Audio Engineering Society
Convention e-Brief

Presented at the 132nd Convention
2012 April 26–29 Budapest, Hungary

Smart microphone sensor system platform

Elias Kokkinis[1], Konstantinos Drossos[2], Nicolas-Alexander Tatlas[3], Andreas Floros[2], Alexandros Tsilfidis[1] and Kyriakos Agavanakis[3].

[1] University of Patras, Audio and Acoustic Technology Group

[2] Ionian University, Department of Audiovisual Arts

[3] BLUEdev Ltd.

Correspondence should be addressed to Nicolas-Alexander Tatlas (ntatlas@bluedev.eu)

ABSTRACT

A platform for a flexible, smart microphone system using available hardware components is presented. Three subsystems are employed, specifically: (a) a set of digital MEMs microphones, with a one-bit serial output (b) a preprocessing/Digital-to-Digital converter and (c) an CPU/DSP-based embedded system with I2S connectivity. Basic preprocessing functions, such as noise gating and filtering can be performed in the preprocessing stage, while application-specific algorithms such as word spotting, beam-forming and reverberation suppression, can be handled by the embedded system. Widely used high-level operating systems are supported including drivers for a number of peripheral devices. Finally, an employment scenario for a wireless home automation speech activated front-end sensor system using the platform is analyzed.

1. INTRODUCTION

The widespread use of sensors in a number of application domains is evident: The operation for example of home automation and security, environmental monitoring, healthcare and industrial automation systems increasingly depends on physical parameters that need to be continuously and accurately registered. Moreover, a number of such applications greatly benefit from the use of integrated, or so-called smart sensors such as ambient temperature/pressure, illumination, or presence sensors that include processing capabilities for signal enhancement and usually feature extraction as well as wired or wireless networking subsystems, promoting the ambient intelligence model.

A significant part of the prototyping process for smart sensors is based on the use of flexible embedded platforms [1], providing at least the necessary interfaces to digital sensors and adequate computational power. However, these widely used platforms offer only basic (analogue) connectivity to audio sources, have limited processing power and usually employ low bit rate wireless networking solutions, deeming them insufficient for speech or audio sensing applications.

This work presents a generic embedded system that allows the prototyping and testing of all-digital microphone-based sensing systems. An overview of the hardware and system interconnections is provided in Section 2; Section 3 summarizes the embedded processor/DSP architecture, programming and operation workflow; Section 4 presents two applications that have been developed on the hardware system.

2. SYSTEM OVERVIEW

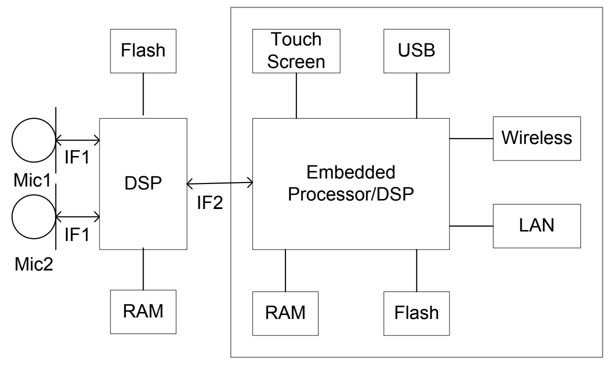

Figure 1: Microphone sensor platform hardware and interface platform

The hardware overview for the platform is shown in Figure 1. A pair of microphones [2] consisting of a MEMS element, an impedance converter amplifier, and a fourth-order Σ-Δ modulator are used. According to [2], the microphones have a high SNR of 61dBA, sensitivity -26dBFS, and a flat frequency response from 100Hz to 15KHz (±3dB). Their output provides a digital Pulse Density Modulated (PDM) signal. The interface IF1 consists of the one-bit PDM and clocking signals.

The microphones are connected to a DSP system [3] that offers a fixed library of basic preprocessing functions, such as Automatic Gain Control, a Parametric Equalizer, Noise Gating etc. The DSP also supports the necessary Digital to Digital conversion, providing a standard I2S digital audio signal. IF2 contains two separate entities, the above-mentioned I2S signal and a USB connection to control DSP features from the embedded processor/DSP.

