Automatic Drum Tuner

Design Document

ECE 445 - Senior Design Project Laboratory

March 6, 2025

Team 70

Joseph Bacino

Max Wojtowicz

Jonathan Fejkiel

TA: Shengyan Liu

1. Introduction......3

1.1 Problem......3

1.2 Solution......3

1.3 Visual Aid......5

1.4 High Level Requirements......5

2. Design......6

2.1 Physical Design......6

2.2 Block Diagram......8

2.3 Functional Overview and Block Diagram Requirements......9

2.3.1 Power Subsystem......9

2.3.2 Striking Subsystem......10

2.3.3 Pitch Detection Subsystem......12

2.3.4 Tuning Subsystem......14

2.4 Component Selection and Cost Analysis......16

2.5 Tolerance Analysis......16

2.6 Schedule......16

3. Ethics and Safety......16

References......18

1. Introduction

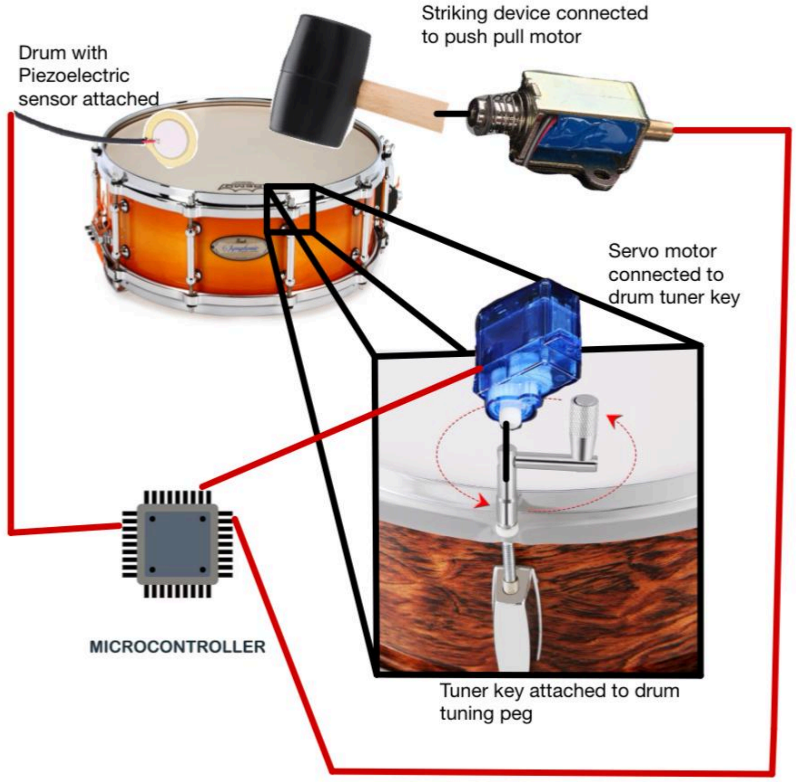
1.1 Problem

Playing instruments is a pastime enjoyed by millions of people across the world. A task that almost every musician must endure before playing is tuning their respective instrument. For many, this is done easily if they are of able body and have good pitch. However, turning lugs and listening for the right tune can be difficult if someone is weaker, such as a child or the elderly, or if they are inexperienced in hearing perfect pitch such as a beginner. Drum tuning requires precise hand movements and finger strength, which can be a challenge for individuals with arthritis or limited dexterity. Some of these disadvantaged people may be those that have physical disabilities or impairments. Music should be accessible to all people as it is a way humanity has expressed itself for thousands of years, however, many aspiring musicians are limited because instruments are not made with accessibility in mind. About 66% of Americans have played or know how to play an instrument (YouGov). This number shows how important music is to the world, and physical ability should not be a limiting factor in self expression. With our senior design project, we aim to solve a limiting problem in the accessibility of playing the drums for anyone who may need it. We want to make playing the drums an inclusive form of art for those of all different skill levels and ages.

