Final Project ECE 470 Microcontroller Based Guitar Tuner

Benjamin Bachman, Clifford Villerot, Karlou Putrus
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
E-mails: babachma@oakland.edu, cpviller@oakland.edu, kgputrus@oakland.edu

***Abstract*—The purpose of this project is to create a system which allows an accurate tuning of a guitar. This system will be implemented using the Dragon-12 light board and with an external amplifier, low-pass filter, peak detector, and comparator.**

I. Introduction

The project will cover the design and implementation of an accurate guitar tuner. This project was chosen because it represents a real world application of the concepts that have been taught in class. We have all played the guitar at some point in our lives, so incorporating our passion of music with a school project was a good idea. Some of the concepts that will be used will be timer input and output compare, interrupts, LED's, ADC, and using the LCD. We will discuss how we receive a signal, amplify that signal, set up our circuit, and how we manipulate that signal in order to read frequency and display it on the LCD.

II. Methodology

A. Amplifier and Low pass filter

The first step in tuning a guitar is receiving the signal from the guitar. Since the guitar has magnetic pick-ups, this will be accomplished by using an amplifier to amplify the signal from the guitar. The output of the guitar was measured to be a signal with amplitude of between 250mv and 500mv. Since only the distance between the peaks is being calculated, only the positive half of the signal needs to be amplified. This means that we would need to include a non-inverting amplifier in order for the overall amplification to be approximately 33 V/V. The second step will be to diminish the higher frequencies so that the fundamental frequencies will be easier to obtain. Since the highest frequency on a guitar is ~330 Hz, the project will need a low-pass filter tuned slightly above that frequency. The resulting signal that is to be obtained after passing through the amplifier and filter should be the positive half of the raw signal that stays between 0V and 5V. The labeled circuit is shown in figure 14 (end of the report).

Figure 1: Circuit flow diagram

B. Peak Value Detection and waveform comparison

We originally wanted to input our waveform into the ACD and use an algorithm to find the peak value. Once the peak would be found, a percentage of it would be compared with the raw ADC waveform and when it is greater than that value it will output a 1 to a GPIO. This would generate a square wave with a period the same as the fundamental frequency. We modified our plans and added in a peak detector and comparator in our circuit in order to simplify our code, as you can see in figure 14.

C. Timer module frequency detection

This wave will then be used as an input to the timer module and the frequency can be found using either pulse accumulation or by the input compare function. Several readings will be averaged to provide a more accurate reading. Below shows the step by step pseudo code to convert from timer number to frequency.

1. capture rising edge if there is a previous edge captured
2. period = rising edge – previous edge
3. period sum = period sum + period
4. previous edge = rising edge
5. edge counter ++
6. $frequency = \frac{timer\ period}{signal\ period}$
7. $signal\ period = \frac{period\ sum}{edge\ counter - 1}$

D. User Interface

Once the frequency is found it can be compared to the frequency of the string of the guitar and which note is being tuned will be displayed on the LCD.

E. BCD to ASCII

We wanted to convert from BCD to ASCII because we wanted the result displayed on the LCD to be numerical. Where if we were to convert from BCD to HEX, there would be letters displayed on the LCD. In order to fulfill project requirements, we used ASM code for this section. The ASM function is shown below in figure 2.

```
BCDtoASCII:
        ADDB #$30
        STAB ASMreturn
        RTS
```

Figure 2: ASM Code Preview

III. Experimental Setup

A. Hardware

To very that our setup was functional, we used code warrior for the HCS12 board and also these hardware items:

- Oscilloscope: This was used to verify correct output signals from the guitar. We verified our circuit did what it was supposed to do.

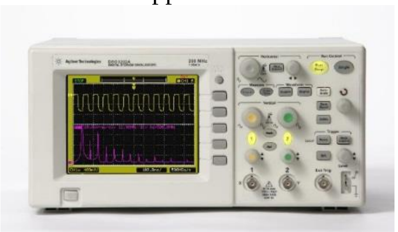

Figure 3: Oscilloscope Used

- Function generator: This was used to test our circuit with a waveform so we didn't have to rely on bringing the guitar.

