UNIVERSITY OF CALIFORNIA,
IRVINE

Impedance Spectroscopy for Medical Sensing Applications

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical Engineering

by

Seyed Ali Hashemi

Thesis Committee:
Professor Rahim Esfandyarpour, Chair
Professor Peter Burke
Professor H. Kumar Wickramasinghe

2021

TABLE OF CONTENTS

Table of Contents

LIST OF FIGURES .... *iii*

ACKNOWLEDGEMENTS .... *iv*

ABSTRACT OF THE THESIS .... *v*

INTRODUCTION *1*

Biosensing Applications and Impedance Spectroscopy 2

Theory of Operation 4

Hardware Design .... *Error! Bookmark not defined.*

Top Level Design 7

Power Supply 7

Microcontroller 11

USB Bridge 13

Connectors 14

Sampling System 16

Printed Circuit Board (PCB) 27

Firmware Design .... *Error! Bookmark not defined.*

Firmware Structure 30

Sampling System Algorithm 30

USB Communication 32

Software Design .... *Error! Bookmark not defined.*

Communication Interface and Structure 33

Capabilities 33

Device Evaluation and Specifications .... *Error! Bookmark not defined.*

Live Plot 35

Frequency Sweep 37

Real Time Response 38

CONCLUSION *40*

Works Cited *41*

LIST OF FIGURES

Figure 1: Schematic Diagram of Interdigitated Electrode Transducer [2] 2
Figure 2: Transducer Equivalent Circuit [2] 3
Figure 3: Circuit Hierarchy 7
Figure 4: Ferrite Bead Circuit Model [8] 8
Figure 5: LDO PSRR Graph [8] 9
Figure 6: Power Supply 10
Figure 7: MCU Power Good Circuitry 11
Figure 8: MCU Schematic 12
Figure 9: USB Bridge 14
Figure 10: Sensors on FFC PCB 15
Figure 11: FFC PCB 3D Model 15
Figure 12: AD5933 Block Diagram 17
Figure 13: AD5933 Improved Block Diagram 18
Figure 14: Multiplexer Circuit 21
Figure 15: AD5933 Calibration MUX 23
Figure 16: Sampling System 25
Figure 17: Gain Factor for Single Point Calibration 26
Figure 18: Gain Factor Variation for 2-Point Calibration 26
Figure 19: PCB 3D Model 28
Figure 20: PCB Picture 29
Figure 21: AD5933 Algorithm 31
Figure 22: Software Screenshot 34
Figure 23: PCB Live Plot Test 36
Figure 24: Zurich Live Plot S0 36
Figure 25: Zurich Live Plot S8 36
Figure 26: PCB Frequency Sweep 37
Figure 27: Zurich Frequency Sweep 38
Figure 28: PCB Real Time Response 39
Figure 29: Zurich Real Time Response 39
Figure 30: Sensor with Beads 39

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my committee chair, Professor Rahim Esfandyarpour, for his patience and knowledge. Without his guidance and persistent help this dissertation would not have been possible.

I would like to thank my committee members, Professor Peter Burk, and Professor H. Kumar Wickramasinghe whose work inspired us to keep going when the going got tough.

In addition, a massive thank you to Xiaocheng Pei and Mahsius Sumi of the University of California, Irvine, who supported me in every step of this journey. This thesis would not be possible without them.

ABSTRACT OF THE THESIS

Impedance Spectroscopy for Medical Sensing Applications
by
Seyed Ali Hashemi
Master of Science in Electrical Engineering
University of California, Irvine, 2021
Professor Rahim Esfandyarpour, Chair

Most sensors that serve important purposes in the medical industry are impedance based. This has created the need for impedance sampling systems in the medical industry. Most of these systems are rather large, difficult to use, and can only sample one item at a time. In this paper we explore the design of an impedance sampling system that can take input from 100 sensors simultaneously. The system communicates and is powered through a micro-USB connection. The front end can provide an excitation voltage of 200mV, 400mV, 1V, and 2V peak-to-peak and frequency of 0.1Hz – 100kHz. There are two modes of operation: frequency sweep and live plot. The frequency sweep mode is used to test an unknown impedance to see what voltage and frequency it responds best to. The data collected is then used to setup the live plot. Conversions are done by providing the excitation signal to an unknown impedance and measuring the current through the impedance to back calculate the impedance via a propriety discrete Fourier transform (DFT) algorithm. To maximize dynamic range and performance, a multiplexer is used at the transimpedance amplifier to provide variable gain based on the maximum input range and selected peak-to-peak excitation voltage. The impedance range measurable by the system is 1Ω - 1MΩ. Due to the complexity and low-noise nature of the device, calibration is an extremely important step to getting the best performance out of the system. To facilitate easier calibration, a multiplexer is tied to the input that connects known, accurate, and low drift impedances so that users can call for autocalibration of the device. Software is provided so that users can adjust start/end frequency, number of steps, excitation voltage, and excitation frequency depending on which mode they are using the device in. Users can also recalibrate a device for a specific voltage through the software by the click of the button. Plots are provided for both modes and can be exported through the software along with raw data.

INTRODUCTION

The evolution of sensing technology has been one of the main catalysts to pushing the medical industry forward. Advancements in micro-electromechanical systems (MEMS) have brought forth vast improvements to biomedical sensors. Silicon-based MEMS sensors have the capability to measure various biological properties including glucose, estrogen, ion concentration, etc. These sensors are usually made of two or more electrodes and are generally impedance based. While voltage and current based sensors exist, they are less common than impedance-based sensors and they are not as sensitive and do not provide as much information as an impedance-based sensor, either. The challenge that comes with impedance-based transducers is making sense of the signal. Voltage-based sensors are the easiest to work with as you simply measure the voltage they output. Bandgap sensors don't require an excitation input, either, which makes them even easier to work with. Current-based sensors are also easy to work with. The current is converted to a voltage and then measured. Impedance-based sensors become tricky as they require an alternating current (AC) excitation signal and their readback must also be done in AC. Converting the readback values to something meaningful to the user is also a challenging task. This paper will discuss a simple and compact solution to sampling these impedance-based sensors. We will discuss the theory of operation, circuit and PCB design, firmware design, and software that all work together. We will also check the results of this design versus well-known impedance analyzers.

Biosensing Applications and Impedance Spectroscopy

One of the biggest and most important applications of measuring impedance is for electromechanical impedance spectroscopy (EIS) which is a method for translating complex chemical responses to an electrical signal that can be measured and analyzed [1]. EIS measures the impedance of a system as a function of frequency and an applied excitation, normally a small AC signal [2]. This method is generally applied to biosensors that are formed utilizing interdigitated electrodes. EIS analyzes the resistance and capacitance that occur at the electrode surface, which are very sensitive to biological binding events. Figure 1 shows a very simple example of a two-electrode interdigitated electrode transducer [3].

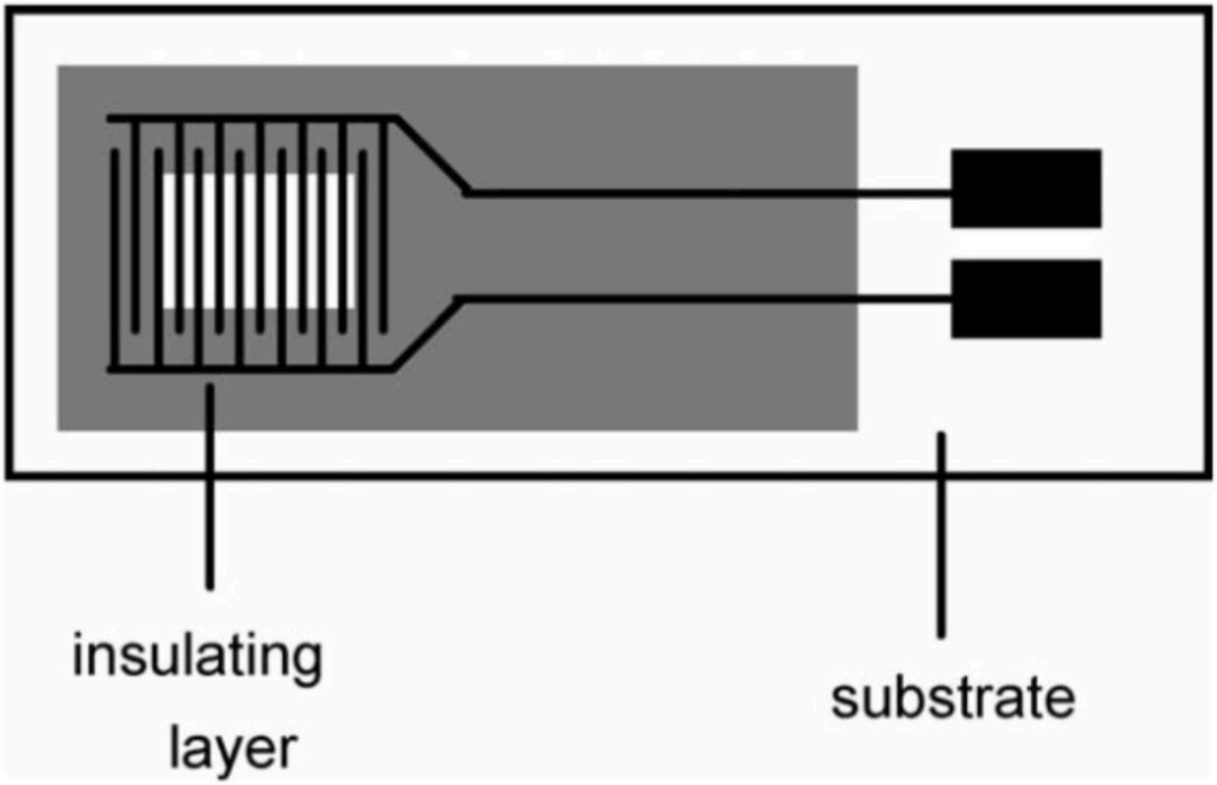


Figure 1: Schematic Diagram of Interdigitated Electrode Transducer *[2]*

An equivalent circuit model for an immunoglobulin detection circuit can be found below.

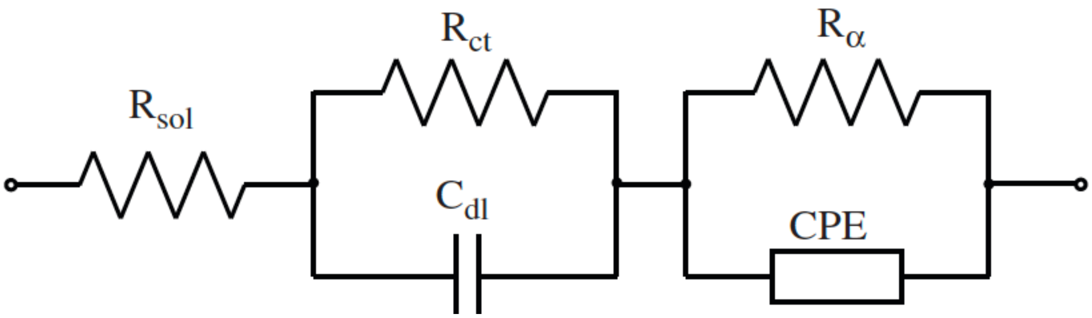


Figure 2: Transducer Equivalent Circuit *[2]*

There are a wide range of biosensors/transducers that can provide incredibly helpful information to doctors, labs, and patients. These types of sensors can measure hemoglobin levels, glucose levels, ion concentration, detection of immunoglobulin, etc. to name a few. Traditional applications include immunosensing protocols, utilizing electrode-immobilized antibodies to register protein-protein interactions, or to detect smaller molecules, such as antibiotics in complex biological media [2]. The field is growing rapidly and there are many new publications on the use of this method for Geno sensing utilizing electrodes with immobilized ss-DNA probes which detect complementary ss-DNA target molecules through hybridizations. The uses aren't just limited to protein-protein or DNA-DNA interactions, either. Electrodes with immobilized dsDNA were used to detect specific binding of anticancer drugs. The method in its entirety is very sensitive and can be used for a "label-free" detection of a multitude of molecular recognition events happening at the electrode surface [4]. Additionally, there have been recent reports of EIS sensors for Endocrine disruptors (ED). For humans, these compounds may cause cancerous tumor and infertility. Various EIS based biosensors have been recently developed for detection of EDs.

Theory of Operation

Before we get into the electronic theory of this application, let's take a step back and look at two fundamental transformations in calculus.

$$s = \frac{d}{dt} \tag{1}$$

$$\frac{1}{s} = \int dt \tag{2}$$

In equations (1) and (2), s is the Laplace frequency. These transformations are what allow EIS to be feasible. The general method to measuring an unknown impedance is like measuring an unknown resistance. We start with Ohm's Law:

$$V = IZ \tag{3}$$

where if we have purely real impedance, $Z = R$. We now re-write (3) to get (4).

$$V(s) = I(s) * Z(s) \tag{4}$$

We now re-write (4) to get (5) in terms of impedance.

$$Z(s) = \frac{V(s)}{I(s)} \tag{5}$$

Since we are looking for Z, we need to provide V and measure I [5]. The excitation voltage, V, is dependent on the experiment we are running. It needs to be variable both in terms of amplitude and frequency. Ideally, V would be centered around 0VDC. Since we are digitizing this signal, the voltage needs to be offset the mid-point. Analog-to-digital converters (ADC) generally do not accept negative voltage inputs. While this isn't the case for all ADCs, we adjust the midpoint so that we can work with all ADCs regardless of negative input capability. The mid-point needs to be chosen as the midpoint of the highest voltage the circuit is providing. We need to ensure that this is also matched to the highest voltage the ADC can measure. This can usually best be done by using a circuit to cut the ADCs reference in half and then add it to the measured AC signal. To measure the current through the unknown impedance, a transimpedance stage is needed. The excitation voltage is fed into one leg of the unknown impedance and the other leg is the input to the transimpedance stage. It is possible to convert a current into voltage by simply using a resistor, but the advantages of using a transimpedance amplifier are that the voltage bias across the input is constant and the input/output impedance is very low even with very high gain [6]. The output of the transimpedance stage is then fed into an ADC so that the collected data can be processed. Note that the ADC provides a voltage value at a particular time. To make the information useful to the user of the end device, we need to use the voltage and current data together. The voltage provided to the sensor (excitation voltage) needs to also be recorded. This data, along with the current data at the same point in time, are used to back-calculate the impedance. To derive the real and imaginary part of the unknown impedance, a discrete Fourier transform is applied. The algorithm is represented by (6).

$$X(f) = \sum_{n=0}^{1023} \left(x(n)\left(\cos(n) - jsin(n)\right)\right) \tag{6}$$

Where $X(f)$ is the power of the signal at the frequency point f, $x(n)$ is the ADC output, $\cos(n)$ and sin (n) are the sampled test vectors provided by the DDC core at the frequency point f. The multiplication is accumulated over 1024 samples for each frequency point. This algorithm then provides a real and imaginary value for each sample in time. From the real and imaginary parts of the unknown impedance we can calculate magnitude $|Z|$ and phase ϕ_z with (7) and (8) [7].

$$|Z| = \sqrt{\{Re(Z)\}^2 + \{Im(Z)\}^2} \tag{7}$$

$$\phi_z = \tan^{-1}\frac{\{Im(Z)\}}{\{Re(Z)\}} \tag{8}$$

HARDWARE DESIGN

Top Level Design

This project aimed to create a device that could perform EIS on 100 sensors at a time, have an easy mechanism to plug and unplug sensors from, and communicate and be powered through USB. To make this happen, we created a few fundamental building blocks: power supply, sampling system, microcontroller, USB bridge, and connectors. Figure 1 provides insight on how the hierarchy of the system is design and interconnected.