1.2 Solution

The solution we propose is an automatic tuner for drums that will adjust the drumhead until the desired pitch is reached. The device will strike the drum with a push pull solenoid and determine its pitch at the current lug. As the drumhead is continuously struck at a constant interval, a piezoelectric sensor will capture the vibrations of the drum. This will be transmitted to the microcontroller, which will calculate the frequency of the drum based on the vibrations. LED's will display red if the drum is not tuned. Whether it determines if the drum is too low or high based on a preset pitch, the microcontroller will determine the direction a tuning motor will turn. A servo motor attached to the lug will rotate to tighten or loosen the drumhead. As this is occurring, the drum is still being struck and its pitch is being determined. Once the correct pitch is detected from the drum, the microcontroller will tell all subsystems to stop, and the LED's will become green. At this point, the user can remove the device and place it on the next lug. The device will be connected together as one unit. The device will clip onto the lip of the drum and the servo motor will fit into the lug of the drum. The push pull solenoid will be adjustable to raise or lower so that it strikes the drum and the correct height without damaging. The piezoelectric sensor will extend and rest onto the center of the drum.

1.3 Visual Aid



1.4 High Level Requirements

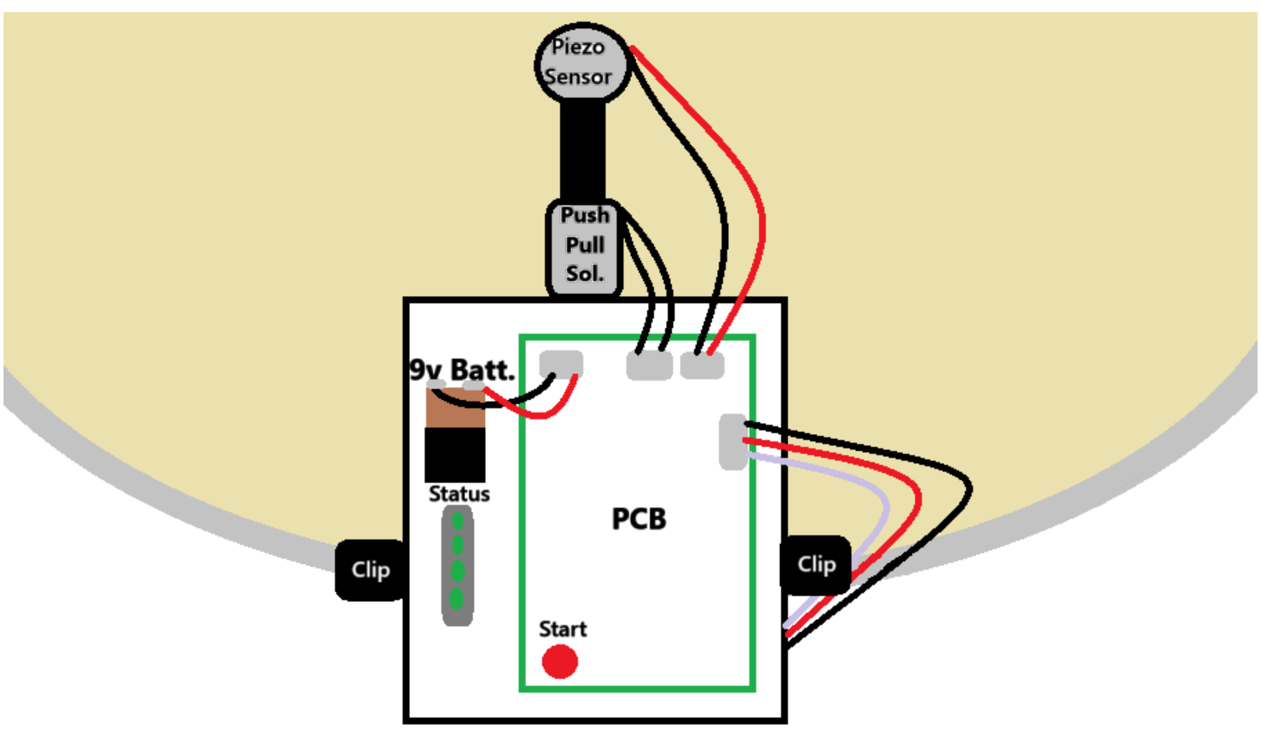
1. The first requirement for our project is that the mechanism strikes the drum correctly and for the same amount of time and intensity for each cycle as needed. It must be strong enough to create frequency that can be picked up from the piezoelectric sensors and repeated if more tuning is needed. The motor rod should make contact with the drumhead for less than 10 milliseconds to maximize resonance for the drum.
2. The tuning mechanism should be able to apply the correct amount of torque needed to turn the lug to reach the desired pitch/frequency. It should also operate at a constant slow speed of about 5 rotations per minute. We want to avoid any cases of the motor under or

