MIDI Controller Sequencer

ECE 445 Design Document -- Spring 2019

Team: Devin Alexander (dbalexa2),
Martin Lamping (mdl3),
Nathan Zychal (nzycha2)
TA: Christopher Horn

Date: 2/21/2019

Team #: 2

1 Introduction

1.1 Objective

Modern day software interfaces give a solution to the issue of music accessibility and creation; however, we believe a dynamic *physical* solution has not been implemented. Our goal is to create an easy to use physical interface for prototyping and designing music and melodies for the modern-day producer. We will be creating a sequencer that works based off of MIDI communication. The “steps” in each sequence will play a sound given a pitch and root note, and pitches will either be set on a continuous or quantized scale. In order to create a quantized scale, the voltage from each motorized potentiometer must be read into an ADC. The frequency for each step will be set from positions of the motorized potentiometers and the note will eventually be output of the device.

1.2 Background

Music has undoubtedly been an important part of human development. Every known culture, past and present, has expressed some form of music or melodies. Theories suggest that music has been around from dates as early as 500 A.D. [1]. The four widely accepted eras of music are, the Baroque Era, the Classical Era, the Romantic Era, and the Contemporary Era [2][3]. Music variety and accessibility increased dramatically toward the end of the Romantic Era/beginning of the Modern Era (~1880) with the creation of the gramophone [4] and start of radio broadcasting [5]. Radio broadcasting had allowed for music to reach nearly 40 percent of American households [6]. In comparison to other fields such as the sciences, music development had been relatively slow. The 20th century not only led to inventions that could play music such as the gramophone, but also music recording [7] and later distribution. The use of sheet music was largely limited to that of the middle and upper class because the people using sheet music needed to be able to sing, play and read music [8]. Accessibility and the creation of music has always been an issue and continues to be in modern day.

There have been various attempts at creating a dynamic sequencer with many features in modern day (particularly software based DAWs). There are far fewer hardware-based implementations of sequencers. Software based sequencer implementations (included in DAWs) have been implemented by big name companies such as Sony or Yamaha, however they are not the most popular. Instead, recording software such as FL Studio, Ableton, or Logic Pro are more commonly used and have in some cases been ported to physical synthesizers [9].

1.3 High-Level Requirements

- User inputs data (e.g. pitches, root notes, tempo, and scale) this is output as MIDI data representing a sequence of notes at a consistent tempo.
- User input data is used to determine the duration of the motor pulses and bring the potentiometer sliders into position.
- Sequencer status information is output accurately onto the LCD screen.

2 Design

2.1 Block Diagram

The block diagram will meet the high level requirements mentioned in the Section 1.3 above. The motor controllers are connected directly to the potentiometer motors and I2C communication protocol will be used to optimize data transfer rates. The arrows below (in the block diagram) indicate which sections of the sequencer will need to communicate with each other and over what protocol. Different power (5 or 9 volts) will be routed to the appropriate module depending on its logic (or lack thereof). The 16 motorized potentiometers will be used to output specific tonal data which will be transferred through the ADC. The ADC's output will be read and processed by the microcontroller and MIDI data will be output. The LCD will show current user settings and processed music information. A *maximum* of 16 motor potentiometers will be used .

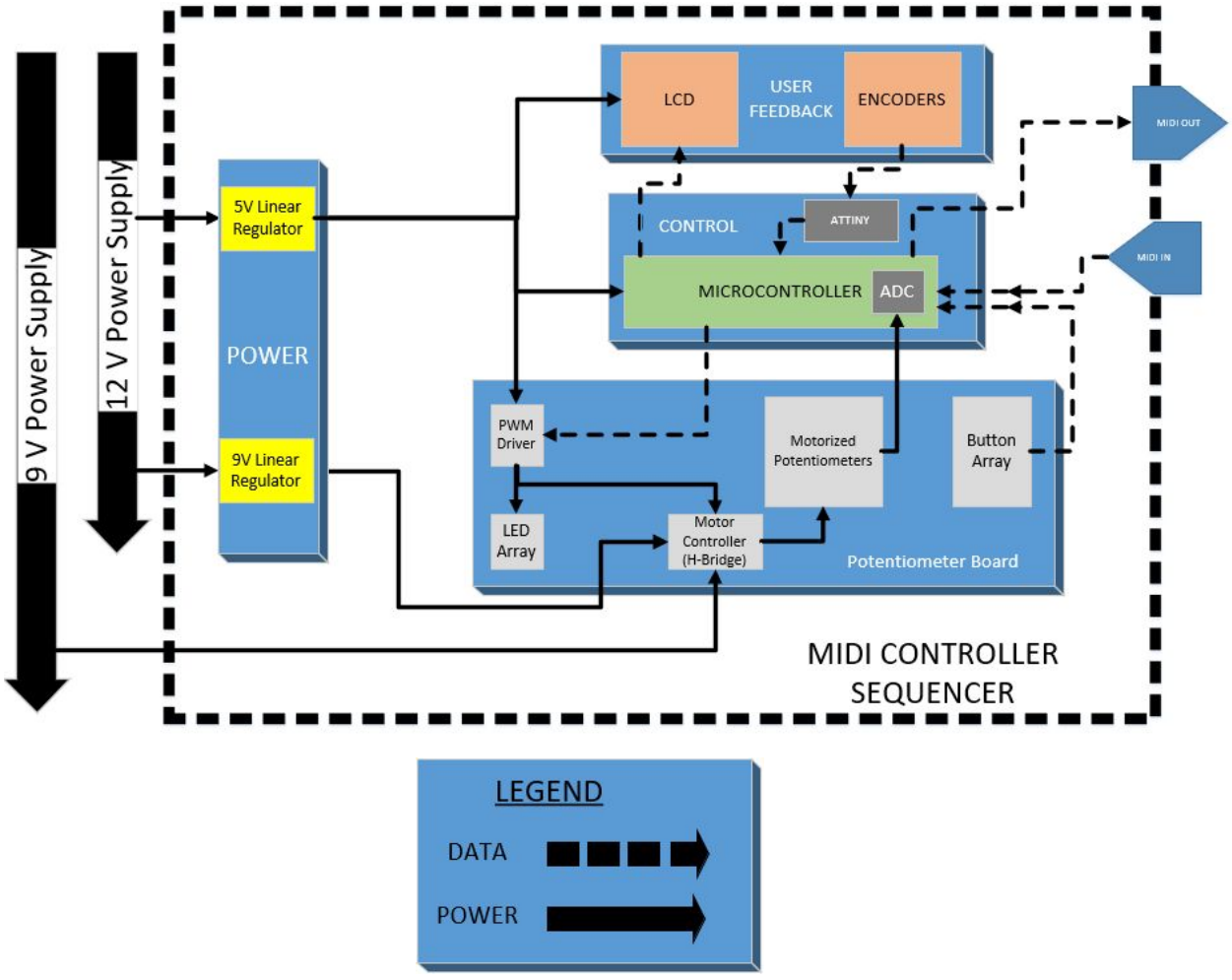


Figure 1: Block diagram implementation of the MIDI controller sequencer.
Input: MIDI In **Output:** MIDI Out

2.2 Physical Design

A 4u rack unit (4 * 1u) with a faceplate will be used to house all the PCBs internal to the sequencer and also hold the motorized potentiometers and LCD screen. It will also be retrofitted to house the buttons and rotary encoders. (Dimension(s): 4u= 7 inches height x 19 inches wide.)

Note: This physical casing of the sequencer will be considered a reach goal

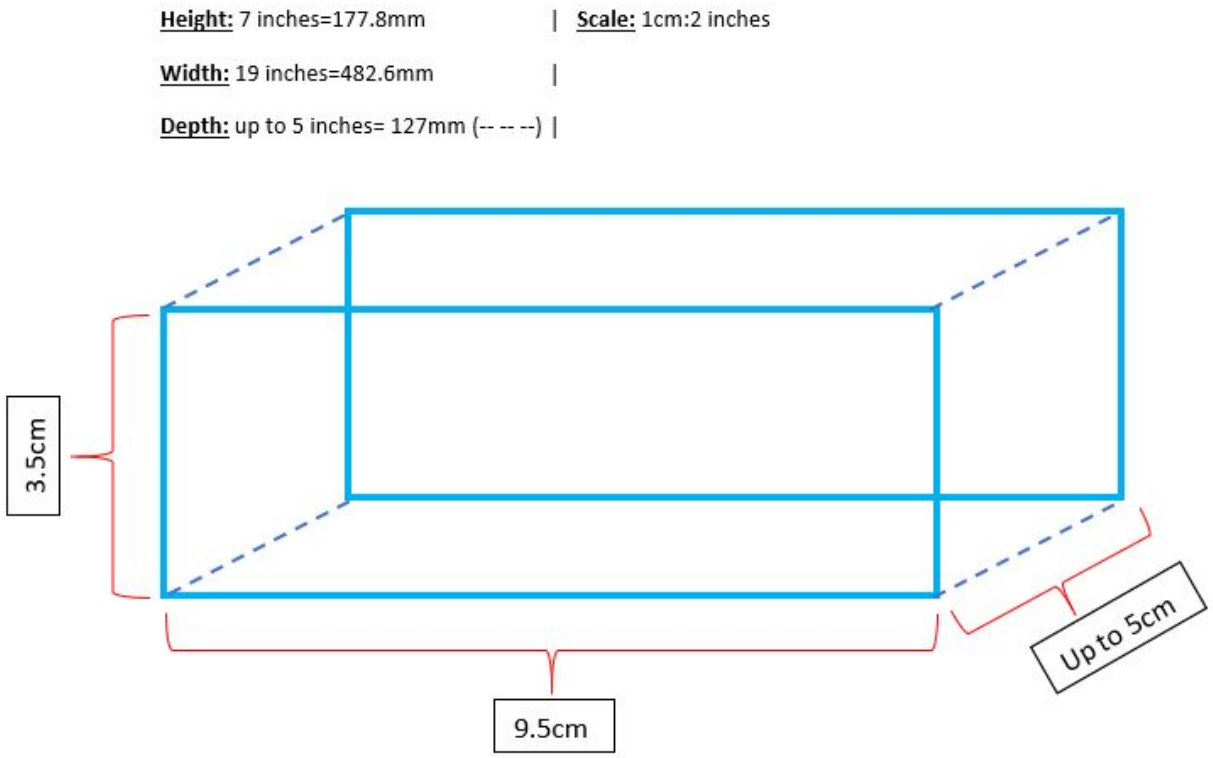


Figure 2: Potential Spatial Requirements

Figure 3: 4u Rack Unit Without Faceplate. Potential housing unit
Note: *If* a faceplate is designed it will be retro-fitted to hold all the motorized potentiometers, LCD screen, buttons, and encoders.

2.3 Block Design

2.3.1 / 2.3.2 Functional Overview and RV Tables

2.3.3 Power Supply

This module is external to our design. We will be using a pre-existing power supply blocks to provide power to the system.

12 V Power Supply Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Provides +/- 12 V with a tolerance of +/- .1 V .	A. Set up mock test circuit with rated resistor. B. Apply 12 V power supply to rated resistor. C. Use digital multimeter to test that the voltage drop across the resistor does not exceed 12.1 V.

2.) Supplies 4.5 A +/- 2A	A. Set up mock test bench circuit with resistive load of 1.10 □ (comparable to circuit). B. Calculate current across resistor.

9 V Power Supply Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Provides +/- 9 V with a tolerance of +/- .1 V .	A. Set up mock test circuit with rated resistor. B. Apply 9 V power supply to rated resistor. C. Use digital multimeter to test that the voltage drop across the resistor does not exceed 9.1 V.
2.) Supplies 10 A +/- 2 A.	A. Set up mock test bench circuit with resistive load of .9 □ (comparable to circuit). B. Calculate current across resistor.

2.3.4 Linear Voltage Regulators

Two types of linear voltage regulators are required to step down the 9 V from the power supply. This is necessary because all of the ICs will need to be supplied with 5 V power. The 5 V linear regulator will be used to to power the microcontroller and therefore will need to function within +/- .5 V deviation. The 9 V linear regulator will be used as a reference voltage, therefore its tolerance will be much lower than the 5 V. We contend a deviation of +/- .1 V.

