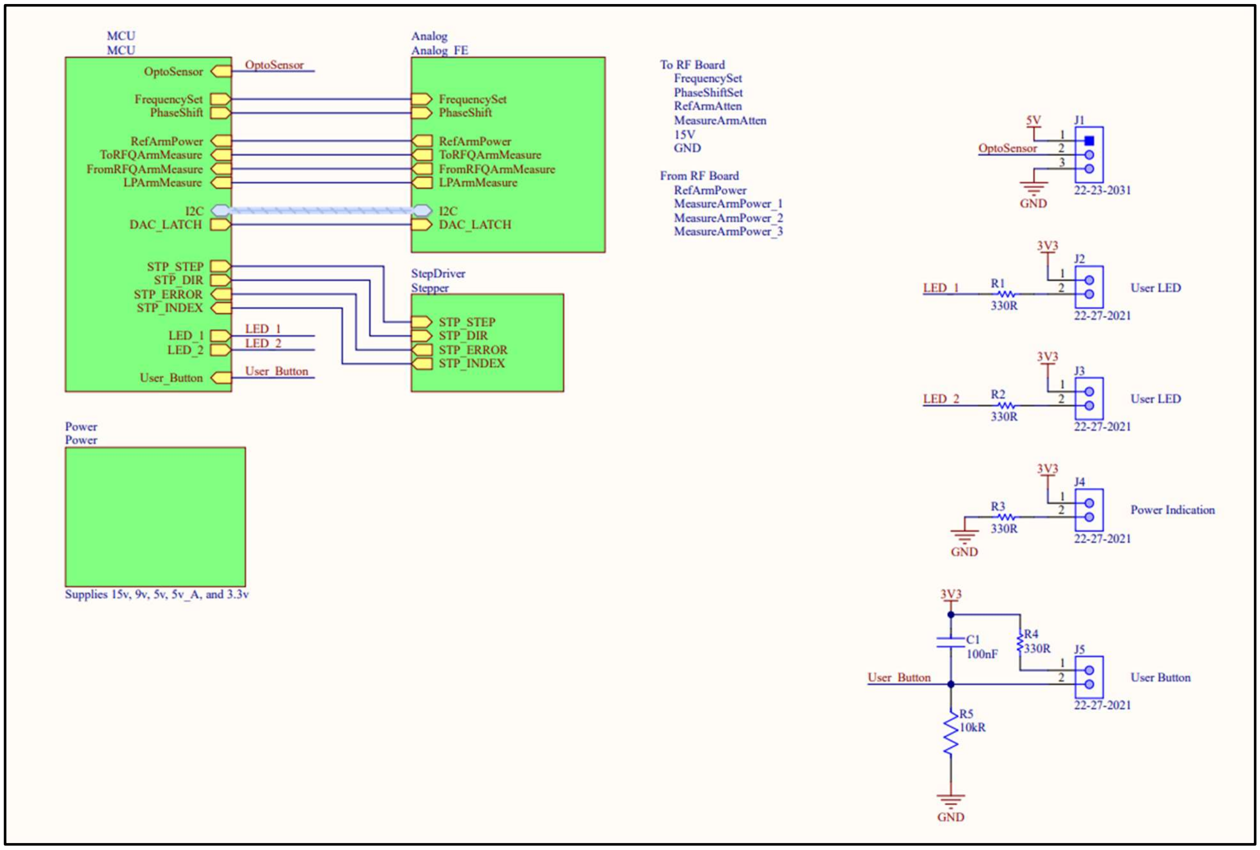

Figure 20: Top Level Schematic

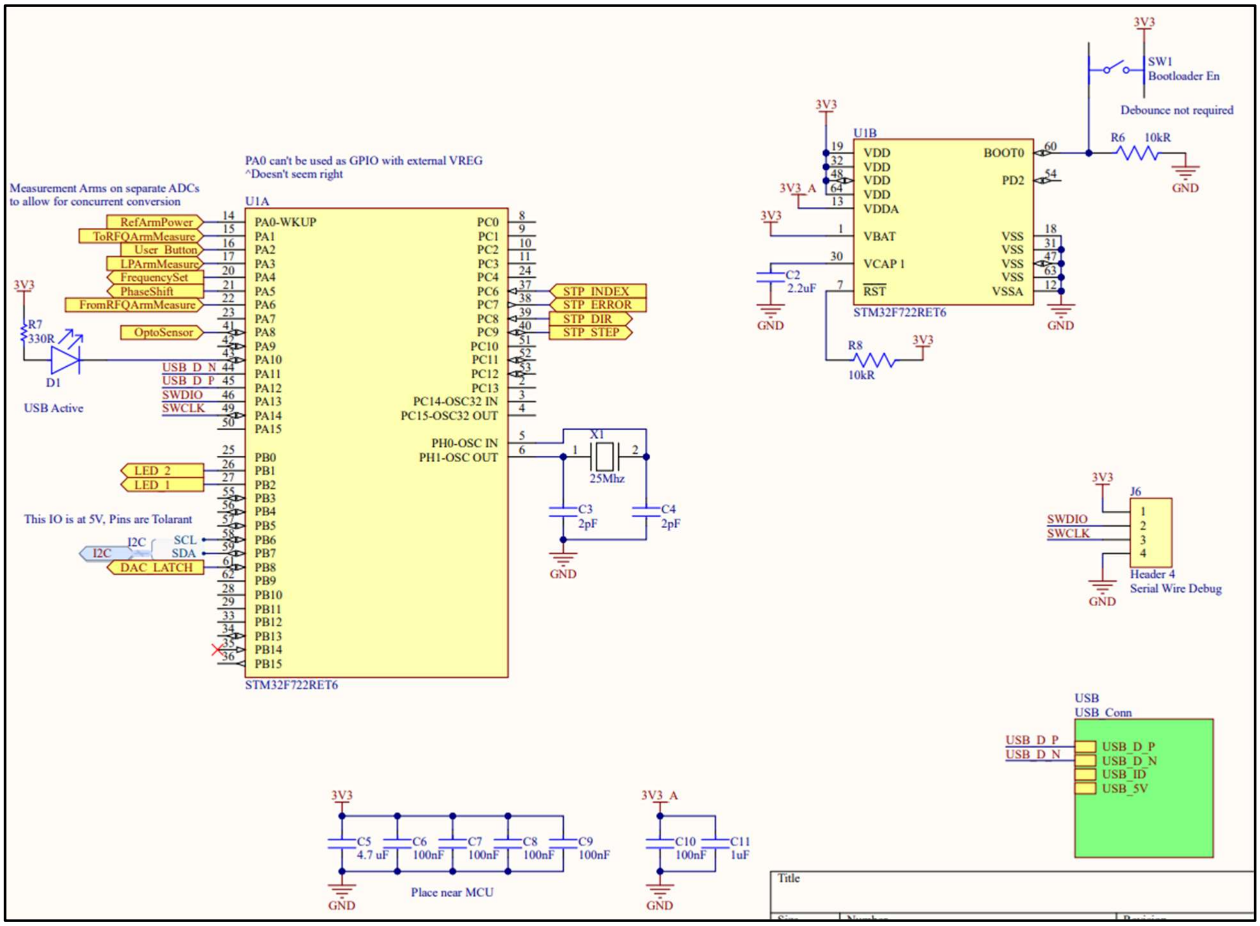