over shooting to keep the entire process streamlined and efficient, without unnecessary corrective turns. The motor should be able to produce at most 8-10 in-lbs of torque to reach higher frequencies, but not go over as to not damage the drum hardware.

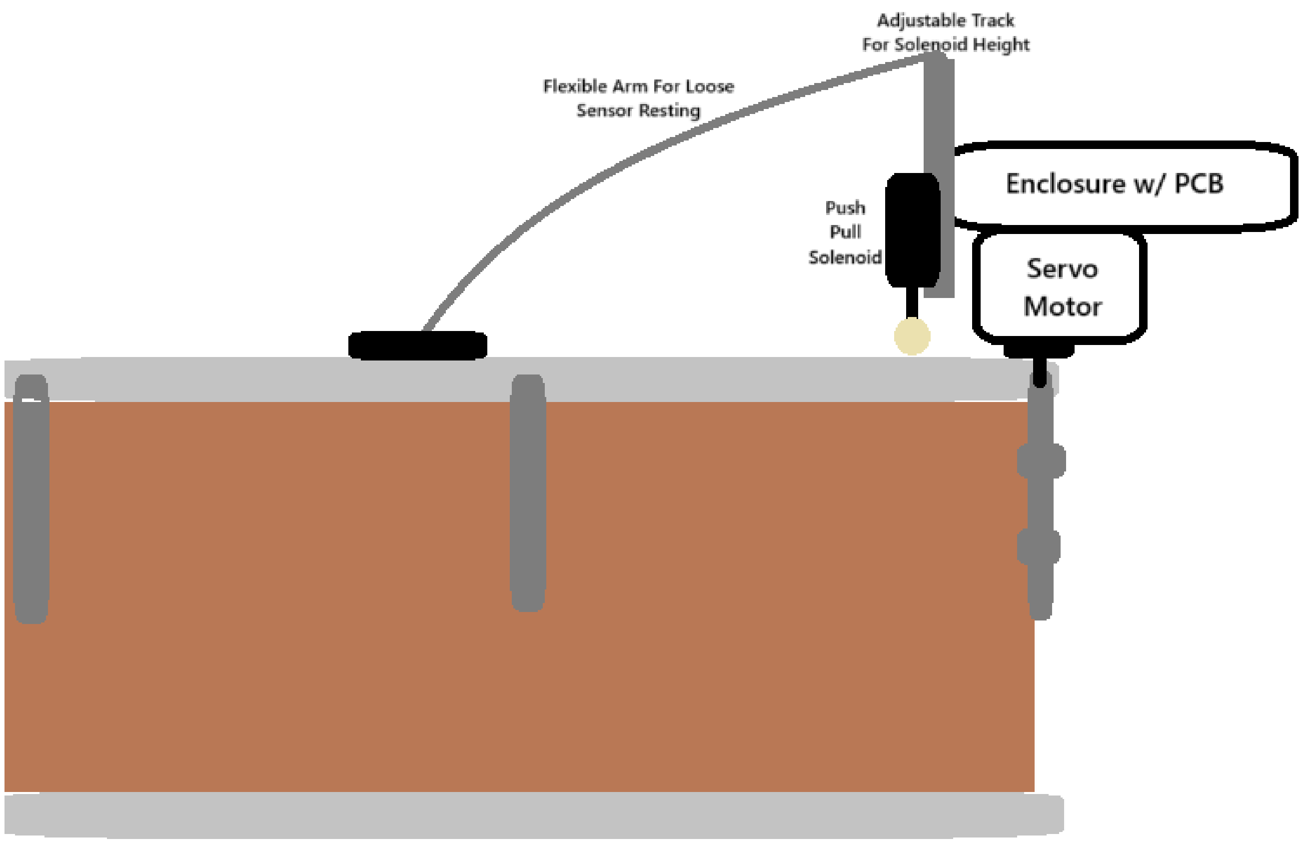
3. Finally, the pitch detection subsystem should accurately find the pitch of the drum after the hammer has struck. It should only pick up sound from after the hit of the drum and not any softer background audio. To consider the lug to be tuned, we have a tolerance of 5Hz for each pitch. Based on the frequency for notes in the range of the second octave, 5Hz is an adequate tolerance to create a unified sound among all the lugs of the drum.

