uTune: An Automated Guitar Tuning Solution

By: Christian Femrite, Joe Palasek, Peter Rymsza, Kevin Schneier, Ben Wdowik
University of Notre Dame Electrical Engineering Senior Design 2019

Project Overview

A practical issue faced by guitarists is the constant need to tune and re-tune the guitar. Firstly, guitar strings constantly go out of tune with increased play time, changes in temperature, and time spent unused. On top of this, many songs use tunings different than the standard *EADGBE*, such as *DADGBE* (known as "drop D"), *DADGAD* for fingerstyle songs, full downward shifts like *D#G#C#F#A#D#* or *DGCFAD*, and song-unique tunings like *CGCGGC* or *CFCFAC*.

The six tuning pegs located on the headstock of a guitar must be turned manually by the guitarist to adjust the pitch of each string, one by one. This tuning process takes significant time and effort in a guitarist's performance, day, and week due to how often the above situations call for new tuning.

This project automates the tuning process through the use of frequency analysis, a feedback-loop control system, and high-torque motors. uTune successfully solves the problems of the time and effort to tune faced by guitarists.

Solution

The final product, uTune, is an automated guitar tuner. The user has the option to interact with a touch screen or iPhone app to select their tuning scheme for the six strings. uTune then uses a microphone to sample the sound of each guitar string, takes the Fast Fourier Transform (FFT) of the input audio signal, and uses an embedded control algorithm to instruct a body of motors to turn each of the six tuning pegs, bringing the strings to the pre-selected tuning setting.

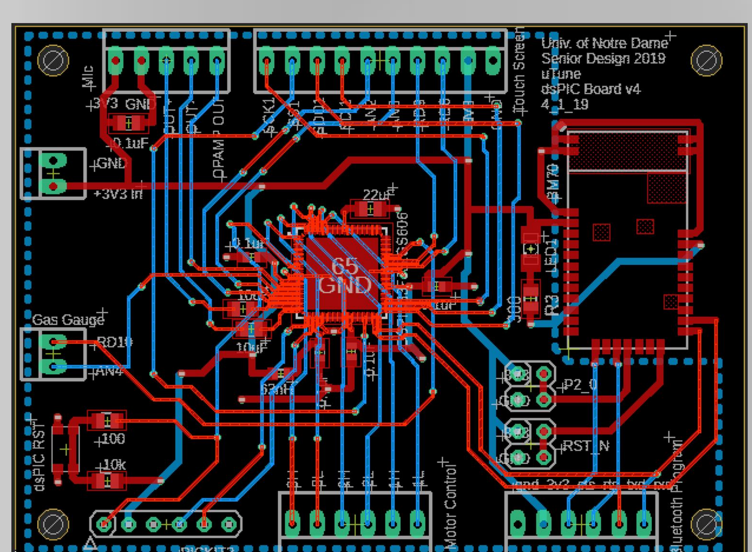


Figure 1: Main Board Layout

The final product implements three separate PCBs to allow for greater structural flexibility: (1) the main board with the microcontroller and Bluetooth module, (2) the power management board, and (3) the microphone board.

The design approach for the final product was to consider a variety of subsystems ranging from the microphone input signal through the motor control algorithm to provide end-to-end functionality.

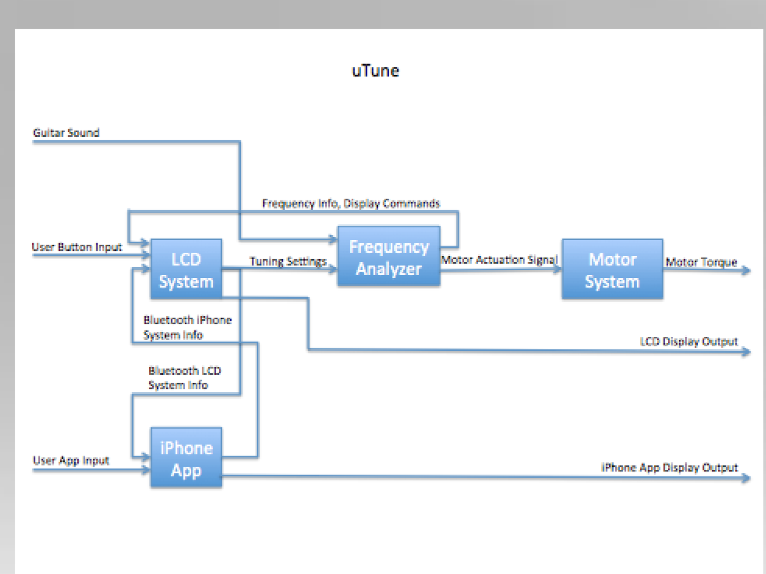
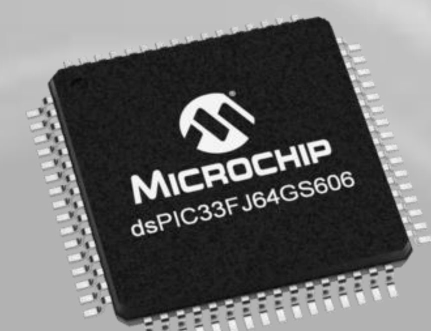


