MICROCONTROLLER COMPENSATED MICROMACHINED OSCILLATOR

Megan Driggers, Heather Hofstee, and
Michaela Pain

University of Central Florida, Department. of
Electrical and Computer Engineering, Orlando,
Florida, 32816, USA

***Abstract* — Oscillators are the heartbeat of electronic devices. Microelectromechanical systems (MEMS) oscillators are usually smaller in size than the industry standard crystals and can be fabricated using conventional semiconductor methods. However, their performance tends to vary with temperature. This paper describes a method of digital resistance (and thus temperature) control for a thin-film piezoelectric-on-substrate (TPoS) oscillator to explore the behavior of the device with microcontroller -based temperature compensation.**

***Index Terms* — MEMS, microcontroller, oven-control, resonators, temperature coefficient of resistance, TPoS.**

I. INTRODUCTION

Microelectromechanical systems (MEMS) resonators have been shown to have applications as a smaller, more easily fabricated, and sometimes less expensive oscillator compared to the current crystal oscillators that dominate the market. The challenge with MEMS oscillators currently centers around their performance, especially since they have a relatively high temperature coefficient of frequency (TCF). The TCF details the changes in resonance frequency with respect to temperature and is minimized at the resonance frequency.

Because resonator performance is affected by temperature, having a method of stabilizing the temperature to the value corresponding to the resonance frequency should help optimize and stabilize the resonator performance, minimizing changes in the frequency by keeping the resonator operating at a temperature where the TCF is stable. However, because the temperature coefficient of resistance varies from resonator to resonator, this project has been designed to keep the resistance of the resonator constant since it is directly related to temperature.

To keep the resonator resistance stable, current will be passed through it to elevate its temperature using Ohmic resistive heating. The resistance will be controlled with a microcontroller and set via user input. The microcontroller will be used to program a control loop to keep the resonator temperature stable. The output of the resistance of the resonator will be displayed to an LCD screen so that the user can know the values.

A. MEMS Resonator

A resonator is a device that naturally oscillates at its resonant frequencies with a greater amplitude than at other frequencies. These oscillations can be generated either mechanically or electromagnetically. This allows the resonator to generate and detect specific frequencies. Mechanical resonance is when a system absorbs more energy when the frequency of oscillation matches its resonance frequency. Microelectromechanical systems (MEMS) resonators are microstructures that resonate at high frequencies and can be excited electrically.

Their performance can be analyzed using mechanical and electrical equations, with the quality factor serving as a metric of the device performance.

Another performance metric, used to determine the thermal stability of a resonator, is called the temperature coefficient of resonance frequency (TCF). The resonance frequency changes with temperature and shifts in the modulus of elasticity, structural damping, and thermal expansion or contraction of different materials. Thus, thermal stability is important to consider because temperature change affects the resonance frequency of the system. The TCF is found by placing a test sample within a cavity on a low-loss, low-dielectric constant, and low-thermal expansion material. The cavity is then placed within a temperature chamber, and the resonant frequency is measured at each temperature over the desired range of temperatures. The TCF can then be calculated and expressed in parts-per-million-per-degree Celsius (ppm/°C) [1].

B. MEMS Oscillators

Oscillators are devices used to produce periodic electric currents or voltages through exchanges in kinetic and potential energy. The electronic signals generated by these devices are often a product of the circuit design and the values of the components; however, they generally assume sine or square waveforms. Oscillators can convert direct current (DC) from a given power supply to an alternating current (AC) signal. They are often distinguished by their output signal frequency and output signal type, and this lends to different applications. For example, oscillators can generate clock signals exercised in computers and broadcast signals utilized in transmitters.

Currently, quartz crystal oscillators are the most common type of oscillator. The output signal frequency of a quartz oscillator is affected by the temperature of the quartz crystal, which can impact its resonance frequency. An oven-controlled crystal oscillator (OCXO) is a specific type of oscillator that controls the temperature of an oscillator circuit using an oven. This type of oscillator is often used to provide improved temperature stability and frequency accuracy with respect to a standard crystal oscillator. The drawback to using an oven-controlled crystal oscillator is that it often consumes a lot of power and space, which can be expensive.

