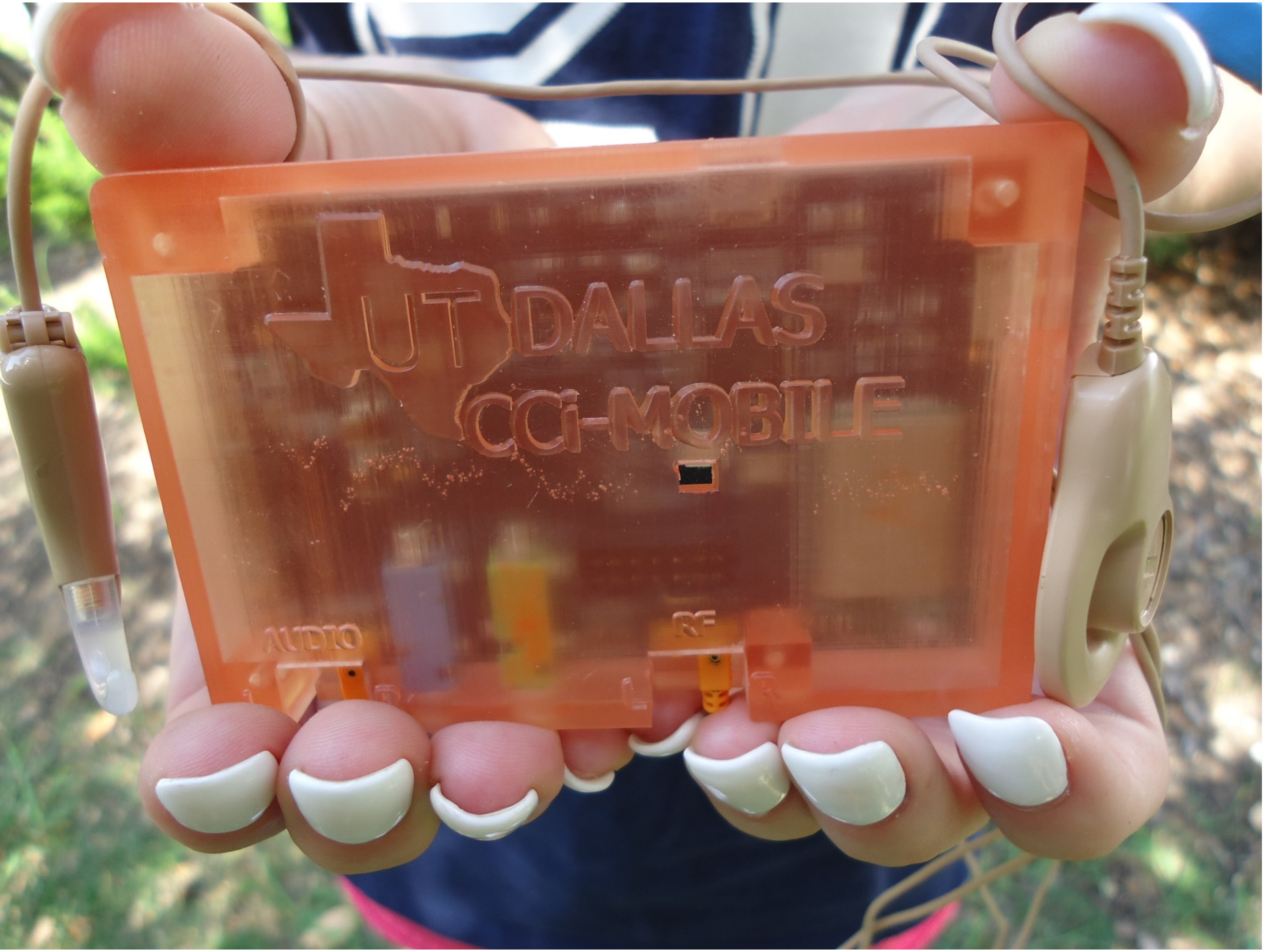


CCi-MOBILE Research Platform for Cochlear Implants and Hearing Aids

HANDS-ON WORKSHOP

July 18, 2017 CIAP-2017

John H.L. Hansen, Hussnain Ali, Sandeep Ammula, Juliana Saba

Cochlear Implant Laboratory
Center for Robust Speech Systems
The University of Texas at Dallas

UTD
Cochlear Implant Laboratory
UTD CRSS

Acknowledgements - Sponsors

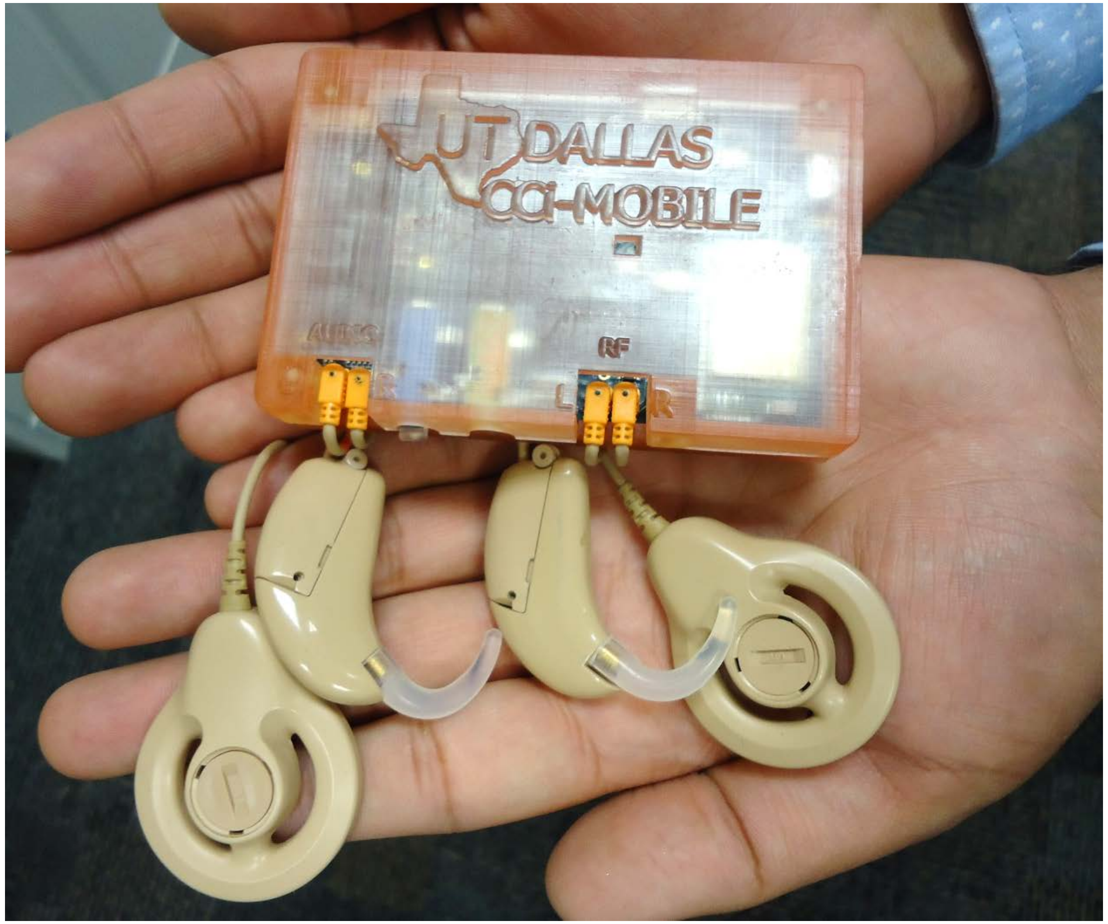
This work was supported by Grant R01 DC010494-01A awarded from the NIH/NIDCD

Special thanks to Cochlear Corp. for their sustained contributions and collaborations

Agenda

- Getting to know the hardware
  - Multiple configurations to use the platform
- Software
  - MATLAB
  - Android
- How to use the platform in your research
  - Equipment needed
- How to conduct human testing with the platform
- Resources
- Demos + Hands-on session

Platform – at a glance



Compatible with:

- Cochlear implant Cochlear Corp. (CI24)
- Hearing aid transducers

Supports:

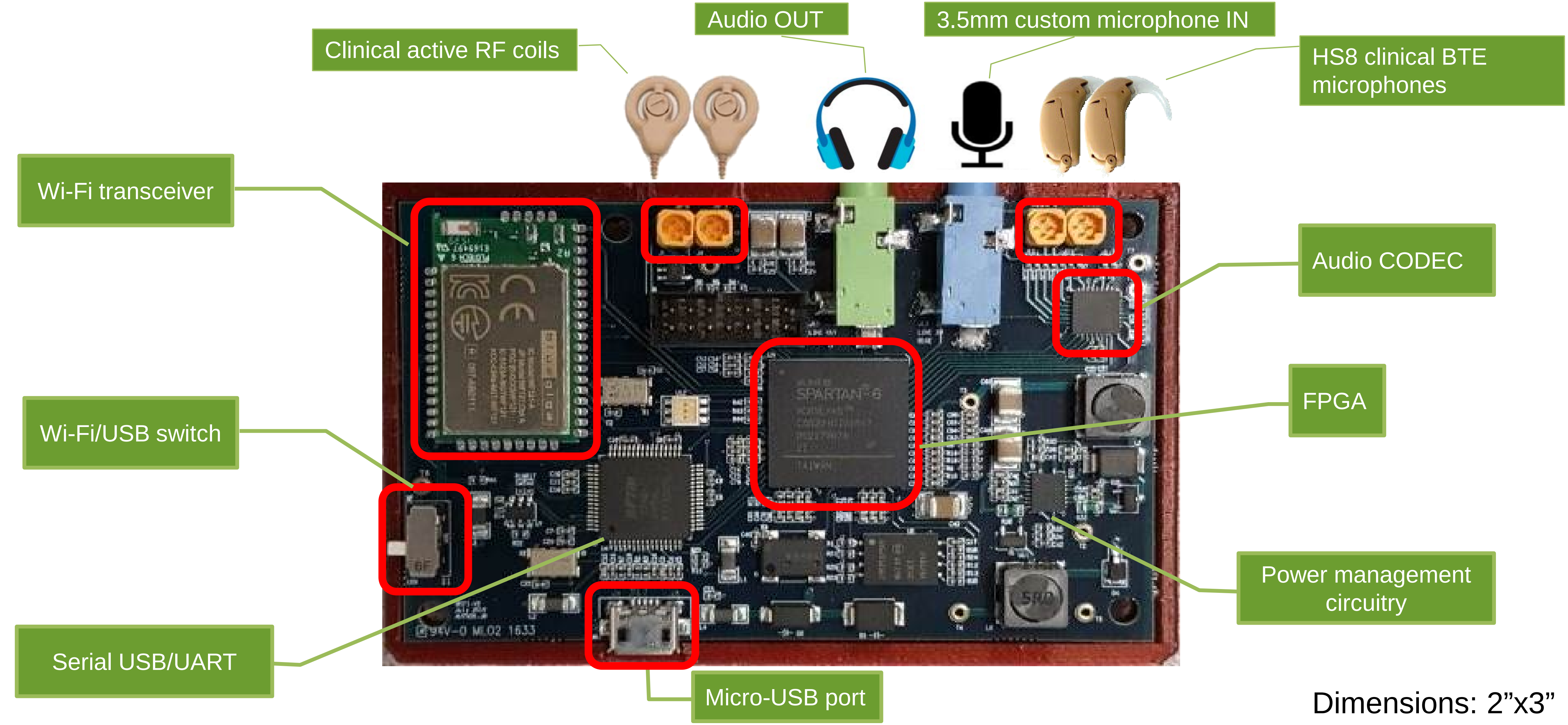
- Unilateral and time synchronized Bilateral electric stimulation
- Synchronized acoustic stimulation – bimodal/electric+acoustic stim.

