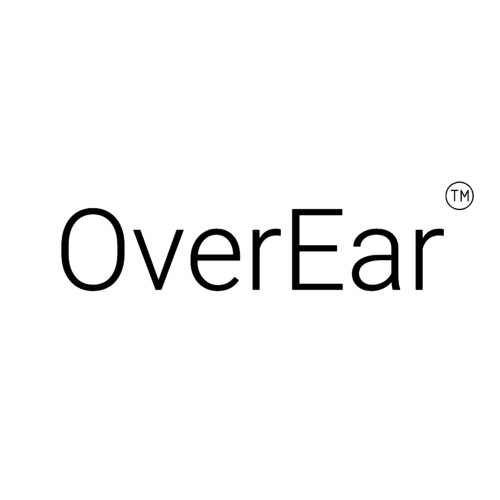
**Boston University**

**Electrical & Computer Engineering**

**EC464 Senior Design Project II**

**Second Prototype Test Report**



**Team Members**

Hannah Gold

Jonathan Ngo

Benjamin Li

Guillermo Ao

**Required Materials**

**Hardware:**

* Teensy v3.5 Development board (microcontroller)
* 2x Electret Microphone Amplifier - MAX4466 with Adjustable Gain
* Audio Adaptor Boards for Teensy v3.5 (Audio Shield)
* Earbuds/Headphones with AUX male port

**Software:**

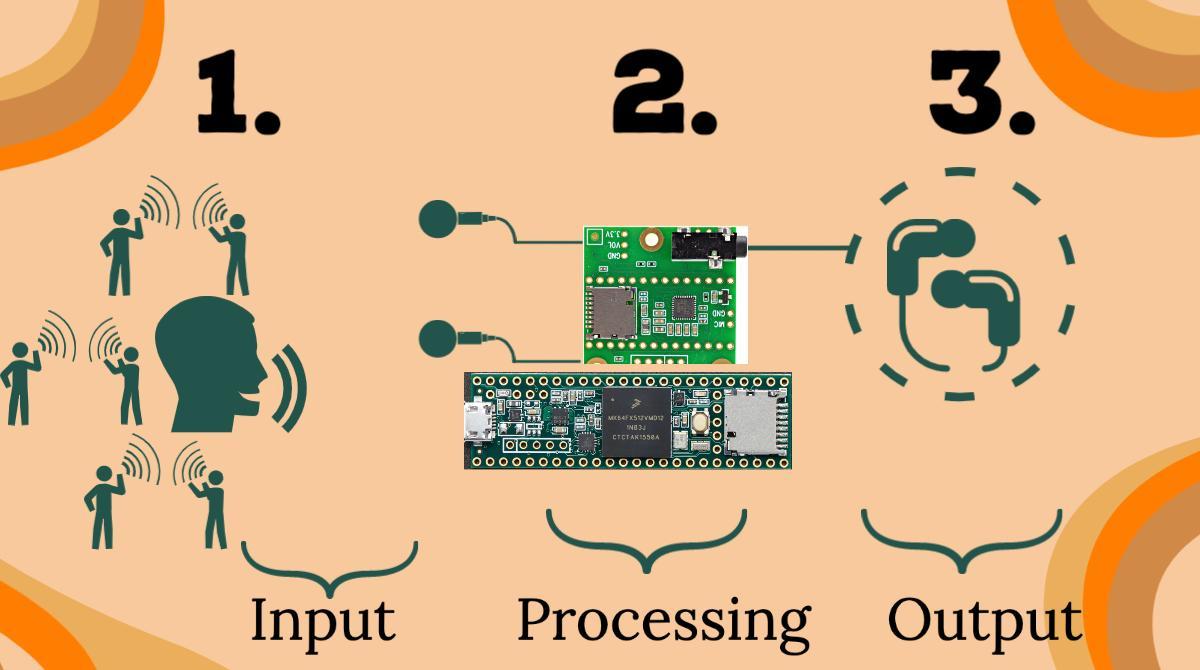
* Arduino IDE + Teensyduino add-on
* Sound-Segregation Algorithm code

