Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx-14), Erlangen, Germany, September 1-5, 2014

FREEDSP: A LOW-BUDGET OPEN-SOURCE AUDIO-DSP MODULE

Sebastian Merchel and Ludwig Kormann,

Chair of Communication Acoustics,
TU Dresden
Dresden, Germany
sebastian.merchel@tu-dresden.de

ABSTRACT

In this paper, the development and application of a universal audio-DSP (digital signal processor) board will be described. It is called freeDSP. Our goal was to provide an affordable real-time signal processing solution for researchers and the do-it-yourself community. Easy assembling and simple programmability were the main focus. A solution based on Analog Devices' ADAU1701 DSP chip together with the free graphical development environment SigmaStudio is proposed. The applications range from active loudspeaker compensation over steerable microphone arrays to advanced audio effect processors. The freeDSP board is published under a creative commons license, which allows the unrestricted use and modification of the module.

1. INTRODUCTION

The development and application of DSP boards has been associated with high cost, difficult soldering and low-level programming in the past. The current project investigated if this is still the case. Therefore, the different steps from the selection of a suitable DSP chip to final application examples will be presented in the following. Before describing the design of the DSP module, basic demands and commercial alternatives will be discussed.

1.1. Requirements

It was decided that the board should have at least two audio inputs (stereo) and more than two outputs (e.g., to implement active cross-over networks for multi-way loudspeakers). A scalable number of input and output channels would be preferable. Additionally, the board shouldn't be too difficult to solder with as many through-hole components as possible. The overall cost should be as low as possible. An easy to learn programming language would be beneficial to allow quick experimentation and a short familiarization time. Finally, the overall processing power and functionality should allow for easy adaptation to different application scenarios.

1.2. Available DSP solutions

Few commercial solutions exist on the market. E.g., the company MiniDSP offers several easy to use DSP boards, mainly for audio filtering applications [1]. In order to configure the hardware, special software plugins have to be purchased separately - which increases the overall price. Another option is the use of evaluation boards from semiconductor companies. They are usually expensive and contain only the minimal support logic needed to learn programming the DSP or microcontroller. Several other companies provide costly audio amplifiers or loudspeakers with build-in DSPs. In those cases, configuration of the signal processing is quite limited. Many more DSP products exist which are specialized on one specific purpose like room compensation, audio effects or feedback suppression in a public address system. A low-cost open-source DSP solution provides much more flexibility. It enables the curious student to experiment with digital signal processing. Hobbyists can implement and share various projects using the same platform. Additionally, such a board could be easily adapted for professional prototyping or research projects.

2. DESIGN

First of all, a suitable DSP or microcontroller had to be selected.

2.1. Selection of the DSP

The requirements listed above are met by the SigmaDSP series from Analog Devices. E.g., the ADAU1701 has two integrated AD converters (inputs) with a sampling rate of 48 kHz and a bit depth of 24 bit. Four extra bits are added internally for better signal processing (28 bit total). Before outputting of the signal these extra bits are removed again. Four build in output DA converters can be used for various application scenarios. Additionally, it is possible to connect external AD/DA converters via the I^2S interface. Therefore, a maximum of 10 inputs and 12 outputs is realizable. A functional block diagram of the ADAU1701 is shown in Figure 1. Although the ADAU1701 comes in an SMD package (48-Lead LQFP), it can still be soldered by hand.

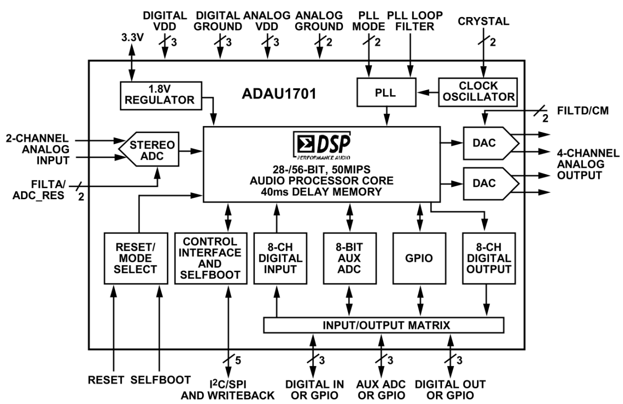

Figure 1: Functional block diagram of the ADAU1701 from Analog Devices [2].

