A Teaching Tool for Understanding Different Audio Filters

Alireza Kavianpour, William Nguyen, and Paul Wagner
DeVry University
901 Corporate Center Dr.
Pomona, CA 91768-2642

Abstract

A teaching tool for understanding different audio filters is developed. Texas Instrument Digital Signal Processing (DSP) board, 16-bit Motorola micro-controller (MC68HC12), and C++ programming are used to develop a teaching tool for explaining different types of audio filters. One application of this tool is for eliminating audio feedback in a Public Audio (PA), or sound reinforcement system. An audio noise occurs when an input device (microphone) picks up its own signal from the output device (speaker) thus creating a loop. The result is a loud noise coming from the speakers. With the use of this system this type of problem will be eliminated.

In this new tool, user through keypad can select a specific filter. The system then takes an incoming analog signal and converts it to digital form. The digital signal is then "scanned" using a Fast Fourier Transform Algorithm (FFT) to find the unwanted frequencies. These offending frequencies are then filtered out. Next the processed digital signal is converted back to analog and sent to the speaker. The system has the capability to display data in a graphic form. A simple procedure could be carried out to update the software and add more features like digital effects (delay, and pitch shifting). The ability to make these changes by this software package also makes expansion very inexpensive and affordable. Most importantly this system can be widely used for educational purposes.

Key Word: DSP, FFT, Filter, Low-Pass, High-Pass, Band-Pass, Micro-controller

Introduction

Audio feedback has been a problem for many audio systems that has real-time input. Eliminating this type of feedback or noise requires lots of time and experience in this field to achieve the task; also it requires changing in the physical component layout.

Audio feedback happens when there is public address system such as sound system at concerts or at home. These feedbacks cause unpleasant feeling and possible damaged to your ear. To prevent it from happening, it is required to have professional background in this field and lots of time in physical setup. Because of cost, most of the public sound system at concerts or at home does not come with audio feedback suppressors. There are many different ways to eliminate feedback, however, using DSP is the most efficient, and convenient way. This method requires simple setup by the users.

Design Specification

Figure 1 displays different parts of this project. In this project 16-bit Micro-controller 68HC12, DSP board by Texas Instruments, Liquid Crystal, Keypad, Speaker, and a Microphone are used. The audio feedback suppressor is designed to do real-time filtering noise of an audio signal using DSP. The 68HC12 is used as the mean of interfacing with the users and display system status. The 68HC12 interfaces with a keypad and a LCD. The keypad is for the user to select the filtering modes to improve the quality of the audio output. The LCD is to display the welcome message as the system power up and the user selection mode. In this project 68HC12 Micro-controller 68HC12 is used to control the DSP via user switches. The DSP board is for real-time signal processing by calling different types of filters base on users' selection.

The software portion of this project is involved of writing different types of filters and spectrum analyzer using Fast Fourier Transformation method. Programming tool used C++ programming. Other software languages such as System View, MATLAB and Code Composer were very

Helpful during the development process. With the help of this software, all the filters were designed and verified prior to implementing them in the DSP system.

Motorola 68HC12 Micro controller

The CMD-9S12DP512 is a low cost full-featured development system with on-board USB BDM development port for the free scale MC9S12DP512. The board provides operation in Single-Chip Mode or Expanded Wide Mode with memory bus available for expansion and development memory access. System is supplied with a powerful IDE software suite (AxIDE4) with GNU “C++” compiler and assembler, and the USB BDM debugs support for a seamless development environment. The integrated USB-BDM provides background development control of the HCS12, optional board input power, and optional serial communication via the USB port for

Single point connection to host computer. Development software is Windows XP compatible and provides source code level development support, selection of Target operation modes, Target code loading, Target flash memory programming, and multiple window views for Target registers and data. When not required for development use, the USB BDM port may still be optioned as a USB Serial port. The Axiom CMD-12DP512 development system provides a full seamless hardware and software application development environment. The onboard BDM circuit allows the user to locate code in the On-Board RAM or HC12 Flash, set Break Points, Step code, and display or modify registers and / or memory. After application is operational the user may apply the board for dedicated operation of new software. I/O Port headers provide easy connections to breadboard area or user option components with 24GA solid core jumper wires (provided). Pin headers may also be installed to apply IDC ribbon or other connections. The Keypad and LCD ports are compatible with the Axiom HC-KP 16 key (4x4) keypad and HC-LCD 80 character (4x20) LCD module. Example software is provided to operate all user components and accessories.