2. Design

2.1 Physical Design

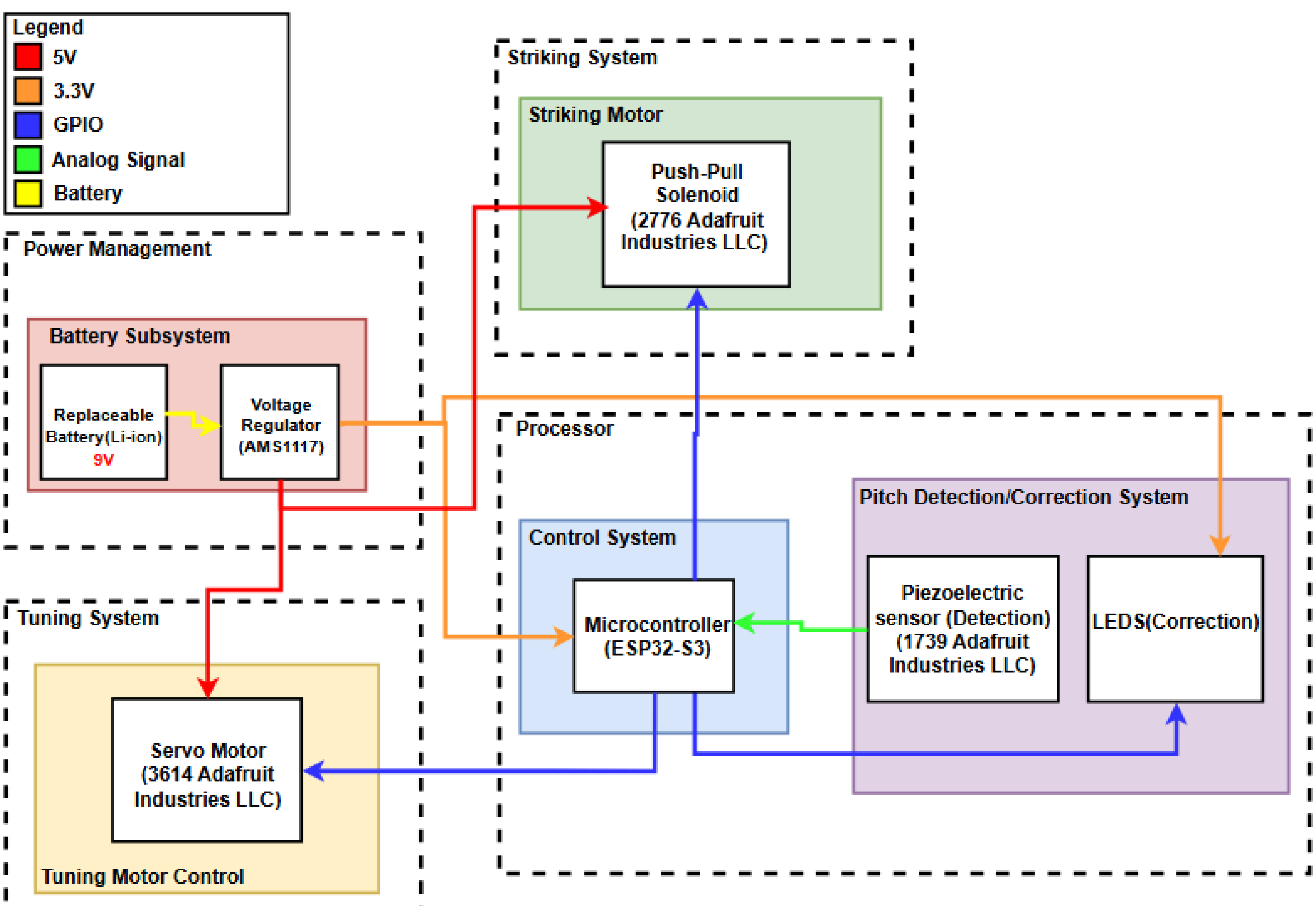


Above we have a top down view of our device, with the enclosure exposed to see everything inside. The PCB is in the center of the enclosure screwed in by mounting holes. The user powers the device by attaching a 9V battery to the battery harness which is connected to our PCB, and then is placed into the enclosure. At a resting position, the LED's will display a standby neutral color that neither indicates a correctly or incorrectly tuned drum. The entire device is held by clips on either side of the enclosure that hold onto the lip of the drumhead. Extending outward toward the middle of the drum we first have the push pull solenoid. Extending all the way to the middle of the drum we have the piezo sensor. For a side view we have the following figure.



With this side view, we can see how the device rests over each lug of the drum, one at a time. The servo is fitted to be placed directly on the lug, with the socket at the end of the motor having to fit perfectly on the lug. The weight of the device is allowed to rest on the drum without damaging it. The push pull solenoid near the edge of the drum is attached to a track. This track which moves vertically allows the solenoid to be raised to lowered so that the tip of the actuator touches the drumhead perfectly. The user must set this for their desired drum. This allows the actuator to strike the drum near the lug, with the same distance each time. Finally we have the piezo sensor. The sensor should hang freely, so it is attached to a flexible arm that swings allowing the piezo sensor to rest on the center of the drum.

2.2 Block Diagram



2.3 Functional Overview and Block Diagram Requirements

2.3.1 Power Subsystem