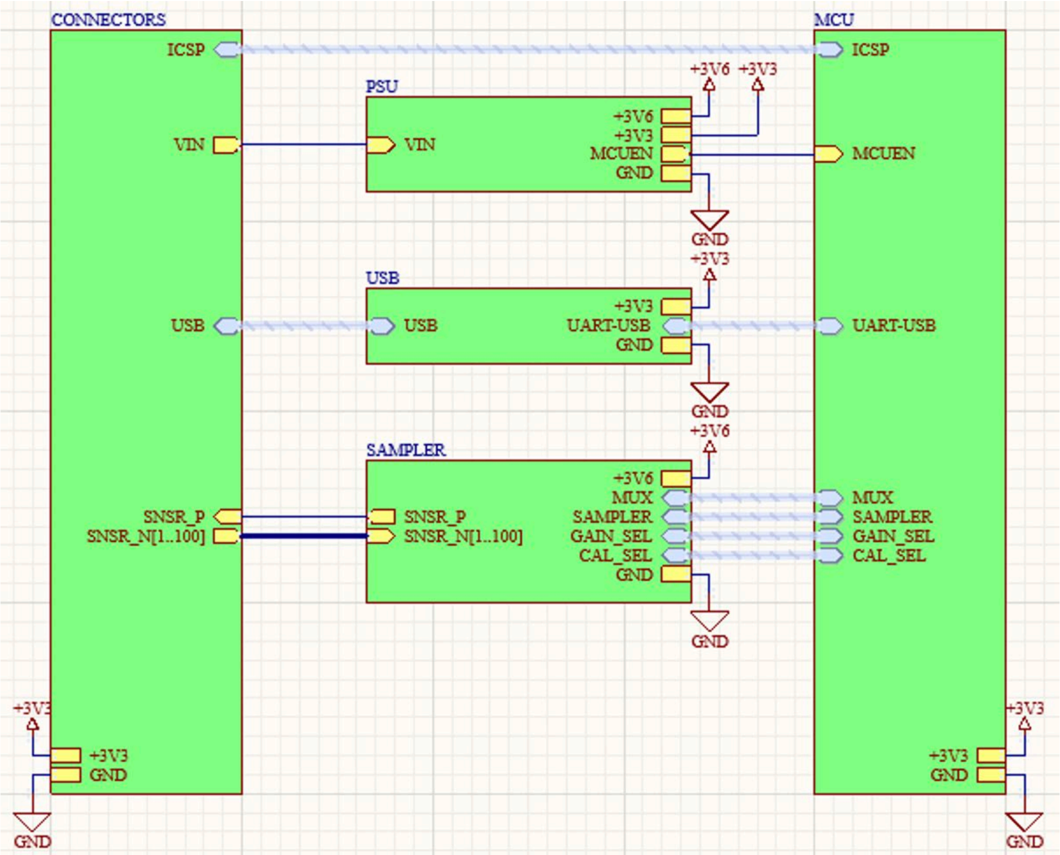


Figure 3: Circuit Hierarchy

Power Supply

The device is design to work with USB. It contains a USB micro connector that facilitates both power and communication. Since the worst-case USB power specification is 4.4VDC – 5.25VDC, an Analog Devices LT8609A DC-DC converter with a buck topology is used to create a 3.6VDC rail. None of the components on this board require a rail higher than 3.3VDC. We chose 3.6VDC to be the output of the switching stage for efficiency. This voltage is right where the maximum dropout of the low-dropout linear regulators (LDO) we will use is. The DC-DC input has a low pass filter to filter any current spikes coming back to the supply. This helps keep the USB bus clean from noise and helps avoid crashing the bus for any power related reasons. The output is also filtered with a low pass filter to eliminate high frequency noise on the board. This is specifically important for a large design like this since we have small analog AC voltages and currents being measured across a very large surface. The filters are realized using ferrite beads instead of inductors or resistors as ferrite beads are better at rejecting noise than inductors and they don't have the voltage drop issues that come with resistors. Ferrite beads not only shunt noise to ground, but they also give noise off as heat whereas inductors only shunt noise to ground. The equivalent circuit model of a ferrite bead can be found in figure 4.

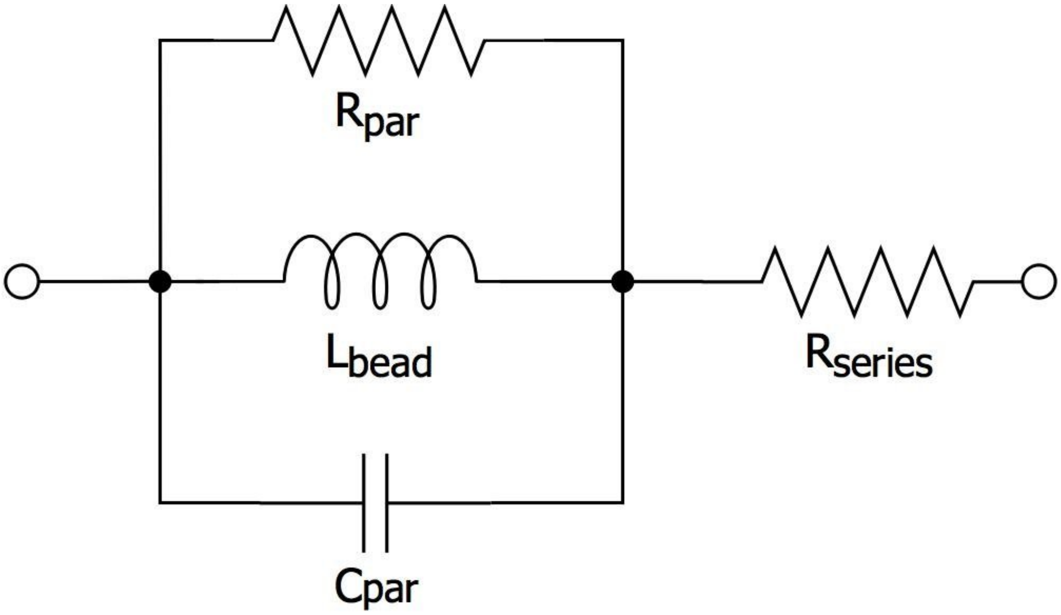


Figure 4: Ferrite Bead Circuit Model *[8]*

Microchip MIC5319 LDOs are used to generate the 3.3VDC rails the circuit requires. LDOs are used for their superior noise and regulation performance in comparison to a DC-DC converter. These devices are specifically used as they have very good power supply rejection ratio (PSRR) across a wide frequency range. The PSRR graph for this particular regulator can be found in figure 5.

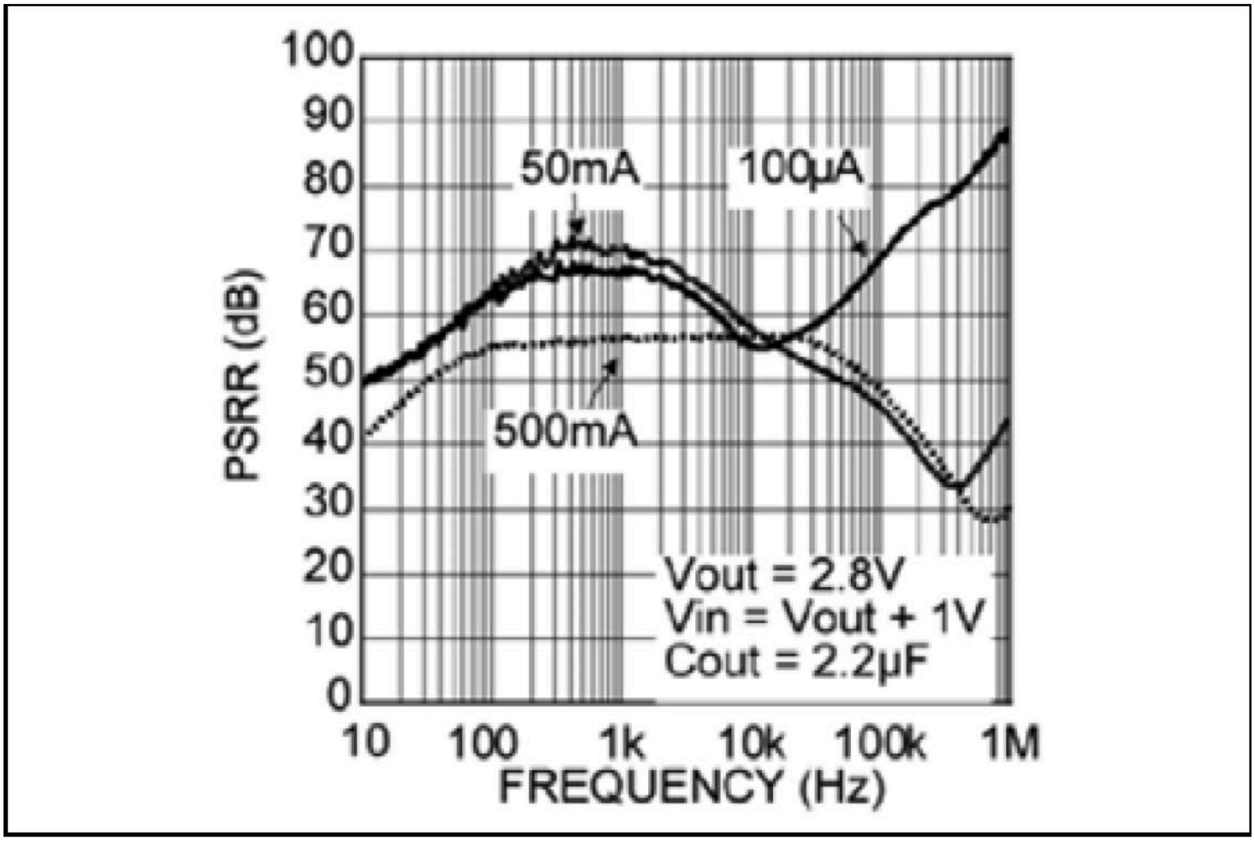


FIGURE 2-1: *PSRR (Bypass Pin Capacitor = 0.1 µF).*

Figure 5: LDO PSRR Graph *[8]*

The operating frequency of the LT8609A is set so that it falls in the high rejection area of the PSRR. This method helps reject as much switching noise from the DC-DC as possible. A global 3.3VDC LDO is used to supply power to all the digital components not associated with the sampling system. This is done to localize the 3.6VDC voltage to only the LDO right next to it and not across the entire board. The sampling system has its own 3.3VDC supply to help localize its current meshes and avoid pickup up switching noise from the digital circuitry. This helps keep the sampling system as low-noise as possible. The overall power supply design can be found in figure 6.

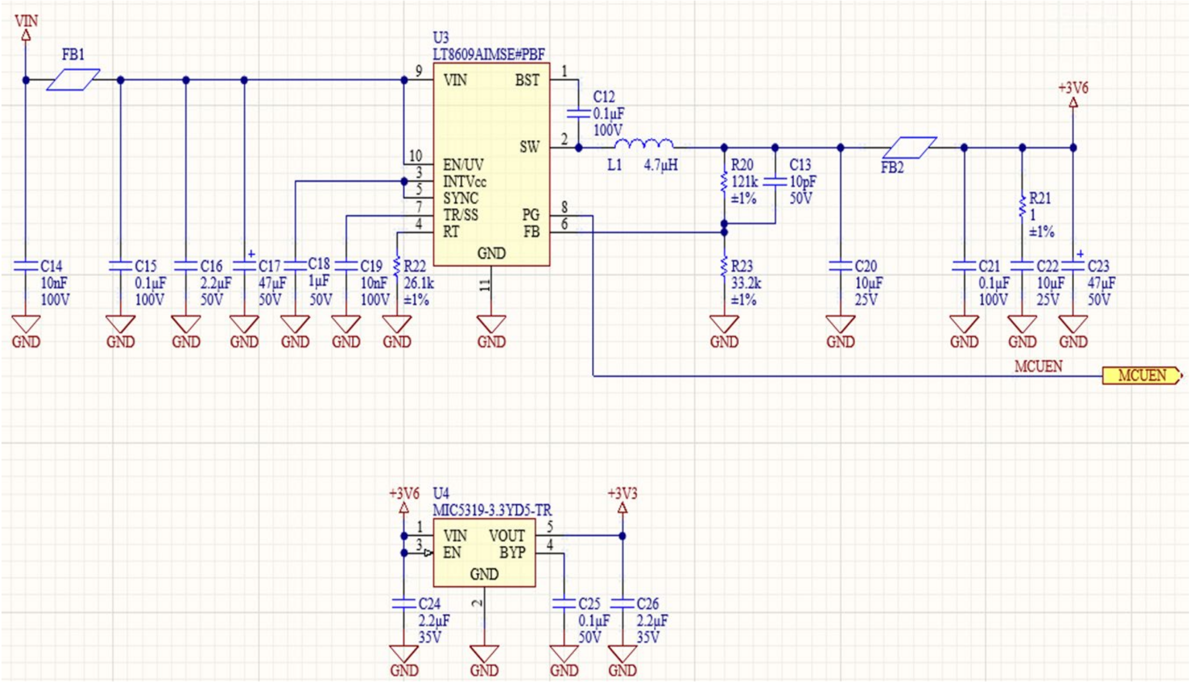


Figure 6: Power Supply

The LT8609A provides a power good signal which is also used to turn the rest of the circuitry on. It holds the MCU in reset until power is good which in turn halts all execution. This ensures that the MCU doesn't start when power isn't stable to avoid any and all initialization errors. The circuitry used to achieve this can be found in figure 7. R4 serves as a pull up for the MCU reset pin and it also biases Q1. R3 serves as a pull-up on the power good pin since it is active low. R5 is used to bias the base of Q1.

