EE3102

9 Febuary 2012

University of Minnesota

Preliminary Design Report

Project Members:
Patrick Barrett
Mark Pennebecker
Kayiita Johnson
Prashant Dhakal

Submitted to:
Professor Higman

Abstract

The general purpose of this project is the design and construction of a music tuner. Specifically, this tuner will be checking whether or not an input pitch matches 440Hz (A). The input will be read from either a microphone or a coax cable fed from an oscilloscope. The purpose of a tuner is not only to test on whether or not a pitch matches a note, but also to see if the input pitch is sharp or flat (whether the pitch is above or below the given pitch). Our tuner will indicate this information through a series of LEDs. We plan to use a microcontroller to analyze the input and control the output LEDs.

1 Introduction

(Explanation of how cents and such work, pitches etc... More to come on this later)

Our tuner will take an input signal to be tested. When the input pitch matches A-440Hz, the center LED will light up. As the pitch moves out of tune, other LEDs will light up to signify how sharp or flat the pitch is compared to A. These LED's will be set to 5 cent intervals ranging from -30 to 30 cents around A. Anything above or below these ranges will simply light up the leftmost or rightmost LED depending on whether they are above or below the required pitch. The decision of which LED to light up will be handled by a microcontroller.

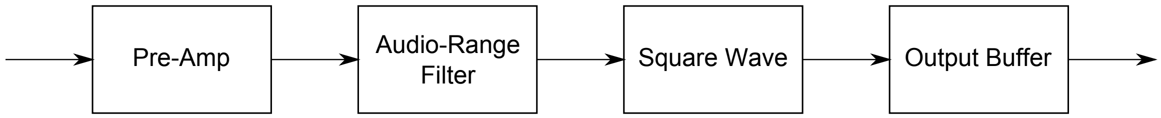

Figure 1: Block Overview

The basic flow of the signal is shown in Figure 1. After the input is received from the oscilloscope (microphone input will need a few more steps), the signal will pass through a bandpass filter to help remove any noise from the signal. After being filtered, the signal will pass through a preamp, to boost the signal so that it can be further processed by the later stages. The signal is then sent to a square wave converter, which will take the analog signal and convert it into a series of square pulses to be handled by the microcontroller. If nessary there may also be an output beffer stage. The microcontroller will take the signal generated from this stage and measure the frequency of the square pulses over a certain period of time. Using this information, the microcontroller will choose light up an LED to display how close the signal is to the desired pitch.

2 Body

For the microcontroller aspect of the datapath, we decided to use a PIC18 device. Given the relative simplicity of the work required by the microcontroller, we chose this device do to our familiarity with the inner workings of the microcontroller rather than any special functionalities of the PIC18.

The CCP module of the PIC18 can be used to measure the time in clock cycles between events. In this application we'll define an event as the falling edge of the square wave generated from the previous stage. Using one of the internal timers of the PIC18 we'll be able to measure the time between events and we can calculate the frequency from that time measurement. Since it is possible for the frequency to fluctuate we will take an average over a (relatively) number of samples. Based upon the frequency calculated, the microcontroller will turn on an LED corresponding with the calculated frequency. To accomplish this, we plan on using a single pin per 5 cent range. (Note that we may end up using a shift register to cut down on the number of pins required).

3 Microcontroller Timing

Cent calculations for 220hz-440hz:

$$\frac{440-220}{1200} = .1833\frac{Hz}{cent} \tag{1}$$

Time difference between cents:

$$\frac{1}{440Hz} - \frac{1}{439.817Hz} = 1\mu S \tag{2}$$

Microcontroller time per colck cycle:

$$\frac{1}{16MHz} = 62.5nS \tag{3}$$

Clock cycles per cent:

$$\frac{1\mu S}{62.5nS} = 16 \tag{4}$$

Time of one cycle of input signal:

$$\frac{1}{440Hz} = 2.27mS \tag{5}$$

Clock Cycles per Signal Cycle:

$$\frac{2.27mS}{62.5nS} = 36363.6 \tag{6}$$