Modes:

- PC-based for Laboratory studies
- Android for field trials

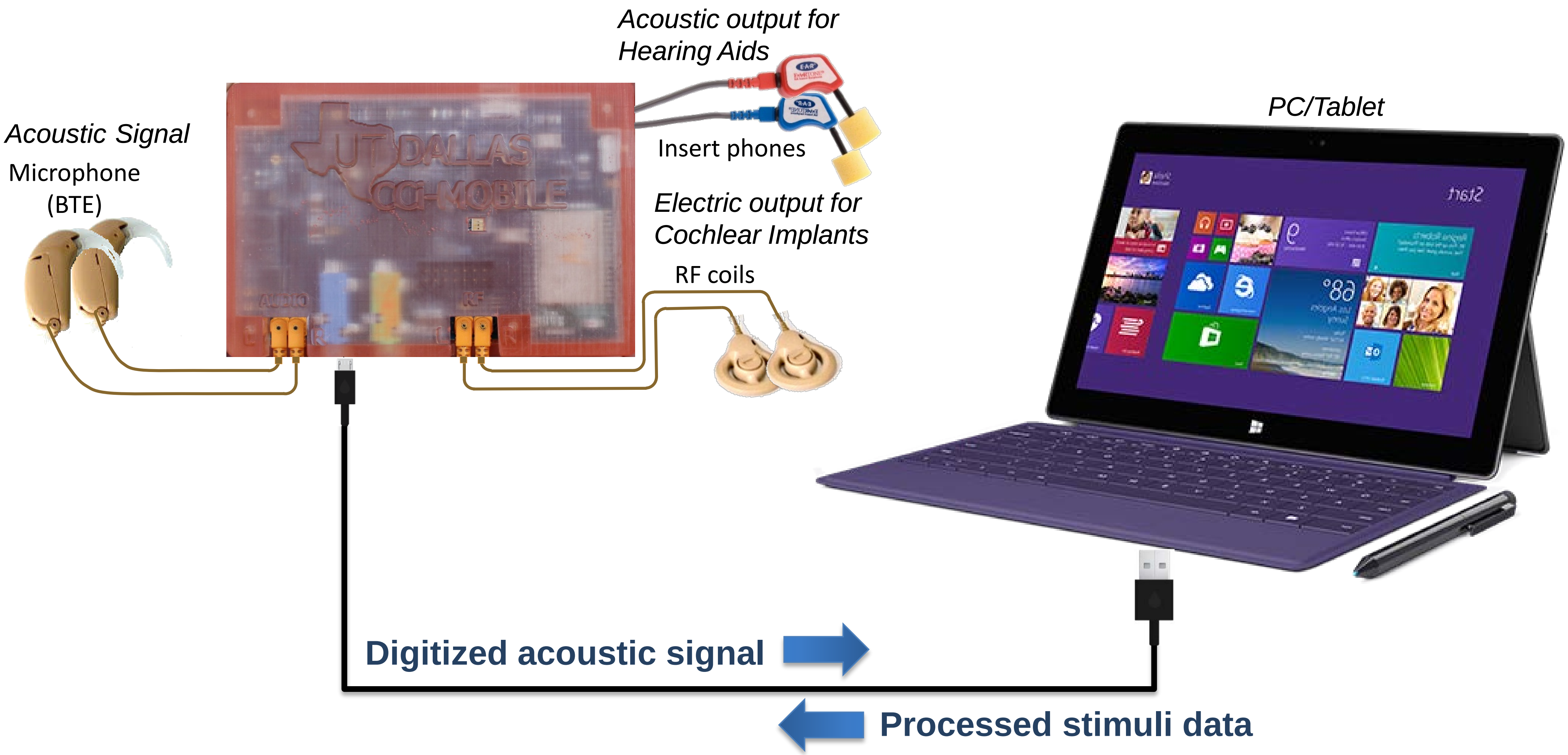
- Plug-and-play
- Portable, wearable
- On-the-go adjustment of sound processing parameters

Interface Board

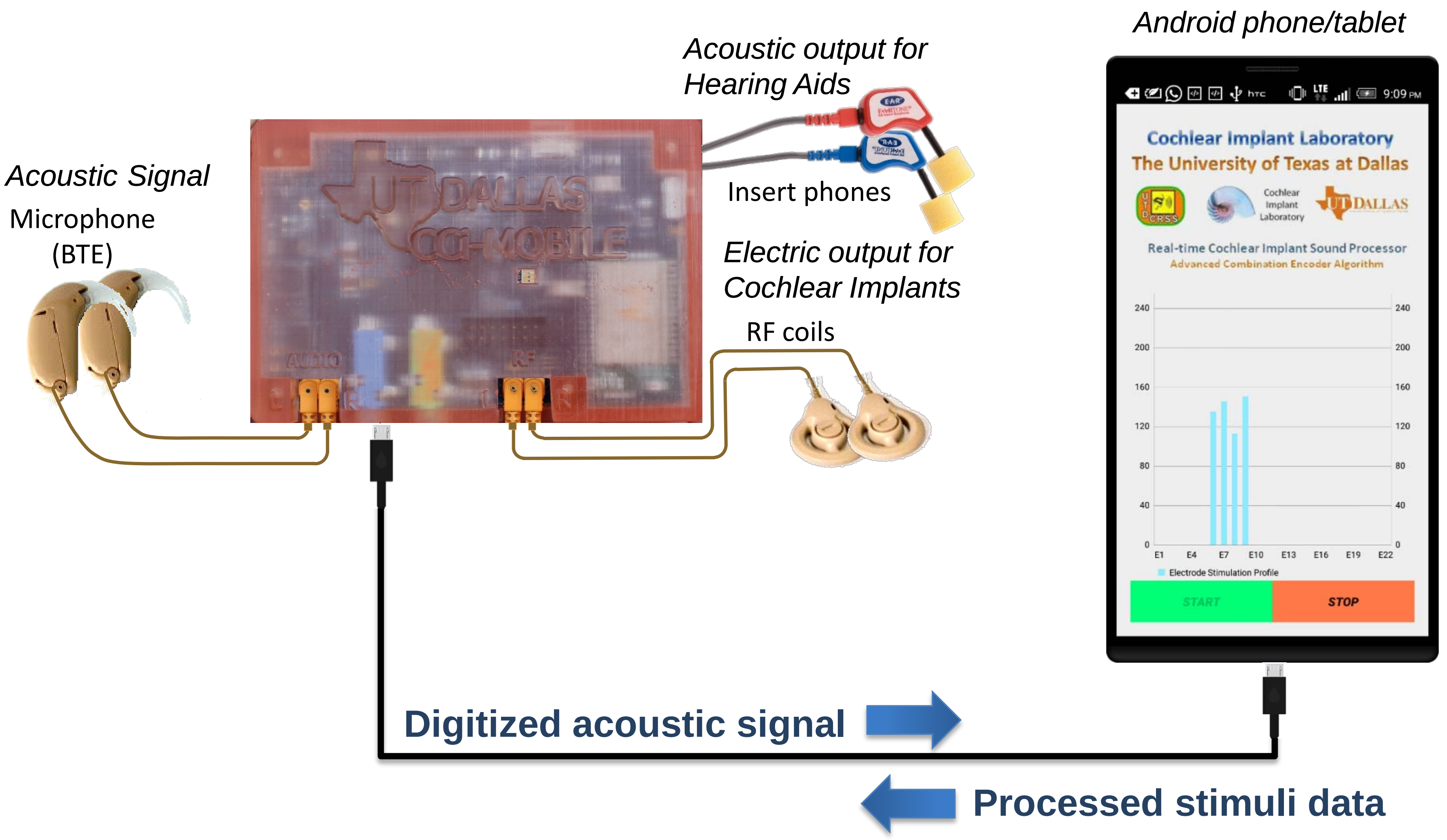


Dimensions: 2”x3”