2.2. Board Layout

The design fits on a two-layer board that has a width of 100 mm and a length of 80 mm. The components used on the board are mainly through-hole components, which are easier to solder. Only two SMD parts appear on the board, the ADAU1701 and AD8608, which is the amplifier used within the active output filter. Figure 2 shows the schematic of the audio-DSP board. The freeDSP supports a wide range of input voltages from 5 V up to 24 V. A voltage regulator supplies the 3.3V needed for all circuitry on the board. Additionally, the board might be powered over the USBi connector, which can be used for programming the processor and the EEPROM. Two integrated audio inputs and four audio outputs of the DSP can be accessed via RCA-connectors. The connectors are located on one side of the board for easier access. Figure 3 shows a photo of the assembled freeDSP in the current version. Jumpers allow fast and easy configuration of the input voltage range. One out of three user-defined configurations can be chosen individually for each input. This allows for easy adaptation to different signal levels. Active output filters are used to provide better audio performance. GPIO (general purpose input/output) pins, I²S and I²C interfaces are accessible via the GPIO header. Additional boards can be easily stacked on top of the freeDSP to add more functionality. This includes buttons, potentiometers, additional inputs and outputs or any other peripherals.

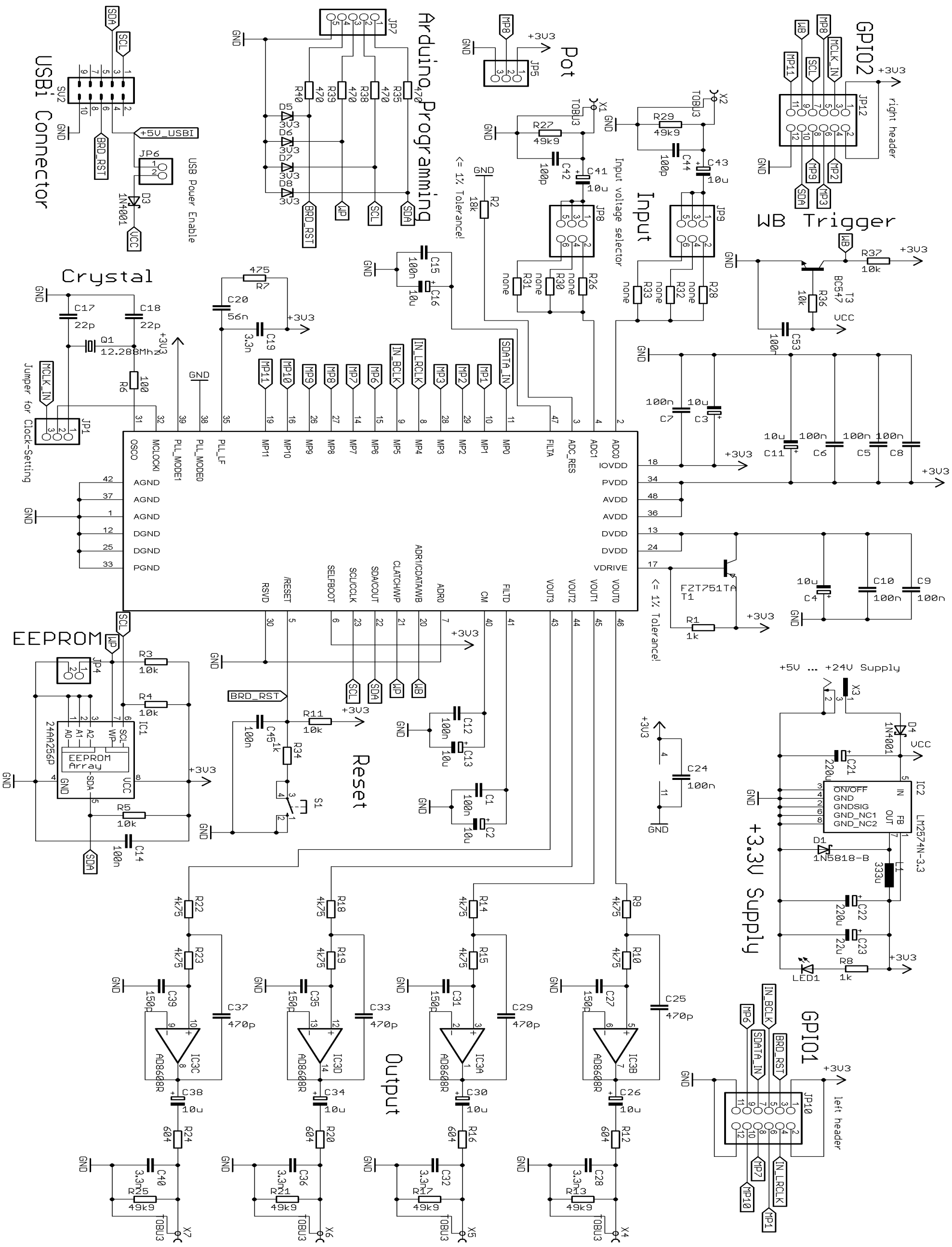

Figure 2: *Schematic of the freeDSP board version 0.3.*

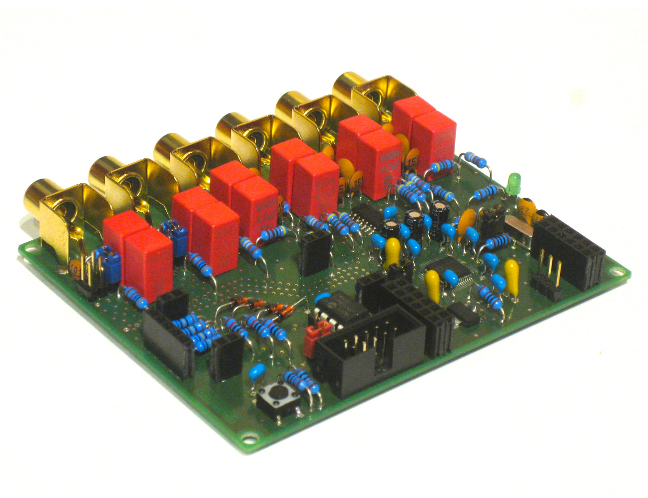

Figure 3: *Completely assembled freeDSP module (V 0.4).*

2.3. Programming

To get started with a SigmaDSP no advanced programming skills are necessary. A free development environment called SigmaStudio is available for Windows. It can be downloaded from the Analog Devices website [3] with no charge (account needed). The programming model is function-block based – comparable to other graphical programming languages like PureData [4] or Max/MSP [5]. Many prebuilt blocks (e.g., filters, compressors, effects, or logic) can be placed in the signal path via drag and drop. If the included libraries do not have the functions needed, low-level blocks, such as multipliers and delays, can be wired together to create custom algorithms. Figure 4 shows a simple example patch with two input channels. The incoming signals are filtered using an equalizer with adjustable frequencies and filter characteristics (band pass, low shelf, …). Finally, the summed up signal is played back via two outputs.

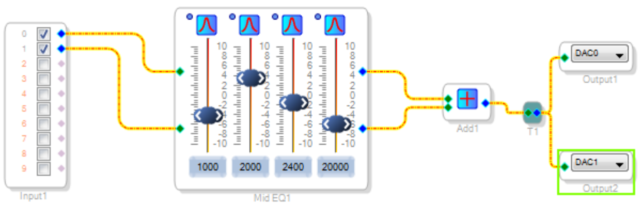

Figure 4: *A simple example patch in SigmaStudio.*

The final program needs to be compiled and transferred to the DSP. Therefore, an USBi programming adapter is available from Analog Devices. With this adapter the board can be controlled in real-time from the PC using SigmaStudio. Alternatively, the program can be written to the onboard EEPROM (electrically erasable programmable read-only memory) and booted in stand-alone mode. This offline programming can be done using a microcontroller, e.g. an Arduino or Raspberry Pi. Alternatively, a dedicated I^2C EEPROM programmer can be used, which supports the EEPROM mounted on the board.

2.4. Cost Calculation

The overall cost of the freeDSP depends on the number of boards build at the same time. Lets assume a total of five boards in the following. The exemplary calculation in Table 1 provides an overview of the expected costs per module.

Table 1: *Component overview with price examples.*

Components	**Price / Module**
Board	5 $
DSP ADAU1701	9 $
Resistors	3 $
Capacitors	4 $
Semiconductors	8 $
Miscellaneous	10 $
Total	**39 $**

Additionally, a suitable programmer is needed once. To offer real-time programmability the USBi adapter from Analog Devices is recommended, which is currently sold for approximately 80 $. If the application is intended to run the board only in self-boot mode, a cheaper programmer can be used.

3. APPLICATION EXAMPLES

The board was extensively tested and has been used successfully in several applications so far. Two exemplary projects will be described in the following.

