

# Anima

## A Brain-Computer Interface

Sebastien Romane Charles

Eric Mendoza-Conner

Darwin Huang

Professor Toby Cumberbatch

Senior Projects

May 2017

## Abstract

Anima is a non-invasive electroencephalogram (EEG) that provides a brain-computer interface (BCI) headset. Anima measures and filters brain waves using custom designed active electrodes. Anima's artificial neural network uses features extracted from selected brain waves to move a computer cursor in real time. This paper outlines the theory, design, and results behind the construction of such a device.

## Table of Contents

<b>Introduction</b>	<b>4</b>
<b>Background</b>	<b>5</b>
2.1. The Human Brain	5
2.2. The Human Body	5
2.3. Brain-Computer Interfaces	6
2.4. Active Sensor	6
2.5. Brainwaves	7
2.6. Machine Learning	9
2.7. Graphical User Interface	10
<b>Overview</b>	<b>11</b>
<b>Electrodes</b>	<b>12</b>
4.1. Electrode Choices	12
4.2. The Flex Electrode	14
<b>Filters</b>	<b>16</b>
5.1. Notch	16
5.2. Low-Pass	19
5.3. High-Pass	20
<b>Instrumentation Amplifier</b>	<b>22</b>
6.1. Implementation	22
<b>Combined Active Sensor</b>	<b>23</b>
<b>Microcontroller</b>	<b>25</b>
8.1 Microcontroller Selection	25
<b>Digital Signal Processing</b>	<b>26</b>
9.1. Signal Separation	26
9.2. Localization	26
9.3. Feature Extraction	27
<b>Machine Learning - Artificial Neural Network</b>	<b>28</b>
10.1. Artificial Neural Network Background	28
10.2. Hidden Layer Size Determination	30

10.3. Backpropagation	37
<b>Graphical User Interface</b>	<b>40</b>
11.1. Implementation	40
<b>Experimental Procedure and Results</b>	<b>41</b>
<b>Conclusion</b>	<b>43</b>
<b>Works Cited</b>	<b>44</b>

## 1. Introduction

The way humans currently interact with computers range from computer mice to touchscreens, and almost always involve many unnecessary steps which reduce immersion for the user. In such methods, a user's intent must be translated into muscle movement before the computer can understand it. It is envisioned that one day a user may simply think about how they wish to interact with a computer, and directly see that interaction affect the computer in real-time, without the need for any physical muscle movement of the user. Removing this physical layer by directly reading the brain's intent to interact with the computer helps achieve this goal. Anima proposes to bridge the gap between the brain and computer input control by providing a platform to read brainwaves directly in order to move a computer cursor. Anima employs the electroencephalogram in order to read such brain waves directly from a user's scalp.

Electroencephalography (EEG), throughout the 20<sup>th</sup> century, had an important place in the diagnosis of neurological conditions like epilepsy, brain death, and coma. Recent technological advancements have made EEGs more portable, inexpensive, and accurate, so that their use outside of medical diagnosis has become feasible. Such EEGs have found recent use as a brain-machine interface (BMI), mapping various brain wave patterns to corresponding control signals to control systems like video games [1] and quadcopters [2]. As the EEG is the basic building block that provides the control for these various systems, the development of an inexpensive portable EEG headset is crucial for further exploration into the mind, and its capabilities in operating various interfaces.

While there are currently EEGs which are capable of providing input control using brain waves, current EEGs are expensive and often rely on auxiliary signals, such as eye tracking, in order to supplement their function as input devices. Anima is a unique in that it focuses on being a relatively inexpensive EEG input device capable of solely sending commands to a computer, without assistance from any other components.

## 2. Background

### 2.1. The Human Brain

The design of an EEG is informed by the operation of the brain, the surrounding human body, and the expected environment surrounding a human subject. Before creating the design of an EEG headset, whose function is to map different brainwave patterns to intentional commands supplied via the thought of a user, Anima had to understand some basic inner workings of the brain, and consider artifacts that may be generated by the surrounding human body and environment.