CML12S-DP256 Specifications

Upward code compatible w/ 68HC11

4K Bytes EEPROM

512K Byte Flash EEPROM

14K Byte SRAM

2 Enhanced SCI Ports

3 SPI Port (Synchronous Serial)

5 CAN 2.0 A or B Interface

Two 8 Channel 10 Bit Analog Converters

Background Debug Port

Enhanced 16 bit Timer w/ 8 channels

16 Bit Pulse Accumulator

8 PWM Channels

Two 8 bit Key Wake-up ports

PLL Clock Oscillator Support

RTC and COP features

Up to 91 I/O

4Mhz reference oscillator for up to 24MHz operation.

External Memory: 256K Bytes (128K x 16) SRAM

COM1 Port – HC12 SCI0 w/ RS232 and DB9S connector

COM2 Port – HC12 SCI1 w/ RS232 and 3 pin header

INDICATORS – Power and RESET.

BUS-PORT – 40 Pin Socket Header

MCU I/O PORT - 60 pin Socket Header

Analog Port – 20 pin Socket Header

CAN PORT – CAN 0 I/O with 1M Baud Transceiver

LCD Module and Keypad Ports

Solder less Prototype Area and Connections

Easy Power Connection and Tap points'

Back Ground Debug (BDM) Port – 6 Pin standard

Power Specifications: 7 to 25VDC input to 5V Power Supply

Operating Power: 60ma @ 5V

DSP Board Specifications

Very long instruction word (VLIW) architecture

Load-store architecture

32 32-bit general-purpose registers

900 million floating-point operations per second (MIPS)

6.7-ns cycle time

Up to eight 32-bit instructions per cycle

32-bit address range

Little- and big-Indian support

Bit-counting

Memory/peripherals

L1/L2 memory architecture:

32K-bit (4-K byte) L1P program cache (direct mapped)

32K-bit (4K-byte) L1D data cache (2-way set-associative)

512K-bit (64K byte) L2 unified map RAM/cache (flexible data/program allocation)

32-bit external memory interface (EMIF):

Glue less interface to synchronous memories: SDRAM and SBSRAM, and EPROM

Enhanced direct-memory-access (EDMA) controller

16-bit host-port interface (HPI) (access to entire memory map)

Two multichannel buffered serial ports (McBSPs)