Figure 21: Microcontroller Schematic

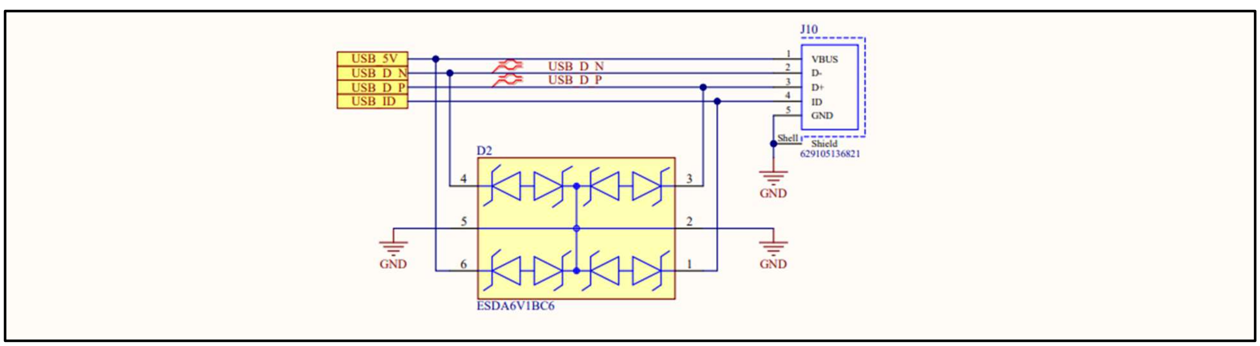

Figure 22: USB port Schematic