9 V Linear Regulator Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Outputs at a voltage level of 9 V with a tolerance of +/- .1 V at a maximum current output of 500 mA.	A. Set up linear voltage regulator mock up circuit with 500 mA load. B. Connect known 12 V power supply to voltage regulator. C. Connect voltage regulator to 24 □ in order to pull 500 mA. D. Test output is operating within acceptable range of 8.9 V -9.1 V and the output current does not exceed 500 mA.

5 V Linear Regulator Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Outputs at a voltage level of 5 V with a tolerance of +/- .4 V at a maximum current output of 1500 mA.	A. Set up linear voltage regulator mock up circuit with 1500 mA load. B. Connect known 12 V power supply to voltage regulator. C. Measure output is operating within acceptable test range of 4.6 V - 5.4 V and the output current does not exceed 1500 mA.

2.3.5 LCD

The LCD will show the current states of user input such as the current note, octave,scale type, root note, tempo (clock rate), and duration (reach goal). The size of this LCD display will be made of (4 rows by 20 columns) that can print off (9 rows x 5 columns) ascii / sprites in each space. Minimal LCD refresh rate of 4 frames for second is a necessary condition to ensure time relevant musical data is displayed promptly. If there is unnecessary delay in data output to LCD then the displayed data will not be a real time representation. The LCD exchanges data with the microcontroller with I2C.

LCD Module Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Make sure that the contrast of the LCD display is such that the message can be seen on it from multiple viewing angles.	A. Configure microcontroller test circuit with LCD interface. B. Send known text to the LCD from microcontroller. C. Provide an analog voltage to the pin on the display controller that is responsible for controlling the contrast. D. Adjust contrast until viewable from top, sides and bottom with viewing angle of roughly 45 degrees from the z axis.

2.) LCD backlight brightness is great enough to see the text on the screen in a dark environment.	A. Configure microcontroller test circuit with LCD interface. B. Send known text to the LCD from microcontroller. C. Take test setup into dark room. D. Make sure that the voltage going to the LCD I2C to parallel backplate has the correct voltage to power the LED backlight. E. Ensure backlighting is bright enough
3.) Real time display of messages	A. Configure microcontroller test circuit with LCD interface. B. Supply the LCD with a series of known messages. C. Increase the refresh rate of that.
4.) The LCD display can not take up too many pins on the microcontroller.	A. To save IC digital pins for use of connecting to other modules, we will be using a Parallel to I2C BUS converter chip to free up digital pins for other requirements.
5.) LCD screen will run off a 5 V regulator +/- 0.5 V.	A. Connect linear regulator to 12 V power supply. B. Using multimeter measure the output of the linear input to the LCD screen and confirm voltage is 5 V +/- 0.5 V.

2.3.6 Encoders

The rotary encoders will be used to select scale, tempo, root, and duration. Once an output is turned to, the encoder's button can be pressed to confirm that selection. The encoder outputs Grey Code to allow for quicker selection cycling. Grey Code will allow for quicker cycling, because unlike binary count encoders, for Grey Code, only one bit is switching when the output selection changes. Two bits will be used for Grey Code count (as opposed to 4 bit) so that the microcontroller will have enough vacant pins. Encoder reach goals considered include a dial specifically for menu navigation. Each encoder used will have its own dedicated microcontroller separate from the main one to ensure there will be no latency issues.

Encoder Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Enough detent for active cycling.	A. Create list of all desired outputs that encoder needs to cycle through. B. Connect *every* encoder output to Arduino board pins. C. Use digitalRead(pin) syntax to read *every* pin output with Arduino IDE software. D. If detent of encoder is less than or greater than the number of desired outputs, ensure that the encoder cycles through the list.
1.) Tempo encoder has a range of 60 bpm - 140 bpm (1 Hz - 2.33 Hz.)	A. Turn encoder to minimum tempo of 60 bpm. B. Connect minimum frequency output of encoder to oscilloscope. C. Check that minimum frequency selection operates at or below 60 bpm. D. Turn encoder up to maximum tempo of 140 bpm. E. Connect maximum frequency output of tempo encoder to oscilloscope. F. Check that maximum frequency output of tempo encoder operates at or above 140 bpm.

2.3.7 Microcontroller

The microcontroller's main purpose will be to control the system as a whole and synthesize the MIDI data from analog and digital signals. It will need to complete various tasks, some of which are timing critical. The microcontroller will need to be capable of processing the MIDI data at a rate of 31.25 Kbaud (industry standard). It will use a host of protocols to talk to ICs on this device. We also require that the microcontroller has at minimum, two seperate UART channels to use for the MIDI IN and MIDI OUT. We may be using a dedicated IC to translate the USB data into serialized binary data that the microcontroller uses. The USB would be placed on the same PCB as the microcontroller. Otherwise we will use an external USB UART to serial programmer/debugger. Also, the ADC will be internal to the microcontroller.

Microcontroller Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Supports an I2C frequency of at least 100 KHz.	A. Input I2C test signal using computer and USB input port of breakout board. B. Measure output frequency and confirm it reaches at least 100 KHz +/- .5 KHz using oscilloscope.
2.) Crystal oscillator will run off a 5 V regulator +/- 0.2 V.	A. Connect linear regulator to 12 V power supply. B. Using multimeter measure the output of the linear oscillator and confirm voltage is 5 V +/- 0.2 V. C. Apply output of 5 V linear oscillator to the input of the crystal oscillator. D. Confirm normal operation of 16.000 MHz output of crystal oscillator via oscilloscope.
3.) Samples voltage levels at 1000 KHz.	A. Input square wave with peak to peak voltage 3.3 V from function generator. B. Connect microcontroller output to oscilloscope. C. Ensure that microcontroller is capable of sampling signal at 1000 KHz by measuring period of output sample.
4.) Microcontroller must be capable of reading a sample of at most 10 ms +/- 5 ms.	A. Set up microcontroller test bench and read signal values via oscilloscope. B. Provide microcontroller with known analog signal. C. Make sure that the known signal and values read into the microcontroller at the required 10 ms +/- 5 ms match.
5.) Samples voltage levels at 1000 KHz.	A. Input square wave with peak to peak voltage 3.3 V from function generator. B. Connect microcontroller output to oscilloscope. C. Ensure that microcontroller is capable of sampling signal at 1000 KHz by measuring the period of the output sample.

7.) ADC being internal to the microcontroller.	A. Check the datasheet for the microcontroller to see if there are any pins being utilized for analog to digital conversion. This will reduce design complexity.

2.3.8 Motorized Potentiometers

The user will be able to control the movement of the potentiometer sliders either manually or through the use of rotary encoders. This will in effect change the tone data that will be output through MIDI. The motorized potentiometers must have a resistive range from 0 k□ to 10 k□ +/- 2 k□ in order to have enough quantization steps representing the various note steps. The motorized potentiometers will have responsive feedback that will provide the user with a form of touch information. The fader will slide to specific locations based on the voltage reading, and will require variable pressure to move it from one of these predetermined locations. If the user leaves a fader in between two of the predetermined quantization locations, then the software will move it to the closest available position within a given musical scale.

Motorized Potentiometer Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Potentiometer ranges from 0 K□ - 10 K□ with a tolerance +/- 2 K□.	A. Configure individual potentiometer testbench. B. Apply known voltage across potentiometer. C. Measure output current as the potentiometer position is adjusted. D. Ensure that all potentiometers operate in at 10 K□ +/- 2 K□.
2.) Ability to drive motors such that positions can be “quantized” to localized positions along potentiometer movement range.	A. Configure individual potentiometer testbench. B. Configure Microcontroller subcircuit with test code that will slide through all necessary positions. C. Ensure smooth transition between locations. D. Ensure lack of movement in between movements.

3.) Each individual motor needs to have noise from surrounding motors filtered out from the output voltage before being sent to ADC.	A. Connect output from the potentiometer to an oscilloscope. B. Configure microcontroller subcircuit that will slide through all necessary potentiometer positions. C. Inspect that the amplitude of any periodic spikes in voltage from the motor from that potentiometer (or nearby potentiometers) is less than 8 mV +/- 2 mV.
4.) Potentiometer sliders will need to be moved at a speed that seems fluid but does not harm users.	A. Configure individual potentiometer testbench. B. Configure microcontroller subcircuit with test code that will slide through all necessary note positions. C. Test out the motor controllers with human hands touching the knobs. D. Determine that movements do not cause discomfort.

2.3.9 PWM Driver (Used to drive the LEDs)

We will be using an IC to provide PWM power signals for controlling the LEDs. We must ensure that the output of the PWM can reliably apply 20 mA of current per each PWM output pin. Higher “frequencies” can also be reached if needed, switching the PWM drivers’ mode of operation will allow for this. Fast mode allows for variable pulse width while operating within the same required frequency range of normal mode.

PWM LED Driver Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) I2C bus on the microcontroller, that functions at a frequency of 100 KHz.	A. Connect mockup PWM driver circuit to oscilloscope. B. Run data through the I2C bus. C. Measure the frequency at which I2C signal oscillates. D. Check If the signal oscillates at a frequency within +/- .1 KHz of the required 100 KHz.
2.) PWM driver will run off a 5 V regulator +/- 0.2 V.	A. Connect linear regulator to 12 V power supply. B. Using multimeter measure the output of the linear oscillator and confirm voltage is 5 V +/- 0.2 V. C. Apply output of 5 V linear oscillator to

	the input of the crystal oscillator.
3.) The LED driver provides pulses that are able to drive LEDs at 20 mA per channel.	A. Assemble microcontroller and LED driver subcircuit. B. Use multimeter to check the LED's channel current is within +/- 5 mA of the required 20 mA.

2.3.10 PWM Driver (Used to drive the motors)

The driver specific to the motors will use a PWM signal interact with the H-Bridge. The PWM signal will pulse differently based on if the positions have been pre-set (quantized) along the potentiometer slider or if there are no quantized positions (continuous). The H-Bridge in **2.3.11** has an internal pull down resistor with a value of 200 K□. In order to ensure current flows through the PWM channel to trigger the H-Bridge, a resistance lower than 10 K□ will be necessary in between the input to the H-Bridges PWM input and the corresponding PWM output channel.

PWM Motor Driver Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) I2C bus on the microcontroller, that functions at a frequency of 100 KHz.	A. Connect mockup PWM driver circuit to oscilloscope. B. Run data through the I2C bus. C. Measure the frequency at which I2C signal oscillates. D. Check If the signal oscillates at a frequency within +/- .1 KHz of the required 100 KHz.
2.) PWM driver will run off a 5 V regulator +/- 0.2 V.	A. Connect linear regulator to 12 V power supply. B. Using multimeter measure the output of the linear oscillator and confirm voltage is 5 V +/- 0.2 V. C. Apply output of 5 V linear oscillator to the input of the crystal oscillator.

2.3.11 H-Bridge (Motor Driver)

We will be using the an H-Bridge for this project to drive the motors. Each H-Bridge (motor driver) will be used to control the movement of 2 motors simultaneously. The H-Bridges will be placed as close as possible to each set of two motors to maximize physical space on the PCB. The H-bridge IC is necessary because the motors will need to be driven both backwards and forwards. Additionally, the H-bridge will also need to be powered off a 5 V voltage regulator. In accompanying **2.3.17** Support materials

H-bridge subsection you can find a simulation that confirms the H-bridge will have the necessary functionality we desire.

H-Bridge Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Receive an isolated power source (not from the 9 V linear regulator) with a voltage rated for both +9 V and -9 V to drive the motors.	A. Configure H-Bridge testbench on breadboard. B. Configure two test motors to testbench on breadboard and connect in parallel to H-Bridge. C. Using a multimeter measure the voltage across the motor and confirm voltage is 9 V +/- 1 V. (8 V -10 V range).
2.) Motors will need to be driven in either direction.	A. Configure H-Bridge testbench on breadboard. B. Configure two test motors to testbench on breadboard and connect in parallel to H-Bridge. C. Apply +/- 10 V to each motor and check that the potentiometers are driven by the motor in opposite directions.

2.3.12 Button Array

Buttons will be used for various functions throughout the device such as turning a selected motorized potentiometer on and off and encoder output selection. Reach goals considered include a randomization button and power button. The randomization button would randomized selected potentiometers to change to note values within the selected scale. The power button would cut power to the entire device. Typical / tolerable button bounce times range from 400 µs (more common) to 3000 µs between switch contact [10]. Time between samples will thus be controlled to keep button press readings accurate and over sampled at a rate over the typical bounce times.