Thin-film piezoelectric-on-substrate (TPoS) resonators have been the subject of the research of Dr. Reza Abdolvand, an associate professor at UCF, for over a decade. These resonators are a type of lateral bulk acoustic resonators (which vibrate via expansion and contraction due to electric signal converted into a force) specifically shown to have high Q-factors in the MHz range [2]. TPoS resonators involve creating piezoelectric components (which will translate electrical energy into mechanical energy) as a part of the silicon bulk [3].

They have been researched as a potential replacement for crystal oscillators due to their smaller size, ability to be fabricated using conventional semiconductor methods, and potential to be more energy efficient. However, a major issue with these devices currently is their lack of temperature stability. Thus, the printed circuit board design of this project was intended to be used with a TPoS MEMS oscillator to explore the resistance (and thus temperature) compensation of a TPoS MEMS oscillator because of its unique TCF properties.

II. SYSTEM DESIGN

To keep the resistance of the resonator constant, a controlled current will be passed through the resonator to elevate its temperature. This current will be controlled by a microcontroller, which will receive data about both the voltage across and the current through the resonator. Based on that information, it will adjust the current accordingly. The temperature of the resonator relates to its resistance so the user will be modifying or stabilizing the temperature of the resonator by doing the same to its resistance value. This allows more flexibility in testing different types of resonators because each type has its own correlation between resistance and temperature.

The program will operate in three modes to either display the state of the resonator in terms of resistance at the default current or a selected current or regulate the resistance of the resonator. For the latter, a control system will be used to stabilize the resistance. The user will be prompted to set the desired resistance using switches, and the LCD display will serve as a communication interface between the user and the microcontroller to receive input from the user and display the results.

The actual heating of the resonator will be done using an analog circuit setup according to the schematic in Figure 2. V_{in} is the voltage from the microcontroller that will be converted to a current. This current will then pass through the resonator and a precision resistor, marked here as R_{mea}. This resistor will be largely insensitive to changes in resistance as its temperature increases; thus, the voltage across it can be measured using an instrumentation amplifier and then divided by the resistance value to get the current passing through the resonator. The divider circuit on the instrumentation amplifiers are in place to match the voltage input range of the microcontroller. The voltage reference source provides for stable biasing of the resistors and the BJT.

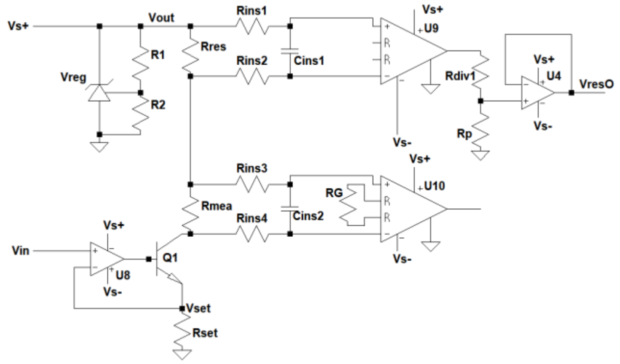

Fig. 2. Analog portion of printed circuit board design.

III. HARDWARE

The resonator, microcontroller, and LCD were each chosen to best suit this project.

A. Resonator

The resonator to be used in this project is a TPoS resonator. Research has shown them to offer high-Q (quality factor) and high-power handling. They have a thin layer of piezoelectric material on a substrate (often silicon) [3]. The largest drawback of these types of resonators is the relatively high temperature coefficient of frequency (TCF). Lightly doped silicon's TCF is usually about -30ppm/°C while MEMS resonators on silicon usually have a TCF of about -50ppm/°C.

The turnover temperature is where the TCF changes polarity and is the point where the TCF is minimized. This turnover temperature is dependent on the doping concentration and resonant mode [4]. The desire of this project is to stabilize the resonator temperature so that it operates near the turnover temperature and thus operates with a minimized TCF.

B. Microcontroller

The microcontroller chosen to be used in this project is the Texas Instruments MSP430. This family of microcontrollers has peripheral sets designed for a range of different applications. These microcontrollers are simple 16-bit devices that encompass the desired communication interfaces in order to display the information to the user. The architecture allows for extended battery life for portable measurement applications such as this one.

