A binaural demonstrator on a single board computer

Fabian Schlieper, Lukas Aspöck, Michael Vorländer
Institute of Technical Acoustics, RWTH Aachen University, fabian.schlieper@rwth-aachen.de

Introduction

Digital signal processing (DSP) has become indispensable when recording, storing, transmitting and reproducing audio signals. Still, DSP hardware is expensive, may require expert knowledge and is not always suitable for generic filter prototyping.
Single-board computers (SBC) like the Raspberry Pi are powerful small devices available at a low price. Standard interfaces such as USB, Ethernet and even I^2S allow connecting arbitrary audio hardware – raising the idea to use them for audio DSP.
We evaluated some of these computers on their ability of real-time audio processing and developed a software library that implements FIR signal convolution with the fast overlap-save FFT method. This is the only mechanism which can perform arbitrarily complicated filtering and equalization in real-time [2]. In this paper we discuss the usage of Linux as operating system, the audio driver stack, high-level operations such as binaural synthesis and crosstalk cancellation for loudspeaker reproduction. Finally, we present performance benchmarks and optimization.

Single-board computers

In the last years Single-board computers have become popular not only for experts but also for hobbyists and home users. The availability of various preconfigured operating systems and large repository of compiled software packages makes working with single-board computers more convenient for these user groups. Single-Board Computers usually include a system-on-a-chip (SoC) which contains a general purpose processor, typically based on ARM-architectures. Compared to desktop PCs and most laptops running on x64 architecture, ARM uses a reduced instruction set making chips smaller and more energy efficient. For multimedia applications ARM processors feature a floating-point unit (FPU), that computes up to eight single-precision operations with a single instruction. Current ARM FPU named NEON is available on the Raspberry Pi Model 2 and 3. The Ne10 project [5] provides some DSP relevant functions optimzed for the ARM architecture

Soundcards

Most stereo soundcards just work plug-and-play and offer good quality, such as the Behringer UCA 202 (SNR of around 90 dB). We did not find any compatible USB sound card with more than two channels. Some I^2S sound cards are available for popular boards, such as the Cirrus Audio Board for the Raspberry Pi 2 (figure 1). They can run at lower block sizes than USB devices, thus with

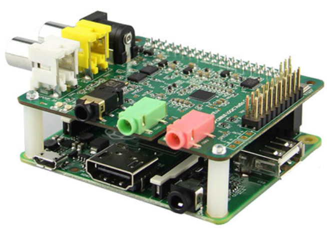

Figure 1: The Raspberry Pi 2 with mounted Cirrus Audio Card 24-bit DAC, 192kHz sampling rate, SNR 95 dB

lower audio latency. We found that most boards have sufficient CPU power to process more than 2-channel data, the bottleneck is audio input and output.

Audio Signal Processing on a SBC

Instead of using a dedicated digital signal processor, we implemented a real-time audio processor in C++. Figure 3 drafts the whole system, with a possible analog audio source or a network stream of audio data that is received by the SBC. A web server allows to control all parameters with any browser. Filters are stored on the SD-card and the user can exchange them during runtime. Arch Linux [6] operating system with special task scheduling configuration and JACK2 [7] facilitate access to the sound card with a delay less than 3 milliseconds. Figure 2 illustrates the audio stack. Audio Processing Software

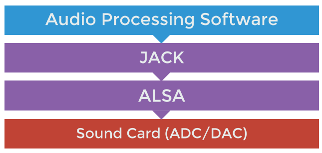

Figure 2: Overview of the linux audio layers in our DSP task.