Button Array Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Output digital HIGH (1) when pressed or digital LOW (0) when pressed depending on if the button is active high or active low.	A. Configure button testbench on breadboard. B. Use a multimeter to read voltage across button when it is pressed. C. Check if voltage is above the critical point of 2.5 V within +/- .2 V.

2.) Debounce buttons.	A. Build a lowpass filter with capacitor in parallel with the button and resistor in series. B. Connect oscilloscope to the output of button. C. Apply 3 V input to button. D. Map button activation on oscilloscope. E. Ensure that the button remains in the high position with tolerable bounce (listed in **2.3.12** summary above).

2.3.13 LED Array

LEDs will be used as indicators, placed at various locations on the faceplate of the rack unit. They will indicate the current potentiometer(s) that is (are) currently being used. The LEDs will clearly indicate when a step is occurring in the sequence of notes. An individual LED will illuminate as a particular potentiometer value is read and played. Then the next LED, in the sequence will illuminate as the next potentiometer is activated, so on and so forth. An LED will also illuminate when a particular potentiometer is selected to change the value.

LED Array Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) The LEDs are RGB and can be changed to any color in the the RGB colour space .	A. Cycle through absolute RGB values. B. Ensure correctly display R, G and B C. Variation in LEDs will be judged by running the R , G, and B values. through the standard 256 color value ranges.
2.) The LEDs must be driven correctly by PWM signals generated by an external	A. Set up I2C connection protocol between microcontroller and LED array. B. Ensure an I2C data signal from the Microcontroller has individually addressed PWM channels. C. Test I2C communication with all three color channels on each LED individually.

2.3.14 MIDI Ports (IN & OUT)

The MIDI Ports will provide the means of speaking MIDI to external controllers and devices. Ports will be labeled for easy identification for the user. They also need to be

firmly fastened into their enclosure in the case. MIDI encoding uses eight bits to specify data modes such as polyphonic aftertouch, channel aftertouch, pitch bend, controller change, patch change, and active sense data transfer. MIDI encoding has become one of the most widely used industry standards for musical data communication applications.

MIDI Ports Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION**
1.) Successfully outputs MIDI data with industry standard bit rate of 31.25 Kbaud ~ 3 bytes / ms.	A. Mock up microcontroller test board. B. Connect MIDI input and output data lines to UART port (solder). C. Supply MIDI port with known sequence of MIDI data from computer DAW. D. Check that data is being sent at a rate of 31.25 Kbaud~ 3 bytes / ms.
2.) Ports connected to microcontroller on two of the four UART ports.	A. Mock up microcontroller test board. B. Connect MIDI input and output data lines to UART port (solder). C. Supply MIDI port with known sequence of MIDI data from computer DAW. D. Check that data is being transferred to the microcontroller ensure successful connection.
3.) Provide a means to communicate with external MIDI devices.	A. Mock up microcontroller test board. B. Connect MIDI output port to computer DAW software. C. Flash know MIDI sequence that probes the ‘status bit’ onto microcontroller and output through MIDI output port. D. Using a DAW (digital audio workstation), check that the MIDI encoding standard “status bit” is sent with regularity.

4.) MIDI will output on, off, duration, velocity, and tempo data.	A. Synchronize the MIDI output data to that of a MIDI input device to verify if sound is coming out of a MIDI module. B. If status byte of the MIDI output data, is flashed high every 1 ms +/- 1 ms for each parameter.
5.) MIDI input will only be used for syncing with external tempo data and nothing else.	A. Mock up microcontroller test board. B. Connect MIDI input and output data lines to UART port (solder). C. Supply MIDI port with known tempo data from computer DAW. D. Ensure correct synchronization of the MIDI input clock with that of a MIDI output via the DAW using a MIDI out port on a digital interface for testing.

2.3.15 Barrel Port Connector with Input Fuse Protection

A barrel port connector will be used to interface the sequencer with the 9 V power supply. Due to potential user negligence we will provide fused circuit protection on this input in case of overvoltage. This fuse will allow for up to 10 A (per potentiometer PCB) to be supplied to the circuitry but no more. Correct failure conditions will be tested (listed in the RV table below.)

9V Barrel Port Connector Input Fuse Protection Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Apply over current conditions (10 A) at power supply input and confirm fuse functions properly.	A. Mock up barrel port connector with added fuse protection. B. Connect fuse protection to 0.9 □ resistor to pull 10 A. C. Connect multimeter across resistor to measure voltage. D. Ensure that fuse blows and voltage across resistor goes to 0.

12V Barrel Port Connector Input Fuse Protection Requirement and Verification Table	
REQUIREMENT:	**VERIFICATION:**
1.) Apply over current conditions (5 A) at power supply input and confirm fuse functions properly.	A. Mock up barrel port connector with added fuse protection. B. Connect fuse protection to 3 □ resistor to pull 4.5 A. C. Connect multimeter across resistor to measure voltage. D. Ensure that fuse blows and voltage across resistor goes to 0.

2.3.16 Software

The software for our design has five main functional goals. The first is to process user input and display user input settings to the LCD screen. The second is to process the MIDI data that was either input to the system or output to the DAW. The third process necessary for the software to handle will be to help mitigate internal noise errors. This is detailed more extensively in the subsection of tolerance analysis below. Possible software implementations to mitigate this noise are in our sampling frequency, or a data averaging technique. We can also implement functions that will isolate and remove outliers in the voltage data. The fourth functional goal is to output user feedback, whether that is illuminating LED's or current note selections. The fifth and final software function is to quantize the fader positions and produce user feedback to these locations. In other words, when the user moves a fader out of position, they will encounter a resistance that is much greater in comparison to the intrinsic physical resistance of the potentiometer. For quantization, the algorithm implemented in software functions by consistently checking the position of the potentiometer and moving the slider to the closest quantized position. The further a potentiometer slider from a quantized position, the greater the duty cycle of the PWM will be. Once the quantized position is reached, the pulses are halted. The fifth functional goal of the software will snap the potentiometer to the closest available location when it is left in between positions. In short, the motors have the potential to induce an EMI current on unshielded traces or IC's. An attached software flowchart can be found in the **2.3.17** supporting materials flowchart subsection.

The software will interpret three (with the possibility of adding two more) main functions from the rotary encoders. Those functionalities are: tempo adjustment, root note (octave) selection and scale selection. If time permits, a menu select, randomization button and note duration functionality will be implemented.

Furthermore, the software will be tasked with determining which potentiometers are activated via push buttons underneath each potentiometer. When a button is pressed, the software needs to illuminate an LED above that specific motorized potentiometer.

2.3.17 Supporting Material

Circuit Schematics

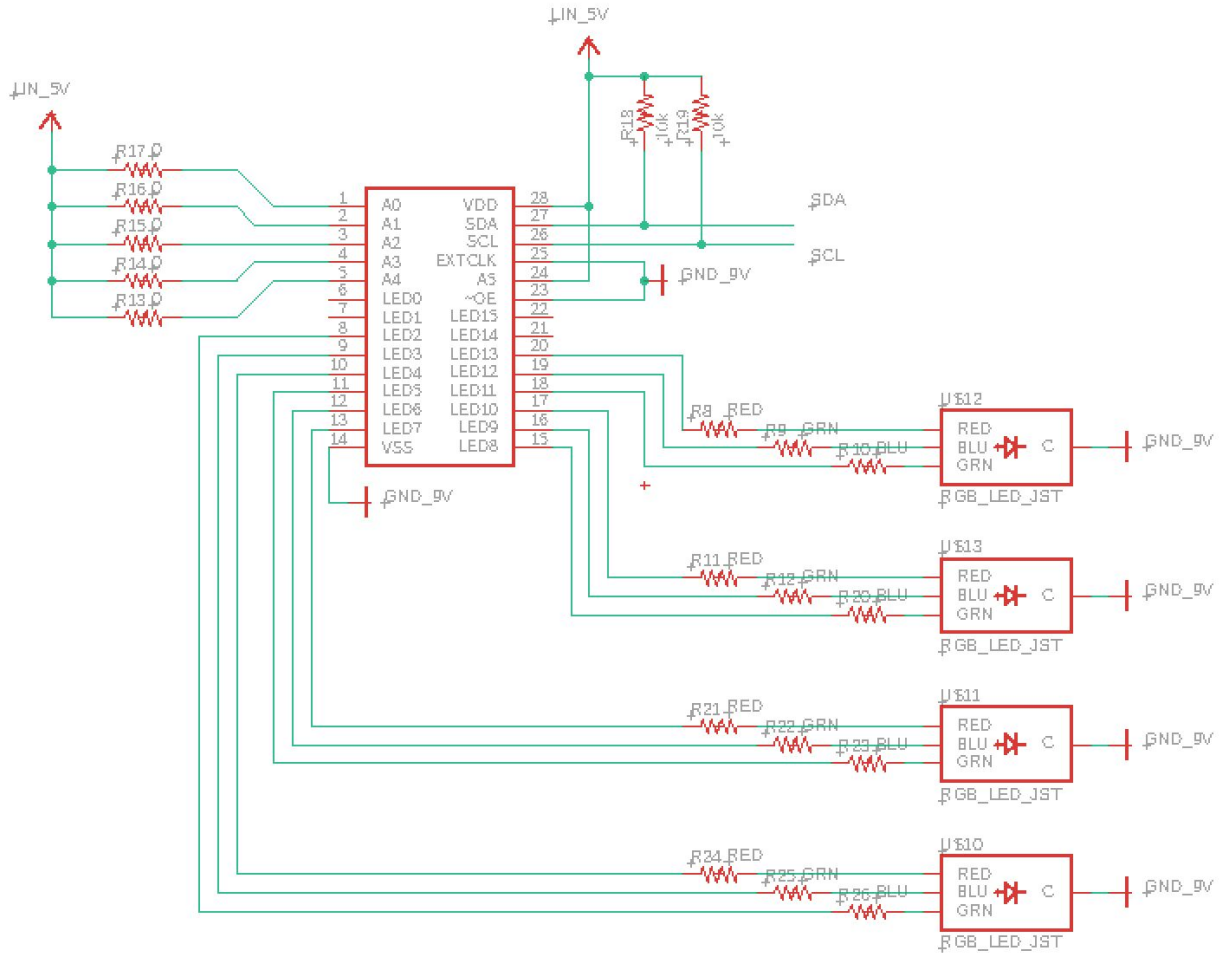


Figure 4: RGB LED Driver Circuit:

Description: This circuit will be used to display the current potentiometer being used for the output sequencer step.

