Published in final edited form as:

Conf Rec Asilomar Conf Signals Syst Comput. 2018 October ; 2018: 223–227. doi:10.1109/ACSSC.2018.8645557.

A Wearable Platform for Research in Augmented Hearing

Louis Pisha[*], **Sean Hamilton**[†], **Dhiman Sengupta**[†], **Ching-Hua Lee**[*], **Krishna Chaithanya Vastare**[*], **Tamara Zubatiy**[‡], **Sergio Luna**[§], **Cagri Yalcin**[‡], **Alex Grant**[‡], **Rajesh Gupta**[†], **Ganz Chockalingam**[‡], **Bhaskar D. Rao**[*], **Harinath Garudadri**[*,‡]

[*]Department of Electrical and Computer Engineering, University of California, San Diego

[†]Department of Computer Science and Engineering, University of California, San Diego

[‡]California Institute for Telecommunications and Information Technology (Calit2), University of California, San Diego

[§]Department of Mathematics, University of California, San Diego

Abstract

We have previously reported a realtime, open-source speech-processing platform (OSP) for hearing aids (HAs) research. In this contribution, we describe a wearable version of this platform to facilitate audiological studies in the lab and in the field. The system is based on smartphone chipsets to leverage power efficiency in terms of FLOPS/watt and economies of scale. We present the system architecture and discuss salient design elements in support of HA research. The ear-level assemblies support up to 4 microphones on each ear, with 96 kHz, 24 bit codecs. The wearable unit runs OSP Release 2018c on top of 64-bit Debian Linux for binaural HA with an overall latency of 5.6 ms. The wearable unit also hosts an embedded web server (EWS) to monitor and control the HA state in realtime. We describe three example web apps in support of typical audiological studies they enable. Finally, we describe a baseline speech enhancement module included with Release 2018c, and describe extensions to the algorithms as future work.

Keywords

Hearing aids; hearables; Open Speech Platform (OSP); speech and audio processing

I. Introduction

Open Speech Platform (OSP) [1] is designed as a research tool for improving hearing healthcare through two related paths. First, OSP provides a hardware and software platform for the development of new speech and audio processing algorithms and techniques, which can ultimately lead to the creation of more advanced consumer-grade hearing assisted devices such as HAs and hearables. Second, OSP facilitates both conventional and novel clinical audiological investigations, in particular by giving researchers and listeners unprecedented control over the behavior and parameters of the HA, and by allowing the studies to continue out of the lab into real-world acoustical environments. Thus, our

lpisha@ucsd.edu.

development on OSP has focused on on replicating the functionality of existing systems for hearing loss diagnosis, treatment, and outcomes assessment on a portable, open platform; and then providing extensibility and features far beyond the capabilities of current systems —all to enable new discoveries in hearing healthcare.

The platform (Fig. 1) includes a real-time speech processing engine configured as a master hearing aid (RT-MHA), an embedded web server to monitor and control the HA state, and example web apps for audiologists to conduct research studies and listeners to fine-tune HA parameters in real-world usage. At a high level, signal processing researchers can extend features by modifying and/or adding individual modules to RT-MHA; audiologists can adapt example web apps towards specific research questions and also provide simplified interfaces to listeners to optimize intelligibility in real-world acoustical situations. This contribution describes new developments in each of the three blocks comprising the platform, which serve to bring this vision closer to reality.

The hardware used in OSP as previously published [2] was a laptop, studio audio interface, breakout box, and analog ear-level assemblies. Here, we present a new, wearable version of OSP that facilitates investigations both in the lab and the field. This includes digital ear-level assemblies (Section II) and a wearable pendant unit (Section III) which contains the processing, wireless connectivity, and battery. We characterize the performance of this new hardware from a systems and an audiological perspective (Section IV). Next, we discuss two major software changes present in OSP Release 2018c [1]. We present the embedded web server (EWS) (Section V), and give three examples of provided web apps which are usable in audiological studies, each highlighting a different feature of the EWS. Signal processing researchers can extend the baseline speech enhancement (SE) module (Section VI) by leveraging the OSP infrastructure. Finally, we discuss future extensions to the algorithms provided in OSP (Section VII).

