ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Musical Hand

Team #24

RAMSEY FOOTE
(rgfoote2@illinois.edu)
MICHELLE ZHANG
(mz32@illinois.edu)
THOMAS MACDONALD
(tcm5@illinois.edu)

TA: Hanyin Shao

May 4, 2022

Abstract

This final report explains the Musical Hand: its purpose, functionality, design, and fabrication. It details the process from start to finish of how this portable, wearable synthesizer went from an idea to a fully realized final product demonstrated for ECE 445 senior design laboratory. All design considerations, figures, motivations, and details are included to assist the reader in understanding the engineering behind the instrument.

Contents

1 Introduction

1.1 Purpose

Musical instruments come in all shapes and sizes; however, transporting instruments often involves bulky and heavy cases. Not only can transporting instruments be a hassle, but the initial purchase and maintenance of an instrument can be very expensive. For example, let us consider stringed instruments. The initial purchase can easily reach 3-4 digits while strings and bows often need replacements or repairs. We addressed these issues of cost and portability by creating an instrument using electronic synthesis that is lightweight, compact, durable, and low maintenance.

1.2 Functionality

Our project involves a wearable system consisting of two gloves and a chest mount. The left glove is used to dictate the pitches of three “strings” using relative angles between the palm and fingers. For example, from a flat horizontal hand a small dip in one finger is associated with a low frequency. A greater dip corresponds to a higher frequency pitch. The roll of the right glove controls the vibrato effect added to the generated sound, and the pitch of the right hand acts selects the finger that is outputting sound. For sensors, both gloves utilize accelerometers instead of flex sensors to limit wear and tear, which helps limit the cost of maintenance typical of more physical synthesis methods. The processing, power, and audio modules are contained within a chest mount worn with strap over both shoulders, which provides a highly portable product.

1.2.1 Subsystem Overview

The overall system, as shown in Figure 1, consists of three main subsystems: the note hand, the effects hand, and the central subsystems. The note hand subsystem corresponds to the left hand glove that controls which notes are generated. It contains four analog accelerometers: one on each of the index, middle and pointer fingers and one on the back of the hand. The effects hand subsystem correspond to the right hand glove that determines vibrato and finger select. This subsystem contains a single analog accelerometer on the back of hand. The final subsystem is the central unit that houses the power, audio, and microcontroller modules. The central subsystem processes all of the sensor values and generates an output audio signal while powering the entire system.

2

Figure 1: Block Diagram

2 Design

2.1 Note Hand

2.1.1 Design Procedure

The note hand was designed with durability in mind. Breaking free from the physical constraints of sensors that rely on flexibility or physical contact, the design choice of using accelerometers was made. For the case of the musical hand, these accelerometers rely largely on gravitational force as a reference metric.

Two types of printed circuit boards (PCBs) were designed for the note hand with the purpose of acquiring and transmitting analog accelerometer data. The finger boards were designed with the constraint of size in mind, as they are required to fit on the back of a finger tip without encumbering the user. The back of the hand board did not have as tight of a space requirement, and therefore was determined to be useful for consolidating data lines.

Another consideration for the note hand's design was making it wearable and not cumbersome. To achieve this, the PCBs and connections were mounted onto a glove with Velcro and nylon straps sown onto the design to keep cables from interfering with movement. Finally, to allow the full extension of the arm when this glove is worn, the cabling was sized longer than minimally required and strapped to the arm with elastic bands.

2.1.2 Design Detail

Analog Device's ADXL335 accelerometer was determined to be the best fit for this design for a few reasons. Firstly, its small, low power design was perfect for a portable, battery powered product such as the musical hand. The footprint is 4mm x 4mm and it typically draws only 350A . Secondly, it runs on the same 3.3V logic that the microcontrollers do, creating ease of integration. Finally, this accelerometer is sensitive enough at a typical rating of 300mV/g to provide the resolution needed for microchip ADCs to encode data[1].

In the PCB design, the accelerometers were paired with the necessary 47 F capacitors to provide a bandwidth of 100Hz. This was chosen since the human response to touch stimuli is on the order of 100ms, so a 100Hz bandwidth will make it so the user will not be able to notice any lag due to accelerometer delay, while also not being so fast as to be susceptible to high frequency noise. Each accelerometer requires power and ground and outputs an analog signal for x, y, and z axis. For this reason each finger PCB uses a 5-pin header with the appropriate cabling attaching them to the back of hand PCB, which acts as a hub. The 4 combined accelerometers' signals as well as power and ground motivate a 20 pin header that connects the note hand to the central subsystem[1].

