Open Portable Platform for Hearing Aid Research

Hendrik Kayser
Caslav Pavlovic, Volker Hohmann, Louis Wong, Tobias Herzke, S.R. Prakash, Paul Maanen, Zezhang Hou

Digital hearing devices
Medizinische Physik and Cluster of Excellence "Hearing4all"
Carl von Ossietzky University Oldenburg

03/03/2018

Project aims / NIH initiative

Open

Design

Tools

for

Speech

Signal

Processing

Goals

- Lower barriers for hardware and software development
- Enhancement of tools for acoustic psychophysical research studies
- Facilitate translation into hearing aids, cochlear implants, consumer devices

Project aims / NIH initiative

Open Design Tools for Speech Signal Processing

Goals

- Lower barriers for hardware and software development
- Enhancement of tools for acoustic psychophysical research studies
- Facilitate translation into hearing aids, cochlear implants, consumer devices

Approaches and Tools

- Algorithms that exploit computing power of future hearing devices
- Portable acoustic signal processing tools
- Reconfigurable real-time speech enhancement software for real-world operation
- Open source design principles
- Outreach and dissemination activities

Overview: Open Portable Platform for Hearing Aid Research

Research Project (R01)

open Master Hearing Aid (openMHA)

- Open-source software for hearing aid algorithm development and evaluation
- real-time signal processing
- Low-delay (< 10 ms)
- programmable in C++

+

Small Business Innovation Research (SBIR)

Portable Hearing Laboratory (PHL)

- Mobile hardware platform
- Running openMHA under Linux
- remotely accessible for configuration on runtime

Overview: Project partners

research
publications
algorithms
software
evaluation

hardware design
enhacement
user requests

University of Oldenburg

HörTech gGmbh (Oldenburg)

Bat&Cat SoundLabs (Palo Alto)

Advanced Sensor Integrations

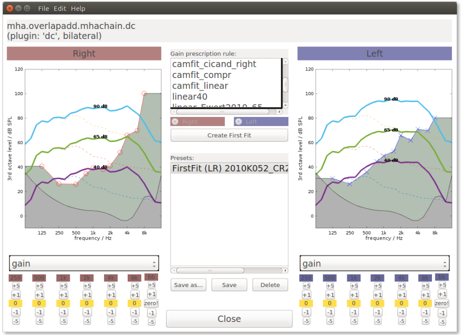
openMHA (R01DC015429)

PHL (R44DC016257)

Target groups

Audiological researchers

- Plug-and play software modules
- Easily change processing parameters on high level



mha.overlapadd.mhachain.dc
(plugin: 'dc', bilateral)

Gain prescription rule:
camfit_cicand_right
camfit_compr
camfit_linear
linear40
linear_Ewert2010_65

Presets:
FirstFit (LR) 2010K052_CR2

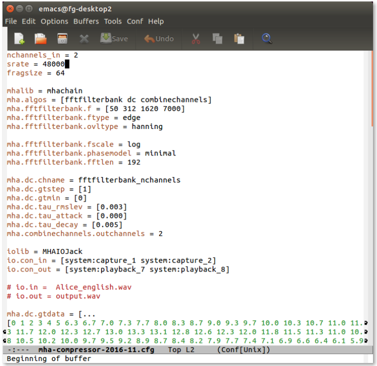
Target groups

I Audiological researchers

- Plug-and play software modules
- Easily change processing parameters on high level