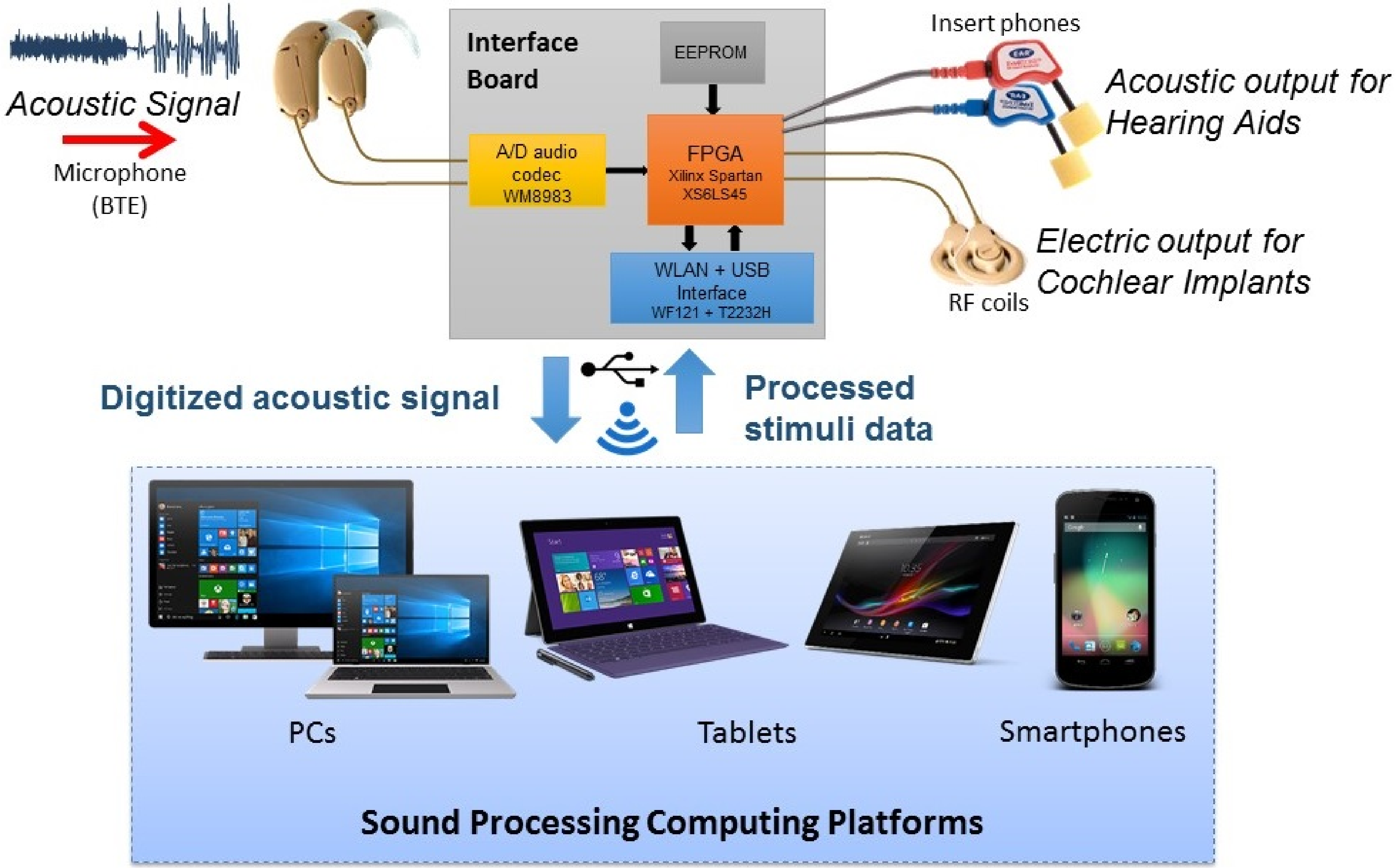
PC-based Setup



Android-based Setup



High-level Overview



Software

PC

MATLAB-based

Applications for:
- Real-time Processing
- Audio File Processing
- Audio Recording
- Signal visualization
- Psychophysics

Android

JAVA

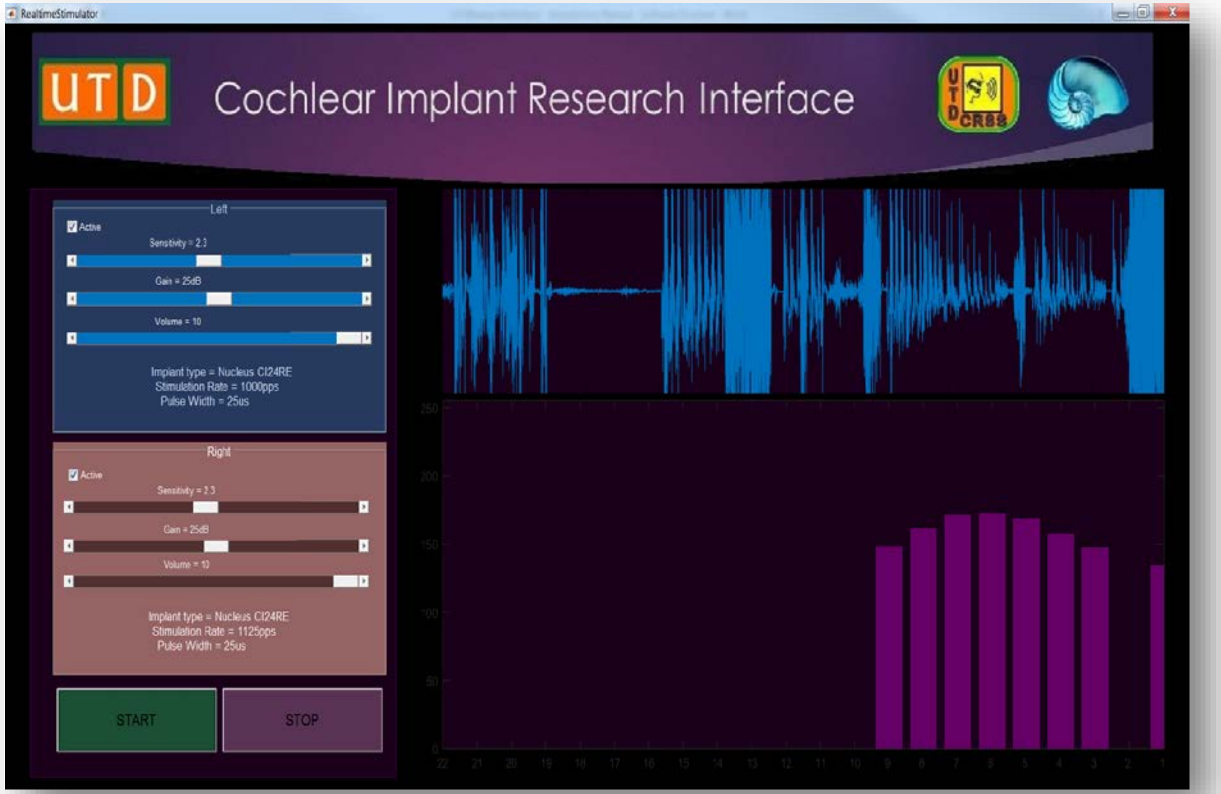
Applications for:
- Real-time Processing
- Audio Recording
- On-the-go map changes
- DEMOs

PC – MATLAB-based software

Real-time MATLAB capability

- Integrate your existing MATLAB algorithms
- Access to microphone signals
- Complete control on stimuli
- **Modular** software design
- Simple, **intuitive,** easy-to-understand, and straight-forward

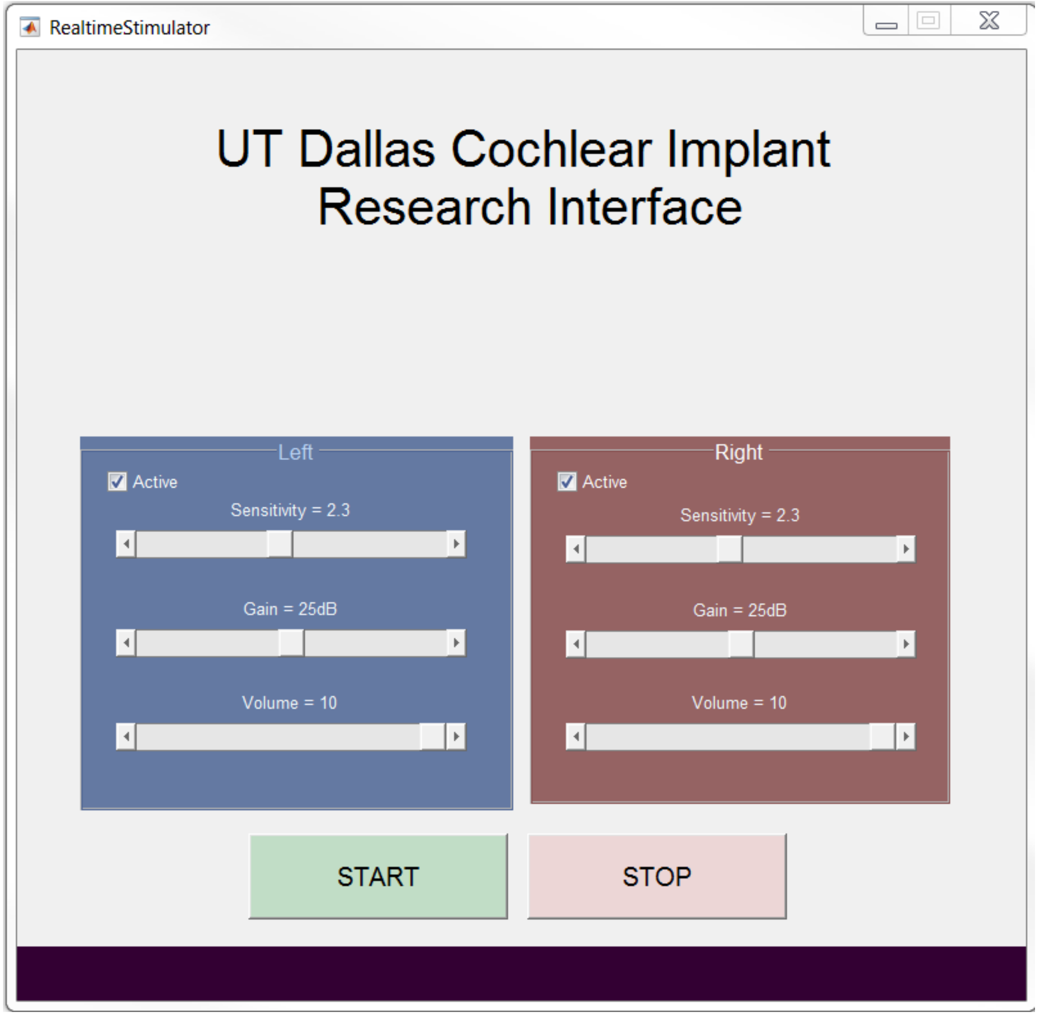
MATLAB applications – DEMO



- Visualize acquired acoustic signal waveform in realtime
- Bilateral ACE processing
- Visualize pulse sequence generated by processing
- Change parameters on the go