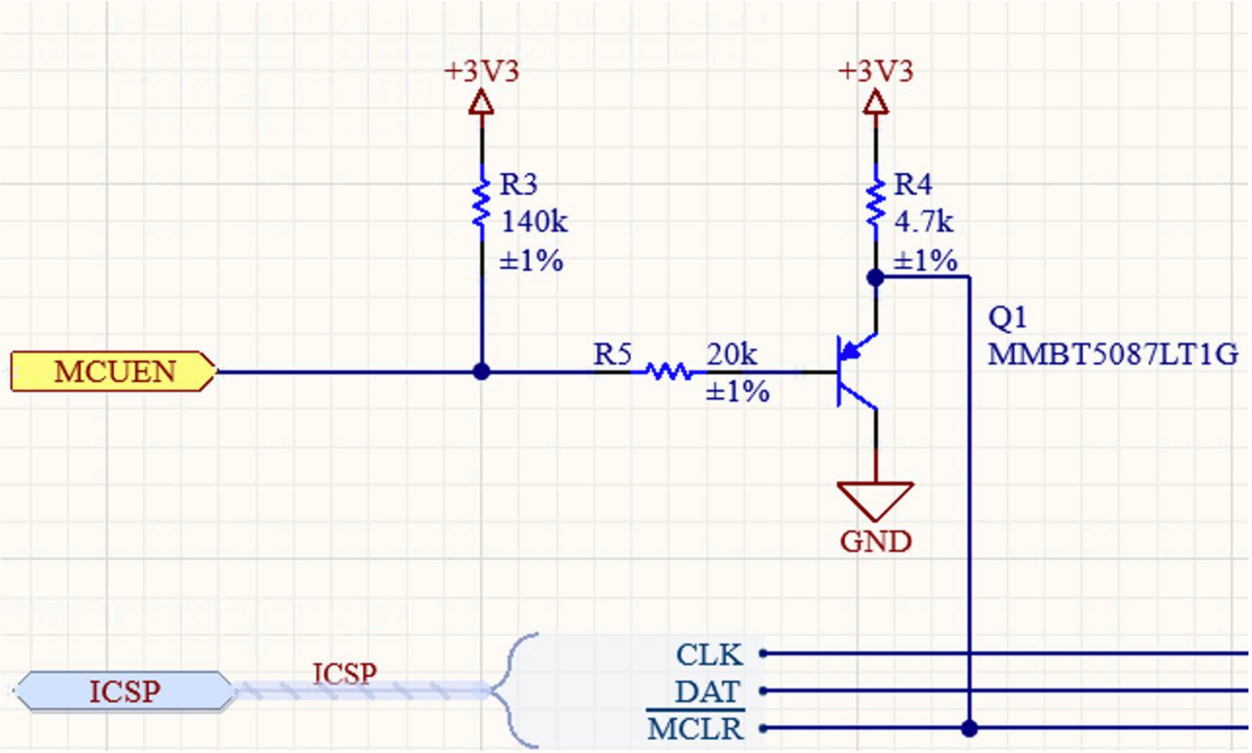


Figure 7: MCU Power Good Circuitry

Microcontroller

The chosen microcontroller for this device is the Microchip PIC32MZ2048EFG100. The 2 megabit (Mb) 100-pin version was used just for future expandability and because there is plenty of room on the board, but it is not necessary. Smaller versions with less random-access memory (RAM) can be used as well. The PIC32MZ is a great fit for this project as it has plenty of peripherals, like multiple SPI, UART, and I2C ports, as well as a very powerful FPU incase post processing of data is required. A Microchip external EEPROM is also used to store data like the device serial number and calibration points. The MCU schematic can be found in figure 8.

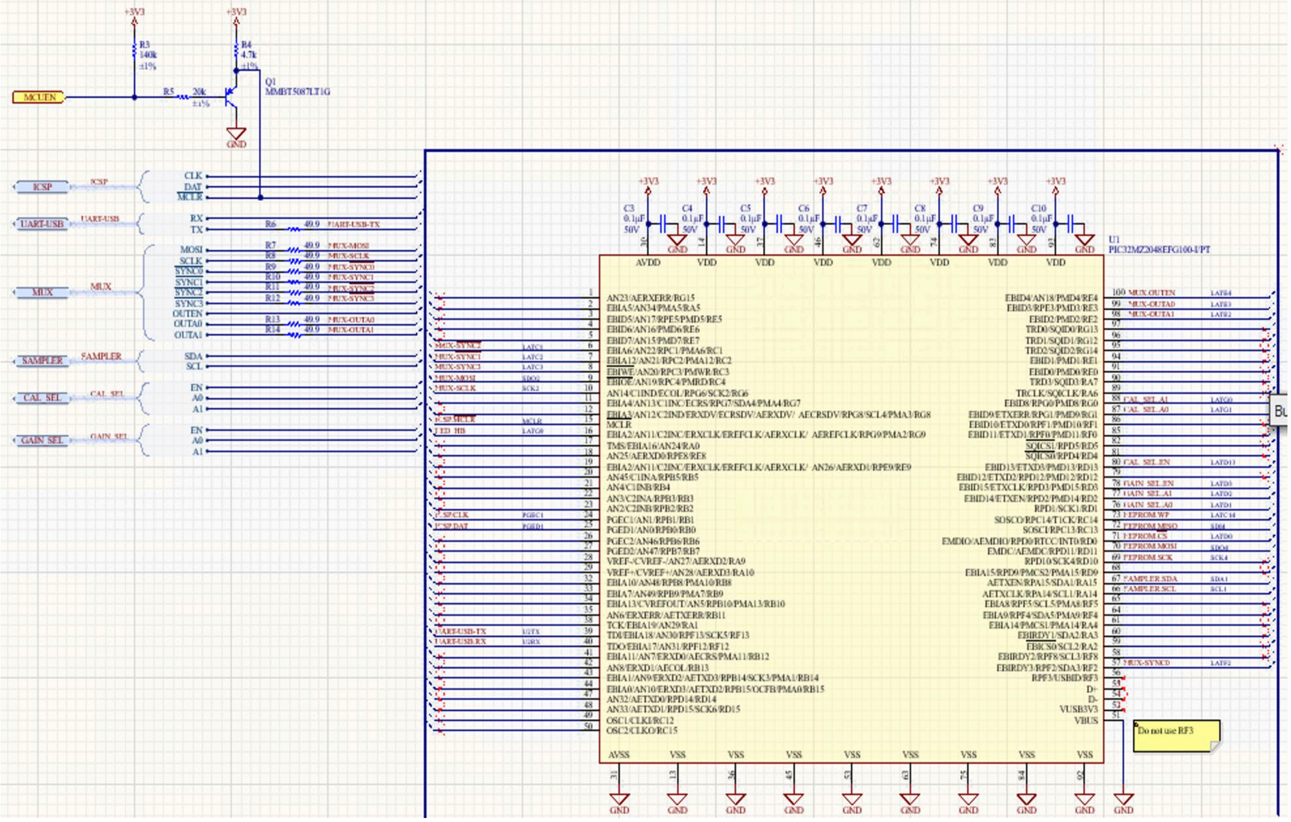


Figure 8: MCU Schematic

One of the biggest benefits of using a PIC MCU is the flexibility of the pins and peripherals. With these devices, you can map nearly any pin to any peripheral. This greatly helps the PCB design become streamlined and clean. The peripherals are also very easy to use and save tons of development time. The I2C peripheral handles all start/stop sequences and provides errors on the bus as they happen. It is extremely easy to follow the special function registers (SFR) and configure them to your liking. The SPI bus is also fairly streamlined. Both of these peripherals have deep buffers. Instead of single byte buffers, they offer up to 16 bytes deep. This helps in firmware development as all data can be written out together, then all responses can be collected. The firmware doesn't need to write/read, write/read, etc. All data can then be processed while waiting on the next sample. Since the device has multiple SPI buses, we don't need to share any buses and write schedulers, either. The UART peripheral is also very streamlined. It offers baud-rates of up to 4,000,000, which can come in handy when expanding the board to more channels. Each channel needs to send 32 bytes of data out along with the channel number; thus, this can add up fairly quickly. As mentioned earlier, the MCU uses a global 3.3VDC supply that is also shared with the USB bus and EEPROM chip. For best performance, decoupling capacitors are placed as close as possible at every single power pin.

USB Bridge

An FTDI FT230X UART to USB bridge is used to communicate between the board and a computer. This chip was chosen as it is easy to use and requires no configuration. This isn't a high throughput USB bridge but for this application, that is irrelevant. While there is data coming in from 100 channels, the data rate per second is not high, thus making this chip the perfect candidate for this design. The circuit for this block can be found in figure 9. The device can be powered in many ways, but we've chosen to power it off the board rather than straight from the USB port to avoid the chip feeding any voltage to the MCU while the MCU is off or in a reset state. A resistor divider is used to sense the USB bus voltage and feed it to a preconfigured pin to let the device know when the bus is connected and operational. Additionally, the chip has preconfigured pins for RX and TX LED indication which are used for diagnostics. Due to the size of the board and length of traces, a termination resistor is used on the UART pins to avoid any ringing. Ringing can cause incorrect interpretation of data on both the RX and TX sides of the communication. Decoupling capacitors are used as well on this chip since the supply is shared with the MCU and EEPROM. A bulk capacitor is also required by the datasheet. The USB bus is noted as a differential pair and is routed as such for best performance. Since the USB signal is high frequency, keeping the differential pairs equally spaced and long provides the best and most safe performance. The reset pin on this chip is not used. It can be tied to the MCU to reset in case of communication issues, but this is extremely unlikely. There is also no way to issue a reset command if the MCU can't communicate with the computer via USB.

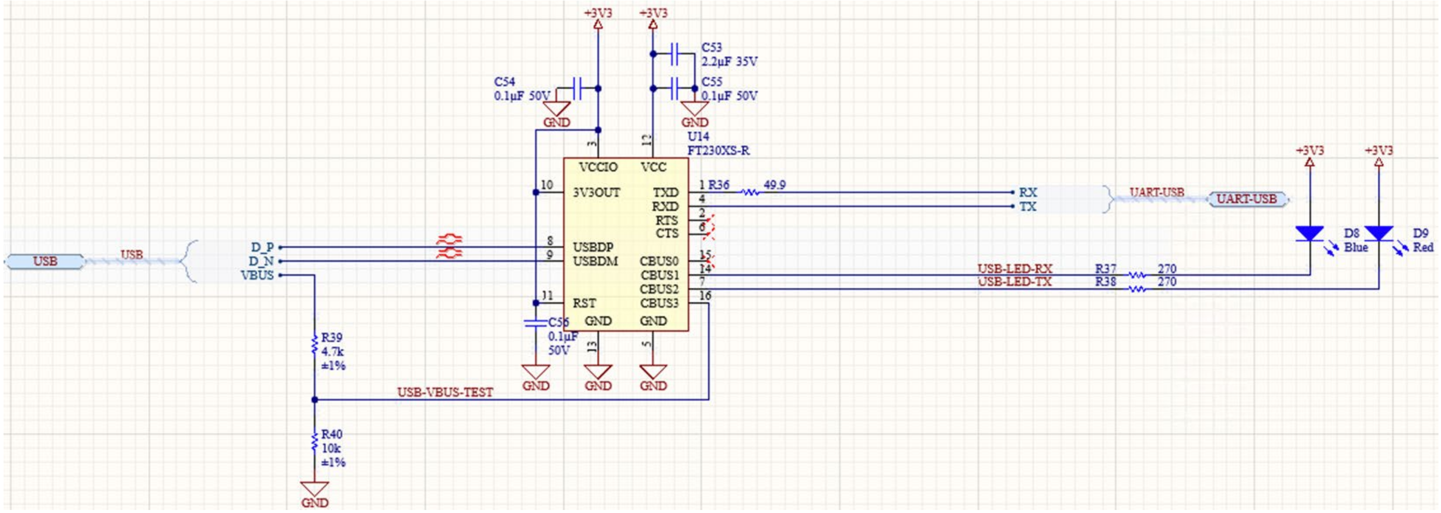


Figure 9: USB Bridge

Connectors

There are 102 connectors on this board: 100 flexible flat connectors (FFC) for the sensors, 1 programming connector to program the MCU, and 1 USB micro connector for power/communication. The FFC connector is used to allow for ease of plugging and unplugging sensors. The sensors are made on a silicon wafer and wire-bonded to a custom PCB that will plug into the board. The board isn't flexible, but it has the same front plug that a flexible circuit would have. The stackup is designed to act in such way that it will work with the FFC connectors. A picture of the sensor on the custom FFC PCB attached to the PCB can be found in figure 10. A 3D model of the adaptor can be found in figure 11. Note that the two through-hole pins that are seen towards the connecting end are for putting in pin-headers for experiments and testing. They are not used otherwise.