2.1.3 Verification

Verifying the note hand after soldering the PCBs required modular testing. Each individual PCB was powered by a benchtop DC power supply and the 3 accelerometer axes were probed by oscilloscopes. For the design to work, an analog signal had to be generated. Such a signal is demonstrated in Figure 2 with sufficient difference between the lowest and highest voltages, corresponding to an unbent finger parallel to the floor and a 90 bent finger perpendicular to the floor respectively. Figures 3 and 4 demonstrate this difference, calculated to be roughly 300mV, which is consistent with the data sheet's rated sensitivity to a 1g acceleration difference [1]. More information on the requirements and verification of the note hand can be found in Appendix A Table 4.

Figure 2: An analog waveform created by moving a finger accelerometer along the z-axis

Figure 3: The output of a finger accelerometer taken parallel to the floor

Figure 4: The output of a finger accelerometer taken perpendicular to the floor

2.2 Effects Hand

2.2.1 Design Procedure

For the Effect Hand, durability was yet again key. However, this subsystem only needed one accelerometer and had more space. We decided to use an additional finger module from the Note Hand in order to reduce on number of total boards we needed.

A 5-pin header is needed to interface with the central unit, so the same MTA100 connectors used in the Note Hand could be used in order to keep a the same standard across all 5-pin connections.

2.2.2 Design Detail

Since we were using the same board as used on the Note Hand finger tip and did not need any special functionality of the accelerometer, we were able to use the same components as on the finger modules of the note hand. The bandwidth considerations made previously still held true here, so the 47 μF capacitors were used for the bandwidth setting.

2.2.3 Verification

Verifying the effect hand after soldering the PCB was done in a similar manner to the finger boards of the Note Hand. The PCB was powered by a benchtop DC power supply and the 3 accelerometer axes were probed by oscilloscopes. For the design to work, an analog signal had to be generated. Such a signal is demonstrated in Figure 5 with sufficient difference between the lowest and highest voltages, corresponding to a hand with the thumb pointing down and up respectively. Figures 7 and 6 demonstrate this difference, calculated to be roughly 700mV, which is consistent with the data sheet's rated sensitivity to a 2g acceleration difference. [1]. More information on the requirements and verification of the Effect Hand can be found in Appendix A Table 5.

Figure 5: An analog waveform created by moving the Effect Hand accelerometer along the X-axis

Figure 6: The output of the Effect Hand accelerometer taken with the thumb pointing up

Figure 7: The output of the Effect Hand accelerometer taken with the thumb pointing down

2.3 Power Module

2.3.1 Design Procedure

The different components of the design need different voltages to power them. However, the devices that draw the most power need 3.3V to power the system, and the rest of the design needs more than 6V. To account for this, we decided that a 3.3V linear regulator that can have an input of 9V would be ideal to be able to provide the necessary 3.3V to the bulk of the components while also powering the parts that need a higher voltage.

2.3.2 Design Detail

From the datasheets for the microchips[2] and accelerometers[1], the maximum current draw was determined to be 520mA. To ensure no errors, a linear regulator with a maximum output current of at least 1A was needed. With this requirement along with the 9V input and 3.3V output requirements in mind, the LT1963 linear regulator was decided to be a good fit. Once this component was chosen, the necessary stability capacitors were placed according to it's datasheet[3].

2.3.3 Verification

The procedure to verify correct operation of the power subsystem involves using a voltmeter to check for the appropriate output voltages of the battery and linear regulator while under load conditions. Figures 8 and 9 demonstrate that the battery and regulator are operating correctly, respectively. Another important verification is that the linear regulator does not operate at extreme temperatures. Figure 10 uses a thermal camera to demonstrate that under operating conditions, the regulator operates safely. More information regarding the requirements and verification of the power subsystem can be found in Appendix A Table A

Figure 8: The voltage output of the 9V battery under load conditions

Figure 9: The voltage output of the linear regulator under load conditions

Figure 10: Thermal camera demonstrates that the regulator operates at safe temperature

2.4 Audio Module

2.4.1 Design Procedure

With the output of the microcontrollers determined to be a PDM signal, a circuit to filter out the switching was needed to extract the audio waveform. To do this, a second order low pass filter was determined to be the best method of filtering as it allows for more attenuation of switching noise when compared to a single order filter while keeping layout size small.