NOT recommended for human use

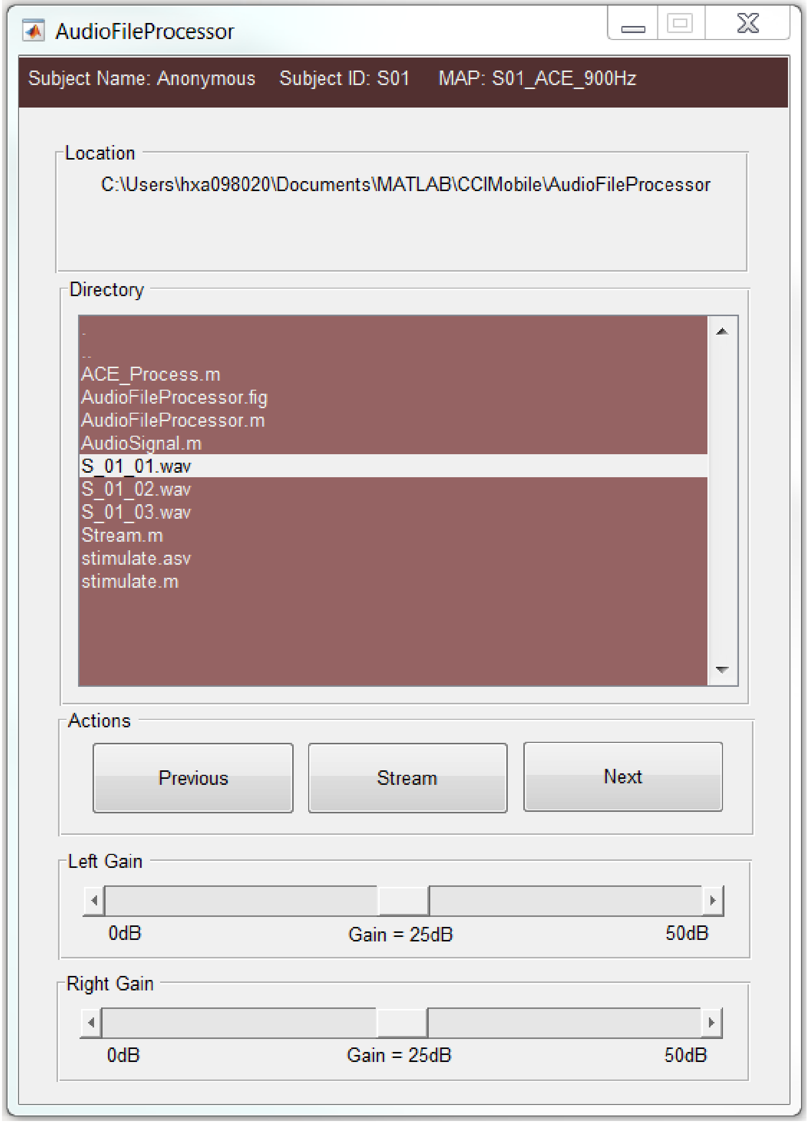
MATLAB applications – Realtime ACE



- For real-time bilateral ACE processing **in Free field**
- Audio signal from microphone
- Complete control on processing, parameters, and stimulation.
- Variable rate supported on frame-by-frame basis
- Bimodal version also available

- Perfect for HUMAN use

MATLAB applications – Offline ACE

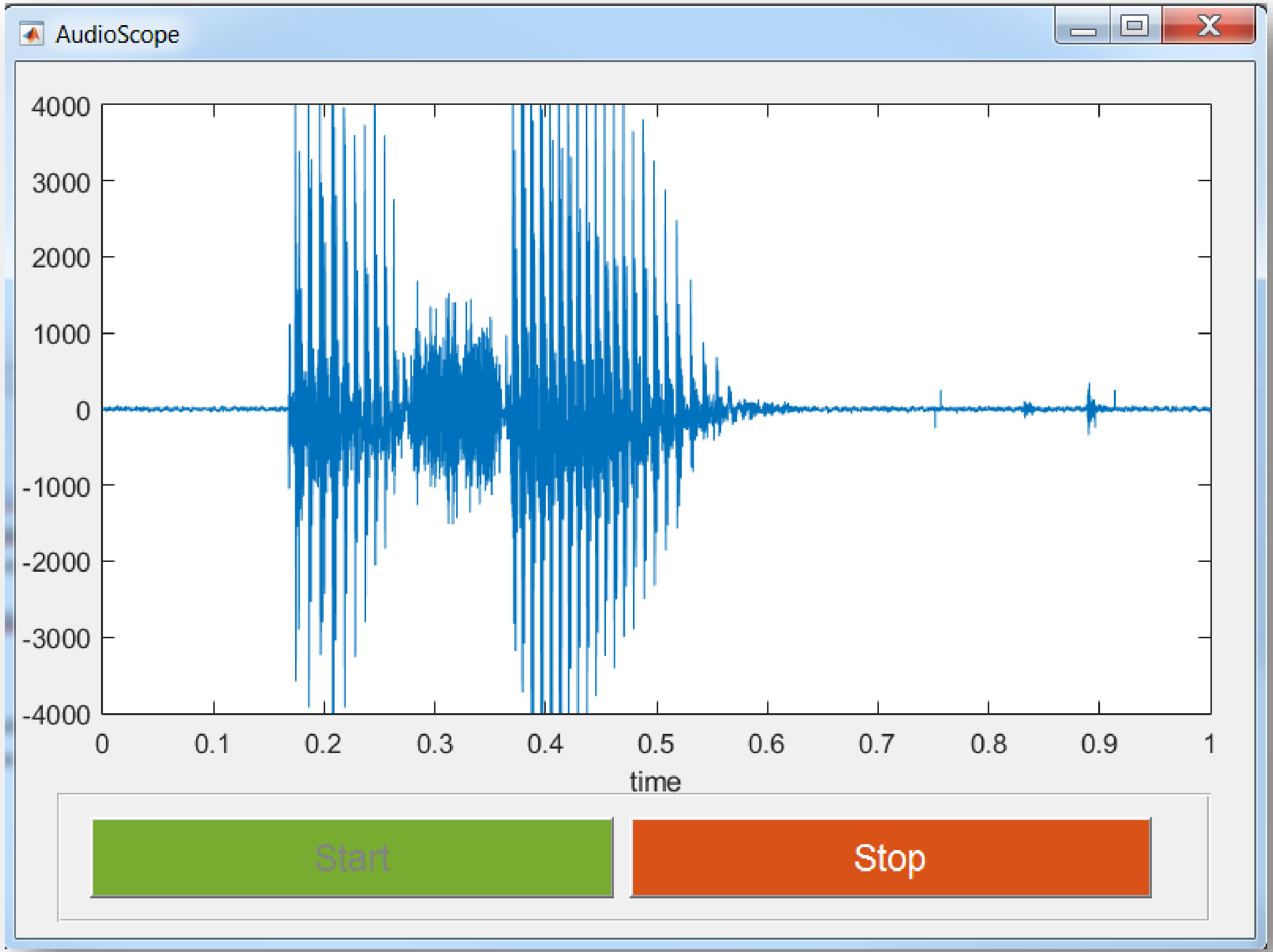


- For processing audio wave files from your PC
- streaming processed files directly to implant
- Complete control on processing, parameters, and stimulation.
- Variable rate supported on frame-by-frame basis
- Bimodal version also available