**Important equipment description:**

* Teensy v3.5 Microcontroller
  + Used to run the sound-segregation algorithm as well as process the input signals and output them.
* Teensy Audio Shield
  + Hardware interface to simplify sound I/O connections to the Teensy microcontroller.
* MAX4466 Electret Microphone Amplifer
  + Used to record incoming bi-channel audio and provide pre-processing amplification
* Earbuds/Headphones
  + Used to output processed sound to user

**Set Up**

The hardware setup requires attaching the Teensy microcontroller to the audio shield through a series of pins, much like any other Arduino shield. The microphones will be connected to the respective ADC pins on the audio shield to receive the input signals and the earbuds will be connected to the 3.5mm jack on the audio shield for output. The audio shield uses the I2S interface to control the ADC and DAC connections. The software includes the self-developed sound-segregation algorithm which is in charge of processing the incoming noise signals. The code for the algorithm will already be pre-loaded onto the Teensy device. The general test will be to speak to the user in 3 different angles: first in front of the user at 0 degree, then at 90 degree offsets (directly left or directly right). The sound processing is almost immediate and will be outputted through the earbuds as live audio.



*Figure 1: Set up and process flow visual*

**Pre-testing Setup Procedure**

* Push the sound segregation algorithm code to the Teensy microcontroller.
* For the testing, have all the boards connected with the microphones and the earbuds.

**Measurable criteria**

The criteria for successful running and output is as follows:

1. The Teensy should be able to receive input sound signals through both left and right microphones.
2. The incoming signals should be processed through the sound processing algorithm, blocking sounds from directly left and right.
3. The device is able to operate for at least 1 hour on battery.
4. The device fits inside a 120x80x40 mm package.
5. Sound output from forward direction should be significantly greater than from the sides for frequencies within the range of human speech.

**Measurement Methodology**

**Criteria 1-4: General Test**

1. Turn on and set-up the Teensy microcontroller to start receiving input sound signals.
2. Start speaking in front and on the sides of the user.
3. The signals will start to be processed through the algorithm as the device activates.
4. Audio is processed almost immediately and it will automatically reproduce the output sound through the earbud speakers.
5. Listen to the output sounds.
6. Allow the device to run for 1 hour.

**Criteria 5: Quantitative Test:**

**Objective**

Our goal is to quantitatively measure the effectiveness of our device. Since the device outputs speech, it’s difficult to measure coherency of words numerically, and as such, we aim to measure the muffling capability of the device.

**Materials**

* OverEar device
* Phone speaker

**Variables**

* Independent:
  + Frequency
  + Angle of source
* Dependent:
  + Amplitude over time
* Constants:
  + Distance of source from user
  + Voice recording