Other considerations needed were the type of components and the supply voltage needed for the circuit. Since this module was the most likely to need changing of parts, through hole components were used for easier de-soldering if needed. In addition, the system needed to be powered by more than 6V in order to avoid clipping. To achieve this, we decided to use a 9V battery due to its small form factor, capacity, and portability.

2.4.2 Design Detail

The multiple feedback filter used is similar to that described by this document by TI[4] which has a cutoff frequency of 10kHz. The PDM output will be switching at about 500kHz and there wont be audio above 2kHz, so this 10kHz cutoff means that there can be significant error in the cutoff before there is any worry about cutting off audio or letting through switching. In addition, this cutoff means that the switching will be attenuated by nearly 60dB, which will make the switching noise negligible.

The 9V battery was used to have no clipping in the filtering process. If 4.5V source were to be used, the 3.3V PDM input would have some clipping at the filtered output due to it getting too close to the power or ground rails of the op-amp.

2.4.3 Verification

To verify the functionality of the filter module, the output needed to be shown to have negligible switching noise while not exceeding operating temperature range. Originally, the plan was to do extensive testing of the filter on a breadboard to verify the cutoff and fall-off of the filter. However, it became clear that this method was unnecessarily complex. Instead, the output was probed and inspected to ensure that the switching noise was eliminated. While this made the exact cutoff and fall off unknown, those parameters had a large range of error before there was any noticeable effect. Figure 11 shows that the filtered output is a sine wave and that the switching effectively removed. In addition, figure 10 shows that the temperature of the op-amp remains in nominal operating temperatures.

Figure 11: Filtered audio before AC coupling

2.5 Microcontroller Module

2.5.1 Design Procedure

The central subsystem utilizes two microcontrollers, whose general structure is shown in Figure 12. They are interconnected by eight parallel data lines and two acknowledge line. The microcontrollers are used to process input from the Effects and Note Hand subsystems and to output a PDM signal to the Audio Module. Microcontroller 1 (MC1) is dedicated to processing all of the accelerometer inputs and performing any necessary calculations while Microcontroller 2 (MC2) is dedicated to continuously generating a PDM signal. This separation in roles allows the audio output signal to update at a faster rate and avoid latency due to processing sensor values.

2.5.2 Design Detail

The microcontrollers needed a lot of thought put into how they were used in the project. The first major detail is the use of two microcontrollers. The human response time is on the order of 100ms, and given that the microcontrollers have internal clocks on the order of MHz, there would need to be significant delay before the user would notice it. However, audio needs a faster switching in order to get clean filtering and high bandwidth. Offloading the math to one microcontroller while the other one does very fast math for the wavetable synthesis allows MC2 to go at audio speeds while MC1 can be running

Figure 12: Microcontroller Structure

slower due to heavy computation.

The next detail is the physical layout of the controllers. With the parallel interface for sending frequency information between the two microcontrollers, we need to find an orientation that allows for the cleanest pin connections while keeping distance short so the overall board size doesn't get too large. To do this, we put the microcontrollers diagonal from each other. With 8 bits of data per finger and 3 total fingers, we had 24 total I/O pins for frequency data. Since the microcontrollers are 64 pin devices, a diagonal orientation allows for two sides to be connected with short and neat traces.

Finally, we needed to create all of the supporting circuitry for the microcontrollers. Each microcontroller needed .1 F capacitors between power and ground pins for stability, programming headers were needed to allow us to interface with the controller, and some pins needed pull up/down circuits to ensure proper functionality.

2.5.3 Verification

The verification for this part of the design is not something that is easily measured. Since most of things that are probe-able are indication of software functionality, the only real verification is that we can program the microcontrollers and get the correct outputs from them. Figures 13, 14, and 15 show that we can get correct frequency data and PDM data from MC1 and MC2 respectively.

2.6 Software

2.6.1 Design Procedure

The logic required by the project lies within the software design for the system. As discussed in Section 2.6.1, MC1 is programmed to use the accelerometer values to calculate the frequency of the output signal. To account for potential shaking in a user's fingers, MC1 maps input voltages to discrete notes and utilizes hysteresis to stabilize the output note. The determined frequency is encoded as an index value and transmitted to MC2. The index corresponds to a ratio look up table in MC2's program that contain values for two parameters which control the frequency of the sine wave used in the PDM algorithm. Alternatives such as using multiple sine wave tables and directly including octave and frequency information in transmitted messages were considered. However, the index and ratio table approach provided a more compact method of information encoding and more flexibility in terms of frequency determination.

