A Tuning Device for a Musical Instrument

7289441

School of Physics and Astronomy
The University of Manchester

Third Year BSc Dissertation

May 2011

Written by William Poynter

Abstract

This design essay discusses a new design for an electronic musical instrument tuning device. The device uses an ADC built into its PIC18F4550 to convert a musical note into digital format so that the microcontroller can perform fast Fourier transform on the signal to determine the base frequency of the note being played. The device determines the closest note and displays how accurately the note is tuned. The recommended range of use is E_3 (164.8Hz) to E_5 (659.3Hz). Within the recommended range the program can accurately place the note being played within the 17 bins that make up the accuracy meter of the tuning device. The program will attempt to determine the accuracy of notes down to E_2 (82.4Hz), but the accuracy is compromised. The sampling frequency of the ADC and the FFT are 4096Hz and the program will hold up to 8192 samples at one time; allowing a frequency resolution of 0.5Hz.

1. Introduction

The science of audio acoustics has a wide range of practical uses within the modern world, from very important applications, such as bridge building,[1] to everyday applications such as an electronic instrument tuner.

The purpose of an electronic instrument tuner is to aid musicians who cannot tune their instrument by ear alone, without using a reference, such as that produced by a tuning fork.

Not all musicians can tune their instruments just using a reference frequency and their ears, and even fewer musicians can tune without any reference at all, but research indicates that musicians and even non-musicians can tune a musical instrument while using an electronic tuner.[Appendix A] Even though many musicians can tune using a reference this requires them to carry around a tuning fork of the correct frequency. A much more multipurpose and accurate method for tuning would be an electronic instrument tuner,[Appendix A] which would be able to tune to all semitones over multiple octaves.

There are currently several electronic tuners on sale within music shops in the United Kingdom. Therefore it is important to address why this design of musical tuner is better, cheaper, or more convenient than current designs and thus has a reasonable chance of being successful in the market.

Current electronic tuners retail for prices around £15.00.[2] Most retail stores add somewhere between 20 and 30 percent to the cost they buy their tuners at, implying a cost to them of about £12.[3] This implies that in order to sell the new product for less than current products, it needs to be produced for less than £11.50, to allow the production company to make a profit. The two largest ongoing cost factors in producing the new product are the components that the tuner is made from and the production cost of building the product. In addition, the project needs to recover its design and setup costs over a period of time. Other major ongoing costs include marketing and management.

For the purpose of this project it is assumed that the tuner will be batch produced in quantities of 2000.

2. Theory

This project will only be concerned with tuned musical instruments, but non-tuned instruments also need to be taken note of, examples of these are snare drums, cymbals and the triangle.

2.1 Note Production

A note is produced within a musical instrument in one of two ways, via continual force or via acute force. The first is where the sound is created by a continuous driving force causing a resonance and continuing until the force ends, e.g. a flute or a violin when used with a bow.[4] The second is where energy is transferred to the instrument in an impulse, and this energy is then allowed to dissipate over time, by damping [4]. Examples of the second method of note production include harps, pianos and guitars.

Both types, continual force and acute force, construct notes in the same way. The lowest harmonic frequency is the base frequency, which relates to frequency of the note. But instruments do not create notes that just consist of the base frequency; harmonics are also created at the same time. For example, a plucked string that is tuned perfectly, to 440Hz, will produce waves of frequencies 440Hz, 880Hz, 1320Hz, 1760Hz, etc. These higher harmonics are considered to be overtones. Different instruments have different proportions of these overtones; this causes instruments playing the same note to sound significantly different.

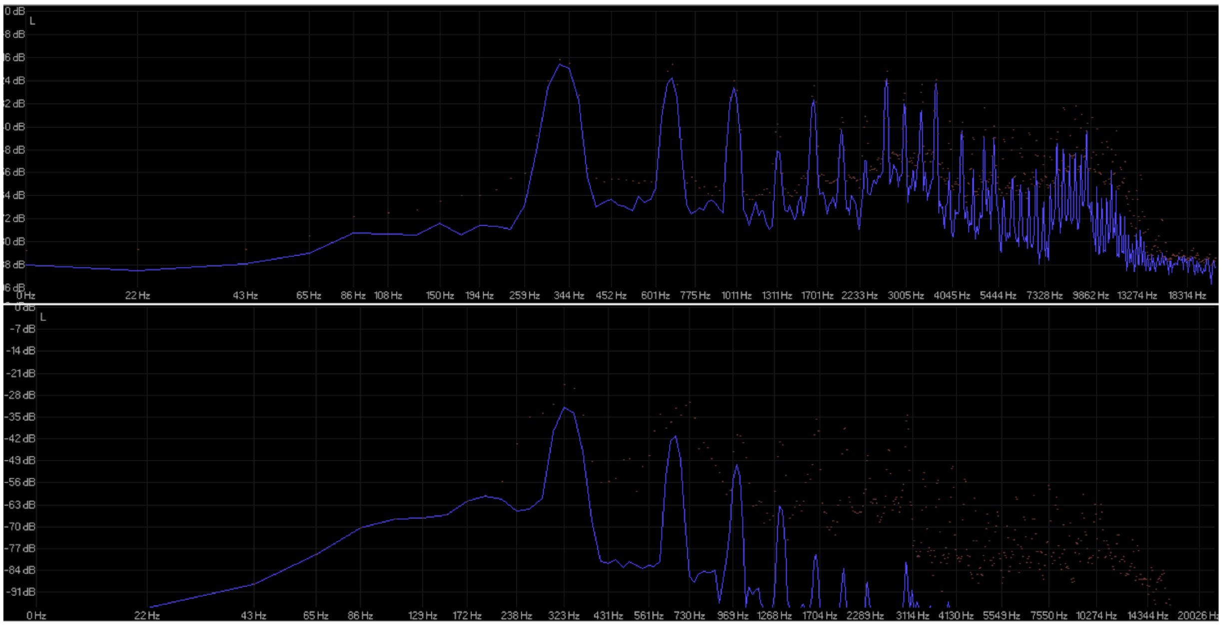

Figure 1 - (top) An FFT taken from the recording of a harmonica playing an E4 (329.6Hz). (bottom) An FFT of a ukulele playing the same note. It is clear to see that both notes are made up of the same frequencies but different proportions of them.