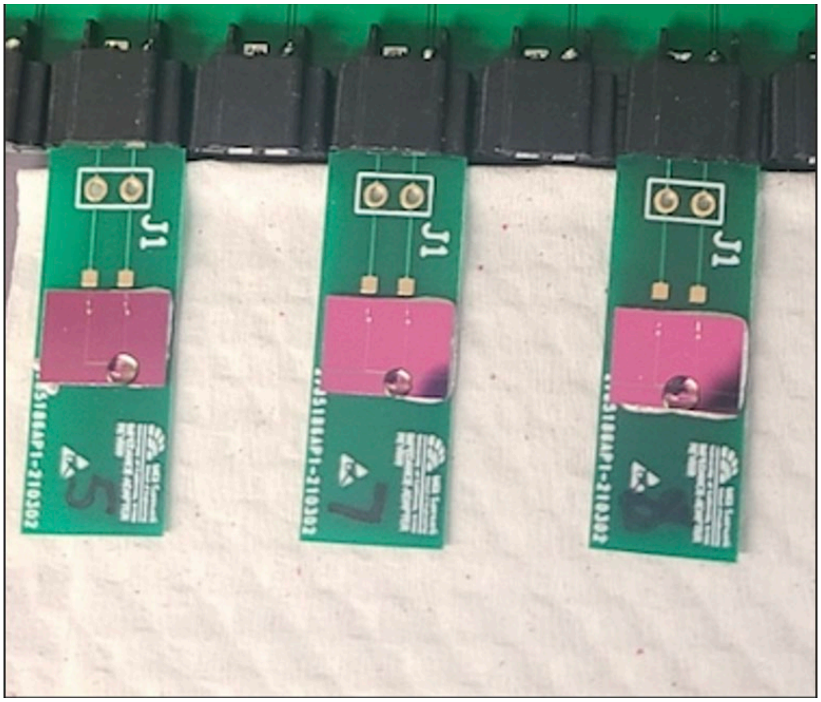


Figure 10: Sensors on FFC PCB

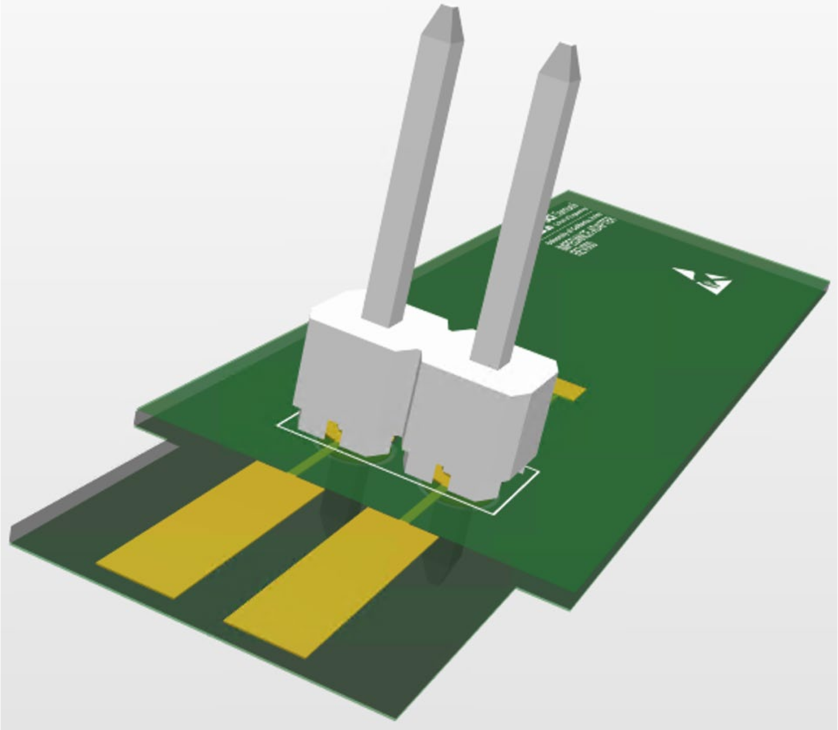


Figure 11: FFC PCB 3D Model

Sampling System

The heart of the sampling system for this device is the Analog Devices AD5933. This integrated circuit (IC) provides us with the capability to provide an excitation signal at 200mV, 400mV, 1V, and 2V peak to peak with a variable frequency from 0.1Hz – 100kHz. Additionally, it has a built in 12-bit 1M sample-per-second (SPS) ADC that is tied to a DFT engine that performs the conversion from a sampled AC signal to real and imaginary parts of an impedance. A block diagram of the device can be found in figure 11. The system also has a built-in gain block that is adjustable to 1x and 5x. The gain stage is followed by a low-pass filter stage. This stage is pre-set to abide by the Nyquist law to ensure proper sampling. There is also an internal temperature sensor that can be used to measure temperature on the die. This can be used to adjust for drift on both the excitation voltage and the gain stage, if desired. Users must be careful with how the algorithm for compensation is done. If done incorrectly, the algorithm can hurt the signal by causing further inaccuracies rather than helping correct it.

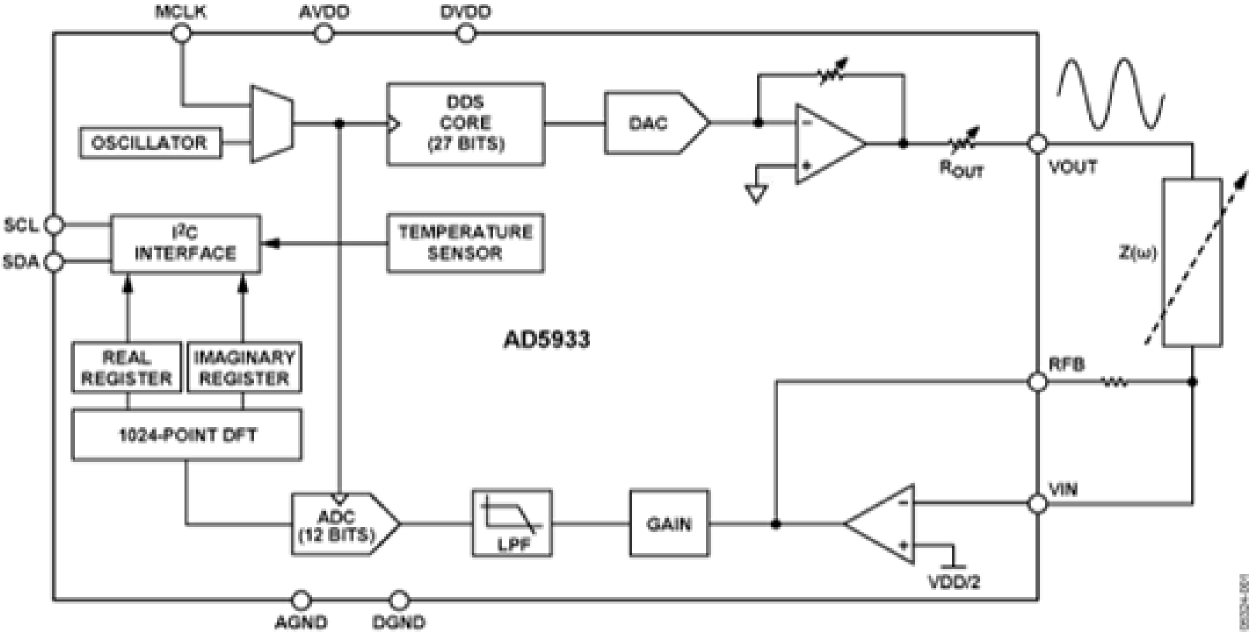


Figure 12: AD5933 Block Diagram

The device has good performance, but to get the best out of it, two external blocks are required: an external transimpedance amplifier on the input and an external biasing amplifier on the output. The excitation voltage supplied to the sensors cannot have any DC bleed-through and needs to be biased at VDD/2 as previously discussed. VDD is the supply voltage to the AD5933. We use an Analog Devices AD8606 which has two internal operational amplifiers for these improvements. A capacitor is used to block DC from the excitation signal generator and a divider is used to bias the signal. Ideally, the voltage to the divider should be from a voltage reference and not the LDO to avoid noise, drift, and any other inaccuracies. The resistors also should be the same value and part number to cut the value in half and to ensure drift happens in the same direction for both resistors. Drift on these resistors will affect the excitation signal which in turn affects the measurement accuracy. An external transimpedance amplifier is required for better noise and input/output impedance performance. We use the second amplifier in the AD8606 for this purpose. For even better performance, a separate IC can be used based on what is required for the design. The TIA must be biased to VDD/2 so that there are no common-mode input issues with the ADC as previously discussed. The manufacturer's suggested design can be found in figure 12.

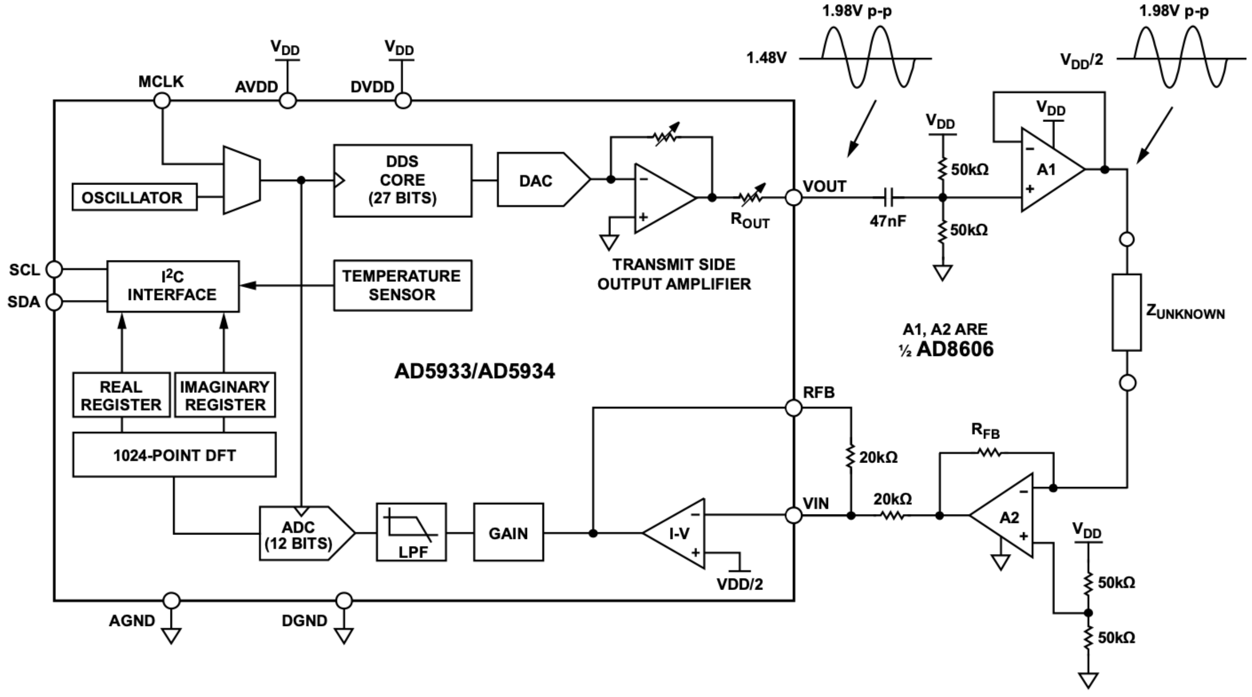


Figure 13: AD5933 Improved Block Diagram

Additionally, variable gain is needed on the TIA since excitation voltage is also variable. Since the device has 4 excitation voltage choices, we provide an external MUX with 4 low drift, high accuracy resistors to set the gain of the TIA based on the selected excitation voltage. The Analog Devices ADG704 MUX is used for this purpose as it is low noise, readily available, and easy to use. The device has poor on-state resistance, but this is irrelevant since we calibrate our device for all gain values. Equation 9 provides the method for deriving the gain resistance needed for each excitation voltage. V_{pk} is the excitation peak-to-peak voltage and Z_{min} is the minimum impedance that the design is required to measure down to. PGA is a value that is set in the AD5933 and is adjustable to 1 or 5. $V_{dcoffset}$ is fixed to VDD/2 in this design. This is done to ensure that the maximum dynamic range of the ADC is utilized so that we can maximize the resolution of the overall sampling system [8].

$$R_{gain} = (((\frac{VDD}{2}) - 0.2) * Z_{min}) / ((V_{pk} + (\frac{VDD}{2}) - V_{dcoffset}) * PGA) \quad (9)$$

Since we are using this device to sample many sensors at the same time, multiplexers need to be used on the input and an accurate external clock needs to be used to speed up the device so that we can time-multiplex. Time-multiplexing means that we multiplex the inputs to the sampling system and switch to the next channel once the current channel is finished sampling. To achieve proper sampling rates and throughput for time-multiplexing, we use a Microchip DSA633 pre-programmed clock. It is low noise and drift which aides in accurate sampling. We speed up the AD5933 to 200 SPS to maintain 1 SPS per channel. In theory, it should only be increased to 100 SPS. Since each channel needs time to settle before it can be sampled, we increase to 200 SPS to allow for the ADC conversion to settle. To achieve the capability of 100 inputs for 100 sensors, multiplexers were used to create more inputs. Since multiplexers come in powers of 2, we used 4 32-input multiplexers and 1 4-input multiplexer. The 100 sensor inputs are tied to the 4 32-input multiplexers and the 3 32-input multiplexers are tied to the 4-input multiplexer which is then connected to the input of the TIA stage. The multiplexers are chosen to have a very low input capacitance and low propagation delay since the signals we are working with are AC. The input/output impedance are not very important for our design. The input resistance does cause inaccuracy since we are measuring current, but it is calibrated out during calibration. The output resistance also is not important because it is again calibrated out and connected to the TIA stage. The calibration and gain multiplexers are also chosen to be the same device as the 4-input multiplexer used on the inputs to provide similar performance, drift, and inaccuracy as the input to increase overall accuracy. Note that R46 in figure 15 is a place holder and is not populated. It only exists in case the user wants to bypass the gain multiplexer. The multiplexing circuitry can be found in figure 13.