In consideration of synchronization, two acknowledge lines between MC1 and MC2 are used in a handshake protocol when transmitting and receiving messages. This ensures that MC2 does not use an incomplete message to index into the ratio lookup table, which would result in an incorrect output frequency.

2.6.2 Design Detail

Microchip's MPLAB X IDE along with Harmony v3 acting as a hardware abstraction layer were used to develop the software for both microcontrollers. The following section will discuss the software implementations for MC1, MC2, and the communication protocol. Refer to Appendix A for high level flowcharts of the MC1 and MC2 programs.

MC1 contains three lookup arrays, one for each note finger, that contain the corresponding value to be transmitted to MC2 for each discrete note within a finger's range. These arrays are used in combination with sensor input to determine the value transmitted to MC2. To take in accelerometer input, MC1 polls the ADC's connected to the note hand's z axes and the effect hand's x and y axes. The relative difference in voltage values between the note hand's back of hand accelerometer and each of the note hand's finger accelerometers are used to calculate an index into the corresponding finger's lookup array as follows:

$$\text{arrayIndex} = (\text{uint8\_t})(\text{abs}((\text{int})((\text{backOfHandV} - \text{fingerV} + \text{offset})*\text{scalar}))) \qquad (1)$$

The offset and scalar are chosen for each finger such that arrayIndex ranges from 0 to the last index of the corresponding lookup array for that finger. Array indices are calculated for each finger (i.e. pointer, middle, index). The above calculation is reversed to calculate the upper and lower bound voltages for the previous array index values of each finger. These bounds are used to implement hysteresis. In order for a finger's array index to be updated, the finger's relative voltage to the back of hand must be outside of the bounds plus an additional margin.

The x axis of the effect hand accelerometer determines the extent of the vibrato effect by controlling the value of a vibrato threshold. MC1 keeps track of a vibrato index that constantly oscillates between the negative of the vibrato threshold and vibrato threshold. A single vibrato index is calculated for each iteration instead of three because it is independent of the note hand and depends on the effects hand.

Once all three finger's array indices are updated and the vibrato index is determined, the y axis of the effect's hand is used to select the output finger. The following message is then sent to MC2:

$$\text{uint8\_t tx = finger\_lookup\_array[finger\_array\_index] + vibrato\_index} \tag{2}$$

The index that is transmitted by MC1 is used by MC2 to index into a ratio lookup table. The ratio lookup table contains pairs of values for two parameters used in the PDM algorithm: counter and increment. These two parameters are used to control the output frequency as shown in Equation 3

$$\text{desired\_freq = fundamental\_freq * increment / count} \tag{3}$$

By choosing appropriate increment and count values, a single sine wave table can be used to create many different output frequencies. The fundamental frequency is the output frequency when increment and count both have have a value of one. However, empirical results showed that the fundamental frequency tended to drift with different desired frequencies. To counter this drift, an estimation of the fundamental frequency was utilized to calculate approximate ratios in addition to manual tuning for finer accuracy.

The count and increment values, as specified by the index transmitted by MC1, are substituted into Algorithm 1 to generate a PDM signal.

Algorithm 1 PDM algorithm

```
procedure GENERATEPDM
    while true do
        while counter < count do
            accum    accum + waveTable[sinIndex] - out    16
            out    (accum    16)&0x1
            if out == 1 then
                toggle output high
            else
                toggle output low
            counter++
        sinIndex    sinIndex + increment modulo waveTable size
        counter    0
```

Another component of the software design is the handshake protocol for sending and receiving messages between the two microcontrollers. The handshake process relies on the two acknowledge lines. Each acknowledge lines relays the status of a single microcontroller. When MC1 ACK is low, it indicates that no message has been sent, and when it is high it indicates that MC1 has sent a message. As for MC2, a low ACK signals that MC2 is ready to receive a new message, and a high ACK signals that MC2 is not ready to receive. The handshake protocol is illustrated in Algorithm 2.

