Master's Thesis

Auditory Evoked Potentials with Mobile EEG

Marc Vidal De Palol

12.03.2019

Neuroinformatics

Cognitive Science

First supervisor: Johannes Leugering, M. Sc.

Second supervisor: Prof. Dr. rer. nat. Gordon Pipa

Abstract

Electroencephalography (EEG) is a non-invasive technique for studying the electrophysiological dynamics of the brain that can be acquired by measuring electrodes placed on the scalp. In the past, EEG measurements were only done on a lab under exact and limited conditions. Nowadays, thanks to the use of mobile EEG devices, these brain dynamics can be captured and analyzed on natural contexts.

In this present work, an Android App for real-time EEG visualization and recording for working with the Traumschreiber EEG system is developed and tested. Subsequently, an experiment for trying to assess the capability of the device for recording Auditory Evoked Potentials (AEPs) is designed and run.

The resultant App empowers Android devices for being used as EEG visualizers and recorders anywhere at any time, expanding the possible experimental setups to a new level.

The data analysis results highlight the necessity of a more in-depth exploration of waveforms processing techniques and the requirement of a better experimental design to ensure the acquiring and visualization of AEPs. However, the results also indicate that the Traumschreiber system is capable of recording ERPs as a standard EEG system.

List of Abbreviations

ABR	Auditory Brainstem Response
ADC	Analog-to-Digital Converter
AEP	Auditory Evoked Potential
AER	Auditory Evoked Response
ALR	Auditory Late Response
AMLR	Auditory Middle Latency Response
API	Application Programming Interface
BCI	Brain-Computer Interface
BLE	Bluetooth Low Energy
CD	Compact Disc
CSV	Comma-Separated Values
DFT	Discrete Fourier Transform
ECG	Electrocardiography
ECochG	Electrocochleography
EEG	Electroencephalography
EOG	Electrooculography
EMG	Electromyography
ERP	Event-Related Potential
FFT	Fast Fourier Transform
FPS	Frames Per Second
GATT	Generic Attributes
GNU	GNU's Not Unix
HiWi	Wissenschaftliche Hilfskraft
IDE	Integrated Development Environment
IFFT	Inverse Fast Fourier Transform
LED	Light-Emitting Diode
OS	Operative System
PCM	Pulse-Code Modulation
RGB	Red Green Blue
SD	Standard Deviation

SDK	Software Development Kit
SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level
UI	User Interface
USB	Universal Serial Bus
UTC	Coordinated Universal Time
UID	Unique Identifier
UUID	Universally Unique Identifier
WAVE	Waveform Audio File Format

List of Figures

List of Tables

Table of contents

1. Introduction

Understanding the human brain is one of the most significant milestones in science nowadays. For instance, because of its number of neurons, analogous to the number of stars in our galaxy, the organ is often described as one of the most complex structures in the known universe.

To better comprehend how this organ works (organization, functions, and relations), many techniques have been developed and are being used at present. One of these methodologies is EEG, a non-invasive method for measuring the activity of the brain on the scalp. Traditionally EEG recordings were exclusively done on a lab. Thus, with exact conditions and in a closed environment with limited possibilities. However, as electronic components such as mobile technologies are continuously getting cheaper, more powerful and smaller (Appel, 2018), also mobile EEG devices did. Thanks to these last, the possible experimental setups increased exponentially, and brain activity can now be recorded and analyzed in natural contexts.

Mobile EEG devices are commonly used in the field of BCIs, aiming to distinguish specific patterns of electrical brain activity and use them as control commands, studying sleep cycles, facial expressions, and performance metrics such as excitement, engagement, interest, focus, stress, and relaxation. Some examples of these devices are the Emotiv Epoc+ (EMOTIV, 2015), the Mindwave Mobile 2 (NeuroSky, 2015) or the Smarting (MBT, 2013).

1.1. Electroencephalography

EEG signals mirror the dynamics of electrical activity in populations of neurons. These populations can work in synchrony since they are connected forming a network. Therefore, EEG signals are a readout of voltage oscillations when these groups of neural cells are working together (Niedermeyer & Lopes Da Silva, 2005). Different sets of neurons interacting, rather by mutual excitation or inhibition feedback, can result in the oscillatory phenomena shown in EEG signals. The dynamic behavior of the population is then reflected in the subsequent waveform (Niedermeyer & Lopes Da Silva, 2005).

EEG systems measure the electrical activity present on the scalp by the use of electrodes. The recorded electrical activity is essentially the summation of excitatory and inhibitory postsynaptic potentials at the dendrites (Cohen, 2014).

Configurations of electrodes (reading input) to channels (calculated output) used in EEG, commonly known as montages, can be divided into two main categories. A possible setup is that all electrodes placed on the scalp are referenced to one single electrode. This arrangement implies the resultant voltage at each channel is always the difference recorded between one electrode and the reference one. The other montage setting is the scalp-to-scalp bipolar linkage. In this way, the resulting amplitude value of each channel signifies the difference registered within each pair of linked electrodes.

EEG is primarily a useful technique when it comes to the study of neurocognitive processes in high-temporal resolution. On the one hand, because it can capture these dynamics within the timespan they occur. Most of cognitive, emotional, linguistic, motor and perceptual processes chance within hundreds of milliseconds. On the other, because as the voltage fluctuations measured are biophysical phenomena at the level of populations of neurons. Thus the observed oscillations are reflections of neural fluctuations in the cortex. Hence, what is measured is the direct neural activity (Cohen, 2014).

1.2. Auditory Evoked Potentials and Event-Related Potentials

EEG as a methodology for studying the functioning of the human brain can also be used for comprehending the response of the brain to different stimuli. When the EEG changes are related to particular events, these are called ERPs. These variations are often studied when they are sensory evoked. Hence, they are auditory, somatosensory or visually induced (Niedermeyer & Lopes Da Silva, 2005).

Although ERPs are time-locked events categorized depending on the sensory type of the eliciting stimuli, they are often described, decomposed, and analyzed instead in terms of their temporal relation or region of origin. It is also the case of the AEPs, as the ERPs of interest for this present work, being essentially brain waves, sometimes denoted as electrical or evoked potentials, formed by sound stimulation on a person.

The activity produced by sound stimulation comes from structures within the ear, nerve, and brain (W. Hall III, 2015).

Looking at the time-based relation and region of origin of AEPs categories (see Figure 1), first we encounter the ECochGs, as the earliest potentials generated on the cochlea, followed by the ABRs, generated on the brainstem. The former occurs within 2 or 3ms after the stimulus onset, whereas the latter appear within a period of 6 to 10ms.

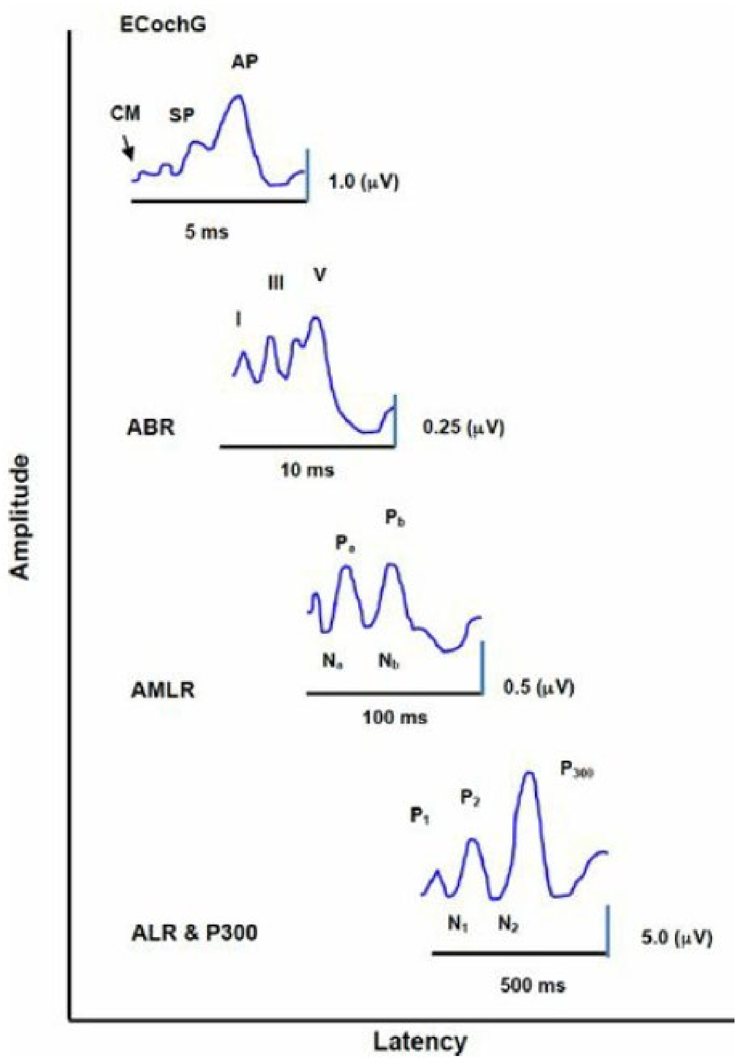


Figure 1. Representation of clinically-recorded AEPs subtypes (James W. Hall III, 2015).

In the case of the long-latency subtypes, on the one hand, we find the AMLRs, arising within the first 100ms. On the other, we have the ALRs, occurring during the first 500ms. As it is also denoted in Figure 1, they are comprised of different components with dissimilar amplitudes. However, although they are decomposed in Figure 1, they should be seen as a continuous phenomenon as it is shown in Figure 2 (Picton, Hillyard, Krausz, & Galambos, 1974). The P components indicate a positive peak whereas the Ns indicate a negative one. Note that they are named in order of appearance but also designating the latency of their occurrence, as it is the case of the P300 (P3) component of ALRs.

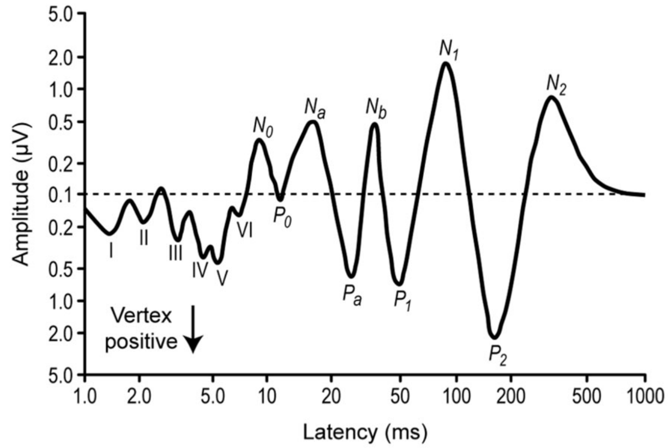


Figure 2. Diagrammatic representation of the auditory evoked potential components.

The P300 or P3 is an ALR component with positive charge amplitude often observed with 300ms of latency after the stimulus presentation (Ilardi, Atchley, Enloe, Kwasny, & Garratt, 2007). It is named 300 because of the appearance time in milliseconds after the stimulus onset. Also known as P3 due to be the third wave component after the positive waves P1 and P2 of the ALRs components. However, it is acknowledged that P300 can also be recorded between 250ms and 400ms in healthy subjects (W. Hall III, 2015). To better illustrate and summarize the characteristics of each AEP type Table 1 is presented.

Table 1. AEPs components properties summarized.

	ECochG	**ABR**	**AMLR**	**ALR**
Region of origin	Cochlea	Brainstem	Auditory cortex	Auditory cortex
Timespan after stimulus onset	2-3ms	6-10ms	100ms	500ms
Components	SM, SP, AP	I, III, V	Na, Pa, Nb, Pb	P1, N1, P2, N2, P300

As a method for assessing the status of the central and peripheral auditory system, in comparison to other ERPs where a behavioral subject response is required, AEPs measurements reflect mainly electrophysiological responses. Thus, no behavioral task is needed to take appropriate measurements of AEPs except the P300 (P3). In contrast, this response is known to be cognitively evoked in attention-driven listening tasks for the subject (W. Hall III, 2015).

There are no standardized protocols for AEPs measurements, neither generally accepted principles for defining or analyzing responses. Thus, no single technique is considered the best approach when it comes to recording AEPs. The best method should always be the one capturing the most accurate, reliable and well-formed response (W. Hall III, 2015). However, it is known that at least one electrode array is necessary to record an AEP, understanding it as a combination of two recording electrodes (W. Hall III, 2015).

One of the main difficulties of detecting ERPs is the habitually much larger ongoing background activity, usually not associated with the stimulus presented. The electrical activity not related to the stimulus presented is referred to as noise. In order to be able to distinguish the ERP among the background noise, the SNR must be improved. Several techniques are used to achieve this, for example, signal averaging. This method is used under the assumption that the pattern of auditory brain activity, such as the P and N components of the AEPs should be considerably similar. Thus, the voltage amplitude at the latency elicited after each same stimulus presentation should be almost the same. In this way, by averaging the epochs of the signal recorded after each appearance, the background noise should cancel out or be drastically reduced (as random waveform patterns) whereas the ERP signal should be strengthened (W. Hall III, 2015).

1.3. *The Traumschreiber*

The Traumschreiber is a mobile EEG device designed by the Ph.D. students Johannes Leugering and Kristoffer Appel from the University of Osnabrück. Initially, the device was meant to be a high-tech sleep mask for research purposes. Especially, for measuring and manipulating human sleep in sleep experiments (Appel & Leugering, 2017). However, the device features include eight channels that are usable not only for EOG but also for EMG, ECG, and EEG.

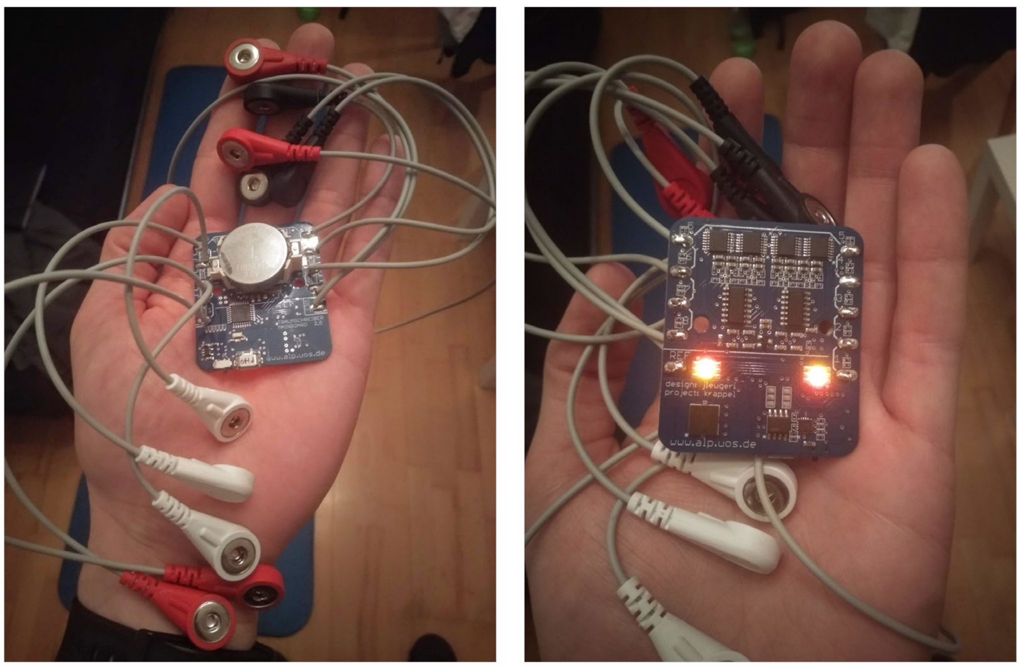


Figure 3. The Traumschreiber device seen from the upper side (left) and the lower (right).

This device is precisely the one used in connectivity with the developed Android App and also for the recordings of the experiment of this present work. Due to its BLE capabilities, it can send the data measured from its electrodes via Bluetooth. Thanks to this feature, the acquisition, presentation and storing of data are possible in real-time on the App.

1.3.1. Device Components and Characteristics

The device consists of a 55 mm x 47 mm x 1 mm motherboard (smaller than a debit or credit card), with a mini-USB power charging port, a power on/off switch, a

LIR2477 3.6V button cell battery slot, two RGB LEDs indicating the status of the device (on, connected, paired, charging, charged) and, ten lead female electrode wires soldered on the motherboard.

Table 2. Channel-electrodes distribution of the Traumschreiber.

Channel	1	2	3	4	5	6	7	8	Ground
Electrodes	1 - 2	2 - 3	3 - 4	4 - 5L	5L - 5R	5R - 6	6 - 7	7 - 8	9

The first nine numbered electrodes measure the electric potential (voltage) wherever placed. The micro-controller of the device computes the difference of the electric potential in pairs of electrodes and returns the octet of values representing 8 EEG measured channels. More precisely (see Table 2), it calculates the difference between the electrodes 1 and 2 (channel 1), 2 and 3 (channel 2), 3 and 4 (channel 3), 4 and 5 left (channel 4), 5 left and 5 right (channel 5), 5 right and 6 (channel 6), 6 and 7 (channel 7) and, 7 and 8 (channel 8). The remaining electrode, tagged as REF on the motherboard, is the non-measuring ground electrode, aimed to prevent power line noise from interfering with the measured small biopotential signals of interest (Biopac Systems, Inc., 2014).

For working with it, the device requires a charged LIR2477 3.6V button cell battery placed on its slot and a hydro-gel foam electrode pad clipped with each lead female electrode wire aimed to use for measuring.

Figure 4. Ten hydro-gel foam electrode pads with a LIR2477 3.6V button cell.

1.3.2. Device Specifications

The Traumschreiber specifications describe a resolution of 250 Hz, meaning it can capture 250 consecutive octets of values in a second. It can be considered an excellent measure since standard digital EEG systems are known to have a sample rate of 240 Hz (ACNS, 2006). It is recommended a minimum sampling rate three times the high-frequency filter (72 Hz in this case). Thus, the sampling rate of the device is above the minimum recommended according to the American Clinical Neurophysiology Society.

In practice, during this work, it has been estimated that the device tops at 223Hz. Hence, it measures and sends a new octet of values each 4.5ms on average. It is important to note that this estimation has been calculated considering the mean time between the acquisition of the data as a sampling frequency on the App developed. Examining the real data point octets obtained time, this ranges between 1ms and 10ms. This time window is explained because of the Bluetooth delay while sending data and the data buffering on the device.

A 12bit resolution ADC and two active analog band-pass filters (0.04Hz high pass, 72 Hz low pass) are incorporated in the device. The former, as the name indicates, is used for converting the analog signal, in this case, the measured voltage, into a digital number. As the resolution of it is 12bit, it means the total voltage range represents 2^{12} = 4096 possible values. Thus, as EEG signals are waveforms, it actually can represent values ranging from -2048 to 2047. Analog filters of 0.04Hz as high-pass and 72Hz as low-pass are included into the hardware of the device since most of the frequencies of interest on EEG measurements are between 0 and 100Hz. Hence, these analog filters establish the bandwidth of the data that is going to be visualized or recorded. In case of the above filter, 72Hz implies getting a less steep roll-off of the frequency response, on the one hand, but also to make sure that the signal is completely attenuated when it reaches 100Hz.

Another relevant spec to mention is the amplification range. The device contains a gain amplification factor of 1000, which makes it possible to measure from microvolts up to millivolts (factor of 1000). Moreover, there is also a modifiable extra gain parameter adjustable via Bluetooth. This one can be set to 0.5, 1, 2, 4, 8, 16, 32 or 64.

The primary function of it is to change the voltage amplitude limit to record or analyze. In other words, a higher gain crops the signal, like a zoom in effect while a lower one produces a zoom out of the outcome. Thus, if we set a high gain parameter value and we want to measure an amount more prominent than the limit, the result will be a clipped signal, preventing us from seeing the actual voltage but toping constantly.

Finally, it is also important to highlight the power autonomy of the device since it can last up to 8 working hours, and the production price as it is under 100€. Thus, in comparison with a standard EEG recording device that can rise from 1000€ up to 25.000€, it is, in any case, a low budget option.

Table 3. Specifications summary of the Traumschreiber device.