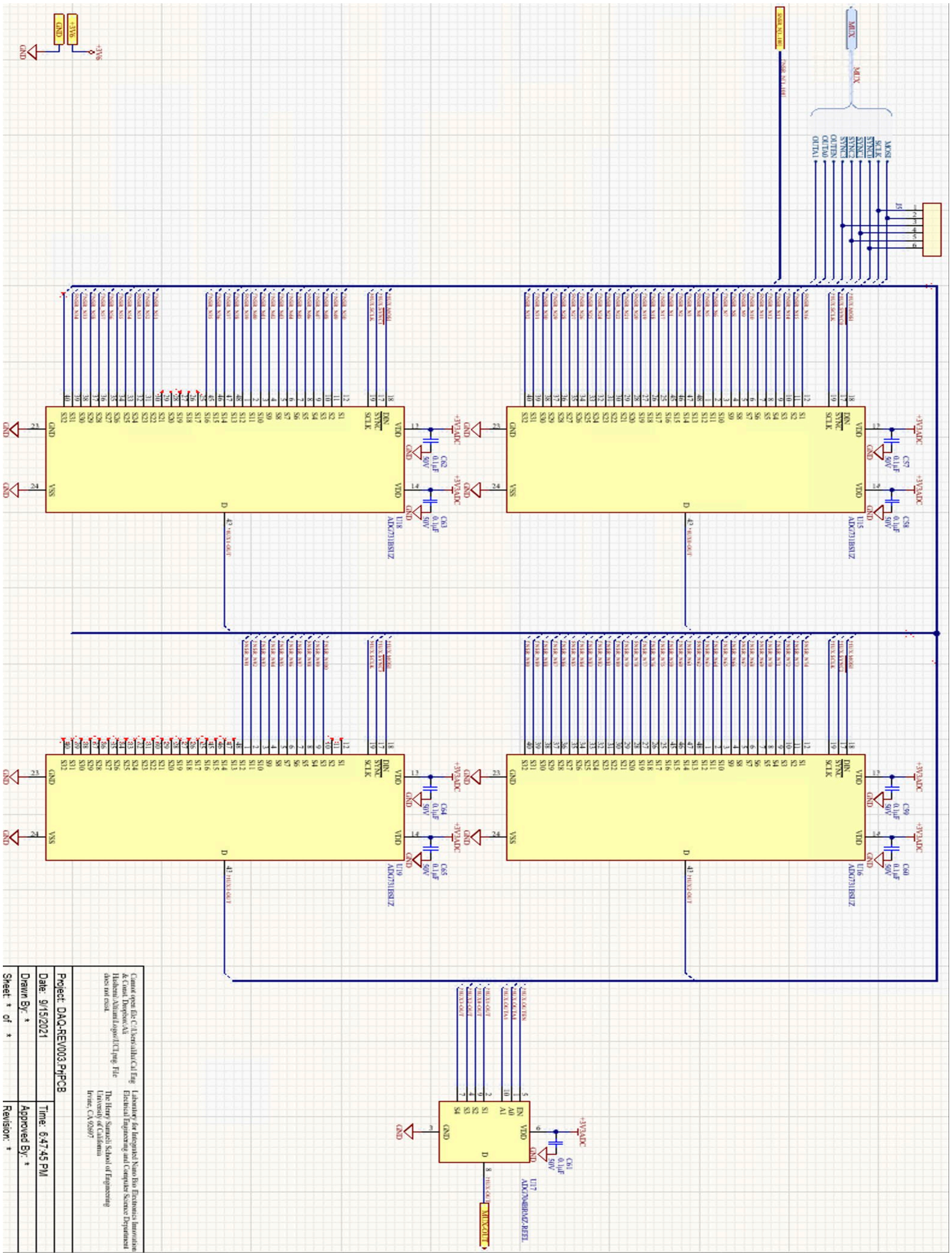


Figure 14: Multiplexer Circuit

As discussed in the power supply section, a standalone LDO is used to power this entire block to avoid digital noise crossover from the digital ICs. Since this block is mixed signal block, the digital signals are all filtered with a ferrite-bead based low pass filter to avoid switching current from going back into the LDO and creating switching noise. This helps keep the resolution of the sampling system as high as possible. Being that this is a sampling system, calibration is also extremely important in getting the best performance out of the system. Another MUX is tied to the input of the system with 4 low drift, high accuracy resistors. The circuit recommended by the manufacturer can be found in figure 14. The manufacturer recommends connecting the unknown impedance through the switch as well, but this isn't feasible for our design. We use a 4 input MUX and have 4 calibration values. Using a bigger MUX would take more room, power, and would provide more noise into the circuitry. Adding another ADG704 isn't feasible for noise reasons, either. While the calibration becomes more accurate when the unknown impedance is also measured through the MUX, it just doesn't make sense overall for this particular design.

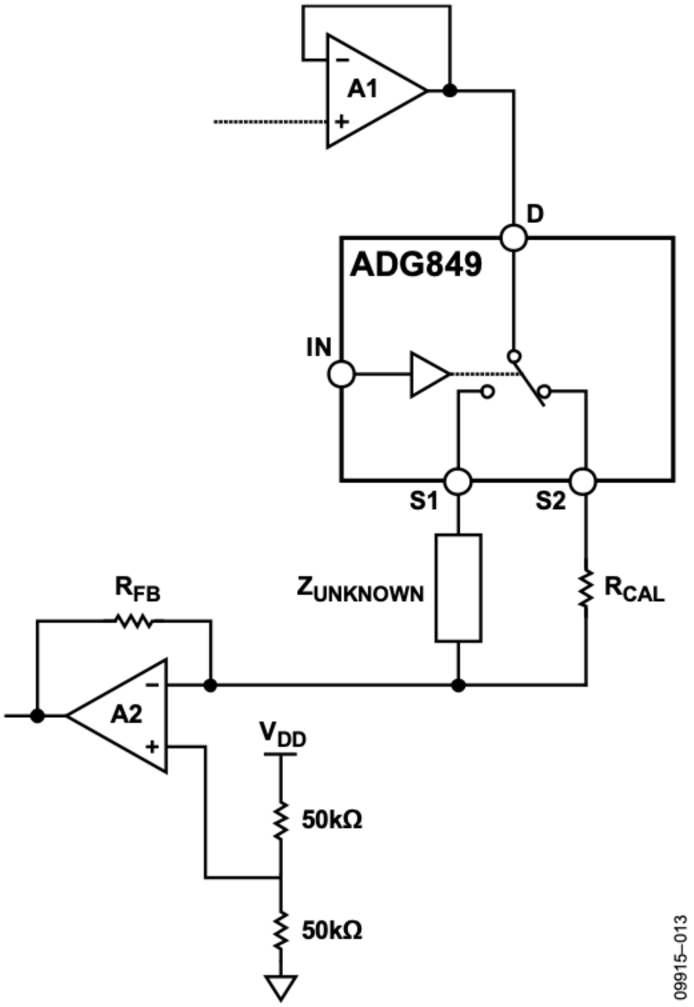


Figure 15: AD5933 Calibration MUX

With this added MUX, the software can auto-calibrate each excitation voltage/gain combination by connecting the corresponding calibration resistor, taking readings, and recording to EEPROM. Since we did not attach the unknown impedance to the MUX, users must ensure that no sensors are connected to the system when calibrating. Equation (10) can be used to calculate the calibration value.

$$Gain\ Factor = \frac{Admittance}{Code} \tag{10}$$

Now we further convert (10) to equation (11) to get impedance.

$$Gain\ Factor = \frac{\frac{1}{impedance}}{Magnitude} \tag{11}$$

Now to back calculate impedance for any conversion, not just for calibration, we use equation (12).

$$Impedance = \frac{1}{Magnitude * Gain\ Factor} \tag{12}$$

Calibration should be done in a temperature/humidity-controlled environment with isolated USB power for best results. The temperature and humidity control allow the device to be calibrated at a happy medium. When calibration isn't done at typical room temperatures, the delta between the calibration and the measured values in worst-case environments becomes big, which causes inaccuracies in the overall sampling. Isolated USB ensures there are no ground loops or computer switching noises coming into the circuitry during calibration and measurement. Calibration resistance values can be calculated using equation 13. Z_{min} and Z_{max} are the minimum and maximum impedances that are required to be measured, respectively. The full schematic for the sampling system can be found in figure 14.

$$R_{cal} = (Z_{min} + Z_{max}) * (\frac{1}{3}) \tag{13}$$

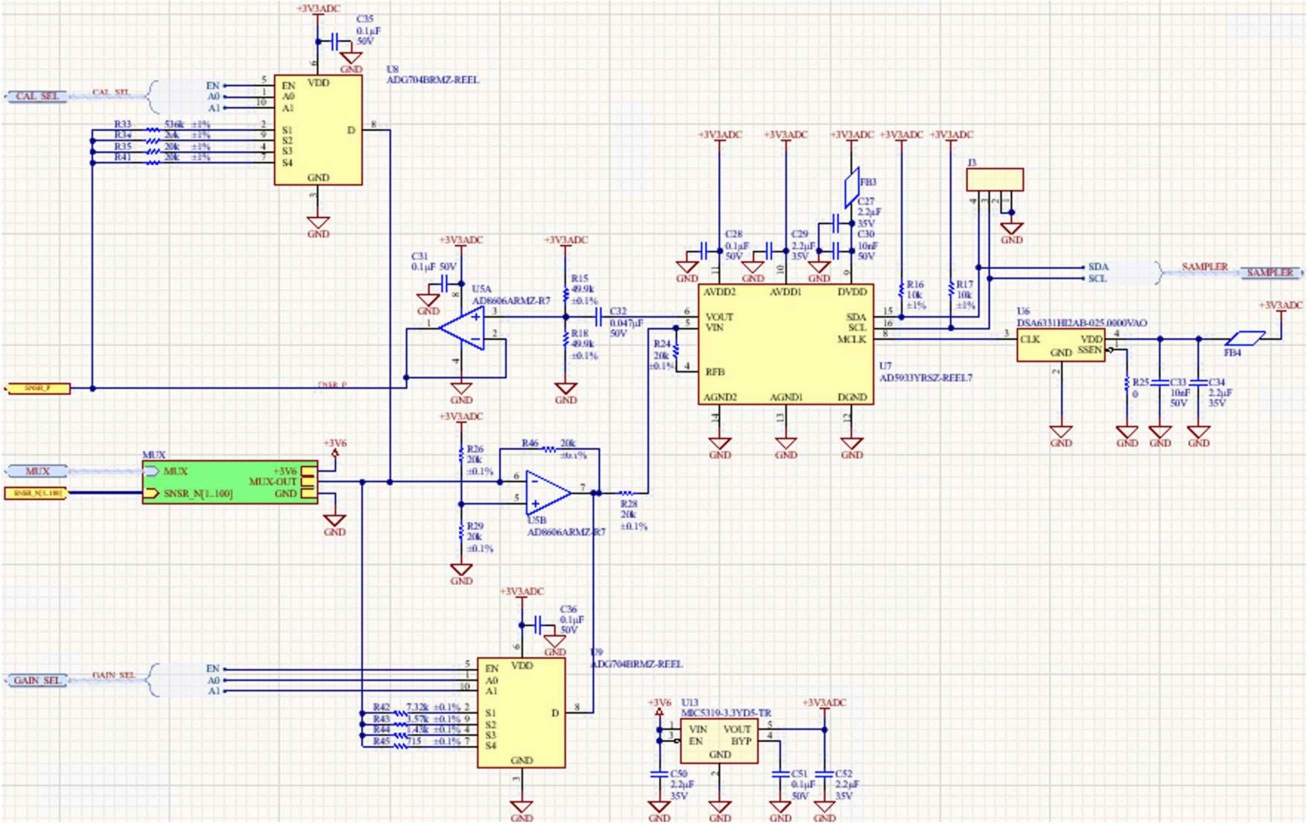


Figure 16: Sampling System

We must note that the gain factor has variations with frequency. A graph of the variations can be found below in figure 16. If we perform a 2-point calibration rather than a single point calibration, we can greatly increase the accuracy of the gain factor which in turn increases the accuracy of the entire system. The graph of the gain factor versus frequency with a 2-point calibration can be found in figure 17.

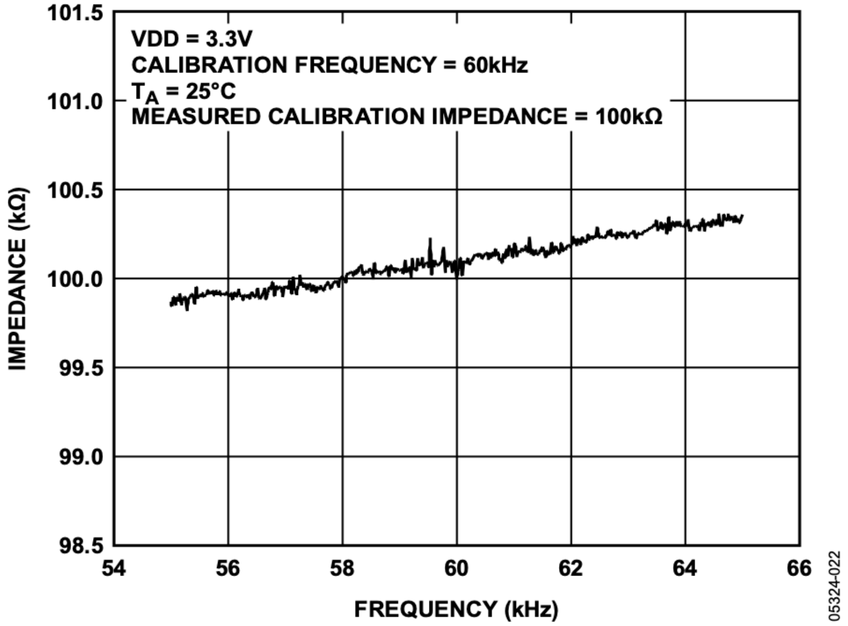


Figure 22. Impedance Profile Using a Single-Point Gain Factor Calculation

Figure 17: Gain Factor for Single Point Calibration

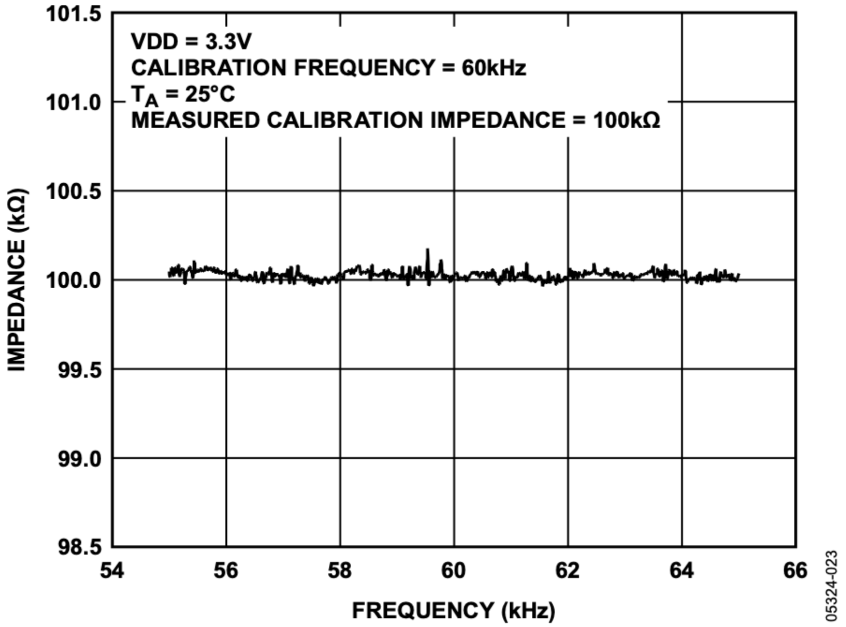


Figure 23. Impedance Profile Using a Two-Point Gain Factor Calculation

Figure 18: Gain Factor Variation for 2-Point Calibration

Printed Circuit Board (PCB)

Special attention must be paid to the PCB when designing a low noise sampling system. The main difficulties of the PCB came from its mixed signal nature and its size. For power, a partial plane was used for the 3.6VDC that went to each of the regulators. The ground plane for the DC-DC and each LDO was placed directly underneath the parts in the layer below the top layer. These ground planes were then anchored via multiple vias to a global ground plane that runs the length and width of the board. This is done to ensure that current meshes are localized but the overall potential of the ground plane is the same throughout the board. Since the multiplexers were spread out, the 3.3VDC of the sampling system also ran the length and width of the board. The mixed signal components were placed away from the purely digital components for better separation. To deal with the large size of the board and to prevent warping and cracking, dead copper (copper regions not connected to a net) was used to balance the board. The 3D model of the board and a physical image of the board can be found in figures 18 and 19 respectively.