Algorithm 2 Handshake Protocol

```
procedure HANDSHAKE
    MC1 updates data pins with new value if M2 status is 0
    MC1 sets status to 1
    M2 sets status to 1
    M1 resets status to 0
    M2 reads input message and updates appropriate variables
    M2 resets status to 0
```

2.6.3 Verification

For the project to be successful, different finger positions must result in different output frequencies; therefore a change in accelerometer values. must correspond to different data line values between MC1 and MC2. In order to verify this requirement to data lines were probed at two different finger positions. Figure 13 shows the second and fourth bits of the transmitted message are set high for the finger position corresponding to ratio table index 26. Figure 14 shows the values of the same data lines for a different finger position. In this case, both the second and fourth bits are zero. By demonstrating changing data line values with changing input, MC1 is successfully verified.

Figure 13: Second and fourth data lines for index 26

In regards to MC2, the requirements and verification concern its output. Since MC2 feeds into the Audio Module, it must output a PDM signal. To verify that MC2 outputs a valid

Figure 14: Second and fourth data lines for index 32

PDM signal, an oscilloscope was used to probe the output of MC2. Figure 15 shows the results and verifies that the output is a PDM signal.

Figure 15: Probed output of MC2

3 Costs

Accounting for labor costs requires an analysis of the average starting salary for UIUC ECE graduates as well as a more project-specific positional average across relevant companies. To calculate an accurate figure for this, the differing engineering disciplines of creation need to be considered. Overall, the finished product is a fully contained audio synthesizer. Moog is a relevant company to this style of electrical product, so the starting salary average of an electrical product engineer has been included. The DSP coding requirements of our microcontroller are important for synthesis so the starting salary of an Analog Devices DSP engineer is also included. The analog nature of the accelerometers and audio output used needs to be integrated into the overall design, explaining the inclusion of a Texas Instruments Analog Applications Engineer. Finally, the starting salary of an audio software designer for Shure is used. To begin calculating overall labor cost, the aforementioned salaries were averaged and divided by 2080 hours (the amount of hours worked in a year based off of 40 hour work weeks and 52 weeks/year) for a $/Hour figure. Then, the calculated rate was multiplied by 2.5 and 132 hours (based on 12 hour work weeks and 11 weeks of actual product design and assembly in ECE 445). This yielded a project salary figure for each student in the group, which was multiplied by 3 students for a total labor cost of $42,149.09. All relevant calculations are shown below in Table 3.

The Musical Hand's component cost calculation is shown below in Table 3, with a total cost of $191.28. All capacitor and resistor prices were based off of bulk pricing per unit, as this product could be mass produced for the market.

The total cost for the Musical Hand's research and development as well as material cost is given as:

$$\$42,149.09 + \$191.28 = \$42,340.37 \tag{4}$$

Company	Position	Starting Salary	$/Hr	Salary/Person	Total Labor
Moog	Electrical Product Engineer	$76,610.00	$36.83	$12,154.47	$36,463.41
Analog Devices	DSP Engineer	$109,807.00	$52.79	$17,421.30	$52,263.91
Texas Instruments	Analog Applications Engineer	$85,188.00	$40.96	$13,515.40	$40,546.21
Shure	Software Engineer	$83,023.00	$39.91	$13,171.92	$39,515.75
Grainger College of Engineering	Electrical Engineer	$79,714.00	$38.32	$12,646.93	$37,940.80
Grainger College of Engineering	Computer Engineer	$96,992.00	$46.63	$15,388.15	$46,164.46
Average	Musical Hand Engineer	$88,555.67	$42.57	$14,049.70	$42,149.09

Table 1: Salaries of positions relevant to Musical Hand project

Part Name	# of Units	Cost/Unit	Total Cost
Dual OP Amp	1	$1.54	$1.54
1/4” Female TS Jack	1	$2.09	$2.09
10cm x 10cm PCB Housing	1	$11.10	$11.10
Electrolytic Capacitors	2	$0.22	$0.44
Ceramic Capacitors	3	$0.44	$1.32
0805 Surface Mount Capacitors	34	$0.09	$3.06
0.5W 5% Carbon Resistors	9	$0.07	$0.63
0805 Surface Mount Resistors	8	$0.08	$0.64
1' 5-Conductor Ribbon Cable	6	$2.10	$12.60
3.3' 20-Conductor Ribbon Cable	1	$9.00	$9.00
Microcontroller	2	$9.32	$18.64
Accelerometer	5	$7.00	$35.00
20-Position Through-Hole Header	2	$0.50	$1.00
5-Position Through-Hole Header	8	$0.38	$3.04
5-Conductor Receptacle Connector	8	$0.40	$3.20
Linear Voltage Regulator	1	$6.91	$6.91
Pair of Cotton Gloves	1	$1.00	$1.00
Plastic Chest Plate	1	$25.00	$25.00
1” Woven Nylon Straps with buckles	1	$8.95	$8.95
3/4” Adhesive Velcro Tape - 1'	2	$0.63	$1.26
9V battery	1	$12.60	$12.60
9V battery connector	1	$0.57	$0.57
2 x 3 header	2	$0.16	$0.32
2 x 4 header	2	$0.19	$0.38
Programmer	1	$30.99	$30.99
PCBs	6	$0.85	$5.10
Total			$196.38