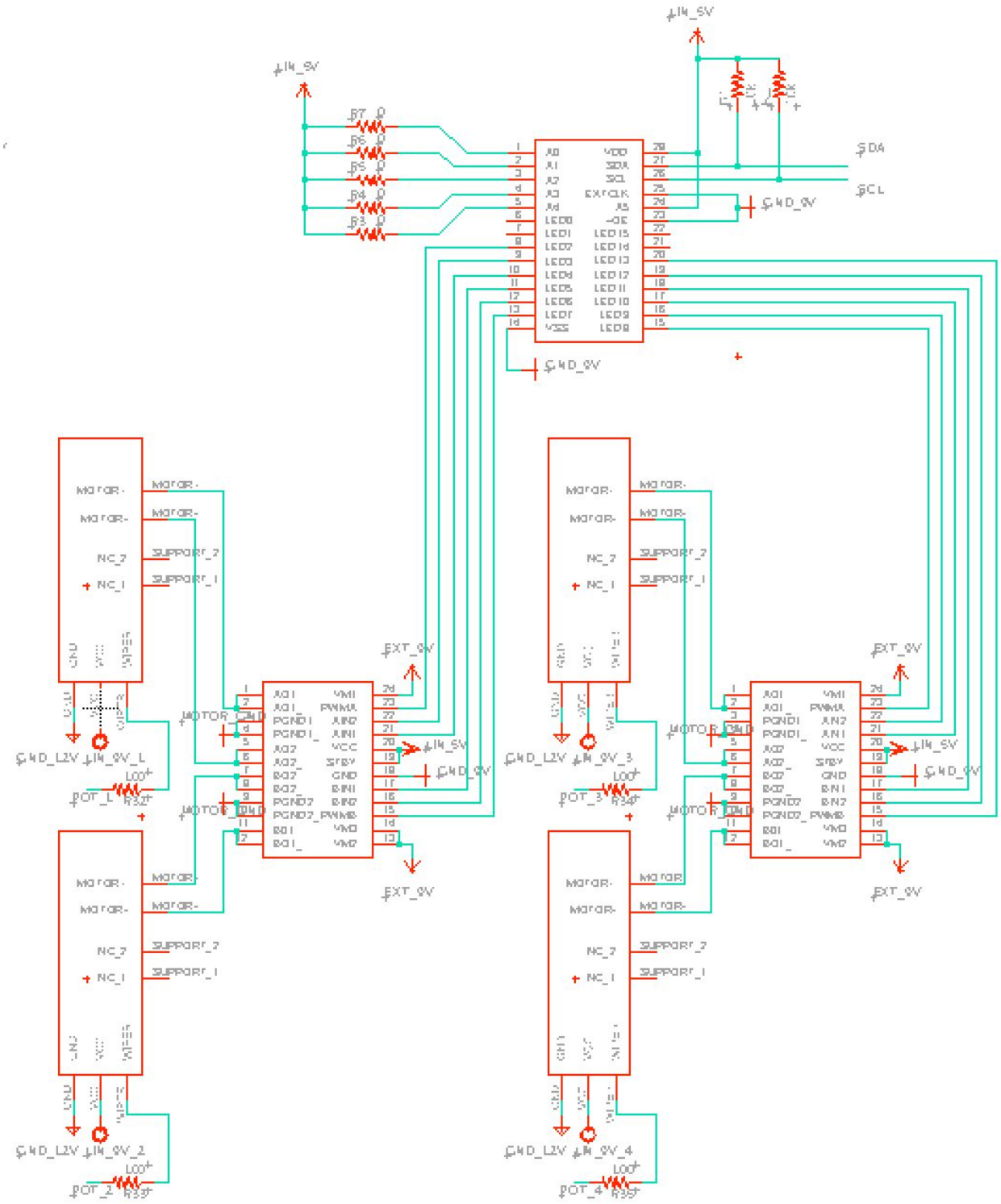


Figure 5: Motor Controller and Potentiometer Circuit:

Description: This is an individual potentiometer submodule for the system. In total, there will be four of these subcircuits that then make up the total potentiometer module.

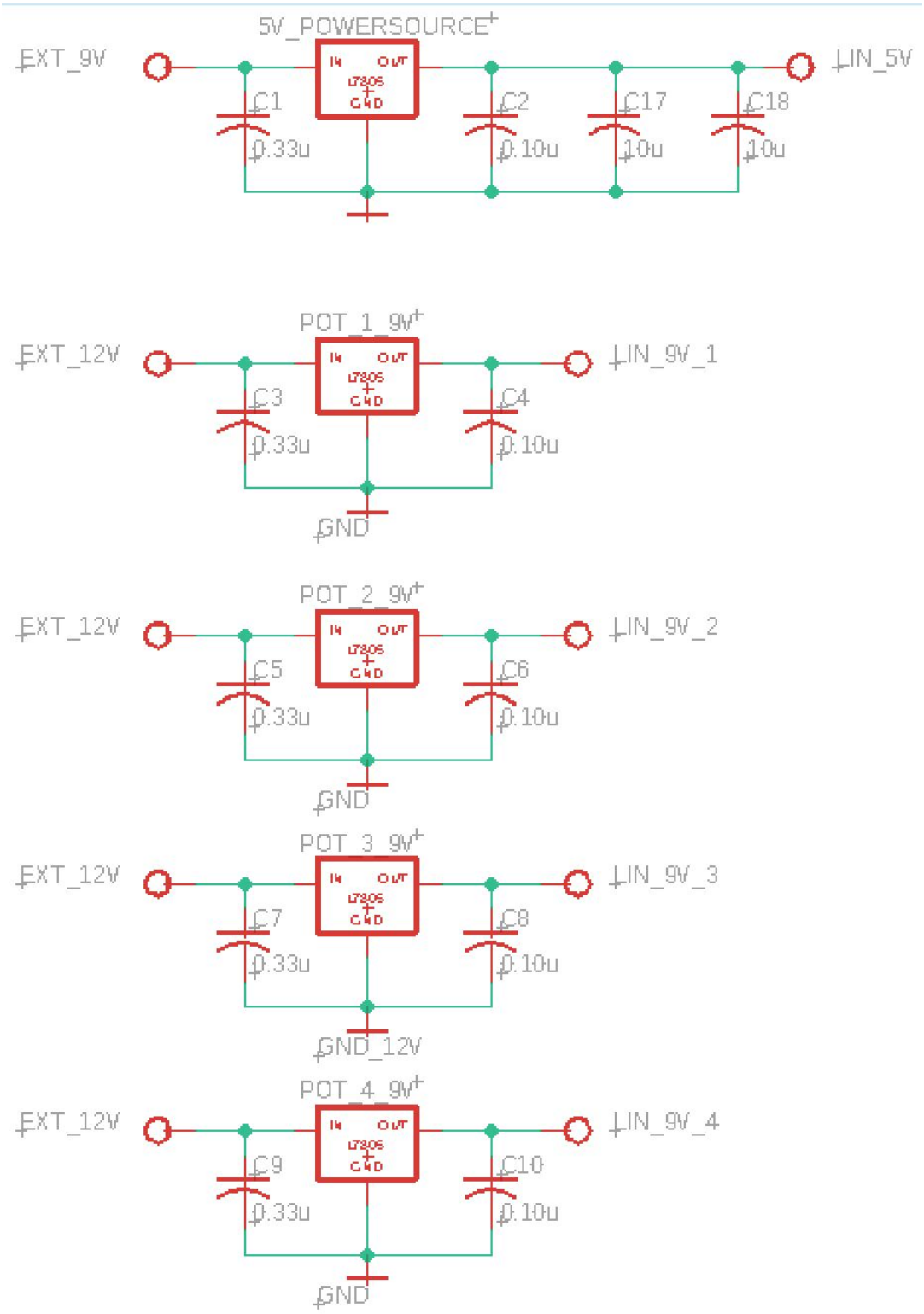


Figure 6: Linear Regulator Array

Description: The four 9 V linear regulators will be used to provide a potential for the potentiometers. The single 5 V linear regulator provides power to all ICs on PCB.

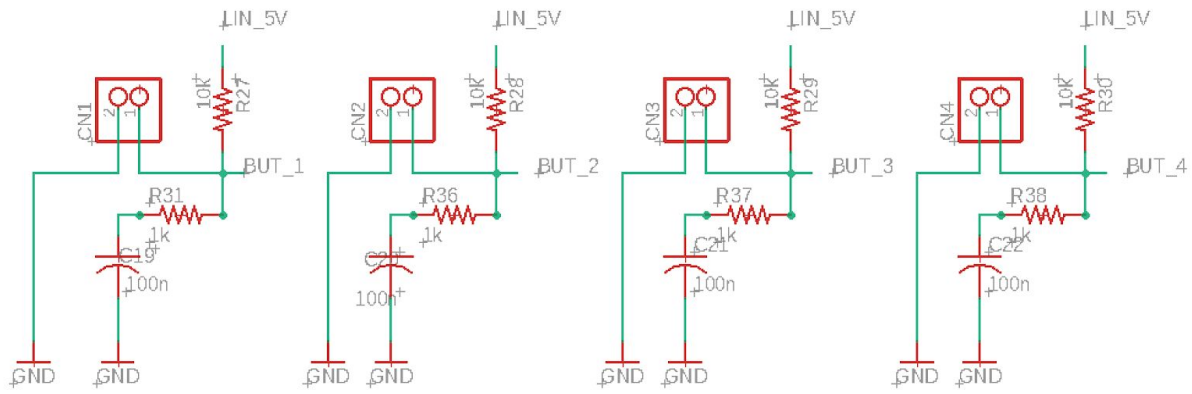


Figure 7: Button Array with low pass filter

Description: These will be 2 pin JST headers for connecting external buttons to a faceplate.

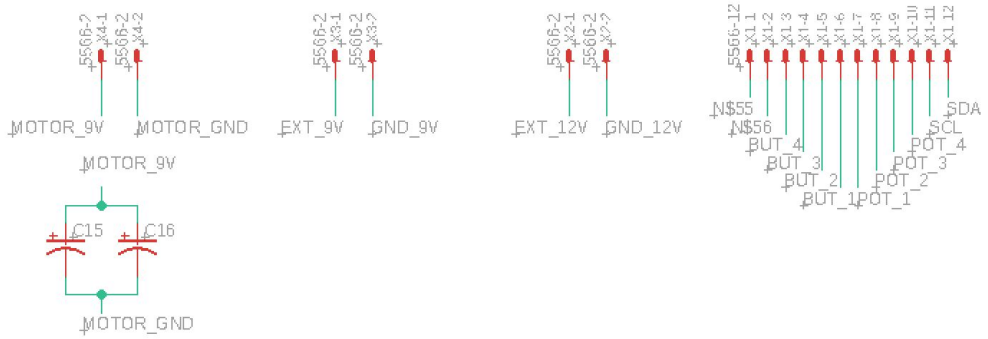


Figure 8: External Headers

Description: This is how the potentiometer PCB interfaces with the Microcontroller PCB.