Figure 4: Function Generator Used

- Guitar: Used as the basis for our input (strings, etc.)

Figure 5: Guitar Used for Tuner

- Jack: Connects our guitar to the board.

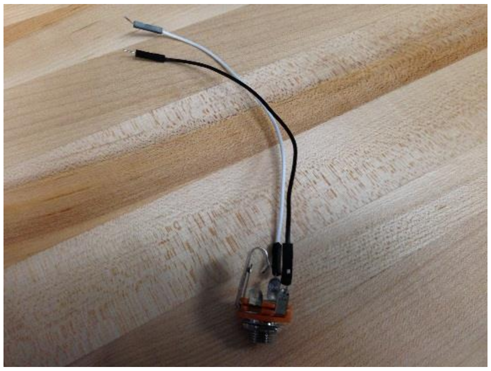

Figure 6: Jack that connected guitar to circuit

- Quad OP Amp (LM324)

Figure 7: OP AMP used for circuit

B. Final Soldered Circuit

After we completed our circuit, we wanted to make it presentable, so we decided to solder our circuit. You can see the soldered circuit below in figure 8.

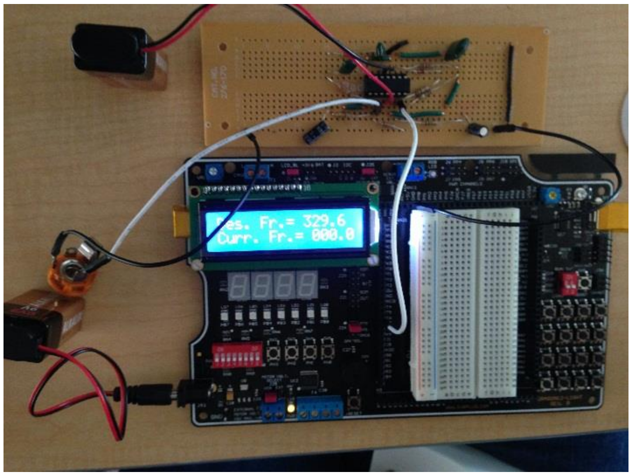

Figure 8: Final Soldered Circuit

IV. Results

A. Project Successes

We were successfully able to wire our circuit and display frequencies of all the strings on the LCD. We are more hardware dependent, but we successfully included all requirements for this project. We implemented interrupts for the timers. The interrupts were mainly used for detecting frequency and producing our 75 millisecond sampling time. The ASM code was implemented in our BCD to ASCII function and that was successful. We expected our circuit to produce a correct input that we could manipulate in our code. What we didn't expect was how accurate our tuner would be. The frequency was accurate to a tenth of a Hertz.

B. Oscilloscope Waveforms (Beginning to End)

Here are the step-by-step waveforms that describes our input to the microcontroller and the circuit as a whole.

1. We first wanted to see what kind of waveform we were working with, so we connected the guitar straight to the oscilloscope. You can see the raw waveform that was generated below in figure 9.

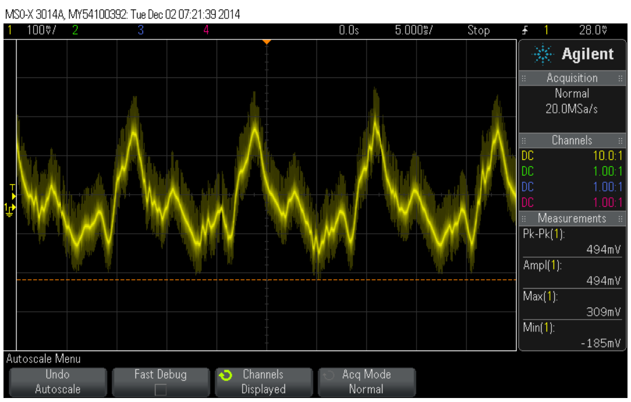

Figure 9: Guitar RAW waveform

2. We then implemented our circuit with our non-inverting amplifier to get a gain of 33V/V. You can see the output waveform that was generated below in figure 10.

