Development of an Open Source Hearing Aid Platform

William (Chip) Audette, Odile Clavier, Creare LLC

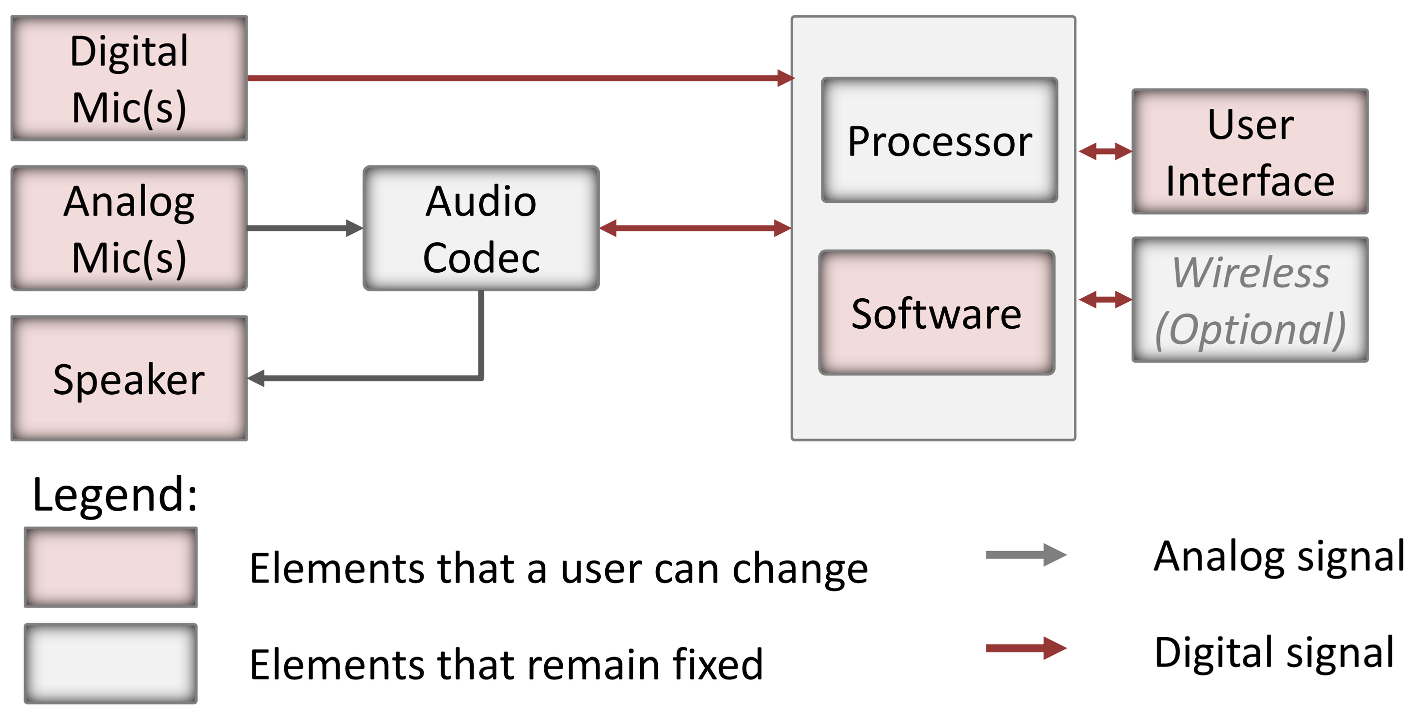
Daniel Rasetshwane, Stephen Neely, Marc Brennan, Ryan McCreery, Boys Town National Research Hospital

