Homework #2: Audio IO, Threads, and FFTW

Design a Windows Phone to fulfill the following specifications. The specifications are presented in a series of phases, however it is not necessary for you to complete each phase and submit them separately. Only the final phase must be submitted, this separation into phases is to provide stepping stones toward the final product.

Phase 1: Develop a Windows Phone application that reads in audio from the mic, and writes that audio out to the speaker, in real time using WASAPI. Recall from the lecture notes all this entails: You must create a C++/CX component to house the WASAPI code, initialize your devices and start a C++ thread to perform the audio processing. Be aware of the differences in format between the input audio and output audio streams.

Phase 2: Link in the FFTW library to perform realtime Fourier analysis on the audio stream. Search the complex-valued Fourier transform for the frequency bin of highest magnitude, and output the frequency this Fourier transform bin corresponds to (Recall from your DSP knowledge how to convert a DFT bin index to a frequency in Hertz). Because we have not yet studied C++ → C# communication yet, the suggested manner in which to output this information is to create a C# DispatcherTimer object that reads the "current loudest frequency" value from the C++ code periodically. We will learn better ways to do this next week.

You should use the phone hardware for testing as much as possible, as parameters such as the length of audio being passed in, etc. will vary between the emulator and the actual device. Additionally, the emulator on the remote server will not be useful for audio input/output as the audio does not survive the remote desktop transition.