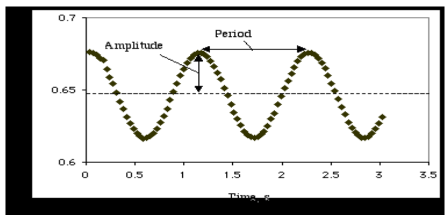
Sound Part 3: All That Noise

Is it suprising that the quality of the sound we have made using our microcontroller is inferior to that produced by a $2 greeting card? In this exercise, we will explore sound and what we can do to change the quality of the sound we produce with our microcontroller.

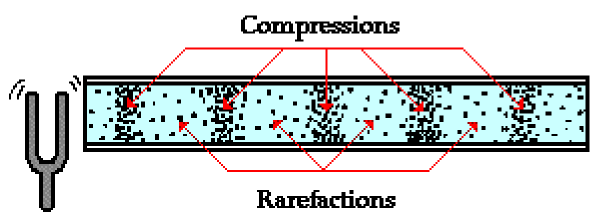
Sound

What is sound anyway? If a tree falls in the woods and nobody hears it---did it make a sound? Ok, that's not really the question we will address here, but, before we can improve the sound made by our microcontroller, we need to understand something about the characteristics of sound.

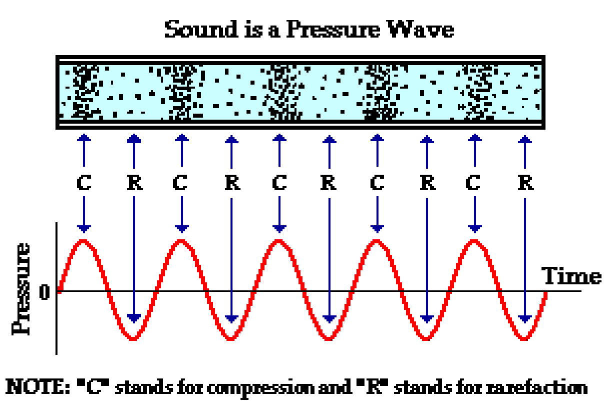
Sound is a wave. Not a wave like this:



But rather, a wave like this:



a longitudinal pressure wave that is detected by your ears. The changes in pressure detected by our ears occur at regular time intervals that can be described as "periodic." A plot of pressure versus time could (for certain sounds) appear as a sine curve. The peak points of the sine curve correspond to compressions; the low points correspond to rarefactions; and the "zero points" correspond to the pressure which the air would have if there were no disturbance moving through it.



Sound is make by movement because movement produces presure waves in the air. When you tap your finger on a table, the vibration causes the table to shake a bit which sends vibrations out through the air around the table. Most movement has associated vibration and vibration implies periodicity like the sinusodial sound wave pictured above. Indeed, almost any sound can be produced by mixing together sound waves with different periodicity or *frequency*. But wait---let's slow down a little.

The sinusoidal sound wave depicted above is produced by something, maybe a guitar string, virbrating at a specific frequency. Frequency is defined by the number of oscillations a sound wave produces per second. If the time range of the graph above were one second, the the frequency of the above wave would be 5 cycles/sec. Electrical engineers will quickly recognize the concept of "cycles per second" in a wave as being measured in hertz. Using this ternimology, the frequency of the above wave is 5 Hertz abrivated 5Hz.

The frequency of a sound wave determines it's *pitch* which defines its musical note. Try it out! Using your Windows account (the plugin used by the applet is not supported on Linux), go to http://www.phys.unca.edu/demos/demos_sound.asp and play with the Sine Wave applet to experience the relationship between frequency and pitch. Note that you can also adjust the *amplitude* of the sounds wave to change its volume.

Referring back to our work in Sound Part 2, what is the frequency of “c” and “C” in the program **Melody**? These notes are said to be an *Octave* apart. What is the meaning of the term *Octave*?

Timbre

The pitch of a sound is relatively easy to understand; it's simply controlled by the frequency of the sound, which is how many cycles the sound wave makes per second. Similarly, the volume of a sound is also pretty easy to understand. But there's a fundamental characteristic of sound that's perhaps more difficult to understand: Quite simply, it's what the sound actually sounds like. This has been given various names, including the "tone" of the sound (although "tone" is sometimes used as a synonym for "pitch"), or "quality" of the sound, but the technical term is the **timbre** of the sound.

Timbre is what makes difference instruments sound differently when they play the same note. It's the shape of the sound wave that makes all the difference. If you could see the sound waves formed by middle C on a piano and compare them with the sound waves made by middle C on a guitar, you'd see that although the two waves have the same frequency, the shape of the waves is different. The shape of the sound wave is what constitutes the timbre of the sound. Again, you can see and hear this on http://www.phys.unca.edu/demos/demos_sound.asp. This time try the Wave Form applet to experience timbre.

You can analyze sound waveforms using *Fourier analysis*. Although serious Fourier analysis is fairly involved mathematically, the basic idea is something like this: Any waveform which is not a sine wave can be produced by imposing several sine waves on top of each other. If you produce a wave using an electronic oscillator, it's a sine wave at one frequency. If you produce a note of the same frequency with a guitar string, the resulting sound is not a simple sine wave, its a more complex waveform. Nonetheless, this complex wave shape can be reproduced by mixing together several sine waves of different frequencies. Again, try using the Fourier Synthesizer Applet at http://www.phys.unca.edu/demos/demos_sound.asp. Try to creat a square wave and notice the effect of the high frequency sine waves.

In the program **Melody**, how did we create a square wave of a specific frequency and why is that the only wave form we could create?

Better Sound

Recall for the previous exercise that we produce our melody by sending “square waves” of current at 5 Volts to a speaker. When we send electrical current through a speaker, it vibrates and creates sound. Given what you have now learned about sound and programming, how are we controlling pitch, amplitude, and timbre in the program **Melody?** Be prepared to answer this question in a class discussion.

How can we improve the sound made with our microcontroller? We could change all three characteristics of the sound waves we produce: the pitch, the amplitude, and the timbre, but that will have to be the topic of another lab. That's all for now.