Joshua Alexander, Purdue University

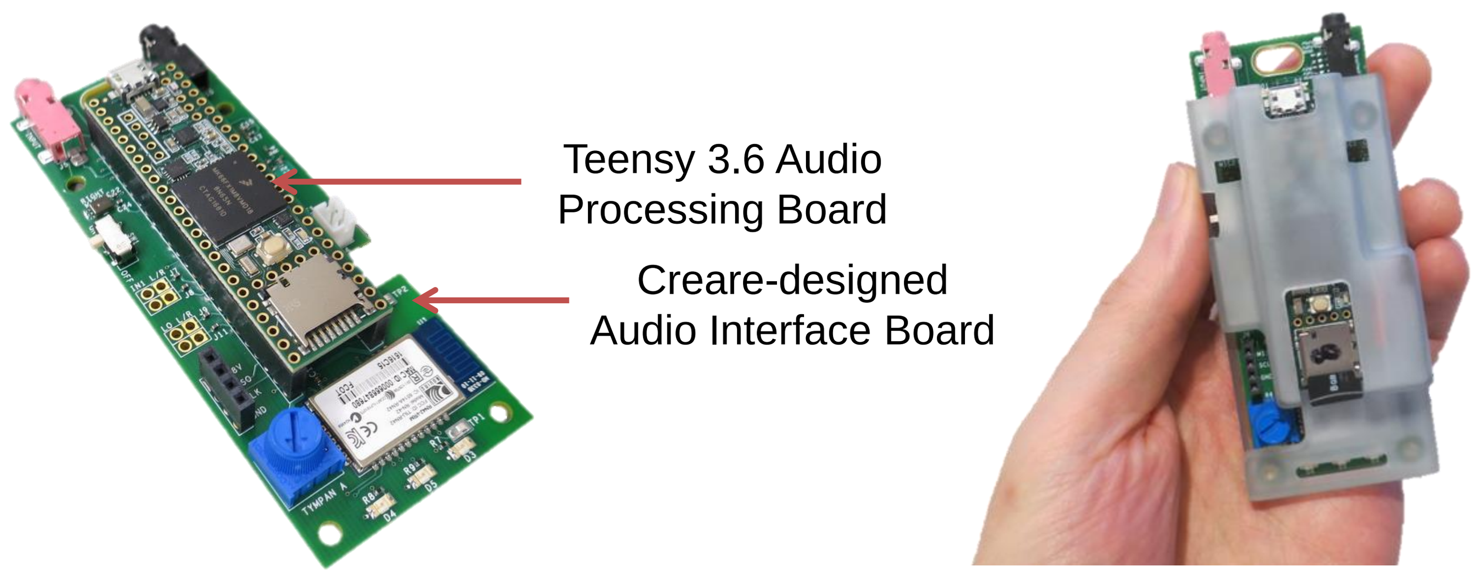
Overview

As the lead federal agency promoting the nation's hearing healthcare, NIDCD is actively seeking to address accessibility from the public health perspective. Aside from cost, one of the primary barriers to hearing aid adoption remains sound quality, and the sense that hearing aids are still not quite able to meet all the needs of hearing impaired consumers.

In this project, we seek to develop an open source hearing aid platform in order to spur innovation in hearing aid research through collaborative development. An overview of the platform architecture is provided below.



Current Hardware Implementation



- Processor: NXP 180 MHz ARM 32 bit processor with Floating Point Unit
- CODEC: Very low power 32 bit TLC320AIC3206
- Bluetooth: Roving Networks RN-42 Bluetooth Classic
- Tool chain: Arduino integrated development environment (IDE)

Basic Electroacoustic Measurements

Basic electroacoustic measurements of this first prototype were made. Using a Knowles SPH1642 MEMS Microphone (on the Tympan board) and Etymotic HF5 earbuds, the performance specifications are :

- Input Referred Noise Floor: 39 dBA SPL
- Input Level at Clipping: 127 dB SPL
- Input Referred Dynamic Range: 88 dBA
- Maximum Output Level: 121 dB SPL