2.2 Musical Scales

In the western world, music is typically written and performed using the chromatic scale. The chromatic scale consists of 12 semitones, 7 naturals and 5 flats, which make up a single octave. Each of the flats are referred to as a semitone below their corresponding natural, e.g. $B^{\flat}$ is a semitone below B. These five flats can also be considered to be a semitone above their corresponding natural, but then they are referred to as sharps (e.g. $A^{\sharp}$).[5]

2.3 Musical Frequencies

The chromatic music scale is a logarithmic base 2 scale, denoted by the letters from A to G. Concert pitch, the pitch that a group of instruments are tuned to, can vary from ensemble to ensemble, but is normally extrapolated from middle A, which is typically taken to be 440Hz. For the purpose of this report the scale will be based on middle A having a frequency of 440Hz. Because of the logarithmic nature of the scale the A one octave below middle A, known as low A, is 220Hz and the A one octave higher, known as high A, is 880Hz.

Scientific pitch notation labels the notes using their letter coupled with a number to indicate which octave they belong to. The middle octave is denoted by the number 4, therefore the octave below middle octave is denoted by number 3. As a result, A4 = Middle A = 440Hz and A3 = Low A = 220Hz.

Note	Frequency /Hz	Note	Frequency /Hz	Note	Frequency /Hz
E_3	164.8	Middle' C_4	261.63	$G^{\#}_4/A^{b}_4$	415.3
F_3	174.6	$C^{\#}_4/D^{b}_4$	277.2	A_4	440
$F^{\#}_3/G^{b}_3$	185	D_4	293.7	$A^{\#}_4/B^{b}_4$	466.2
G_3	196	$D^{\#}_4/E^{b}_4$	311.1	B_4	493.9
$G^{\#}_3/A^{b}_3$	207.7	E_4	329.6	C_5	523.3
A_3	220	F_4	349.2	$C^{\#}_5/D^{b}_5$	554.4
$A^{\#}_3/B^{b}_3$	233.1	$F^{\#}_4/G^{b}_4$	370	D_5	587.3
B_3	246.9	G_4	392	$D^{\#}_5/E^{b}_5$	622.3

Figure 2 - A table displaying the relationship between notes, denoted by the scientific pitch notation, and their frequencies.

Each semitone is divided into 100 cents. Cents are a useful unit when considering tuning, as they are a linear unit relating to the accuracy of a note. For example, E4 (329.6Hz) and F4 (349.2Hz) are separated by 19.6Hz, whereas E5 (659.3Hz) and F5 (698.5Hz) are separated by 39.2Hz. By contrast, the gap between E4 and F4 is 100 cents, and the gap between E5 and F5 is also 100 cents.

In terms of the accuracy required for an instrument tuner, the concept of just noticeable difference is relevant and useful. The just noticeable difference (j.n.d) of the human ear is the smallest detectable difference the ear can detect between two stimuli and it can be measured, in tuning, to be 5 cents.[6]

3. Design

It has been decided that the tuner will be able to tune as many instruments as possible, to increase the potential market for the tuner. This approach would include instruments from multiple families, for example strings, woodwind, and tuned percussion. Therefore, the tuner will be able to recognise all 12 semitones within an octave, since many woodwind and brass instruments tune to flats. Also, even within the same family there can be a several octave difference between two instruments (e.g. cello and ukulele)[7][8]. Therefore the tuner's recommended range of use will span from E3 (164.8Hz) to E5 (659.3Hz). The tuner will not be able to tune notes above this range, but will be able to tune below this range, down to E2 (82.4Hz), but with less accuracy than the accuracy meter suggests.

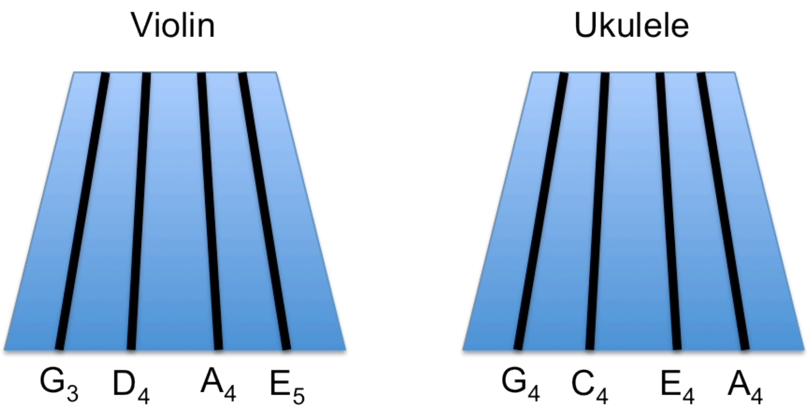

Figure 3 - (left) Represents the neck of a violin displaying the notes the strings are normally tuned to. (right) Shows the normal tuning for a ukulele.