The processing is handled by a subsystem [4], based on an embedded processor [5], that includes a number of peripheral devices such as a touch screen, USB host and Wired/Wireless network connectivity. Multiple open source and proprietary Operating Systems are supported.

3. EMBEDDED PROCESSOR/DSP

The embedded processor [5] provides a flexible architecture, with the integrated ARM and DSP processors able to handle different tasks with contrasting requirements, thus enabling the efficient implementation of demanding applications, such as those of speech and audio processing.

Programming of the ARM processor is implemented in C++ language. Each of the developed functions is designed and implemented in an object oriented (OO) design. For every major component of the system, an abstraction layer with appropriate interfaces has been defined, so that modules from different platforms or technologies can be smoothly incorporated as needed. Thus, the needed functionality of the ARM processor was first designed, secondly implemented in OO code, compiled, linked with the needed libraries and downloaded to the device bearing the processor.

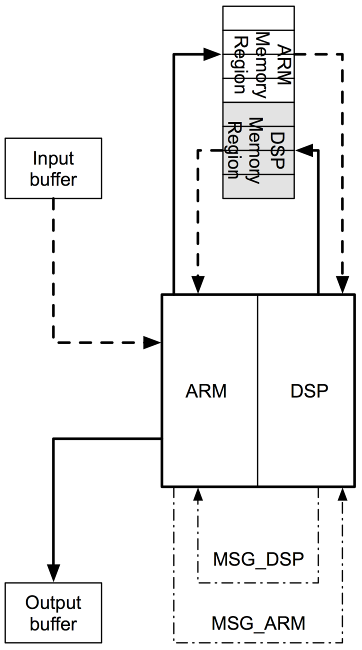

Figure 2: Typical data flow between ARM and DSP processors. Solid lines indicate write operations and dashed lines indicate read operations

A typical data flow for audio applications using the DSP is shown in Figure 2. In order for the processors to be able to exchange data, a dedicated memory region should be reserved for each one. In the example shown in Figure 2, the ARM memory region is write-only for the ARM and read-only for the DSP. The opposite holds for the DSP memory region.

The ARM reads an input buffer from the ADC and writes the data to the corresponding memory region. The DSP reads those data, processes them accordingly and writes the new data to the DSP memory region. The ARM then reads the processed data and writes them to the output buffer of the DAC.

The communication between the ARM and the DSP processors is controlled and synchronized by messages (see Figure 2) that inform each other when data are ready, where they are located and may also pass information about various other parameters.

4. APPLICATIONS

Two demonstrative audio sensing applications were implemented on the embedded processor/DSP; the one utilizes only ARM computational resources, while the second is distributed between the ARM and the DSP.

4.1. Application on embedded processor

A Voice Activity Detection (VAD) sub system was implemented using the ARM processor. In order to keep the complexity and the implied burden for the processor in low levels, a simply VAD algorithm was used as described in [6]. Moreover, due the use of fixed point numbers with the ARM processor, the dominant frequency was estimated with the use of correlation function rather than Fourier Transform (FT). This reduced the use of FT only for just ne feature used in the VAD sub system.

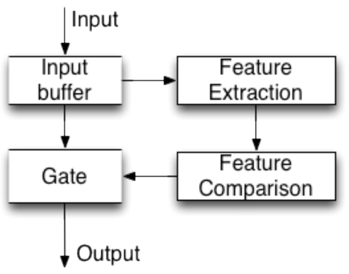

Figure 3: VAD sub-system block diagram

The VAD's procedure is based on the extraction of three features from an input buffer, obtained by the device's audio input, namely: (a) energy, (b) dominant frequency, and (c) spectral flatness. These features were compared to thresholds, decided prior to the start of the system and updated during its functionality if this was necessary. If the value of the extracted features was above the one of the corresponding threshold them there was an indication that voice activity was present. The VAD sub-system is illustrated in **Figure 3**.

Making use of the structured abstraction layers mentioned before, we have provided an implementation of the required audio services based on the host OS functionality. Thus, the acquisition of the input buffer for the VAD sub-system is delegated to the available OS's methods.