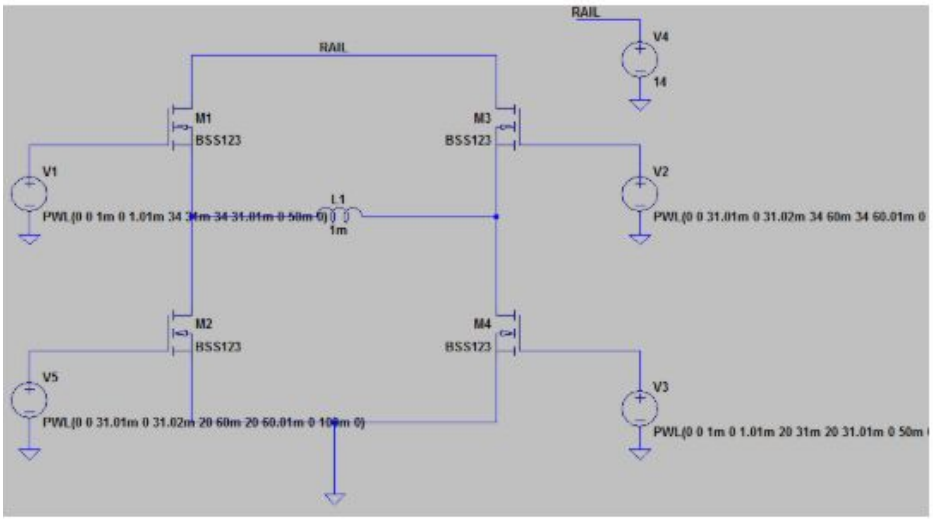


Figure 9: LTspice H-Bridge Schematic

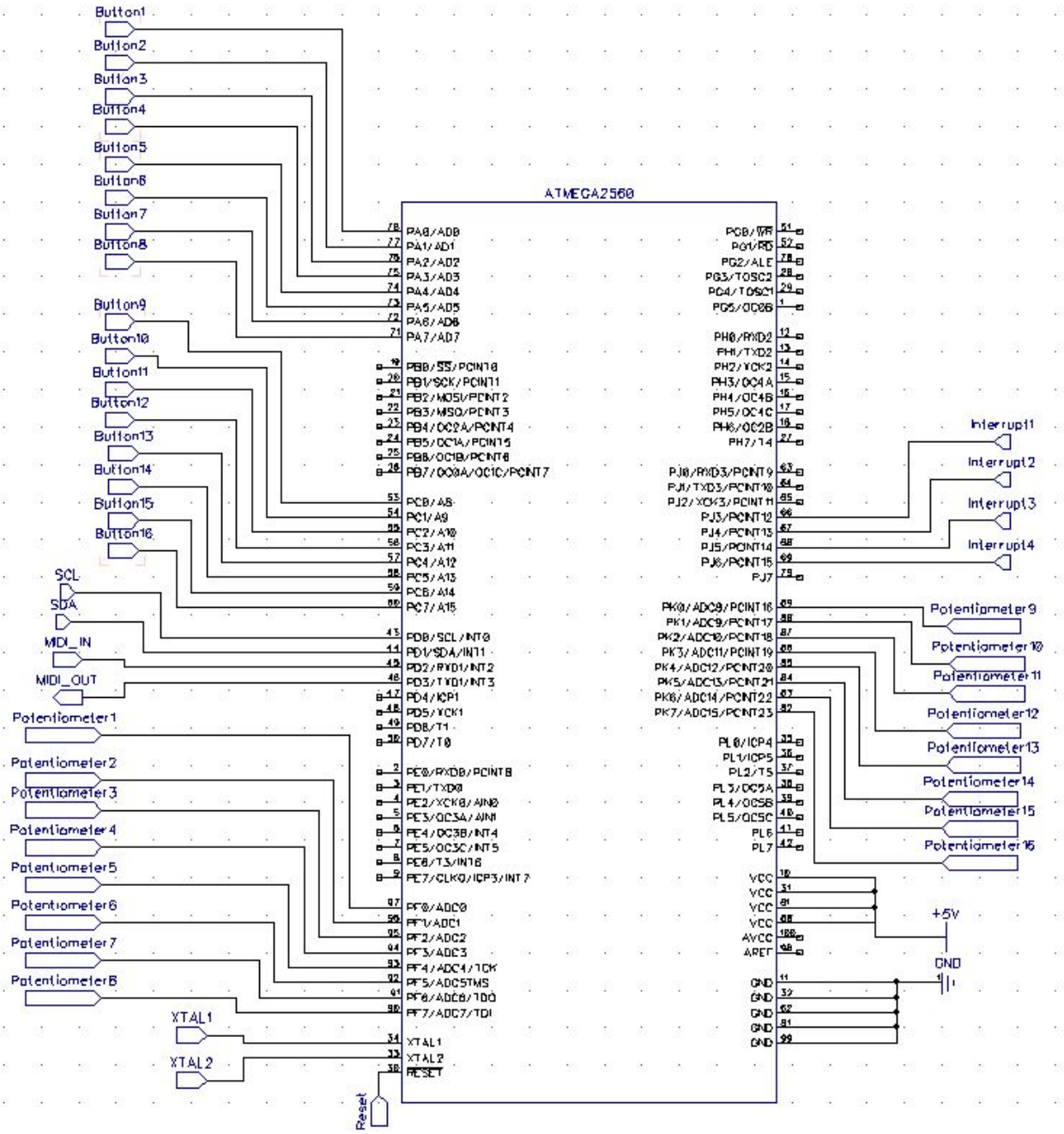


Figure 10 : Microcontroller Module

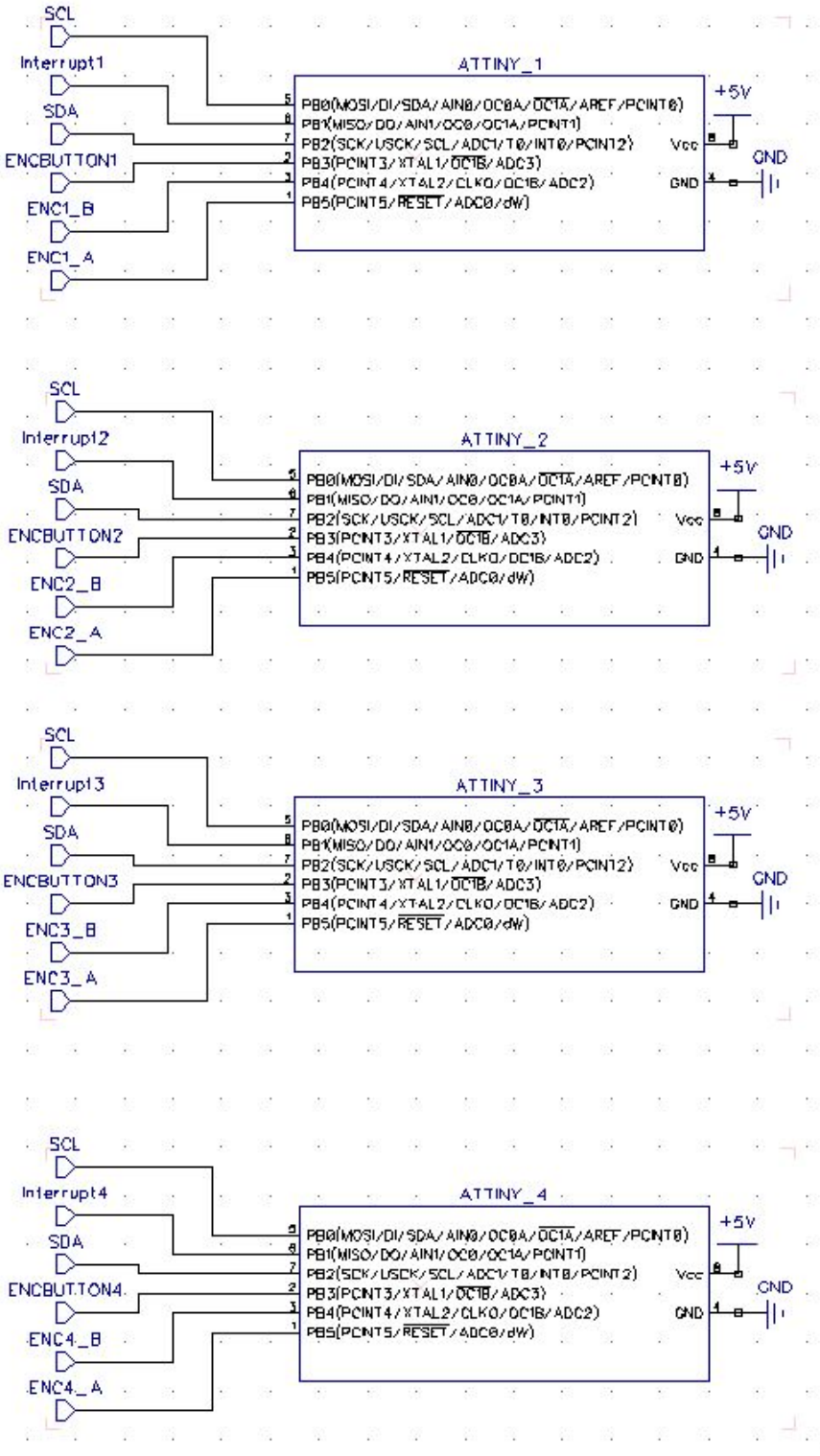


Figure 11: Encoder with connection to 4 dedicated microcontrollers.

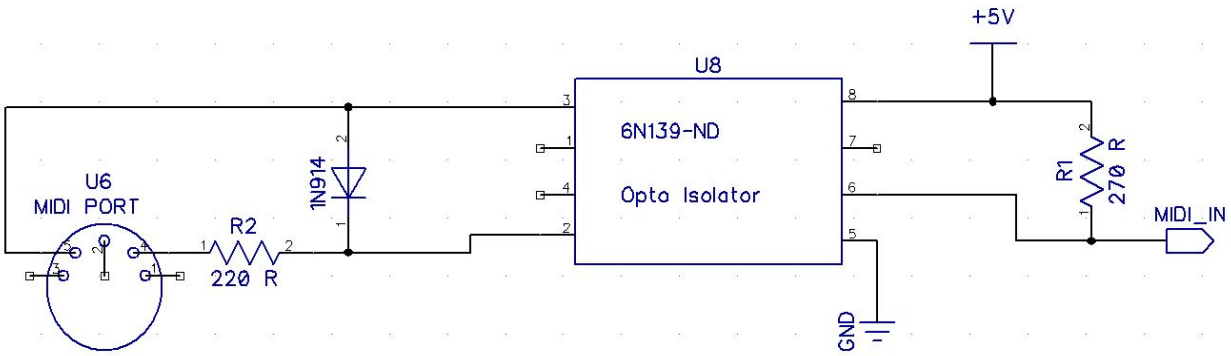


Figure 12: MIDI in connection

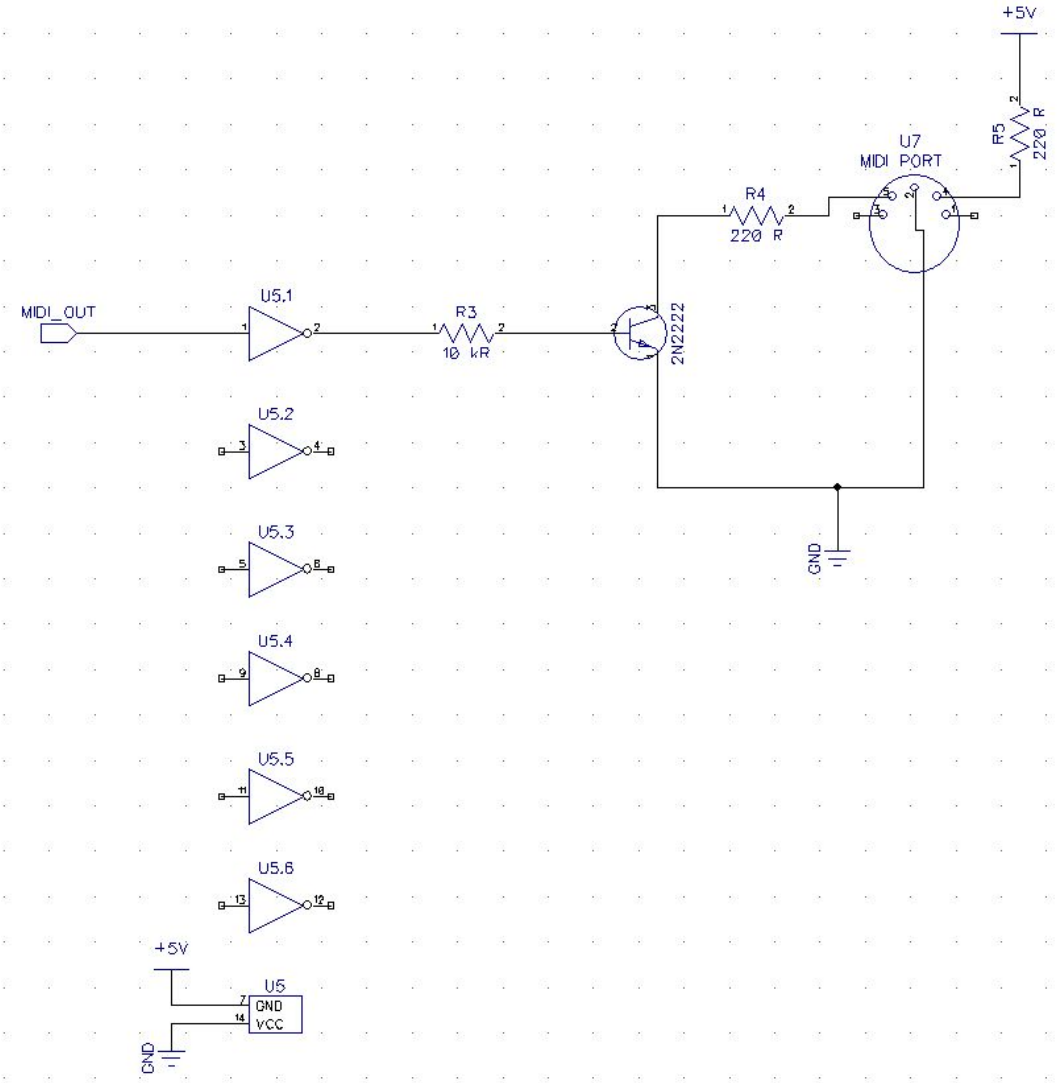


Figure 13: MIDI out connection

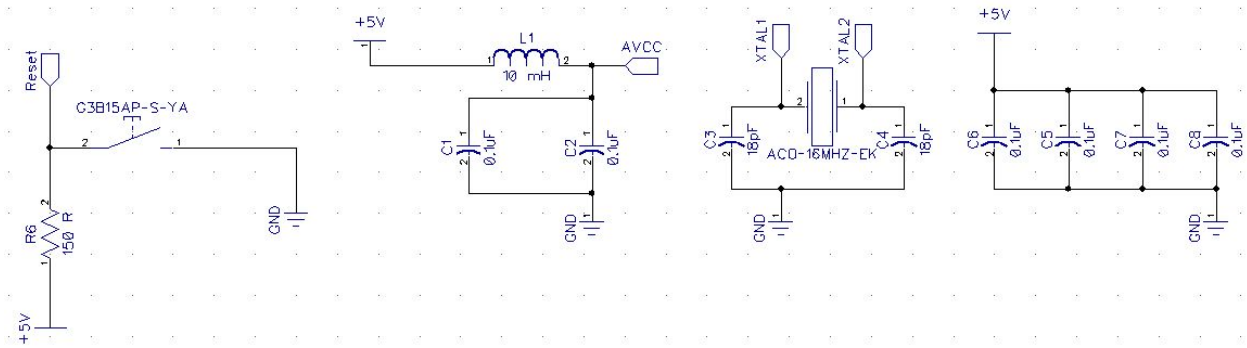


Figure 14: crystal oscillator,LPF , capacitor bank, and reset circuitry.