The power subsystem will be responsible for stepping down to and maintaining the various voltage levels that the other subsystems and devices require to run. The power subsystem must step down from 9V and continuously supply 6V ± 0.1V to the tuning motor system, supply 5V ± 0.1V to the striking motor, and also provide 3.3V ± 0.1V to the microcontroller and pitch correction LEDs. A LM7805 voltage regulator will step down 9V to 5V for powering the push-pull solenoid, with 100µF and 10µF capacitors included for ripple reduction. The AMS1117 voltage regulator will step down 9V to 3.3V to supply power to the ESP32 microcontroller, using 10µF and 1µF capacitors for ripple filtering. Additionally, a LM7806 voltage regulator will step down 9V to 6V for the servo motor, incorporating 1µF and 0.1µF capacitors for ripple reduction. This setup ensures stable voltage levels within the required tolerances.

Power Subsystem Requirements and Verification Table:

Requirements	Verification
The power subsystem must step down from 9V and provide 5V ± 0.1V for the striking motor.	Measure the 5V output with a multimeter under load to confirm it remains within ±0.1V tolerance.
The LM7805 regulator must use 100µF and 10µF capacitors for ripple reduction.	Verify capacitor values and placement in the circuit schematic and measure output ripple

	using an oscilloscope. Peak-to-peak ripple should be less than 100 mV.
The AMS1117 regulator must step down 9V to 3.3V ± 0.1V for the ESP32 and LEDs.	Measure the 3.3V output with a multimeter and confirm voltage stability within ±0.1V under normal operation.
The AMS1117 must use 10µF and 1µF capacitors for ripple filtering.	Inspect the circuit to confirm correct capacitor values and test ripple voltage with an oscilloscope. Peak-to-peak ripple should be less than 50 mV.
The LM7806 regulator must step down 9V to 6V ± 0.1V for the servo motor.	Measure the 6V output with a multimeter to ensure it remains within ±0.1V under typical load conditions.
The LM7806 must use 1µF and 0.1µF capacitors for ripple filtering.	Verify capacitor placement and use an oscilloscope to check for voltage ripple. Peak-to-peak ripple should be less than 100 mV.