A neuron is a specialized cell which transmits nerve impulses throughout the human body in the form of very weak electrical current. The average human brain consists of approximately 100 billion neurons, which create human thoughts as a result of their communication between one another. As long as a human is alive, the neurons in their body, particularly the ones in the brain, are continuously firing. The aggregate electrical activity of these neurons induce changes in the electric fields around them that can be measured as voltage potentials using an electrode. These neurons are tightly packed, and separated from the surface of the scalp by a relatively conductive cerebrospinal fluid which is surrounded by the skull. Due to the relatively high transmittance of the cerebrospinal fluid, neurons' effects on the surrounding electric field are easily propagated throughout the skull. However, the skull's high impedance strongly attenuates these neural potentials as they pass through the skull and on towards the scalp. The cerebrospinal fluid makes it difficult to pinpoint the effect of individual neurons or small groups of neurons, and the high impedance of the skull and inconvenient contact surface of typical human scalps make the neural signals available on the scalp very weak in magnitude.

### 2.2. The Human Body

Compounded with such challenges are the effects of the human body and environments. The human body has strong bio-amplification properties, so that signals from the brain become stronger when they reach their destination muscles. Therefore, events like heart pumping, eyelid blinking, and muscle movements can create unwanted voltage potentials of magnitudes several times higher than that of the brain after attenuation through the skull, when measured at the scalp. Such unwanted voltage potentials, hereafter referred to as artifacts, must be filtered out through various means so that they do not affect or overwhelm the measurement of signals from the brain.

In addition to artifacts from muscles, it is expected that some subjects may reside in locations similar to the Cooper Union labs, where noise from the power grid may exist. The scale of the 60Hz power grid noise induced by the body [3] is orders of magnitude above brain signals Anima reads from the scalp, so in order to prevent that large amplitude signal from being amplified with the brainwaves, a filter at that frequency must be designed. While a filter cannot completely remove the 60 Hz noise, they reduce it enough that the brain signals can be recovered on the computer.

### 2.3. Brain-Computer Interfaces

The term Brain-Computer Interface is usually used to describe any external device with a direct communication pathway to the brain. Those devices have been used in the past mostly to monitor the brain activity of patients or to experiment on animals. The most popular non-invasive method to read data off the human brain is by using an electroencephalogram (EEG). An EEG is a device that measures electrical potentials from a human scalp as a mean of measuring cerebral activity. The use of this technique for day to day applications has received much attention in the past five years. Multiple BCI products implemented via EEGs- such as the OpenBCI and Emotiv- have been released, allowing users to monitor their brain activity or use them to perform simple tasks. Anima is unique in that it focuses on being an input device capable of solely sending commands to a computer, without assistance from any other components.

### 2.4. Active Sensor

Many EEG applications use passive sensors to measure voltage potentials on humans' scalps. Electrodes are metallic probes that, in the context of this project, are passive sensors. The signal that electrodes pick up is very noisy to the point where brain waves are occulted. Applying a polishing powder and an adhesive, conductive gel to a subject's scalp reduces noise. This process, however, is cumbersome and irritates the user's scalp.

An active sensor avoids this irritation by filtering the signal as it comes out of the electrode. The active sensor amplifies the signal a little bit, filters out the noise and then amplifies the brainwaves even more. This removes the need for any uncomfortable preparation of the skin and results in a signal that is less noisy than a passive sensor could ever give you.

## 2.5. Brainwaves

Signals from the brain are propagated as “brainwaves”, or collections of periodic neural impulses, often categorized by their frequency. Those brainwaves relate to different thought processes or mental states. Figure 1 provides a visual representation of the aggregated brain waves as measured by Anima and the table below gives a more complete explanation of what the different brainwaves represent.