EEG Channels	8
Electrodes	10
ADC resolution	12 bits
Sampling rate	250Hz (223Hz measured)
Amplification factor	1000 device set (from µV to V) from 0.5 to 64 modifiable via Bluetooth
Active Analog Filters	High Pass: 0.04Hz – Low Pass: 72Hz
Connectivity	Bluetooth Low Energy (BLE)
Power use	LIR2477 3.6V 120mAh rechargeable button cells
Autonomy	8 hours
Production cost	Under 100€

1.4. The present work

The present work contains two main related goals. On the one hand, to develop a functional Android App for working with the Traumschreiber system, and thus make the mobile EEG possible and usable for research purposes. Hence means, building an app able to connect to the device, and also receive, represent and store real-time measured EEG data. On the other hand, use the App to try to assess if the device can record AEPs as a specific type of ERPs and see what components of the former are observable.

The motivation behind this study started back on January 2018, when Prof. Dr. Gordon Pipa gave me the opportunity to join a group of enthusiasts with the project idea of developing an epilepsy diagnosis Android app working with the Traumschreiber device. The main intention of it would be to help epilepsy-affected people in underdeveloped countries where access to the doctor is often limited due to either geographical or economic reasons.

The starter of the project was Jan Zerfowski implementing a prototype of it with a working Bluetooth connection between the device and the app (Zerfowski, 2018), derived from the *Android BluetoothLeGatt Sample* project (Google Samples, 2018). After that, the project was intensively developed during the Hack4health hackathon between the 12th and the 16th of February of the same year (IKW, RKI & GCO, 2018) by Adrián Rojas, Andrei Achziger, Renato Garita and me (Rojas, Achziger, Garita, & Vidal De Palol, 2018). At this point, the app was able to display some data of a specific channel, including an educational chatbot for epilepsy and EEG. Afterward, Kai Fritsch added a time/frequency analysis into the project, and Adrián Rojas and I continued the development as the task of our HiWi job positions at the Institute of Cognitive Science at the University of Osnabrück (Rojas, et al., 2018).

Because of being involved in this project, I came up with the idea of developing a simple but functional app for displaying and storing EEG data measured with the Traumschreiber. On the one hand, because the EEGdroid project included more than just these two simple tasks and was not only intended for researchers but end-users. On the other, due to the fact the only way to successfully visualize and record data with the device at that point were a set of Python (Python Sofware Foundation, 2019) scripts developed by Johannes Leugering and Pascal Nieters for the course *Edge Computing on Intel Movidius Neural Compute Stick and Mobile EEG* (Nieters & Leugering, 2017). With the constraint that these scripts were only meant to work on GNU/Linux-based OSs. Moreover, the Bluetooth connectivity library used is not as reliable as the Android one, as the Android community updates this last, and moreover many Bluetooth devices are meant to work with Android devices.

Additionally, since the start of the present work I had no background on EEG, it was a huge motivation to work precisely on this. To get into detail about how to make recordings, clean, process and analyze EEG data. AEPs were the type of ERPs chosen because they can in principle be easily elicited, just by presenting audio stimuli to the subject. Moreover, due to my background in music education, this was an extra incentive to use real musical tones as experimental stimuli to try to provoke this AEPs. Also, because they are not the usual stimuli used in when it comes to trying to see and analyze this type of responses.

2. Method

The methodology used for the present work is commented into the next two subsections. First, the development process of the app is described. Second, the experiment design and post data analysis procedure are explained.

2.1. App development

2.1.1. Development platform

Android was the development platform chosen for this work, preferred for different reasons. On the one hand, because it is nowadays the most used OS worldwide, also with an increasing market share tendency (StatCounter, 2019). The platform is leading the market share in both mobiles (74.45% in January 2019) and general OSs (37.55% in January 2019) (StatCounter, 2019). Moreover, this platform supports a very diverse range of devices. From very non-expensive mobile devices starting around 50€, up to very high-end devices of about 1000€. Thus, contributing to the aim of making mobile EEG more affordable and so extensible to researchers and enthusiasts. On the other hand, because this app development work deviated from the EEGdroid project, and a thus substantial part of it was reused and modified for the present.

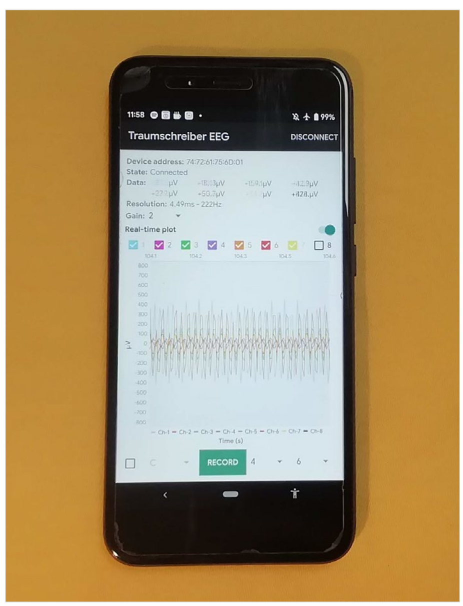


Figure 5. Xiaomi Mi A1 smartphone used for this work.

For the development of the app, Android Studio was the IDE used, as it is the officially supported software for Android development (Google Developers, 2019). When using Android Studio, there are mainly three supported programming languages. These are Java, Kotlin and C++. Java was picked since it is one of the most popular worldwide, and hence there is a big community and lots of support to access on the net (Stackoverflow, 2019). To target a big range of Android devices while making use of current technologies, a compromise for choosing the minimum compatible API for the app project was made. Concretely, the API 21. As statistical data reported from Google on October 26, 2018, this would target the 88.9% of the devices worldwide (Google Developers, 2018). However, as the app was actually debugged and tested mostly on an API 28 (Android 9 or Pie) running device, the SDK target version for compiling the project was set to 28.

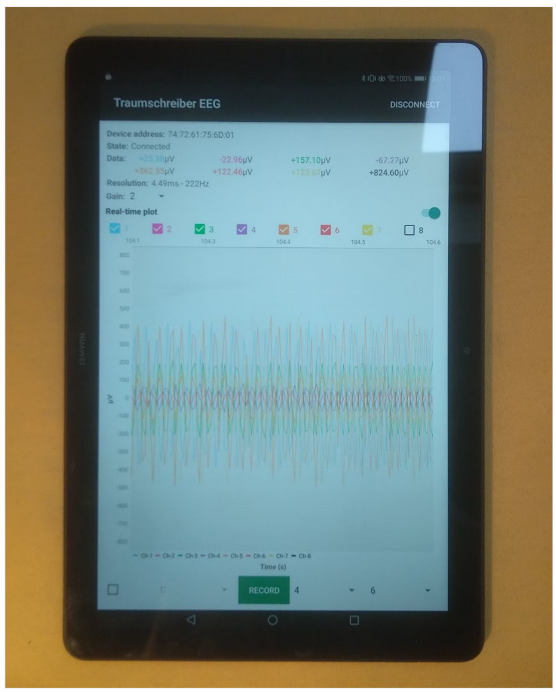


Figure 6. Huawei Mediapad T5 tablet used for the development of the app.

In the end, the App was tested on two different devices. On the one hand, a Xiaomi Mi A1 smartphone (Xiaomi, 2017) with API 28. On the other, a Huawei Mediapad T5 tablet (Huawei Technologies Co., 2018) with API 26. Thus, two well-performant mid-range Android devices (Passmark Software, 2019) released between

September 2017 and September 2018 were the ones used. Mid-range points smartphones or tablets devices with specifications that currently lay somewhere between the high-end and the low-end systems, which means the former category as the current top-performant and the latter the low-performant available models (Chibueze, 2016).

2.1.2. Code definition and structure

As mentioned, the coding process of this app did not start from scratch but forked from the EEGdroid project of the Neuroinformatics research group of the Institute of Cognitive Science of the University of Osnabrück. Precisely, the latest commit of the code of the plotting and recording module with the Bluetooth connection from the 8th of August (Rojas, et al., 2018) was the initial point when the development process started in late September 2018.

When developing with Java on Android, a set of Java files are written. All of them usually stored on the path *src/main/java* of the project. In this case, the project files used are listed in Figure 7.

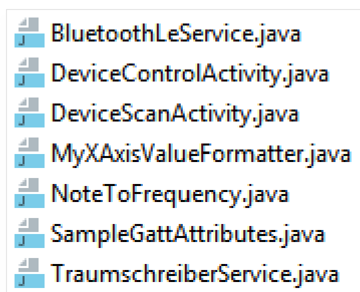


Figure 7. Java file classes defined in the Android project of the App.

Each of these java files represents a different class of the code. The ones named as Activity, as the name indicates, are Android Activities. An Android Activity class is a crucial component of an Android app, and the way activities are launched and put together is a fundamental part of the platform's application model. Unlike programming paradigms in which apps are launched with a *main()* method, the Android system initiates code in an Activity instance by invoking specific callback methods that correspond to particular stages of its lifecycle (Google Developers, 2019).

Thus, specifically for this app, two activities were defined. First, *DeviceScanActivity.java* as the first screen of the app aimed to scan and list BLE devices, precisely only Traumschreibers. Second, *DeviceControlActivity.java* as the next and control screen, for interacting with the Traumschreiber by showing information about the device, reading and plotting the data sent from it, and connecting and disconnecting with the device.

The rest of the java file classes have very distinct functions. *BluetoothLeService.java* is where all methods, functions, and properties for managing the Bluetooth connection, including sending and receiving data between the Android and the BLE device. *MyXAxisValueFormatter.java* is just a class for formatting the X-Axis of the chart in the desired decimal format. *NoteToFrequency.java* is a music key to musical tone frequency converter class. *SampleGattAttributes.java* is a sample class where BLE GATT characteristics are defined as a UID. Finally, *TraumschreiberService.java* is where specific GATT characteristics and other device-specific properties for the Traumschreiber are declared, including a byte array to integer array conversion method.

Concerning the UI of the app, in Android development it can be either designed using its layout editor or in XML files. This former allows the developer to quickly build layouts dragging UI elements into a visual design editor instead of defining it into XML files by hand (Google Developers, 2019). It is precisely on XML files where all the resources of an Android app project are in the end declared. So, not only the UI for the Activities but also of menus, colors, text strings presented on the interface, icons, images, videos, media, etc. The folders and files used for this work are shown in Figure 8.

In more detail, inside the *res* (resources) folder there are three folders: *layout*, *menu,* and *values*. As these names indicate, the designs for all Activities are there. Precisely, *listitem_device.xml* contains the layout definition for the Activity *DeviceScanActivity*, whereas *gatt_services_characteristics.xml* covers the *DeviceControlActivity* one.

The exception here is *actionbar_indeterminate_progress.xml*, which defines an indeterminate circle progress bar element for the menu when the app is looking for BLE devices. On the *menu* folder, *gatt_services.xml* declares the menu bar elements when the Activity is *DeviceControlActivity* whereas *main.xml* when DeviceScanActivity is the one running. For the *values* folder, as its name and inner filenames themselves indicate, the colors and text strings values used on the UI are specified as resources.

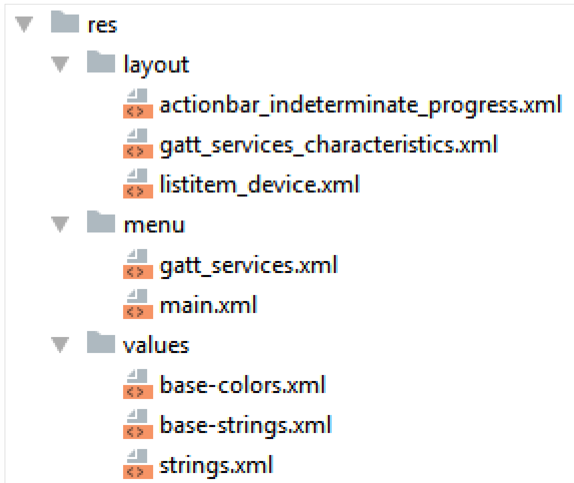


Figure 8. Folder structure and files used for the development of this app.

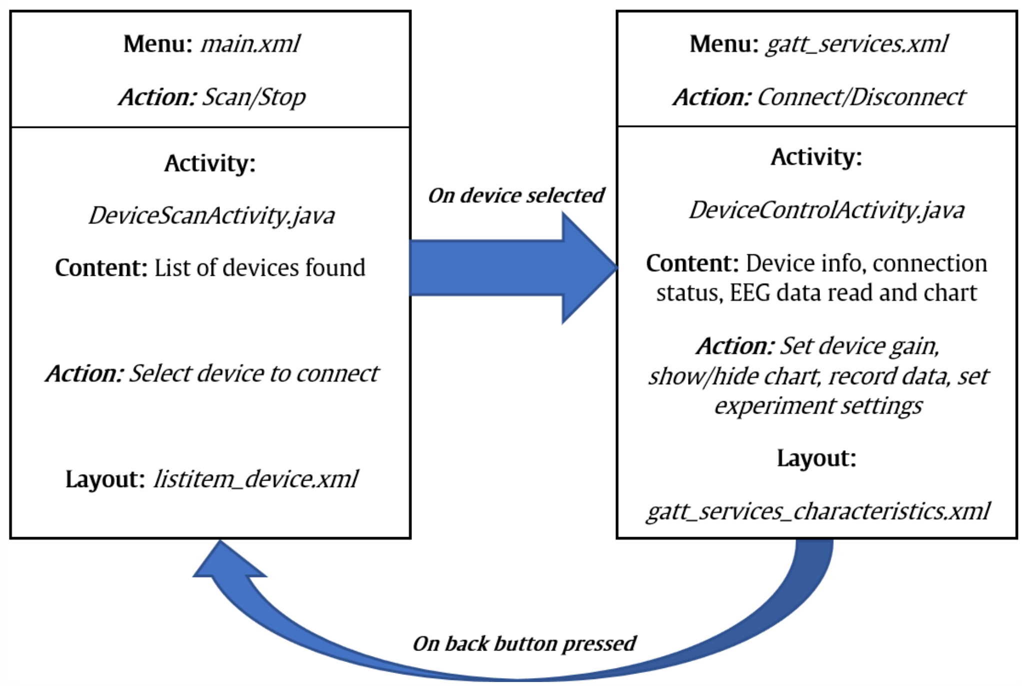


Figure 9. Flow chart representation of the app.

To better illustrate how the Activities, the menus, the UI and other resources are used in the app, the flow chart representation in Figure 9 is presented. There, additionally, the actions for each Activity plus the content shown are specified. Moreover, the blue arrows indicate how the user moves forward from the *DeviceScanActivity* to the *DeviceControlActivity*, and also back from the latter to the former.

The coding process for the development of this app, and so its structure, methods, functions, and flow are not in further detail explained. However, all the code can be accessed and so read on its public GitHub repository *(Vidal De Palol, 2019)*. The resultant App is detailed in the *Results* section.

2.2. Experiment

2.2.1. Aim

The goal of this experiment was to assess the Traumschreiber device capability on recording AEPs. More precisely, the AMLR and ALR components of it. The ABR and ECochG components were discarded to be observable as they are expected to occur within the first 10ms after the stimulus presentation. Hence it is impossible to be measured with the sampling rate of the Traumschreiber, as only two data points are recorded within this timespan.

2.2.2. Subjects

Four healthy students (2 males and 2 females aged 20 to 30 years old) were selected as participants of the experiment. These subjects were asked to not to consume alcohol within the last 24 hours before the trial, to sleep regularly the night before and to not to nap during that day before the participation. They were requested to sign the consent form (Reimann, 2018) (Schalkamp, 2018) before the start.

2.2.3. Materials and procedure

The recordings were done in a closed room for increasing the sound isolation factor. Any electronic device there was turned off to prevent additional signals from other devices being recorded, and thus try to reduce unnecessary noise in the acquired EEG raw data.

Figure 10. Subject of the experiment ready for the EEG recordings.

For acquiring the EEG data, a Xiaomi Mi A1 smartphone with the developed app running on it and the Traumschreiber system were used. Precisely, the channels 7 and 8 of the device (electrodes 6, 7, and 8) were used measuring the difference of the voltage amplitude from the temporal lobe regions behind the left and right ears (next to the mastoids) in respect to the frontal-parietal center. To better illustrate how the electrodes were placed on the scalp, a schematic representation of the International 10-20 system with the location of the recording electrodes highlighted in green is presented (see Figure 11).

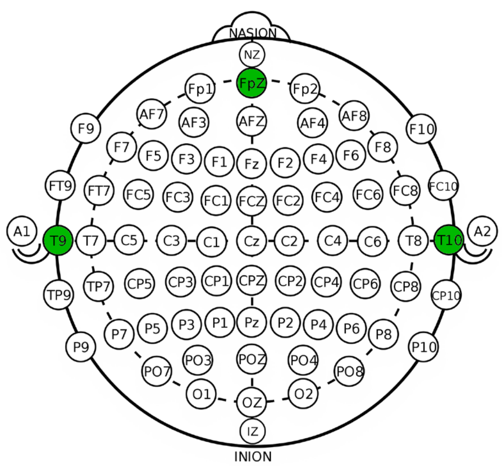


Figure 11. Electrodes placement (green) for this experiment in the International 10-20 system.

Four electrodes on regions without hair on the scalp in a bipolar montage were set. The electrode six was placed at the temporal lobe region behind the left ear (T9 on Figure 11), the electrode seven was located as a reference electrode on the frontal-parietal center region on the forehead (FpZ on Figure 11). The electrode eight was placed on the equivalent location of number 6 but the other side of the head, behind the right ear (T10 in Figure 11).

This configuration was chosen because the Traumschreiber is designed to calculate the channel voltage by subtracting the amplitude read by each pair of electrodes. Hence, input 1 (reference or non-inverting electrode) minus input two (inverting electrode) instead of sharing a common reference electrode between channels. In this way, to get the signal equally referenced, the data acquired from channel seven was posteriorly re-referenced by inverting the signal, hence multiplying all its values by -1. Additionally, the ground electrode (labeled as REF) was set to the top-left side of the forehead due to its short-length cable. This simple montage was also chosen as proven to be good enough and sufficient for capturing the desired events (W. Hall III, 2015). Furthermore, it is also proven that more recording electrodes do not necessarily improve the visibility of the ERPs (Schalkamp, 2018) (Reimann, 2018).

For the presentation of the stimuli, the earphones (see Figure 12) Monoprice Enhanced Bass Hi-Fi Noise Isolating were used (Monoprice, 2013). This model was chosen for being a standard-quality (20 Hz to 20 kHz frequency response) earbuds with a good noise-isolation capability. As they have a length of 122 cm, this was approximately the distance between the participant and the phone during the experiment.

Figure 12. Earphones used in the experiment.

Concerning the experiment design, each participant was recorded for around 30-35 minutes, meaning 15 EEG trials per subject, as each session lasted 112 seconds. This length was precisely defined as each trial consisted of 36 pip tone presentations of 500 milliseconds with a delay time of 3 seconds between presentations, with an additional 4 seconds before the first presentation.

Three different stimuli setups were used, hence five trials per different configuration. On the first five, the musical tone A4 (440 Hz) was the only present. On the second batch, it was A5 (880 Hz). On the third, a set of order-randomized musical tones from C4 to B6 (261.63 Hz - 1975.53 Hz), from semitone to semitone, were presented.

Before starting the experiment, the participants were told to sit on a chair, stay relaxed but aware of what they would hear, to keep their eyes closed, to try not to move and also to avoid swallowing. These recommendations were asked for preventing the recording of artifacts produced by the eyes while blinking, moving or any other muscle movements.

The musical tones used as stimuli of the experiment were generated by coding a Python script that uses the package SoX (Bagwell, 2008). With this, one can quickly generate audio files specifying frequency, length, fade in and fade out, volume gain, and many more parameters just by using commands on a Linux or Unix system. The script used to generate them can be accessed on the net (Vidal De Palol, 2019). For this experiment, the parameter values selected were: one audio channel (mono), 500 milliseconds as length, -0.01 as volume gain (for avoiding audio distortion or clipping), 16 bit and 44100 Hz as encoding and sampling frequency (as standard for digital audio in computers, compact discs, etc.). Moreover, a quadratic fade in and a logarithmic fade out of 10 milliseconds were set. In Figure 13, the resultant waveform for the A4 and A5 musical tones generated in the experiment are shown.

This length and fades waveform (quadratic and logarithmic) were explicitly designed for avoiding click sounds during the presentation of the stimuli (start and stop) as then the AEP components could be elicited from this. Hence the aim was to generate a tone as pure as possible without any extra noise sounds.