2.3.2 Striking Subsystem

The striking subsystem is responsible for striking the drum head near each lug to create vibrations that can be detected by our piezo sensor. The striking subsystem must reliably receive a signal from the microcontroller to initiate a strike, ensuring precise timing for accurate pitch detection. It must also operate consistently when instructed, as failure to strike at the correct moment would lead to incorrect pitch readings from the detection subsystem. The system will use a 2776 Adafruit push-pull solenoid motor to execute the striking action. Additionally, the striker must not remain in contact with the drum for too long, as prolonged contact would alter the drum's pitch and cause faster sound dissipation. To ensure proper operation and longevity of the solenoid, the circuit must be designed to provide a quick voltage on-and-off signal without damaging the solenoid, incorporating necessary components such as flyback diodes and MOSFETs to prevent voltage spikes.

Striking Subsystem Requirements and Verification

Requirements	Verification
The striking subsystem must receive a signal from the microcontroller to initiate a strike.	Use an oscilloscope or logic analyzer to verify that the control signal from the microcontroller is correctly sent when a strike is required.
The motor must always operate when instructed by the microcontroller.	Send multiple strike signals and observe the solenoid's response to ensure consistent operation. We'll use an oscilloscope to measure the current draw to confirm activation.
The striker must not remain in contact with the drum for longer than 10 ms.	Measure the striker's contact time using either slow-motion video or our piezo sensor and adjust timing in the microcontroller's

	programming if necessary.
The circuit must ensure safe operation of the solenoid by allowing a quick on-and-off voltage signal without damage.	We will use an oscilloscope to detect voltage spikes and to measure how quickly the solenoid demagnetizes. Turning off the solenoid quickly after it turns on could cause voltage spikes that could damage itself or other parts of the circuit. Depending on these voltage spikes, we'll adjust our flyback diode setup to one that can handle the voltage spikes and can discharge them quickly so the solenoid is ready to operate sooner. We'll also monitor the solenoids temperature to ensure safe usage.

2.3.3 Pitch Detection Subsystem

The pitch detection subsystem is responsible for translating the time-domain signals from the piezo sensor into frequency-domain information. This will be done using the FFT algorithm through micropython on our ESP32-S3. Micropython has several different efficient FFT modules, micropython-ulab has one that can complete a 1024 point FFT in 2 ms. Once we have the FFT, we can get the pitch from the last strike. We will compare the most dominant pitch from the FFT to our desired pitch, and this comparison will dictate if our tuning subsystem will either tighten or loosen the current lug the device is on. The ESP32-S3 must accurately identify the most prominent frequency or pitch from the 1739 Adafruit piezoelectric sensor readings. This sensor will be strategically placed in the center of the drum to maintain consistent detection as the device moves along the rim, ensuring accurate and consistent pitch readings across each run of tuning each lug. The microcontroller must correctly determine whether the detected pitch is higher or lower than the desired pitch and subsequently send the appropriate signals to the tuning system. Additionally, it must control LED indicators to provide real-time feedback to the user, red for too low, green for within tolerance, and blue for too high, ensuring clear progress tracking during the tuning process.

Pitch Detection Subsystem Requirements and Verification

Requirements	Verification
The microcontroller must correctly identify the most prominent frequency from the piezoelectric sensor readings.	We will use a pitch detecting app to compare the microcontrollers output to the expected values.
The sensor must be placed consistently around the drum for accurate pitch detection.	Our current plan is to keep the sensor in the center of the drum for consistent readings. If this doesn't work as desired since the sensor will be relatively far from where the solenoid strikes the drum, we will move it closer to the solenoid to get better readings. It will be standardized, e.g. 3-4 cm away from the striking point.
The microcontroller must determine whether	We will input test frequencies that are above,

the detected pitch is higher or lower than the desired pitch.	below, and within the target range, then confirm the microcontroller classifies them correctly.
The microcontroller must send the correct signals to the tuning system based on detected pitch.	We will monitor the control signals sent to the tuning system using an oscilloscope or logic analyzer to verify correctness. We can initially input test frequencies as an initial verification, and then use the actual piezo sensor signals after they've been FFT'd to verify further
The system must provide real-time LED feedback: Red = too low, Green = in range, Blue = too high.	We will manually test with various known test frequencies and confirm that the corresponding LED lights up correctly based on pitch deviation.