Simulations

Input: Square wave **Output:** Inverted and delayed square wave

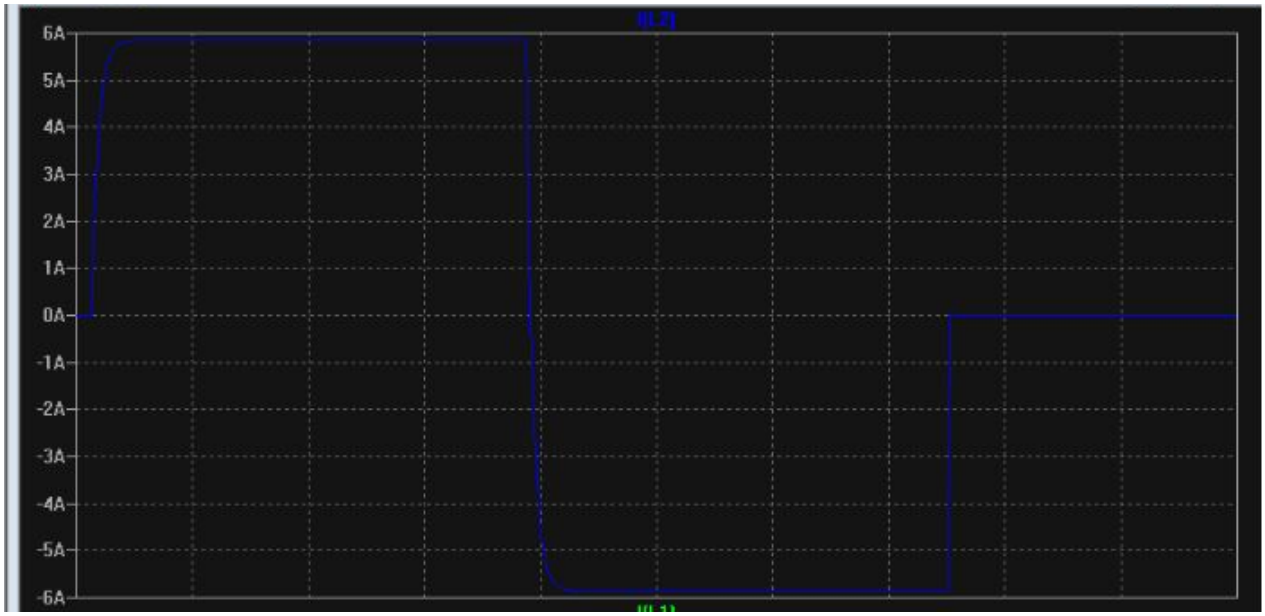


Figure 15: LTspice H-Bridge Simulation

Description: Current across inductor in **Figure 9**.

Flow Chart

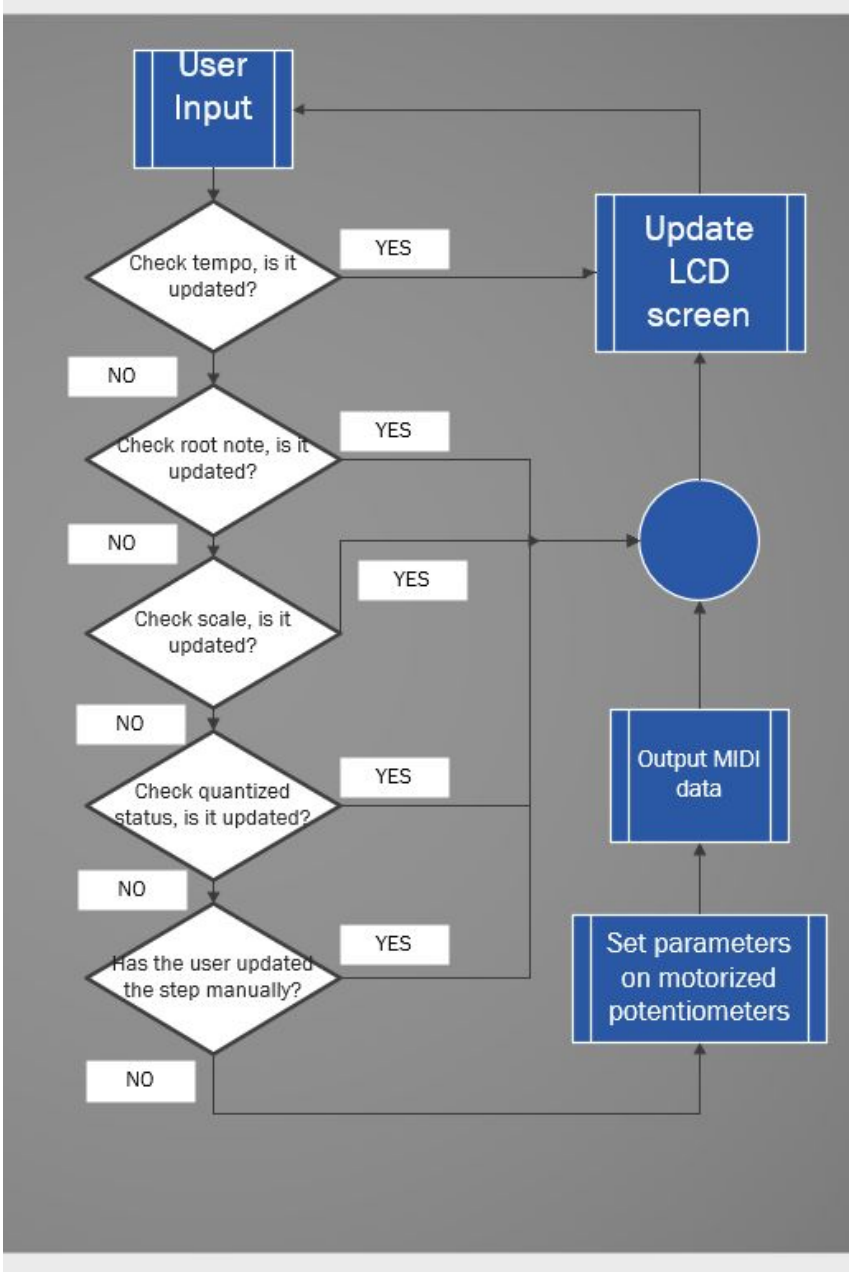


Figure 16: High level LCD Functionality Flowchart

Note: The above figure will most likely be implemented with software interrupts.

Mechanical Diagrams (CAD Drawings)

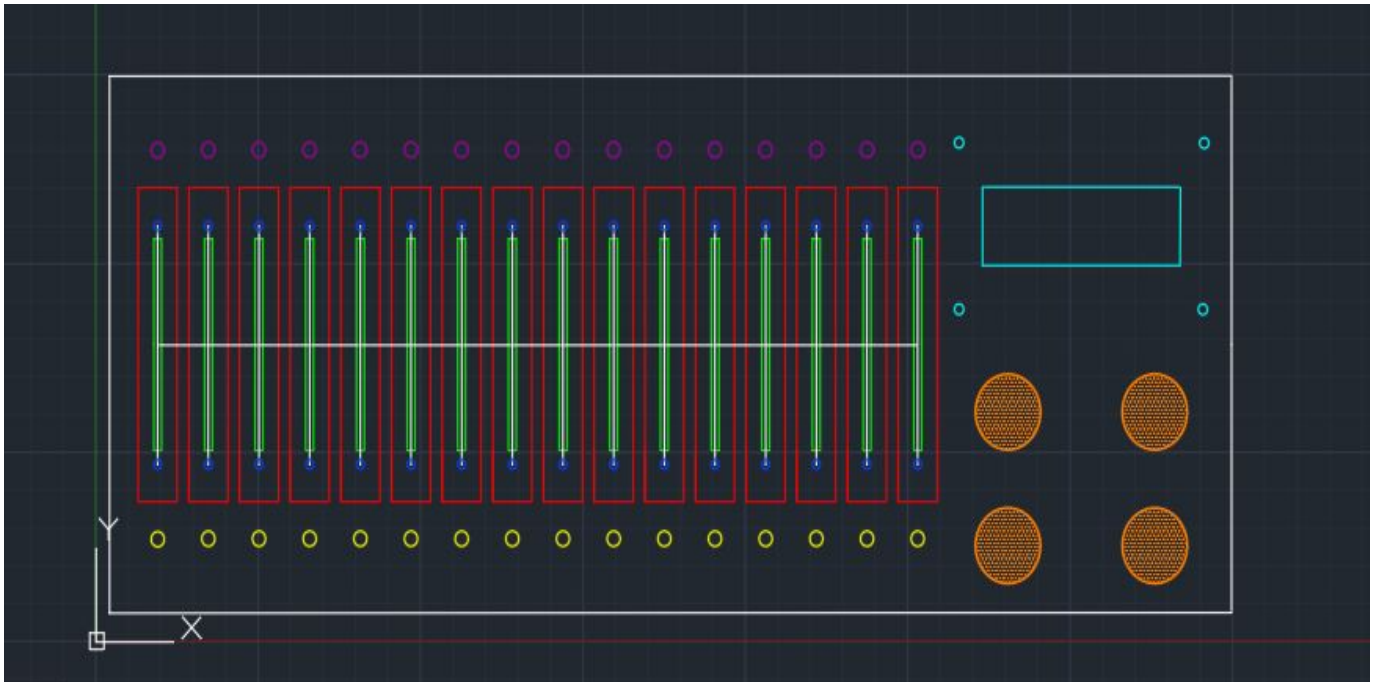


Figure 17: AutoCAD generated drawing of faceplate (2-D View)

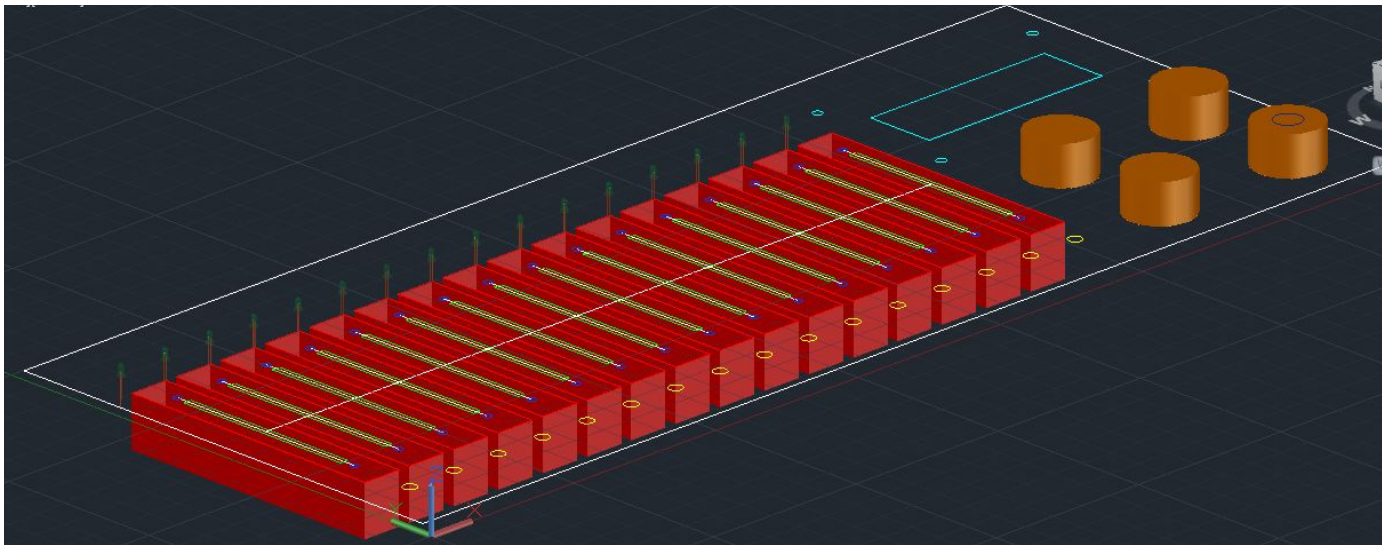
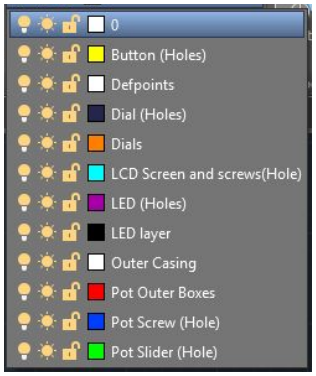


Figure 18: AutoCAD generated drawing of faceplate (3-D View)

LEGEND:



2.4 Tolerance Analysis

The block that poses the greatest risk to the successful completion of the project is the potentiometer board. Four, smaller sub boards with four potentiometers each will comprise the potentiometer module. The potentiometer boards will hold all the motorized potentiometers, selection buttons, display LEDs, as well as the Motor controller, and LED Driver.