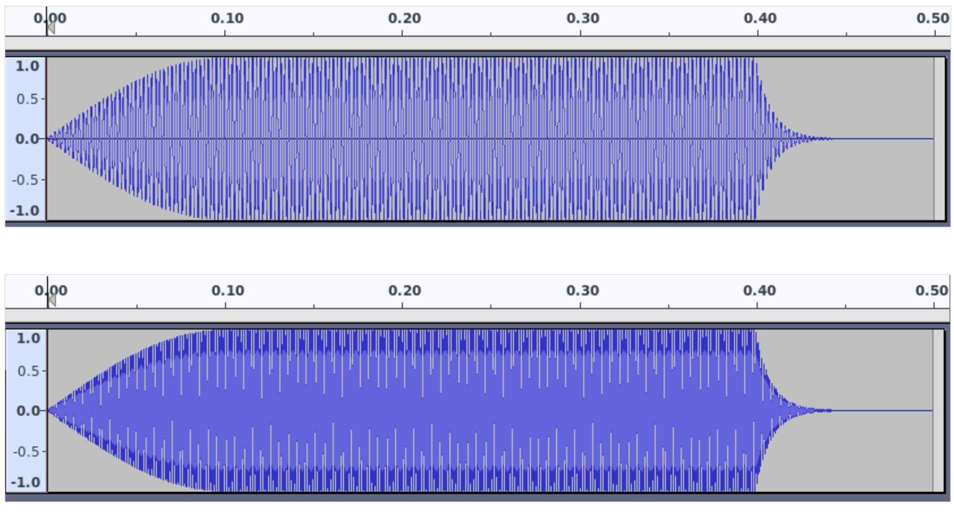


Figure 13. Generated A4 (top) and A5 (bottom) musical tones.

Concerning the data analysis procedure, it was divided into the following steps (Luck, 2014). The stored files containing the recorded data were read and processed using Jupyter Notebooks (Jupyter Team, 2015) written in the Python programming language. All the recordings, plots and scripts of this work can be accessed via GitHub (Vidal De Palol, 2019)

First, the whole trials were plotted to have a general view of the recorded data. This procedure was also done for discarding invalid trials because of non-working electrodes or unusable channels, and for checking the appearance of noticeable artifacts such the ones elicited by muscle or eye movements. Second, descriptive statistics from all trials were obtained to confirm or clear out all doubts about the unusable trials and channels for each subject.

Third, the stimuli appearance timestamps between trials were stored for facilitating the selection and extraction of epochs with maximum precision. This measure was taken because the trial-to-trial differences at the same presentation time can range from 1ms up to 30ms. Then baseline correction was applied at each epoch to ensure all epochs started at amplitude 0. Next, the epochs were averaged at each stimulus condition (A4. A5, C4-B6), and finally, the resultant averaged epoch was compared between subjects and stimuli experimental setup.

Afterward, a spectrum analysis of the valid trials was performed by using the DFT. The DFT or FFT algorithm converts a signal from its original domain (in this case voltage/time) into a frequency domain representation (power spectrum). By making use of it, one can easily see the most dominant frequencies that compose a signal. In the end, the FFT and its inverse were used to try to picture the underlying AEP components over the averaged signal.

3. Results

3.1. *The App*

As mentioned in previous sections, an Android App has been developed to use the Traumschreiber for visualizing, recording and posteriorly analyze EEG data. The App allows the user to interact with the Traumschreiber at distinct levels. On the following subsections, the standard App is described whereas the experiment version of it with its additional features is explained in the last subsection as an extension of the former.

3.1.1. Device search and connect

When the App is launched, on the first screen (see Figure 14), one can look for a Traumschreiber by touching the SCAN/STOP button. This task works primarily by scanning BLE devices that contain the same first ten digits of the physical address of a Traumschreiber. As a consequence, it acts as a filter, and thus no other BLE devices are shown.

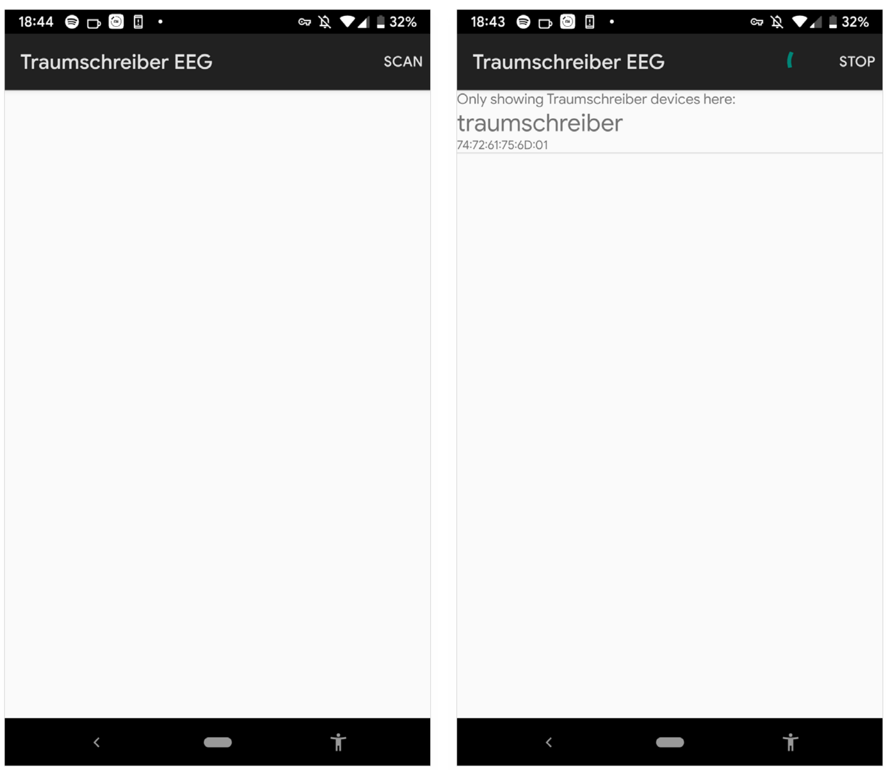


Figure 14. First screen of the app. No scanning (left) Scanning and device found (right).

As it is exposed on the right screenshot of Figure 14 when a device is found it is shown on an item-selectable list. As information, the device name and its physical address are displayed on a list item under the top bar. After this point, the item can be selected meaning that the user wants to connect to the device found. What follows is the main screen of this App for working with the EEG data the Traumschreiber measures and sends via Bluetooth.

3.1.2. Real-time information

From the top to the bottom (see Figure 15), it displays the physical address of the device, followed by the connection state (Connected/Disconnected), the voltage in microvolts from the eight channels at the moment, and the resolution of the device (sampling rate) calculated in milliseconds and in Hertz. This last essentially represents the time average in milliseconds between the octets of data received (e.g., T = 4.5ms), and the number of octets received in a second (e.g. R = 1/T = 222Hz) where T is expressed in seconds.

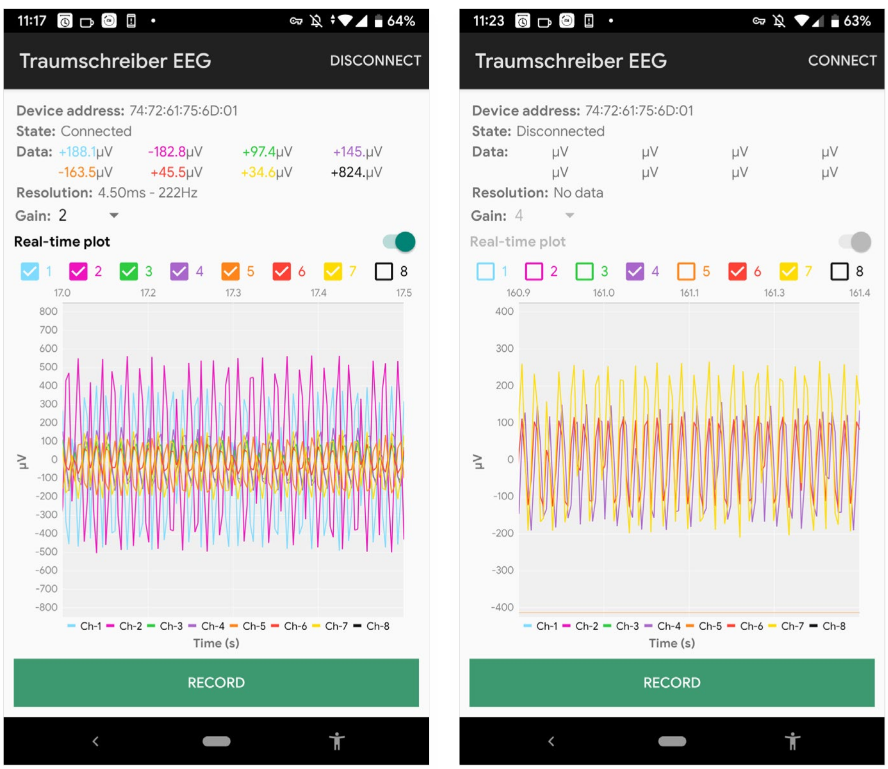


Figure 15. App main screen, plotting 7 channels (left). Data stream disconnected, 3 channels enabled (right).

Note that when the App and the Traumschreiber are not connected (Figure 15, right), there are no values on the data field, the state *Disconnected* is shown, and all controls but for going back to the main screen and the *CONNECT/DISCONNECT* button are disabled. This last was implemented for preventing bugs and app crashes.

Furthermore, the current gain amplification factor value appears in a selectable list (e.g., set to 2 and 4 on Figure 15 screenshots). As explained on the Traumschreiber section of this work, this configurable parameter allows the user to limit the maximum and minimum values producing this "zoom in" effect while increasing the gain and "zoom out" by decreasing it.

To illustrate better how this is possible, the gain conversion formula *V = X * 1.65V / (1000 * GAIN * 2048)* is explained. What we want to calculate is the input voltage. X represents one of the values contained in the received octet of data, as the current channel voltage expressed as a 2-bit integer value after the binary value received via Bluetooth. Then this value is multiplied by 1.65 meaning the maximum positive or minimum negative voltage value the Traumschreiber can theoretically measure. Precisely, since the device has a 3.3V battery range, the amplified signal saturates and clips off at -1.65V and +1.65V. Then we divide this product by the hardware analog amplification factor value 1000, times the currently selected gain set via App, times 2048. This last value comes from the 12-bit accuracy recording of the device, meaning 2^{12} = 4096 possible values. Hence, this digit elucidates the digitalized scaled range from -2048 to 2047 (0 included).

3.1.3. EEG plotting

What is presented below in the App are the data plot and the recording button for storing the EEG measured trials or sessions. As displayed in Figure 15, a switch button is the one in charge to show or hide the real-time plot of the enabled channels. When this one is switched on, the layout expands up to fill the space left on screen with eight checkboxes (one per channel displayed) followed by a 2D axis data representation.

The plot is achieved using the Java library for Android MPAndroidChart (Jahoda, PhilJay/MPAndroidChart, 2019). This library is well-known on the Android developing community for being a powerful and easy to use chart solution for this OS.

Concerning the chart axis, the Y (left) represents the voltage amplitude measured in microvolts, whereas the X (top) signifies the elapsed time in seconds. Note that the decimals on the labels of the axis represent the milliseconds within a second.

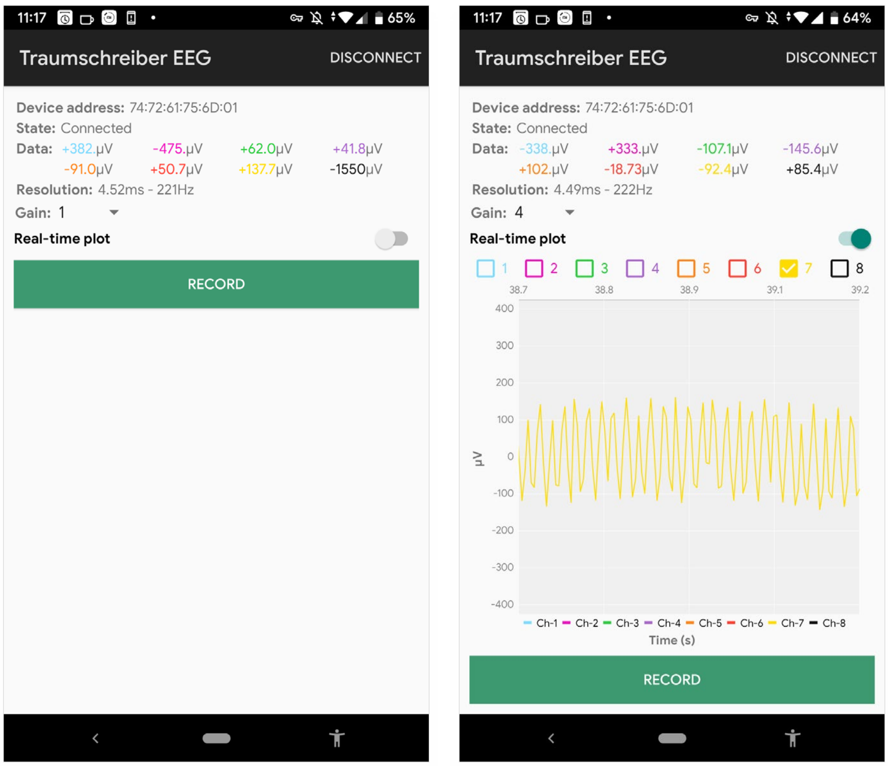


Figure 16. App main screen, hidden plot (left), plotting one channel (right).

The range of the Y-axis is adjusted accordingly with the maximum and minimum possible value measured with the currently selected gain. By formula definition, at gain one, the signal will clip at +1650/-1650 microvolts (1.65 millivolts). Thus, is the maximum it can measure taking into account that the gain is amplified 1000 times on the hardware of the device (so 1.65V / 1000). Hence, the Y-axis scales itself in the following way:

Table 4. Y-axis estimated limit in microvolts per gain distribution.

Gain	0.5	1	2	4	8	16	32	64
Y-axis limit (µV)	±2000	±1650	±825	±412.5	±206.25	±103.13	±51.56	±25.78

Following the same distribution, see Table 4, as each gain factor is value doubled in respect to the previous one, the signal limit gets precisely divided by two from lower to higher gain parameters. The only exception is 0.5, which in fact clips at 2000. The reason behind is the measure limitation by hardware. Because, although following the same principle the value should be 1650 * 2 = 3300, actually the device tops measuring at 2V because while measuring, the supply voltage ranges from 1.65V until 2V maximum. Also, although this distribution is an accurate estimation, the Y-axis limit on the App was set a bit higher than the presented above to show all the allocated points inside the chart and prevent the clipping data from laying out of it.

When it comes to the range set for the X-axis, as we have a continuous time-series representation, it is never fixed since it is continuously receiving a data stream over time. However, a visible on-screen range of 500 milliseconds was chosen since most of ERP components happen between this timespan and half a second can be still considered "human-readable" in terms of speed.

It is important to mention that real-time plotting, meaning allocating all values as they are received from the Traumschreiber was not possible. The library itself is not able to handle that amount of data allocation in a such a small-time range (around 220 plot calls of 8 points within a second in this case). The MPAndroidChart developer confirms this himself answering to a user issue on the GitHub repository of the library. Precisely saying "240 values per second is not going to work (or at least not for very long), even on a high-end device. The library is not feasible for that amount of data plotted at real-time" (Jahoda, 2016). Nevertheless, many attempts to achieve this or to get as close as possible to display a real-time chart were done.

Finally, real-time plotting was achieved by taking into account the human eye limitations. Humans watch movies at the standard frame rate of 24, meaning 24 images are presented within a second to perceive fluid motion. This same idea was taken to prevent too many plot calls and so reduce the CPU load of this task. Moreover, thus in the end, “real” real-time plotting is unnecessary for human eyes to perceive motion without flickering. The solution implemented was to accumulate the received values in epochs of 30 milliseconds, producing a 33.3 FPS plot.

Furthermore, for freeing even more resources of the device computational threads were used. Thus, not only splitting the plotting process into smaller operations but also running it in different threads improved the performance of the App considerably. This advance is explained by the use of the multiple cores contained on current device processors for running these threads in parallel instead of making each sub-operation wait for a chance to run (Google Developers, 2019).

3.1.4. EEG recording

As previously mentioned, and also shown on previous (Figure 15 and Figure 16) and next figures (Figure 17), the App has a recording button. This element allows the user to start and stop the recording of an EEG session.

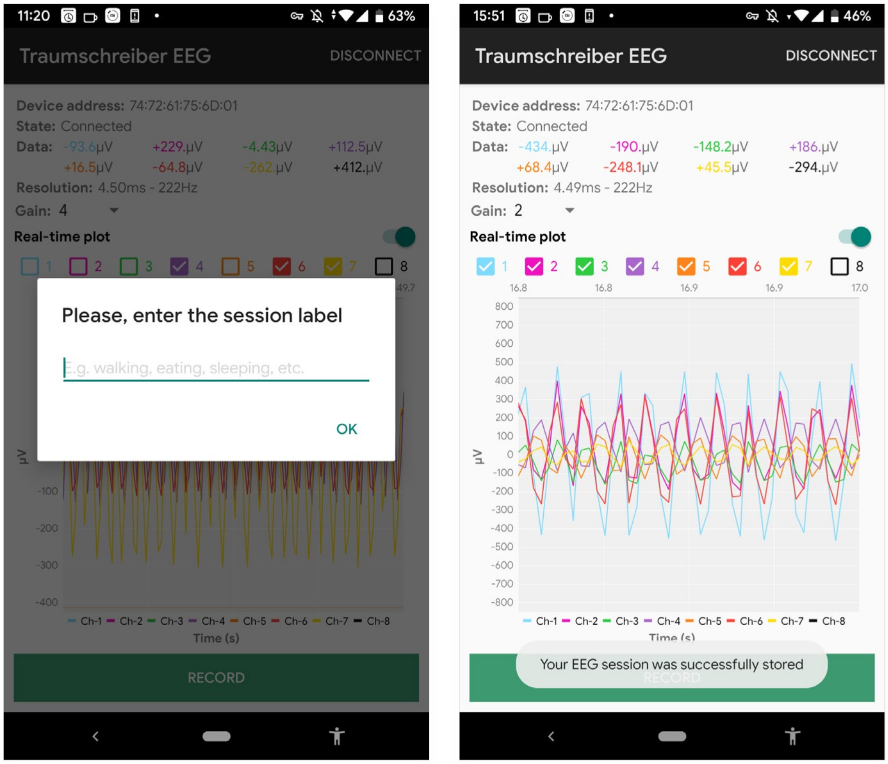
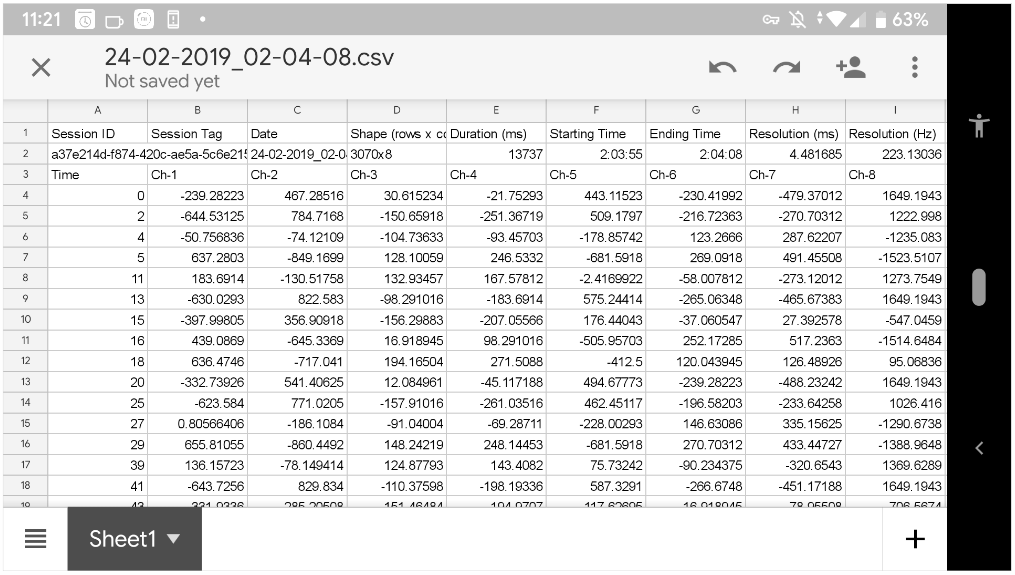


Figure 17. Session labeling input form field (left). EEG session stored with a confirmation message (right).