II. Ear-Level Assemblies

The wearable version of the OSP platform uses new, digital ear-level assemblies of behind-the-ear receiver-in-the-canal (BTE-RIC) form factor, codenamed “DJBs” for Digital JellyBeans. These are worn behind the ears, and each connects to the wearable unit via a four-wire cable. Each DJB contains four microphones: one each in the front and rear placements commonly used in other HAs; one packaged together with the in-ear receiver (loudspeaker) for measuring sound within the ear canal; and the fourth, a Voice Pick-Up (VPU) or bone conduction microphone, mounted within the DJB to pick up conducted vibrations of the listener’s voice. The four microphones and receiver are connected to an Analog Devices ADAU1372 codec, which is a high-quality but inexpensive consumer audio codec that includes programmable microphone preamplifiers and headphone drivers. It samples all channels at up to 96kHz 24bit and is powered by separate analog and digital supplies for minimal noise. The digital audio and control lines to and from the codec are controlled by an extremely small FPGA (2.6×2.6 mm, 1300 LUTs), that converts all the data-plane and control-plane signals to a high-speed bidirectional low-voltage differential signaling (LVDS) pair for communication between the DJBs and the wearable unit.

The DJB also contains an inertial measurement unit (IMU), which is a chip containing a three-axis accelerometer and three-axis gyroscope; it is also connected to the FPGA and communicates with the wearable unit over the same two-wire interface. The IMUs will be used in future research with this platform to detect changes in look direction, which will be used to adjust direction of arrival estimates in beamforming algorithms. In addition, it is well known that people with hearing loss have co-morbidities such as vestibular disorders leading to increased risk for falls [3]. Thus, the IMUs at each ear will also be used in future research to (i) assess listeners' ability to maintain postural balance and (ii) detect unexplained gait disturbances.

III. Wearable Unit

The wearable unit is a box containing processing, battery, wireless communication, and other peripherals. It may be worn as a pendant around the neck, kept in a pocket or on a belt, or used as a portable HA research platform in a lab. The unit also contains two microphones (besides the eight across the two DJBs), spatially separated on the left and right sides of the device, for additional soundfield sampling for beamforming and noise estimation purposes.

The embedded system is organized as a custom PCB hosting a system-on-module (SoM), which contains a system-on-chip (SoC), which contains a CPU. The SoM features a chipset used in commercial smartphones, Snapdragon™ 410c, which consists of the SoC, LPDDR3 and Flash memory, a power-management integrated circuit (PMIC) also containing a multichannel audio codec, and wireless radios and antennas for WiFi, Bluetooth, and GPS. The SoC (APQ8016) consists of a quad-core 64-bit ARM Cortex A53 CPU at 1.2 GHz, a DSP, a GPU, and a variety of peripherals. The CPU runs offthe-shelf ARM64 Debian Linux with custom ALSA drivers provided for the DJBs. The CPU's floating-point performance is several GFLOPS (the exact value depending on whether NEON vector intrinsics [4] are used and whether DSP and GPU performance is counted), enabling algorithm developers to simply #include <math.h> and comfortably code in platform-agnostic ANSI C or standard C ++ through C++14.

Besides the SoM, the wearable unit's custom PCB contains an FPGA to convert the LVDS signals from the DJBs back to I2S and SPI serial interfaces. Peripherals include a microSD card for data logging and two microUSB ports for Linux debug and charging/flashing respectively. The battery is a standard3.7V, 2000 mAh Li-Ion smartphone-style battery, which is both user-replaceable and chargeable in-system.

IV. Hardware Results

a) Latency:

Latency is measured both with RT-MHA processing enabled and disabled; when disabled, the RTMHA software simply outputs an amplified copy of the front microphone signal from each ear. An analog waveform with a noticeable transient is introduced at the microphone, by tapping the microphone with a fingernail. Then, using an oscilloscope, the delay between the transient at the codec's analog input and the corresponding transient produced at its analog output (for the receiver or loudspeaker) is measured. For this wearable OSP system with the 2018c code and the latest OS and drivers, the latency with RT-MHA processing disabled is measured to be about 2.3ms, and with RT-MHA processing enabled it is about 5.6ms. This matches our expectations, since the RT-MHA processing introduces a latency of about 3.3ms due to its FIR filters. The hardware latency is also reasonable, as the current code requests input and output latencies of 1ms each, and the remaining 0.3ms may be due to resampling filters in the codec and other miscellaneous small delays. (The FPGA LVDS scheme only introduces a round-trip latency of one sample, so this does not make a significant contribution to the system delay.) The target latency for a real-time speech processing system is 10ms, so OSP provides a comfortable margin with latency budget available for future algorithms.