Table 2: Comprehensive list of parts, amounts, and costs for Musical Hand

4 Schedule

Week	Ramsey	Thomas	Michelle
Feb 21 - 25	PCB board layout	Design Document	Design Document
Feb 28 - Mar 4	Finalize PCB design	Order components	Development on evaluation board
Mar 7 - 11	Breadboard testing	Breadboard testing	Continue program development
Spring Break	Begin soldering and breadboard test	Begin soldering and breadboard test	Development board prototyping
Mar 21 - 25	Draft PCB v2	Soldering and PCB I/O testing	Continue program development
Mar 28 - Apr 1	Finish soldering and I/O testing	Finish soldering and I/O testing	Convert program to PCB
Apr 4 - 8	Finish physical design	Finish physical design	Debug program
Apr 11 - 15	Prepare for mock demo	Prepare for mock demo	Debug program
Apr 18 - 22	Mock demo, prepare for final demo	Mock demo, prepare for final demo	Mock demo, prepare for final demo
Apr 25 - 29	Final demo, prepare for presentations	Final demo, prepare for presentations	Final demo, prepare for presentations
May 2 - 6	Deliver presentation, finish paper	Deliver presentation, finish paper	Deliver presentation, finish paper

Table 3: Schedule for completion of the musical hand

5 Conclusions

5.1 Accomplishments

Altogether, the Musical Hand's design process from start to finish was a complete success with only minor setbacks. All of the original high level requirements for a minimally viable product were met and only a couple design tweaks were required. A portable, wearable, durable, and inexpensive synthesizer was created that granted a musician full mobility, allows for an output between G3 (196Hz) and A#6 (1864.66Hz), and applies an adjustable vibrato effect. While the original vision for this project included polyphony and a large continuous range of notes for each finger, for musicality's sake these ideas were abandoned. This was decided because moving three fingers independently of each other accurately was deemed too difficult, and landing on a specific pitch with such a large range would prove nearly impossible. Instead a successful finger select program was implemented, allowing the pitch of the right hand to determine which finger is output; one at a time. Also, fewer discrete pitches tuned to a blues scale were implemented and the three fingers were given a low, medium, and high range portion of the scale, corresponding to three different octaves. None of these design changes left an unsatisfactory result.

5.2 Uncertainties

Most of Musical hand's uncertainties were in the physical design and easily corrected. For instance, two pins of the operational amplifier integrated circuit were flipped, but this was fixed by bending these pins, soldering wires to them, and flipping their connections to the IC header. The other uncertainty came in the form of our 1/4” output jack connection. The original design did not couple the output, which is necessary to keep the signal from damaging audio equipment. However, this was easily fixed by soldering the necessary capacitor and resistor between the original output jack and a new one; requiring some extra space in the central subsystem housing.

5.3 Future Work

The design of the musical hand leaves much room for future work. More scales to play in different keys and modes, different wave tables to change timbres, other effects than vibrato, polyphony, and a start/stop function are all easily implementable.

5.4 Ethics

The only ethical consideration of the design was ensuring that the linear regulator did not get too hot for the user, causing injury. This was verified in the power module verifications subsection above.

References

[1] A. Devices. “”ADXL335: Small, Low Power, 3-Axis ±3 g Accelerometer Data Sheet (Rev. B)”.” (2010), [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL335.pdf (visited on 03/30/2022).

[2] Microchip. “”PIC32MK General Purpose and Motor Control (GP/MC) Family Data Sheet”.” (2019), [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/PIC32MK_GP_MC_Familly_Datasheet_60001402G.pdf (visited on 03/30/2022).

[3] L. Technology. “”LT1963 - 1.5A, Low Noise, Fast Transient Response LDO Regulators”.” (2005), [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/1963fc.pdf (visited on 03/30/2022).