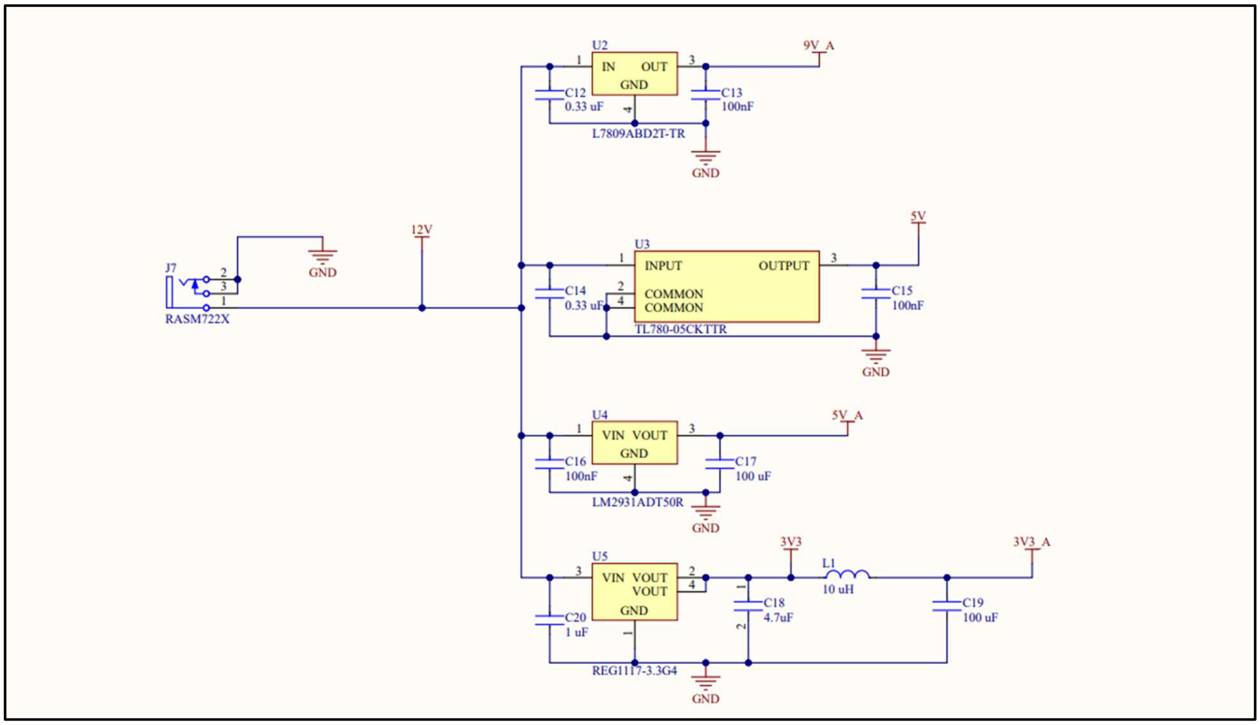

Figure 23: Power Circuitry Schematic

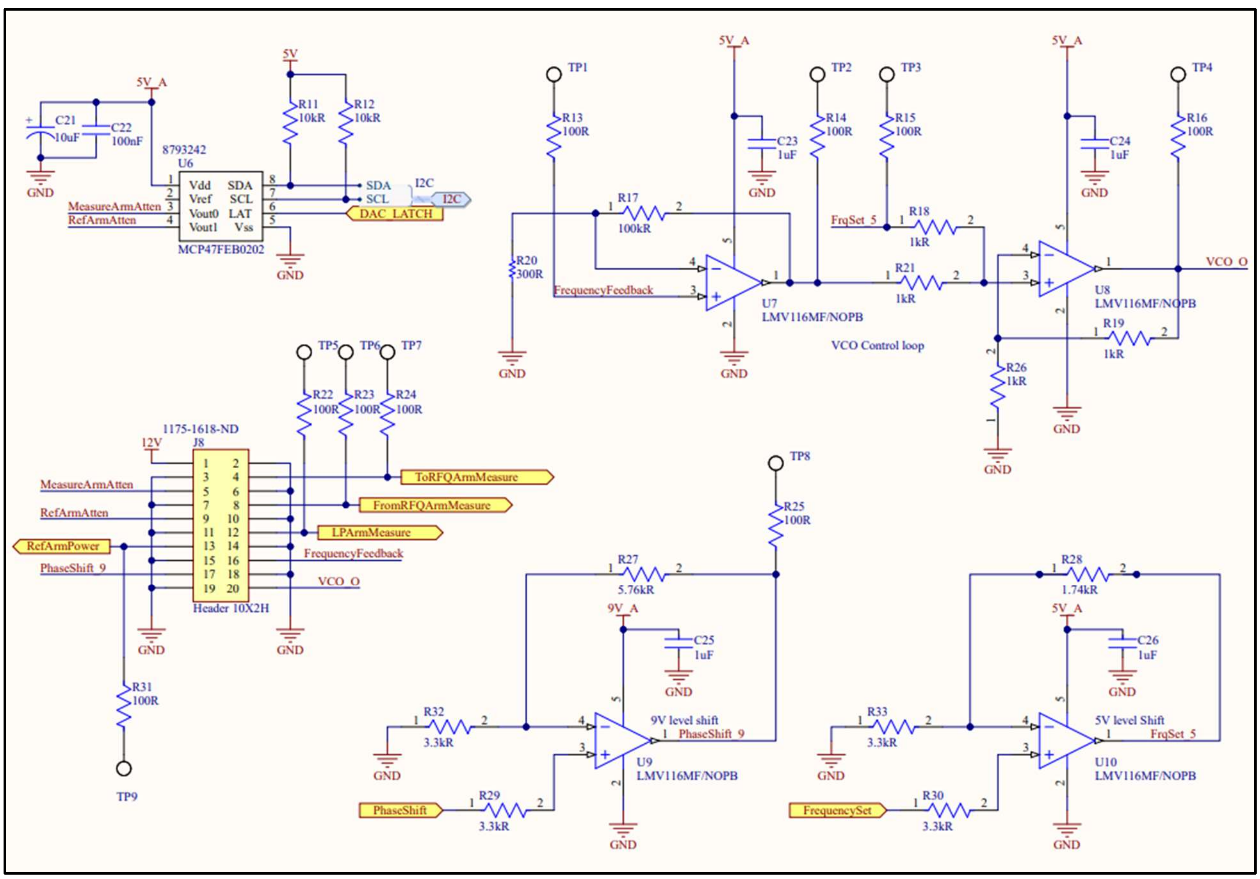