3.1. Initial Signal Analysis Designs

In the early stages of designing the tuner a number of ideas of how to best determine the frequency of the note being played have been considered. These ideas have been narrowed down to three approaches, which are outlined below.

Design 1 – is an analogue design consisting of many narrow band-pass filters, much like the design of a graphical equaliser. These are then fed into a microcontroller and the portion of the initial signal to be passed through each filter gives an indication of the frequency. The key problem with this design is the overall size in comparison with accuracy. In order to be able to determine 36 notes, over 3 octaves, it requires a minimum of 36 RC band-pass filters, consisting of 72 resistors and 72 capacitors.[9] Even a design of 36 band-pass filters does not necessarily guarantee accuracy in determining how flat or sharp a note is. For this reason this design has been dropped.

Design 2 – is a design that involves both analogue and digital systems. This design is inspired from the design of a FM radio tuning system.[10] It has one active narrow band-pass filter that is adjustable. A microcontroller controls an electronically variable capacitor (varicaps) that would alter the capacitance, causing the filter to sweep over the desired frequency range. When the filter reaches a frequency that is being received by the microphone the filter would resonate and cause a spike in the gain. Simultaneously, the microcontroller assesses the amplitude of the signal leaving the filter and identifies the frequency of the peaks. This design has a lot of potential, but assessing it commercially shows that it is more complicated and more expensive than required for this product. This design might be suitable for more expensive professional tuning devices.

Design 3 – is the chosen design. All of the signal analysis is done digitally by the means of the fast Fourier transform (FFT). A program to perform the FFT and other tasks to best identify the note being played would be programmed onto a microcontroller. This design has been selected because of its small number of components, satisfying the objectives of keeping the device both cheap to produce and small in size.

Design 3 also has another advantage over both of the previous designs, in that it is easier to calibrate and verify that it has been made correctly. During construction of design 3 there are two circuits to calibrate and a program needs to be executed on the microcontroller before it is mounted to verify that it was programmed correctly.[11] Design 1 would require, at a minimum, the tuning of 36 separate filters, each one with two tuning points. Design 2 only has one system to calibrate, but it involves running the device over several different input frequencies, re-calibrating and repeating until it is correct.

3.2 Input, Output & Power

All three designs, the two excluded and the chosen design, are powered using the same method and have exactly the same design for how the signal would be input to the tuner and how the accuracy of the note would be output to the user.

3.2.1 Power

The power source for the tuner would come from four AA alkaline cells. All of the components are low current thus AA would be the best compromise between size and charge.[12] The four cells would be connected in series to produce a potential difference of 6V. This is the minimum that can be realistically used in any of the three designs as they contain a microcontroller using TTL technology[1][13] and all three contain op-amps which need approximately 1.5V operational voltage either side of the signal, which has peak-to-peak voltage of 2V [14], resulting in the minimum potential difference to power the op-amps to be $(2 \times 1.5V) + 2V = 5V$.

As the chosen design contains a microcontroller with built in TTL circuitry that requires a power supply voltage in the range of 4.2V to 5.5V, it is vital to divide the supply voltage to gain a potential difference in this range. Microcontrollers can also suffer when given unstable voltage supplies; for this reason the voltage will be divided using a zener diode in reverse bias to maintain a difference of 5.1V.

Finally, the input circuitry requires a supply voltage that is half of that supplied to the microcontroller. To create this supply voltage, as less stability is required, a zener diode is not used but a potential divider, constructed from resistors, is used instead.

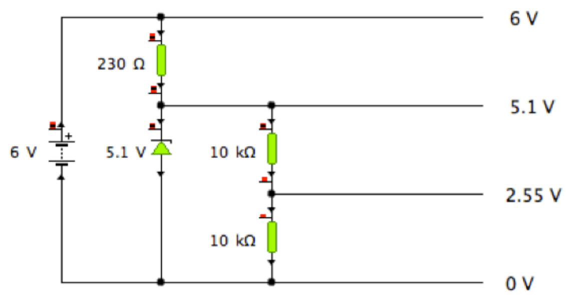

Figure 4 - Schematic diagram of the power distribution circuit. It divides the 6V potential difference into two additional outputs, 2.55V and 5.1V.

3.2.2 Input

The tuner is equipped with a basic internal microphone that would be biased using the power source. By using a microphone, as opposed to an input socket, the tuner is able to work with the

---

[1] TTL, transistor-transistor logic, is a type of circuit built from bipolar junction transistors and resistors.[15]

entire range of instruments, not just electric instruments. The microphone has to be biased to allow both compression and rarefaction of the diaphragm of the microphone.[16]

The signal from the microphone would then be passed through a band-pass filter broad enough to filter out frequencies below 80Hz and frequencies above 2000Hz. This filtering would remove some of the noise in the signal and also protect the components from dangerous low frequencies, which can be interpreted by the components, as unwanted voltage biasing, which could lead to damage of the components. The high cut part of this filter also acts as an anti-aliasing filter for the FFT. Due to the requirement to amplify the signal from the microphone an active Butterworth filter has been selected. The microphone outputs the signal of $1.5V_{rms}$, which is then amplified down to line level, $1V_{rms}$.[17]

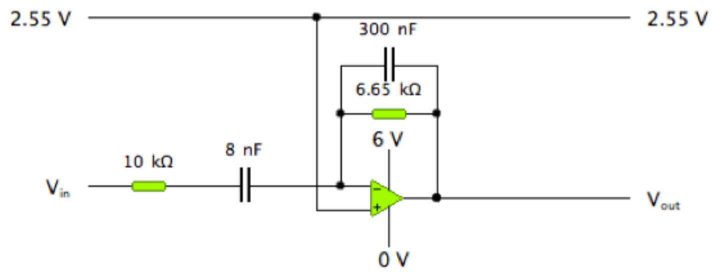

Figure 5 - Schematic diagram representing the input Butterworth filter. Its break frequencies are 80Hz and 2000Hz.