II Application engineers

- Set up measurement tools and customize algorithms
- Access configuration interface at an advanced level



```
nchannels_in = 2
srate = 48000
fragsize = 64

mhalib = mhachain
mha.algos = [fftfilterbank dc combinechannels]
mha.fftfilterbank.f = [50 312 1620 7000]
mha.fftfilterbank.ftype = edge
mha.fftfilterbank.ovltype = hanning

mha.fftfilterbank.fscale = log
mha.fftfilterbank.phasemodel = minimal
mha.fftfilterbank.fftlen = 192

mha.dc.chname = fftfilterbank_nchannels
mha.dc.gtstep = [1]
mha.dc.gtmin = [0]
mha.dc.tau_rmslev = [0.003]
mha.dc.tau_attack = [0.000]
mha.dc.tau_decay = [0.005]
mha.combinechannels.outchannels = 2

iolib = MHAIOJack
io.con_in = [system:capture_1 system:capture_2]
io.con_out = [system:playback_7 system:playback_8]

# io.in =  Alice_english.wav
# io.out = output.wav

mha.dc.gtdata = [...
[0 1 2 3 4 5 6.3 6.7 7.0 7.3 7.7 8.0 8.3 8.7 9.0 9.3 9.7 10.0 10.3 10.7 11.0 11.
3 11.7 12.0 12.3 12.7 13.0 13.3 13.1 12.8 12.6 12.3 12.0 11.8 11.5 11.3 11.0 10.
8 10.5 10.2 10.0 9.7 9.5 9.2 8.9 8.7 8.4 8.2 7.9 7.7 7.4 7.1 6.9 6.6 6.4 6.1 5.9
-:---  mha-compressor-2016-11.cfg   Top L2     (Conf[Unix])
Beginning of buffer
```

Target groups

I Audiological researchers

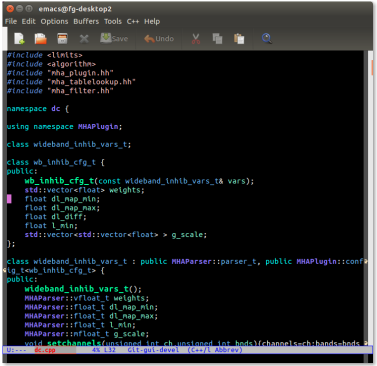
- Plug-and play software modules
- Easily change processing parameters on high level

II Application engineers

- Set up measurement tools and customize algorithms
- Access configuration interface at an advanced level

III Plugin developers

- Develop and implement new plugins in the openMHA framework



```
#include <limits>
#include <algorithm>
#include "mha_plugin.hh"
#include "mha_tablelookup.hh"
#include "mha_filter.hh"

namespace dc {

using namespace MHAPlugin;

class wideband_inhib_vars_t;

class wb_inhib_cfg_t {
public:
    wb_inhib_cfg_t(const wideband_inhib_vars_t& vars);
    std::vector<float> weights;
    float dl_map_min;
    float dl_map_max;
    float dl_diff;
    float l_min;
    std::vector<std::vector<float> > g_scale;
};

class wideband_inhib_vars_t : public MHAParser::parser_t, public MHAPlugin::conf
ig_t<wb_inhib_cfg_t> {
public:
    wideband_inhib_vars_t();
    MHAParser::vfloat_t weights;
    MHAParser::float_t dl_map_min;
    MHAParser::float_t dl_map_max;
    MHAParser::float_t l_min;
    MHAParser::mfloat_t g_scale;
    void setchannels(unsigned int ch,unsigned int bnds){channels=ch;bands=bnds
U:---  dc.cpp       4% L32   Git-gui-devel  (C++/l Abbrev)
```

Target groups

I Audiological researchers

- Plug-and play software modules
- Easily change processing parameters on high level

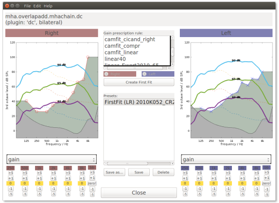
II Application engineers

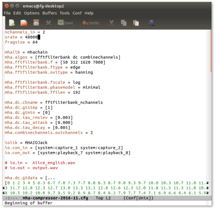
- Set up measurement tools and customize algorithms
- Access configuration interface at an advanced level

III Plugin developers

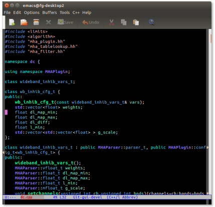
- Develop and implement new plugins in the openMHA framework