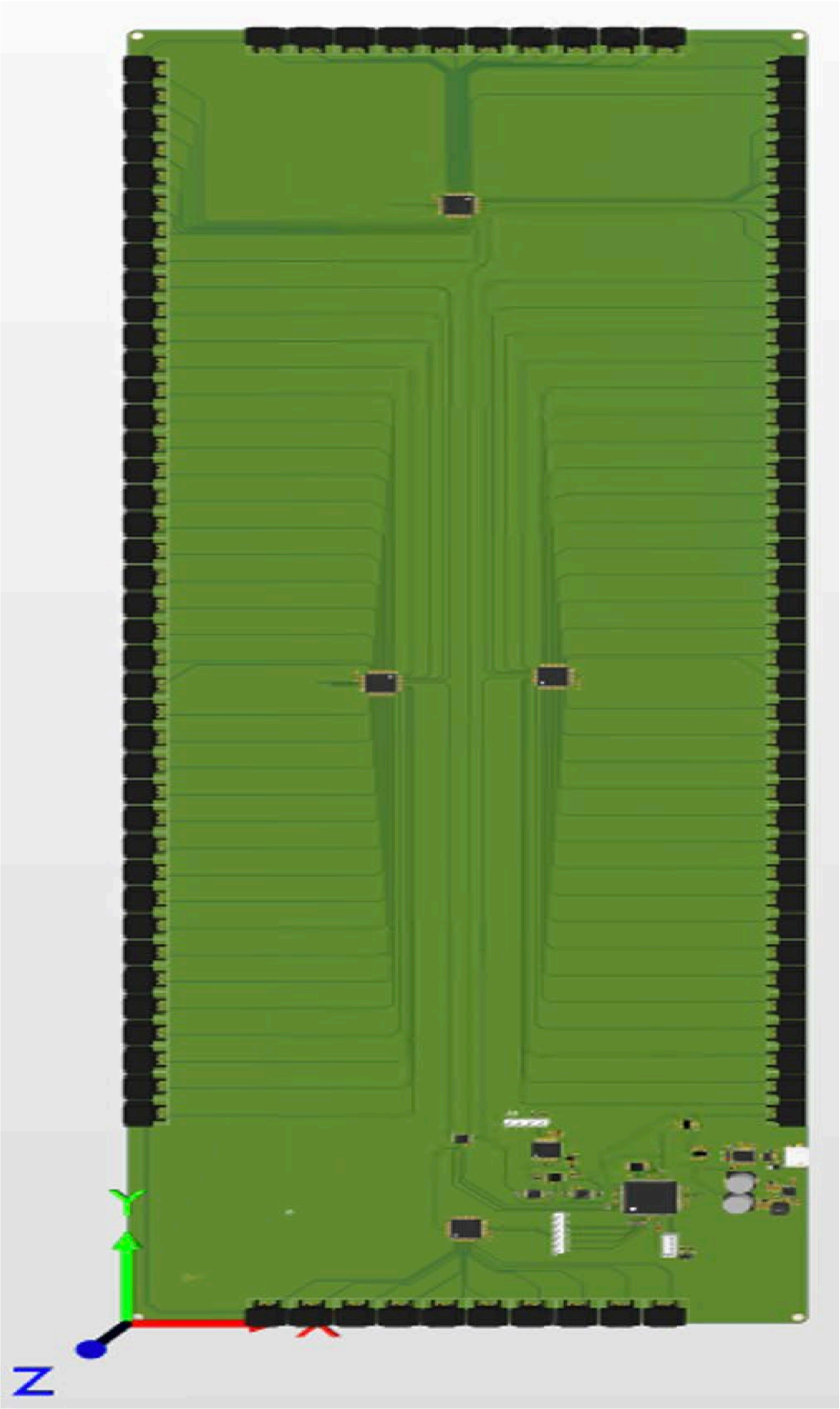


Figure 19: PCB 3D Model

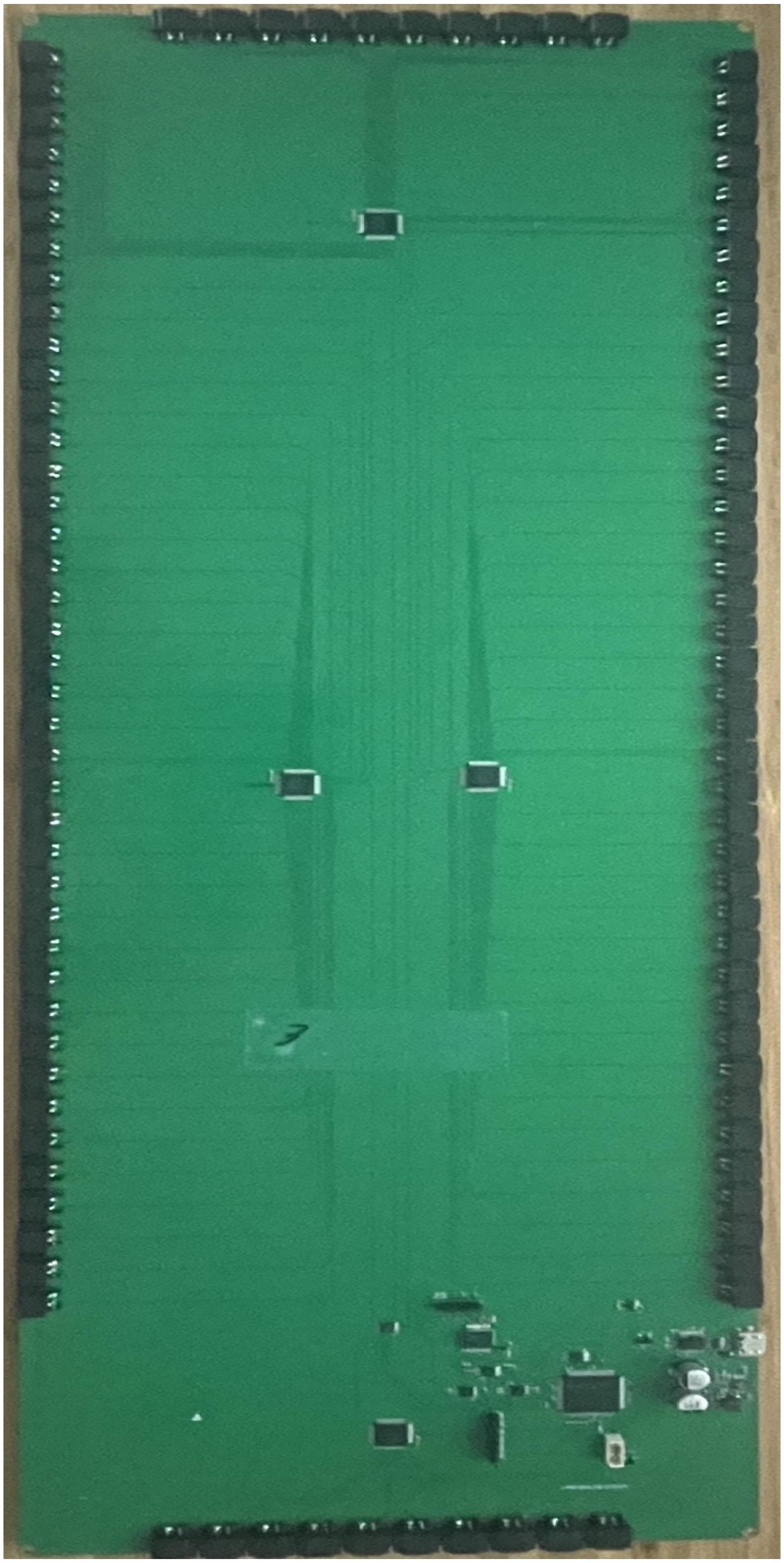


Figure 20: PCB Picture

FIRMWARE DESIGN

The firmware for this device is written entirely in C using the Microchip MP Lab integrated development environment (IDE). Microchip utilizes their own compilers, the XC series that come in 8, 16, and 32-bit. For this project, we use the 32-bit since we have a 32-bit MCU.

Firmware Structure

The firmware for this device is fairly simple. A pin manager file has been created to set the input/output states, initial latch states, and any pull-up/pull-downs. An overall system file is created to call all initialization routines and to configure the MCU and its internal clocks. There are drivers for the EEPROM, sampling system, timers, and the UART. The EEPROM driver utilizes SPI and the sampling system utilizes I2C as previously discussed. The timer is used for handling of a heart-beat LED and the EEPROM. The EEPROM requires certain timing when writing to it to ensure that a write is successful. For that reason, we utilize a timer configured for 10uS interrupts and only service the EEPROM once we have reached a count sufficient for the write to be complete.

Sampling System Algorithm

The sampling system has various firmware blocks that collectively run the algorithm for sampling. The most important block is collection of data from the AD5933. The algorithm for this can be found in figure 20. The device is first programmed for the end-user's desired test parameters. The device is then prepared in standby mode. A start frequency is then sent out to stabilize the attached impedance. After the system has settled, The initial frequency is then sent out. For a frequency sweep, this is the first value in the sweep range, but for a live-plot at fixed frequency, this would be the desired fixed frequency. The device then has a register that needs to be polled to see if the conversion is complete or not. This can be polled as often as desired. We poll this in the firmware as often as we possibly can by running the function that does this in an open-loop nature in our main super loop. Once data is ready, it is read back, packaged, and sent out to the USB bus via UART. If we are in a frequency sweep, we check if the sweep is done. If not, we increment the frequency and repeat. If in live-plot, we increment by 0 and repeat.

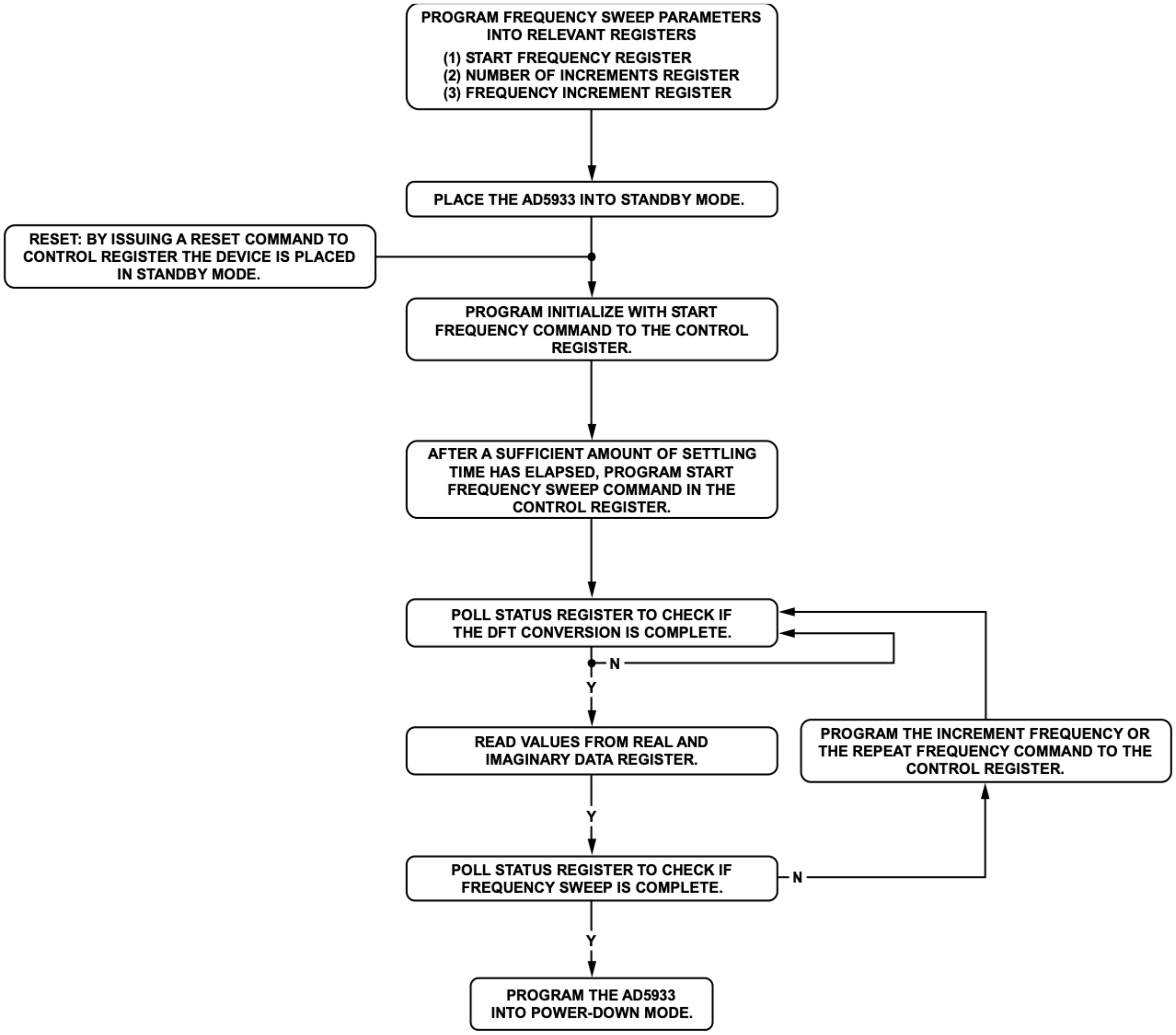

PROGRAM FREQUENCY SWEEP PARAMETERS INTO RELEVANT REGISTERS
(1) START FREQUENCY REGISTER
(2) NUMBER OF INCREMENTS REGISTER
(3) FREQUENCY INCREMENT REGISTER
PLACE THE AD5933 INTO STANDBY MODE.
RESET: BY ISSUING A RESET COMMAND TO CONTROL REGISTER THE DEVICE IS PLACED IN STANDBY MODE.
PROGRAM INITIALIZE WITH START FREQUENCY COMMAND TO THE CONTROL REGISTER.
AFTER A SUFFICIENT AMOUNT OF SETTLING TIME HAS ELAPSED, PROGRAM START FREQUENCY SWEEP COMMAND IN THE CONTROL REGISTER.
POLL STATUS REGISTER TO CHECK IF THE DFT CONVERSION IS COMPLETE.
N
Y
READ VALUES FROM REAL AND IMAGINARY DATA REGISTER.
PROGRAM THE INCREMENT FREQUENCY OR THE REPEAT FREQUENCY COMMAND TO THE CONTROL REGISTER.
Y
POLL STATUS REGISTER TO CHECK IF FREQUENCY SWEEP IS COMPLETE.
N
Y
PROGRAM THE AD5933 INTO POWER-DOWN MODE.