3.2.3 Output

The output needs to meet three main criteria: be easy to understand, be cheap to produce and be small enough to be built into a pocket-sized device. The display needs to do two things; it needs to tell the user which note they are closest to and to tell them how far away from it they are.

3.2.3.1 Displaying the Nearest Note

The output needs to be able to display the 7 letters, A to G, that make up the notes of the chromatic scale plus an indication whether the note is a flat or natural (e.g. B or B♭). Several methods of displaying a signal letter were explored, but none of them, within a sensible price, were able to deal simply with displaying a letter and a “♭”. Because of this the two elements were separated, the letter is displayed using a dot matrix of LEDs and the ‘flat tag’ is indicated with a single LED positioned towards the top left of the dot matrix.

The dot matrix selected consists of 35 red LEDs in a 7 by 5 matrix. It functions by multiplexing the signal across the 5 columns, lighting up the necessary LEDs per column before moving to the next.

The human eye can't detect flickering in a display if it refreshes above 30Hz.[18] This means that the microcontroller needs to be able to change outputs at 150Hz in order to change all 5 columns at 30Hz.

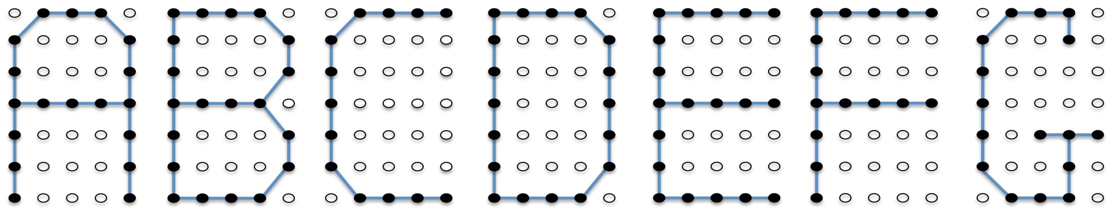

Figure 6 - This demonstrates how each of the 7 letters that make up the chromatic scale are displayed on the dot matrix display.

Using the diagram above it is established that certain columns of LED are repeated over the 7 letters, for example the left vertical lines in B, D, E, and F. This information is used to reduce the size of the program that needs to be loaded to the microcontroller, as it needs fewer presets saved to its memory.

An even more important feature of the display is that multiple columns of LEDs for a single letter can be lit up at the same time without interference. This means that for all the letters apart from G, the number of changes to light up all 5 columns is reduced to just 2 or 3.

It has been decided that the output refresh rate will stay at 150Hz for all letters, meaning the letters complete their refresh cycles at different rates.

The 'flat tag', to indicate whether the note is a flat or not, is a much simpler design. It consists of one red LED mounted just underneath the case. The plastic covering this LED is then be engraved with a " ♭ " so that when the LED is illuminated the light will shine through the engraved part.

3.2.3.2 Accuracy Meter

The accuracy meter is even simpler in design than the 'flat tag', due to the fact it does not require any tricky engraving. It consists of 9 LEDs, 1 green, 4 orange and 4 red. The green is positioned in the middle and indicates when the note is within tolerances of being correct. The other 8 LEDs are positioned, 2 of each colour either side of the green to represent how far out the note is. Either one or two LEDs will be lit when the device is receiving a note.

3.3 The Chosen Signal Analysis Design

The chosen design is a careful balance between low cost components, accuracy and speed. The chosen design is completely digital. The tuner feeds the signal output from the initial band-pass filter, in the input stage, straight into an ADC (Analogue to Digital Converter) input on a

PIC18F4550 microcontroller. The microcontroller then analyses the signal, and chooses the appropriate display to send to the outputs. This processing within the microcontroller will consist of several separate processes, as shown in figure 7.

3.3.1 Analogue to Digital Converter

The chosen microcontroller, PIC18F4550, comes with several built in ADC inputs. For this product only one ADC input is used, allowing the others to be used as programmable inputs and outputs. The ADC can output using 10 bits, which would result in waves being quantised into 1024 different amplitudes at each point. The microcontroller is equipped with an 8 bit by 8 bit hardware multiplier, which reduces the number of clock cycles per calculation. The ADC is programmed to only use 8 bits meaning the signal is quantised into 256 steps instead of 1024. As the 256 steps are spaced over 2 volts this results in a quantisation error of 7.8mV.

$$\frac{2000mV}{256} = 7.8mV$$

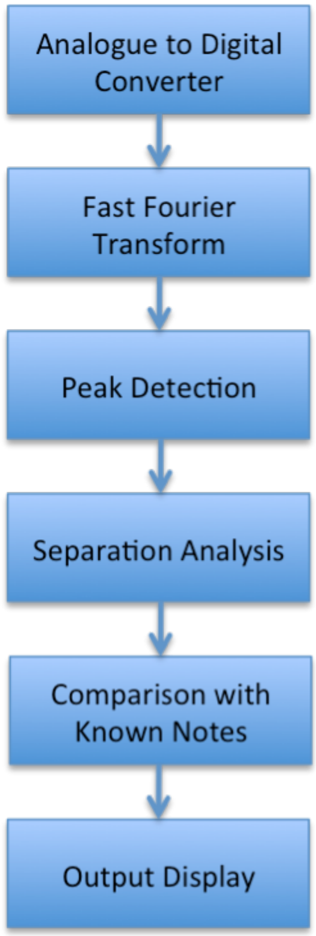

Figure 7 - A flow diagram to illustrate the processes performed by the microcontroller.