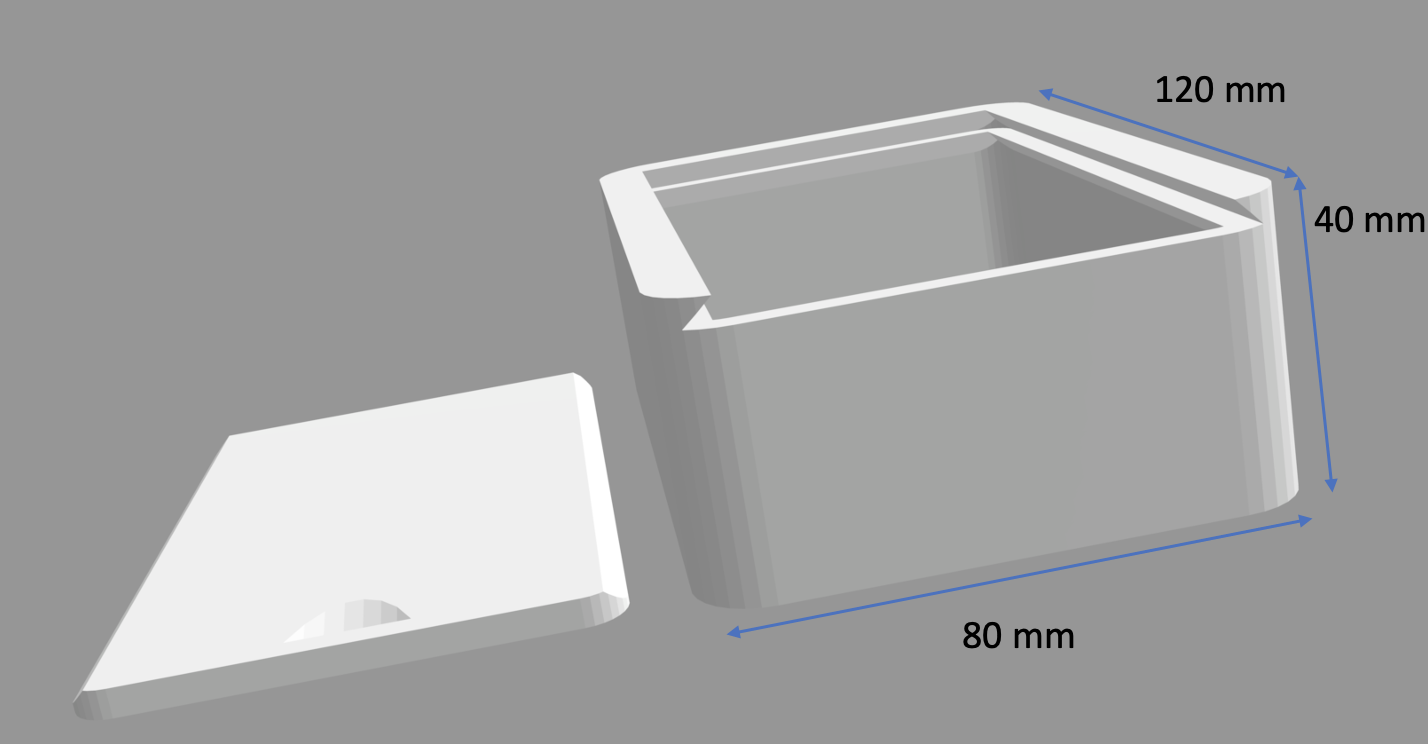
**Procedure**:

1. Connect the device and open the Serial Plotter on the Arduino IDE.
2. Place the phone speaker at the chosen angle relative to the 0th degree azimuth, directly in front of the user, at a distance of 8cm. For simplicity, the speaker will be at the same height as the user.
3. Play the selected sound recording. The sound clips will change across different frequencies and pitches of voice but kept constant across the different angles.
4. Measure and plot the amplitudes of 3 video clips of voices ranging from low to high frequencies at 3 different angles (-90°, 0°, 90°) after they pass through the device, for a total of 9 measurements. Video clips will be played 8cm from the foam head. Note that -90° corresponds to the right of the user, 90° to the left, and 0° right in front, which is assumed to be the direction of a talker.

