IoT Final Project Report

Name: Jack Power

Student No.: 20080169

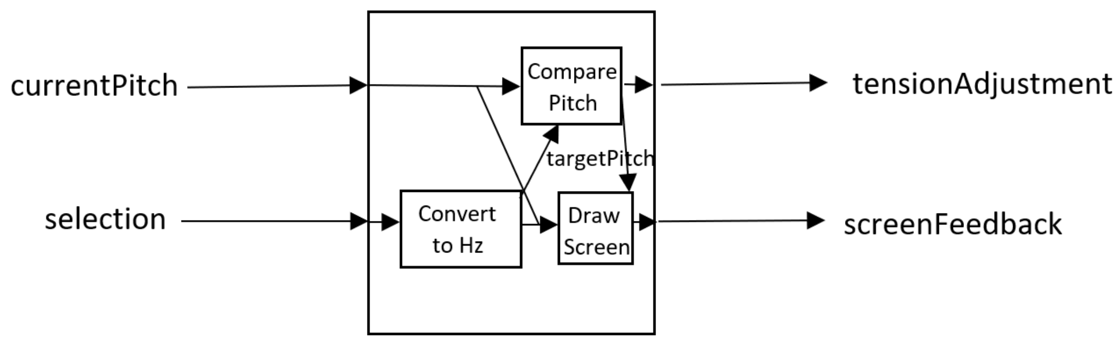
Project Title: Automatic Guitar Tuner.

Introduction: The purpose of this project was to design a device which would tune a stringed instrument automatically. Early ideas included variable motor speed, inducing vibration in a steel string using an electromagnetic field and a complex structure to mount the device. After much thought, refinement and making sure all goals were feasible, the final design forewent these features in favour of simplicity and effectiveness. The result of this is a device which, although certainly having room for future improvement, works satisfactorily well for the purpose it was designed.

Functional Specification:

The three fundamental subsystems of the device are **Convert to Hertz**, **Compare Pitch** and **Draw Screen**.

- **Convert to Hertz** takes the user's selection in a pitch class format (i.e. A_4) and converts it to its corresponding pitch (frequency) in Hertz. This makes use of the equal temperament tuning formula, $f_n = 440(\sqrt[12]{2})^n$.
- **Compare Pitch** compares the values of currentPitch, from the microphone, and targetPitch, from Convert to Hertz. If currentPitch is less than targetPitch, the tension must be increased. If currentPitch is greater than targetPitch, the tension must be decreased. If they are equal, the tension should not be adjusted.
- **Draw Screen** takes selection, currentPitch, targetPitch and tensionAdjustment and displays them to the user coherently.



Design:

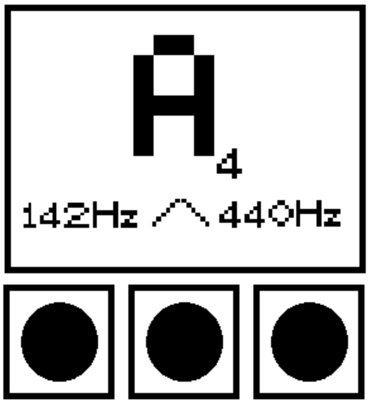
Due to many new and unfamiliar concepts and tools, as well as hardware arriving sequentially, design was largely focused on independent modules. Once a piece of hardware was obtained, what was required of it was precisely determined and code was then written to carry out this task independently. Once two or more modules worked satisfactorily, they could be implemented together in the larger system.

Implementation & Testing:

Using much simulated data, modules were gradually drawn into the larger system so that they could be more effectively tested and debugged. Some parts of the system relied far more heavily on the performance of certain hardware, and therefore had to be more thoroughly explored and protected against error.

- The microphone's purpose was to sample a sound waveform and then the fundamental frequency could be determined using Fast Fourier Transformation. Initial testing using an online tone generator to generate pure sine waves of known frequency revealed that the frequencies that could be sampled ranged from 50-500 Hz. This range fortunately contained all notes in a guitar's standard tuning. It was also observed that while idle, background noise resulted in vastly varying frequencies being read, which was extremely undesirable. To overcome this, the code was changed so that the frequency would only be updated if the voltage read from the microphone exceeded the value produced from background noise level. Final testing with a guitar revealed that variations associated with an actual stringed instrument introduced small error. As a result, the tolerance was adjusted so that frequencies within 5 Hz would be regarded as in tune.
- The motor, being controlled using an h-bridge constructed from individual transistors, was found to experience significant lag in stopping and starting. This problem seemed associated with the motor itself and little could be done to improve this.
- The screen provided no trouble as it served to output data rather than receive it. Using a graphics library which used sufficiently little memory, the screen initially displayed placeholder data before being easily implemented into the real system.

User Guide:



The finished device consists of the buttons, screen and Arduino connected in one unit, with the microphone and motor connected with long, separate wires. The microphone is placed so that it can clearly receive the sound produced e.g. in front of an amplifier or inside the guitar body. The motor is mounted on the machine head of the string to be tuned such that it can rotate the machine head relative to the headstock. The user can now input their note selection by using two buttons to navigate an array of notes. The current pitch and the target pitch are displayed below with a symbol between denoting the adjustment to be made. Once the middle button is pressed the motor will activate, turning in the direction computed while the button is held. Once the instrument is tuned, the symbol displayed is '==' and the middle button should be released. Repeat for the rest of the strings on the instrument.

Appendices:

Git Repository - https://github.com/JackP2112/IoTProjectSemester2

Code – Tuner CodeV1.0.txt in Git Repository.

Presentation Slides – Automatic Guitar Tuner.pptx in Git Repository.