b) Computation Performance:

The RT-MHA software is a hard real-time process requiring a relatively rapid response time: the input and output buffers are 1ms long (48 or 96 samples), so the processing for each frame must complete within 1ms or else artifacts will be audible. Given that this process is running under a non-realtime OS (Debian Linux) and must coexist with the embedded web server (EWS) and other system processes, this is a non-trivial task. On top of these considerations, the RT-MHA uses separate threads to process the data for each ear, so inter-thread communication and synchronization must also occur within these time constraints. Two steps are taken to increase the robustness of this process: first, all OS functionality and all processes besides the RT-MHA are bound to CPU core 0, while the RT-MHA is bound to cores 1 through 3; and second, nice and ionice are used to give the RT-MHA maximum process priority over CPU and I/O resources.

The execution time for each RT-MHA module is measured on each 1ms frame using a high-resolution system timer, and the statistics are aggregated throughout a run and reported at program termination (Table I). Statistics for filtering, speech enhancement, peak detection, and WDRC are logged on a per-band basis and multiplied by 6 to cover the six subbands; this is why the “maximum” total time for a particular channel appears to be higher than the maximum time ever taken for the whole processing. The maximum values reported here do not occur often at runtime (compare the minimum and average values), and they may be partially due to caching or other initialization factors the first time they are run.

c) Power Consumption:

For these measurements, the wearable unit was simply powered from a benchtop supply at 3.7V, and the average current draw was recorded during various operations (Table II). Battery life numbers are computed from the current draw assuming a 2000 mAh battery. Since nominal battery capacities often include capacity not usable in a certain application, usable battery life may be lower. For instance, in our testing, the wearable system shuts off when the battery voltage drops to about 3.2V, yet battery manufacturers may rate the “2000 mAh” claim by drawing current from the battery until its voltage drops to near zero. Nevertheless, we expect the system to consistently provide 3–4 hour battery life with all features operational.

d) HA Performance:

The wearable unit and one DJB were placed in the Audioscan Verifit 2 test box [5] and measured with the ANSI 3.22 test protocol [6]. The results are compared to those from four different commercial HAs, as well as to the previously published results for the laptop-based (“lab”) OSP system.

For the most part, the results match the OSP lab system in meeting or exceeding the performance of commercial systems, providing higher output levels for listeners with severe hearing loss and keeping a low equivalent input noise (EIN). The one metric that is substantially worse is the distortion with the high-bandwidth (i.e. low-power) receiver. With a lower average gain (about 25 dB), this receiver provided good performance with $\leq 1\%$ distortion in all bands, but when the gain was raised to 35 dB it clearly distorted. As seen in Table III, this distortion did not occur with the high-power receiver at similar or higher volume levels, so it is not an output power limitation of the DJBs; nor was the distortion as bad when the OSP lab system drove the high-bandwidth receiver, so it is not an inherent limitation of the receiver. Instead, we believe it is due to the supply rails: the high-bandwidth receiver has a higher impedance than the high-power receiver (117.8Ω vs. 46Ω at 1kHz [7] [8]), and the DJBs can provide only up to6.6Vpp (3.3V supply) to the receiver compared to up to 10Vpp (5V supply) in the OSP lab system. Thus, future DJB designs with wider analog supply rails could increase the maximum non-distorting output level with the high-bandwidth receiver.