... work on **one** software platform.





```
nchannels_in = 2
srate = 48000
fragsize = 64

mhalib = mhachain
mha.algos = [fftfilterbank dc combinechannels]
mha.fftfilterbank.f = [50 312 1620 7000]
mha.fftfilterbank.ftype = edge
mha.fftfilterbank.ovltype = hanning

mha.fftfilterbank.fscale = log
mha.fftfilterbank.phasemodel = minimal
mha.fftfilterbank.fftlen = 192

mha.dc.chname = fftfilterbank_nchannels
mha.dc.gtstep = [1]
mha.dc.gtmin = [0]
mha.dc.tau_rmslev = [0.003]
mha.dc.tau_attack = [0.000]
mha.dc.tau_decay = [0.005]
mha.combinechannels.outchannels = 2

iolib = MHAIOJack
io.con_in = [system:capture_1 system:capture_2]
io.con_out = [system:playback_7 system:playback_8]

# io.in =  Alice_english.wav
# io.out = output.wav

mha.dc.gtdata = [...
[0 1 2 3 4 5 6.3 6.7 7.0 7.3 7.7 8.0 8.3 8.7 9.0 9.3 9.7 10.0 10.3 10.7 11.0 11.
3 11.7 12.0 12.3 12.7 13.0 13.3 13.1 12.8 12.6 12.3 12.0 11.8 11.5 11.3 11.0 10.
8 10.5 10.2 10.0 9.7 9.5 9.2 8.9 8.7 8.4 8.2 7.9 7.7 7.4 7.1 6.9 6.6 6.4 6.1 5.9
-:---  mha-compressor-2016-11.cfg   Top L2     (Conf[Unix])
Beginning of buffer
```

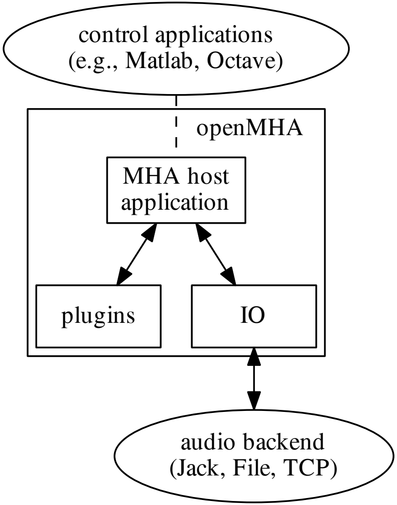


```
#include <limits>
#include <algorithm>
#include "mha_plugin.hh"
#include "mha_tablelookup.hh"
#include "mha_filter.hh"

namespace dc {

using namespace MHAPlugin;

class wideband_inhib_vars_t;

class wb_inhib_cfg_t {
public:
    wb_inhib_cfg_t(const wideband_inhib_vars_t& vars);
    std::vector<float> weights;
    float dl_map_min;
    float dl_map_max;
    float dl_diff;
    float l_min;
    std::vector<std::vector<float> > g_scale;
};

class wideband_inhib_vars_t : public MHAParser::parser_t, public MHAPlugin::conf
ig_t<wb_inhib_cfg_t> {
public:
    wideband_inhib_vars_t();
    MHAParser::vfloat_t weights;
    MHAParser::float_t dl_map_min;
    MHAParser::float_t dl_map_max;
    MHAParser::float_t l_min;
    MHAParser::mfloat_t g_scale;
    void setchannels(unsigned int ch,unsigned int bnds){channels=ch;bands=bnds
U:---  dc.cpp        4% L32   Git-gui-devel  (C++/l Abbrev)
```

openMHA - structure

Basic framework

- MHA host application
- toolbox with plugins
- communication interfaces (control applications, I/O)
- runtime configuration changes



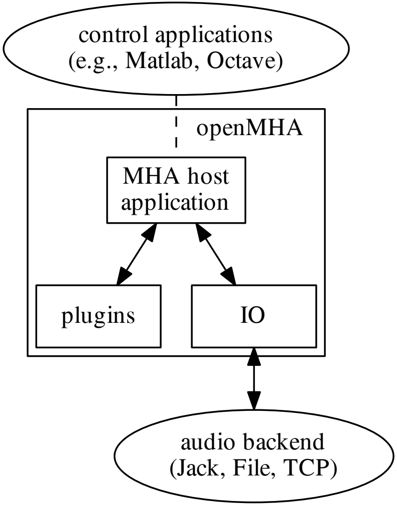
openMHA - structure

Basic framework

- MHA host application
- toolbox with plugins
- communication interfaces (control applications, I/O)
- runtime configuration changes

Tools and documentation

- Matlab/Octave GUI for hearing aid fitting
- manuals for different usage scenarios
- example configuration files for algorithms included

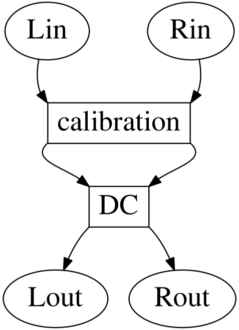


openMHA is available for algorithm development and test

... under Linux and Mac operating systems.