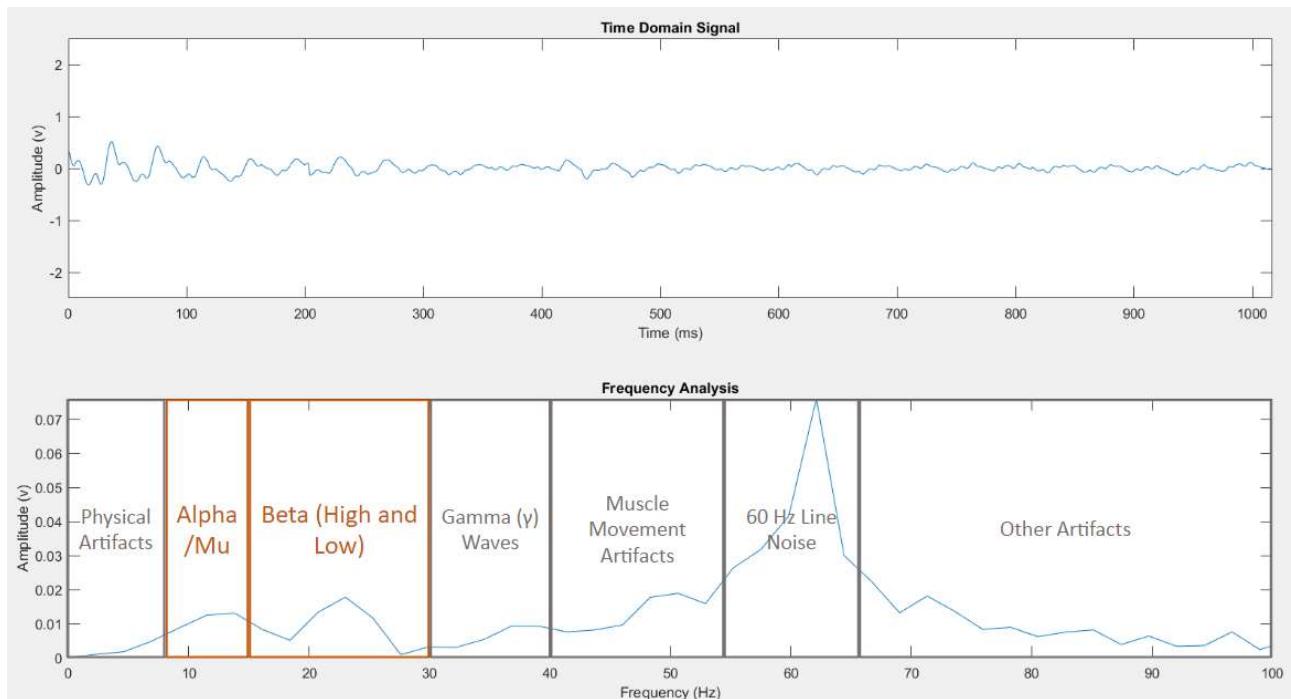


Figure 1: Brainwave Readings Measured By Anima

Brain wave	Frequency range	Mental process
Alpha	8-15 Hz	Associated with relaxation, reflection, and repression as well. Closing the eyes can often cause spikes in alpha waves, and therefore often serve as strong signals for verifications of EEGs.
Beta	16-31 Hz	Often occur during active thinking, focus, and stress.

Gamma	>32 Hz	Displayed by the somatosensory cortex. They often increase in amplitude when several senses are in use, or for short-term memory matching of visual, auditory, or tactile sensations.
Delta	<4 Hz	They are relatively high-amplitude, and prevalent in slow-wave sleep as well as continuous-attention tasks.
Theta	4-7 Hz	Occurs when drowsy, idling, or actively trying to repress responses or actions
Mu	8-13 Hz	Mu waves are suppressed when the subject does a motor action or thinks about a motor action.

Not all above waves are important to this project and being able to single out useful brain waves assists with both the handling of the signal and the placement of the electrodes. Delta waves only occur during sleep or tasks which require long periods of attention. The user of the Anima headset will be anticipated to be awake when using it and continuous attention is too much of burden to require for the use of the headset. Theta waves only occur in periods of drowsiness and idling which also does not fit the expected use of the Anima headset. For all those reasons, delta and theta waves are out of scope of this project.