Right after the button is pressed, an input form field is presented (Figure 17, left). This field lets the App handler to tag the session before the recording process starts. Once the information is there, the storing of the data is done on memory until the button is selected again. The button text *RECORD* changes to *STOP AND STORE* during the recording to increase the information status of the task. In case no session label is inputted, the recording starts and takes the label *Default*.

This recording process ends when the button is again pressed. Then the data stored on memory is written on a CSV file. When this last is succeeds, the message *Your EEG session was successfully stored* appears on screen (Figure 17, right).


11:21 63%
24-02-2019_02-04-08.csv
Not saved yet

	A	B	C	D	E	F	G	H	I
1	Session ID	Session Tag	Date	Shape (rows x cc	Duration (ms)	Starting Time	Ending Time	Resolution (ms)	Resolution (Hz)
2	a37e214d-f874-420c-ae5a-5c6e21!	24-02-2019_02-0	3070x8		13737	2:03:55	2:04:08	4.481685	223.13036
3	Time	Ch-1	Ch-2	Ch-3	Ch-4	Ch-5	Ch-6	Ch-7	Ch-8
4	0	-239.28223	467.28516	30.615234	-21.75293	443.11523	-230.41992	-479.37012	1649.1943
5	2	-644.53125	784.7168	-150.65918	-251.36719	509.1797	-216.72363	-270.70312	1222.998
6	4	-50.756836	-74.12109	-104.73633	-93.45703	-178.85742	123.2666	287.62207	-1235.083
7	5	637.2803	-849.1699	128.10059	246.5332	-681.5918	269.0918	491.45508	-1523.5107
8	11	183.6914	-130.51758	132.93457	167.57812	-2.4169922	-58.007812	-273.12012	1273.7549
9	13	-630.0293	822.583	-98.291016	-183.6914	575.24414	-265.06348	-465.67383	1649.1943
10	15	-397.99805	356.90918	-156.29883	-207.05566	176.44043	-37.060547	27.392578	-547.0459
11	16	439.0869	-645.3369	16.918945	98.291016	-505.95703	252.17285	517.2363	-1514.6484
12	18	636.4746	-717.041	194.16504	271.5088	-412.5	120.043945	126.48926	95.06836
13	20	-332.73926	541.40625	12.084961	-45.117188	494.67773	-239.28223	-488.23242	1649.1943
14	25	-623.584	771.0205	-157.91016	-261.03516	462.45117	-196.58203	-233.64258	1026.416
15	27	0.80566406	-186.1084	-91.04004	-69.28711	-228.00293	146.63086	335.15625	-1290.6738
16	29	655.81055	-860.4492	148.24219	248.14453	-681.5918	270.70312	433.44727	-1388.9648
17	39	136.15723	-78.149414	124.87793	143.4082	75.73242	-90.234375	-320.6543	1369.6289
18	41	-643.7256	829.834	-110.37598	-198.19336	587.3291	-266.6748	-451.17188	1649.1943

Sheet1

Figure 18. Cropped view of a sample CSV file containing an EEG stored session on Google Spreadsheets for Android.

The saved CSV files are organized as follows (see Figure 18). The two first rows contain headers and its correspondent values below about different aspects of the EEG session. A UUID is saved on the first place under *Session ID*, followed by the inputted tag under *Session Tag* , the current date in format dd-MM-yyyy_HH-mm-ss (Oracle, 2018), the shape of the stored data as number of rows times number of columns, the session duration in milliseconds, the starting and ending times (HH:MM:SS formatted), the sampling rate in milliseconds and Hertz as *Resolution*, the unit measure (always in microvolts) and, the starting and ending system timestamp in milliseconds. These last values are a standard measure in Computer Science. They are computed as the difference, measured in milliseconds, between the current time and midnight, January 1, 1970, UTC (Tutorialspoint, 2019).

The rest of rows are nothing less than the actual voltage amplitude measured in microvolts per each channel (column), plus the correspondent timing of these expressed in milliseconds since the first octet of data of the recording was received. The filename is always defined by the actual second-precision datetime when the storing finishes (e.g., from Figure 18: 24-02-2019_02-04-08 as Day-Month-Year_Hour-Minute-Second).

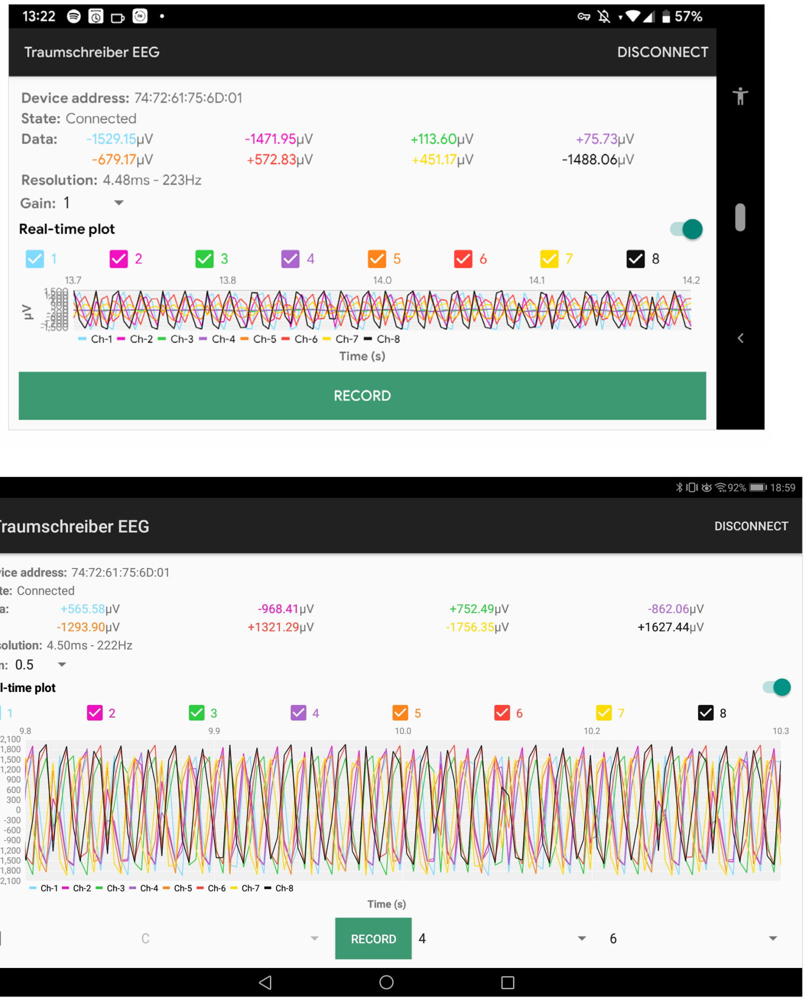


Figure 19. Main activity of the App in landscape orientation mode for a smartphone (top) versus tablet (bottom).

Another last important aspect to mention is that the App has been designed to work in both portrait and landscape orientation modes. This feature has been achieved by defining weights in the UI editor of Android Studio for this project, as it is its method to distribute the presented content in a percentage fashion automatically. Hence, the resolution and the orientation of the screen are irrelevant because the App adapts programmatically to the host screen characteristics (see Figure 19).

However, when it comes to appreciating better the data represented on the chart, the portrait mode is more convenient since there is more space in terms of height, and thus the amplitude of the wave can be drawn with more pixels.

3.1.5. The AEP experiment version

It is mentioned that the App presented in this work was designed not only for the experiment of this study but also keeping in mind the possibility of being used as a standard option with the Traumschreiber device by students, researchers, and enthusiasts. Thus, two versions of the same App were developed in parallel. Since the experiment version is nothing less than the standard one with some experiment-specific additional features, the former is presented first, and the latter (mainly and only the differences) is described next.

Concerning the UI, both apps are identical, except for the record button and the elements presented on both sides of it (see Figure 19, bottom). These are four: a checkbox and a selectable list on the left side, and two selectable lists on the right side.

Since on the experiment musical tones are presented as the audio stimuli expected to evoke an auditory response, the controls described above are precisely the parameters that define what musical tone or tones are going to be played on the ears of the subject.

The first checkbox defines if a single stimulus (checked) or a range of them (unchecked) will be used. The first selectable list contains all musical keys from C to B from semitone to semitone (C-C#-D-D#-E-F-F#-G-G#-A-A#-B). If the checkbox is checked, this control is enabled because the experimenter must select a key for picking the desired stimulus. Otherwise, it is disabled because the user must choose a range instead. The following two selectable lists both contain the numbers from 0 to 11. If the checkbox is unchecked, these numbers allow the researcher to select a range of octaves. Meaning the selected range define the set of stimuli chosen to present. For example, if the chosen values are 4 and 6, it will pick all musical tones (from semitone to semitone) from C to B between the octaves 4 and 6 inclusive. In case the checkbox is checked. The first selectable list sets the octave of the selected key and the second the number of full octaves it will present. Thus, understanding a full octave as 12 presentations of the same stimulus.

What happens internally in the App when a specific experimental setting is chosen, is that a set of number representations are generated. In case of the repetition of a single stimulus, since a key can be represented as a number from 0 to 11 (where 0 is C and 11 is B), and the selected octave is already a number, a sequence of key-octave pairs is generated in a list fashion. The same couple repeats 12 times the number of octaves selected on the last selectable list control. In case of a range of musical tones set as stimuli, the representation would contain a randomly ordered sequence of key-pairs as well, but all of them unique, representing all the musical tones within the range of octaves picked. It is important to highlight that the App displays the current experimental settings not only when the main screen is loaded, but also every time any of these experiment controls is changed (see Figure 20). Precisely, a notification with the stimulus or stimuli range, the number of presentations and the total time of the trial are shown.

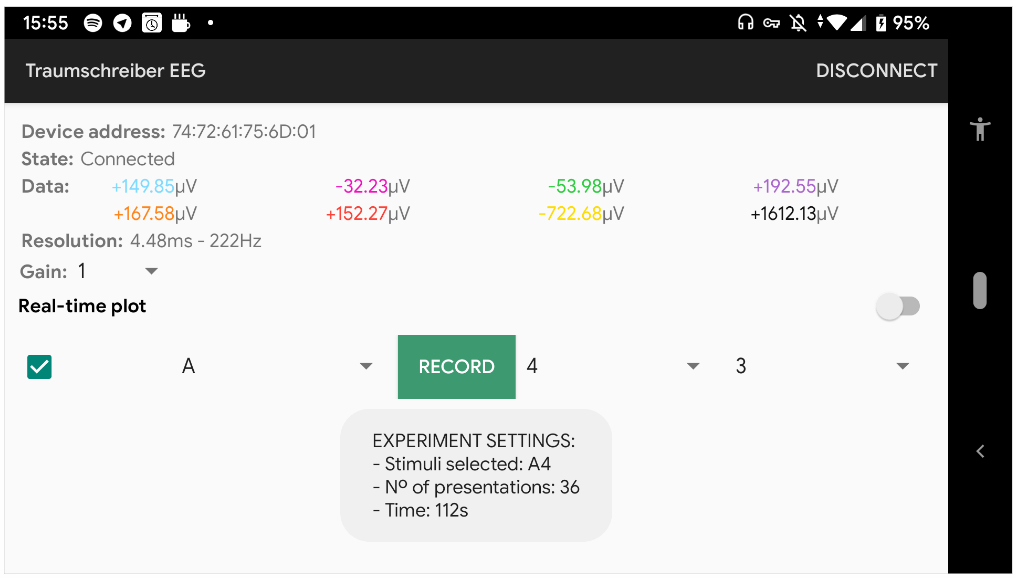


Figure 20. Experiment settings displayed on the screen.

There is one last aspect left to explain about the extra features added onto the experiment version of the App in respect to the standard. Moreover, this is how the key-octave pairs as musical tone representations are translated into the stimuli presented. This conversion is done primarily by turning the key and octave numbers into the WAVE (.wav) filename that is going to be played. It illustrates as follows:

Table 5. Key number representation with its correspondent musical key value translation.

Key number	0	1	2	3	4	5	6	7	8	9	10	11
Key value	C	C#	D	D#	E	F	F#	G	G#	A	A#	B

The stimuli files are already stored on the phone external storage (Google Developers, 2019) precisely under the folder *Tones*. The stimuli files are named using the key followed by the octave and the file extension (e.g., C#5.wav, A4.wav, D#6.wav). Hence, the App plays the correspondent file at every stimulus presentation time after the key-pair number representation to filename translation.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Session ID	Session Tag	Date	Stimuli	Presentations	Volume	Shape (rows :	Duration (ms)	Starting Time	Ending Time	Resolution	Resolution (Hz)
2	f01c9822-5d74-47ea	closed	18-02-2019_16-27-22	A4	36	0	24957x8	112026	25:30.8	27:22.8	4.488582	222.7875
3	Stimuli appearance	4041	7044	10047	13037	16036	19037	22038	25045	28038	31047	34039
4	Time	Ch-1	Ch-2	Ch-3	Ch-4	Ch-5	Ch-6	Ch-7	Ch-8	Key	Freq	
5	48	-7.4020386	51.537323	-49.774933	-34.215546	4.9346924	51.537323	8.182526	0.67977905	silence	0	
6	50	-22.382355	-10.045624	13.671112	14.52713	-36.229706	-51.5625	-0.75531006	19.864655	silence	0	
7	52	-8.686066	-51.5625	-51.5625	29.381561	-13.948059	-51.5625	-12.084961	16.868591	silence	0	
8	54	32.4028	-3.8520813	9.114075	-10.951996	21.07315	51.537323	-29.079437	33.25882	silence	0	
9	55	24.572754	51.537323	-51.537323	-35.877228	20.368195	51.537323	6.2942505	2.2659302	silence	0	
10	56	-14.552307	33.863068	-16.818237	-1.8127441	-21.475983	-8.006287	-7.3516846	22.483063	silence	0	
11	57	-18.73169	-51.5625	51.260376	31.49643	-26.939392	-51.5625	-9.894562	27.065277	silence	0	
12	60	19.285583	-51.5625	41.91971	5.1864624	14.376068	-21.173859	1.0574341	2.3162842	silence	0	
13	62	35.625458	51.537323	-44.68918	-23.993683	28.248596	51.537323	6.3446045	-6.4201355	silence	0	
14	64	-4.0786743	51.537323	-51.537323	-23.76709	-9.088898	50.05188	9.970093	-0.88119507	silence	0	
15	65	-26.939392	-19.487	20.871735	21.627045	-32.276917	-51.5625	-2.8701782	2.5680542	silence	0	
16	67	3.5499573	-51.5625	-51.5625	28.97873	-16.54129	-51.5625	-5.5892944	4.7584534	silence	0	
17	70	41.466522	-0.50354004	6.9236755	-13.318634	22.155762	51.537323	-0.25177002	9.693146	silence	0	

Figure 21. Header of a sample CSV file of an EEG stored session of the experiment App.

Additionally, when it comes to storing the EEG recorded session onto a CSV file, the experiment App contains also some extra information.

On the one hand (see Figure 21), the selected stimulus or stimuli range (musical tones) and the number of stimuli presentations is registered. Also, the internal audio volume value of the device in dB (from -Infinite to 0dB, not the actual SPL) is written. Moreover, the third row of values is added. As the name of the first cell indicates, the stimuli appearance times since the starting of the EEG experiment trial in milliseconds are listed there. Another essential difference to mention in respect to the standard App is the filename of the stored CSV. As explained above, the standard names it as follows *Day-Month-Year_Hour-Minute-Second* whereas the experiment one adds the stimulus or stimuli range (e.g., B#, C3-B7) afterward, plus the session tag. Thus, the naming format is as follows *Day-Month-Year_Hour-Minute-Second_Stimuli_SessionTag*.

	A	B	C	D	E	F	G	H	I	J	K
898	4008	-14.325714	-51.5625	-51.5625	31.924438	-12.940979	-51.5625	-0.52871704	-25.806427	silence	0
899	4010	24.698639	-26.53656	18.454742	-0.60424805	21.148682	30.967712	9.114075	-40.30838	silence	0
900	4012	22.734833	51.537323	-51.537323	-35.92758	17.699432	51.537323	18.630981	-32.025146	silence	0
901	4044	-19.159698	46.829224	-41.768646	-6.7474365	-18.958282	17.97638	24.8497	-44.890594	A5	880
902	4045	-22.105408	-51.5625	41.365814	24.925232	-31.49643	-51.5625	19.260406	-17.447662	A5	880
903	4051	12.462616	-51.5625	51.512146	21.173859	1.9134521	-49.422455	14.804077	-25.177002	A5	880
904	4053	26.007843	32.0755	-29.98581	-23.59085	20.468903	51.537323	20.871735	-27.090454	A5	880
905	4055	-4.3304443	51.537323	-51.537323	-27.669525	7.0999146	51.537323	27.694702	-32.604218	A5	880
906	4057	-29.608154	-5.4130554	-2.1400452	7.3768616	-21.82846	-51.5625	28.47519	-24.648285	A5	880
907	4058	-8.207703	-51.5625	-51.5625	30.841827	-7.0495605	-51.5625	15.861511	-26.083374	A5	880
908	4059	36.078644	-20.015717	14.099121	-2.5428772	25.881958	50.454712	19.638062	-32.4028	A5	880
909	4063	19.713593	51.537323	-49.598694	-24.144745	12.059784	51.537323	22.206116	-29.960632	A5	880
910	4065	-16.692352	39.326477	-32.15103	-4.3556213	-18.404388	19.159698	21.954346	-26.133728	A5	880
911	4067	-15.509033	-51.5625	38.87329	36.657715	-32.100677	-51.5625	3.2730103	-10.24704	A5	880
912	4077	21.57669	-51.5625	43.883514	26.838684	6.6467285	-28.55072	9.466553	-14.325714	A5	880
913	4079	32.17621	40.207672	-34.34143	-24.446869	26.259613	51.537323	24.472046	-32.704926	A5	880
914	4098	-7.8552246	51.537323	-51.537323	-25.453949	4.1038513	51.537323	35.02121	-30.942535	A5	880
915	4100	-27.54364	-20.141602	11.153412	18.354034	-28.651428	-51.5625	12.966156	-29.356384	A5	880
916	4101	1.8127441	-51.5625	-51.5625	35.071564	-10.095978	-51.5625	27.014923	-31.773376	A5	880
917	4103	33.384705	-12.638855	13.595581	-3.3737183	21.299744	51.537323	33.83789	-44.110107	A5	880
918	4105	23.641205	51.537323	-48.21396	-28.324127	11.883545	51.537323	47.584534	-51.5625	A5	880

Figure 22. Cropped sample of an EEG stored session CSV file when a stimulus is presented.

One last relevant thing to comment is the two extra columns of values added after the eight channels. On the *Key* column (see Figure 21), the stimuli annotated as the musical key is registered (see Figure 21 and Figure 22) whereas the translation of this last one as musical tone expressed in the frequency domain (Hertz) is on the *Frequency* column. In the case of no presentation, the key value is represented as *silence*, while *0* is the equivalent chosen for the most-right column.

Although the frequency translation could have been done easily defining a dictionary of key-value pairs to map each musical key to a specific frequency value, this is computed using the physics of music formula (MTU Physics, 2019). This expression is given by $f_n = f_0 * (a)^n$, where f_n is the frequency we want to calculate and f_0 is the frequency of one fixed note which must be defined. In this case, and, as a common choice for standard tuning, A4 is the one, so 440Hz. *n* is the number of half steps away from the fixed note you are (A4 in this case). Thus, if you lay on a higher note, *n* is positive whereas n is negative in case of a lower one. $a = 2^{1/12} = 1.059463094359\ldots$ For instance, if we want to compute the frequency of C5, we obtain $f_n = 440 * (1.059463094359)^3 = 523.3$Hz. In this example, 3 is the number of half steps that separate A4 from C5, and since C5 is higher, the difference is positive.