The EMI (Electromagnetic Interference) or RFI (radio frequency interference) produced by the motors may also interfere with the ADC readings [11]. This EMI will need to be correctly detected and handled appropriately to ensure it does not cause the potentiometers to position wobble. This is caused by the EMI inducing an unwanted surplus of current through a connection or inside an individual IC. In practice, shielded cables are used to ensure the data that passes through the wire is unaffected. Back EMF caused by the motor will have noise shielding in the form of a low pass filter.

To determine if position wobble and EMI produced will pose a significant noise risk, we will use Maxwell's equations (1) - (4) shown below:

$\nabla \cdot \mathbf{D} = \rho$	(1)	Gauss' Law
$\nabla \cdot \mathbf{B} = 0$	(2)	Gauss' Law for magnetism
$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$	(3)	Faraday's Law
$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}$	(4)	Ampère-Maxwell Law

We also ran simulations to determine the real world voltage values of EMI. We used our single potentiometer test bench and applied 9 V to For an unshielded 1 K□ resistor placed directly next to the motor, a 2 mV spike in voltage was measured. This is shown in **Figure 19** below. Following that initial worst case scenario, we placed a piece of plastic in between the potentiometer motor and the resistor. This plastic was used to emulate what shielding effects a PCB plane would have towards mitigating the EMI. This effectively cut the EMI produced by the motor in half. **Figure 20** below shows the results of this test, a 1 mV spike in voltage. Finally we decided to model the effectiveness of a ground plane by placing a piece of aluminum foil in between a paper towel. We placed this in between the motor and resistor and noticed a complete

removal of the EMI. At most, in the third scenario, there is a 250 µV spike in voltage induced across the resistor.

Another tolerance to consider is that with the 9 V supply there are 1024 possible discrete potentiometer positions. Individual positions may vary by as little as 10 mV. Differentiating between user input and noise is also important. It is important because we need to know if the slider position change is desired or not. One reason we decided to use a higher voltage of 9 V is to lower the signal to noise ratio. Lower signal noise ratio may help mitigate any EMI issues as described above.

If the ADC outputs 12 bits we may only have 10 bits of useful data and 2 bits of noise. This loss of information will most likely be negligible and not noticeably affect the voltage.

Various voltage levels must be supplied to the PCBs and thus circuit protection must also be devised to protect both the user and the more difficult sourced components such as the motorized potentiometers. To protect both the user and the components, fuses will be inserted in series with all outputs from the main power source.

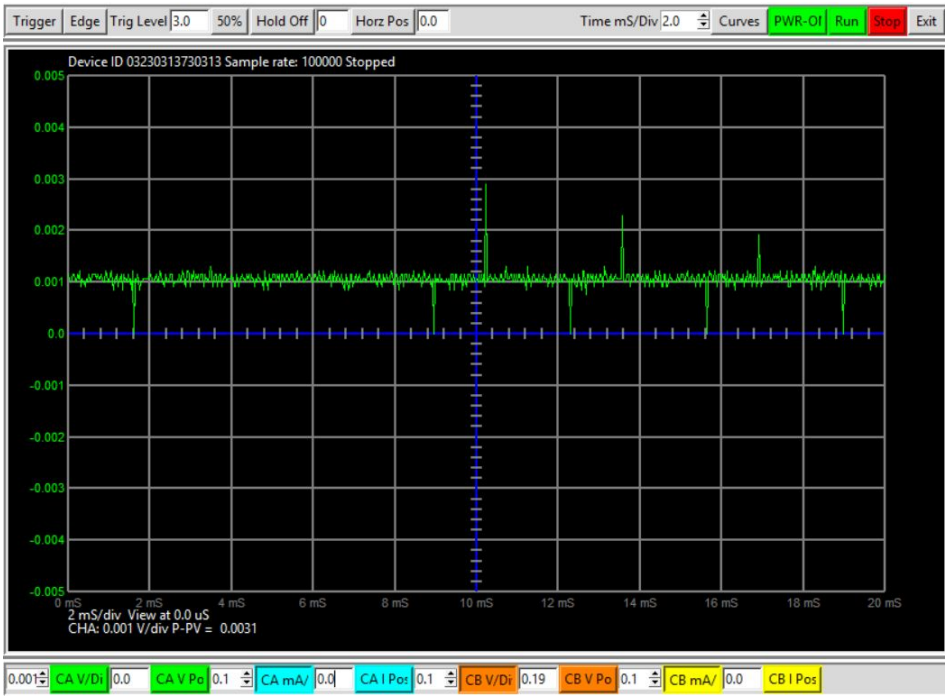


Figure 19: Model of induced
Description: Unshielded PCB test case with 2 mV spike in voltage.

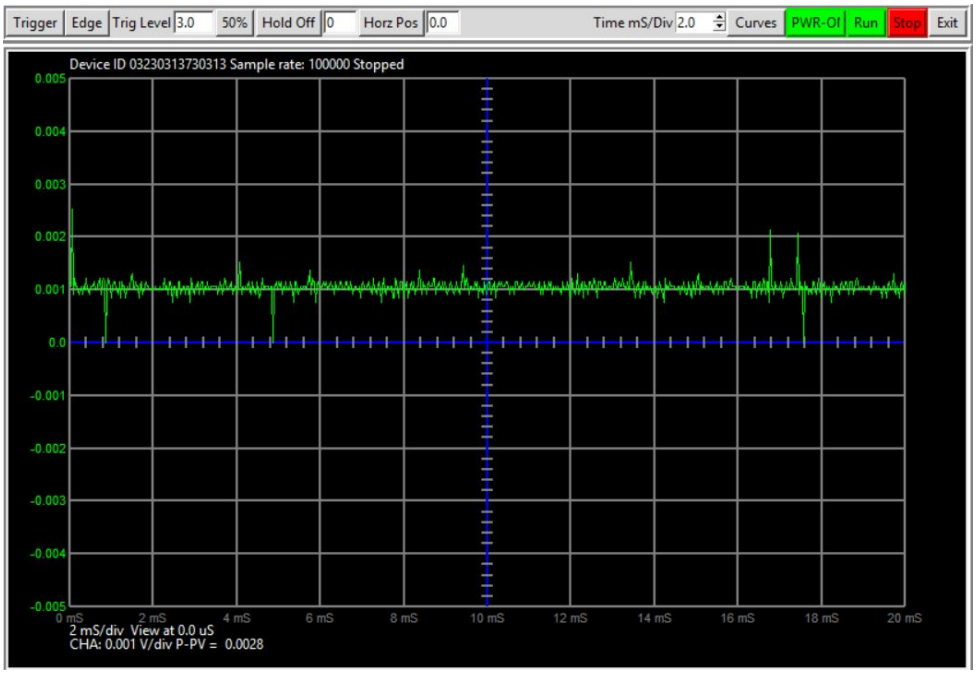


Figure 20: Shielded with plastic
Description: Unshielded PCB test case with 1 mV spike in voltage.

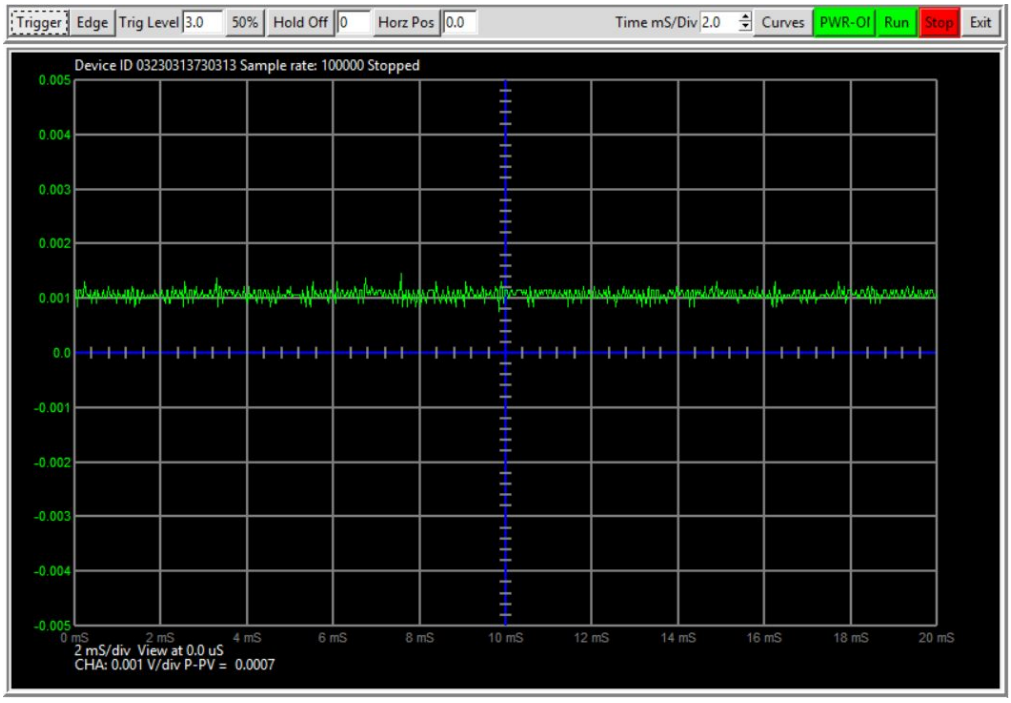


Figure 21: Shielded trial with conductive ground plane
Description: There was a complete reduction in induced spike.

3 Cost and Schedule

3.1 Cost Analysis

Labor

An entry level engineering job makes a starting wage of around $40 / hr or around $76,800 per year (pre-tax). With these numbers in mind, the project cost roughly: $25,600 (with project duration assumed consistent at 4 months.)

Parts

Note: These parts' prices were found on *digikey.com*. Most frequently, the part with the minimum cost was selected.