Figure 24: Analog Circuitry

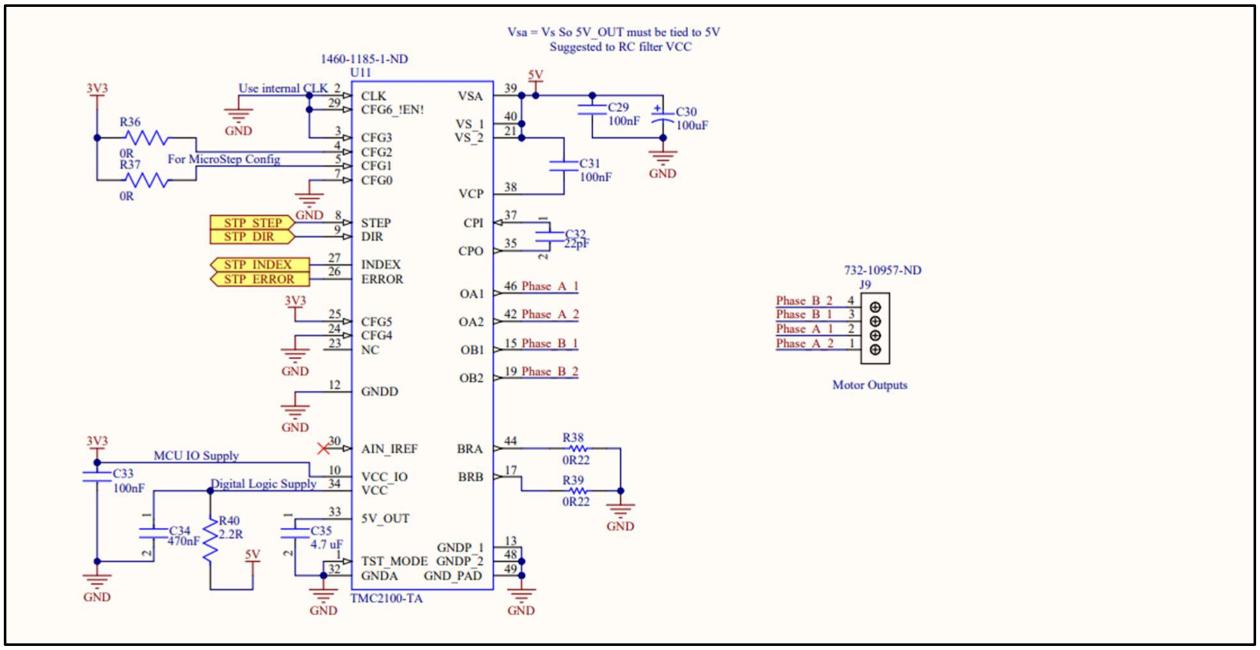

Figure 25: Stepper Motor Schematic