3.1.6. Issues and limitations

Although in general, the App works as expected, this does not apply in all cases. Overall, it connects and disconnects to the device when the action is solicited, and the same applies to the rest of the controls. Including setting the gain, hiding or showing the chart, storing and stopping the recording of the data received, entering a session name, establishing the experiment settings. Even displaying real-time information such as connection state, device address, current sampling rate, and the experiment configuration.

However, the Bluetooth connectivity and previous data acquiring do not always work on the first try. When one tries to connect to the Traumschreiber after the device is just turned on, now and then it does not automatically connect to the data channel. Sometimes it happens with some delay or, from time to time, touching connect on the top menu bar is needed. Occasionally even returning to the device search screen is required in order to connect to it and receive data successfully.

3.2. Data analysis

The results described in this section can be examined in more detail on the appendices A.2 Preliminary Plots, A.3 Descriptive statistics, A.4 Final Plots. The discarded trials are highlighted in red color on A.3.

3.2.1. Trial inspection and selection

Subject 1

Looking at the trial plots, starting with the first subject, the plot and the descriptive statistics show that channel 8 is in all three conditions slightly different than channel 7. Precisely, considerable background noise seems to be present on the signal across the whole experiment. When compared, the tendency mostly indicates that both standard deviation and the expected value of the set are two times bigger on channel 8 in respect to channel 7. Hence, the former was discarded to be used in further steps of the data processing.

Subject 2

In this case, the charts and the statistical data show a prominent noise-saturated signal in most of channel 8 trials except for the first of A1, where the descriptive data confirms the poor or almost not present signal (σ = -0.096, μ = 0.49). Nevertheless, trials 4 and 5 in A5 and 1, 2, 4, 5 in C4-B6 from channel 8 were kept for further processing because of having similar mean and SD values compared to Subject 1's selected trials.

In the case of channel 7, trials 2 from A4 and 2 from A5 were discarded for having SD around 3 times more prominent than valid trials from Subject 1 and 2.

Subject 3

In case of this subject, all trials needed to be rejected for later processing as the descriptive statistics reveal SDs between 5 and 25 times the magnitude of "normal" trials from Subjects 1 and 2. The unreliability of the data recorded can also be seen on the plots, as it appears hugely variant.

Subject 4

For the last subject, surprisingly, the saturated noisy channel was 7 instead of 8, unlike most of the previous cases. Thus, all channel 7 trials were discarded for having SDs between 5 and 8 times the labeled as valid. In the case of channel 8, the first trial of A5 was rejected for having a SD twice as large compared to the rest that seems to follow the norm of the selected ones.

3.2.2. Averaged AMLR epochs

Before start commenting the final results of this analysis, and for having a better picture of the components of interest, in this case, an ideal AMLR waveform showing its major peaks at typical latencies is presented (see Figure 23).

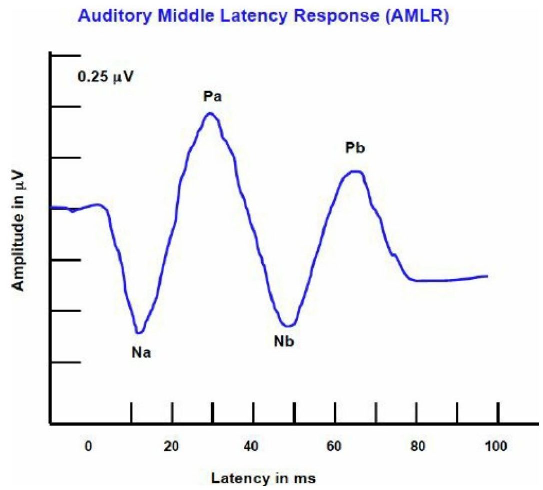


Figure 23. Ideal AMLR waveform (W. Hall III, 2015)

The results plotted seem to indicate that the AMLR components are represented on the outcome of the epochs averaged for Subject 1 (Figure 24). Components Na, Pa, Nb, and Pb look quite clear at around the expected latencies in the three conditions.

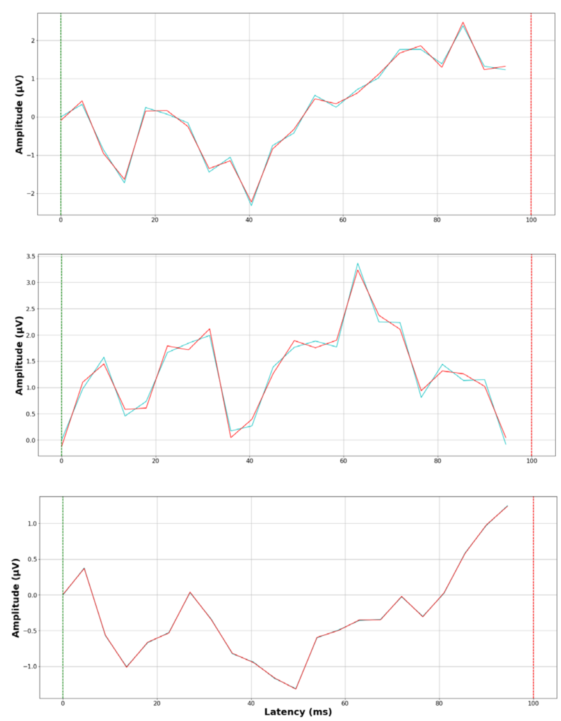


Figure 24. Averaged AMLR epochs for A4 (top), A5 (middle), and C4-B6 (bottom) of Subject 1.

Although the amplitudes do not coincide precisely with the ideal AMLR, they are indeed close to the expectancy. Another critical aspect to highlight is that the signal is undoubtedly stronger in the first two conditions. Also, there is precisely where the same stimulus is presented. Moreover, the amplitude of the components registered on A5 (880Hz) is more significant in comparison with A4 (440Hz). This observation follows the idea that higher frequencies should produce more prominent responses than lower ones.

Subject 1 results for AMLR epochs were the only clear outcomes to distinguish its components.

3.2.3. Averaged ALR epochs

In this description, an ideal ALR is also included to better illustrate the similarities and differences between the findings and the idyllic case.

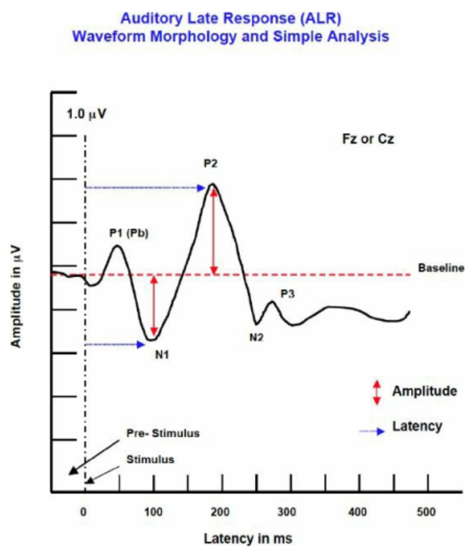
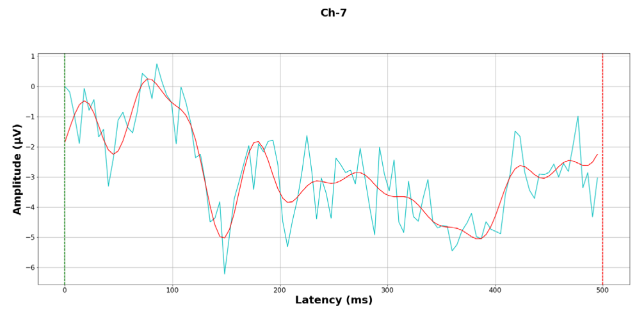
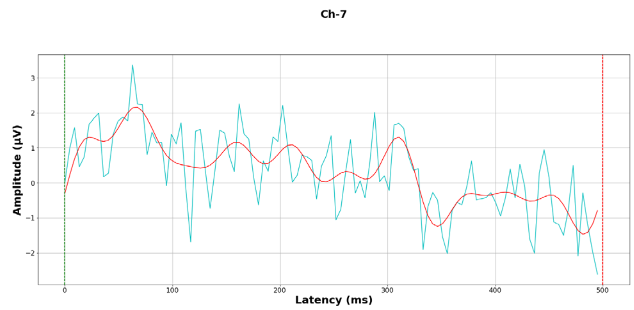


Figure 25. Ideal ALR waveform (W. Hall III, 2015)

In this case, Subject 1 is again the closest outcome when comparing with the theoretical ALR (Figure 25). Here only the peaks from the conditions A4 and A5 can resemble the ALR components. Precisely, P1, N1, and P2. P3 was not expected since it is a cognitive-elicited component that requires specific attentional tasks from the participant.





Once more, Subject 1 results for ALR epochs are the only clear outcomes to distinguish the components of this AEP.

4. Discussion

4.1. App improvements and issues

The fact that from 60 recordings, only one was not stored correctly, denotes that the App is at the point to be ready for daily use, or at least for being tested on an everyday basis. Moreover, although it can still break from going carelessly back and forward in suboptimal situations (e.g., disconnecting data while plotting, trying to connect and disconnect too fast, etc.), it can accomplish its purpose most of the time: display real-time EEG data and store EEG sessions.

In order to improve its current state, and hence reduce the ratio of failure, extensive and careful testing should be done. This means, trying it on different devices in conjunction with a deeper understanding of the Bluetooth handling library. Just as an example, the way how the two mid-range devices used in this present work communicate and react to the Traumschreiber via BLE protocol it is different enough to consider adapting the Bluetooth library. Hence, this would necessarily mean, for instance, to find a compromise by probably reducing the connection speed for increasing the connection compatibility with more devices.

Additionally, the file format the App stores the EEG sessions could also be optimized. Because even though there is no problem at all using CSV files. On the contrary, they are plain text files and a standard format when it comes to reading and storing data. However, to make some numbers as an example, 112 seconds of EEG data on a CSV occupy around 2.5MB. Thus, as the battery of the Traumschreiber can last for some hours and long EEG sessions could be recorded, a compressed format would be optimal to solve this issue.

A future enhancement in this same direction could be to add an option to stream the data via TCP or UDP protocol. In this way, the measured voltage could be directly read by another device, for instance, a laptop, avoiding the need to connect to the smartphone or tablet to copy the files every time new recordings are being done.

4.2. EEG data processing and analysis

Many different procedures for processing and analyzing the recorded data were used in this work even though only some of them have been described until this point. As already stated in a previous section of this work, there is no standard procedure to find AEPs components, and the best way should always be the one producing the best results.

The results of the data analysis showed that with the steps used, only the case for one subject could be considered revealing enough to claim that maybe the components are being seen, and thus they were recorded. However, no clear nor single explanation has been found to explain why the results found are not as promising as perhaps they should.

The stimuli presented to the participants were 500ms-long pips as musical tones. It could be that their length influenced the appearance of the AEPs or masked them with other waveforms reflected on the signal.

Another explanation could be that maybe the uniform time distribution of the data points was a wrong or not accurate enough assumption. For instance, taking 223Hz as a sampling rate, if we compare the assumption of 4.5ms with 4.4ms, in one second the last data point would be laying on 4.5*223 = 1003.5ms whereas the other would lay on 4.5*223 = 981.2ms. Thus, they would be 22.3ms distanced in one second and 220.3ms after 10 seconds. One more decision on the procedure that probably also affected the results significantly, in this case in terms of noise canceling on average, could be the epoch selection. Because the stimuli presentation timestamps never lay at the same point across trials and sometimes the latency between them is as significant as 30ms, the epochs were selected with this in mind. In this way, the epoch selection was made with the maximum precision, but it could have then affected the random noise canceling factor by reducing it instead of being increased as it expected on average. This could have maybe been improved by taking the average of the timestamps across trials instead and thus using epochs starting and finishing at the same time point.

Different methods were used to try to improve the quality of the averaged signal. Unfortunately, none of them helped to improve the visibility of the AEPs components in the end. For instance, the SNR was calculated, defined as $20log_{10}(RMS_{epoch}/RMS_{noise})$ assuming the noise as the pre-stimulus noise selecting 200ms before the stimulus onset. This measure was tried for discarding epochs were the SNR was too small, meaning the signal was not strong enough to be considered a candidate to improve the quality of the averaged signal. A similar but simpler technique was also tried without success by defining an amplitude range (minimum and maximum) for discarding epochs with too prominent amplitudes.

It is also important to highlight that two different baseline correction methods were tried. On the one hand, just subtracting the first value of a selected epoch from each data point in it, shifting the voltage value by making all epochs start at 0 amplitude. On the other hand, subtracting the mean of the last 200ms before the stimulus onset to the whole epoch, as a measure of noise reduction, and then applying the first element subtraction again. Nevertheless, no differences at all were observed in terms of improving the quality of the signal within these two methodologies.

Different band-pass filters were also assessed by first analyzing the power spectrum with the FFT algorithm, depicting the most prominent frequencies, and then reconstructing the signal after removing the unwanted ones. However, no specific useful filter values were found. This is probably because FFT algorithms assume the frequencies that comprise a signal are all continuous from beginning to end. Moreover, this assumption may be useful in certain situations for brainwaves like alpha, beta, delta or theta. However, ERPs usually occur in a short period, thus applying filters may erase partially or entirely the trace of the wanted potential. Nevertheless, FFT and its inverse were used to try better to illustrate the shape of the AEP components among the noise, at it can be considered the case of the findings for Subject 1.

Conclusions

The App developed in this work has proven to be good performant on visualizing and recording EEG data with the Traumschreiber device. However, to ensure the compatibility of the App, there is indeed the necessity of being further tested in more Android devices as the diversity of models, and thus hardware and OSs versions of this is enormous.

The results found from the data analyzed of this experiment underline the need for better exploring EEG data and waveforms processing techniques for obtaining better and more definite results. Moreover, a more standard experimental design for ensuring the finding of AEPs components may be crucial for a successful outcome.

References

ACNS. (2006). *Guideline 8: Guidelines for Recording Clinical EEG on Digital Media.* Retrieved from American Clinical Neurophysiology Society: https://www.acns.org/pdf/guidelines/Guideline-8.pdf

Appel, K. (2018). *The Traumschreiber System: Enabling Crowd-based, Machine Learning-driven, Complex, Polysomnographic Sleep and Dream Experiments.*

Appel, K., & Leugering, J. (2017, January 17). *Neuroinformatics - Institute of Cognitive Science*. Retrieved from The Traumschreiber Project: https://www.traumschreiber.uni-osnabrueck.de/

Bagwell, C. (2008). *SoX - Sound eXchange, the Swiss Army knife of audio manipulation*. Retrieved from Linux man page: https://linux.die.net/man/1/sox

Biopac Systems, Inc. (2014, December 31). *Ground vs. reference for EEG recording*. Retrieved from Biopac: https://www.biopac.com/knowledge-base/ground-vs-reference-for-eeg-recording/

Chibueze, E. (2016, May 16). *How do YOU Define Smartphone Brackets (Mid-range, etc)? Price, or Specs?* Retrieved from XDA-Developers: https://www.xda-developers.com/how-do-you-define-smartphone-categories-price-or-specs/

Cohen, M. X. (2014). Advantages and Limitations of Time- and Time-Frequency-Domain Analyses. In *Analyzing Neural Time Series Data: Theory and Practice* (pp. 15-16). The MIT Press.

Cohen, M. X. (2014). Introduction to the Physiological Bases of EEG. In *Analyzing Neural Time Series Data: Theory and Practice* (p. 51). The MIT Press.

EMOTIV. (2015). *EMOTIV EPOC+ 14 Channel Wireless EEG Headset*. Retrieved from Emotiv: https://www.emotiv.com/epoc/

Google Developers. (2018, October 26). *Distribution dashboard*. Retrieved from Android Developers: https://developer.android.com/about/dashboards

Google Developers. (2019). *Android Studio*. Retrieved from Android Developers: https://developer.android.com/studio

Google Developers. (2019). *Build a UI with Layout Editor*. Retrieved from Android Developers: https://developer.android.com/studio/write/layout-editor

Google Developers. (2019). *Data and file storage overview*. Retrieved from Android Developers: https://developer.android.com/guide/topics/data/data-storage#filesExternal

Google Developers. (2019). *Introduction to Activities*. Retrieved from Android Developers: https://developer.android.com/guide/components/activities/intro-activities

Google Developers. (2019). *Sending operations to multiple threads*. Retrieved from Android Developers: https://developer.android.com/training/multiple-threads/

Google Samples. (2018, February 16). *googlesamples/android-BluetoothLeGatt*. Retrieved from GitHub: https://github.com/googlesamples/android-BluetoothLeGatt

Hu, L., Mourauxd, A., Huc, Y., & Iannetti, G. (2010). A novel approach for enhancing the signal-to-noise ratio and detecting automatically event-related potentials (ERPs) in single trials. *NeuroImage*, 6.

Huawei Technologies Co. (2018, September). *HUAWEI*. Retrieved from MediaPad T5: https://consumer.huawei.com/en/tablets/mediapad-t5/

IKW, RKI & GCO. (2018, February 12). *Cognitive Computing Hackathon on Epidemiology*. Retrieved from Hack4Health: https://www.hack4health.de/

Ilardi, S. S., Atchley, R. A., Enloe, A., Kwasny, K., & Garratt, G. (2007). Disentangling Attentional Biases and Attentional Deficits in Depression: An Event-Related Potential P300 Analysis. *Cognitive Therapy and Research*, 175–187.

Jahoda, P. (2016, April 12). *Delay in plotting real time data*. Retrieved from GitHub: https://github.com/PhilJay/MPAndroidChart/issues/1676

Jahoda, P. (2019, February). *PhilJay/MPAndroidChart*. Retrieved from GitHub: https://github.com/PhilJay/MPAndroidChart

Jupyter Team. (2015). *The Jupyter Notebook*. Retrieved from Jupyter Notebook documentation: https://jupyter-notebook.readthedocs.io/en/stable/

Luck, S. J. (2014). *Order of processing steps*. Retrieved from ERP INFO: https://erpinfo.org/order-of-steps

MBT. (2013). *Smarting*. Retrieved from mBrainTrain: https://mbraintrain.com/smarting/

Monoprice. (2013). *Monoprice Enhanced Bass Hi-Fi Noise Isolating Earbuds Headphones*. Retrieved from Monoprice: https://www.monoprice.com/product?p_id=9927

MTU Physics. (2019). *Formula for frequency table*. Retrieved from Physics of Music Notes: http://pages.mtu.edu/~suits/NoteFreqCalcs.html

NeuroSky. (2015). *MindWave*. Retrieved from NeuroSky Store: https://store.neurosky.com/pages/mindwave

Niedermeyer, E., & Lopes Da Silva, F. (2005). Dynamics of EEGs as Signals of Neuronal Populations: Models and Theoretical Considerations. In *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields* (p. 85). Philadelphia: Lippincott Williams & Wilkins.

Niedermeyer, E., & Lopes Da Silva, F. (2005). Event-Related Potentials: Methodology and Quantification. In *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields* (p. 991). Philadelphia: Lippincott Williams & Wilkins.

Nieters, P., & Leugering, J. (2017, December 20). *pnieters/tflow_edge*. Retrieved from GitHub: https://github.com/pnieters/tflow_edge

Oracle. (2018). *SimpleDateFormat*. Retrieved from Java Platform Se 7: https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html

Passmark Software. (2019). *Android Benchmark Charts*. Retrieved from Passmark: https://www.androidbenchmark.net/passmark_chart.html

Picton, T. W., Hillyard, S. A., Krausz, H. ..., & Galambos, R. (1974). Human Auditory Evoked Potentials. I: Evaluation of Components. *Electroencephalography and Clinical Neurophysiology*, 3.