Hearing aid processing methods, release 4.5.5 (Feb. 14, 2018)

- calibration
- multi-band dynamic compressor
- adaptive feedback cancellation



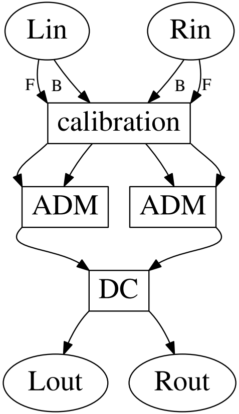
openMHA - current status

openMHA is available for algorithm development and test

... under Linux and Mac operating systems.

Hearing aid processing methods, release 4.5.5 (Feb. 14, 2018)

- calibration
- multi-band dynamic compressor
- adaptive feedback cancellation
- single-channel noise reduction
- adaptive differential microphone



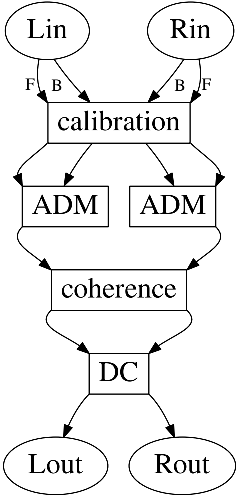
openMHA - current status

openMHA is available for algorithm development and test

... under Linux and Mac operating systems.

Hearing aid processing methods, release 4.5.5 (Feb. 14, 2018)

- calibration
- multi-band dynamic compressor
- adaptive feedback cancellation
- single-channel noise reduction
- adaptive differential microphone
- binaural coherence filter
- binaural beamforming algorithms



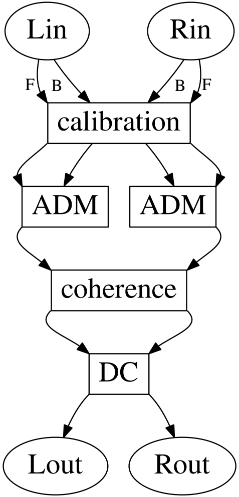
openMHA - current status

openMHA is available for algorithm development and test

... under Linux and Mac operating systems.

Hearing aid processing methods, release 4.5.5 (Feb. 14, 2018)

- calibration
- multi-band dynamic compressor
- adaptive feedback cancellation
- single-channel noise reduction
- adaptive differential microphone
- binaural coherence filter
- binaural beamforming algorithms
- sound source localization



openMHA - current status

openMHA is available for algorithm development and test

... under Linux and Mac operating systems.

Hearing aid processing methods, release 4.5.5 (Feb. 14, 2018)

- calibration
- multi-band dynamic compressor
- adaptive feedback cancellation
- single-channel noise reduction
- adaptive differential microphone
- binaural coherence filter
- binaural beamforming algorithms
- sound source localization

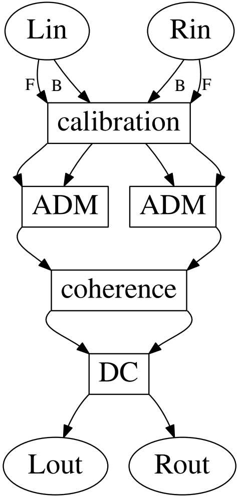
What's next?