Although this would be a huge error for music to be recorded at, if it were to be played again, for this analysis it is deemed an acceptable trade-off, because it does not directly affect the frequency. The highest frequency that needs to be analysed, the Nyquist frequency, is 2048Hz, chosen to be a power of 2 for convenience, therefore the sample rate has been chosen to be 4096Hz. The ADC records 1024 samples with inter-sample separation of 244μs. It is vital to the integrity of the FFT algorithm that the separation is kept uniform. Each sample point ranges from 0 to 255 meaning that the virtual ground is positioned at 127. This is beneficial as the microcontroller can do unsigned multiplication in a single clock cycle. This stage of the processing is the longest and takes 1024 * 244 μs = 0.25s.

The dot matrix display needs to change its output every 6.67ms and it is controlled by the same microcontroller that is collecting the data using the ADC, this means that the display changes need to entwine with the data collection cycle. The program uses one of the microcontroller's four timers to check after each sample is taken whether it has been 6.67ms since the last dot matrix change. It then uses a second timer to make sure that it has been exactly 244μs since the last sample was taken. If a sample point needs to be taken at the same time as updating the display the sample point will be taken at the correct time and then the display will be updated immediately afterwards, as this will not cause any noticeable difference to the user, while delaying the data sample could be catastrophic to the FFT.

3.3.2 Fast Fourier Transform (FFT)

Fast Fourier transform is a faster algorithm that conducts discrete Fourier transform (DFT). DFT takes a discrete sample of signal and performs Fourier transform on it to conclude what combination of sine and cosine waves could re-construct the inputted signal. This makes DFT ideal for digital signal analysis and therefore ideal for determining the frequencies being played by the instrument. The drawback of DFT is the speed of the algorithm. DFT requires N^2 calculations to be conducted, where N is the number of samples.[19] The number of calculations can become quite unruly quite quickly, especially using a microprocessor that has a relatively low clock speed, 48MHz.[13] This is where FFT comes into use. FFT does exactly the same job as DFT but with a lot fewer calculations meaning the whole process can be done quicker. FFT reduces the number of calculations by removing the redundant calculations done by DFT. FFT only requires $2N \cdot \log_2 N$ calculations to do the same analysis as DFT.[19]

N	DFT (N^2)	FFT ($N \cdot \log_2 N$)	Ratio DFT/FFT
64	4096	384	10.7
128	16384	896	18.3
256	65536	2048	32.0
512	262144	4608	56.9
1024	1048576	10240	102.4
2048	4194304	22528	186.2
4096	16777216	49152	341.3
8192	67108864	106496	630.2

Figure 8 - Table demonstrating the time saving ability of FFT over DFT. The right column displays how many times more calculations DFT would perform instead of FFT.

The FFT algorithm being used is the Cooley-Tukey algorithm named after James W. Cooley and James Tukey, who reported it 1965.[20] It computes exactly the same result as a DFT algorithm but in far fewer calculations, as is expressed in figure 8. It does this by dividing a single DFT with N samples into multiple DFTs. For example a single DFT could be divided into two DFTs, each with N/2 samples; it then recombines these two DFTs by performing many DFTs with 2 samples, often referred to as the butterfly operations. This process removes some of the redundant multiplications performed by doing the DFT as a single process.

Aliasing was briefly mentioned earlier in the input section. Aliasing is phenomenon caused by DFT where there is a mirror effect in the output about the Nyquist frequency. If DFT were run on a pure sinusoidal signal of 50Hz with a Nyquist frequency of 256Hz the output would display two spikes. The first spike would be at 50Hz, as would be expected, but it would also display a spike at 462Hz, an exact mirror of the 50Hz spike. This means that the sampling frequency must be at least twice that of the highest frequency being analysed. It is also necessary to use a filter to remove frequencies above the Nyquist frequency, as any frequency above this would also be mirrored. For example, if a pure sinusoidal wave of 400Hz is input into the DFT, two spikes are returned again, one at 400Hz but also one at 112Hz.[21]

A key problem that plagues FFT is leakage. Leakage is where the sample taken is not an exact integer multiple of the basic period of sample taken; when this occurs a discontinuity is introduced which causes high frequency terms in the FTT output. To avoid this problem a window function is used. This program uses a Hanning window. Using a Hanning window broadens the peaks as well as reducing the height of the peaks by a factor of two. [21]

The biggest asset of the microcontroller that was selected for this design is its ability to perform 16 bit calculations, 8 x 8 bits in a single clock cycle. This coupled with FFT greatly reduces the execution time of the signal analysis.

3.3.3 Peak Detection

Using the array of amplitudes created by FFT the next part of the program detects the peaks in this array. Firstly, the program takes the mean of this array; then it ignores all values below $2 \cdot \overline{signal}$ (i.e. twice the mean of the signal output by the FFT), in order to remove background peaks. The microcontroller then runs a function that compares the height of one point with the 10 neighbouring points, 5 on either side; if the point is higher than all 10 comparative points, the frequency is taken to be the summit of the peak and potentially one of the frequencies of the note.