3.1. Active Loudspeaker Concept

Active loudspeakers are speakers with build in amplifiers. Typical examples are subwoofers or studio monitors. For educational purposes, a desktop loudspeaker was designed and built in our audio lab. It was decided to implement a two-way system and to reduce the internal volume of the cabinet because of aesthetical reasons. A picture of the speaker is shown in Figure 5.

Figure 5: *Small active loudspeaker prototypes with digital crossover, loudspeaker protection and bass enhancement algorithms. An early version of the freeDSP can be seen in the foreground.*

An active crossover was built into the speakers using the freeDSP. This enabled fast and simple parameter modifications (like corner frequency or filter type) during the testing phase. A screenshot of one of the prebuild crossover blocks is shown in Figure 6.

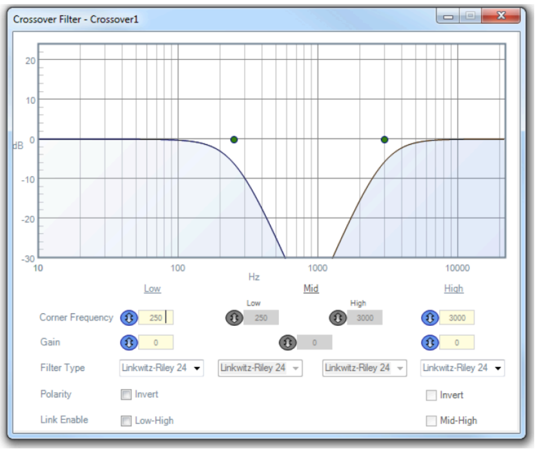

Figure 6: *Adjustable crossover settings in SigmaStudio.*

Further, the freeDSP was used to equalize the frequency response of the system. Therefore, impulse responses have been measured. Matlab was used to create inverse filters with different algorithms. The resulting IIR coefficients have been applied using a biquad filter bank in SigmaStudio. Alternatively, free software could be used for measurement and filter creation, e.g., the Room EQ Wizard (REW) by John Mulcahy [6]. However, there is also an easy to use auto-EQ filter designer available in SigmaStudio.
Additionally, the bass output was improved and the drivers were protected from overload. Experiments were carried out with dynamic bass boost, which provides variable gain for low frequencies dependent on input signal level.
If the general purpose input and output pins are taken into account, many more examples are imaginable. E.g., the GPIO pins of the module could be used to power down the loudspeaker amplifiers when no signal is present. Another useful option would be to connect configurable dip switches and a volume control potentiometer to the board.
The application of the freeDSP in this active loudspeaker example was intuitive and successful. However, also more complex projects can be realized.

3.2. Vibrotactile Feedback for an Electric Violin

When playing a violin, the musician communicates with his instrument not only through his ears but also his fingers, cheek, shoulder and eyes. While playing he uses multiple sensory channels, which are provided by different modalities, such as auditory, tactile, kinesthetic, and visual. In a research project, the influence of violin vibrations on the perceived quality of the instrument was investigated. Therefore, it was necessary to separately control the sound radiation and the vibrotactile feedback of the instrument. A Harley Benton electric violin was selected because of its non-resonant body. A vibration reproduction system was added using small electro-dynamic shakers mounted at the back of the instrument. Figure 7 shows the violin with one of the exciters mounted below the chin rest.

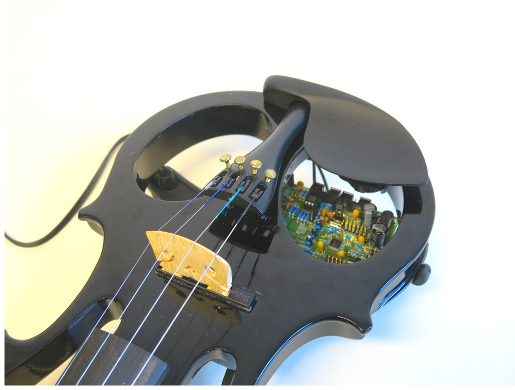

Figure 7: *Vibrotactile feedback for an electric violin using a small eletrodynamic on the backside of the chin rest. The freeDSP was integrated in the free space below.*

The freeDSP board was applied to generate audio-driven vibrations from the sound in real-time. First, the natural vibrations of a classical violin were simulated using the system. Additionally, different algorithms to modify the vibration signal were implemented and investigated, e.g., frequency shifting or dynamic compression. In Figure 8 an exemplary program patch can be seen. The results showed the importance of vibrations on the overall perception of the instrument and provide information on useful vibration features for the player-instrument interaction [7].