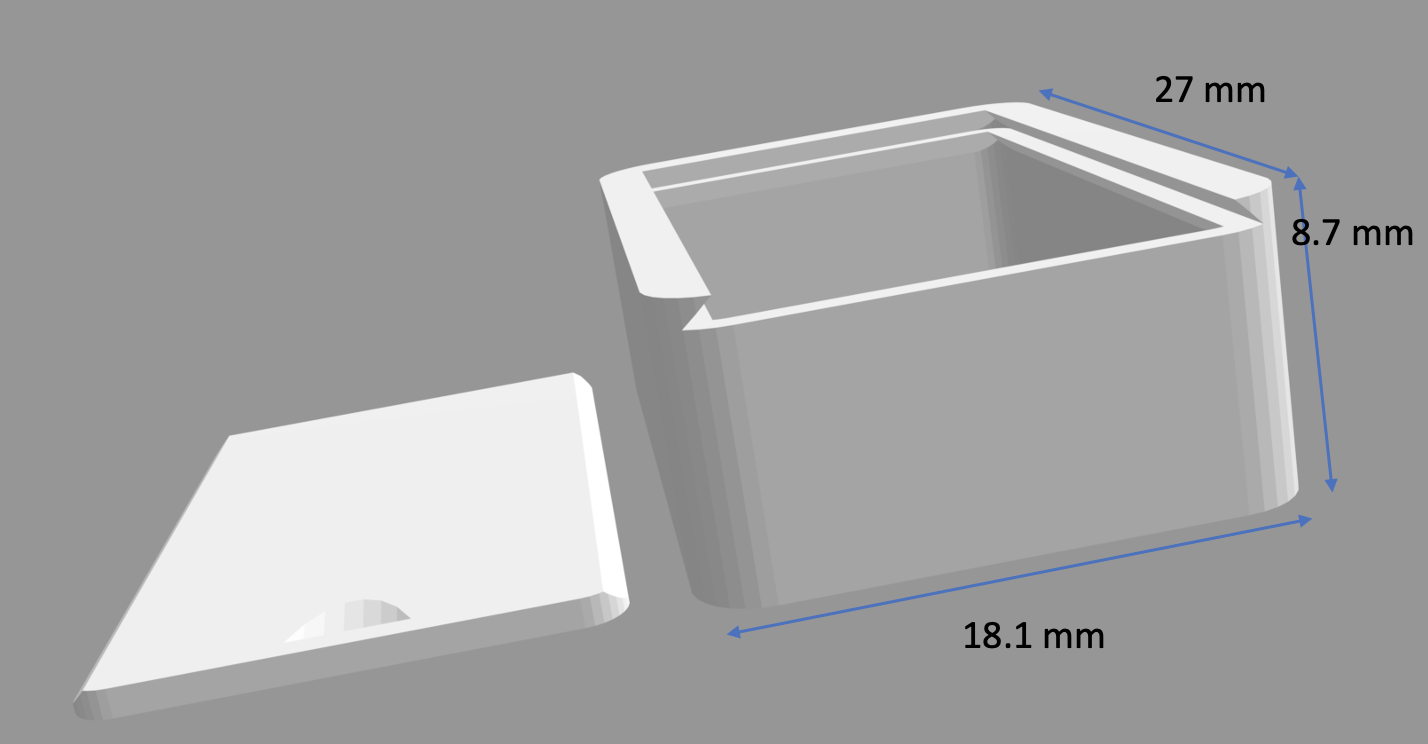
**Results:**

The OverEar device draws 65.5 milliamps during normal operation. This translates to an estimated operating time of just over 15 hours with the 1000 mAh battery. Because the battery outputs a constant 3.7 volts, the power consumption of the device is 242.35 milliwatts.

The main electronics (Teensy + Audio shield + battery) are contained in a 120x80x40 mm case (Figure 2) and each microphone will have its own case of size 27 x 18.1 x 8.7 mm (Figure 3) that has clips to attach earbuds for each side. Additionally, the main case has a clip to attach to the user’s waist band of their pants.