3.3.4 Separation Analysis

This is the most crucial part in deciding which note is closest to the one that is being played by the instrument. The base frequency is the frequency that corresponds directly to the note. The overtones played cause additional spikes output by the FFT. As these are fundamental to the structure of the note, they can be used to best decide which note is being played. The microcontroller calculates the separation between each of the peaks; the separation should be equal to the fundamental frequency. After doing this once, the process is repeated but instead of calculating the separation to the neighbouring peak, the program calculates the separation between one peak and the next but one peak; this is dubbed the second order separation. The means of both the first order separation and half of the second order separation are calculated, and the mean of those two values taken to be the base frequency.

It is assumed that there is a sufficiently large chance that one of the overtones would be so small that it would be removed by the $2\cdot\overline{signal}$ cut, for this reason a check has been added to the program. After the mean of the first order separation is calculated, any separation that is more than 10% away from the mean would be ignored and the mean recalculated. But this separation is then handed on to the second order separation calculation, and if it is caused by missing a single peak it still has the potential to be used. The 10% check is then run again, on the second order separation mean, due to the chance that the anomalous separation size may be caused by an extra, unwanted, frequency. If that were to be the case, the separations related to the extra peak will fail the 10% check and be thrown away, but there should still be enough signal to accurately determine the base frequency.

3.3.5 Comparison

The calculated base frequency is then compared with a list of known frequencies, stored within the program, to best assess which note the recorded frequency is closest to and how close it is. The program stores these notes as a note number and an output function coupled together. The note number corresponds to the notes position in the array of notes programmed to the microcontroller, the notes the tuner can tune to. The note number is used to calculate the frequency of a note using the equation below,

$$f = 77.9\exp[0.0578\cdot\lambda]$$

where λ is the note number.

The recorded note's note number is then compared by incrementing through the list of note numbers until the program has found the closest match, and that is sent to the output function of the program.

The program then calculates the difference between recorded note number and the closest note stored within the program. This difference is labelled Δ_1. Then using Δ_1 the program illuminates the accuracy meter. The accuracy meter has 17 bins for the recorded note number to fall into. A note that receives only the centre green LED, is correctly tuned within ±0.029 of a preset note number (i.e. an integer between 1 and 25), which corresponds to 2.9 cents error either side of the correctly tuned frequency. Each bin has a range of 5.8 cents.

3.3.6 Program Output

3.3.6.1 Note Display

As previously mentioned, the dot matrix of LEDs displays the letter of the closest note and refreshes at a rate of 150Hz. The second part of this display is the 'flat tag', which indicates to the user whether the note is a flat. To be able to control these two displays, from the microcontroller, 13 pins are required; 12 for the dot matrix, 7 by 5, and 1 for the 'flat tag' LED.

The 13th pin for the 'flat tag' will be constant as long as that note is being displayed and therefore does not require any additional programming.

The first twelve pins that operate the dot matrix change output every 6.67ms. If each of the letters were to be displayed using all 5 columns as 5 separate presets and each letter treated individually, the program would be required to store 35 presets, but by condensing multi columns into one preset, using the same presets for multiple letters and adding a single pin within the program, the number of presets required is reduced to 12. It is important to be as efficient with memory as possible, as the PIC18F4550 only has 32KB of programmable memory.[13] All of the presets are demonstrated in the table below.

Note	Stage				
	1	2	3	4	5
A	A1	A2	D3 + 9		
B	A1 + 12	A2 + 6	B3		
C	C1	C2			
D	A1 + 12	D2	D3		
E	A1 + 12	A2 + 6	F3 + 6		
F	A1 + 12	A2	F3		
G	C1	G2	G3	G4	G5

Figure 9 - Table illustrating the presets and the optional additional pin used to illuminate each stage of a given letter. The colours correspond to figure 9 (below).

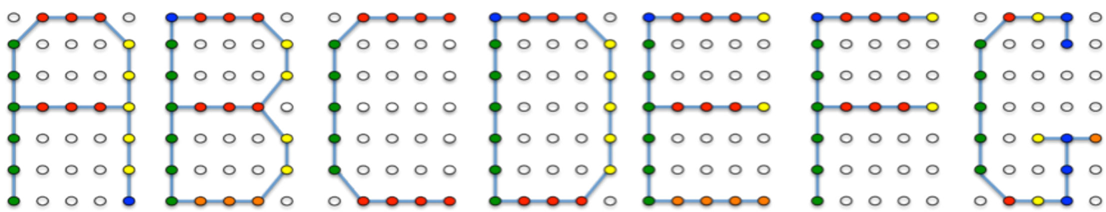

Figure 10 - This schematic diagram demonstrates how each of the 7 letters of the chromatic scale will be displayed using a colour key to indicate each preset or additional pin.

3.3.6.2 Accuracy Display

As previously stated the accuracy meter has 17 bins each of width 5.8 cents (0.058 of a note number). The program has an IF clause, for each of the LEDs, that it performs on Δ_1. A maximum of two IF clauses can return true at one time, this would denote a bin that illuminates two LEDs. For example, the IF clause for the centre, green, LED is "IF ($-0.087 < \Delta_1 < 0.087$) THEN RETURN "true". The clause uses 0.087 instead of 0.029 because the green is also illuminated when both the green and the first orange LEDs are illuminated and thus encompasses the next bin. Each LED clause ranges over 3 bins, 0.174, apart from the two outermost red LEDs where their IF clauses only range over 2 bins, 0.116.