Figure 21: AD5933 Algorithm

USB Communication

The final piece to the puzzle in the firmware is the USB communications. This is done in a very simple manner using a UART module at 115,200 baud. The module has a user interface (UI) file that is custom written that allows us to package data before it is sent out. The device acts entirely as a slave until a test is started. The device will wait for a command and once received it'll process. The package data is configured as a command byte, necessary payload, and a checksum as mentioned previously. When a test is started, to avoid missing any data, the device will stream all data it receives from the sampling system immediately. It has a prefix that allows the user to see what channel the data is corresponding to and a suffix checksum to ensure proper communication.

SOFTWARE DESIGN

The software for this device has been written in Matlab in the interest of time and ease of distribution. Matlab has an easy-to-use graphical user interface (GUI) generator that is easy to configure and customize. The GUI is mainly drag-and-drop but the code attached to each GUI block can be fully customized through Matlab.

Communication Interface and Structure

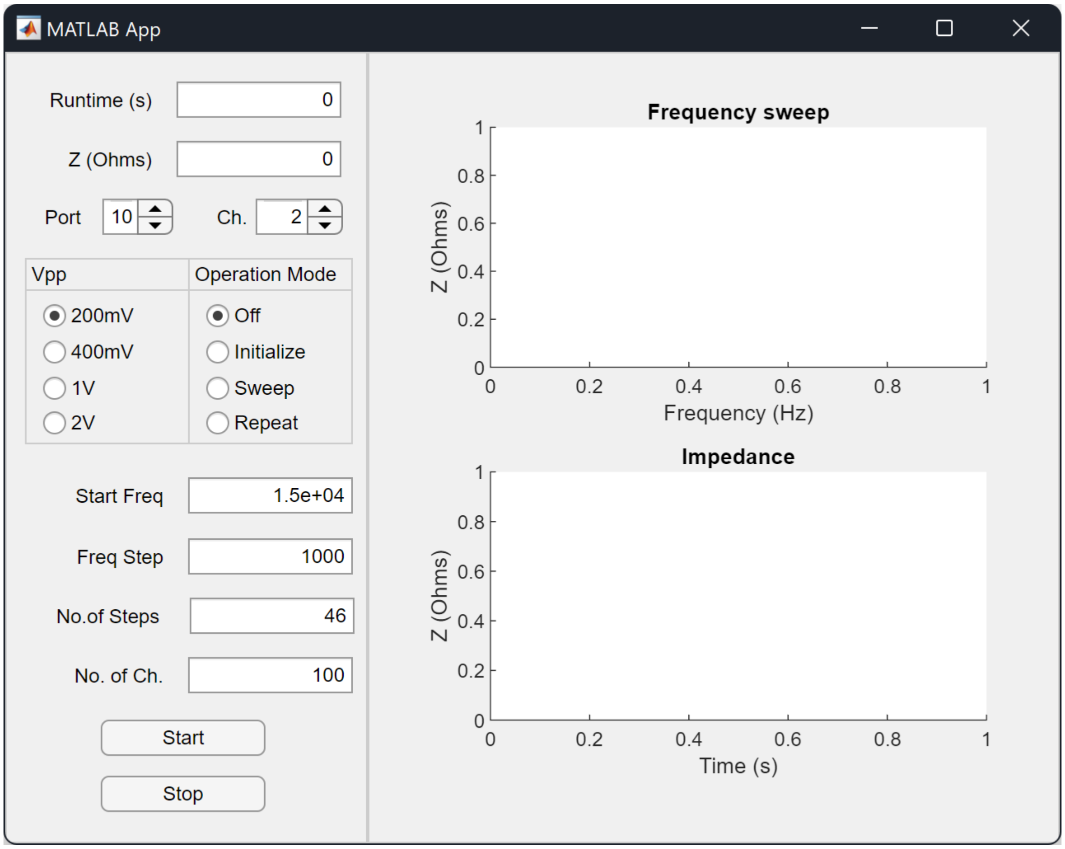
The software communicates to the board via USB. The computer is setup to have a virtual comm-port which enables basic two-way communication through the USB port of the computer down to the UART module of the MCU. A baud rate of 115,200 is chosen for communication and the computer and device can then communicate through a simple interface. The packet structure is fairly simple. The first byte is the command byte, the following bytes are the payload, if any, and the last byte is a checksum. The checksum formula can be found in equation (14).

$$chksm = (b_1 * 1) + (b_2 * 2) + (b_3 * 3) + b_n * n) \tag{14}$$

In (14), n represents the nth byte in the packet. Note that we are only using a single byte for checksum in this algorithm. When the formula goes past the 8-bit limit, the number overflows and so does the checksum. The communication between the computer and the device is fairly rock solid, thus a complicated verification method really isn't necessary. The checksum is just a backup that doesn't hurt.

Capabilities

The software provides the following capabilities: calibration, changing mode between frequency sweep and live plot, setting of voltage, start/end frequency, and number of points when in frequency sweep mode, setting of voltage and frequency when in live-plot mode, number of sensors to sample, and exporting of raw data to an Excel spreadsheet. Settings must be changed before a sweep or live plot are started. When voltage is changed, the firmware of the system automatically changes gain resistors as well. When calibration for the selected voltage is called, the device automatically connects the correct calibration resistor and records the calibration data to EEPROM. The software interface of the system can be found in figure 19. The interface shows a single sensor at a time, thus the "ch" buttons are used to cycle between channels. It should also be noted that the device reads sequentially, thus channels in the middle of a range cannot be disabled.



Runtime (s) 0

Z (Ohms) 0

Port 10 Ch. 2

Vpp
● 200mV
○ 400mV
○ 1V
○ 2V

Operation Mode
● Off
○ Initialize
○ Sweep
○ Repeat

Start Freq 1.5e+04

Freq Step 1000

No.of Steps 46

No. of Ch. 100

Figure 22: Software Screenshot

DEVICE EVALUATION AND SPECIFICATIONS

To test the device, we connected a combination of two electrode sensors printed on copper tape and silicon wafer-based sensors. The copper-based sensors were printed to allow testing of the device when many sensors are connected at the same time so that we can test the time-multiplexing capability. Additionally, because their properties are not as tight and controlled as the silicon-based sensors, they provided a “worst case” testing scenario for the device. A live plot test was performed with both types of sensors connected while dropping various buffers and cleaning the sensors. Frequency sweep tests were performed with the various sensors with a single buffer. The device’s real-time response was also tested by dropping beads on the sensors and ensuring that the device can capture the live change of impedance. To verify results, the same sensors were used to perform the same tests on a Zurich impedance analyzer.

Live Plot

We connected 2 silicon sensors to the board for this test. The sensors were cleaned with DI water in between each sample that was deposited. Multiple DI wash cycles were used to ensure the sensors are thoroughly cleaned after each sample deposit. Tap water, 0.5x PBS, and 1x PBS were used for testing. The exact same test with the same sensors was also done on the Zurich impedance analyzer. Results can be found in figures 22, 23, and 24. We can see that the live plot results are very similar. Slight differences can appear just from how and when the buffers are applied to the sensor as well as how well the sensors are cleaned after each application.

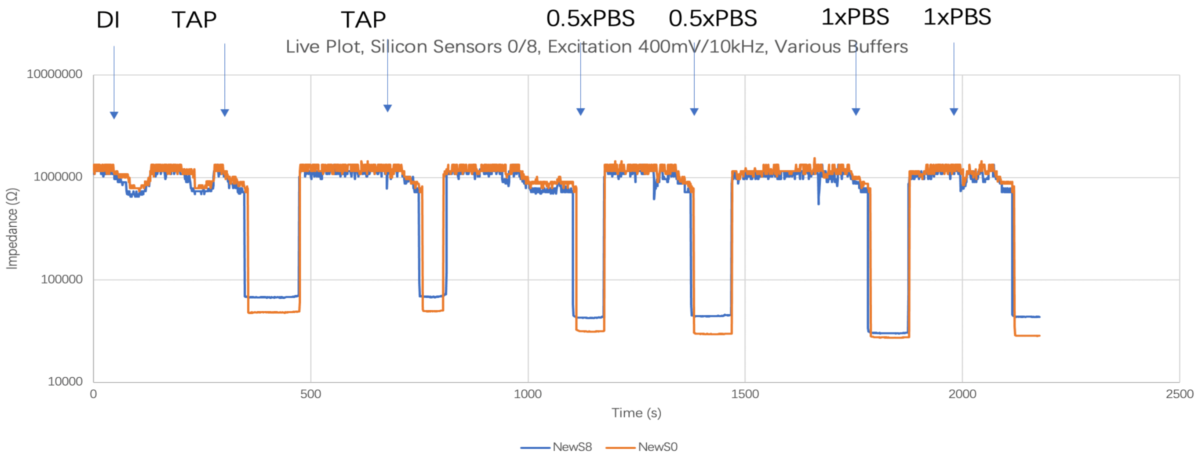


Figure 23: PCB Live Plot Test

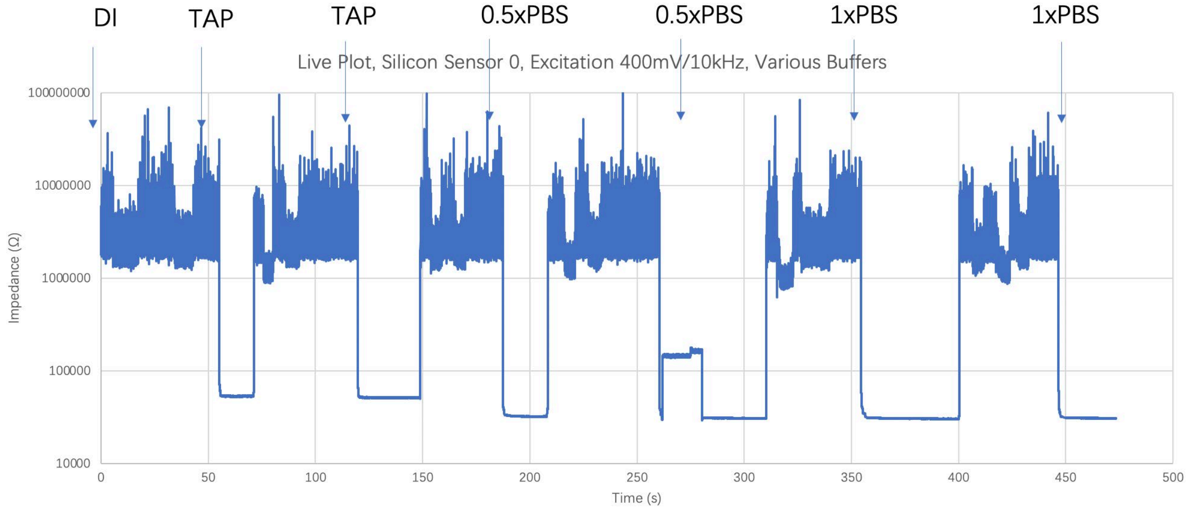


Figure 24: Zurich Live Plot S0

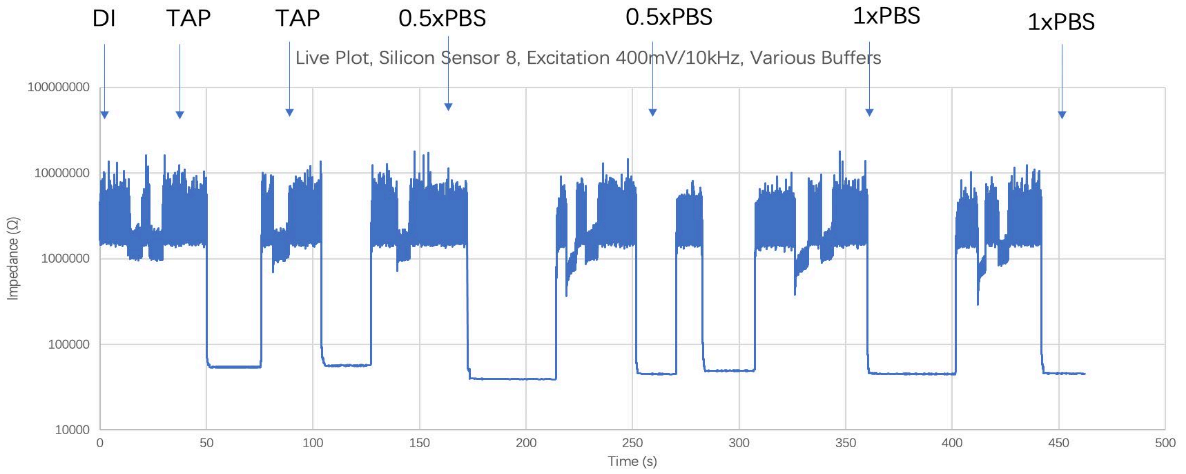


Figure 25: Zurich Live Plot S8

Frequency Sweep

The devices frequency sweep functionality was also tested on the same 2 silicon sensors and compared to the performance of the Zurich. The sensors were thoroughly washed with DI water between tests. 2.9uL of 1x PBS was applied to both sensors and the sweep was performed. Results can be found in figures 25 and 26. We can see bumps in the PCB graph whereas the Zurich is very smooth. This is due to performance at lower frequencies in the PCB not being as good as its performance in higher frequencies. We still see very similar results in frequencies above 6kHz. Overall, the frequency sweeps are fairly similar and within a 0.5% tolerance of one-another. It should be noted that the frequency sweeps across different voltages were also tested and yielded similar results. The copper sensors are capable of withstanding higher voltages but the silicon based sensors could only withstand 400mV which is why they are tested with that excitation value.