4.2. Application utilizing DSP

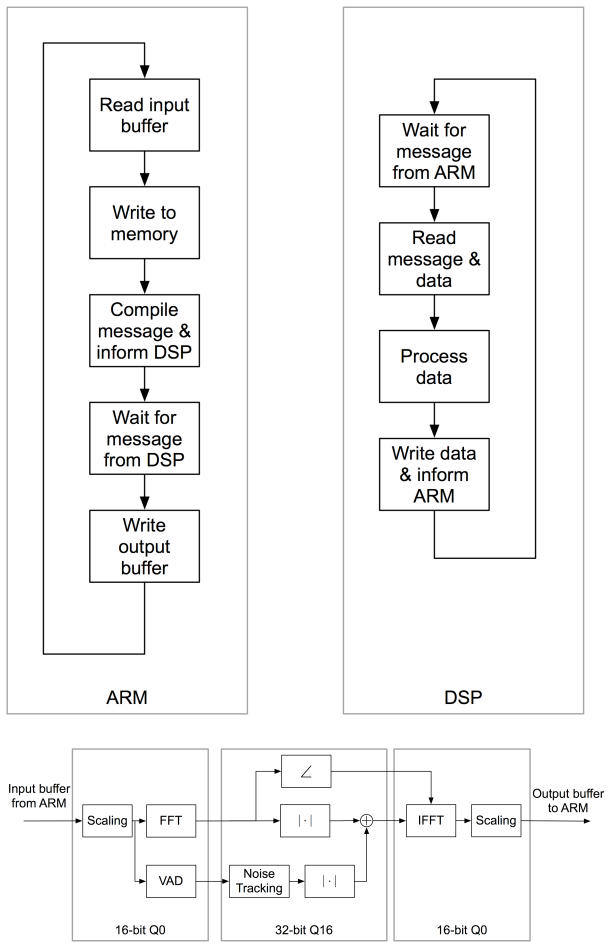

Figure 4: (a) Program flow for the ARM and DSP processors for a simple audio processing application. (b) Implementation diagram of a simple spectral subtraction noise suppression scheme on the DSP.

A simple noise suppression implementation is shown in Figure 4. Figure 4a shows the program flowchart for the ARM and DSP processors. The ARM handles audio I/O and user input. The DSP performs a simple spectral subtraction which is detailed in the block diagram of Figure 4b. The FFT and the energy-based VAD operations are implemented using TI's DSPLIB which operates on 16-bit integers with Q0 format (zero fraction length). The noise tracking process is based on the method described in [7]. All other operations are implemented using the IQmath library for fixed-point processing, which operates on 32-bit numbers with Q16 format.

5. ACKNOWLEDGEMENTS

The work presented here was co-financed by Hellenic Funds and by the European Regional Development Fund (ERDF) under the Hellenic National Strategic Reference Framework (ESPA) 2007-2013, as part of the "MEMSense" project.

6. REFERENCES

[1] Bergmann, N.W.; Wallace, M.; Calia, E.; , "Low cost prototyping system for sensor networks," Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference on , vol., no., pp.19-24, 7-10 Dec. 2010

[2] Analog Devices, "Omnidirectional Microphone with Bottom Port and I2S Digital Output", ADMP421, Data Sheet.

[3] Analog Devices, "SigmaDSP Stereo, Low Power, 96 kHz, 24-Bit Audio Codec with Integrated PLL", ADAU1761, Data Sheet.

[4] Texas Instruments, "OMAP35x Evaluation Module (EVM)", http://www.ti.com/tool/tmdsevm3530/.

[5] Texas Instruments, "OMAP35x Applications Processor", Technical Reference Manual.

[6] M. H. Moattar and M. M. Homayounpour, "A Simple But Efficient Real-Time Voice Activity Detection Algorithm," presented at the 17th European Signal Processing Conference (EU-SIPCO 2009), Glasgow, Scotland, August 24- 28, 2009

[7] Ris, C. and Dupont S., "Assesing local noise level estimation methods: application to noise robust ASR", *Speech Commun.,* 34(1-2), pp.141-158, April 2001