Real-time acoustic FIR filter suite using the LPC4350

Steve Sabram, aHandSoft, Inc. February, 2012

I have been a fan of anything that changes sounds since my school days be it musical instruments, controls on radios or television or record players, acoustic fittings in rooms and so on. My passion for this and electronics has created a professional interest in digital signal processing. The works of Bernard Gold and Charles M. Rader pioneered early digital signal processing in the late 1960's at MIT. Many consider their original book on the subject *Digital Processing of Signals* (Lincoln Laboratory Publications, 1969) to be the cornerstone of DSP theory and practice. I am fortunate to have a copy of this book in my collection and I refer to it occasionally for different projects when an explained theory of operation is under scrutiny.

These first DSP research projects ran on general purpose mini-frame-grade computers that were the size of a small closet and packed a punch to one's electricity bill. Thus, the first DSP algorithm implementations were in software running on these general purpose computers. From this early research, specific semiconductor IC designs for the processing of digital signals involved specialized circuits and "pipe-lining" sampled data read from an analog-to-digital converter (ADC). Thus, hardware implementations of these algorithms were the most cost effect offerings in the first commercial DSP devices in the 1970's and into the 1980's.

As semiconductors become faster and more power efficient, the implementation of these DSP algorithms now go full circle back to the original research implementations on general purpose CPU circuits. Many desktop computer software packages processing digital signals are now quite affordable. With the implementation of the NXP Semiconductor LPC4300 series of ARM Cortex-M4 CPU's, these DSP algorithms implement for mobile DSP solutions challenging cost and performance of dedicated DSP integrated circuit hardware. ARM publishes the CMSIS DSP library providing ease of implementation of fundamental DSP functions for a wide variety of ARM Cortex-M3 and Cortex-M4 processors.

This example project implements acoustic range, finite infinite response (FIR) filters using the LPC4350 demo board and the ARM CMSIS DSP library. A sound resource of a science fiction "zap gun" plays out the headphone jack of the demo board. This sound sample is great for demonstration since it has many low-, mid-, and high-frequency components along the acoustical band. You play the sound resource by touching the capacitive touch buttons on the

Download the sample code here

demo board as explained by the menu shown on the demo board's LCD. It is best to listen to the sound with a headphone or ear buds. Each of the four buttons plays the same sound resource differently:

1) Raw – The unprocessed sound plays according to its format (sample rate of 44.1 kHz, 16-bit sample and mono sound).
2) Low Pass – filtered through a low pass, Butterworth filter with a cutoff of 5 kHz, similar to the bandwidth of a common analog telephone. When played, notice that high pitch components are removed similar to the sound heard over a telephone.
3) High Pass – filtered through a high pass, Butterworth filter with a low frequency cutoff of 8 kHz. Notice that the sound is lower in volume since only the upper harmonic components are played.
4) Backward Mask – The sound resource sample plays in reverse order. The "Zap!" is now a "Zoup!"

The digital filters were designed with a favorite public domain tool, WinFilter (http://www.winfilter.20m.com/) accompanying this example. With exposure of this free DSP design tool via NXP, I hope its author expands its features.

This demo can be expanded in many ways to utilize more resources on the LPC4350 demo board. Here are a few that come to mind:

- Stream input via the microphone input instead just playing a sound resource.
- Play from sound files resident on an SD card read by a file system.
- Change the filters at run-time via a filter description file that can reside on the SD card.
- Expand the LCD driver to draw vertical lines, making for a real-time graphical display of the sound waveform.
- Utilize the CMSIS DSP library's FFT function to show a real-time, frequency domain display of the sound played utilizing the updated LCD driver.
- Use the CMSIS DSP library's Frequency Mixer function to merge two sound sources such as a microphone input and a "background" sound file stored on the SD card.

In conclusion the NXP LPC4350 demonstrates great software implementations that challenge the utility of hardware DSP solutions.

Date of release: February 2012
Published in the USA