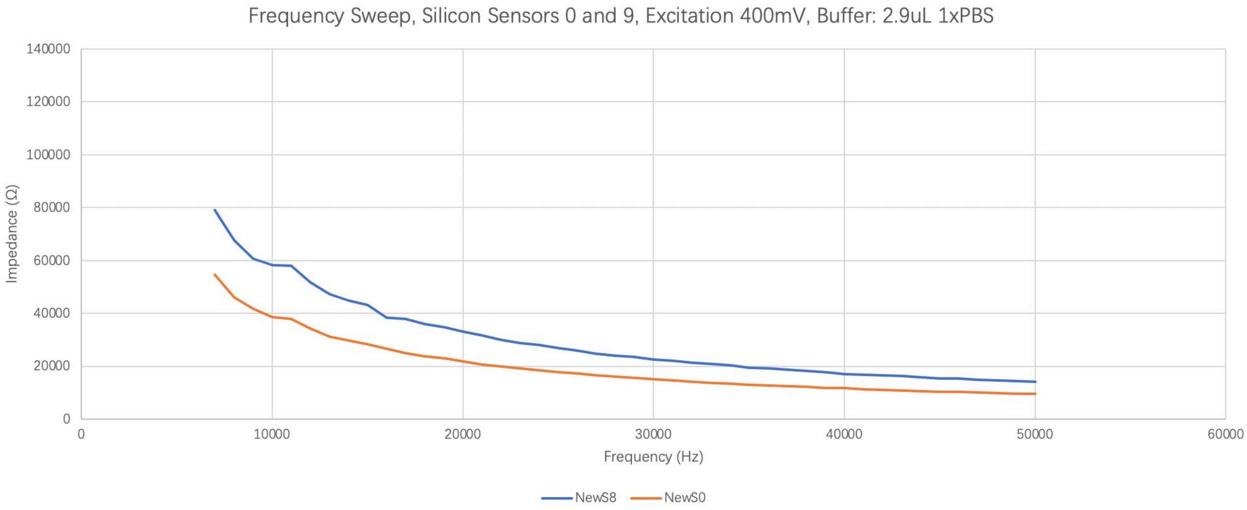


Figure 26: PCB Frequency Sweep

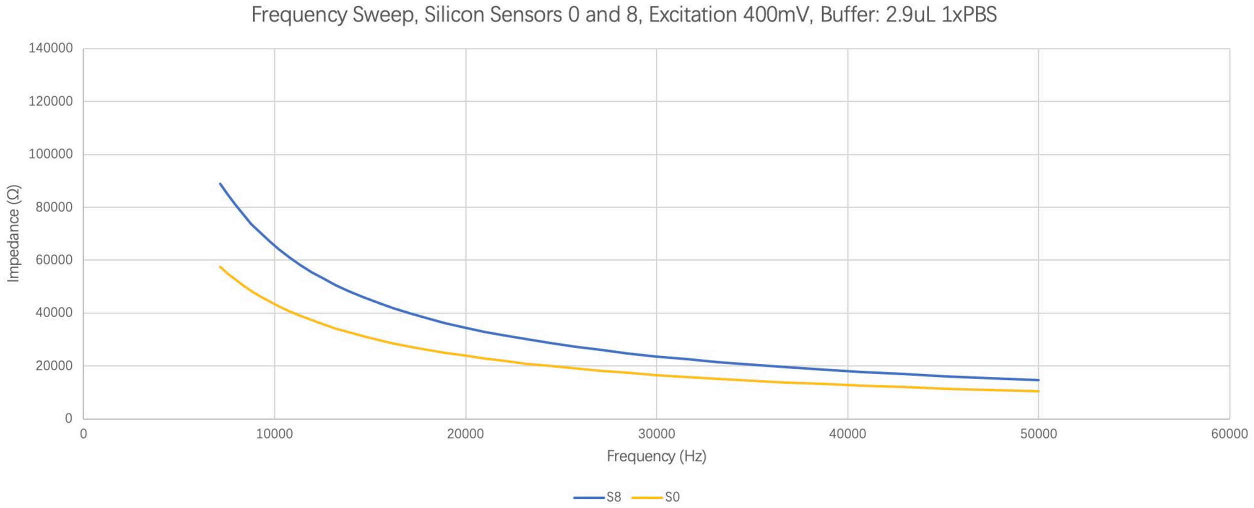


Figure 27: Zurich Frequency Sweep

Real Time Response

To test real time response of the systems, Polystyrene beads were placed on the sensors and the output change was recorded. The same sensors and concentration of beads was used on both the PCB and the Zurich tests. Results can be found in figures 27 and 28 for the PCB and Zurich respectively. Figure 29 shows how the beads are deposited on the sensor. The graphs for the two tests may have different cross points, but this is due to how and when the beads are applied to the sensor. The main goal is to confirm that in both devices, the reaction to the beads is the same. The desired reaction is that the impedance measured starts out the same, then based on the sensor’s impedance curve, we see a gradual increase of impedance and then a flattening of the curve. This is the reaction the beads should have on the sensors based on their properties.

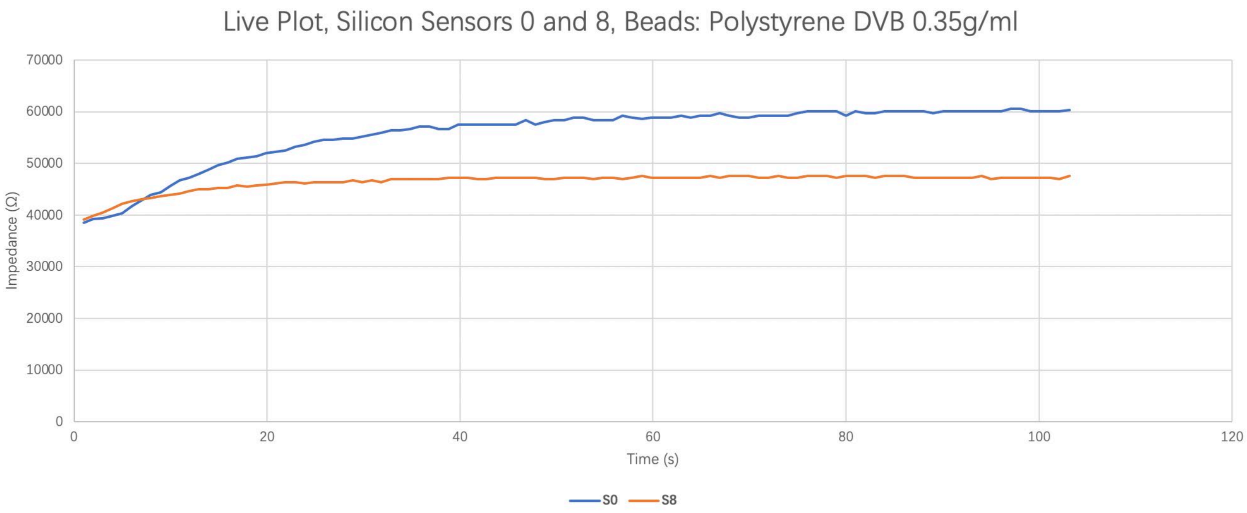


Figure 28: PCB Real Time Response

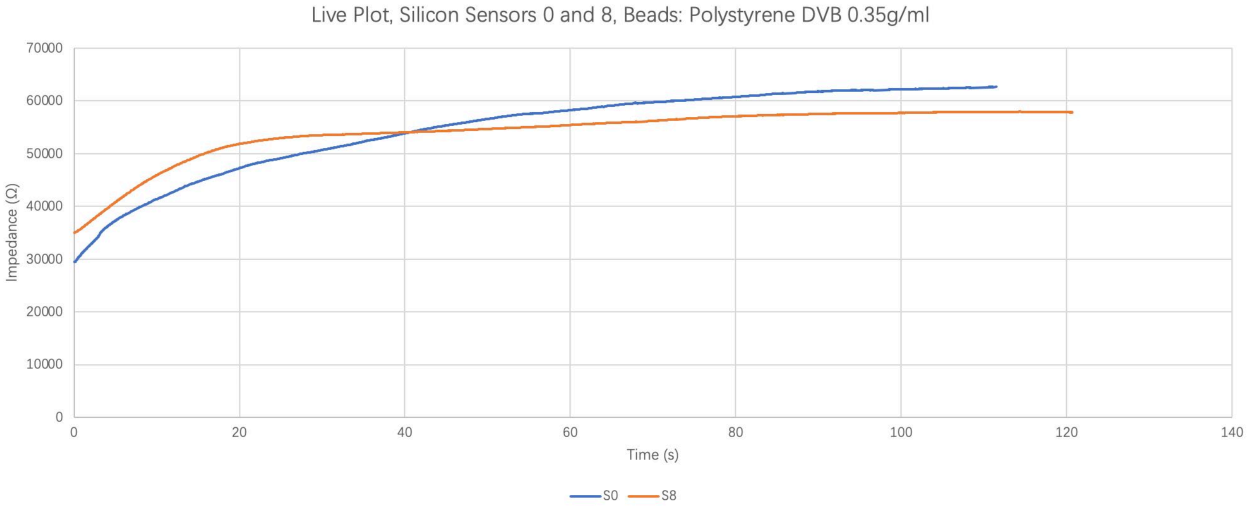


Figure 29: Zurich Real Time Response

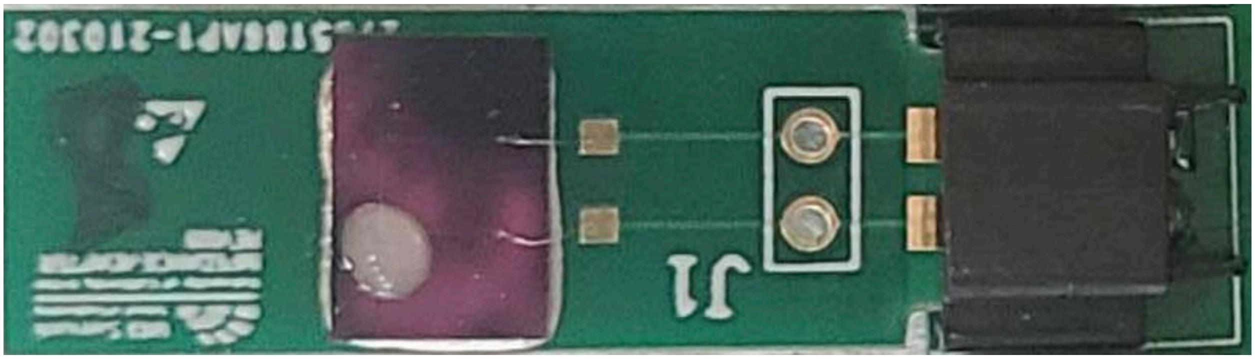


Figure 30: Sensor with Beads

CONCLUSION

In conclusion, we were able to design a USB powered impedance spectroscopy device for easy use with various impedance-based biosensors. The device uses USB for power and communications for user convenience. The system can run frequency sweeps and live plotting 100 sensors at a time. The sensors connect via a custom FFC PCB and are easily pluggable into the main PCB. Frequency is adjustable from 0.1Hz – 100kHz and excitation voltage can be adjusted to 200mV, 400mV, 1V, and 2V peak-to-peak. Results of the system were compared to that of a Zurich impedance analyzer. Although the results are as good as the impedance analyzer, they meet the requirement for most common types of impedance-based biosensors. In this device, ease of use and functionality are traded for the high performance of an impedance analyzer. This device is a proof of concept for a system that can be used in labs for testing of various biosensors.

Works Cited

[1] K. Acord, "Electrochemical Impedance Spectroscopy," 7 September 2021. [Online]. Available: https://eng.libretexts.org/Bookshelves/Materials_Science/Supplemental_Modules_(Materials_Science)/Insulators/Electrochemical_Impedance_Spectroscopy. [Accessed 11 September 2021].

[2] R. Ohno, H. Ohnuki, H. Wang, T. Yokoyama, H. Endo, D. Tsuya and M. Izumi, "Electrochemical impedance spectroscopy biosensor with interdigitated electrode fordetection of human immunoglobulin A," *Biosensors and Bioelectronics,* vol. 40, pp. 422-426, 2013.

[3] M. Cortina, M. Esplanidiu, S. Alegret and M. d. Valle, "Urea impedimetric biosensor based on polymer degradationonto interdigitated electrodes," *Sensors and Actuators,* vol. B, no. 118, pp. 84-89, 2006.

[4] A. Bogomolova, E. Komarova, K. Reber, T. Gerasimov, O. Yavuz, S. Bhatt and M. Aldissi, "Challenges of Electrochemical Impedance Spectroscopy in Protein Biosensing," *Analytical Chemistry,* vol. 81, no. 10, p. 3944–3949, 2009.

[5] D. D. Macdonald, "Reflections on the history of electrochemicalimpedance spectroscopy," *Electrochimica Acta,* vol. 51, pp. 1376-1388, 2006.

[6] Texas Instruments, "Transimpedance amplifier circuit," January 2019. [Online]. Available: https://www.ti.com/lit/an/sboa268a/sboa268a.pdf. [Accessed 10 September 2021].

[7] Analog Devices, "AD5933 Datasheet," 2017. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD5933.pdf. [Accessed 09 September 2019].

[8] R. Keim , "Clean Power for Every IC, Part 3: Understanding Ferrite Beads," All About Circuits, 30 September 2015. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/clean-power-for-every-ic-part-3-understanding-ferrite-beads/. [Accessed 14 September 2021].

[9] Microchip, "MIC5319 Datasheet," 2018. [Online]. Available: https://www.microchip.com/en-us/product/MIC5319. [Accessed 14 September 2021].

[10] Analog Devices, "AD5933 Circuit Note CN-0217," 2013. [Online]. Available: https://www.analog.com/media/en/reference-design-documentation/reference-designs/CN0217.pdf. [Accessed 08 09 2021].