Of the remaining brainwaves, Mu waves is the most important. Mu wave amplitude visibly decreases when the subject either does a motor action, thinks about a motor action or sees someone do a motor action. It has been observed that, given visual feedback, subjects learn how to control the amplitude of their mu waves fairly easily. Mu waves are highly localized and, if sensors are accurate enough, individual patterns can be used to differentiate between parts of his body the subject is thinking about moving. Figure 2 shows where in the brain mu waves are generated and which body part is linked to which section of the motor cortex.

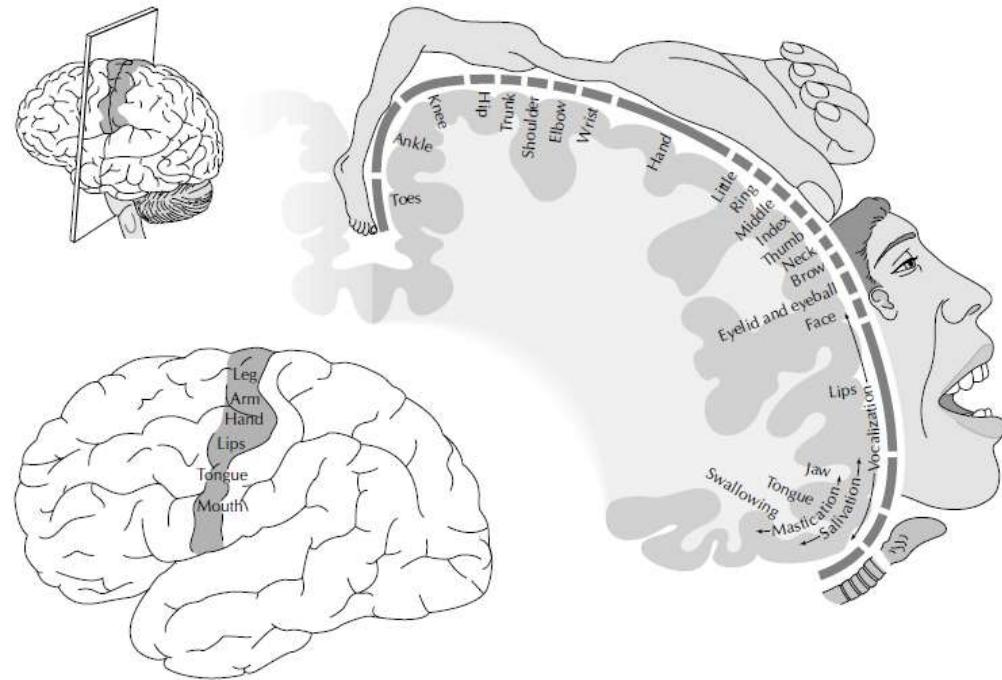


Figure 2: Diagram of the Motor Cortex [4]

## 2.6. Machine Learning

After processing EEG data from the microcontroller, a machine learning algorithm must be applied to get usable input from the processed data. This aspect of the project is indispensable, as without it, understanding the data drawn from the microcontroller would take longer than the maximum acceptable latency between when a user conveys a direction and that direction is implemented on the screen. A machine learning algorithm, by modifying its internal structure over a period of training time, derives a function that minimizes error in predicting the correct output. In this project, the machine learning algorithm reads brainwaves associated with the user thinking/doing something and classify the resulting input data into a given direction or a rest state. This result is then be output to the GUI to display the classified action as cursor movement. The most popular algorithm for this purpose, used often in previous attempts at working with EEG data, is that of artificial neural networks (ANN) [5] [6].

## 2.7. Graphical User Interface

A graphical user interface (or GUI) is any interactive program that involves working outside of a console window (or text user interface) on an electronic device. Examples of GUIs include Facebook, Minesweeper, YouTube, Pokemon and Microsoft Publisher. The purpose of a GUI is to facilitate ease of use on the user-side of the program. A properly designed GUI abstracts away unnecessary and possibly complex details of the underlying system in order to streamline a user's experience, and make their use of this system more efficient. For the purposes of this project, very little knowledge in designing a GUI is necessary for building the program that shows the user the result of the process being done (since it is relatively easy to learn from tutorials online).