2.3.4 Tuning Subsystem

The tuning subsystem is responsible for either tightening or loosening the lugs depending on how the current pitch compares to the desired pitch. The microcontroller must translate pitch deviation signals into commands that the servo motor controller can recognize, ensuring proper tuning adjustments. The motor controller will then direct the servo motor to rotate clockwise or counterclockwise to tighten or loosen the drum lugs accordingly. The servo motor will be controlled via PWM through a GPIO port on the ESP32-S3. After the drum is struck and is resonating, the tuning subsystem must continuously process pitch correction signals and adjust the lugs in real-time until the desired pitch is achieved. The system will use a 3614 Adafruit servo motor to execute the tuning rotations effectively.

Tuning Subsystem Requirements and Verification

Requirements	Verification
The microcontroller must translate higher or lower pitch signals into commands recognizable by the servo motor controller.	We will monitor the output signals from the ESP32-S3 using an oscilloscope or logic analyzer to ensure correct PWM signal generation based on pitch translation from the FFT.
The motor controller must direct the servo to rotate clockwise to tighten and counterclockwise to loosen based on the microcontroller's signals.	We will send test signals to the motor controller and verify that the servo rotates in the correct direction for both tightening and loosening actions.
The tuning subsystem must continuously adjust the lugs as the drum is resonating after the drum head has been struck until the desired pitch is reached.	We will conduct tuning tests where the system automatically adjusts the lugs in real-time. Initially we will measure the final pitch with a frequency/pitch detection app to confirm

	proper tuning. As we develop the project, we will use the piezo sensor itself and our FFT to confirm the desired pitch is reached so our project won't need external resources to determine if the right pitch is reached.

2.4 Cost Analysis

Description		Manufacturer	Quantity	Price	Link
Servo motor		AdaFruit	1	$27.99	https://www.digikey.com/en/products/detail/adafruit-industries-llc/3614/7717299?s=N4IgjCBcoLQBxVAYygMwIYBsDOBTANCAPZQDa4ArAEwIC6AvvYVWSAMwBsYALCA0A
PiezoElectric Sensor		AdaFruit	5	$4.75	https://www.adafruit.com/product/1739
Push Pull Solenoid		AdaFruit	1	$4.95	https://www.adafruit.com/pr

					oduct/2776
Description		**Manufacturer**	**Quantity**	**Price**	**Link**
MCU Dev Board		Espressif Systems	1	16.53	ECE Supply Center , HiLetgo ESP-WROOM-32 ESP32 ESP-32S Development Board
6V Voltage Regulator		STMicroelectronics	1	.96	https://www.digikey.com/en/products/detail/stmicroelectronics/L7806ABD2T-TR/585696
PLA Filament for Enclosure		Elegoo	1	14.99	https://www.amazon.com/ELEGOO-Filament-Dimensional-Accuracy-Printers/dp/B0D421Q2Q2?source=ps-sl-shoppingads-lpcontext&ref_=fplfs&smid=A2WWHQ25ENKVJ1&gQT=1&th=1
			Parts Total	$70.17	

All other components are available in the ECE self service shop and do not need to be paid for and ordered. This includes other voltage regulators, passive components, USB port, and the microcontroller itself.

Name	Rate	Hours	Total (($/hour) x 2.5 x hours to complete)
Joey Bacino	40$	80	$8,000

Jonathan Fejkiel	40$	80	$8,000
Max Wojtowicz	40$	80	$8,000
Total Labor Cost			$24,000

Total Project Cost (Parts + Labor)	$24,070.17

2.5 Tolerance Analysis

In the design of the 'Automatic Drum Tuner,' maintaining precise voltage levels is essential for stable and optimal operation. The voltage regulator chip, which converts the 9V battery to 3.3V, 5V, and 6V, has an inherent output tolerance specified in its datasheet. For instance, the 3.3V output has a tolerance of ±0.1V, ensuring the voltage remains between 3.2V and 3.4V, while the 5V output is maintained within 4.9V to 5.1V, and the 6V is maintained within 5.9V to 6.1V. These specifications are as defined by the regulator's specifications from their datasheets. These tolerances account for variations in the regulator's internal components and external factors such as temperature and load fluctuations. Ensuring that the supplied voltages remain within these limits prevents underpowering, which could lead to unreliable performance, and overvoltage conditions, which could cause overheating or component damage.