Two 32-bit general-purpose timers

Flexible phase-locked-loop (PLL) clock generator

IEEE-1149.1 (JTAG) boundary-scan-compatible for emulation and test support

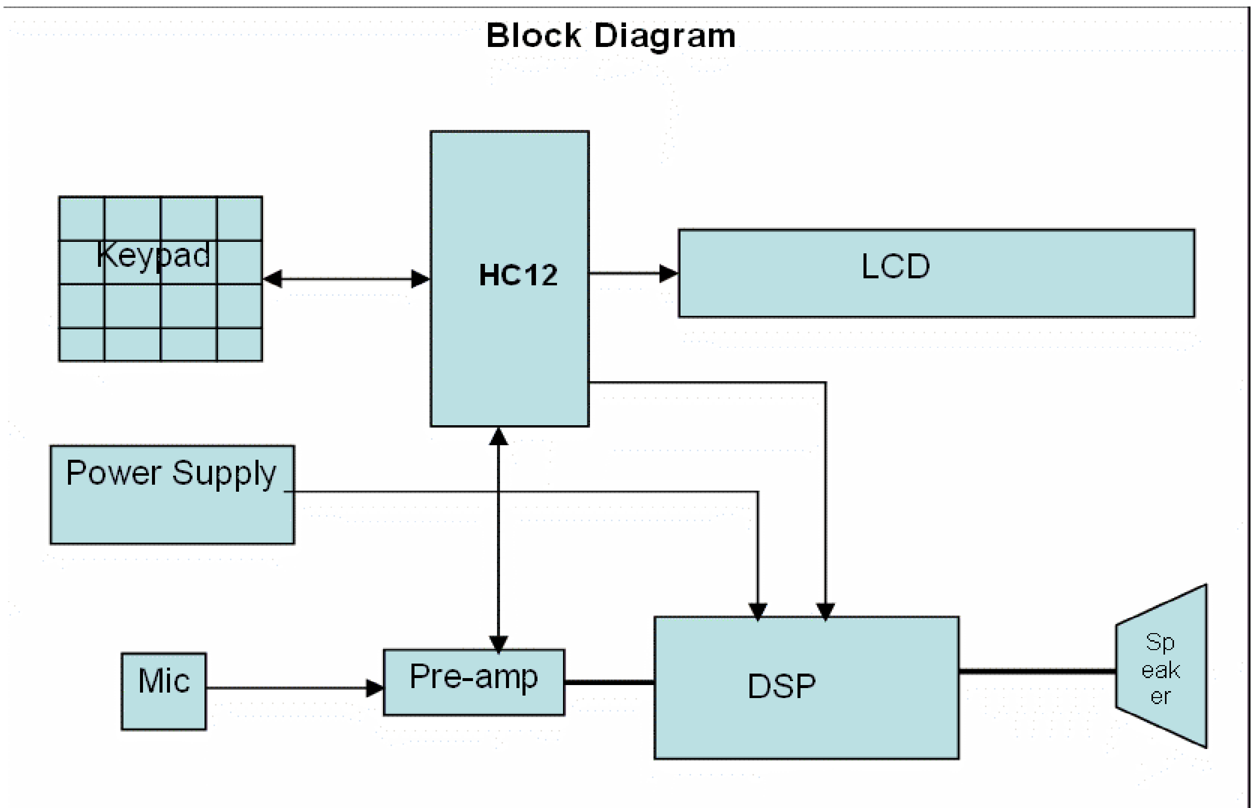

Figure 1. Block Diagram of Audio Filters

Test Results

This project was divided into two different sub-systems – 68HC12 and DSP sub-systems. The 68HC12 section was tested with the keypad and LCD to ensure that this subsystem functions as required. For example, if the users enter “1” on the keypad, the low filter message will display on the LCD and a logic “1” is expected to be at an output port. As a part of the sub-system test, LEDs were installed at the output to verify the operations. For the DSP sub-system, an audio input signal is applied to the audio input port. By selecting different user switches, the audio output was monitored in both time domains using oscilloscope and in frequency domain using spectrum analyzer. The test was done for four different types of filters low-pass, high-pass, band-pass and band-stop. After both sub-systems were fully tested, system test was also done to ensure the communication between the 68HC12 and DSP. Figure 2 displays one of the results.