### 3. Overview

Anima consists of three main components: an active sensor, a microcontroller, and a computer. The active sensor is the most complex, consisting of an electrode, instrumentation amplifiers, and various filters to remove noise. The electrode senses signals directly from the scalp in the microvolt range. From there, filters clean out unwanted artifacts, and instrumentation amplifiers amplify the resulting signal to magnitudes that can be detected by a microcontroller in the range of 0V to 5V. The microcontroller is used primarily as an analog to digital converter, converting analog signals from the output of the active sensor so that they can be sent onwards to a computer for further processing. After processing the analog signal from the active sensor, and turning it into a digital signal, the microcontroller then sends this digital signal towards a computer via the universal serial bus (USB) and corresponding universal active asynchronous receiver/transmitter (UART) protocols. The main purpose of the computer, then, is to make sense of these digital signals representing aggregate brain wave signals from various sensors, and to map patterns in these signals to control signals, for example to move a cursor up or down. To do this, the computer first separates the aggregate digital signals taken in from the microcontroller into their individual brain wave amplitudes. Next, the computer is used to first train a machine learning algorithm, so that it can “learn” patterns and commands from a user’s mind, and then apply these learned patterns for use in moving a cursor, or otherwise. The computer also provides support via a GUI, so that a user-friendly interface exists for software interaction with the EEG. A flowchart illustrating connections between these components is shown in Figure 3.

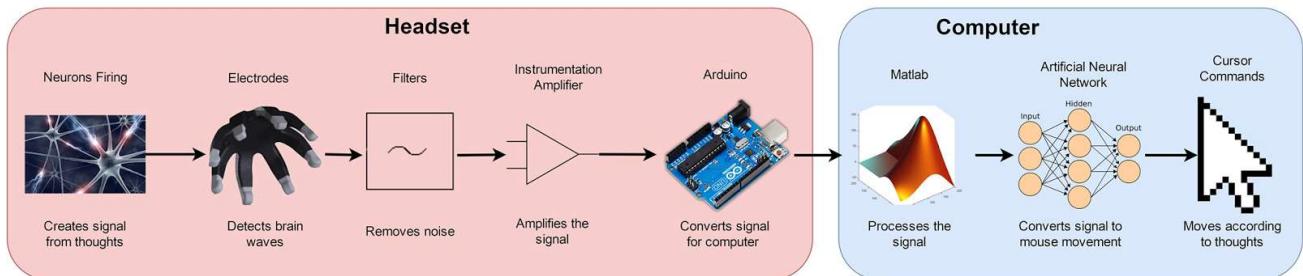


Figure 3: Anima Flowchart

## 4. Electrodes

### 4.1. Electrode Choices

When a well-designed electrode makes physical contact with a human scalp it can pick up brain waves that are as small in amplitude as microvolts. By maximizing contact with the scalp and minimizing impedance, electrodes can pick up more and more useful information from brain waves. Electrodes come in three main forms: dry, gel, and wet.

Traditional dry electrodes, as illustrated in figure 4, consist of multiple clustered conductive spikes that push into the skin of the scalp to maintain contact. They are uncomfortable due to the necessity of having to push spikes into a user's scalp, have less contact area than other types of electrodes, and have the highest impedance of all electrode types, but are relatively inexpensive and convenient, as a user doesn't need to apply conductive paste or other possibly undesirable liquids to aid with conduction. However, a user can potentially apply conductive paste if they wish in order to decrease the impedance of the electrode, although the use of conductive pastes and gels does tend to be inconvenient due to the time required to apply and remove them, as well as potential discomfort in the use of it. According to Lopez-Gordo, Sanchez-Morillo, and Valle, dry electrodes typically exhibit impedances of 150-200 k $\Omega$ s when used without conductive gels or pastes, or 5-10 k $\Omega$ s when used with them [7]. Therefore, the use of conductive gels and pastes can reduce the impedance by several orders of magnitude.