-The current estimated cost and bulk cost for our prototype, not including the physical casing or PCB itself are listed below:

INDIVIDUAL COMPONENT COST TABLE:			
PART:	**MODEL:**	**QUANTITY:**	**COST:**
Microcontroller	ATMEGA 2560	1	$12.35 / unit
Rotary Encoders	PEC11R-4220K-S0024	6	$ 1.73000 / unit
LCD Module	Hitachi HD44780 20x4	1	$17.95000 / unit
I2C to Parallel Adapter	NXP PCF8574AT	1	$3.50 / unit
5 V Linear Regulator	LM7805	5	$1.58000 / unit
9 V Linear Regulator	L7809CV	17	$1.65000 / unit
16.00 MHz Crystal Oscillator	ACO-16.000MHZ-EK	3	$1.91000 / unit
MIDI Connectors	SDS-50J	4	$1.96000 / unit
Toshiba H-Bridge	TB6612FNG	10	$ 2.11000 / unit
PWM 16- Channel LED Driver	NXP PCA9685PW	10	$ 2.26 / unit

RGB LED	LED RGB DIFFUSED T-1 3/4 T/H	25	$2.09000 / unit
4-Pin JST Connectors	B01DUC1S7S	20	$7.99 / 20
2-Pin JST Connectors	AT383-6	20	$1.89 / unit
Motorized Potentiometers	Yamaha AW4416	16	$24.20 / unit
16 Pin JTAG Connector for I/O	DC3-16P	20	$0.20 / unit
Fuses	BK1/TDC10-250-R	10	$1.69000 / unit
Fuse Holder	D0347RA	2	$9.24000 / unit
Break Away Male Header for Power for Motor	Generic	20	$0.75 / unit
Ribbon Connectors	FC-16P	10	$0.57 / unit
4u Flat Aluminum Blank Panel **(Reach Goal)**	N / A	1	$37.95 / unit
_____	_____	_____	**Total:** $ 879.88 **No Panel:** $ 841.93

BULK COMPONENT COST TABLE:			
PART:	**MODEL:**	**QUANTITY:**	**COST:**
Microcontroller	ATMEGA 2560	1	$10.24855 / 1000
Rotary Encoders	PEC11R-4220K-S0024	6	$ 1.73000 / unit
LCD	HD44780 4x20	1	$17.95000 / unit
I2C to Parallel Adapter	NXP PCF8574AT	1	$ 0.65445 / 1500
5 V Linear Regulator	LM7805	5	$0.91176 / 1000
9 V Linear Regulator	L7809CV	17	$0.94212 / 1000

16.00 MHz Crystal Oscillator	ACO-16.000MHZ-EK	3	$1.91000 / unit
MIDI Connectors	SDS-50J	4	$1.13850 / 1000
Toshiba H-Bridge	TB6612FNG	10	$ 0.89100 / 1000
PWM 16- Channel LED Driver	NXP PCA9685PW	10	$ 1.03626 / unit
5 mm RGB LED	LED RGB DIFFUSED T-1 3/4 T/H	25	$2.09000 / unit
4-Pin JST Connectors	B01DUC1S7S	20	$7.99 / 20
2-Pin JST Connectors	AT383-6	20	$0.59 / (12 or more)
Motorized Potentiometers	RS60N11M9	16	$24.20 / unit
16 Pin JTAG Connector for IO	DC3-16P	50	$9.80 / 50
Fuses	BK1/TDC10-250-R	10	$1.69000 / unit
Fuse Holder	D0347RA	2	$9.24000 / unit
Break Away Male Header for power for Motor	Generic	500	$0.45 / unit
Ribbon Connectors	FC-16P	10	$0.57 / unit
4u Flat Aluminum Blank Panel (Reach Goal)	N / A	1	$37.95 / unit
_____	_____	_____	**Total:** $373.07 **No Panel:** $335.12

Grand Totals (Note: Labor is per person)	
Grand Total = Labor + Cost	**Bulk Grand Total = Labor + Bulk Cost**
$ 26,441.93 (No Aluminum Panel)	$ 25,935.12 (No Aluminum Panel)
$ 26,479.88 (Aluminum Panel)	$ 25,973.07 (Aluminum Panel)

3.2 Schedule

WEEK:	Nathan:	Devin:	Martin:
Week of 2/18	1. Design Document Check - 2/19 2. Begin initial potentiometer board PCB diagram/design in Eagle 3. Start soldering assignment 4. Plan design review presentation details 5. Begin software design/debugging for 1 test potentiometer	1. Design Document Check - 2/19 2. Start soldering assignment 3. Begin initial potentiometer board PCB diagram/design in Eagle 4. Plan design review presentation details 5. Begin software design/debugging for 1 test potentiometer	1. Design Document Check - 2/19 2. Start soldering assignment 3. Begin software design/debugging for 1 test potentiometer 4. Microcontroller breakout board design 5. Plan design review presentation details
Week of 2/25	1. Design Review - 2/27 2. Finish potentiometer board PCB diagram/design in Eagle	1. Design Review - 2/27 2. Finish potentiometer board PCB diagram/design in Eagle	1. Design Review - 2/27 2. Continue work on microcontroller circuit
Week of 3/4	1. Teamwork evaluation 1 - 3/4 2. Soldering assignment due - 3/8 3. Finish microcontroller board PCB diagram/design in Eagle 4. Begin software design/debugging for microcontroller 5. Finish initial PCB	1. First round of PCBway orders must pass audit - 3/14 2. Submit final revisions to machine shop - 3/15 3. Begin software design/debugging for LCD and MIDI data	1. Teamwork evaluation 1 - 3/4 2. Soldering assignment due - 3/8 3. Finish microcontroller board PCB diagram/design in Eagle 4. Begin software design/debugging for microcontroller 5. Finish initial PCB

	designs and (microcontroller and potentiometer board) for PCBway	4. Improve/fix initial PCB design	designs and (microcontroller and potentiometer board) for PCBway
Week of 3/11	1. First round of PCBway orders must pass audit - 3/14 2. Submit final revisions to machine shop - 3/15 3. Begin software design/debugging for LCD and MIDI data 4. Improve/fix initial PCB design	1. First round of PCBway orders must pass audit - 3/14 2. Submit final revisions to machine shop - 3/15 3. Begin software design/debugging for LCD and MIDI data 4. Improve/fix initial PCB design	1. First round of PCBway orders must pass audit - 3/14 2. Submit final revisions to machine shop - 3/15 3. Begin software design/debugging for LCD and MIDI data 4. Improve/fix initial PCB design
Week of 3/18	1. Finish final PCB design 2. Finish software design/debugging for LCD and MIDI data	1. Finish final PCB design 2. Finish software design/debugging for LCD and MIDI data	1. Finish final PCB design 2. Finish software design/debugging for LCD and MIDI data
Week of 3/25	1. Individual progress reports due - 3/25 2. Final Round PCBway orders must pass audit - 3/28 3. Prepare for mock demo	1. Individual progress reports due - 3/25 2. Final Round PCBway orders must pass audit - 3/28 3. Prepare for mock demo	1. Individual progress reports due - 3/25 2. Final Round PCBway orders must pass audit - 3/28 3. Prepare for mock demo
Week of 4/1	1. Mock demo 2. Debug/improve all software applications	1. Mock demo 2. Debug/improve all software	1. Mock demo 2. Debug/improve all software applications

		applications	
Week of 4/8	1. Prepare for mock presentation 2. Add “reach goal(s)” hardware and software functionality if possible	1. Prepare for mock presentation 2. Add “reach goal(s)” hardware and software functionality if possible	1. Prepare for mock presentation 2. Add “reach goal(s)” hardware and software functionality if possible
Week of 4/15	1. Prepare for mock presentation 2. Prepare final paper document	1. Prepare for mock presentation 2. Prepare final paper document	1. Prepare for mock presentation 2. Prepare final paper document
Week of 4/22	1. Mock presentation 2. Prepare for final presentation 3. Prepare final paper document	1. Mock presentation 2. Prepare for final presentation 3. Prepare final paper document	1. Mock presentation 2. Prepare for final presentation 3. Prepare final paper document
Week of 4/29	1. Final presentation 2. Final paper due - 5/1 3. Lab notebook due - 5/2 4. Teamwork evaluation II due - 5/2	1. Final presentation 2. Final paper due - 5/1 3. Lab notebook due - 5/2 4. Teamwork evaluation II due - 5/2	1. Final presentation 2. Final paper due - 5/1 3. Lab notebook due - 5/2 4. Teamwork evaluation II due - 5/2

4 Discussion of Ethics and Safety

IEEE's code of ethics article 1 is, *"*to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment." [12] Our group also strives to uphold IEEE's article 1 safety standards throughout our project. Synthesizers/sequencers generally operate at 15 V [13], these voltage levels are considered a safe range for users to operate the device at. Our synthesizer implementation will run at an even lower voltage, with a maximum operation voltage of around +10 V. Even with the device operating at a low voltage level, there is always a potential for risk. To mitigate the risk of electrical shock, the device inputs and outputs, and even some passive elements will be clearly labeled. An on/off switch *may* also be built in to the device with a led response to keep users informed of the sequencers current status.

Additionally, non-electrical issues taken into consideration are users jamming their fingers into the slider holes, sharp edges, or otherwise rough use of the device. The proposed physical faceplate will be designed such that the holes are not large enough for a user to insert their fingers into. Sharp edges will either be noted by a warning label on the front of the faceplate or the edges will simply be ground down so the edges feel smooth. In following with Underwriter Laboratories, " We will create and maintain environmental, health and safety (EHS) work practices and secure work environments that enable employees to work injury free." [14] Significant amounts of soldering will pose a safety risk for the team but we will handle it accordingly. In keeping with the Underwriter Laboratories statement above, safety precautions such as personal protective equipment, adequate ventilation, and lighting will always be available and in use during the assembly stages of this project. Furthermore cleanliness and proper workspace organization will lessen the risk of accidents.

The main ethical dilemma that may need to be considered is the way in which our device is used. The types of music, sounds, and melodies, a user chooses to produce can have an effect on a listener's behavior and emotions. Another safety hazard/ ethical dilemma would be if a user operates the MIDI controller sequencer with malicious or harmful intent or otherwise forcing the device to operate under abnormal conditions (e.g. running the motors too often or quickly, forcing potentiometers into place, etc.).

5 Citations

[1] “Was ‘Earliest Musical Instrument’ Just a Chewed-Up Bone?,” *National Geographic*, 15-Dec-2017. [Online]. Available: https://news.nationalgeographic.com/2015/03/150331-neanderthals-music-oldest-instrument-bones-flutes-archaeology-science/. [Accessed: 22-Feb-2019].

[2] J. Paterson, “Classical Music Periods,” *Hans Zimmer - overview of the film composer and his music*. [Online]. Available: https://www.mfiles.co.uk/classical-periods.htm. [Accessed: 22-Feb-2019].

[3] J. Paterson, “Composer Timelines for Classical Music Periods,” *Hans Zimmer - overview of the film composer and his music*. [Online]. Available: https://www.mfiles.co.uk/composer-timelines-classical-periods.htm. [Accessed: 22-Feb-2019].

[4] “The Gramophone | Articles and Essays | Emile Berliner and the Birth of the Recording Industry | Digital Collections | Library of Congress,” *Planning D-Day (April 2003) - Library of Congress Information Bulletin*. [Online]. Available: https://www.loc.gov/collections/emile-berliner/articles-and-essays/gramophone/. [Accessed: 22-Feb-2019].

[5] *American RadioWorks*. [Online]. Available: http://www.americanradioworks.org/segments/radio-the-internet-of-the-1930s/. [Accessed: 22-Feb-2019].

[6] “A Chronology of AM Radio Broadcasting 1900-1960,” *Earliest Known Uses of Some of the Words of Mathematics (S)*. [Online]. Available: http://jeff560.tripod.com/chrono1.html. [Accessed: 22-Feb-2019].

[7] M. Fabry, “First Recorded Sound: Scott, Edison and History of Invention,” *Time*, 01-May-2018. [Online]. Available: http://time.com/5084599/first-recorded-sound/. [Accessed: 22-Feb-2019].

[8] “How Was Musical Notation Invented? A Brief History | WQXR | New York's Classical Music Radio Station,” *WQXR*. [Online]. Available: https://www.wqxr.org/story/how-was-musical-notation-invented-brief-history/. [Accessed: 22-Feb-2019].

[9] “Evolving DAWs,” *The History Of Korg: Part 1* |, 01-Jan-2019. [Online]. Available: https://www.soundonsound.com/reviews/evolving-daws. [Accessed: 22-Feb-2019].

[10] J. G. Ganslle, “A Guide to Debouncing.” The Ganslle Group, Baltimore, 2004.

[11] Getz, Robin and Bob Moecker. “Understanding and Eliminating EMI in Microcontroller Applications.” Literature Number: SNOA382. Texas Instruments. August 1996. http://www.ti.com/lit/an/snoa382/snoa382.pdf

[12] “IEEE Code of Ethics,” *IEEE - Advancing Technology for Humanity*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 22-Feb-2019].

[13] “What is Voltage? | The Synthesizer Academy,” *synthesis tutorial*. [Online]. Available: http://synthesizeracademy.com/what-is-voltage/. [Accessed: 22-Feb-2019].

[14] “Ethics and Compliance,” *UL - Empowering Trust*. [Online]. Available: https://www.ul.com/aboutul/standards-of-business-conduct/. [Accessed: 25-Feb-2019].