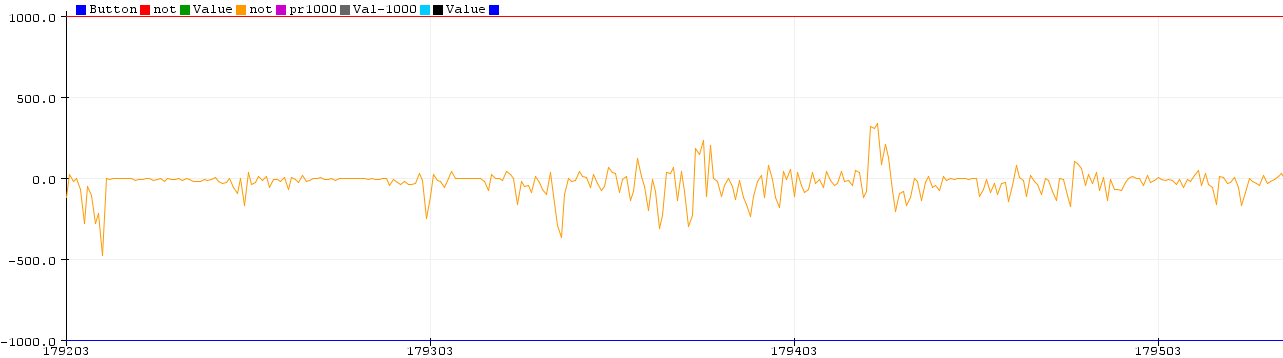
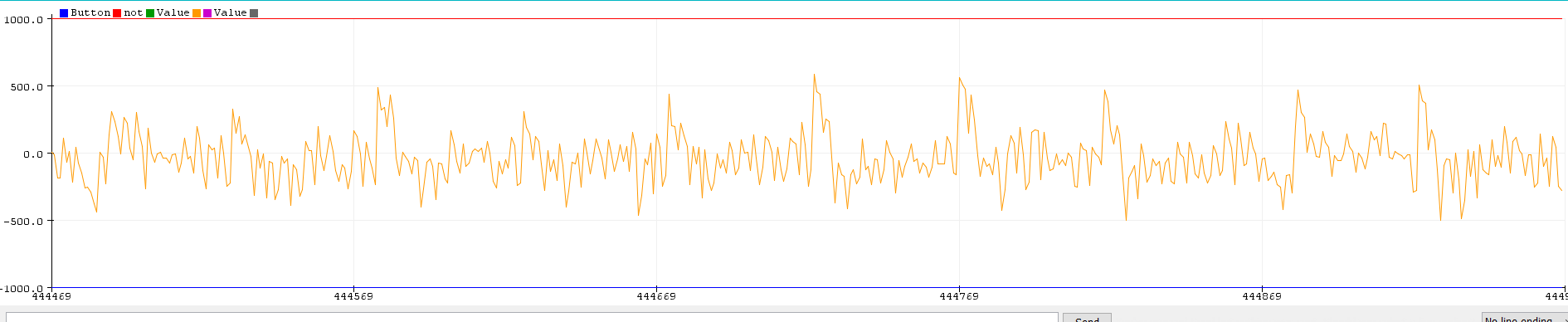
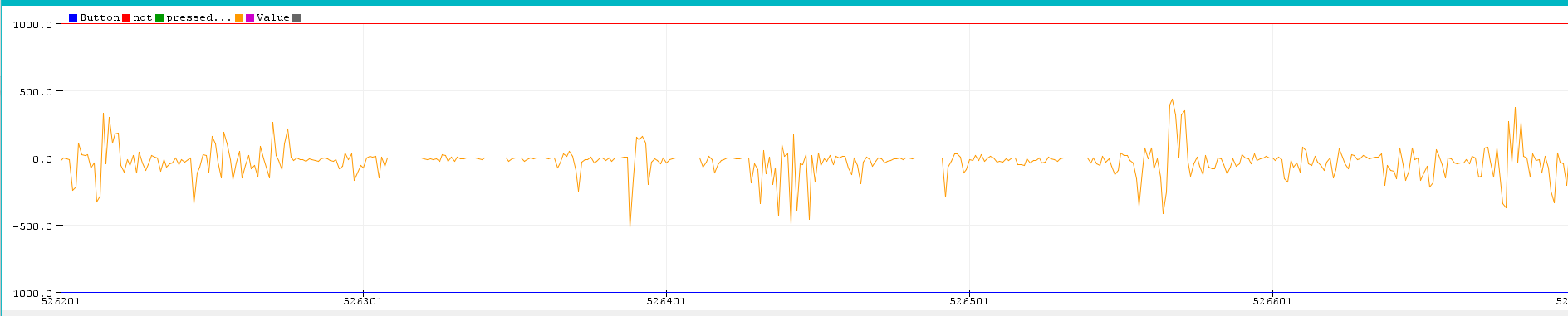
**Figure 2:** Case for OverEar electronic components