Figure 2: High-Level Block Diagram

Microcontroller

To select the best microcontroller for the uTune device, it was necessary to find one that could properly handle the battery management and mic inputs as well as interface with the six tuning peg motors, not to mention the need for heavy integrated DSP capabilities to run the FFT algorithm. Ultimately, the dsPIC33FJ64GS606 was selected with the following features:



- Code-Efficient (C and Assembly) Architecture
- Twelve SMPS PWM Outputs
- High speed ADC module, 10-bit resolution
- RAM Capacity
- Digital Signal Processing Capabilities

Figure 3: dsPIC Microcontroller

Device Apparatus

Size and weight were substantial challenges of the uTune design. One option would have been to replace the uTune guitar's tuning pegs with custom pegs, but that option proved to be mechanically unfeasible for the motors' structure. Instead, a 3D-printed housing and six motor chucks were custom-measured and designed for the uTune guitar head. This apparatus is also detachable so that the performer can choose to not always have to play with uTune attached, even though the structure does not interfere with normal playing.

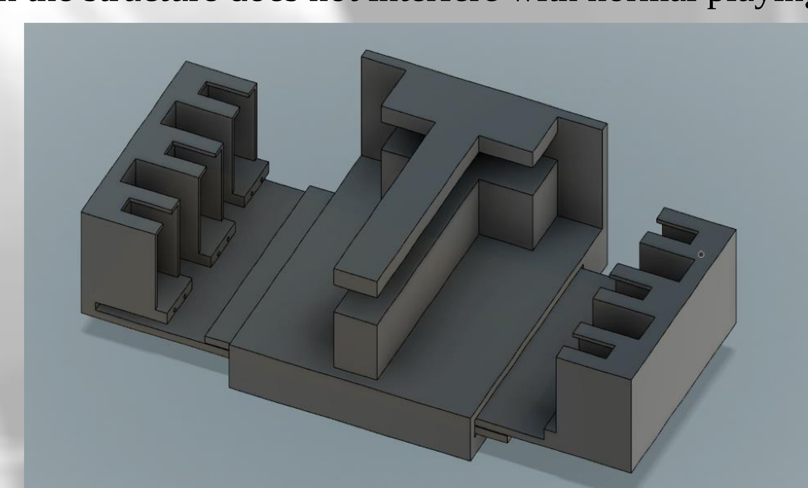


Figure 4: uTune Structural Apparatus

Microphone Board

A MEMS microphone was implemented on a separate PCB, mounted inside the guitar body. This board outputs an analog audio signal to the ADC input of the dsPIC microcontroller. The microphone outputs a (+) and (-) signal, so a differential amplifier was implemented to provide a single strong signal.