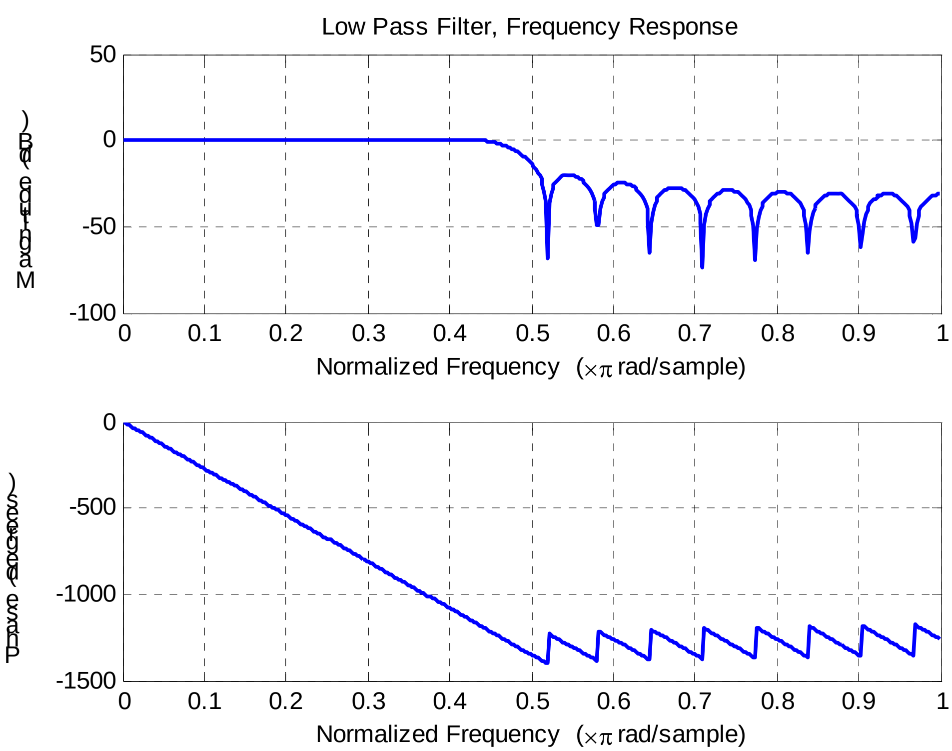

Figure 2. Results of audio filtering

Software Descriptions

Motorola assembly language was utilized to initialize and open I/O port connections. Keypad is connected to the 68HC12 via Port H, which is the main component to interface with users for user options. LCD is connected to the LCD port, which acts like an information center for the user to let the users from filter mode. Port T is used to interface with the DSP user switches. Depending on the user switches selection; the desired filter mode will be selected.
The C++ program for the DSP constantly monitors at the user switches and response to them as they got change by the 68HC12 program. These user switches represent as binary number in the C++ program of the DSP. Depending on the binary weighted of the switches, the program will call designated filter to filter the input and send audio.

Summary and Conclusion

A teaching tool for understanding different audio filters is developed. Texas Instrument Digital Signal Processing (DSP) board, 16-bit Motorola micro-controller (MC68HC12), and C++ programming are utilized to develop a teaching tool for explaining different types of audio filters. This tool is utilized for eliminating audio feedback. Audio noise occurs when an input device (microphone) picks up its own signal from the output device (speaker) thus creating a loop. The result in a PA system is a loud noise coming from the speakers.

References

1. Embree, Paul M. and Damon Danieli. *C++ Algorithms for Digital Signal Processing 2nd Ed.* Upper Saddle River: Prentice Hall PTR, 1999.

2. Lane, John, et al., Eds. *DSP Filters*. Indianapolis: PROMPT Publishing, 2001.

3. Welch, Thad B., Cameron H.G. Wright, and Michael G. Morrow.*Real Time Digital Signal Processing-From Matlab to C with the TMS320C6x DSK.* Boca Raton: CRC Press, 2006.

4. Texas Instrument, Inc. 2007. http://www.ti.com/

5. Axiom Manufacturing. "Education Center: 68HC12." http://axman.com/education/edc_hc12.html

6. Han-Way, Huang. MC68HC12 An Introduction. Thomson Delmar Learning. 2006.

7. Kavianpour, Alireza "CORDIC Calculation" International Congress of Mathematicians, Aug. 1986, Berkeley, U.S.A

Alireza Kavianpour

Dr. Kavianpour is a professor at DeVry University, Pomona, CA. He received his Ph.D. from University of Southern California. His research interests are in the field of microprocessors' applications. He is the author of over thirty research papers.

William Nguyen

Mr. Nguyen is an EET student at Devry University, Pomona, CA

Paul Wagner

Mr. Wagner is an EET student at Devry University, Pomona, CA