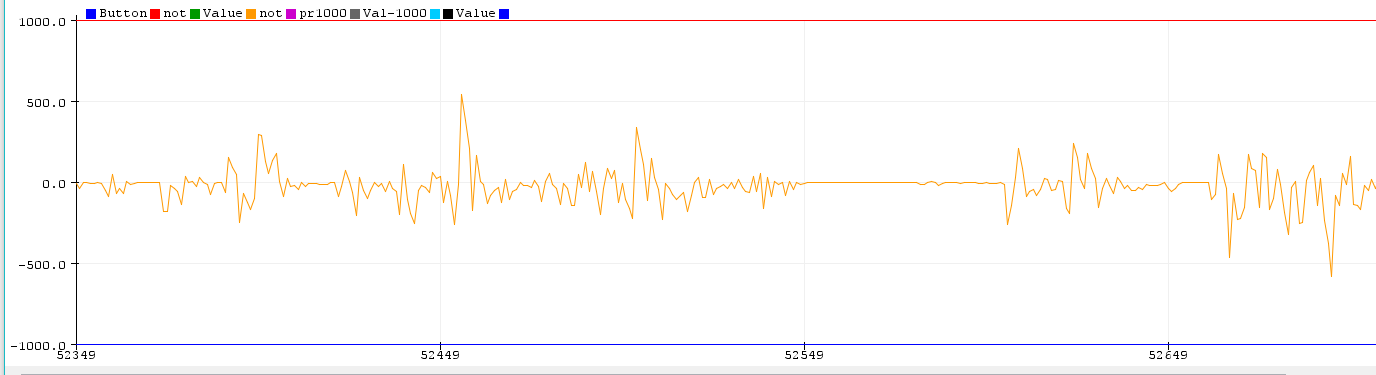
**Figure 3:** Case for microphones

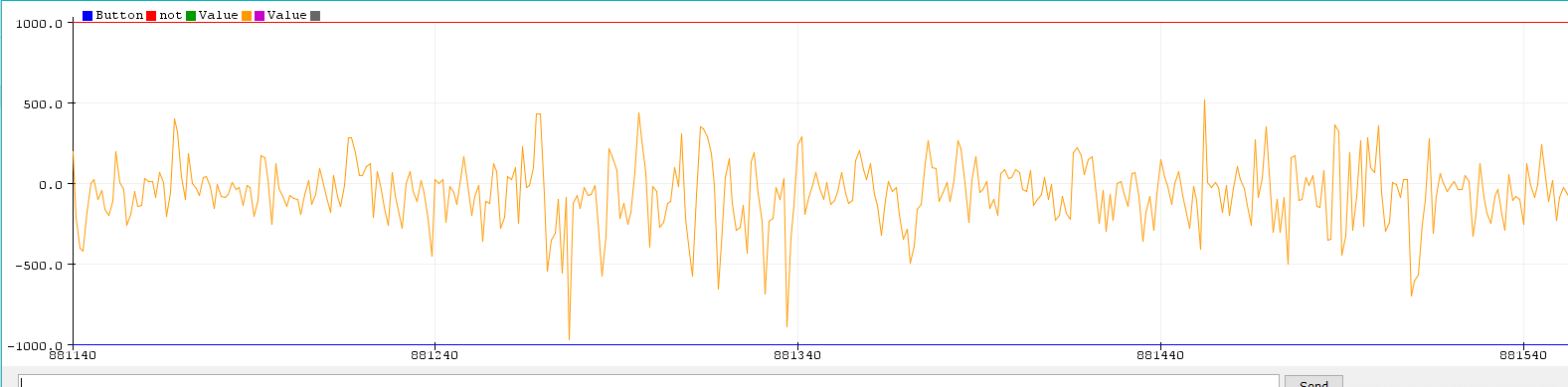
The following graphs were produced using the Arduino Serial Plotter. The values measured in these graphs (orange line) correspond to the output of the device and are the same values passed to the earbuds. Please note that the blue and red lines are constants used to fix the vertical scale of the Serial Plotter.