The advantages for the MSP430FG479 include its low cost and simplicity. This microcontroller is perfect for low-power applications. This project requires the conversion of signals from analog to digital and digital to analog, and this microcontroller provides both of those functionalities with a high resolution. The MSP430FG479 offers a 16-bit Analog-to-Digital converter which will be used to read in analog voltage signals from the resonator and a resistor. The microcontroller also offers a 12-bit Digital-to-Analog converter which will be used to output a voltage to modify the read-in voltage across the resonator and resistor. The amount of flash and RAM for this microcontroller is sufficient for measuring and displaying values. It has a 32MHz internal oscillator and employs a 16-bit RISC CPU, 16-bit registers and constant generators that allow for efficient code implementation. [5]

C. Liquid Crystal Display (LCD)

The LCD chosen to be used in this project is the TC1602A-09T. The purpose of the LCD is to display the resistance readings and prompt user input. This LCD is a 16x2 character-based screen compatible with the selected microcontroller. It contains its own controller with the HD44780 interface. This will allow textual information to be presented to the user. The resistance will be constantly updated on the first line and the user prompts will be displayed on the second line. The screen can be interfaced via a 4-bit or 8-bit parallel data bus [6]. Furthermore, it is available in COB (Chip on Board) packages and is RoHS compliant. For this application, parallel display connections will be used to interface the LCD to the microcontroller, and a preexisting C library will be used to define the connections and display characters to the LCD.

IV. SOFTWARE DESIGN

There are several software components that will run sequentially to achieve the desired results for this project. The purpose of the software can be divided into the following tasks: calculating the resistance of the resonator, displaying the desired information to the user and prompting settings input, and controlling the current passed through the resonator if necessary. These tasks were accomplished through functions programmed into the microcontroller. The software environment and programming language that have been selected to compile and implement the code are the Code Composer Studio Integrated Development Environment (IDE) and C programming language, respectively.

The program implemented a PI controller in order to achieve the desired resistance. The program initialization begins by outputting a default voltage to the entire circuit. These values are output as a digital signal, so they are subsequently converted to analog values using the Digital-to-Analog converter on the microcontroller. Then, the voltage across a 0-TCR resistor in series with the resonator is measured. This value is input as an analog signal, so then it is converted to digital values by the Analog-to-Digital converter. This enables the program to calculate the current through the resonator. Then, the voltage across the resonator is measured and the resistance of the resonator will be calculated using the resonator voltage and current values. The resistance is continuously processed due to the voltage and current being oscillated until the desired resistance is achieved. Thus, the resistance is updated on a periodic basis whenever an updated current is passed through the resonator. The resistance value is output to the LCD screen. Then, the user is prompted to select a mode to operate in.

A. Software Functionality

The microcontroller program operates in three modes: standby, characterization and operational. This was a feature requested by the research group under Dr. Abdolvand. The standby mode limits the program loop to read relevant resonator values and outputs the calculated resistance to the LCD screen. The characterization mode allows the user to select a desired current value for the circuit to operate in, provided it passes the checks. Then, the program continuously reads relevant resonator values and outputs the calculated resistance to the display. The operational mode allows the user to input the desired resistance value for the program to stabilize the resonator at and performs the necessary calculations to determine the current for the next iteration of the loop. This allows the user to control whether the device is operating to stabilize the resistance or to simply display the state of the resonator to the user. In addition, it conserves power without sacrificing the time to power down and startup the device again. This contributes to the low-power quality of the application when the program is in standby mode.

The microcontroller continues to display the state of the resonator to the user in standby and characterization modes while the user is prompted to enter a desired resistance value in the operational mode. For the operational mode, the program calculates a new current to be passed through the circuit using a control system. The described program logic is shown in Figure 3.

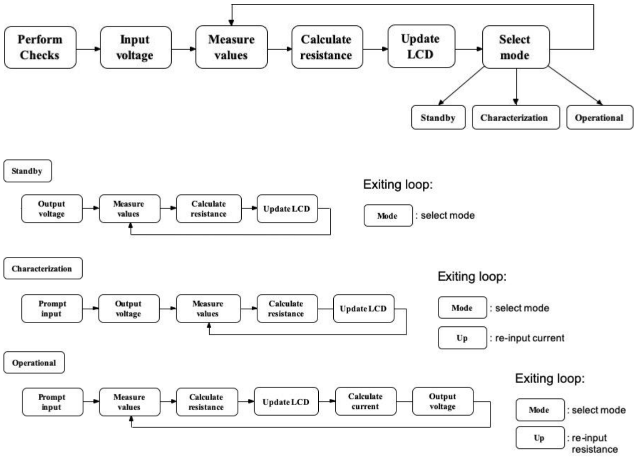

Fig. 3. High-level software flow chart.