Python Sofware Foundation. (2019). *Python 3 documentation*. Retrieved from Python Documentation: https://docs.python.org/3/

Reimann, M. (2018). *Measuring vMMN, P300 and AEPs with a mobile EEG system.*

Rojas, A., Achziger, A., Garita, R., & Vidal De Palol, M. (2018, February 16). *StatefulMind/eegdroid*. Retrieved from GitHub: https://github.com/StatefulMind/eegdroid/tree/72894eb1d7cac4397c3be07e545aeccd1f1b3f80

Rojas, A., Achziger, A., Garita, R., Vidal De Palol, M., Fritsch, K., & Zerfowski, J. (2018, August 8). *StatefulMind/eegdroid*. Retrieved from GitHub: https://github.com/StatefulMind/eegdroid/tree/7252284aad1f879f91d2d9695d74261da0509d5a

Schalkamp, A.-K. (2018). *Analyzing event-related potentials in 8-channel EEG data using machine learning methods.*

Stackoverflow. (2019). *Developer Survey Results 2018*. Retrieved from Stackoverflow: https://insights.stackoverflow.com/survey/2018/#most-popular-technologies

StatCounter. (2019, January). *Mobile Operating System Market Share Worldwide*. Retrieved from StatCounter Global Stats: http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201501-201901

StatCounter. (2019, January). *Operating System Market Share Worldwide*. Retrieved from StatCounter Global Stats: http://gs.statcounter.com/os-market-share#monthly-201501-201901

Tutorialspoint. (2019). *Java.lang.System.currentTimeMillis() Method*. Retrieved from Tutorialspoint: https://www.tutorialspoint.com/java/lang/system_currenttimemillis.htm

Vidal De Palol, M. (2019, 03). *mvidaldp/mscthesis*. Retrieved from GitHub: https://github.com/mvidaldp/mscthesis

Vidal De Palol, M. (2019, March 1). *mvidaldp/pytonegen*. Retrieved from GitHub: https://github.com/mvidaldp/pytonegen

Vidal De Palol, M. (2019, March 1). *mvidaldp/Traumschreiber-mobileEEG*. Retrieved from GitHub: https://github.com/mvidaldp/Traumschreiber-mobileEEG

W. Hall III, J. (2015). Chapter 1: Introduction to Auditor Evoked Responses. In *Handbook of Auditory Evoked Responses* (pp. 23-65). Missy Hall.

W. Hall III, J. (2015). Chapter 10: Auditory Middle Latency Response (AMLR). In *Handbook of Auditory Evoked Responses* (p. 552). Missy Hall.

W. Hall III, J. (2015). Chapter 11: Auditory Late Responses (ALRs). In *Handbook of Auditory Evoked Responses* (p. 638). Missy Hall.

Xiaomi. (2017, September). *Mi Global Home*. Retrieved from Mi A1: https://www.mi.com/global/mi-a1/

Zerfowski, J. (2018, January 12). *StatefulMind/eegdroid*. Retrieved from GitHub: https://github.com/StatefulMind/eegdroid/tree/0a1049db127f91eae590b1f7c4ae055145be23e9

Appendix

A.1 Consent Form

Teilnahme an der Studie: Traumschreiber als EEG-System

Hiermit bestätige ich, ______________________________ geboren am ________________, dass ich dieses Dokument vollständig gelesen und verstanden habe. Mir ist bewusst, dass die Teilnahme an der Studie freiwillig ist und jederzeit ohne nachteilige Auswirkungen, oder Nennung von Gründen abgebrochen werden kann. Mir ist bewusst, dass die Speicherung der gemessenen Daten in anonymisierter Form erfolgt und bei möglichen Veröffentlichungen keine Rückschlüsse auf meine Person möglich sind. Ich wurde informiert, dass ich jederzeit meine Einwilligung zur Speicherung und Nutzung meiner Daten entziehen kann. In diesem Fall werden die Daten unverzüglich gelöscht.

Die gesammelten Daten werden folgenden Personen zur Verfügung gestellt: Marc Vidal De Palol und der Neuroinformatik-Gruppe der Universität Osnabrück.

Ich bin damit einverstanden, dass während des Experiments Messungen mit einem Elektroenzephalogramm (EEG) gemacht werden. Dabei werden vom Gehirn erzeugte Spannungen auf der Kopfoberfläche gemessen. Ich stimme zu, dass hierzu der Traumschreiber verwendet werden darf. Der Traumschreiber wurde von Johannes Leugering und Kristoffer Appel zur Erhebung von Daten während des Schlafs entwickelt. Diese Studie soll dazu dienen herauszufinden, ob der Traumschreiber auch als mobiles EEG-System verwendet werden kann.

Ich bestätige, dass ich keine neurologischen Erkrankungen habe.

Ich bestätige, dass ich die obige Erklärung vollständig gelesen und verstanden habe und ihr zustimme. Alle eventuellen Fragen wurden von den Versuchsleiterinnen beantwortet.

____

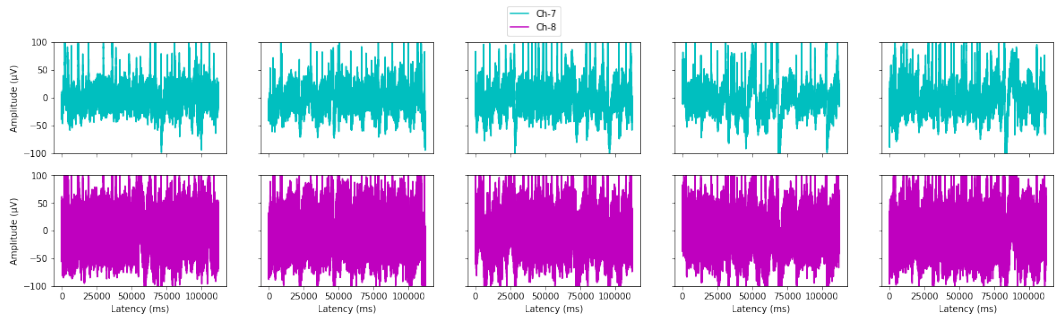
Datum, Unterschrift

mvidaldepalo@uos.de

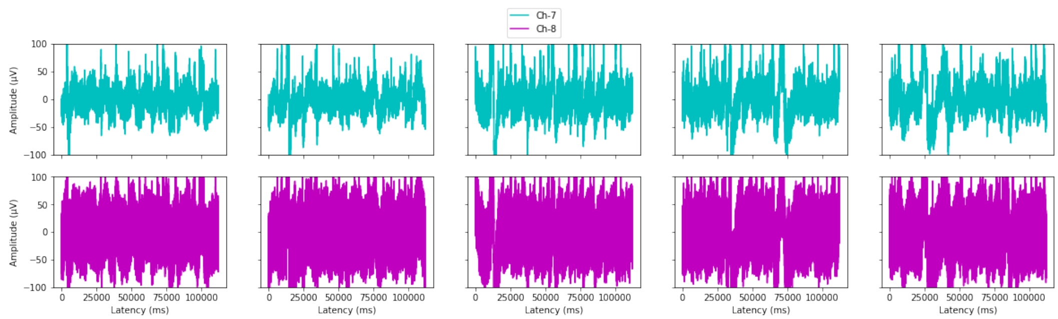
A.2 Preliminary Plots

Trials subject 1

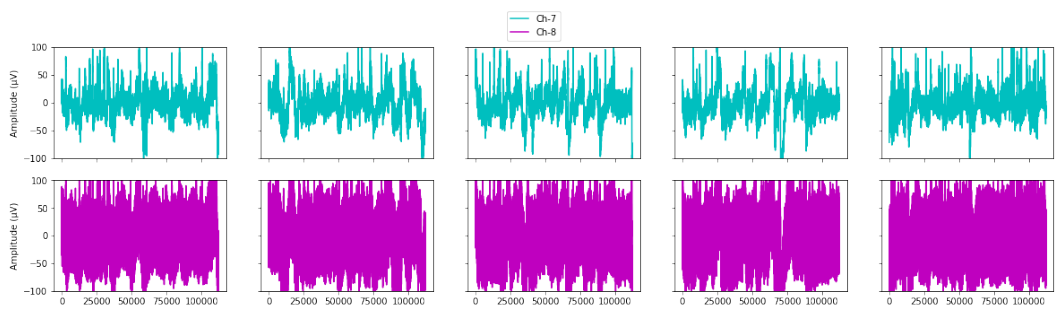
Stimulus A4 (440Hz):



Stimulus A5 (880Hz):

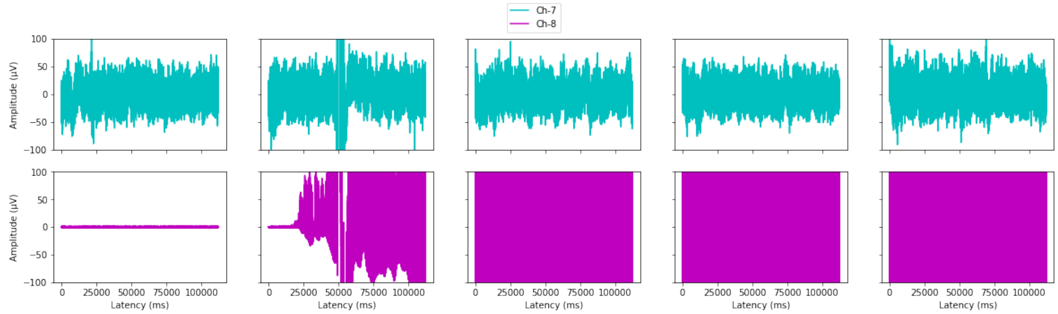


Stimuli C4-B6 (261.63 Hz - 1975.53 Hz):

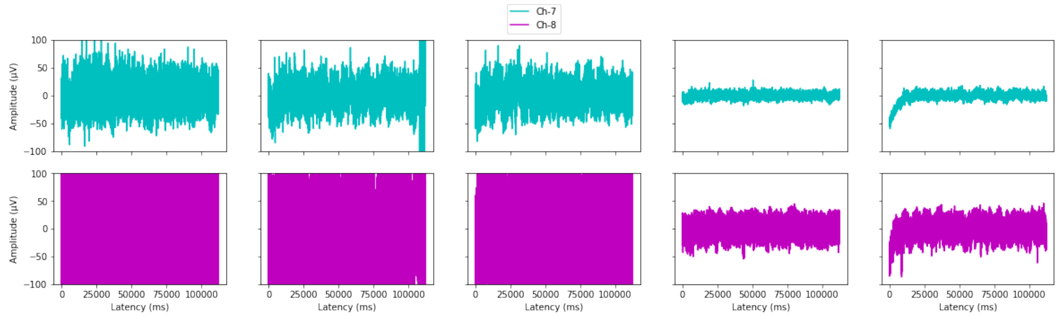


Trials subject 2

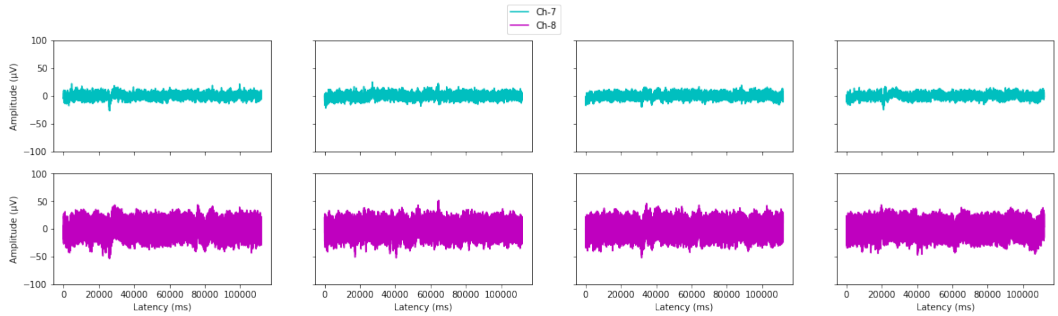
Stimulus A4 (440Hz):



Stimulus A5 (880Hz):

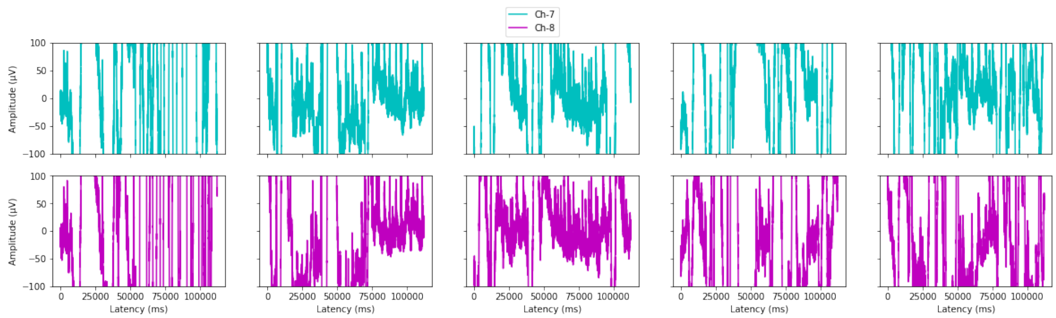


Stimuli C4-B6 (261.63 Hz - 1975.53 Hz):

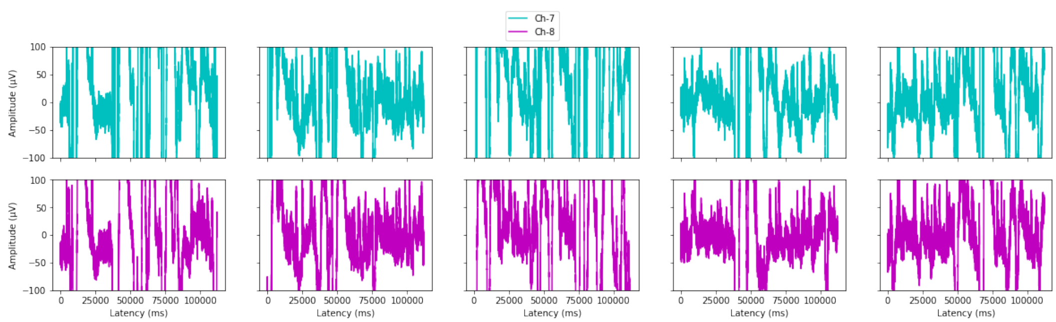


Trials subject 3

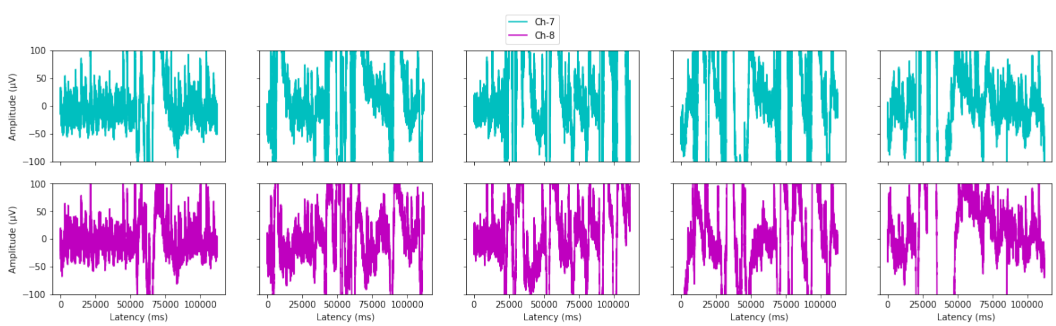
Stimulus A4 (440Hz):



Stimulus A5 (880Hz):

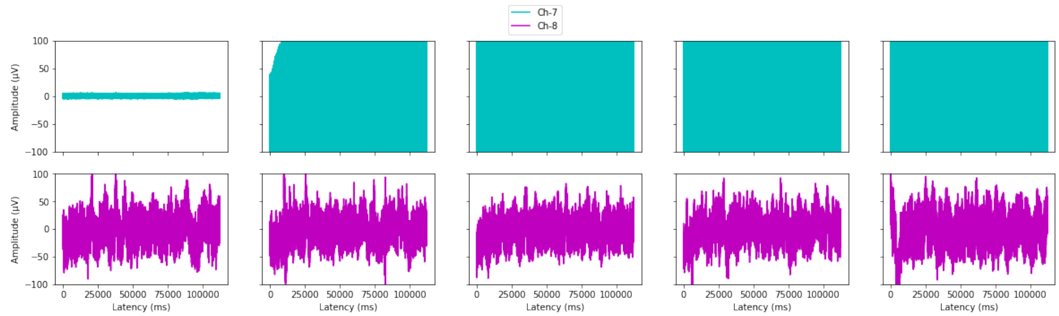


Stimuli C4-B6 (261.63 Hz - 1975.53 Hz):

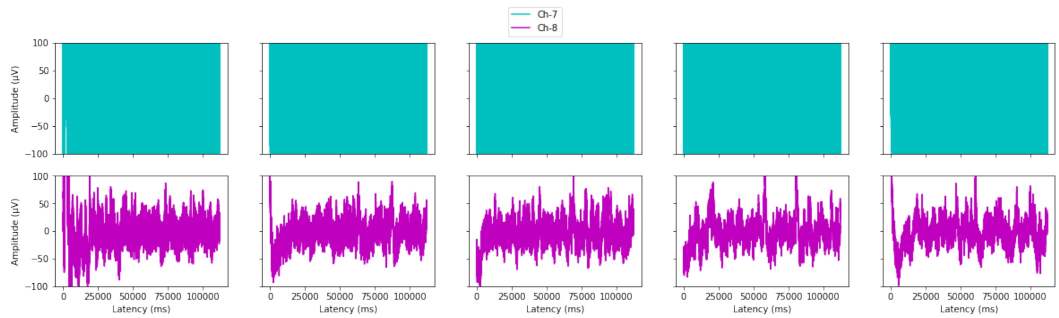


Trials subject 4

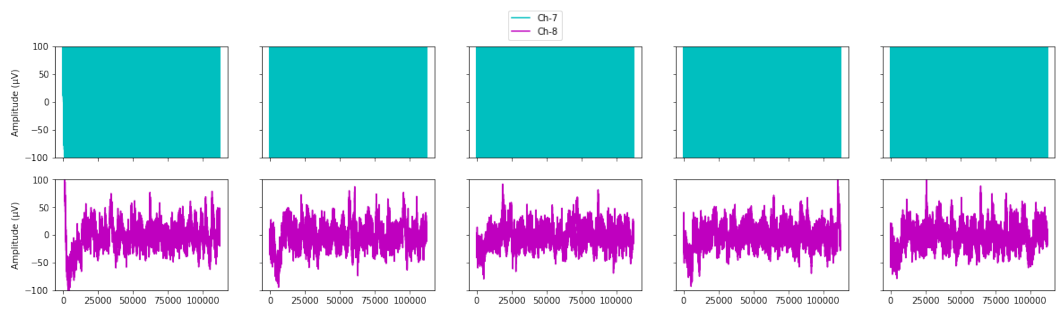
Stimulus A4 (440Hz):



Stimulus A5 (880Hz):



Stimuli C4-B6 (261.63 Hz - 1975.53 Hz):



A.3 Descriptive statistics

Subject 1

Channel	Stimulus	Trial	mean	std	min	25%	50%	75%	max
7	*A4*	1	0.516	23.6	-99.097	-13.696	-1.208	11.884	139.38
		2	-0.668	23.437	-95.068	-15.106	-1.007	12.689	146.429
		3	-0.403	31.98	-102.319	-17.523	-2.417	12.689	290.039
		4	0.35	34.532	-123.468	-18.53	-2.82	13.495	240.289
		5	-1.323	32.429	-167.78	-20.142	-4.834	12.488	225.183
	A5	1	-1.052	22.776	-116.821	-15.509	-2.417	11.481	133.337
		2	-0.339	28.114	-143.005	-14.502	-1.813	11.884	220.551
		3	-0.658	34.248	-202.423	-17.322	-3.223	12.085	168.585
		4	-0.533	31.651	-121.655	-18.732	-2.216	15.71	157.306
		5	-0.548	31.299	-111.987	-19.537	-2.618	15.308	138.574
	C4-B6	1	0.61	25.17	-110.779	-13.696	-0.604	12.891	239.282
		2	-0.88	28.427	-123.267	-18.732	-1.41	14.905	112.994
		3	-0.65	27.808	-128.906	-16.718	-1.208	14.301	109.973
		4	-1.326	30.36	-146.631	-16.718	-2.82	12.085	126.288
		5	-0.955	26.032	-113.599	-16.113	-2.82	10.474	196.381

Channel	Stimulus	Trial	mean	std	min	25%	50%	75%	max
8	*A4*	1	0.189	42.559	-134.747	-35.046	0.504	32.629	187.921
		2	-1.426	41.598	-140.991	-33.838	-1.611	31.018	162.946
		3	-0.482	44.508	-134.344	-32.428	-2.014	28.198	305.347
		4	0.446	45.855	-128.906	-33.032	-1.41	30.414	272.919
		5	-1.046	45.46	-203.43	-35.651	-1.41	30.212	243.512
	A5	1	-1.167	40.884	-152.271	-33.636	-1.007	30.817	169.189
		2	-1.03	47.896	-180.67	-39.679	-1.208	35.852	268.085
		3	-0.976	48.239	-232.031	-37.463	-1.208	33.636	191.949
		4	-0.409	46.683	-149.652	-36.255	-0.604	33.435	175.232
		5	-0.613	46.475	-135.15	-37.061	-0.806	33.435	159.723
	C4-B6	1	0.431	45.52	-151.666	-36.859	0.806	36.053	205.847
		2	-0.963	51.451	-175.836	-41.29	-0.604	39.075	170.398
		3	-1.014	51.111	-169.592	-42.7	-0.806	39.478	163.348
		4	-1.123	54.407	-196.985	-45.319	-0.201	41.29	188.727
		5	-1.254	51.46	-173.016	-44.714	-1.007	40.283	270.703

