So far the following has been completed:

1. Interpreted a sine wave using a digital input.
2. Determined the frequency of the sine wave at the input.

In order to apply the above to a guitar tuner the following tasks are foreseeable and will need to be completed. Undoubtedly other tasks required will become apparent as the project progresses and problems are encountered.

**1. Determine the frequency of a note generated from a real guitar:**

So far we have determined the frequency of a clean sine wave generated from a signal generator. Unfortunately the signal from a guitar may not be as clean. It will contain harmonics that are inherent in a signal that is generated from physical motion, it may also contain noise. We know that the frequency range of the open strings of a guitar is between 82Hz and 330Hz so we can design a crude low pass filter to remove the high frequency noise and harmonics. This will probably just need to be a capacitor that has a low capacitive reactance at a frequency of say 1kHz to be placed between the signal and ground. This would effectively short the higher frequencies to ground.

The amplitude of the signal may also be an issue. In order to exceed the high level input threshold of the arduino we may need to boost the signal or design some circuitry that interprets the signal using different thresholds. The following circuit may be suitable:

This is just a rough idea of what this circuit will probably look like. The 200k potentiometer determines the high level voltage threshold. If the input signal exceeds this the comparator will give a 5V output. This behaves like an adjustable schmitt trigger.

**2.Determine the torque required to turn a machine head on a guitar and find a suitable motor / gearbox:**

This can be determined by using a torque wrench on the machine head. We need to know how many newton meters (NM) it takes to turn the machine head to specify a suitable stepper motor. Being as it is easy to obtain stepper motor / gearbox assemblies with significant gearbox ratios it would probably be suitable to have a large gearbox and therefore substantially more torque than required. It is still worth measuring the torque required to ensure that we have a reasonable factor of safety above this. Using a high ratio gearbox would also give the added benefit a lower polar step displacement and hence higher resolution capabilities.

**3.Drive a stepper motor with the arduino board**

There is already a stepper motor library available for the arduino board on the arduino website so the software side should be trivial. We will however need to implement some electronics to drive the stepper motor. This will just be a simple H-Bridge amplifier as shown below:

<http://arduino.cc/en/uploads/Tutorial/bipolar_schem.jpeg>

**4. Determine the ratio of stepper motor counts to change in frequency:**

In order to ensure we have the resolution needed to tune the guitar to within 1hz we need to determine the amount of steps required to result in a frequency change of 1hz. This could be done by creating on a guitar, measuring the frequency and turning the machine head to result in 1hz of frequency change. This may be impractical as 1hz may only require a very small rotation. To make this more practical and even more applicable we could attach a geared stepper motor and tweak that in motor steps. This would give us our value in motor steps. As a rule of thumb we should aim to have a step displacement of 10% of the smallest step size required, in this case 100mHz.

**4.Create a prototype of the hardware:**

Now that the torque and resolution requirements have been determined a suitable stepper motor and gearbox can be specified. Once this is done a prototype of the system can be designed and created. The prototype will be required to test the final functionality of the system.

**6.Implement a control loop in the arduino:**

Now that the prototype has been completed and both an input device (guitar frequency) and an output device (Stepper motor) are present a control system can be developed and implemented. There are a few ways to do this so some experimentation will be required to determine the best method. We could for example implement a full PID loop and which adjusts the frequency to a frequency velocity profile, ie, we generate a profile of how the frequency transitions between the start frequency and end frequency and follow that. This may give us the best performance but may not be completely necessary, we could just have an algorithm that tweaks in 1hz steps until the desired frequency is achieved.

The control system will also need to decide what frequency to tune to, for example, if a string is plucked the system must determine which string has been plucked and therefore what is the desired frequency of that string.

With the above tasks completed the prototype should be functional. A final product can then be designed and fabricated. More tasks will come to light as problems are encountered.