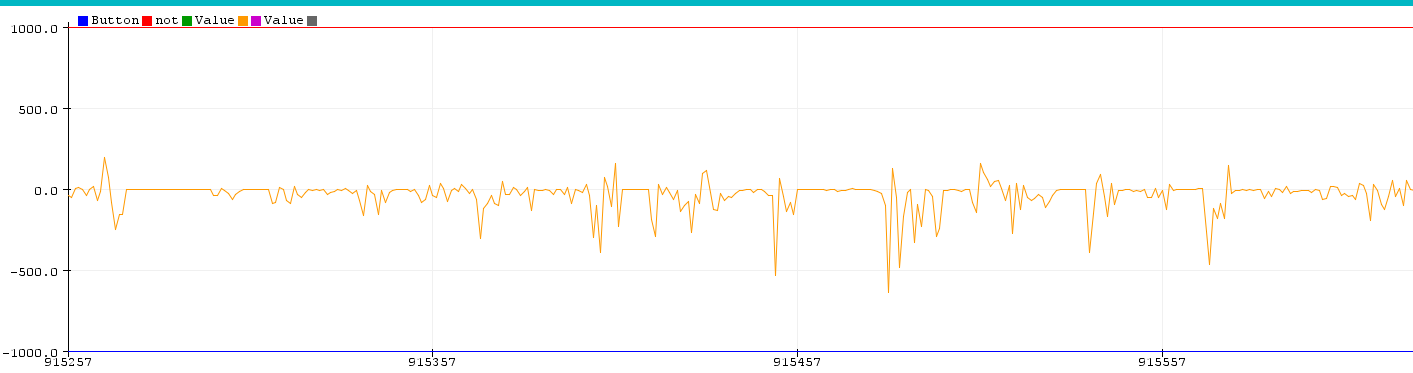
Low Frequency - Speech by Elizabeth Holmes (<https://www.youtube.com/watch?v=5tEeSHy1x98&t=113s> 2:00)  
It can be seen that with Holmes’ speech, there are large periods where almost no noise is seen on the -90° (Figure 4) and 90° (Figure 6) angles. This is due to the sound-segregation algorithm applying a low-frequency filter, which was a deliberate choice to also reduce background noise.

  
**Figure 4.** Measurement of Holmes’ audio amplitude from the -90° angle  
  
**Figure 5.** Measurement of Holmes’ audio amplitude from the 0° angle  
  
**Figure 6.** Measurement of Holmes’ audio amplitude from the 90° angle

Middle Frequency - Interview with Mark Cuban (<https://www.youtube.com/watch?v=vrl5PFB35Ec> 2:42)  
With Cuban’s interview, similar large spaces of small amplitudes are seen in the -90° (Figure 7) and 90° (Figure 9) angles. These can be attributed to Cuban’s voice dipping into lower range frequencies. Otherwise, the amplitudes are dampened by the sound-segregation algorithm.