V. Embedded Web Server

The embedded web server (EWS) is based on a Linux, Apache, mySQL and PHP (LAMP) software stack [9] running on the wearable device. PHP [10] is a server-side scripting language, which we use to provide rich features for browser-enabled devices to monitor and control the state of the real-time master hearing aid (RT-MHA) software. The web apps send parameters to the RT-MHA using PHP, and also store data in a mySQL database for later analysis. The other parts of the web apps are written in HTML, CSS, and Javascript, so they can be easily edited to facilitate a diverse array of research and support the activity of both researchers and listeners. To demonstrate the potential of the EWS, we describe three applications: the Researcher Page, the 4AFC web app, and the two Goldilocks web apps.

a) Researcher Page:

This web app allows researchers to change amplification, noise management, and feedback management parameters in real time. Researchers can upload existing parameter settings from a mySQL database, and can also save new parameter settings. A key benefit of using the EWS model is that all of the web apps work across operating systems and form factors, with any browser-enabled device. Clicking Transmit directly results in a PHP call to the RT-MHA, in which the JSON object containing all of the parameters is communicated and applied in real time by the RT-MHA.

b) 4 Alternate Forced Choice (4AFC):

For diagnosis or evaluation of a particular HA configuration, this web app provides a multiple choice task in response to a sound being played. The questions and associated sound files can be specified by simply modifying a JSON object file. We provide a sample file, and instructions on making the requisite changes. This simple text change is enough to customize this web app for many studies. For example, researchers may change the number of choices, play raw sounds, or run any number of noise or feedback cancellation algorithms on the audio live; any of these experiments will be equally simple for the researcher. The listener must simply enter the appropriate URL into their browser-enabled device, log in, and their actions are automatically logged to the database. Researchers can easily see the results of the studies, and can customize how they would like to receive them.

c) Goldilocks:

A still greater example of the potential of our system for the listener are the two “Goldilocks” web apps. Goldilocks is a paradigm for self-fitting of hearing aids, developed by Carol Mackersie and Arthur Boothroyd at SDSU[11]. It lets listeners adjust the spectral shape and loudness using a simple and intuitive GUI. The “Goldilocks Researcher Page” web app performs similar functionality to the general researcher page, but with specific features for the Goldilocks app. A researcher can log in with credentials for a particular listener, which establishes this listener in the database. Then, the researcher manipulates the settings on the Goldilocks Researcher Page—from HA parameters to the sequence of adjustments used in the study to one of five possible formats for the study—and saves all of them to the database. The listener can now log into the “Goldilocks Listener” web app, at which point they are automatically directed to the version of Goldilocks which has been programmed for them in advance by the researcher. As they go through the task, all of their actions are automatically recorded to the database. The researchers can choose to monitor the listener’s actions in real time, and can also download the logs in comma separated value (CSV) format. To summarize, the web apps on the EWS communicate in real time with the RT-MHA and the mySQL database. The web apps are accessible to listeners and researchers from any browser-enabled device, and the web apps can be easily customized to support diverse research activity.

VI. Speech Enhancement in Real-Time Master Hearing Aid (RT-MHA)

Reduction of the background noise remains an important and challenging task for HAs. The microphone of the HA device picks up the desired voice of the target speaker as well as ambient noise, often resulting in a degraded speech signal that could be difficult to understand for the listener. It is therefore crucial to have reliable and robust SE systems. OSP Release 2018a introduced a baseline version of the SE module for improving intelligibility in noisy environments. More specifically, the SE module is based on a version of the SE systems investigated in [12] by Kates and developed in MATLAB; this module has been ported to ANSI C and provided as a library in source code. The module utilizes the output of each of the six subbands, consisting of (1) peak and valley detection, (2) noise power estimation, and (3) Wiener filtering. For noise power estimation, we provide the following options: (1) power averaging by Arslan *et al.*[13], (2) weight averaging by Hirsch and Ehrlicher [14], and (3) minima controlled recursive averaging (MCRA) by Cohen and Berdugo [15]. After obtaining the noise power estimate using any of the above techniques, the classic Wiener filter is performed to suppress the noise contents for enhancing the speech. Then, the WDRC block takes the enhanced subband speech signals as input and performs non-linear amplification to generate the output of the HA process.