In the figure above, it is evident that the first priority when the microcontroller is switched on is to perform checks to determine the state of the device before continuation of the program execution. Then, the microcontroller reads the relevant voltages from the desired pins and calculates the current and resistance. The current state of the resonator is significant as it used to determine whether the current through the resonator needs to be adjusted. The user is then able to select a mode to operate in while the microcontroller continues to calculate and update the resistance.

For the standby mode, a default voltage value is output to the circuit. Then, a loop is used to continuously calculate the resistance value and update the LCD. For the characterization mode, the user selects the desired current value, and this is converted to a voltage to be output to the circuit. Then, a loop is used to continuously update the resistance to the LCD. For the operational mode, a control loop will be implemented. A proportional-integral (PI) controller is used to continuously control the current through the circuit. The methodology is as such: the system output is assessed, and the system is reconfigured based on the desired output response. The implementation follows the steps of processing information passed from the measurement device to the controller which then evaluates the input and calculates a correcting output that is passed back into the process.

The control loop will receive the current and desired resistance as parameters and perform calculations to determine the updated current value. This value will be converted to a voltage which will be output to the circuit to update the resistance. The process will continue until the user chooses to exit the control loop. The control system used for this application is described in the following section.

B. Control System

This section discusses the design of the PI Controller which will be used to stabilize the resistance and thus the current passing through the resonator. The user will input a desired resistance value and the PI controller will adjust the system's output response to minimize the error. The controller has a proportional and an integral term which are both functions of the error. The error in this case is given by the desired resistance value minus the actual resistance value (for a positive TCR curve). An example of a system's response containing a PI controller is shown in Figure 4 below.

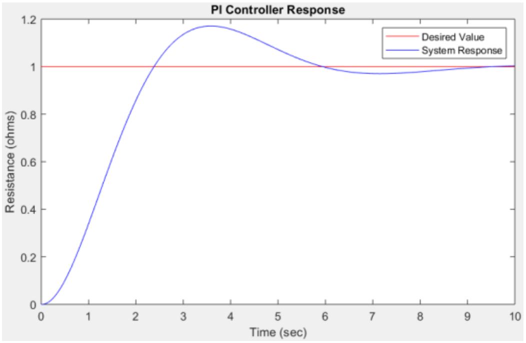

Fig. 4. Example PI Controller System Response

Notice how there can be a significant overshoot from the desired value. There is a tradeoff with PI controllers between the maximum overshoot and the rise time. This means that in order to minimize the maximum overshoot, then there will be a longer rise time for the system to rise to the desired value. In order to approximate values for the proportional gain (K_p) and the integral gain (K_i) for the PI controller, the control system's transfer function must be found. The block diagram given in Figure 5 is used to find the control system's transfer function

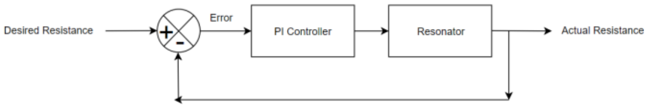

Fig. 5. Control system block diagram.

The PI controller's transfer function is given by $PI(s) = K_p + \frac{K_i}{s}$. The resonator's transfer function is unknown and different for each device. This project is intended to work for different types of resonators so approximations will need to be made to account for this. The resonator's transfer function is approximated by a first order transfer function to be $R(s) = \frac{b}{s+a}$ where ‘b’ and ‘a’ are constants that vary for each resonator. Now, the control system’s transfer function can be found using **Error! Reference source not found.**.

$$T(s) = \frac{Actual\ Resistance}{Desired\ Resistance} = \frac{PI(s) * R(s)}{1 + [PI(s) * R(s)]}$$

The transfer functions for PI(s) and R(s) are then substituted to give the transfer function

$$T(s) = \frac{b(K_p s + K_i)}{s^2 + (a + bK_p)s + bK_i}$$

The denominator of the transfer function is the system’s characteristic equation.