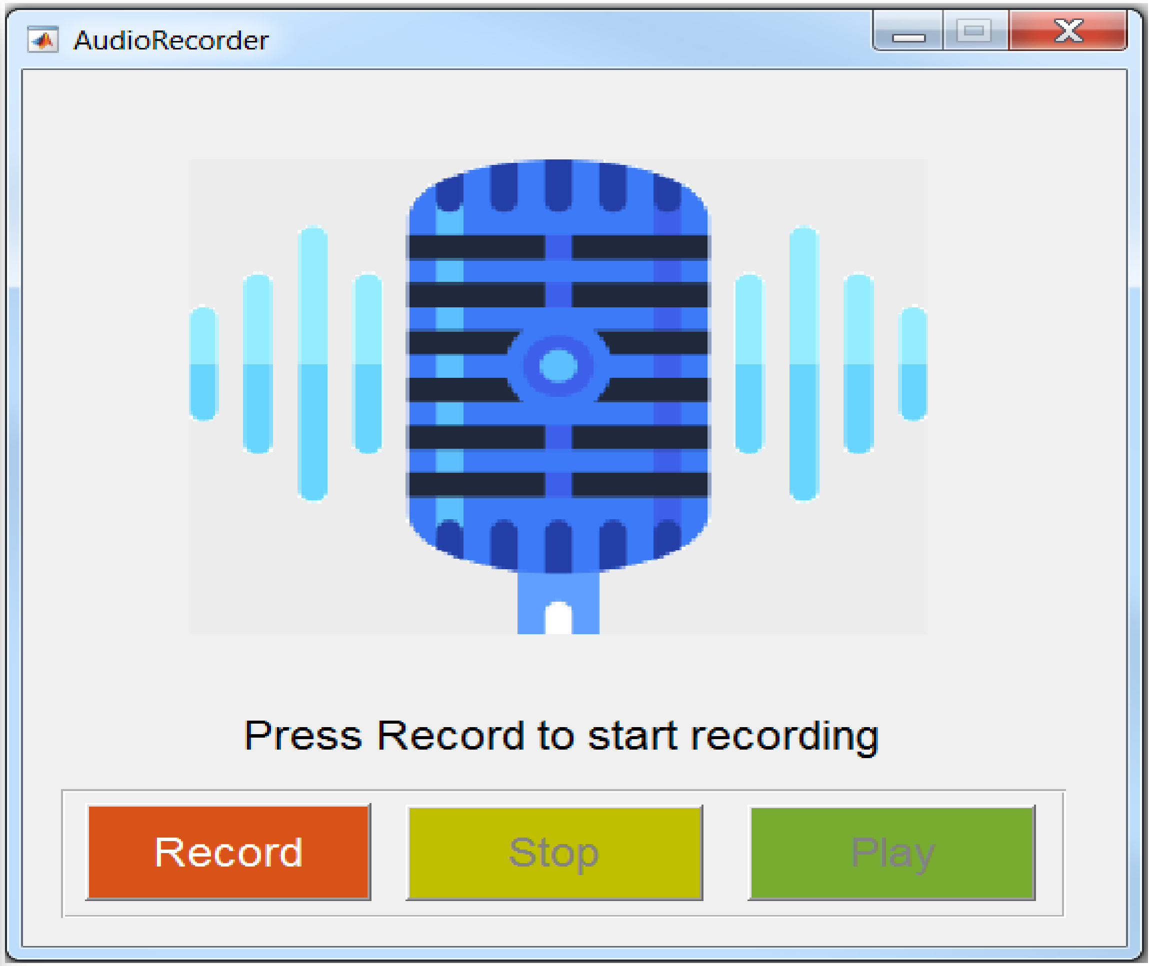
- Perfect for HUMAN testing

MATLAB applications – Audio Recorder/Visualizer

Audio Scope



Audio Recorder

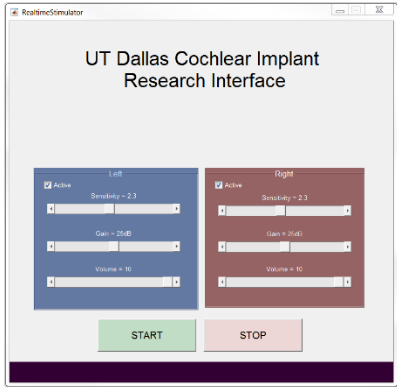


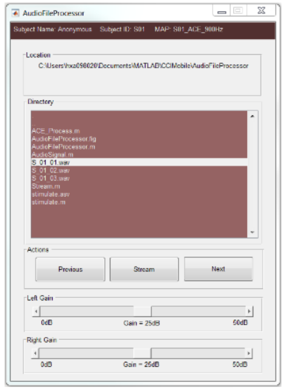
Recommended for Debugging

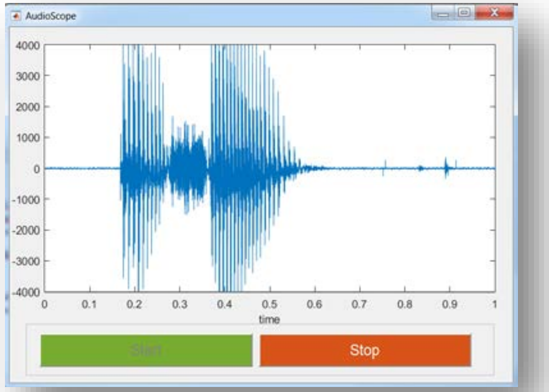
MATLAB bimodal (EAS) applications

- 4 channels of time-synchronized stimulation
  - 2 channels of electric stimulation (left and right)
  - 2 channels of acoustic stimulation (left and right)

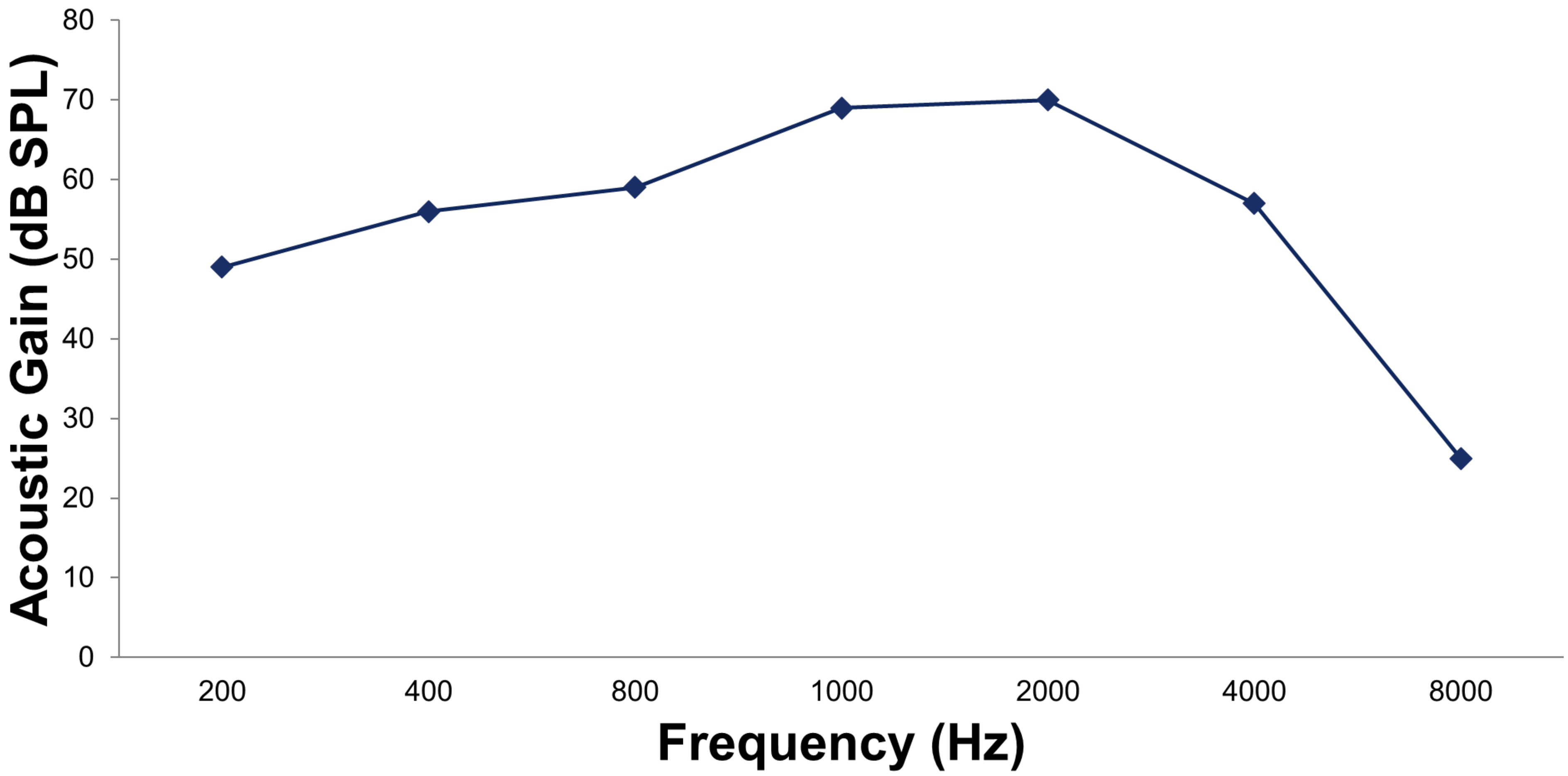
- Hearing-aid routines can be implemented as per your needs.
- Similar suite of applications available in bimodal mode.