VII. Future Work: Extensions to RT-MHA

The RT-MHA not only provides a platform for audiologists to perform clinical studies, but also serves as a baseline system for engineers to make extensions and comparisons with their own signal processing algorithms. For example, for SE one can develop advanced noise estimation techniques and modify the SE module accordingly to compare with the three baseline algorithms provided by the current version of RTMHA, without the overhead of developing a substantial new test bench for SE research. Similarly, for AFC one can put in new algorithms and compare against the basic AFC algorithms [16] or even against advanced features like the sparsity promoting least mean squares (SLMS) [17] algorithm. Other extensible features include the subband filters and WDRC, as well as extending the microphone array processing from a traditional delay-and-sum beamformer separate on each ear to beam forming using signals from both ears.

VIII. Summary

In this contribution, we describe new features in several important aspects of OSP, all of which help advance the platform’s mission of enabling new research and discoveries in hearing healthcare. The complete real-time master hearing aid (RT-MHA) now runs on a fully wearable platform, comprised of a small wearable unit for processing and battery, and two wired digital BTE-RIC ear-level assemblies (DJBs). The DJBs provide high-performance (up to 96kHz / 24 bit), low-noise audio with four microphones per ear, and meet or exceed the performance of commercial HAs on most metrics. The wearable unit processes the audio with only a 2.3ms hardware / system latency, plus 3.3ms for delays due to filtering in the RT-MHA processing for a total of 5.6ms. The CPU is able to complete the audio computation in real time with processing power to spare, and the wearable unit is expected to run for 3–4 hours on a charge with all features enabled. The embedded web server (EWS) runs on the wearable unit and allows for real-time control of all RT-MHA parameters, providing these controls to researchers and listeners in the form of web apps that facilitate audiological studies and allow for listener self-adjustment of HA parameters. Finally, the set of baseline algorithms provided in the RT-MHA has been enhanced with the addition of the speech enhancement (SE) module, which amplifies speech-like signals while suppressing background noise. All the hardware and software within OSP, from the DSP algorithms to the EWS scripts, is open source [1] and easily extensible for the development of novel algorithms and research studies.

Acknowledgment

This work is supported by Nation Institute of Health, NIH/NIDCD grants R33DC015046, “Self-fitting of Amplification: Methodology and Candidacy,” and R01DC015436, “A Real-time, Open, Portable, Extensible Speech Lab” to University of California, San Diego.

References

[1]. Open Speech Platform, http://openspeechplatform.ucsd.edu/, University of California, San Diego, 2018.

[2]. Garudadri HarinathA realtime, open-source speech-processing platform for research in hearing loss compensation; 2017 51st Asilomar Conference on Signals, Systems, and Computers; 2017. 1900–1904.

[3]. Lin FR and Ferrucci L, “Hearing loss and falls among older adults in the united states,” Archives of Internal Medicine, vol. 172, no. 4, pp. 369–371, 2012. [PubMed: 22371929]

[4]. Jang Minwoo, Kim Kukhyun, and Kim Kanghee, “The performance analysis of ARM NEON technology for mobile platforms,” in Proceedings of the 2011 ACM Symposium on Research in Applied Computation, New York, NY, USA, 2011, RACS ’11, pp. 104–106, ACM.

[5]. Verifit Audioscan 2, https://www.audioscan.com/verifit2,Audioscan / Etymonic Design Inc., 2014.

[6]. American National Standards Institute, Specification of Hearing Aid Characteristics (ANSI 3.22–2014), ANSI New York, 2014.

[7]. Knowles Electronics, “RVA-90020-NXX datasheet,” 2009.

[8]. Knowles Electronics, “RVA-90080-NXX datasheet,” 2014.

[9]. Wikipedia contributors, “LAMP (software bundle) – Wikipedia, the free encyclopedia,” 2018, Online; accessed 3-December-2018.

[10]. Laracasts, “Laravel 5.7 from scratch,” https://laracasts.com/series/laravel-from-scratch-2018, 2018.

[11]. Boothroyd Arthur and Mackersie Carol, “A “goldilocks” approach to hearing-aid self-fitting: User interactions,” American journal of audiology, vol. 26, no. 3S, pp. 430–435, 2017. [PubMed: 29049625]

[12]. Kates JM, “Modeling the effects of single-microphone noise-suppression,” Speech Commun, vol. 90, pp. 15–25, 2017.