A general second-order system transfer function is given by:

$$C(s) = K * \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2}$$

Where K is the system’s desired steady state value, ζ is the damping ratio, and ω_n is the undamped natural frequency. For an undamped system $\zeta = 0$, for an underdamped system $0 < \zeta < 1$, for a critically damped system $\zeta = 1$, and for an overdamped system $\zeta > 1$. For the project application, a underdamped response is desired because it allows for the fastest rise time. The denominator of the general transfer function is the desired characteristic polynomial of the control system, so the system’s characteristic equation is set equal to this then solved for the proportional gain (K_p) and the integral gain (K_i).

$$s^2 + (a + bK_p)s + bK_i = s^2 + 2\zeta\omega_n s + {\omega_n}^2$$

By comparing coefficients for the ‘s’ term, K_p is found.

$$a + bK_p = 2\zeta\omega_n \ \rightarrow \ K_p = \frac{2\zeta\omega_n - a}{b}$$

By comparing the constant terms, K_i is found.

$$bK_i = {\omega_n}^2 \ \rightarrow \ K_i = \frac{{\omega_n}^2}{b}$$

Keep in mind, the resonator’s transfer function was approximated by a first order transfer function to be $R(s) = \frac{b}{s+a}$ where ‘b’ and ‘a’ are constants that vary for each resonator. The ‘b’ term depends on the resonator’s highest resistance value when applied with a unit pulse whereas ‘a’ depends on the output’s rate of exponential decay when applied with a unit pulse. Therefore, the proportional gain (K_p) and the integral gain (K_i) are both inversely related to the resonator’s resistance. To test these theories, constants are applied to K_p and K_i where K_p=0.02 and $K_i = 0.002$. The maximum overshoot is analyzed when the different resistance values are used in place of the resonator. If K_p and K_i and inversely related to resistance, it is expected that the maximum overshoot will increase as the resistance increases which means that the gains need to decrease as resistance is increased (instead of being constants). The data is graphed and shown in Figure 6.

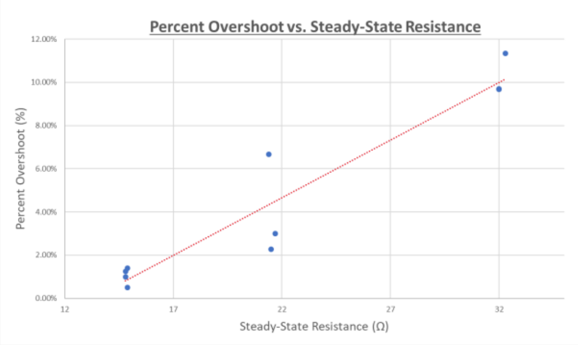

Fig. 6. Control measurements using constant gain values.

Notice how the maximum overshoot increases as the resistance increases which means that the gains need to decrease as resistance is increased. The PI controller gains are set to be inversely dependent on the resistance of the resonator

V. SCHEMATIC AND BOARD DESIGN

The PCB schematic and board design consist of several sections that tie all aspects of the project together. Important components of the schematic and board design include DC-to-DC conversions, the programming interface, schematic design, and board design.

A. DC-to-DC Conversions

Several DC-to-DC conversions are needed to provide all components with the necessary operating voltage and current values. There are four different voltage rails which will be needed to power all the components and they are 10V, 5V, 3.3V, and -10V as shown in Table 1 below. The main power supply source will be the Agilent E3631A triple output DC power supply. One terminal will be set to +10V and a second terminal will be set to -10V. The 10V rail will then be stepped down directly to 3.3V and stepped down directly to 5V using voltage regulators.

Table 1. Component supply voltages.

Component	**Supply Voltage(s)**	
Instrumentation Amplifiers	+10V	-10V
Operational Amplifier	+10V	-10V
LCD Display	5V	
Microcontroller	3.3V	

When selecting the best voltage regulators for the project application, low noise is the most important feature. Switching and Zener voltage regulators contribute a lot of noise into the circuit, therefore linear regulators were chosen.

