**Q-Tune: Beta Test Plan**

Yenessa Avalos-Maldonado, Timothy Chiles, William Trigg, Giovanni Cornejo, and Dakota Cappel

Github Link: <https://github.com/avalosy8/qtune>

**Alpha Test Results**

1. Powering/Turning servos
   1. After determining that we had enough available pins on the Adafruit board, we connected all six servo motors and used different power sources levels to find the proper voltage level we needed to power the motors. After testing, we concluded that 5V was enough to power the motors. We no longer use the default 3.3V on the Adafruit board in our circuitry and instead use the USB pin to obtain 5V.
2. Determining total power required
   1. After testing, we concluded that all components of our guitar tuner could be powered simultaneously via the 5V and 3.3V pins on the Adafruit board. The Adafruit board could successfully power its internal components, the LCD display, the piezo sensor’s circuitry, and at least 1 servo motor at a time via a USB power bank plugged into its USB-C port. We did not test turning more than 1 servo motor at a time as our project does not require this functionality. We determined that a USB power bank is a suitable power source for our final tuner design as it can make the whole tuner rechargeable and easily able to be turned on and off.
3. Testing 3D printed tuning attachment
   1. Once the tuning attachment design has been finalized and printed, we screwed the attachments onto each six servo motors and tested them against each tuning peg. We turned the motor in small increments and tested how much strength was needed to actually turn the tuning peg. After testing, we’ve concluded that the tuning attachment design was enough to properly turn the tuning pegs when the motors are signaled move. Further modification may be made for the tuning attachments, but the design is finalized for the most part.
4. Testing tuner for each string
   1. As we began testing the accuracy of our tuner as a whole, we determined that more circuitry was required to condition the signal from our piezo sensor (amplification, filtering, noise reduction, removal of DC offset). After implementing this new functionality, we encountered an issue where the second harmonic of each string was appearing more so than the fundamental frequency we were looking for in code. We added more checks in our code to account for this issue. When testing our tuner after these progressions, we achieved our first successful tuning of the low E string. Despite this breakthrough, more testing and tweaking of our tuning procedure needs to take place during the next stage of the project.

**Expected Behavior**

1. The tuner should be able to tune all guitar strings to within 1 Hz of the actual in-tune frequency 90% of the time.
2. The tuner should produce accurate results to within 2 Hz of the actual in-tune frequency while the guitar is placed in different environments (outdoors, indoors, with surrounding noise, etc.) and while the guitar is held in different positions 80% of the time.
3. The 3D-printed tuning device should attach to the motor with ease and be secure as to properly tune the guitar peg. Motion from tuning, and sudden motion from the guitar (within reason) should not dislocate or negatively affect the device

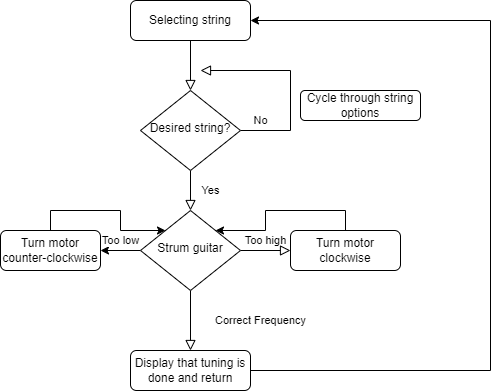


Figure 1: The overall flow of the expected behavior of the beta build.

For each specified expected behavior, a test procedure for it is further outlined below

**Test Procedures**

1. For each guitar string:
2. Manually turn the string’s tuning peg until the oscilloscope reports the string is at the desired starting frequency
3. Select the relevant string on the LCD and strum the string when prompted
4. Measure the frequency of the string with an oscilloscope once the tuner is finished tuning
5. Record the end frequency of the string. Repeat steps 1-3 until starting frequencies of in-tune frequency +/- 15 Hz, +/- 10 Hz, and +/- 5 Hz have been tested
6. Compile results, make adjustments to hardware/software, and repeat testing procedure until desired accuracy has been reached
7. For each environment/position:
8. Manually turn the string’s tuning peg until the oscilloscope reports the string is at the desired starting frequency
9. Hold the guitar in different positions and move into different environments
10. Select the relevant string on the LCD and strum the string when prompted
11. Measure the frequency of the string with an oscilloscope once the tuner is finished tuning
12. Record the end frequency of the string, the environment, and the position of the guitar. Repeat steps 1-3 until starting frequencies of in-tune frequency +/- 15 Hz, +/- 10 Hz, and +/- 5 Hz have been tested
13. Compile results, make adjustments to hardware/software, and repeat testing procedure until desired accuracy has been reached
14. Check that sudden motion does not dislocate or harm the 3D-printed device while tuning
15. Perform many tuning to observe the durability of the 3D-printed parts and ensure that there is minimum wear
16. Ensure that screws remain tight and unaffected by turns made with the tuner.
17. Manually turn each string’s tuning peg until all are out of tune
18. Tune all strings in one sitting to ensure that the motors do not affect each other