Figure 4: Dry Electrodes (12 mm diameter) [8]

Wet electrodes (figure 4b) are named thus because they must be used with conductive pastes, and are therefore the least convenient. They are often just flat conductive metal sheets that touch the skin of the scalp, and while inexpensive, may require undesirable shaving of the head in order to maintain contact. However, they do boast the greatest signal quality, due to their large surface area, a flat expected contact surface, and the expected use of conductive gels or pastes. Due to the necessity of applying conductive gels or pastes, they will typically exhibit impedances of 5-10 k $\Omega$ s [7]. Gel electrodes (figure 5) are the most expensive option, consisting of a conductive gel substance that can squish against a user's head. They are a good compromise between dry and wet electrodes, offering better signal quality than dry electrodes while not requiring conductive paste or shaving of heads. Due to the inconvenience of shaving heads, wet electrodes were taken out of consideration. Additionally, prohibitive cost and the need to keep gel electrodes moist took gel electrodes out of consideration. Thus, Anima uses dry electrodes.



Figure 4b: Wet Electrodes (12mm diameter) [9]



Figure 5: Gel Electrodes (12 mm diameter) [10]

At first, Anima had been planned to use dry spike electrodes, which are both inexpensive and easy to use. However, these electrodes proved to be uncomfortable to use, as the electrodes were comprised of groups of spikes that would sustain contact with the scalp via pressure. This meant that use of these electrodes required the user to have groups of spikes pushing against their scalp, which was at best uncomfortable, and at worst painful. Therefore, after further research and consideration of other electrodes, it was found that a relatively new type of dry electrode existed, which not only, like conventional dry spike electrodes, didn't require any form of moistening, application of conductive paste, or shaving of a subject's head, but also didn't comprise of spikes that would be uncomfortable on a subject's head.

#### 4.2. The Flex Electrode

This dry flex electrode, sized at approximately 20 mm wide by 10 mm high, consists of eight rounded, silver-plated points of contact, each connected to a central hub via a flexible and conductive elastomer, as seen in figure 7 [11]. The rounded points-of-contact make for a more comfortable wearing experience for the subject, and the flexible material allows for the points-of-contact to conveniently slide under hair and maintain solid contact with the scalp. Additionally, the silver contact material provides for a contact impedance as low as  $100 \text{ k}\Omega$ , allowing for brain wave signals almost as strong as those of wet electrodes, as shown in Figure 6.

Therefore, due to their convenience, ease of use, and comfort, Anima decided to use dry flex electrodes.

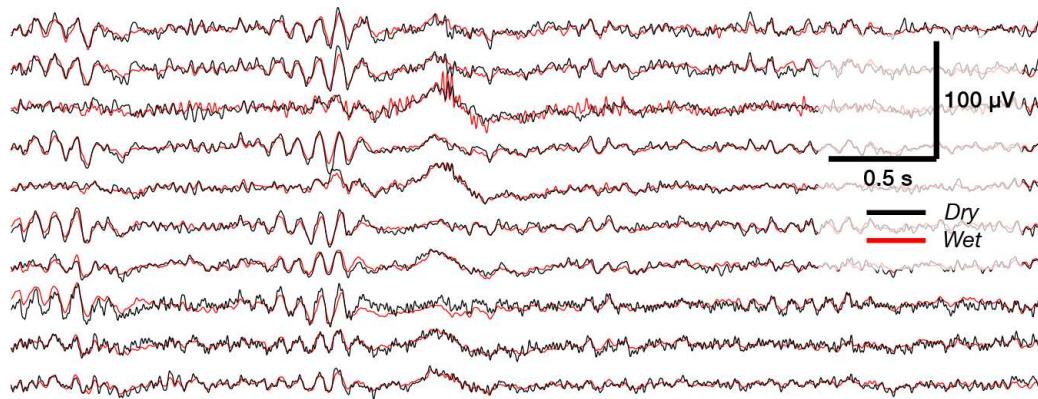


Figure 6: Comparison of brain signals taken from wet and flex sensors [12]



Figure 7: Dry Flex Electrodes (10mm by 20mm) [12]

## 5. Filters

### 5.1. Notch

60 Hz power line noise, as mentioned in section 2.2, is a big issue for reading brain waves because of its high amplitude in relation to brain waves that would rail all other signals if not reduced in amplitude. Therefore, in order to attenuate such artifacts, Anima uses a notch filter centered at 60 Hz. The low-pass filter, mentioned in a following section, provides additional attenuation of the 60 Hz line noise. A successful 60 Hz notch filter design must attenuate 60 Hz line noise while keeping a tight stop-band so as not to attenuate any signals of interest below 30 Hz.