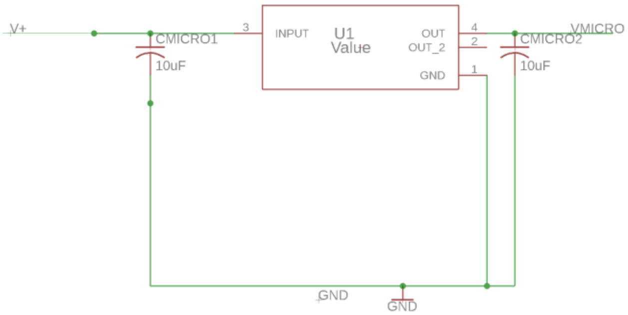

Fig. 7. 10V to 3.3V conversion with linear regulator with filter capacitors.

The LM1117- 3P3NDP regulator was chosen for the 10V to 3.3V conversion because it is a linear regulator with low noise and is large enough to be soldered by hand.

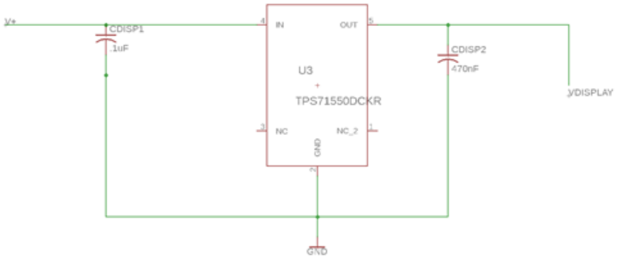

Fig. 8. 10V to 5V conversion with linear regulator and input bypass and output capacitors.

Furthermore, the TPS71550 regulator was chosen for the 10V to 5V conversion because it is a linear regulator with low noise and is large enough to be soldered by hand.

B. Programming Interface

The programming interface is a crucial aspect of the project because it allows the user to program the microcontroller. The MSP430FG47x has four supported developmental tools that could be used for programming. However, the MSP-FET430UIF (USB) was chosen because it allows the microcontroller to be programmed while soldered to a PCB board. [5] It also works with Code Composer Studio which is the programming software chosen for the project. The MSP-FET430UIF connects to the microcontroller through the PCB board using the JTAG connections shown below. [7]

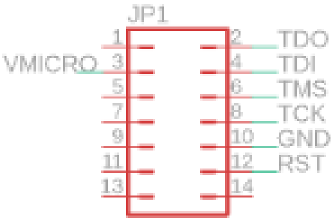

Fig. 9. Programming interface connections

C. Schematic Design

The schematic design consists of several sections including the DC-to-DC conversions in Fig. 7 and Fig. 8, programming interface from Fig. 9, and the analog portion of the design from Fig. 2. Additional sections include the LCD connections to the microcontroller as well as the user interface to the system implemented through several buttons. Autodesk EAGLE software was used for the schematic and PCB board design.

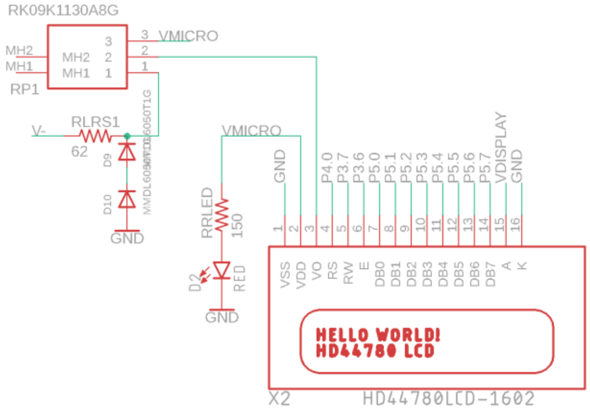

Fig. 10. LCD connections to microcontroller.

Fig. 10 shows the LCD connections to the microcontroller for programming. A red LED was connected to the display's supply voltage for troubleshooting purposes. This allows users to easily tell if the LCD is being properly powered. A potentiometer is connected to the third pin (VO) to adjust the contrast of the LCD.