[13]. Arslan L, McCree A, and Viswanathan V, “New methods for adaptive noise suppression,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP). IEEE, 1995, vol. 1, pp. 812–815.

[14]. Hirsch H-G and Ehrlicher C, “Noise estimation techniques for robust speech recognition,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.(ICASSP). IEEE, 1995, vol. 1, pp. 153–156.

[15]. Cohen I and Berdugo B, “Noise estimation by minima controlled recursive averaging for robust speech enhancement,” IEEE Signal process. lett, vol. 9, no. 1, pp. 12–15, 2002.

[16]. Lee C-H, Kates JM, Rao BD, and Garudadri H, “Speech quality and stable gain trade-offs in adaptive feedback cancellation for hearing aids,” J. Acoust. Soc. Am, vol. 142, no. 4, pp. EL388–EL394, 2017. [PubMed: 29092590]

[17]. Lee C-H, Rao BD, and Garudadri H, “Sparsity promoting LMS for adaptive feedback cancellation,” in Proc. Europ. Signal Process. Conf. (EUSIPCO), 2017, pp. 226–230.

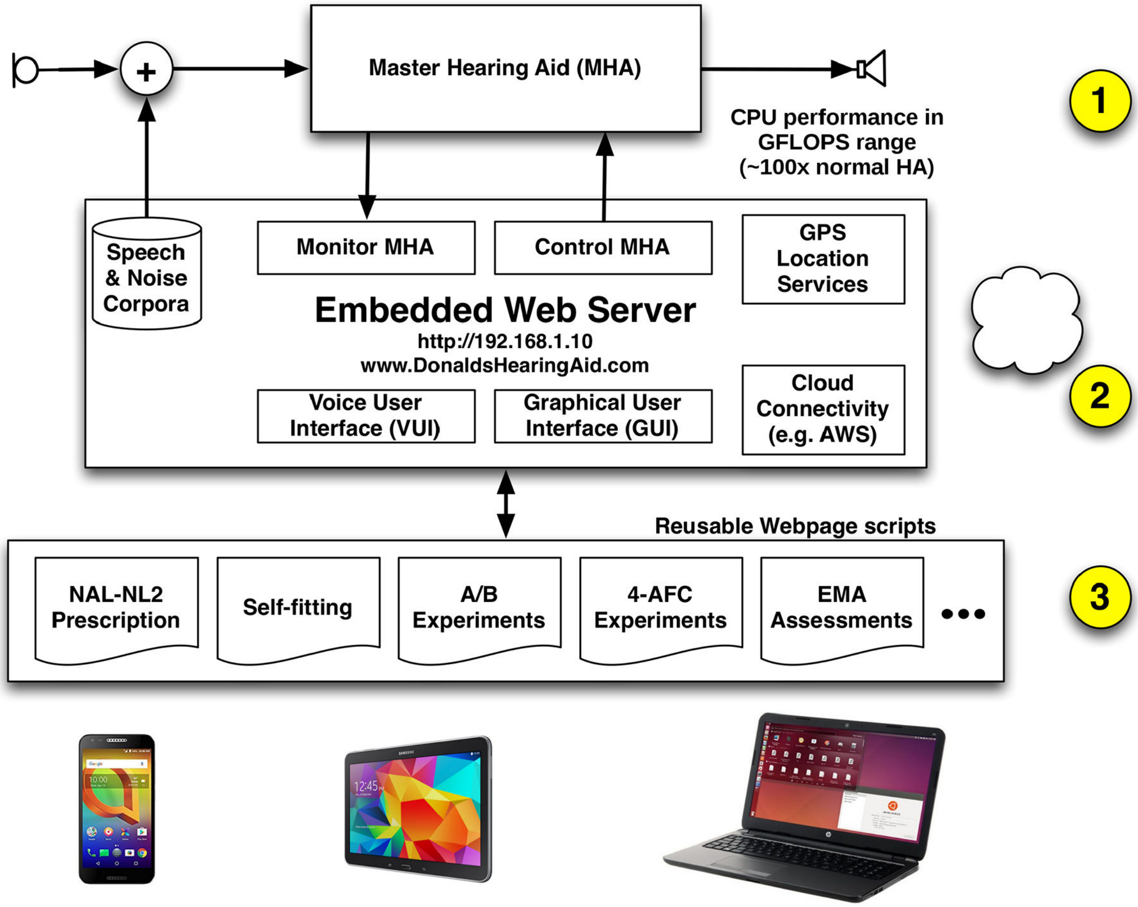

Fig. 1.