[4] T. Instruments. “”Single-supply, 2nd-order, multiple feedback low-pass filter circuit”.” (2021), [Online]. Available: https://www.ti.com/lit/an/sboa231/sboa231.pdf?ts=1648683390966&ref_url=https%253A%252F%252Fwww.google.com%252F (visited on 03/30/2022).

Appendix A

Note Hand	
Requirements	Verification
Accelerometers must output a voltage between -0.3 - 3.6V	Using a power supply, connect the PCB-attached accelerometer to 3.3 V and ground. Connect the analog voltage outputs to an oscilloscope for monitoring. Quickly moving the accelerometer along the x, y, and z axis, ensure that the output remains between -0.3 and 3.6 V.
From a flat hand, a finger dip of 90 ± 10 degrees must cause a change in output voltage of 0.55V ± 0.3V	Using a power supply, connect the PCB-attached accelerometer to 3.3 V and ground. Connect the analog voltage outputs to an oscilloscope for monitoring. Wearing an accelerometer PCB on the back of a gloved finger, dip the finger from parallel to perpendicular to the floor, ensuring the voltage changes by 0.55V ± 0.3V

Table 4: Note Hand Requirement and Verification Table

Effect Hand	
Requirements	Verification
Accelerometers must output a voltage between -0.3 - 3.6V	Using a power supply, connect the PCB-attached accelerometer to 3.3 V and ground. Connect the analog voltage outputs to an oscilloscope for monitoring. Quickly moving the accelerometer along the x, y, and z axis, ensure that the output remains between -0.3 and 3.6 V.
Using a flat hand as the 0 degree reference point, when the wrist is rotated from -90 degrees (thumb down) to 90 degrees (thumb up) there must be a change in the output value of 1.1V with a ± 0.55V tolerance.	Using a power supply, connect the PCB-attached accelerometer to 3.3 V and ground. Connect the analog voltage outputs to an oscilloscope for monitoring. Attaching the accelerometer PCB to the back of a glove, rotate from -90 to 90 degrees, ensuring an output change of 1.1V with a ± 0.55V tolerance.

Table 5: Effect Hand Requirement and Verification Table

Power Module	
Requirements	Verification
The linear regulator needs to supply 3.3V ± 0.05V tolerance to power the system.	Connect output of linear regulator to oscilloscope or voltmeter, observing the output voltage is 3.3V ± 0.05V.
The 9V battery must be able to supply 9V ± 2V tolerance to the amplifier under operating conditions before a voltage divider creates a 3.3V ± 0.3V DC offset.	Connect the 9V battery of the power module to the amplifier. Drive the amplifier with a 3.3V peak-to-peak sine wave from a function generator. Using an oscilloscope, observe that 9V ± 2V is provided and the DC offset of the amplified signal is 3.3V ± 0.3V.
Under operating conditions, the regulator will not exceed 125°C	Using an IR thermometer during verification 1, observe the linear regulator stays beneath 125°C

Table 6: Power Module Requirement and Verification Table

Microcontrollers	
Requirements	Verification
Microcontroller 1 needs to map finger angles to frequencies between 196 ±15 Hz (G3) to 1760 ±120 Hz (A6)	Hold sensor input stable with flat hand Use SNAP debugger to iterate through the program and observe output values. Connect oscilloscope or voltmeter to probe test points/breakout pins on Microcontroller 1 and confirm the output matches the software output Repeat b and c for a sensor input of a completely bent hand
Microcontroller 2 needs to output PDM signal	Connect output of Microcontroller 2 to oscilloscope to observe signal shape

Table 7: Microcontroller Requirement and Verification Table

Filter Module	
Requirements	Verification
Output waveform voltage must be between 0.25V - 3.3V when driven by a 3.3V Peak- Peak PDM signal.	With a function generator set to a 3.3V peak-peak square wave, measure the output voltage of the filter with an oscilloscope, ensuring that it stays between 0.25V - 3.3V sweeping the DC% from 0-100% during the test.
Cut-off frequency of filtered signal is 10 kHz +/- 11% with 40dB/dec +/- 10dB/dec attenuation after cut-off point.	Connect filter input to a function generator set to drive a 3.3 V peak-peak sine wave. Connect filter output to oscilloscope. Starting at 7 kHz, sweep the frequency upwards to 20 kHz. Record oscilloscope output data, verifying the location of cut-off is within +/- 11% of 10 kHz and the attenuation follows a 40dB/dec slope within +/- 10dB/dec.
The Op-Amp IC must not exceed the maximum temperature rating of 70°C under operating conditions	Using an IR thermometer, monitor the temperature of the Op-Amp during Verification 1 and 2, verifying an under 70°C temperature throughout.