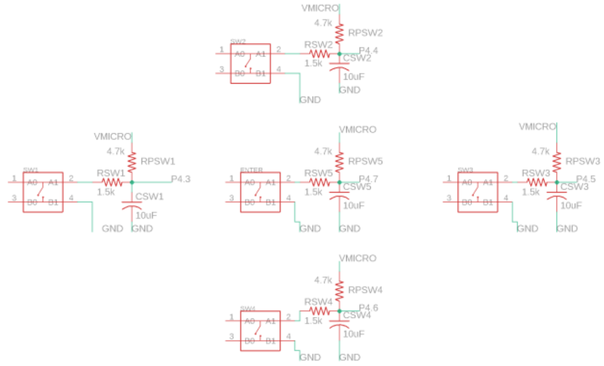

Fig. 11. User interface using buttons.

Fig. 11 shows the user interface to the system implemented through several buttons. The user will be able to set the desired resistance values pressing the 'resistance up' and/or the 'resistance down' buttons. The buttons were attached to RLC circuits and pull-up resistors to ensure more accurate readings from the mechanical connection to the microcontroller.

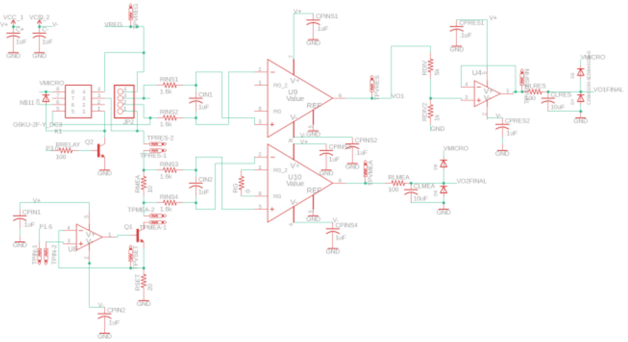

Fig. 12. Analog schematic.

Fig. 12 shows the analog portion of the schematic, which incorporates the analog portion of the design from Fig. 2. This design also includes a relay for operation protection and voltage limiters to ensure that the microcontroller is protected from voltages over 3.6V on its I/O pins. The designs for each portion of the board are all in included in the overall schematic design on separate sheets.

D. Board Design

From the final project schematic, the PCB can be designed. The Autodesk EAGLE footprints for each chosen component were found and downloaded to ensure proper connections to the PCB. Components were placed strategically to ensure best operating practice. For example, the bypass capacitors were placed as close to the voltage sources as possible to minimize noise at that point.

Fig. 13 shows the final populated printed circuit board for the project. The design is compact in an attempt to minimize noise from excessively long traces. Each portion of the circuit (the analog portion, each voltage regulator, and the switches) have been compartmentalized for ease of troubleshooting.

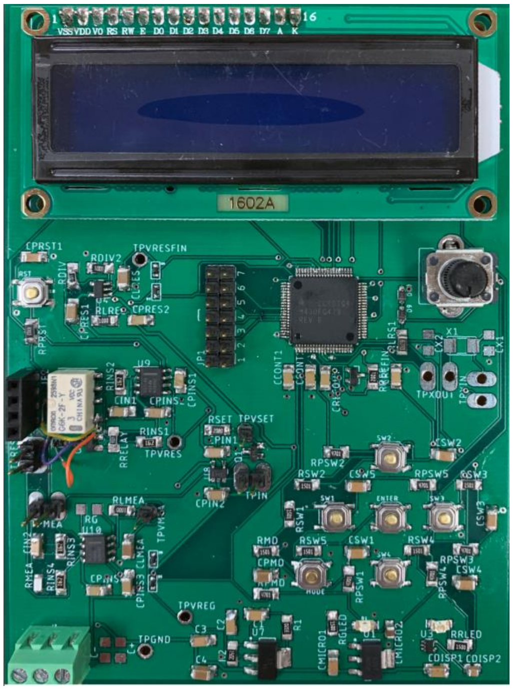

Fig. 13. Project PCB.

VII. RESULTS

The following chart depicts the results of the control loop system. Once the resistance reached its desired setpoint, the overshoot can be seen on Figure 14 as a percentage of the desired resistance value.

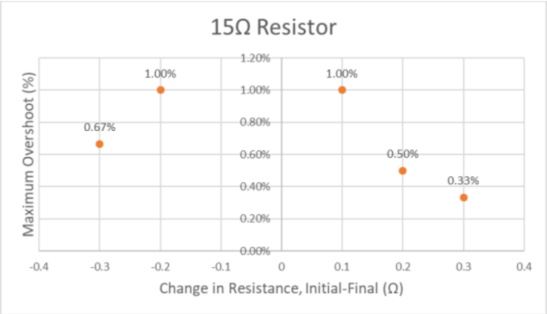

Fig. 14. Percent overshoot based on change in resistance for a 15Ω resistor.