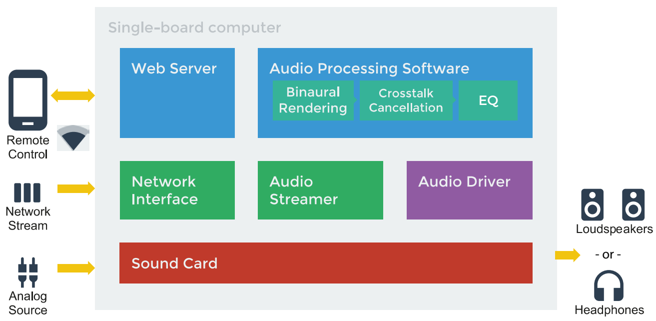

Figure 3: Concept of the SBC in an audio-processing application

Software real-time constraint

Linux Operating System is originally designed for high-throughput (e.g. server applications). It requires special scheduler configuration to reliably process real-time data within a given deadline.
The audio driver splits the audio stream into blocks of arbitrary length. Lower block sizes decrease the latency, but due to shorter deadlines this requires a precise scheduling. With block size of 48 samples at 48 KHz sampling rate the latency of the task scheduler must be lower than 1 ms. System calls (e.g. a write to the SD-card) can take several milliseconds, which causes heavily audible drop-outs of the audio stream. The Linux RT patch [8] deals with this problem by making system calls preemptable. Interrupt handlers from the kernel and normal threads (such as the audio processing program) will run in the same priority pool, and the priority for each interrupt handler and thread can set to arbitrary values. In our DSP application, we give higher priorities to USB/I^{2}S interrupt, JACK and the DSP program. The Linux RT patch also includes debugging tools to trace long-blocking system calls, mostly caused by device drivers. Once a critical blocking driver is found, it should be disabled. For example, let the system boot into RAM and disable the SD-card module after booting.

Binaural rendering

The library can (besides FIR filtering for equalization) synthesize a mono signal to a binaural signal. It uses head-related tranfer functions (HRTFs) stored as DAFF [3] for virtual sound spatialization. This allows 3D positioning of a virtual sound source when using headphones [1]. Users can control virtual sound source position through the browser of their smart phone (see figure 4).
In general, the concept of creating binaural sounds can be applied in several different fields, ranging from musical projects over acoustic virtual environments to creating spatial audio tracks for movies. An interesting application is the virtualization of a loudspeaker setup, to listen to or create multichannel mixes.

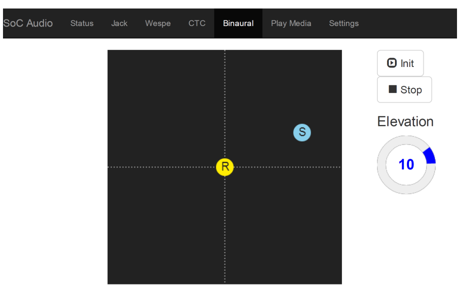

Figure 4: Web interface of the cross talk cancellation to setup loudspeaker positions and listener position.

Crosstalk cancellation

A static crosstalk cancellation (CTC) [4] improves channel separation for reproduction on loudspeakers. Filters are computed on the SBC. Users can adjust the CTC parameters of the loudspeaker array and the listener position (see figure 5).

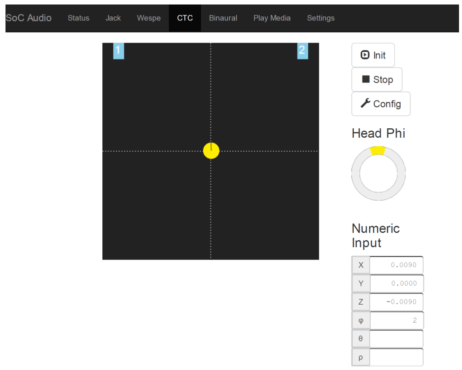

Figure 5: Web interface of the binaural synthesis to control the virtual source position.

More applications

Obviously a SBC can be used for loudspeaker equalization. With wireless network streaming capabilities it can be integrated into a loudspeaker enclosure to create a wireless loudspeaker management (assuming proper time synchronization between multiple SBC). We implemented the binaural synthesis and crosstalk cancellation for demonstration purposes. Considering more use cases, the generic software implementation and the low hardware costs make the system a universal prototype.