Table 8: Filter Requirement and Verification Table

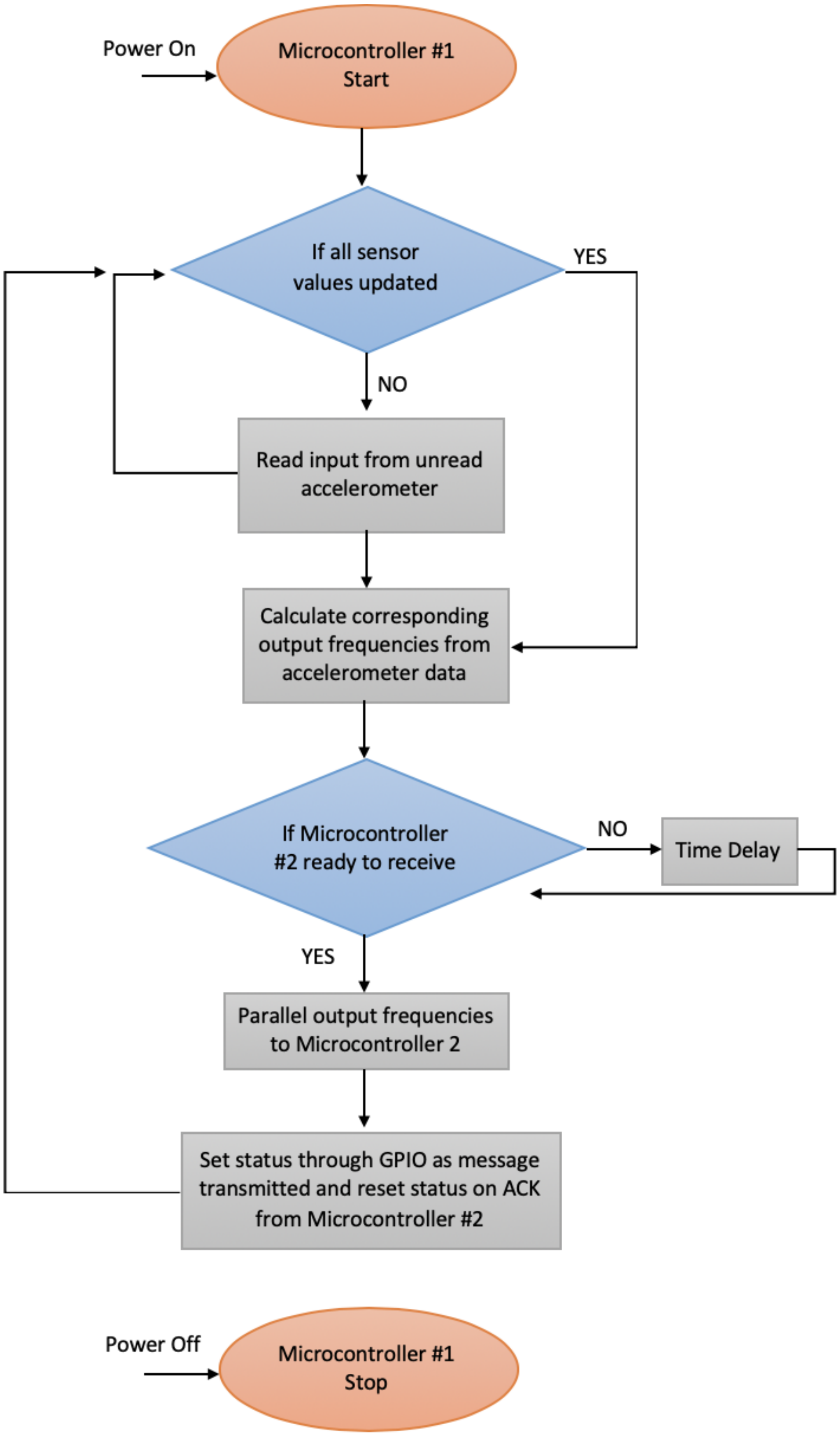

Figure 16: MC1 Software Flowchart