The stages of the program after the FFT (peak detection, separation analysis, comparison and program output) are predicted to collectively take less than 48,000 clock cycles, thus taking less than a millisecond to compute.

3.3.7 Repeat

When using FFT there is a phenomenon known as 'the picket fence effect'. The picket fence effect is caused by the FFT's samples being taken with non-zero spacing. The name of the effect refers to looking at something through a picket fence as you move past it, if a fluctuation occurs inside the spacing, i.e. behind a single picket of the fence, it will be unobserved.[22] This leads to a quantity known as the frequency resolution, which is calculated using the equation below.

$$f_{res} = \frac{f_s}{N}$$

where f_s denotes the sample frequency and N denotes the number of samples.[22]

The program repeats this cycle causing the display to change output just over every 0.25s, just under 4Hz and achieving a frequency resolution of 4Hz. But on the second cycle, the program does not only listen and receive another 0.25s of audio signal; it also uses the previously recorded 0.25s in addition to the newly recorded 0.25s giving the program 2048 samples to use, which improves the frequency resolution to 2Hz. The program continues to add 1024 samples at a time up until it holds 8192, 2 complete seconds worth of recording. But it can only perform FFT on the most recent power-of-2 number of samples (e.g. when the program is holding 3062 it will only perform FFT on the most recent 2048). Once it has reached 8192 samples it removes the first 1024 samples to make way for the newest 1024 samples.

Time since beginning of analysis /ms	Task	No. of Samples Held	No. of Samples Used	Frequency Resoultion / Hz
250	Collecting 1024 Samples			
252	Calculating FFT	1024	1024	4
502	Collecting Another 1024 Samples			
507	Calculating FFT	2048	2048	2
757	Collecting Another 1024 Samples			
761	Calculating FFT on a Newer Data Set	3062	2048	2
1011	Collecting Another 1024 Samples			
1021	Calculating FFT on All the Samples	4096	4096	1
1261	Collecting Another 1024 Samples			
1271	Calculating FFT on a Newer Data Set	5120	4096	1
1521	Collecting Another 1024 Samples			
1531	Calculating FFT on a Newer Data Set	6144	4096	1
1781	Collecting Another 1024 Samples			
1791	Calculating FFT on a Newer Data Set	7168	4096	1
2041	Collecting Another 1024 Samples			
2063	Calculating FFT on All the Samples	8192	8192	0.5
2063	Removing the Oldest 1024 Samples			
2313	Collecting Another 1024 Samples			
2335	Calculating FFT on All the Samples	8192	8192	0.5

Figure 11 - This table displays the iterative cycle the microcontroller undergoes. It ignores the time required by the chip to update the display as it takes less than 1ms.

This system of updating the display, and thus the user, approximately every 0.25s has one main drawback; if the user changes the tuning of the instrument and then re-emits the note or if the user plays an entirely new note the program will take over 2 seconds to clear the old data out of its system. This problem was deemed unacceptable, because it could mislead the user. In order to deal with this issue a final circuit was added to the design, to determine when the user replays a note and then informs the program to begin the entire process from the beginning again.

This can be performed by a simple peak level detector fed into a Schmitt trigger. The peak level detector outputs a voltage in proportion to the amplitude of the signal input to it. This output voltage is then calibrated to trigger the Schmitt trigger whenever the volume rises above a certain value, causing the Schmitt trigger to output logic high which the microcontroller.

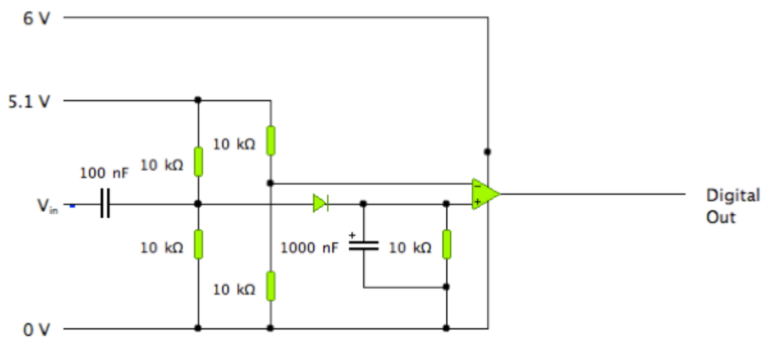

Figure 12 - A schematic diagram showing the design of the peak level detector fed straight into a Schmitt trigger in order to inform the program a new note may have begun.

4. Cost Analysis

The largest cost elements in producing this product are the components that make up the device and the case for the device.

Item	Quantity	Price
Microcontroller (PIC18F4550)	1	£2.83
LED Dot Matrix	1	£1.04
LEDs (1 green, 4 orange, 5 red)	10	£0.027
Zener diode	1	£0.02
Diode	1	£0.009
Microphone	1	£0.21
Variable Resistor	1	£0.084
Resistors	18	£0.008
Non-Electrolytic Capacitors	2	£0.09
Electrolytic Capacitor	1	£0.014
741 Op-amp	2	£0.69
4x AA Battery Holder	1	£0.57
Case	1	£2.74

Figure 13 – A cost assessment table showing the quantity of each component used in the product. For the pricing, it is assumed 2000 products would be made, if less products were to be made the components would cost more.[23]

The total component cost of each product is £9.49 if the tuner is produced in a batch of 2000 products. This is over £2 less than the overall target of producing this tuner for less than £11.50. This suggests that if the cost of construction, management and marketing can be kept below £2 there is potential to make a profit.