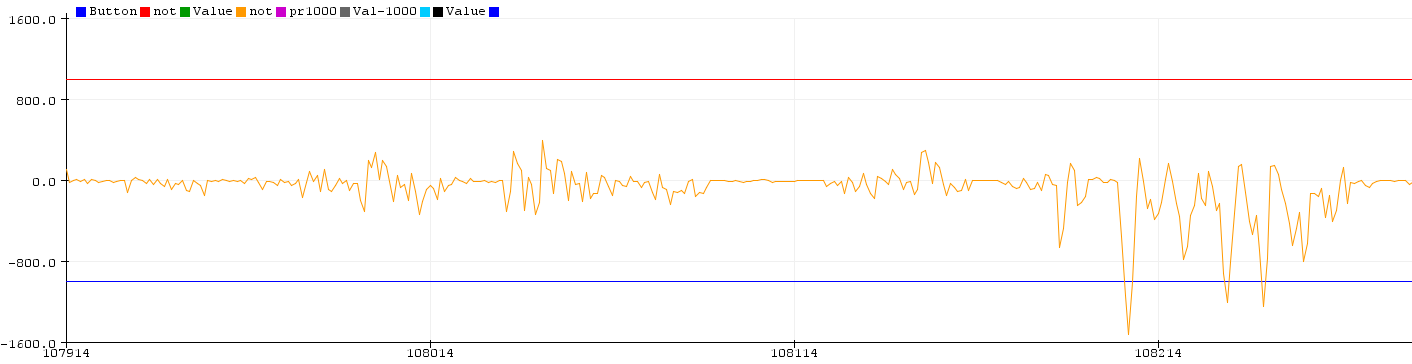
  
**Figure 7.** Measurement of Cuban’s audio amplitude from the -90° angle

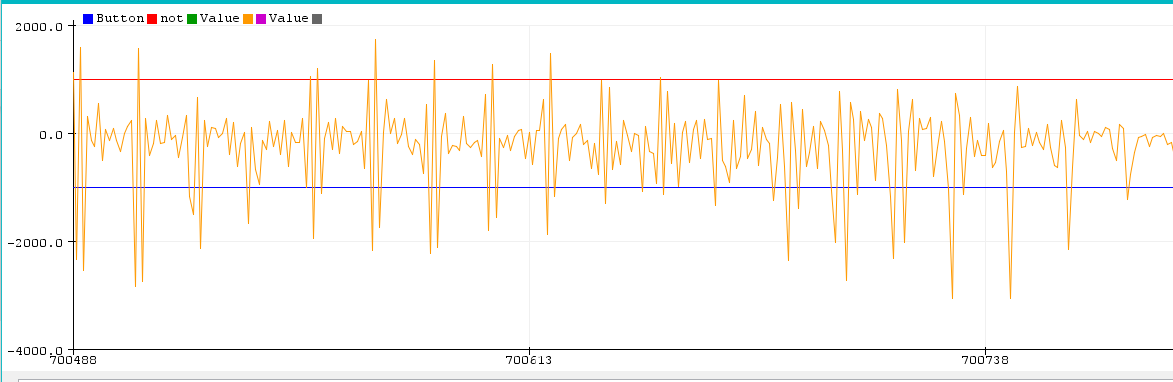
  
**Figure 8.** Measurement of Cuban’s audio amplitude from the 0° angle

  
**Figure 9.** Measurement of Cuban’s audio amplitude from the 90° angle

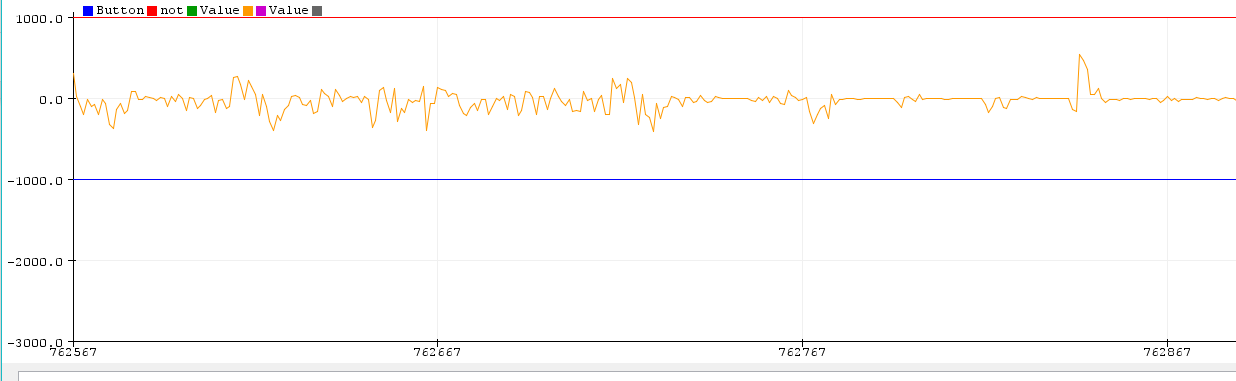
High Frequency - Interview with Khloe Kardashian (<https://www.youtube.com/watch?v=BtftQFxyzIQ> 0:47)

For the graphs of Kardashian’s interview, it can be seen that from the -90° angle (Figure 10), there are large negative spikes. This can be attributed to the blocking performed by the sound-processing algorithm: due to the sound-memory size, blocks of data are retrieved from the microphones by the Teensy, and there can be data drops because of this, similar to dropped internet packets. This will then change the calculated cross-correlation value. It’s not a large issue as these are very short segments in time, and the usability of the device remains unchanged.

  
 **Figure 10.** Measurement of Khloe’s audio amplitude from the -90° angle



**Figure 11.** Measurement of Khloe’s audio amplitude from the 0° angle

  
 **Figure 12.** Measurement of Khloe’s audio amplitude from the 90° angle

In conclusion, we can see that at the -90° and 90° angles, the OverEar device dampens noise while retaining amplitude ranges for the 0° angle. Quantitatively, this shows that our device works as intended. For future testing, we can test more angles and distances to replicate different use cases and determine angle tolerances of our device. Additionally, we can introduce different noise scenarios to observe effectiveness across a range of environments.