Subject 2

Channel	Stimulus	Trial	mean	std	min	25%	50%	75%	max
7	*A4*	1	-1.366	20.234	-89.429	-14.905	-1.007	12.488	116.62
		2	3.903	68.169	-411.896	-14.905	1.41	17.725	412.5
		3	-0.761	20.452	-76.135	-14.905	-1.007	13.293	95.27
		4	-0.186	19.432	-76.74	-13.696	-0.403	13.092	71.1
		5	0.044	20.799	-91.241	-14.301	-0.403	13.293	112.793
	A5	1	-0.287	22.333	-91.443	-15.71	-0.604	14.703	109.57
		2	7.151	69.532	-411.896	-15.106	-1.007	13.495	412.5
		3	-0.949	20.002	-82.983	-14.502	-1.41	12.286	90.033
		4	-0.459	4.638	-18.329	-3.625	-0.403	2.618	27.594
		5	-2.489	8.965	-59.821	-4.633	-1.007	2.417	17.725
	C4-B6	1	-0.302	4.653	-26.99	-3.424	-0.201	2.82	21.954
		2	-0.446	4.736	-22.156	-3.625	-0.403	2.82	24.976
		3	failed	failed	failed	failed	failed	failed	failed
		4	-0.449	4.554	-20.343	-3.424	-0.403	2.618	19.135
		5	-0.366	4.602	-25.378	-3.424	-0.403	2.82	16.718

Channel	Stimulus	Trial	mean	std	min	25%	50%	75%	max
8	*A4*	1	-0.096	0.49	-1.813	-0.403	0	0.201	1.813
		2	0.01	102.921	-412.5	-41.693	-0.806	37.866	412.299
		3	-0.178	107.784	-247.137	-101.514	-0.302	101.312	273.322
		4	0.003	121.113	-291.852	-115.814	0.201	115.613	250.763
		5	-1.805	119.023	-265.466	-113.145	-2.014	109.117	291.046
	A5	1	-0.413	137.729	-292.255	-133.136	-0.604	131.726	279.968
		2	-2.03	93.927	-412.5	-80.566	-2.417	77.495	412.299
		3	-0.975	85.07	-220.551	-76.135	-1.41	74.725	202.625
		4	-0.754	18.499	-54.382	-17.07	-0.806	15.71	44.916
		5	-1.799	19.259	-86.81	-16.516	-1.611	14.099	45.721
	C4-B6	1	-0.477	17.678	-53.577	-15.912	-0.604	15.106	43.506
		2	-0.51	16.661	-52.57	-14.905	-0.403	14.099	51.16
		3	failed	failed	failed	failed	failed	failed	failed
		4	-0.376	17.179	-52.368	-15.509	-0.403	14.703	45.923

		5	-0.528	17.225	-47.333	-15.71	-0.604	14.703	43.304

Subject 3

Channel	Stimulus	Trial	mean	std	min	25%	50%	75%	max
7	*A4*	1	25.178	428.66	-1259.253	-172.412	34.644	238.477	1363.989
		2	17.689	350.894	-1649.194	-41.895	-5.64	45.923	1650
		3	-2.069	268.824	-1025.61	-31.421	16.113	139.38	470.508
		4	-5.112	320.063	-1157.739	-69.287	58.008	199.805	547.046
		5	-1.637	228.529	-1322.095	-24.976	23.364	106.348	403.638
	A5	1	93.428	195.672	-412.299	-18.933	51.361	227.147	412.5
		2	30.339	126.267	-412.097	-23.969	8.057	58.813	412.5
		3	114.905	188.053	-412.299	6.042	86.005	266.826	412.5
		4	36.409	127.846	-412.097	-16.113	9.265	44.11	412.5
		5	22.97	119.9	-412.299	-22.76	8.258	56.396	412.5
	C4-B6	1	-1.694	64.634	-391.351	-22.357	-5.237	16.516	257.611
		2	36.494	134.922	-412.299	-17.322	17.12	92.047	412.5
		3	72.102	169.56	-412.299	-17.322	18.732	156.903	412.5
		4	66.376	172.904	-412.299	-18.329	34.845	164.758	412.5
		5	31.538	140.264	-412.097	-20.947	7.654	43.304	412.5

Channel	Stimulus	Trial	mean	std	min	25%	50%	75%	max
8	*A4*	1	21.312	527.873	-1650	-174.023	26.587	333.545	1590.381
		2	-0.515	266.852	-1017.554	-101.514	-13.696	33.838	1105.371
		3	-2.498	110.731	-477.759	-24.17	2.417	47.534	249.756
		4	-3.396	284.274	-1650	-141.797	-21.753	83.789	974.048
		5	-1.213	190.547	-796.802	-102.319	-34.644	93.457	863.672
	A5	1	-3.447	171.284	-412.5	-49.548	-9.869	65.863	412.299
		2	-1.872	93.698	-412.5	-30.615	-3.021	31.219	293.262
		3	2.677	148.078	-412.5	-42.499	17.523	99.298	355.298
		4	4.712	99.017	-412.5	-22.559	-2.618	21.552	412.299
		5	-0.742	84.826	-412.5	-27.191	-0.403	36.658	276.544
	C4-B6	1	-1.517	44.787	-277.551	-19.135	-3.424	16.315	166.168
		2	-0.42	84.004	-412.5	-32.227	-1.41	36.003	281.378
		3	2.716	102.649	-412.5	-32.227	2.417	55.188	320.654

		4	-1.904	103.804	-372.217	-45.52	-4.028	60.828	265.869
		5	2.074	167.163	-412.5	-21.753	16.315	59.619	412.299

Subject 4

Channel	Stimulus	Trial	mean	std	min	25%	50%	75%	max
7	*A4*	1	0.322	3.193	-5.64	-2.82	-0.403	3.625	6.244
		2	-1.824	133.62	-310.584	-128.453	4.028	115.21	271.106
		3	-1.328	155.485	-290.039	-153.479	-1.208	149.249	276.746
		4	-0.238	171.191	-362.146	-166.168	-0.604	165.161	326.898
		5	-0.77	227.302	-411.493	-220.752	1.208	214.709	412.5
	A5	1	1.032	117.87	-411.291	-106.75	1.208	106.348	412.5
		2	-2.391	116.524	-233.844	-112.994	-2.618	108.563	268.488
		3	-1.878	118.366	-252.777	-115.21	-1.41	110.98	234.65
		4	-2.651	122.647	-387.323	-117.023	-2.417	113.397	261.841
		5	-0.411	123.324	-253.986	-118.03	0	115.411	407.465
	C4-B6	1	-2.099	126.325	-274.731	-121.655	-1.007	116.62	405.249
		2	-2.012	127.779	-260.028	-124.072	-1.41	120.044	281.982
		3	-1.418	129.049	-248.547	-125.482	-1.007	122.662	251.166
		4	-1.762	128.896	-260.229	-125.684	-0.403	122.612	254.59
		5	-2.303	130.675	-267.883	-127.698	-2.014	122.26	265.265

Channel	Stimulus	Trial	mean	std	min	25%	50%	75%	max
8	*A4*	1	-1.838	26.61	-90.637	-20.947	-2.618	16.113	100.104
		2	-2.032	25.736	-113.397	-19.739	-2.417	15.106	127.094
		3	-2.334	23.368	-89.227	-19.336	-2.216	14.301	82.178
		4	-1.612	25.724	-133.337	-19.336	-1.41	16.113	92.249
		5	-3.251	31.573	-155.493	-24.774	-2.216	18.933	114.203
	A5	1	0.294	47.514	-309.979	-18.53	-1.813	14.905	394.977
		2	-2.627	22.481	-93.256	-16.919	-2.417	11.481	113.196
		3	-2.809	23.171	-108.563	-16.516	-3.021	11.078	121.857
		4	-2.69	25.642	-83.386	-18.732	-4.431	10.876	121.454
		5	-1.815	28.907	-98.895	-18.732	-3.223	12.488	262.646
	C4-B6	1	-1.928	30.262	-120.85	-16.718	-3.223	11.279	271.71
		2	-2.758	21.725	-94.666	-15.308	-2.82	9.869	86.81

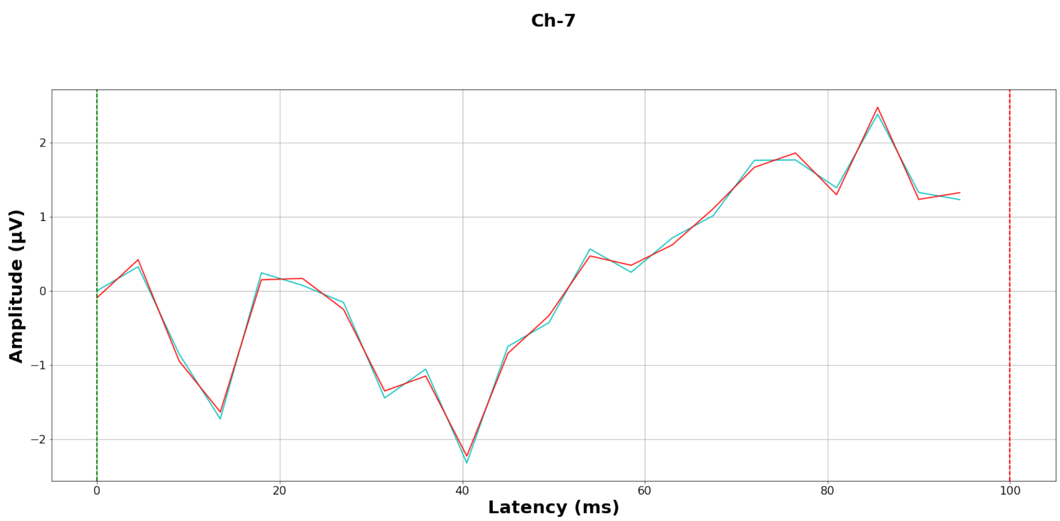
		3	-2.316	18.906	-79.559	-15.106	-3.021	9.668	91.443
		4	-2.451	19.968	-93.054	-14.502	-2.618	9.467	104.736
		5	-2.384	20.695	-78.955	-15.912	-2.82	10.675	100.305