System Architecture (The top blocks labeled (1) correspond to the realtime sound processing subsystem. We show only one channel, but it is understood that this is a binaural system with multiple microphones on each ear. The middle blocks (2) depict the embedded web server (EWS), responsible for all monitoring and control of the realtime system. The EWS subsystem is also responsible for providing graphical user interfaces (GUIs) as web pages running on remote, browser-enabled devices, such as smartphones, tablets and computers. The web pages are hosted on the wearable device itself in the EWS subsystem. Block (3) depicts typical web apps that encapsulate various audiological studies.

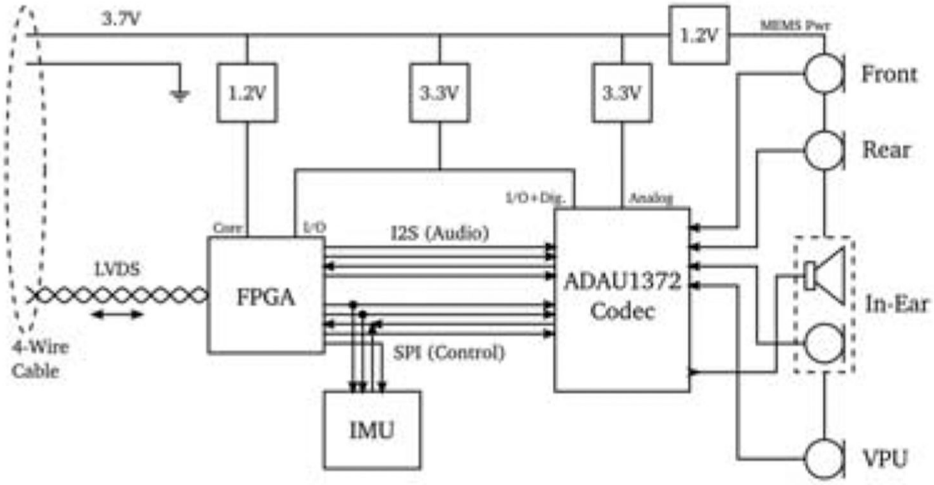

Fig. 2.

Digital ear-level assembly (DJB) block diagram for one ear.

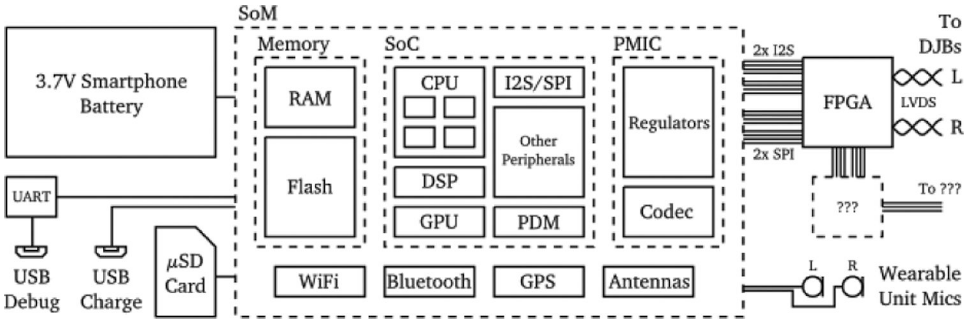

Fig. 3.

Wearable unit (main embedded system) block diagram.