5. Conclusion

The electronic instrument tuner can tune to 25 recommended notes, i.e. those with the best accuracy, and 37 notes in total, the additional 12 having less accuracy. These notes include both natural notes and flats, which allows the tuner to be used with a wide range of instruments. Within the recommended range of notes the tuner can tune to an accuracy greater than that of the standard human ear, 5 cents. Research has indicated that a market for this product exists. Cost analysis shows that this design has the potential to succeed within a competitive market.

To further assess whether this product will be a successful design, research can be done to determine how user friendly the design is and whether users determine that the device is both fast and accurate enough.

There are a number of ways to refine or extend this design. One potential improvement would be to use an elliptical band-pass filter instead of using a Butterworth band-pass filter. This would have the benefit of improving the drop off characteristics; the trade-off being that an elliptical filter requires more components and thus will cost more. Design 2, mentioned in the design evaluation stage and which was based on the design of a radio tuning circuit, could be explored further as a more expensive more professional device for tuning instruments like pianos, harpsichords and harps, which all have a very wide range of notes.

References

[1] K. Billah and R. Scanlan, *Resonance, Tacoma Narrows Bridge Failure, and Undergraduate Physics Textbooks*, American Journal of Physics, vol. 59(2), pp. 118-124 (1991)
[2] musicroom.com/se/ID_No/067079/details.html: The Music Room (Viewed: May 2011)
[3] The Music Room Store: Interview with employee. (April 2011)
[4] phys.unsw.edu.au/jw/Bows.html: The University of New South Wales (Viewed: May 2011)
[5] D. Andrew, "Combinatorial Music Theory", *Journal of the Audio Engineering Society*, vol. 39, pp. 427-448. (June 1991)
[6] hyperphysics.phy-astr.gsu.edu/hbase/music/cents.html#c4: The Use of Cents for Expressing Musical Intervals (Viewed: May 2011)
[7] cellowiki.wetpaint.com/page/Cello: Cello - Cello Wiki (Viewed: May 2011)
[8] ukulelehunt.com/2008/04/10/ukulele-tuning-notes: Uke Hunt (Viewed: May 2001)
[9] P. Horowitz and W. Hill, in: The Art of Electronics, 2nd ed., (Cambridge University Press, 1989)
[10] Mini Project- Radio Receiver Design & Evaluation, University of Hertfordshire (2009)
[11] electronicecircuits.com/circuit-tutorials/ic-prog-programmer-settings-and-programming-pic: IC PROG Programmer Settings and programming PIC (Viewed: May 2011)
[12] duracell.com/oem/Pdf/new/MX1500_US_UL.pdf: LR6 Specification (Viewed: May 2011)
[13] docs-europe.electrocomponents.com/webdocs/0806/0900766b80806cfb.pdf: PIC18F4550 Datasheet (Viewed: May 2011)
[14] docs-europe.electrocomponents.com/webdocs/0b51/0900766b80b51800.pdf: 741-OP-AMP Datasheet (Viewed: May 2011)
[15] L.W. Turner, Electronics Engineer's Reference Book, 4th ed. Newnes-Butterworth (London, 1976)
[16] G.M. Sessler and J.E. West,"Self-biased condenser microphone with high capacitance", *Journal of the Acoustical Society of America* 34: 1787–1788 (1962)
[17] Eddie Ciletti,"Level Headed: Nominal Level (explained) plus an SV-3700 modification"(1998)
[18] frames-per-second.appspot.com: Compare frames per second: which looks better? (Viewed: May 2011)
[19] W. T. Cochran and J. W. Cooley et. al., "What is Fast Fourier Transform?", IEEE Transactions on Audio and Electroacoustics, vol. AU-15 No. 2, pp. 45-98, June 1967
[20] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Of Comput., vol. 19, pp. 297-301, April 1965
[21] P.A. Lynn, in: An Introduction to the Analysis and Processing of Signals, 3rd Ed., Macmillan (1989)
[22] A. H. Kaiser, in: Digital Signal Processing using the Fast Fourier Transform (FFT), GRIN Verlag (1997)
[23] G. H. Jonah and I. J. Whale, in: *Quantum Fluids*, ed. By D.F.Brewer (North-Holland, Amsterdam, 1973), p.146.

Appendix A: Research conducted using Facebook to determine the existence of a market for an electronic instrument tuner.

Option	Count / Dimensionless
I can tune using an electric tuner.	5227
I can tune using another reference (e.g. a piano)	4558
I can tune by relative pitch	3938
I can tune without any reference, just using my ears	2165
Total	15888
Participants were allowed to choose multiple options. When a selection of participants were interviewed, several of them who claimed they could tune without reference stated that they would not go on stage or perform for an audience without checking their tuning with an electric tuner.	

Appendix B: The highlighted column indicated the key properties of the microcontroller used.

TABLE 1-1: DEVICE FEATURES

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Data Memory (Bytes)	2048	2048	2048	2048
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/ Compare/PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1	1	1	1
Streaming Parallel Port (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Comparators	2	2	2	2
Resets (and Delays)	POR, BOR, $\overline{\text{RESET}}$ Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT	POR, BOR, $\overline{\text{RESET}}$ Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT	POR, BOR, $\overline{\text{RESET}}$ Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT	POR, BOR, $\overline{\text{RESET}}$ Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

Appendix C: The full schematic diagram of the product.