Furthermore, the time to the overshoot can also be seen in Figure 15. Once the peak overshoot is reached, the resistance again begins to approach its target value. Once the target value is reached, the resistance varies only ±1mΩ from the desired value.

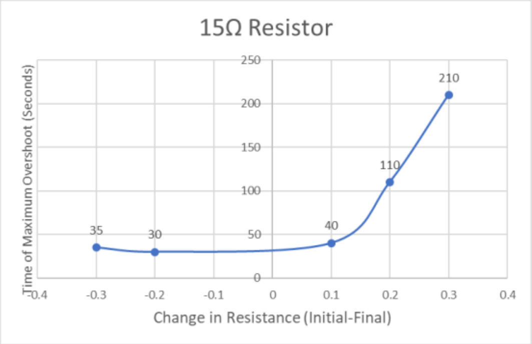

Fig. 15. Time to maximum overshoot based on change in resistance for a 15Ω resistor.

VIII. CONCLUSION

For this project, a printed circuit board design has been completed to control the resistance of a TPoS MEMS oscillator. The unique temperature properties of a TPoS MEMS resonator indicate that its performance (stabilization of the resonator frequency) could be optimized using an oven-control circuit. Hence, the resonator has been heated through Ohmic Resistive Heating for this project. A microcontroller adjusts the current through the resonator to compensate for ambient temperature variations. However, since the temperature data requires characterization of each device, this project will instead seek to control the resistance of the resonator, which is directly related to the temperature.

Thus, this project has sought to set the resistance value of a MEMS TPoS resonator to examine its performance at a stable resistance (and thus temperature).

IX. TEAM MEMBERS

Megan Driggers is an electrical engineering major who will be working with the Naval Warfare Center in Panama City, Florida. She intends to begin a master's in electrical engineering through Florida State University-Panama City Campus in the Fall of 2019.

Heather Hofstee is an electrical engineering major with a minor in math who will be a semiconductor manufacturing engineer with Texas Instruments in Dallas, Texas. She intends to pursue a graduate education in engineering or computer science in the near future.

Michaela Pain is a computer engineering major who will be returning to Intel, where she was previously an intern, as a full-time platform application engineer in Portland, Oregon.

ACKNOWLEDGEMENTS

The authors wish to acknowledge Dr. Reza Abdolvand for his sponsorship and support. Testing was done in his research lab at the University of Central Florida under his guidance and encouragement.

REFERENCES

Adafruit. (2009, June 4). *Data Sheet TC1602A-09T SpecV00*. Retrieved October 26, 2018, from https://cdn-shop.adafruit.com/product-files/181/p181.pdf

Charmet, J., Daly, R., Thiruvenkatanath, J., Woodhouse, J., & Seshial, A. (2014). Observations of modal interaction in lateral bulk acoustic resonators. *Applied Physics Letters*, *105*, pp. 1-4. doi:10.1063/1.4887015

Khamoushi, K. (n.d.). *TEMPERATURE COEFFICIENT OF RESONANT FREQUENCY (TCF)*. Retrieved from DIELECTRIC MATERIAL: http://www.students.tut.fi/~khamousk/TCF.htm

R. Abdolvand et al. (2016). Micromachined Resonators: A Review. (J. Lötters, Ed.)

Shahraini, S., Abdolvand, R., & Fatemi, H. (2018). Temperature Coefficient of Frequency in Silicon-Based Cross-Sectional Quasi Lamé Mode Resonators. *IEEE International Frequency Control Symposium.* Olympic Valley.

Texas Instruments. (n.d.). *MSP430 Hardware Tools User's Guide (Rev. AE)*. Retrieved 03 26, 2019, from http://www.ti.com/lit/ug/slau278ae/slau278ae.pdf

Texas Instruments. (n.d.). *MSP430FG47x Mixed Signal Microcontroller datasheet (Rev. D)*. Retrieved 03 26, 2019, from http://www.ti.com/lit/ds/symlink/msp430fg479.pdf