Y2 Windows support, ARM board AD/DA converter, usability

Y3 More algorithms and tools

Y4 Updates based on community

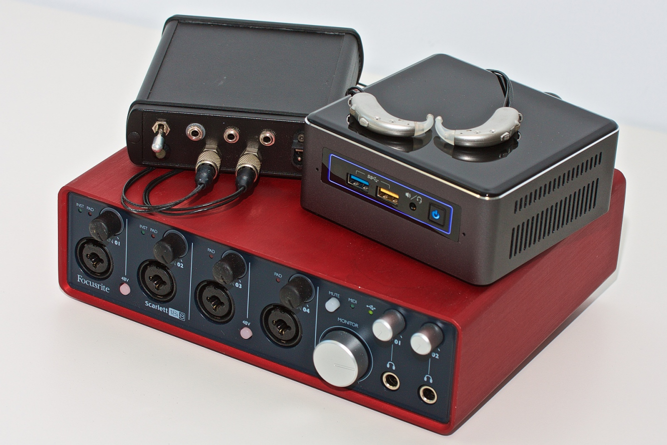
Y5 Further updates and extension



Hardware platforms

Desktop setup

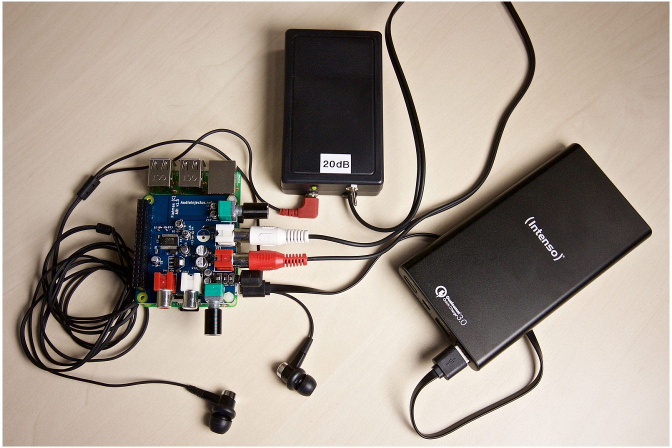
- hearing aids + preamplifier
- small-form-factor PC
- multi-channel audio interface



Hardware platforms

openMHA user project

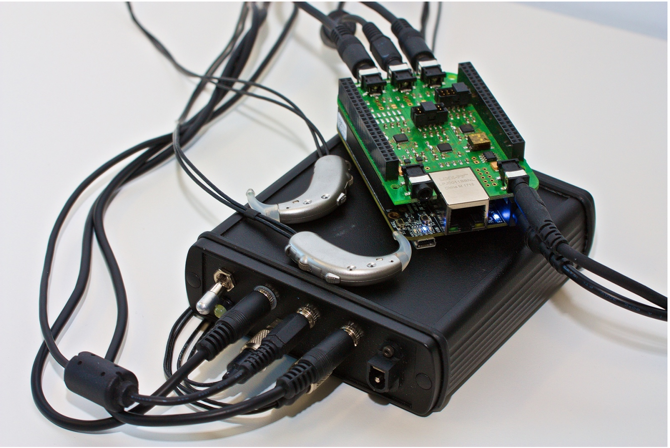
- binaural microphones / earphones
- single-board computer (Raspberry Pi)
- Audio injector
- power bank



Hardware platforms

Self-developed setup

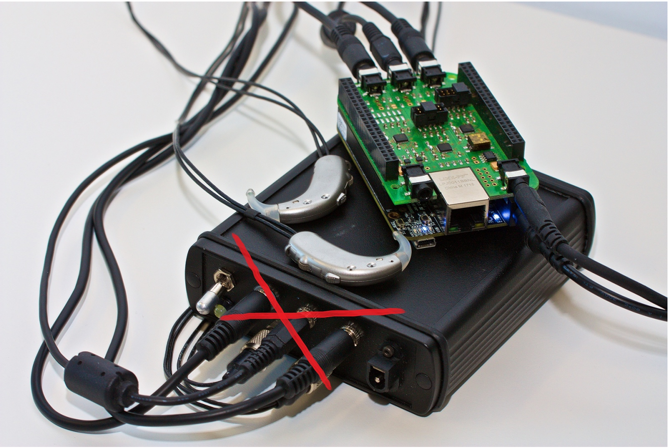
- hearing aids + preamplifier
- single-board computer (Beaglebone Black)
- open hardware audio extension board **"Cape4All"**