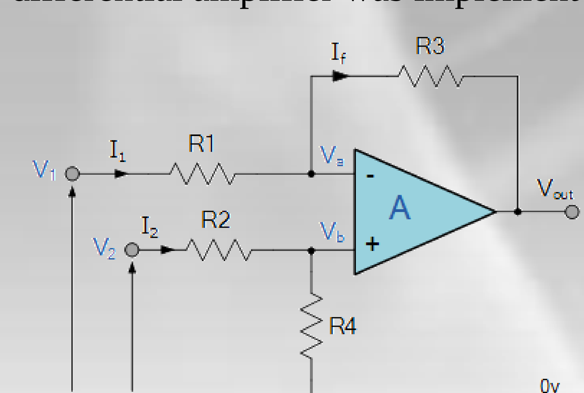


Figure 5: Differential Amplifier

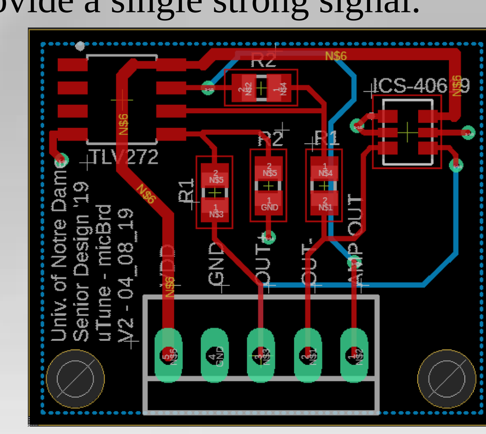


Figure 6: Mic Board Layout

Fast Fourier Transform (FFT) Algorithm

The dsPIC samples the output from the analog mic using its built-in ADC, and an algorithm creates an FFT from this data, with a resolution of about 1.30 Hz based on the sampling frequency and buffer size. This FFT is complicated by the presence of additional harmonic frequencies, or multiples of the fundamental string frequency. A processing algorithm locates the FFT peaks, analyzes for harmonics, and determines the fundamental string frequency. This information is then used to control the motor's turning such that the string frequency approaches its setpoint.

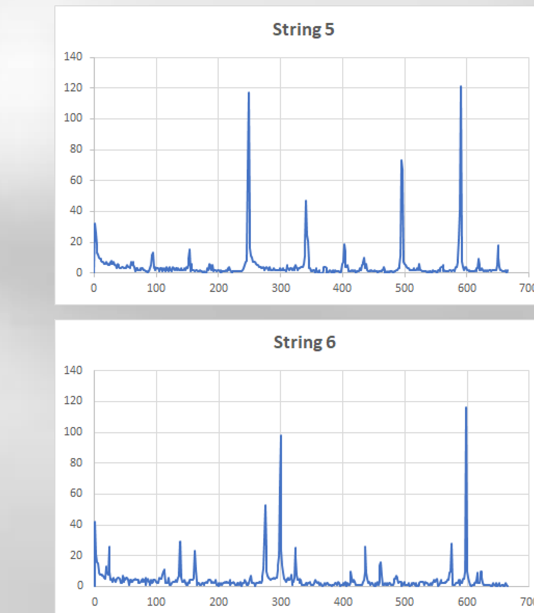


Figure 7: FFT Plots

iPhone App

The Xcode IDE was used to create an iPhone App using the Swift programming language. The app allows the user to create, select, and save tuning schemes. uTune may be activated to tune to any scheme within the range of physically acceptable string frequencies from the app using BLE GATT and transparent UART communication.

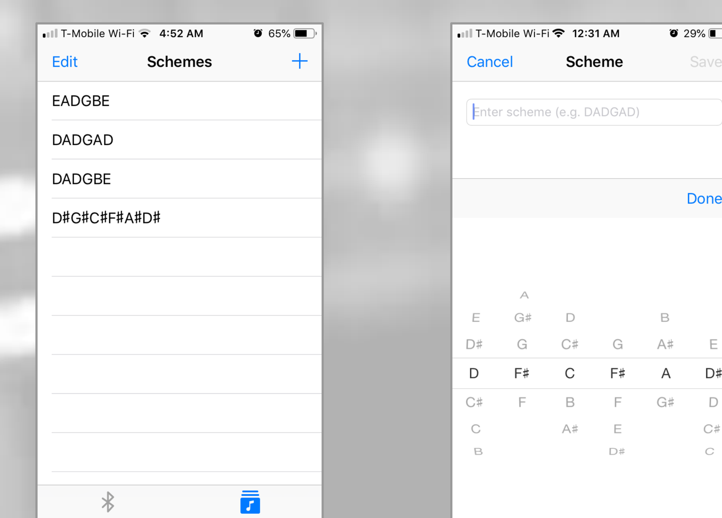


Figure 8: iPhone App Functionality

Touch Screen Display

The touch screen display provides user feedback on string tuning progress and status of each string. It interfaces with the dsPIC via a SPI connection and provides the uTune user a quicker method to change tunings which would prove useful in a live performance.