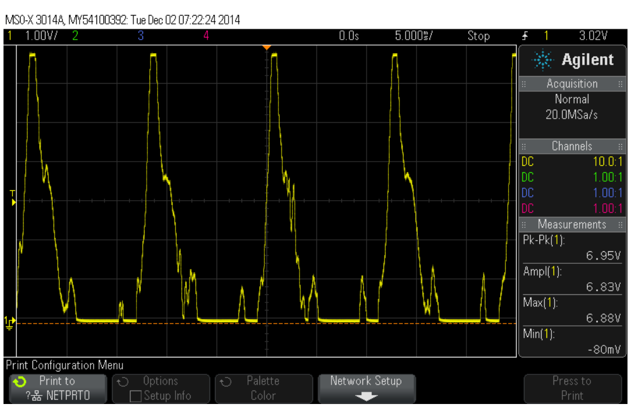

Figure 10: Waveform with Non-Inverting Amplifier

3. After we wired up our low pass filter, the output was a lot cleaner as you can see in figure 11.

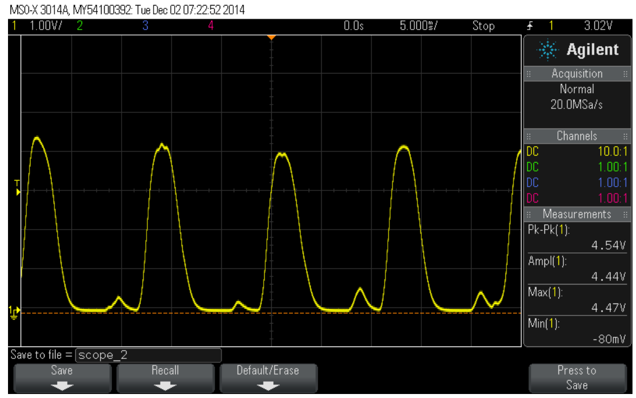

Figure 11: Waveform with low pass filter

4. Adding the peak detector you can see the correct output waveform we were able to generate in figure 12.

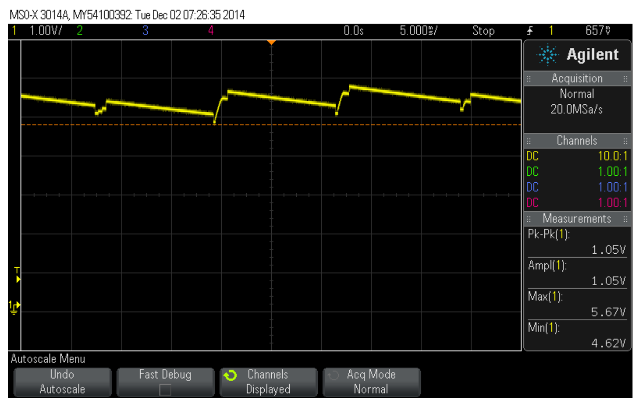

Figure 12: Waveform including peak detector

5. After it runs through the comparator, you can see the resulting waveform in figure 13.

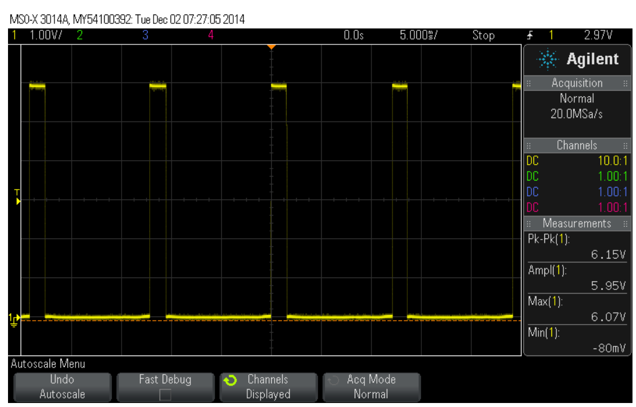

Figure 13: Final waveform including comparator

CONCLUSION

The main thing we take from this project is that it is a real world application of what we learn in class. We also furthered our learning about interrupts and the ACD. The issues that remain to be solved was the tuning of all six strings. The main thing we wanted to accomplish was the proof of concept of being able tune one string according to the frequency. We know that it is doable, it just takes a little more tweaking. Some improvements that can be made are adding an external screen that displays the frequency, filter apparatus, and the hardware apparatus.