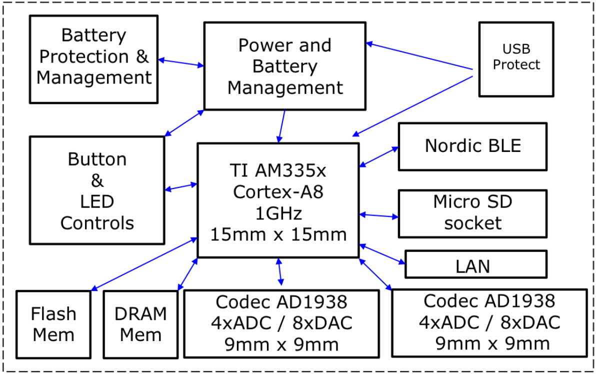
Hardware platforms

Self-developed setup

- hearing aids + preamplifier
- single-board computer (Beaglebone Black)
- open hardware audio extension board **"Cape4All"**
- features power supply for microphones



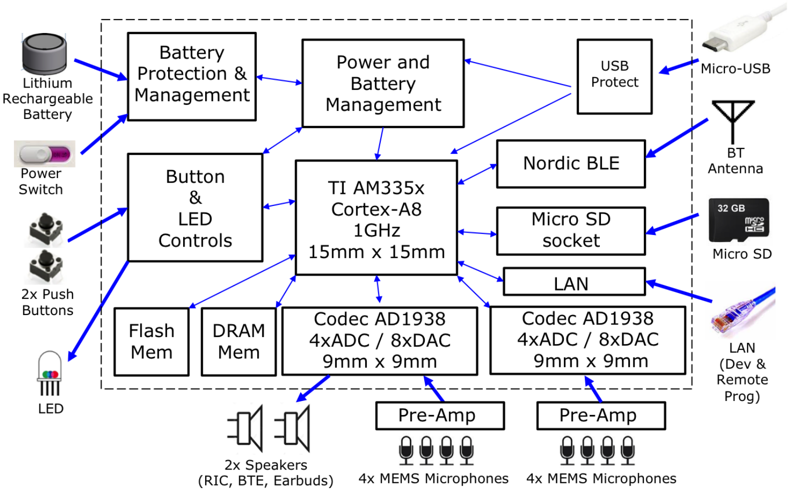
Portable Hearing Laboratory - PHL


Battery Protection & Management
Power and Battery Management
USB Protect
Button & LED Controls
TI AM335x Cortex-A8 1GHz 15mm x 15mm
Nordic BLE
Micro SD socket
LAN
Flash Mem
DRAM Mem
Codec AD1938 4xADC / 8xDAC 9mm x 9mm
Codec AD1938 4xADC / 8xDAC 9mm x 9mm

1st Generation

- ARM Cortex A8 Core
- Linux environment

Portable Hearing Laboratory - PHL



Lithium Rechargeable Battery

Power Switch

2x Push Buttons

LED

Battery Protection & Management

Power and Battery Management

USB Protect

Micro-USB

Button & LED Controls

TI AM335x Cortex-A8 1GHz 15mm x 15mm

Nordic BLE

BT Antenna

Micro SD socket

32 GB

Micro SD

LAN

LAN (Dev & Remote Prog)

Flash Mem

DRAM Mem

Codec AD1938 4xADC / 8xDAC 9mm x 9mm

Codec AD1938 4xADC / 8xDAC 9mm x 9mm

Pre-Amp

Pre-Amp

2x Speakers (RIC, BTE, Earbuds)

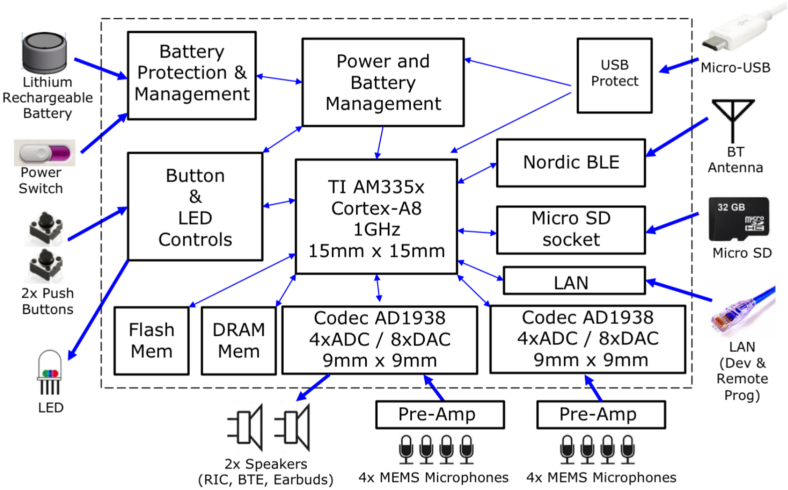
4x MEMS Microphones

4x MEMS Microphones

1st Generation

- ARM Cortex A8 Core
- Linux environment
- Bluetooth low energy (BLE) module module
- 3 alternative output interfaces: RIC, BTE, earbuds