A.4 Final Plots

Subject 1

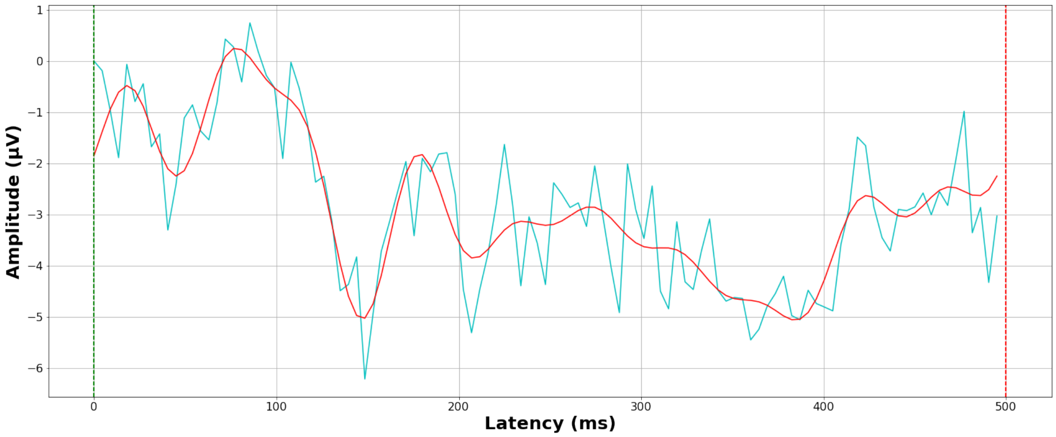
A4

AMLR



ALR

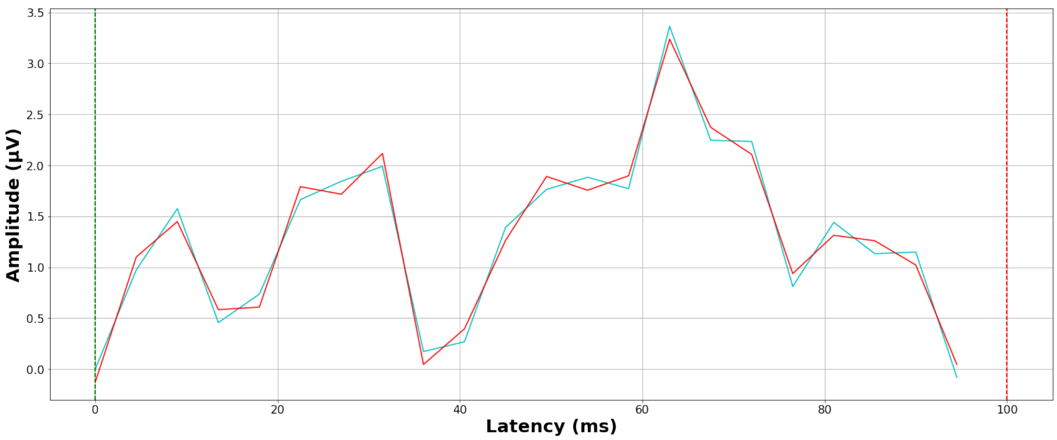
Ch-7



A5

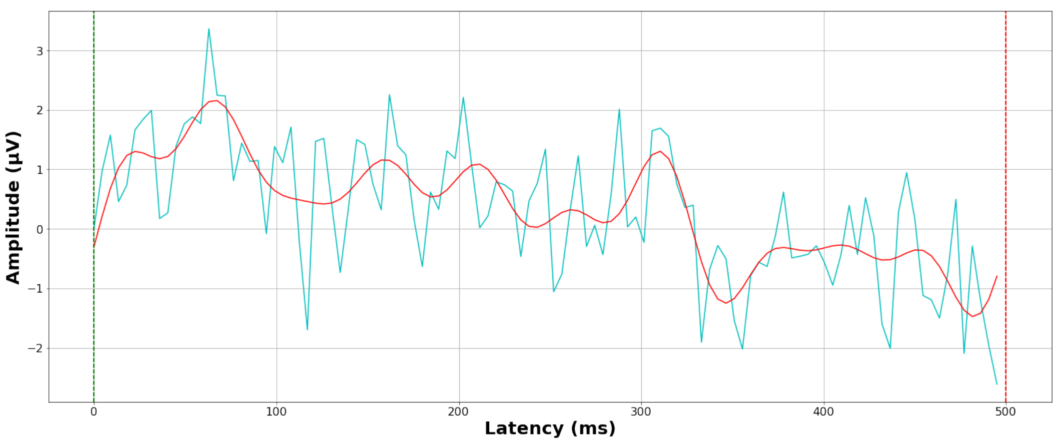
AMLR

Ch-7



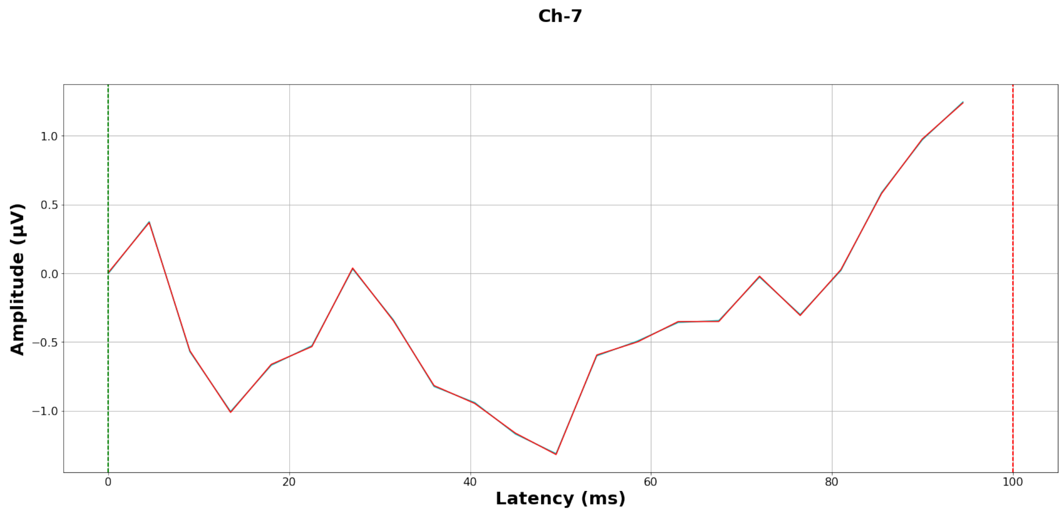
ALR

Ch-7

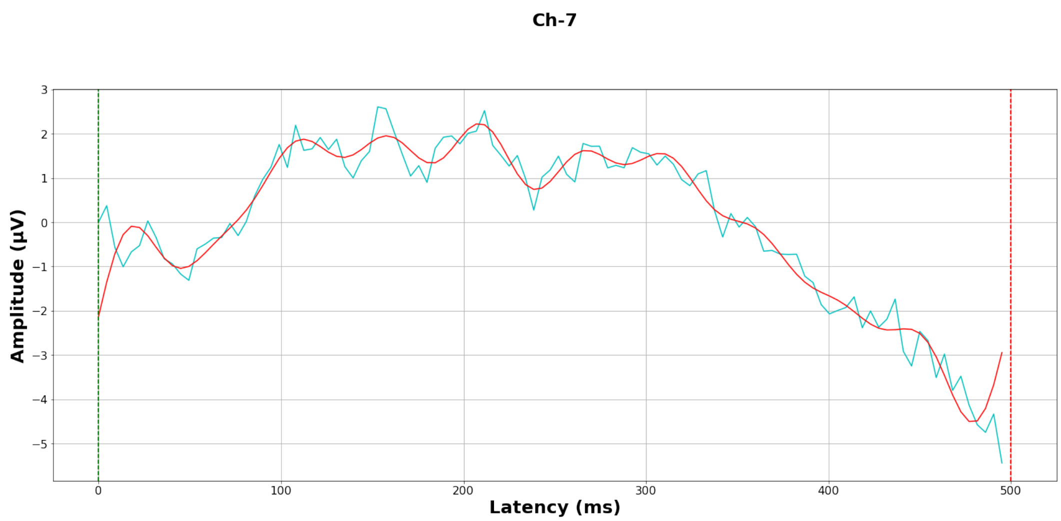


C4-B6

AMLR



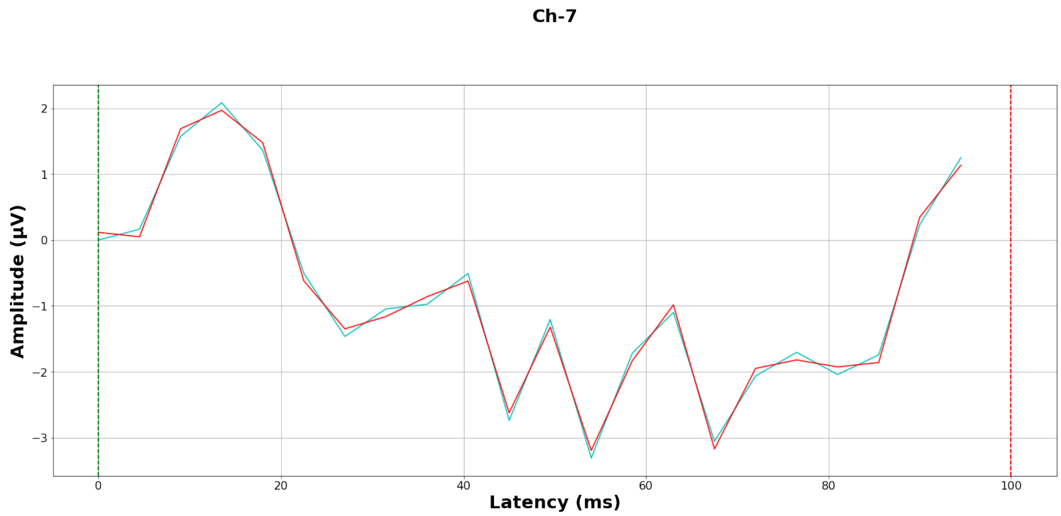
ALR



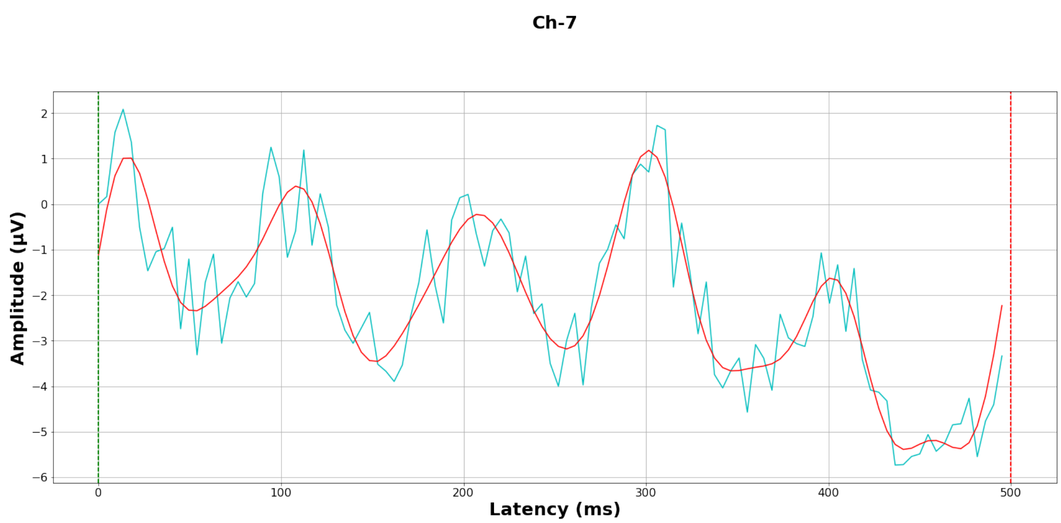
Subject 2

A4

AMLR

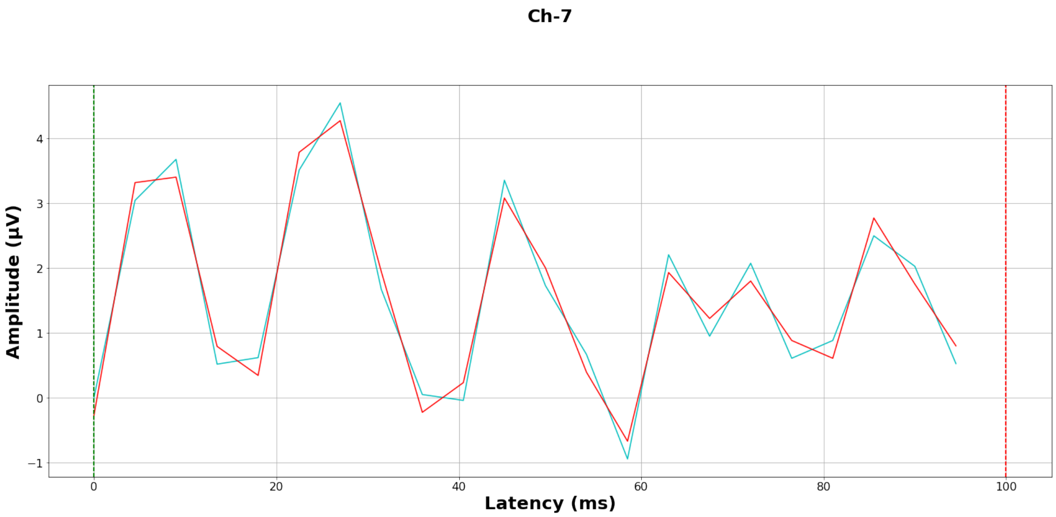


ALR

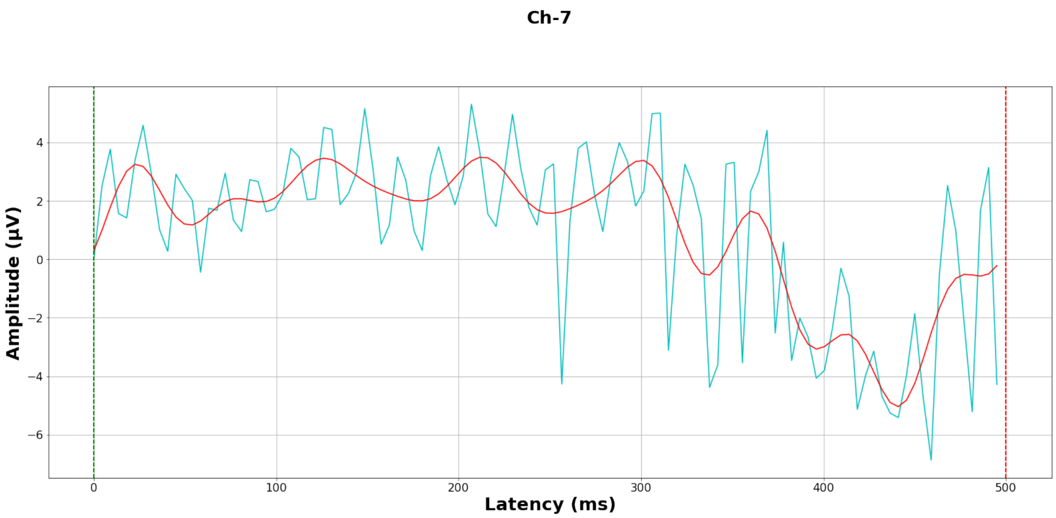


A5

AMLR

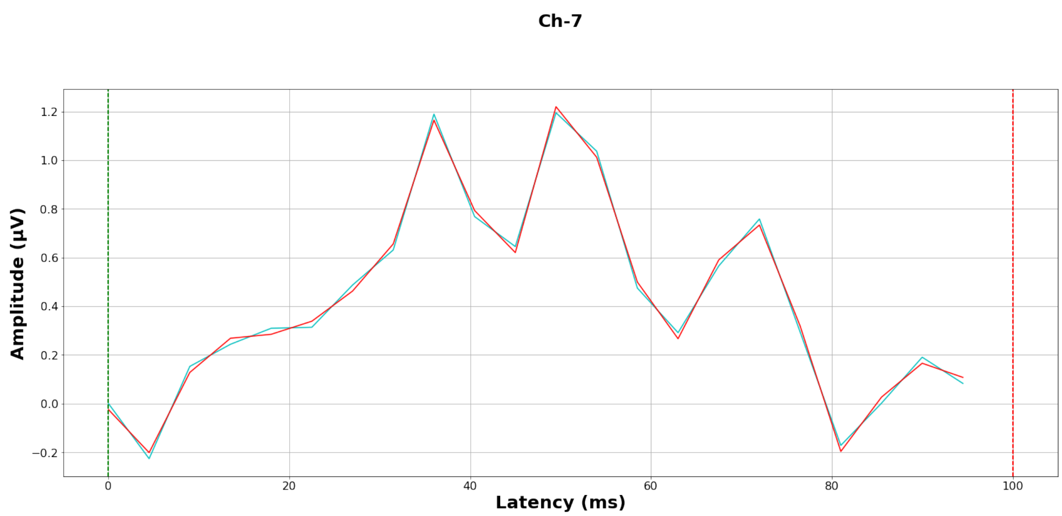


ALR

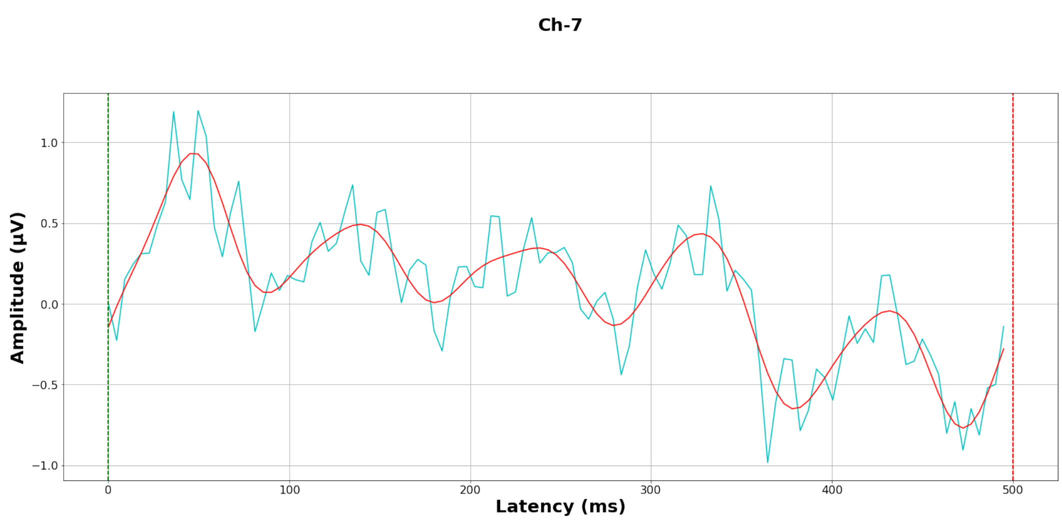


C4-B6

AMLR



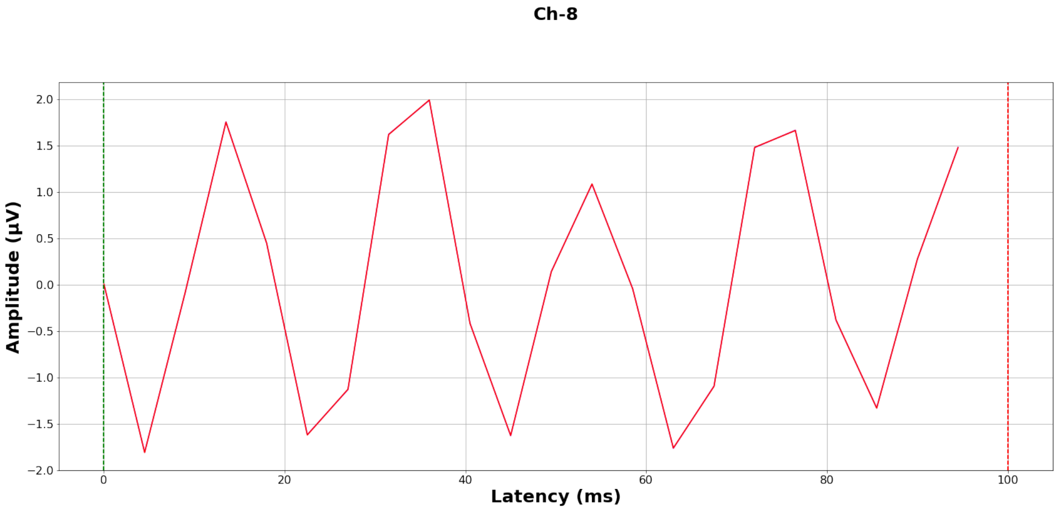
ALR



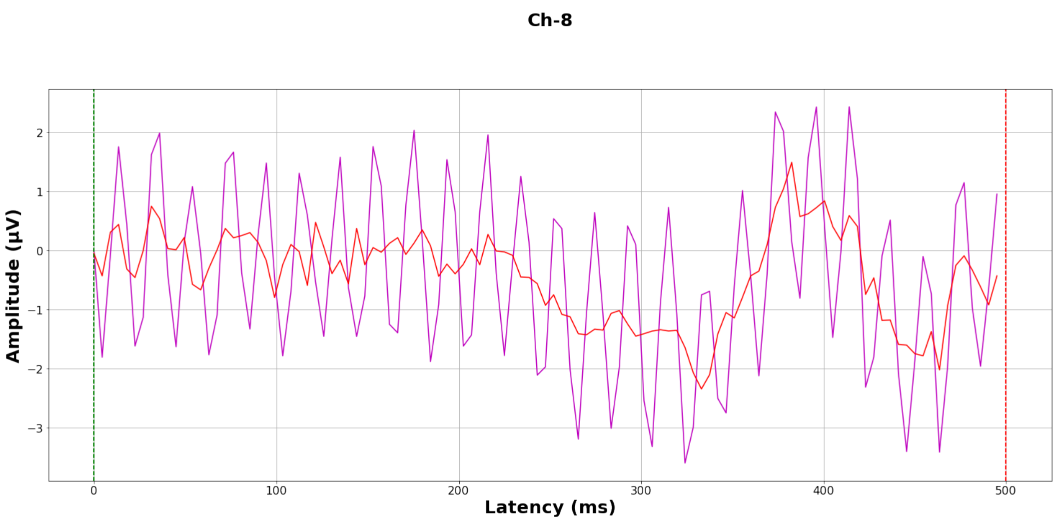
Subject 4

A4

AMLR

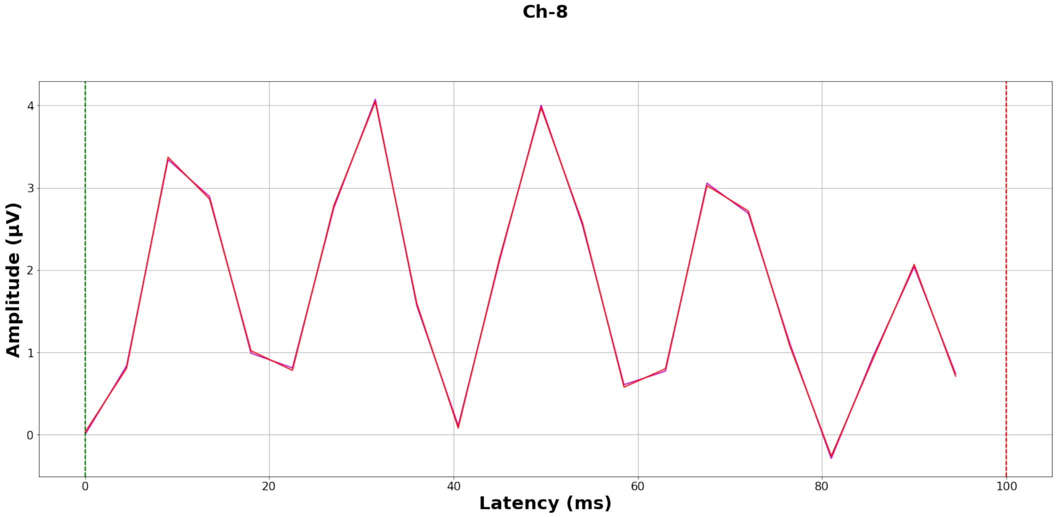


ALR



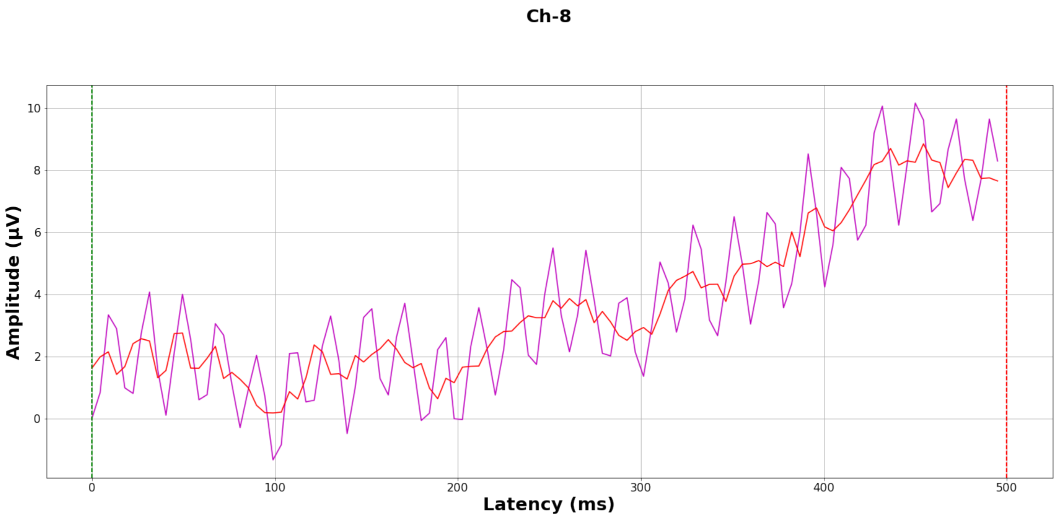
A5

AMLR



Ch-8
Amplitude (µV)
Latency (ms)
0
1
2
3
4
0
20
40
60
80
100

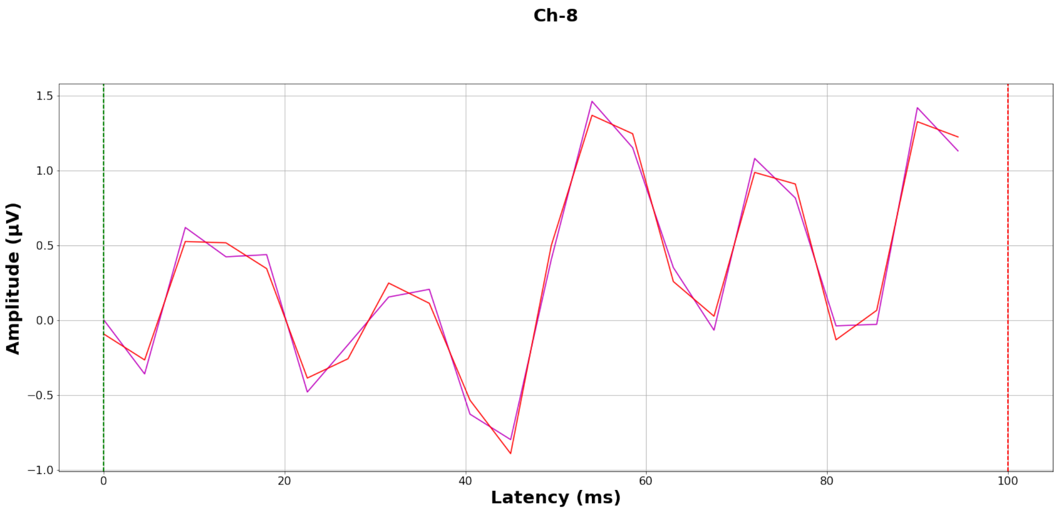
ALR



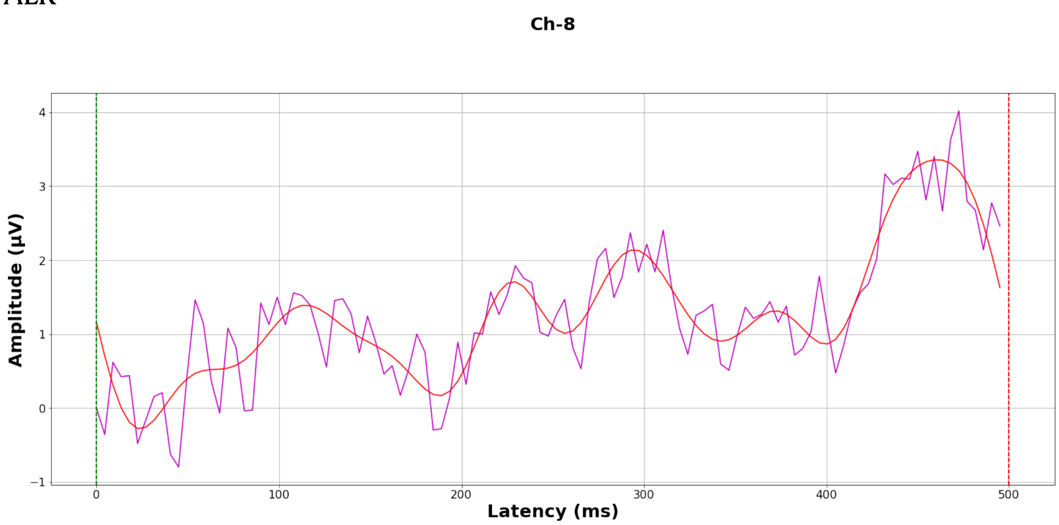
Ch-8
Amplitude (µV)
Latency (ms)
0
2
4
6
8
10
0
100
200
300
400
500

C4-B6

AMLR



ALR



Declaration of Authorship

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.

____

Osnabrück, 12.03.2019