ORIGINAL GOALS

The following list details the original scope of the project and a brief description and explanation for deviation from the original design.

Peak Detection and Comparator: The original design utilized programming algorithms to detect the peaks of the incoming signals and filter out unwanted waveforms. This was implemented in hardware due to time constraints and the project team's greater familiarity with operational amplifiers.

All-Chord Auto Detect: Due to hardware constraints the tuner functions best for three chords and poorly for any of the others. Additional circuits with filters more specific to the other chords would provide the desired inputs with the software algorithms already in place.

LCD Chord Display: Originally the played chord would be displayed on the LCD with the LED's sliding to the right or left indicating how to tune the guitar. This was discarded for displaying the target frequency and current input frequency to the tenth of a hertz, which is significantly more accurate.

REFERENCES

[1] Unnamed Author, "Just two ICs can create an accurate guitar tuner," http://electronicdesign.com/displays/just-two-ics-can-create-accurate-guitar-tuner, September 14, 1998

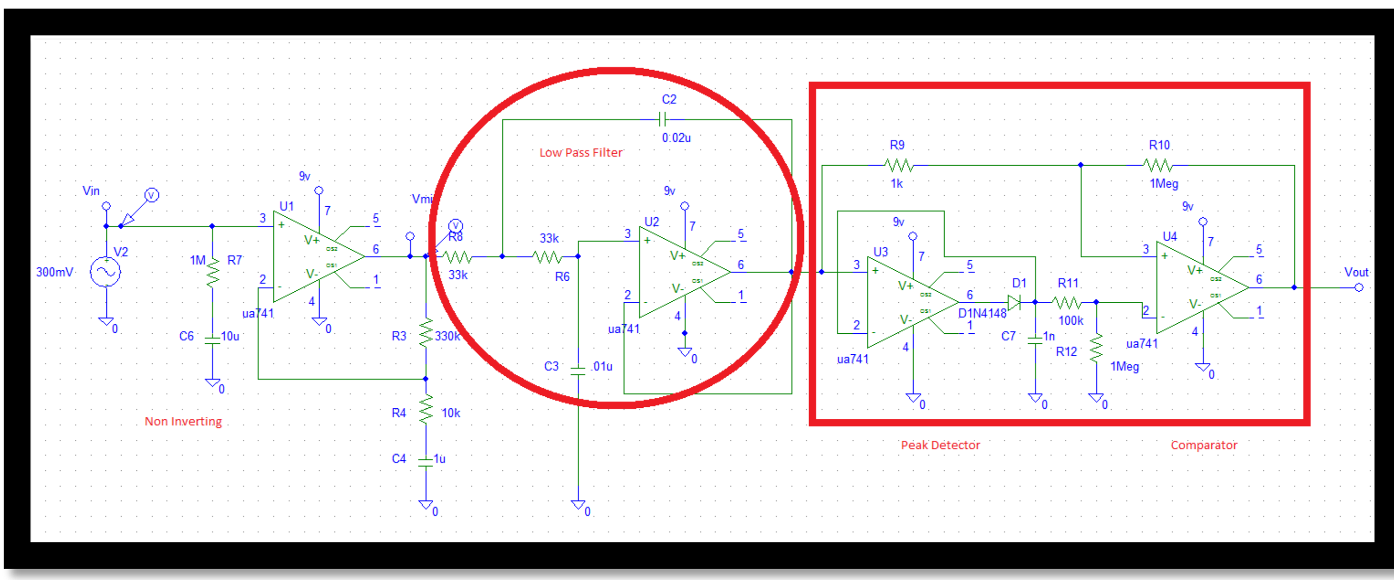

Figure 14: Final Circuit (Includes Non-Inverting Amp, Low Pass Filter, Peak Detector and Comparator)