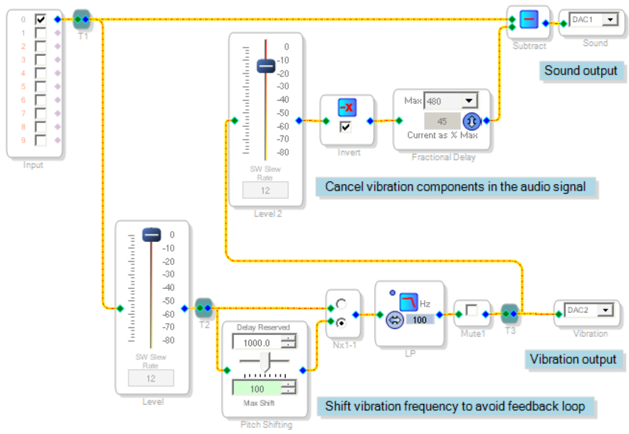

Figure 8: *Simplified program patch in SigmaStudio, showing one approach for vibration generation by frequency shifting the violin sound (bottom path). Additionally, the vibration crosstalk in the audio output is canceled actively using an inverter and a fractional delay (top path).*

In the area of vibrotactile feedback, many more innovative application scenarios for the freeDSP module exist. E.g., it can be used to process vibrations for a touch screen based audio mixer or a groove box [8]. It was found that different music instruments could be distinguished if vibrotactile feedback is rendered from the audio signal in an appropriate way. This helps to improve recognition of an audio source that is assigned e.g. to a specific mixing channel. Another example is the improvement of the concert experience in a music reproduction system. This is possible by generating seat vibrations in real-time from the audio signal using various signal processing algorithms [9].

4. SUMMARY

In this paper, the development and application of a universal low-budget audio-DSP board called freeDSP was described. The main features are summarized below:

- 2 integrated AD and 4 DA converters,
- Sampling rate 44.1 / 48 kHz (max. 96 kHz),
- Bit depth 24 bit (28 bit internal),
- Upgradeable up to 10 inputs and 12 outputs,
- Graphic oriented programming software available,
- Input voltage 5 to 24 V,
- 8 GPIO (general purpose input/output) pins,
- 4 AUX 8 bit ADC inputs (e.g. for potentiometers)
- Variable input range configurable via jumpers,
- Active filtered outputs.

The different application examples show that the freeDSP is a versatile tool that can be applied in various scenarios. The authors hope that it can be of use for others too. Therefore, the board layout is published under a creative commons license (CC BY-SA). Everyone is free to use the module as it is or to modify it as required. A complete list with all necessary components and the latest circuit board design will be published on our website [10]. A forum for discussions and the possibility for centralized buying are planned.

5. REFERENCES

[1] http://www.minidsp.com, Accessed February 27, 2014.

[2] Analog Devices, “Datasheet of the ADAU1701 Digital Signal Processor,” Available at http://www.analog.com/static/imported-files/data_sheets/ADAU1701.pdf, Accessed January 05, 2014.

[3] http://www.analog.com/en/content/cu_over_sigmastudio_graphical_dev_tool_overview/fca.html, Accessed February 27, 2014.

[4] M., Puckette, “Pure Data: another integrated computer music environment.” in *Proc. of the Second Intercollege Computer Music Concerts*, Tachikawa, Japan, 1996.

[5] M., Puckette, “The patcher” in *Proc. of the Computer Music Conference*, San Francisco, USA, 1986.

[6] http://www.roomeqwizard.com, Accessed December 07, 2013.

[7] M.E. Altinsoy, S. Merchel, S. Tilsch, “Perceptual evaluation of violin vibrations and audio-tactile interaction,” in *Proc. of Meetings on Acoustics (ICA 2013)*, Vol. 19, 015026, 2013.

[8] S. Merchel, M.E. Altinsoy, “Vibration in Music Perception,” *in Proc. of the AES 134th Convention*, Rome, Italy, 2013.

[9] S. Merchel, M.E. Altinsoy, M. Stamm, “Touch the Sound: Audio-Driven Tactile Feedback for Audio Mixing Applications,” in Journal of the Audio Engineering Society, 60(1/2), 2012.

[10] http://www.ias.et.tu-dresden.de/ias/kommunikationsakustik, Accessed March 13, 2014.