Portable Hearing Laboratory - PHL



Lithium Rechargeable Battery

Power Switch

2x Push Buttons

LED

Battery Protection & Management

Power and Battery Management

USB Protect

Micro-USB

Button & LED Controls

TI AM335x Cortex-A8 1GHz 15mm x 15mm

Nordic BLE

BT Antenna

Micro SD socket

32 GB

Micro SD

LAN

LAN (Dev & Remote Prog)

Flash Mem

DRAM Mem

Codec AD1938 4xADC / 8xDAC 9mm x 9mm

Codec AD1938 4xADC / 8xDAC 9mm x 9mm

Pre-Amp

Pre-Amp

2x Speakers (RIC, BTE, Earbuds)

4x MEMS Microphones

4x MEMS Microphones

1st Generation

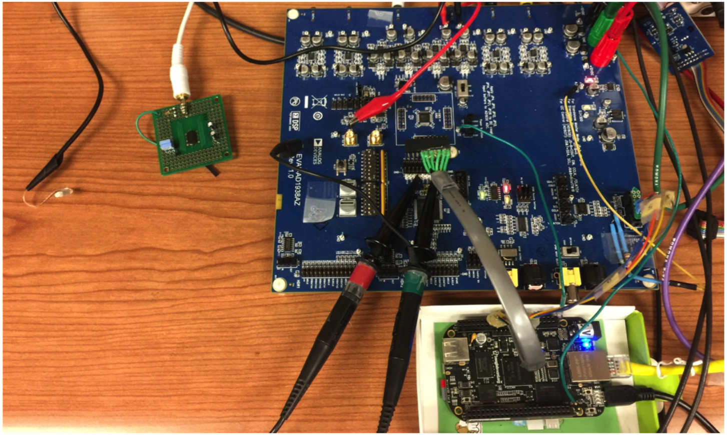
- ARM Cortex A8 Core
- Linux environment
- Bluetooth low energy (BLE) module module
- 3 alternative output interfaces: RIC, BTE, earbuds

2nd Generation

- Miniaturized form factor
- Licensed soft core on own silicone

PHL prototype hardware

PHL hardware development platform



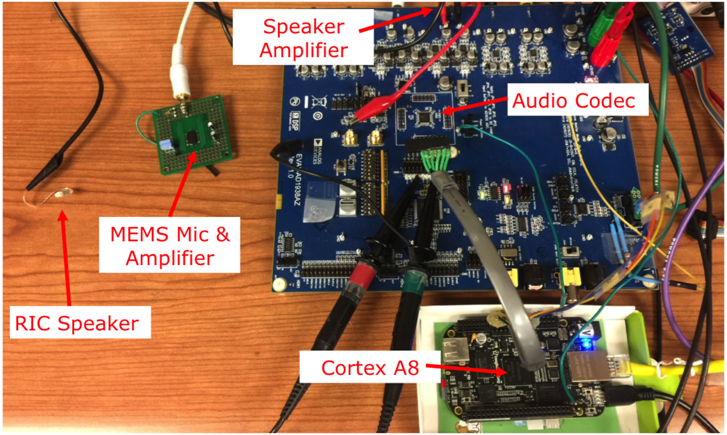
Year 1 schedule

Feb 07 Circuit designs & bench prototype testing (Hardware & Linux)