2.6 Schedule

Week	Task	Team Member
March 9 - March 15	Integrate pitch detection for breadboard demo	Everyone

	Order servo to hex bit adapter + hex bit to drum key adapter	Max
	Revise PCB design	Everyone
	SECOND PCB ORDER 3/13	Everyone
March 16 - March 22	Research optimal solenoid operation with proper discharge diode circuit	Joey
	Model and 3d print first version of housing and attachment clips	Max
	Integrate servo motor control through our microcontroller	Joey
	Assemble and test second PCB	Everyone
March 23 - March 29	Test power subsystems for stability and voltage ripple	Jonathan
	Initial test of servo motor and ability to turn drum tuning lugs	Max
	Assemble and test second PCB	Everyone
March 30 - April 5	Model and print second version of housing and attachment clips	Max
	Revise PCB design	Everyone
	Connect servo motor control system and pitch detection system through microcontroller firmware	Joey and Jonathan
	THIRD PCB ORDER 3/31	Everyone
April 6 - April 12	Integrate solenoid operation through the microcontroller	Max
	Integrate pitch detection LEDs	Jonathan
	Assemble and test third PCB	Everyone
	Revise PCB design	Everyone
	FOURTH PCB ORDER 4/7	Everyone
April 13 - April 19	Integrate ‘start’ button and desired pitch input	Joey

	through initial drum strike	
	PCB assembly with all components	Max and Joey
	Integrate naive pitch detection and correction/tuning with all components	Jonathan and Joey
	Integrate ideal(continuous) pitch detection and correction with all components	Everyone
April 20 - April 26	Finalize housing and attachment clip design	Max
	Add vertical rail attachment to housing for vertical solenoid adjustments	Joey
	First complete test on completely tuning an entire drum	Everyone
	Bug fixes and quality of operating device adjustments	Max and Jonathan
April 27 - May 3	Finalize adjustments to PCB assembly	Joey and Max
	Final live test on completely tuning a drum	Everyone
	FINAL DEMO	Everyone

3. Ethics and Safety

Since our ‘Automatic Drum Tuner’ is a fairly simple device, it has minimal ethical and safety concerns. However, we will formally adhere to the IEEE Code of Ethics [1] and remain mindful of any potential risks throughout the development and testing process. We will also actively consider and implement feedback from our TA and professors to ensure that our device meets the highest safety and ethical standards. Every team member will be treated with respect, and open communication will be prioritized to address any concerns that may arise during development.

Our project design incorporates a 9-Volt Alkaline battery, which necessitates strict adherence to safety and regulatory standards to mitigate fire hazards and potential injuries associated with lithium-powered devices. We will ensure that the battery operates within a safe temperature range of 30°F to 100°F and take necessary precautions to prevent overcharging, short-circuiting, or exposure to moisture. Additional precautions include avoiding actions that could damage the battery, such as dropping, crushing, or puncturing the device, as well as ensuring proper disposal procedures for end-of-life batteries to minimize environmental impact. Overall we will be taking necessary precautions, and using a 9-Volt Alkaline Battery Safety Sheet [2] as general guidance with battery safety.

To further enhance safety, we will document all procedures necessary to handle the device properly. A lab safety document will be created to outline best practices for battery handling, electrical safety, and general lab safety protocols. This will ensure that all team members are aware of potential hazards and know how to respond to emergencies.

Although our device does not involve high-voltage components, or direct human/animal interfaces, we will conduct thorough testing to ensure it does not pose unforeseen risks. Additionally, if any safety concerns arise during the development or testing phases, we will reassess our design and implement mitigation strategies to protect both users and developers. By maintaining a proactive approach to safety and ethics, we can ensure that our project remains both responsible and reliable.

References

[1] *IEEE - IEEE Code of Ethics*, www.ieee.org/about/corporate/governance/p7-8.html. Accessed 13 Feb. 2025.

[2] GPI International Ltd. (2017). *Safety Data Sheet for Alkaline 9V battery (Model: 1604A)*. https://data.kleintools.com/sites/all/product_assets/documents/msds/klein/SDS-GP-battery.pdf