Performance

In this real-time task we measured two performance characteristics: throughput (e.g. max. filter-length the system can process in limited time, see figure 6) and latency (the time a sound signal requires to pass through the system, see figure 7). Table 1 shows the maximum possible filter length at a given block size, comparing Raspberry Pi 1 and Odroid X2.
We tuned the FFT-based convolution by exploiting some of the SIMD (Single Instruction, Multiple Data) features of ARM. The computational cost of the real-time convolution depends at least on the filter length and the block size – therefore its always a trade-off between filter length and lower latency. Operating at 96 kHz sample rate, the system can process with very low latency (3 ms).

Table 1: Performance limits of the convolution

Block Size	Max. FIR filter length	
	Raspberry Pi 1	Odroid X2
64	1024	-
96	2048	20k
128	4096	30k
256	12k	80k

Conclusion

Single-chip computers are a generic alternative to dedicated DSP hardware, although compatible audio

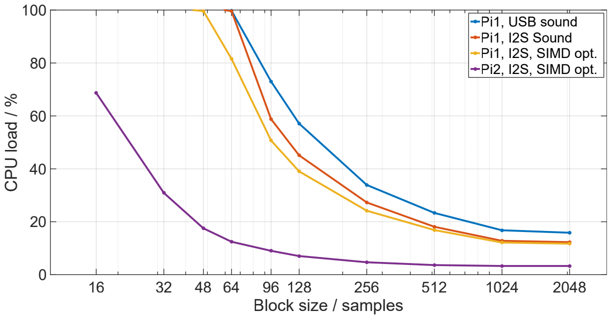

Figure 6: CPU load at block sizes, const. filter length of 1024 samples at 44.1 kHz sampling rate. Raspberry Pi1/2, USB or I^{2}S sound card, SIMD optimizations on/off. At CPU loads close to 100 % the system fails to process in time and outputs distorted audio.

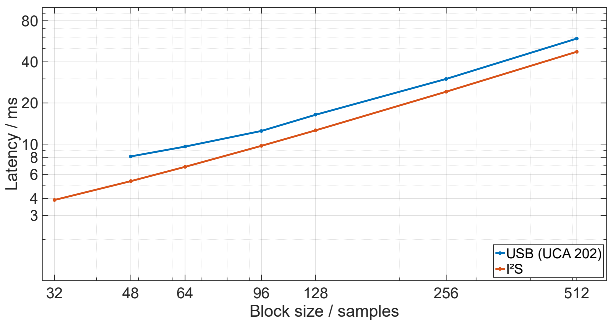

Figure 7: Total DAC/ADC latency at block sizes (using a buffer size of 3 blocks) at 44.1 kHz sampling rate

hardware with more than two channels is currently rare. With new boards and increasing compatibility to hardware – especially due to improvements in the Linux kernel – they become even more suitable for real-time audio processing. Networking capabilities enable wireless transmission of the audio data.
Setting the SBC up for a stable real-time convolution requires a lot of system configuration and optimizations, otherwise it will not work reliable (drop-outs may occur). Of course one can always increase the block size, but this causes higher latency.

References

[1] B. S. Masiero, Binaural technology for virtual reality, RWTH Aachen University, 2008

[2] D. McGrath and A. Reilly, "Huron - A digital Audio Convolution Workstation" 1995

[3] F. Wefers, "OpenDAFF - Ein freies quell-offenes Software-Paket für richtungsabhängige Audiodaten" Proceedings of DAGA Berlin, 2010

[4] B. S. Atal and M. R. Schroeder, "Apparent sound source translator". Patent US3236949 A, 1966

[5] Ne10 project, ARM optimized library, URL: `http://projectne10.github.io/Ne10/`

[6] Arch Linux, URL: `http://www.archlinux.org/`

[7] JACK Audio Connection Kit, URL: `http://www.jackaudio.org/`

[8] Real-Time Linux Wiki, URL: `https://rt.wiki.kernel.org/`