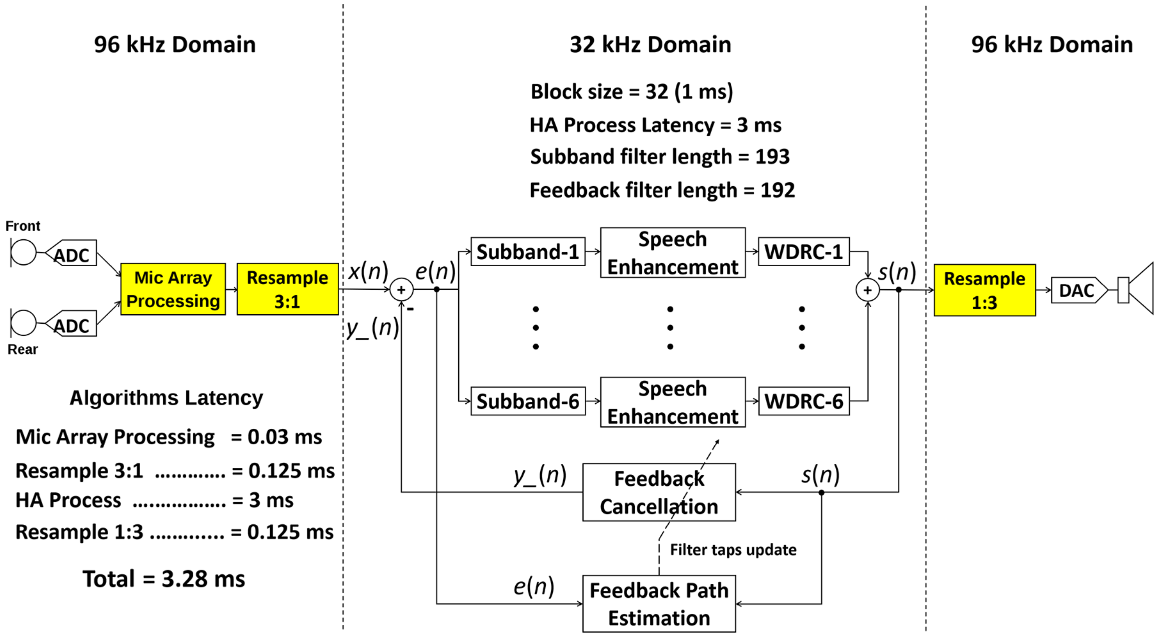

Fig. 4.

RT-MHA software block diagram with signal flows. The I/O is operating at 96 kHz sampling rate and the main processing is carried out in 32 kHz sampling domain to realize subband decomposition, SE, WDRC, and AFC. Note that this system is duplicated for both ears.

TABLE I

Performance statistics for real-time processing of 1ms audio buffers, over ≈ 3 minute sample.

Operation	**Times (μs)**		
	Minimum	**Average**	**Maximum**
Downsampling	32	32	91
Filtering	282	282	822
Speech enhancement	12	12	408
Peak detection	24	30	384
WDRC	18	18	384
AFC	180	190	3661
Upsampling	32	33	91
Total channel:	580	597	5841
Overall	626	694	4635

TABLE II

Current draw (at nominal 3.7VDC) and battery life (assuming 2000mAh Li-ion battery) for common system use cases.

Test Conditions	Average Current (mA)	Battery life (h)
Idle (no WiFi)	0.243	8.23
Idle	0.284	7.04
RT-MHA (HA process disabled)	0.317	6.31
RT-MHA (all processing enabled)	0.440	4.55

TABLE III

ANSI 3.22 test results for OSP system configurations as measured by Audioscan Verifit 2, as compared to results from four commercial HAs.

Metric	**Units**	**Commercial HAs**				**OSP Lab Sys.**		**OSP Wearable**	
		A	**B**	**C**	**D**	**X**	**Y**	**X**	**Y**
Average Gain	dB	40	40	25	35	40	40	35	38
Max OSPL90	dB SPL	107	112	110	111	121	130	119	129
Average OSPL90	dB SPL	106	109	108	106	112	126	111	125
Average Gain @ 50 dB	dB	37	39	25	35	35	41	34	38
Low Cutoff	kHz	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
High Cutoff	kHz	5	6	5	6.73	8	6.3	8	6.73
Equivalent Input Noise	dB SPL	27	26	30	27	29	28	28	27
Distortion @ 500 Hz	% THD	1	1	0	0	2	1	5	1
Distortion @ 800 Hz	% THD	1	1	0	0	3	2	5	1
Distortion @ 1600 Hz	% THD	0	0	0	0	1	1	2	0

X: with high-bandwidth Knowles receiver [7]

Y: with high-power Knowles receiver [8]