Electric + Acoustic Stimulation Gain Profile



MAP

```matlab
MAP.General.SubjectName      = 'Subject 01';        % Optional: Subject Name
MAP.General.SubjectID        = 'S01';               % Optional: Random Subje
MAP.General.MapTitle         = 'S01_ACE_900Hz';     % Optional: Map Title

MAP.General.NumberOfImplants = 2;                   % '1' for Unilateral and
MAP.General.ImplantedEar     = 'bilateral';         % 'Left' for left side;
MAP.General.StimulateEars    = 'Both';              % 'Left' for left only;

%% Left Ear Parameters
%  remove this section if left side does not exist
MAP.Left.ImplantType          = 'CI24RE';      %Implant chip type, e.g., CI24R
MAP.Left.SamplingFrequency    = 16000;         % Fixed
MAP.Left.NumberOfChannels     = 22;            % 22 fixed for imlants from Coc
MAP.Left.Strategy             = 'ACE';         % 'ACE' or 'CIS' or 'Custom'
MAP.Left.Nmaxima              = 8;             % Nmaxima 1 - 22 for n-of-m str
MAP.Left.StimulationMode      = 'MP1+2';       % Electrode Configuration/Stimu
MAP.Left.StimulationRate      = 1000;          % Stimulation rate per electrod
MAP.Left.PulseWidth           = 25;            % Pulse width in us
MAP.Left.IPG                  = 8;             % Inter-Phase Gap (IPG) fixed a
MAP.Left.Sensitivity          = 2.3;           % Microphone Sensitivity (adjus
MAP.Left.Gain                 = 25;            % Global gain for envelopes in
MAP.Left.Volume               = 10;            % Volume Level on a scale of 0
MAP.Left.Q                    = 20;            % Q-factor for the compression
MAP.Left.BaseLevel            = 0.0156;        % Base Level
MAP.Left.SaturationLevel      = 0.5859;        % Saturation Level
MAP.Left.ChannelOrderType     = 'base-to-apex'; % Channel Stimulation Order t
MAP.Left.FrequencyTable       = 'Default';     % Frequency assignment for each
MAP.Left.Window               = 'Hanning'      % Window type
MAP.Left.El_CF1_CF2_THR_MCL_Gain = [
  % El  F_Low   F_High  THR     MCL     Gain
    22  188     313     100     200     0.0
    21  313     438     100     200     0.0
    20  438     563     100     200     0.0
    19  563     688     100     200     0.0
    18  688     813     100     200     0.0
    17  813     938     100     200     0.0
    16  938     1063    100     200     0.0
    15  1063    1188    100     200     0.0
    14  1188    1313    100     200     0.0
    13  1313    1563    100     200     0.0
    12  1563    1813    100     200     0.0
    11  1813    2063    100     200     0.0
    10  2063    2313    100     200     0.0
    9   2313    2688    100     200     0.0
    8   2688    3063    100     200     0.0
    7   3063    3563    100     200     0.0
    6   3563    4063    100     200     0.0
    5   4063    4688    100     200     0.0
    4   4688    5313    100     200     0.0
    3   5313    6063    100     200     0.0
    2   6063    6938    100     200     0.0
    1   6938    7938    100     200     0.0
    ];
```

```matlab
%% Right Ear Parameters
%  remove this section if right side does not exist
MAP.Right.ImplantType        = 'CI24RE';
MAP.Right.SamplingFrequency  = 16000;           % Fixed
MAP.Right.NumberOfChannels   = 22;              % 22 fixed for imlants from Cochle
MAP.Right.Strategy           = 'ACE';           % 'ACE' or 'CIS' or 'Custom'
MAP.Right.Nmaxima            = 8;               % Nmaxima 1 - 22 for n-of-m strate
MAP.Right.StimulationMode    = 'MP1+2';         % Electrode Configuration/Stimulat
MAP.Right.StimulationRate    = 1000;            % Stimulation rate per electrode
MAP.Right.PulseWidth         = 25;              % Pulse width in us
MAP.Right.IPG                = 8;               % Inter-Phase Gap (IPG) fixed at
MAP.Right.Sensitivity        = 2.3;             % Microphone Sensitivity (adjustal
MAP.Right.Gain               = 25;              % Global gain for envelopes in dB
MAP.Right.Volume             = 10;              % Volume Level on a scale of 0 to
MAP.Right.Q                  = 20;              % Q-factor for the compression fu
MAP.Right.BaseLevel          = 0.0156;          % Base Level
MAP.Right.SaturationLevel    = 0.556;           % Saturation Level
MAP.Right.ChannelOrderType   = 'base-to-apex';  % Channel Stimulation Order type:
MAP.Right.FrequencyTable      = 'Default';      % Frequency assignment for each
MAP.Right.Window             = 'Hanning'        % Window type
MAP.Right.El_CF1_CF2_THR_MCL_Gain = [
  % El  F_Low   F_High  THR     MCL     Gain
    22  188     313     100     200     0.0
    21  313     438     100     200     0.0
    20  438     563     100     200     0.0
    19  563     688     100     200     0.0
    18  688     813     100     200     0.0
    17  813     938     100     200     0.0
    16  938     1063    100     200     0.0
    15  1063    1188    100     200     0.0
    14  1188    1313    100     200     0.0
    13  1313    1563    100     200     0.0
    12  1563    1813    100     200     0.0
    11  1813    2063    100     200     0.0
    10  2063    2313    100     200     0.0
    9   2313    2688    100     200     0.0
    8   2688    3063    100     200     0.0
    7   3063    3563    100     200     0.0
    6   3563    4063    100     200     0.0
    5   4063    4688    100     200     0.0
    4   4688    5313    100     200     0.0
    3   5313    6063    100     200     0.0
    2   6063    6938    100     200     0.0
    1   6938    7938    100     200     0.0
    ];
MAP.Right.NumberOfBands              = size(MAP.Right.El_CF1_CF2_THR_MCL_Gain, 1);
MAP.Right.Electrodes                 = MAP.Right.El_CF1_CF2_THR_MCL_Gain(:, 1);
MAP.Right.LowerCutOffFrequencies     = MAP.Right.El_CF1_CF2_THR_MCL_Gain(:, 2);
MAP.Right.UpperCutOffFrequencies     = MAP.Right.El_CF1_CF2_THR_MCL_Gain(:, 3);
MAP.Right.THR                        = MAP.Right.El_CF1_CF2_THR_MCL_Gain(:, 4);
MAP.Right.MCL                        = MAP.Right.El_CF1_CF2_THR_MCL_Gain(:, 5);
MAP.Right.BandGains                  = MAP.Right.El_CF1_CF2_THR_MCL_Gain(:, 6);
```

.m file for MATLAB
Easy to read structure

Nucleus MATLAB toolbox + CCiMOBILE

- Option to use Nuclues MATLAB toolbox (NMT) to directly stream stimuli via CCiMOBILE platform
- Example codes are available (see DEMO)

How easy is it to modify the codes
Directory Structure

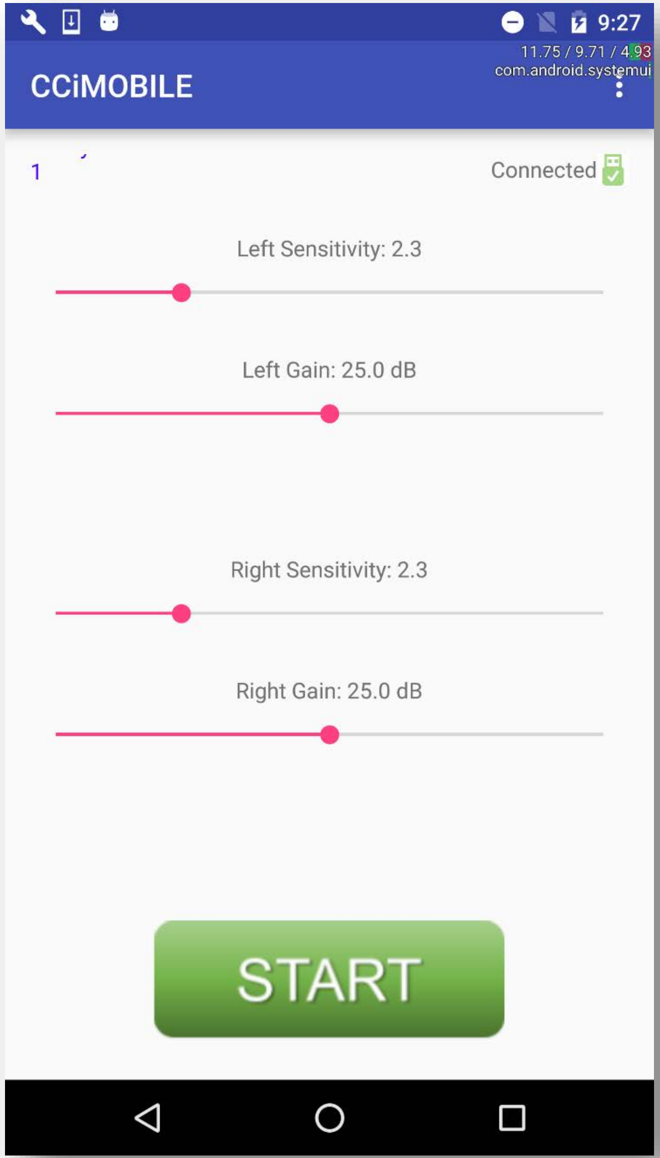
- AudioFileProcessor
- AudioRecorder
- AudioScope
- CommonFunctions
- Demo
- MAPs
- Realtime
- test
- UTDResearchInterface - Introductory Manual - Software Routines.docx
- UTDResearchInterface - Introductory Manual - Software Routines.pdf

- AudioRecorder
  - AudioRecorder.fig
  - AudioRecorder.m
  - BTEaudio.wav
  - mic.png
  - READ ME.txt
- AudioScope
  - AudioScope.fig
  - AudioScope.m
  - READ ME.txt
- CommonFunctions
- Demo
  - ACE_Processing_Realtime.m
  - interface.jpg
  - READ ME.txt
  - RealtimeStimulator.fig
  - RealtimeStimulator.m
- MAPs
- Realtime
  - ACE_Processing_Realtime.m
  - READ ME.txt
  - RealtimeStimulator.fig
  - RealtimeStimulator.m
  - RealtimeStimulator_Script.m
  - SampleMap.m

How easy is it to modify the codes – General Structure

```
while frame_no<nframes-1 % use while else timing won't be right
    if (Wait(s)>= 512)

        %% STEP 1: READ Audio Signal
        AD_data_bytes = Read(s, 512);                          % Read audio from BTE
        AD_data=typecast(int8(AD_data_bytes), 'int16');        % Type cast to short 16 bits
        audio_left = double(AD_data(1:2:end));                 % Type cast to double for processing
        audio_left =  (p.Left.scale_factor).*audio_left;       % Adjust sensitivity

        %% STEP 2: PROCESS AUDIO SIGNAL FRAMES
        stimuli.left = ACE_Processing_Realtime(audio_left, bufferHistory_left, p.Left);

        %% STEP 3: Stream Stimuli
        stimulus = UART_output_buffer(stimuli, p);                 % Fill output Buffers
        Write(s, stimulus,516);

    end
    frame_no = frame_no+1;
end % end while loop for nframes
```