Mar 2 Details schematics and component specifications

PHL prototype hardware

PHL hardware development platform



Speaker Amplifier

Audio Codec

MEMS Mic & Amplifier

RIC Speaker

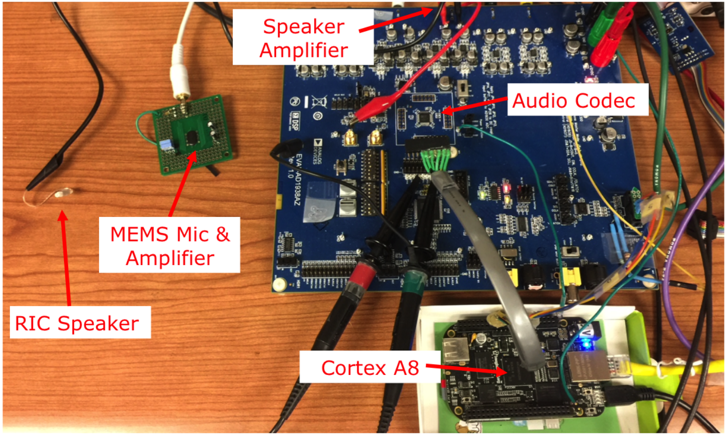
Cortex A8

Year 1 schedule

Feb 07	Circuit designs & bench prototype testing (Hardware & Linux)
Mar 2	Details schematics and component specifications

PHL prototype hardware

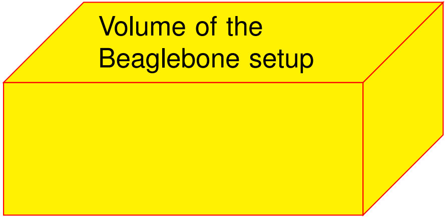
PHL hardware development platform



Year 1 schedule

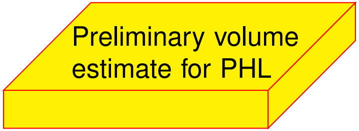
Feb 07	Circuit designs & bench prototype testing (Hardware & Linux)
Mar 2	Details schematics and component specifications
Mar 15	Layout
Apr 1	PCBs produced and parts sourced
Apr 15	Board assembled
May 25	PCB 1st version tested (Hardware, Linux, MHA)
Jun 30	PCB 2nd version produced

PHL specifications



Volume of the Beaglebone setup

PHL specifications

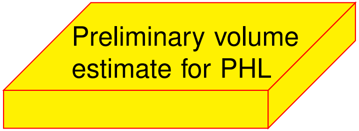

Preliminary volume estimate for PHL

- Length: 7 cm
- Width: 6 cm
- Thickness: 1 cm

(-75%)

(actual form factor and microphone positioning are still to be determined)

PHL specifications



- Length: 7 cm
- Width: 6 cm
- Thickness: 1 cm

(-75%)

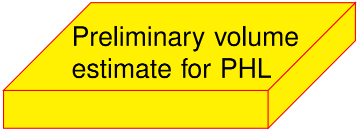
(actual form factor and microphone positioning are still to be determined)

openMHA on PHL

Basic MHA algorithms to be optimized for A8

- Calibration
- STFT filter-bank
- Directional microphone (ADM)
- FFT-based multi-band compression
- Wind noise reduction
- Binaural coherence filter

PHL specifications


Preliminary volume estimate for PHL

- Length: 7 cm
- Width: 6 cm
- Thickness: 1 cm

(-75%)

(actual form factor and microphone positioning are still to be determined)

CPU	current	RT factor	headroom
A8	500 mA	**2.0x**	**50 %**
A7	175 mA	1.3x	17 %

openMHA on PHL

Basic MHA algorithms to be optimized for A8

- Calibration
- STFT filter-bank
- Directional microphone (ADM)
- FFT-based multi-band compression
- Wind noise reduction
- Binaural coherence filter

- 2x real-time execution of basic algorithms

→ 50% headroom for custom algorithms

Milestones

Portable Hearing Laboratory

Y1 First engineering prototype in a non-final form factor exposing test points and connections for detailed testing

Y2 First engineering prototype in the actual form factor ready for initial subject testing

Y3 Final Device ready for field testing

Y4 Community testing and modifications

News and software

openMHA is open source under AGPL-3.0 license

Code is available here:

https://github.com/HoerTech-gGmbH/openMHA

Latest news:

- www.openMHA.org
- http://batandcat.com/newsdownloads-on-open-platform.html

Disclaimer Research reported in this publication was supported by the National Institute On Deafness And Other Communication Disorders of the National Institutes of Health under Award Numbers R01DC015429 and R44DC016257. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.