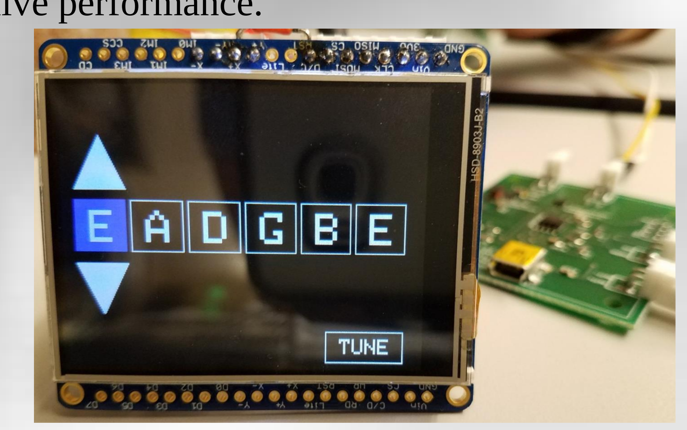


Figure 9: Touch Screen Interface

Power Subsystem

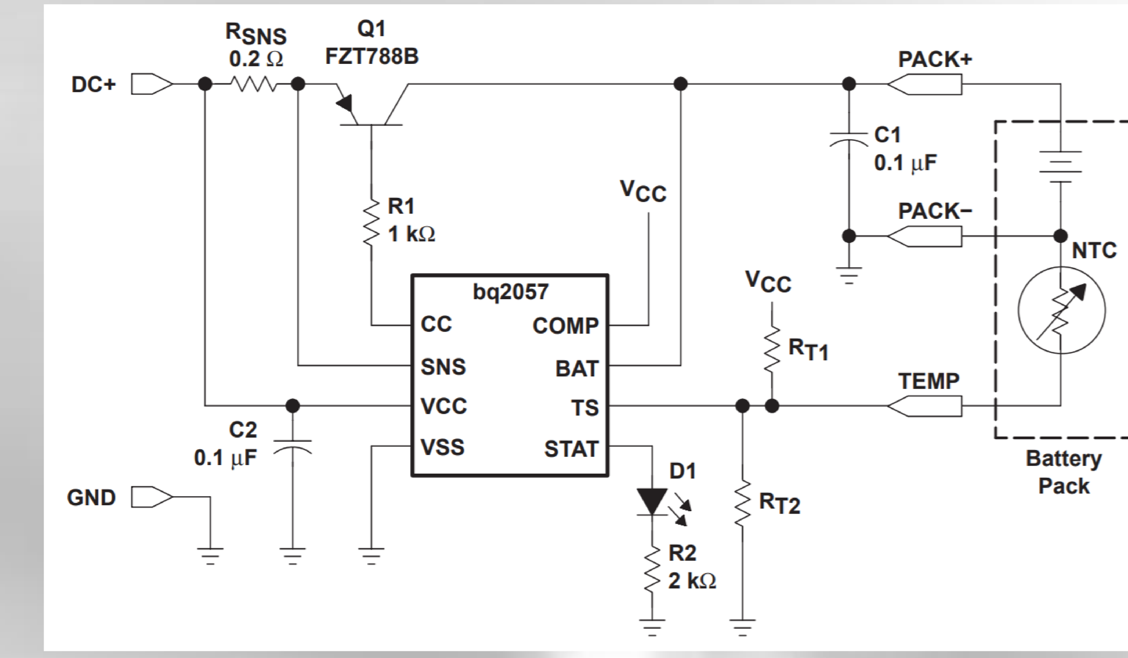


Figure 10: Power Subsystem Schematic

Given the portable nature of the device, a battery-based approach was the natural choice for the design. Li+ technology proved to be the most advanced battery technology (in terms of storage density), but that came with the challenge of a 3.7V source. The BQ2057 power management chip offered the flexibility of multiple packages and one- or two-cell charging options. It serves to control the USB-enabled charge sequence through the precharge, current regulation, and voltage regulation phases. A thermistor was also used as a method to stop charging in the event of overheating.

Results and Conclusions

- Created a final product capable of implementing a user-specified tuning scheme with end-to-end functionality
- Differential amplifier implementation was just as important as the initial selection of an audio broadband microphone
- iPhone app development has a lot of potential to create a fantastic user experience, but requires significant attention to detail
- Physical sampling/prototyping and component selection form a circular process that require time for design iteration
- Software implementation of theory/algorithm is always more intricate and time-consuming than it initially seems

Further Development: If uTune were to be mass-produced or retailed, it would need to have smaller motors and housing as well as be sold in a variety of shapes/sizes for the wide variety of guitars on the market.

Figure 11: Final Assembly