Android applications – Realtime ACE

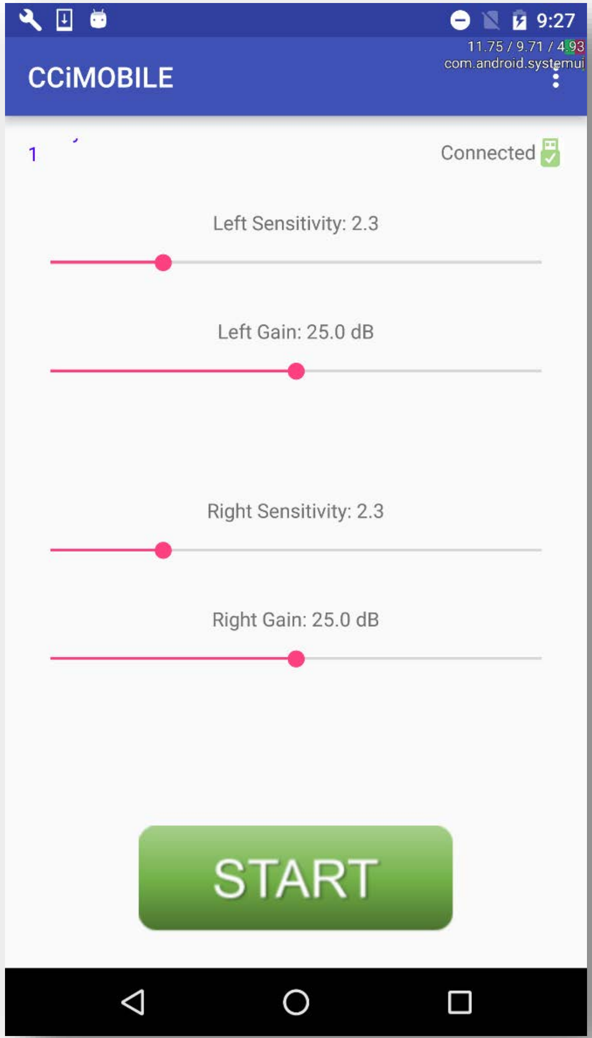


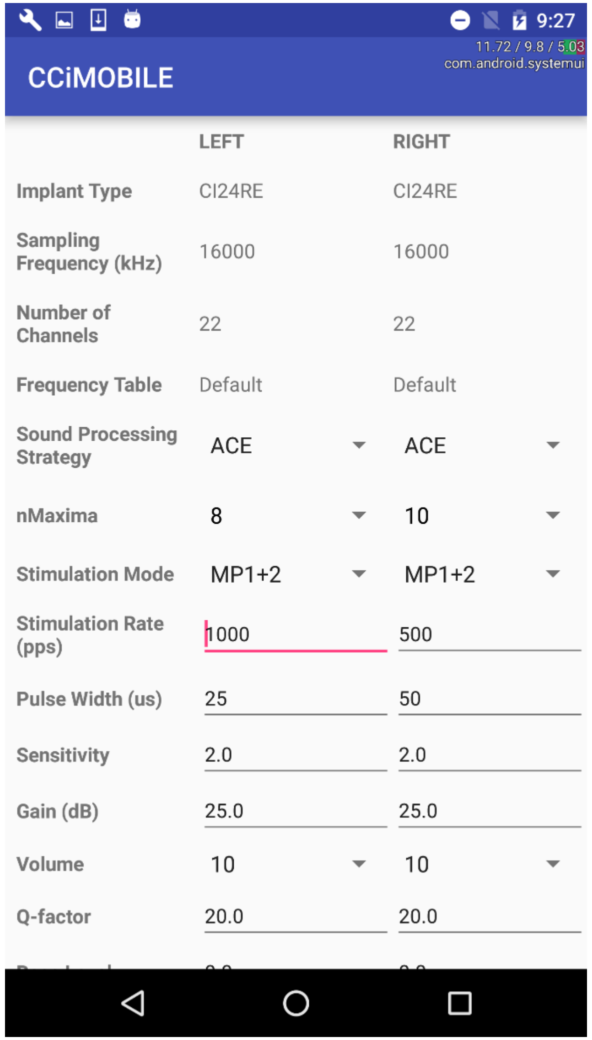
- For real-time bilateral ACE processing
- Audio signal acquired from microphone
- Complete control on sound processing, parameters and stimulation.
- Variable rate supported on frame-by-frame basis
- Change parameters on-the-go
- Bimodal version is currently in works

- Perfect for HUMAN testing in free field

Android applications – Realtime ACE

Change parameters on-the-go



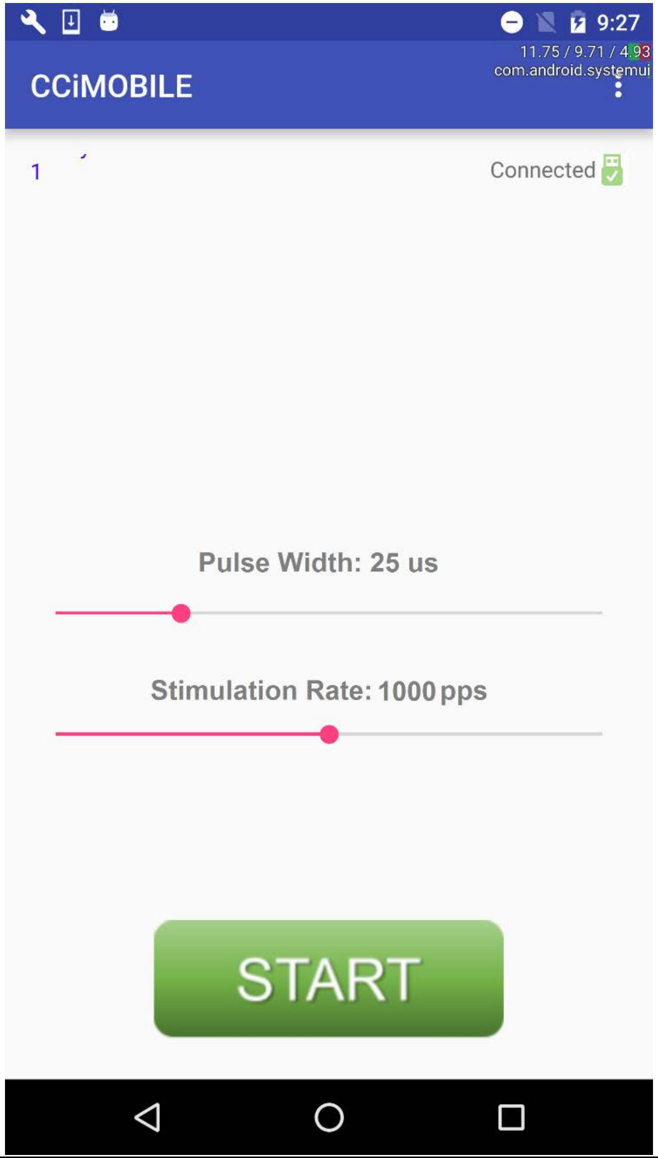


Android applications – Audio Recorder

- For recording the audio signals acquired from BTE microphones
- Could potentially be used for capturing real-world acoustic environments

- Perfect for HUMAN testing in free field

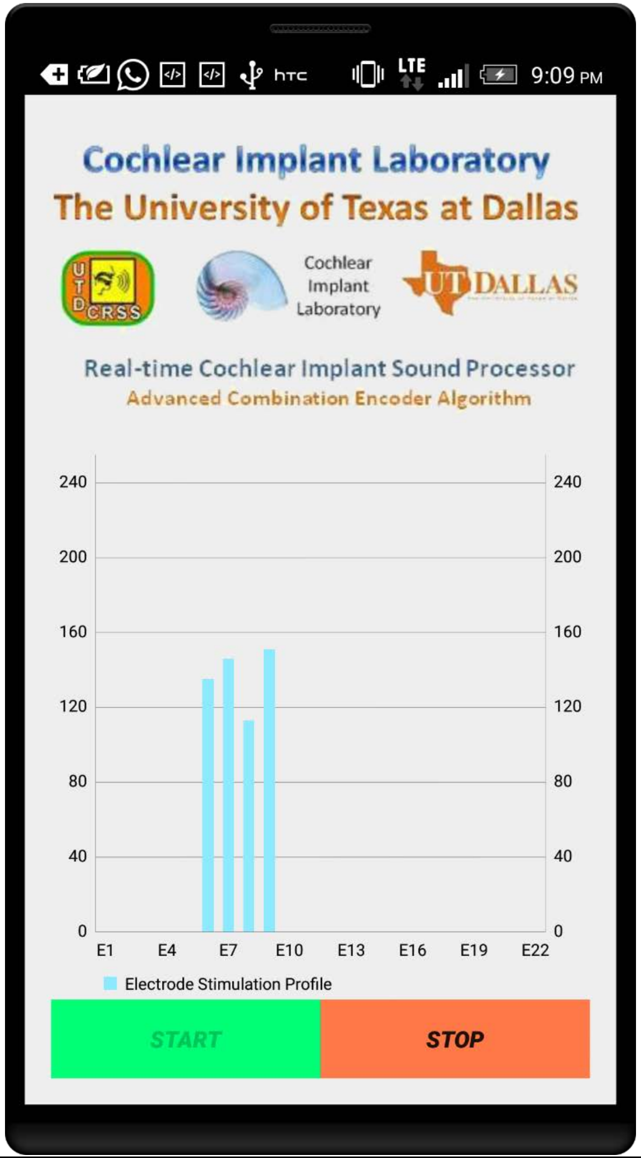
Android applications – DEMO variable rate



- Change stimulation rate and pulse width in real-time
- Pulse sequence updates in realtime - visualize on an oscilloscope
- For DEMO only

DEMO
CCi-MOBILE
Variable Rate

Android applications – RT ACE



- Audio signal acquired from phone's microphone
- Signal processing (ACE) in realtime
- Output pulse visualization
- For DEMO only but can be configured for experimental use
- Available on Google Play Store

CCiMOBILE

Android code - JAVA

```java
while(start) {

    int iavailable_0 = ftDev.getQueueStatus();
    if(iavailable_0>=512) {

        // STEP 1: READ AUDIO SIGNAL
        ftDev.read(readBuffer, 512, 0);
        System.arraycopy(readBuffer, 0, buffbyte2, 0, 512);
        ByteBuffer.wrap(buffbyte2).order(ByteOrder.BIG_ENDIAN).asShortBuffer().get(shortbuffer);

        for (int m = 0; m < leftData.length; m++) {
            leftData[m] = ((double) shortbuffer[2 * m]) * leftScaleFactor;
            rightData[m] = ((double) shortbuffer[(2 * m) + 1]) *rightScaleFactor;
        }

        // STEP 2: Process Audio Signal
        leftStimuli = leftACE.processAudio(leftData);
        rightStimuli = rightACE.processAudio(rightData);

        // STEP 3: Stream Stimuli
        updateOutputBuffer();
        rc = ftDev.write(writeBuffer, 516, true);
    }
}
```