Software Implementation

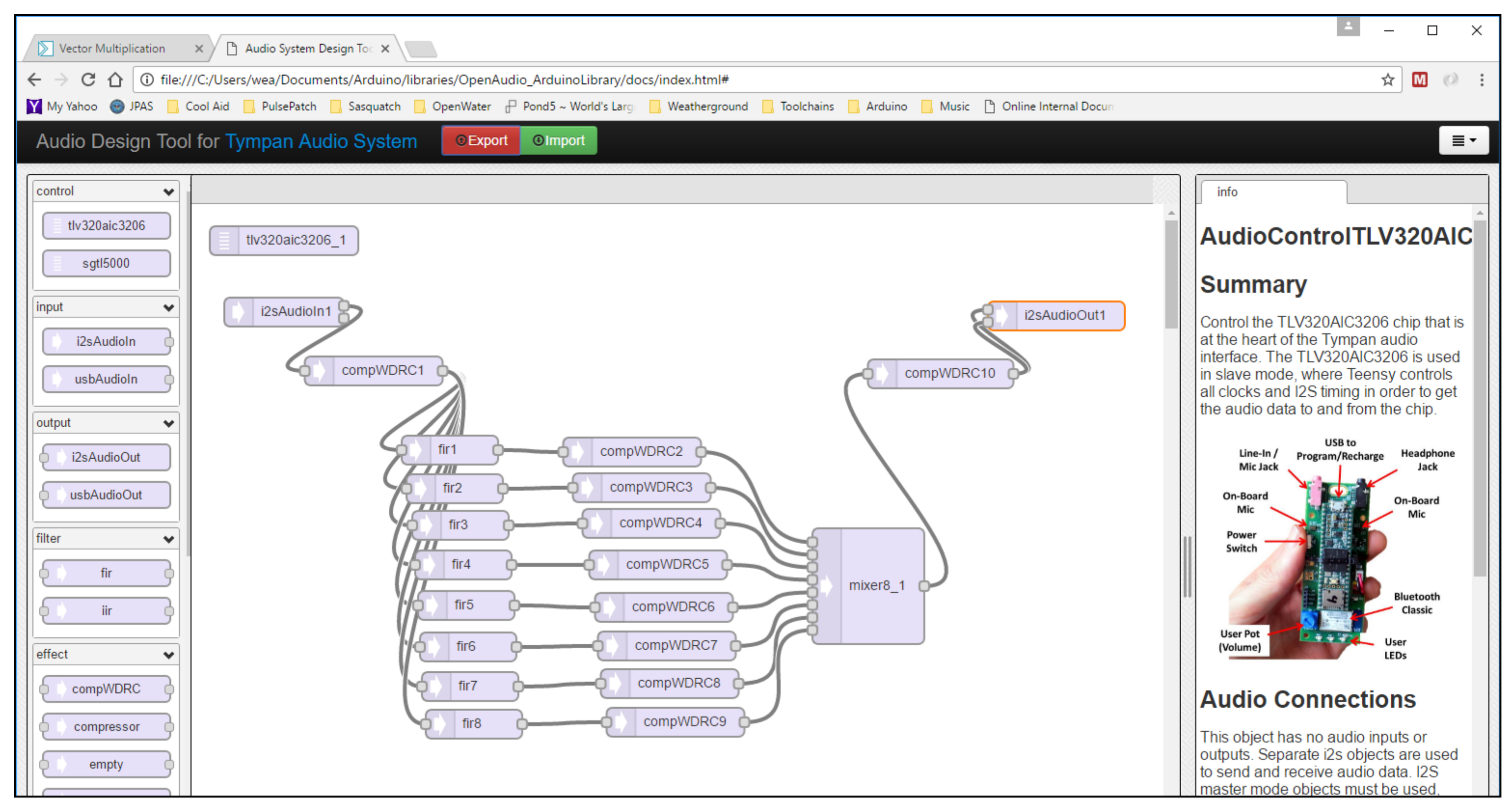
The key challenge is to develop a software platform that allows external users with diverse objectives to customize the hearing aid. We achieve this by providing access at two different levels:

- Audiology researchers modify the parameters of individual algorithms through a simple user interface
- Developers build their own algorithms directly onto the hardware to minimize latency