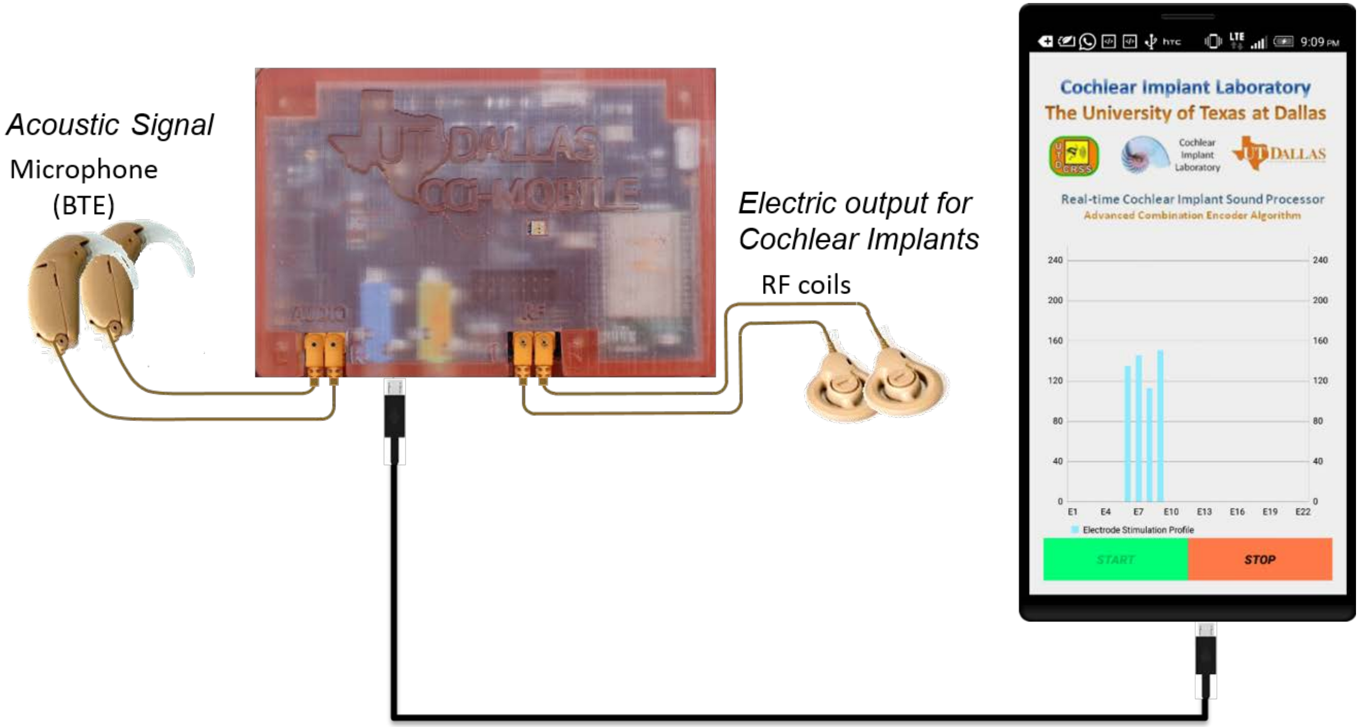
Android code – JAVA – Sound Processing

```java
for(int subframe = 0; subframe < map.pulsesPerFramePerChannel; subframe++)
{
    workingData = new double[BLOCK_SIZE];

    j = 0;
    for(int i = offset; i < BLOCK_SIZE + offset; i++) {
        workingData[j++] = inputBuffer[i]; }

    applyWindow(); // Apply Window

    fft();

    magnitudeSquaredSpectrum();

    weightedSquareSum();

    applyChannelGains();

    sorting();

    loudnessGrowthFunction();

    applyPatientMap();

    stimulationOrder();

    for(int i = 0; i < map.nMaxima; i++) {
        stimuli.Amplitudes[indx] = current_levels[i];
        stimuli.Electrodes[indx] = electrode_numbers[i];
        indx=indx+1;
    }
    offset+= blockShift;
}
```

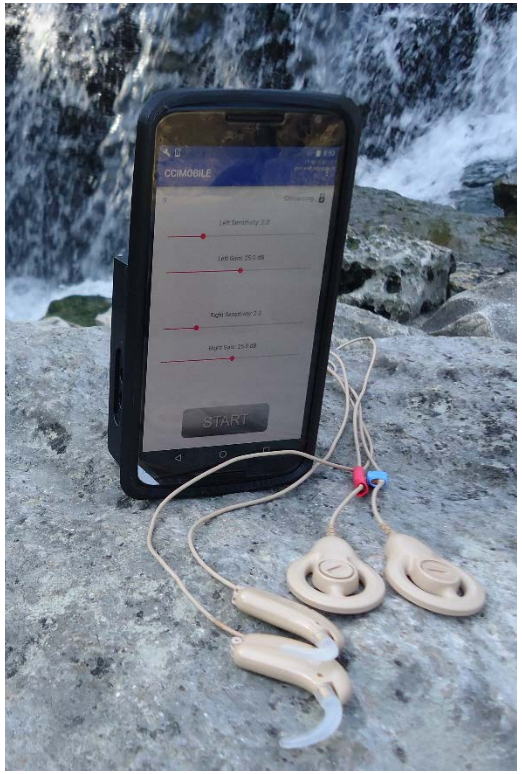
Android specifications

- Plug and play hardware
- Board derives power from the smartphone
- Connect with any commercial off-the-shelf USB OTG cable
- Any commercial Android smartphone can be used, but it has to be rooted, Google-version of Android is recommended.
- For reliable real-time operation, run one application at a time

- Delay: 16ms
- Power Requirements:
- Battery Life: 4 – 5 hours



Portability/Wearability – 3D Sleeve



Safety

- Charge-balanced symmetric pulses only
- Embedded routines for error checking and correcting stimulation parameters
- Pulse-width limited to 400us
- Current levels are checked against the implant type for safe operation.
- Standard stimulation mode
- For sound processing, each pulse's amplitude is checked against subject's MAP and limited in case of erroneous stimulation.

Recommendations for Human testing

- Set a stand-alone PC/phone specifically for conducting human testing.
- Avoid running parallel applications.
- Avoid running extra applications/software on phone.
- Test stimuli on scope before conducting human testing.
- Strongly recommended to have access to implant-in-box emulator
- Test applications are provided to test the integrity of signal and real-time performance.
- Always double check the MAP parameters to ensure you have typed in correct values.

Required Equipment

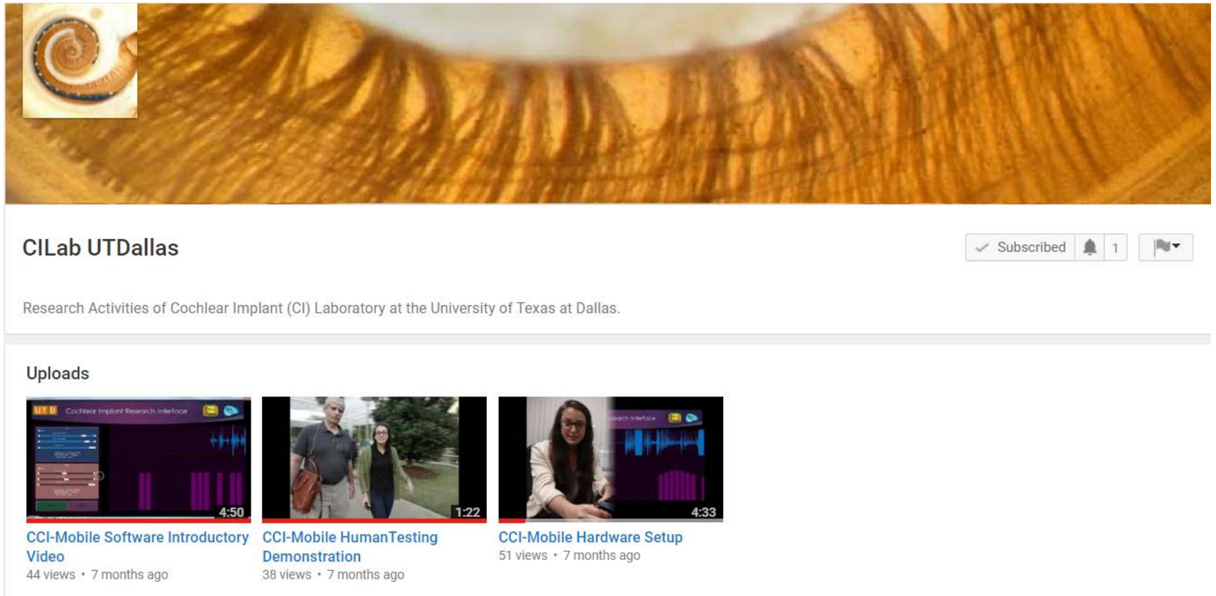
- CCi-MOBILE interface Board
- PC running MATLAB (for benchtop studies)
- Android smartphone (newer recommended with plain Android)
- BTE microphones + RF coils + cables
- USB cables
- Optional: Implant-in-a-box (recommended)

Please fill the Request form and return to us. Reach out to our team on how to get the platform

Resources

- User Manuals and Technical documentation
- Software updates on the website
  - Utdallas.edu/~hussnain.ali/CCIMOBILE.html
  - GitHUB – cilabutdallas

    https://github.com/cilabutdallas/CCiMOBILE.git
- How-to videos
  - Youtube Channel – CILab UTDallas
- Social media
- Annual Workshops (ASA/CIAP)
- Boot-camp at UT-Dallas
- Always feel free to email our team
  - John Hansen: john.Hansen@utdallas.edu
  - Hussnain Ali: hussnain.ali@utdallas.edu



DEMOS

MATLAB (Electric)

Juliana (Juli) Saba Angell

MATLAB (EAS)

Sandeep Ammula

Android + Tech.

Hussnain Ali

Resources

John Hansen

Two DEMO systems available for you to experiment with