Developer Interface

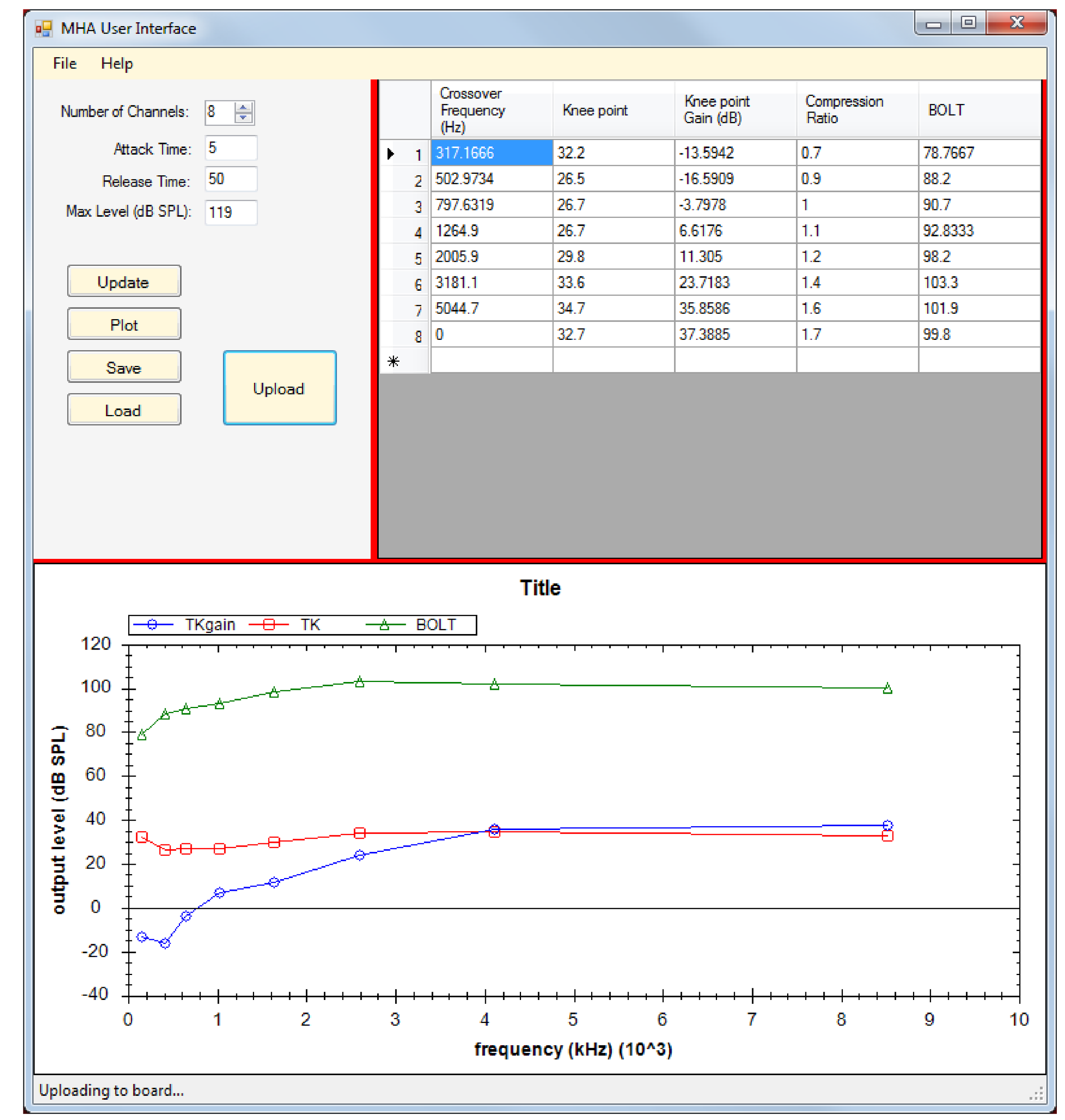
The Teensy board leverages the Arduino software development environment, providing a rich user community ready to contribute, with many examples already available. With this, a developer can create entirely new algorithms.

To help get started, we have also created a graphical interface (GUI) for making new audio processing algorithms. The example below shows an 8-band compression system built from blocks. The GUI writes the Arduino-compatible software, which the developer can then learn from and extend.



Research User Interface

Several algorithms have already been implemented at the embedded level, and researchers can modify the parameters through a Graphical User Interface (GUI). Below we show an 8-band compression filter that can be entirely customized by an audiologist before testing the hearing aid.



Future Developments

In this multi-year effort, a new version of the hardware and firmware will be released yearly. Each version will incorporate user-requested features as well as updated chips (processor, AIC, etc.). Each new release will include full hardware designs and source code available on GitHub, as well as hardware kits at Tympan.org. Users can provide feedback through the Tympan forum or through the GitHub repositories.

For the next release we are considering two different processors

TMS320C6748	ARM M7
32-bit, 456 MHz, 320K RAM	32-bit, 400 MHz, 1MB RAM
3600 DMIPS (if parallelized)	860 DMIPS
Fixed or floating point	Fixed or floating point
TI software tools	Arduino software tools

How to Participate: Tympan.org

Visit the Tympan website for access to all hardware designs and schematics as well as embedded software source code, user interface source code, and documentation: https://tympan.org/.

Make your own or buy a board through Tympan (available Summer 2017). Visit the Tympan user forum to ask questions, get answers, and participate in the development of this platform: https://forum.tympan.org/.

This work has been supported by the National Institutes on Deafness and Communication Disorders under SBIR Phase I Grant #1R44DC015445-01