Two notch filter designs have been considered, both variants of the Twin-T Notch Filter, one passive and one active. The passive filter, shown below in figure 8, was the first attempt, designed to maximize stop-band attenuation, and successfully reaching 52 dB of attenuation at 60 Hz in simulations (see figure 9). However, as one can see in figure 9, the -3 dB down points at 16 Hz and 260 Hz were unacceptable, as they would infringe on frequencies of interest between 16 Hz and 30 Hz.

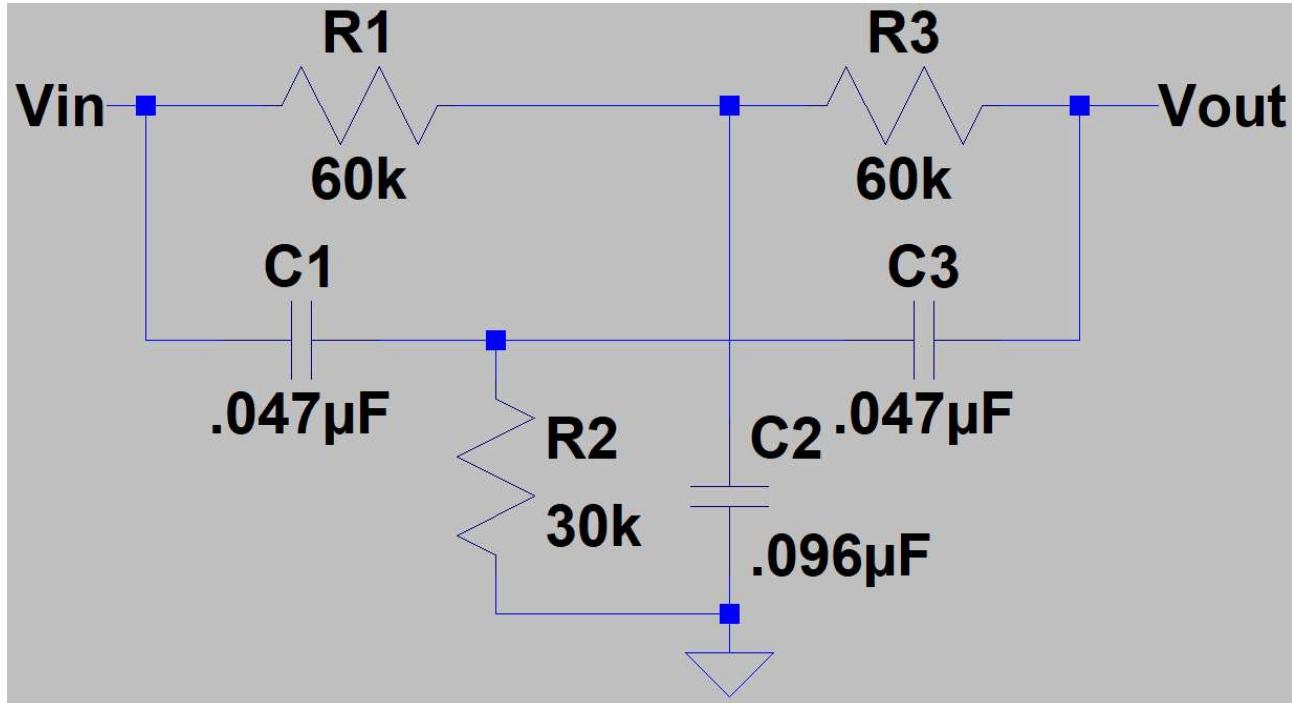


Figure 8: Passive Filter Design

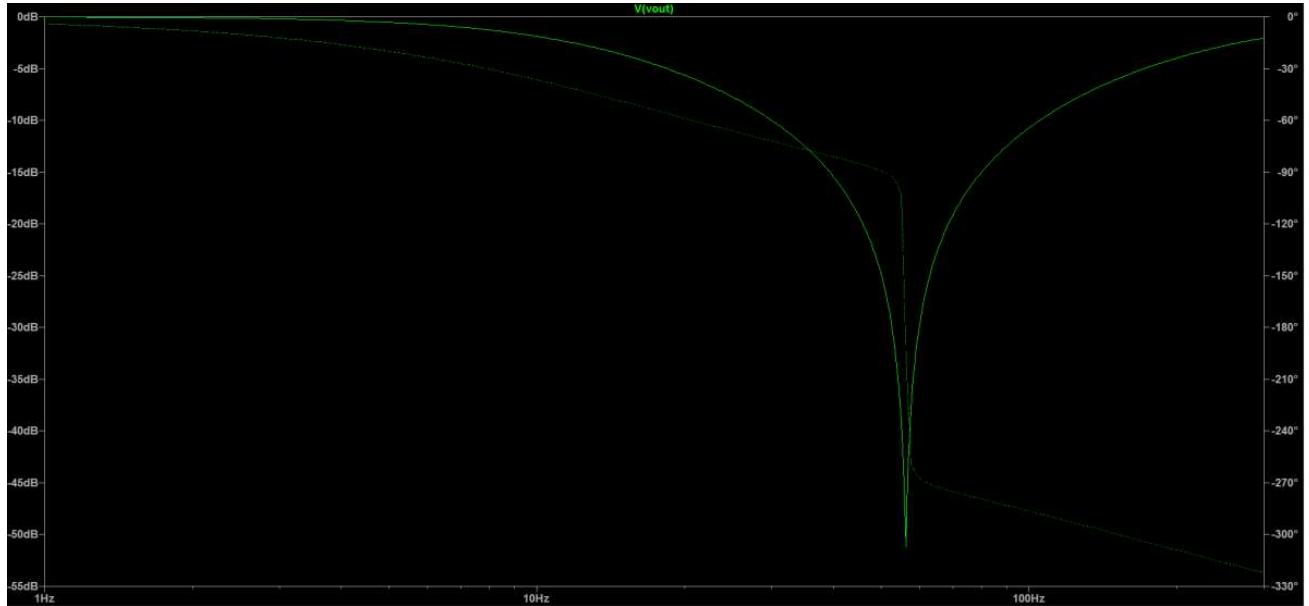


Figure 9: Passive filter simulated frequency response

The next attempted filter design was similar to that of the passive design, but involved use of an operational amplifier to reduce the frequency range of the stop-band. This active filter design is shown in Figure 10, and achieved a reduced stop band, with -3 dB frequencies of 27 Hz above and below the stop point at 60 Hz, as seen in Figure 11. It is important that this filter design achieves a much tighter stop band, one that no longer infringes upon the signal of interest at 30 Hz. However, this filter design only achieves a theoretical attenuation of 37 dB at the stop point, as opposed to the previous, much higher value of 60 dB. Additionally, without the use of high-precision, custom-valued resistors, with tolerances at or below 1%, it is impossible to design a notch filter with a stop frequency at precisely 60 Hz (plus or minus 1 Hz). Therefore, it is necessary to either purchase high-precision resistors, which would increase Anima's cost of production, or to use lower-precision resistors and risk reduced attenuation and variation in stop frequency.

In the end, it was concluded to purchase 1% precision resistors at the increased cost of approximately 3 precision resistors per circuit \* \$0.20 per resistor \* 2 circuits per active electrode \* 3 active electrodes = \$3.60 dollars total for Anima. This cost was determined to be reasonable, given that it was significantly less than that of even one flex electrode (\$30). An active electrode

constructed with such resistors would have an attenuation of approximately 37 dB, so two were chained for each active electrode to allow for a theoretical 74 dB of attenuation at 60 Hz, while maintaining the thin stop band of the active notch filter design.

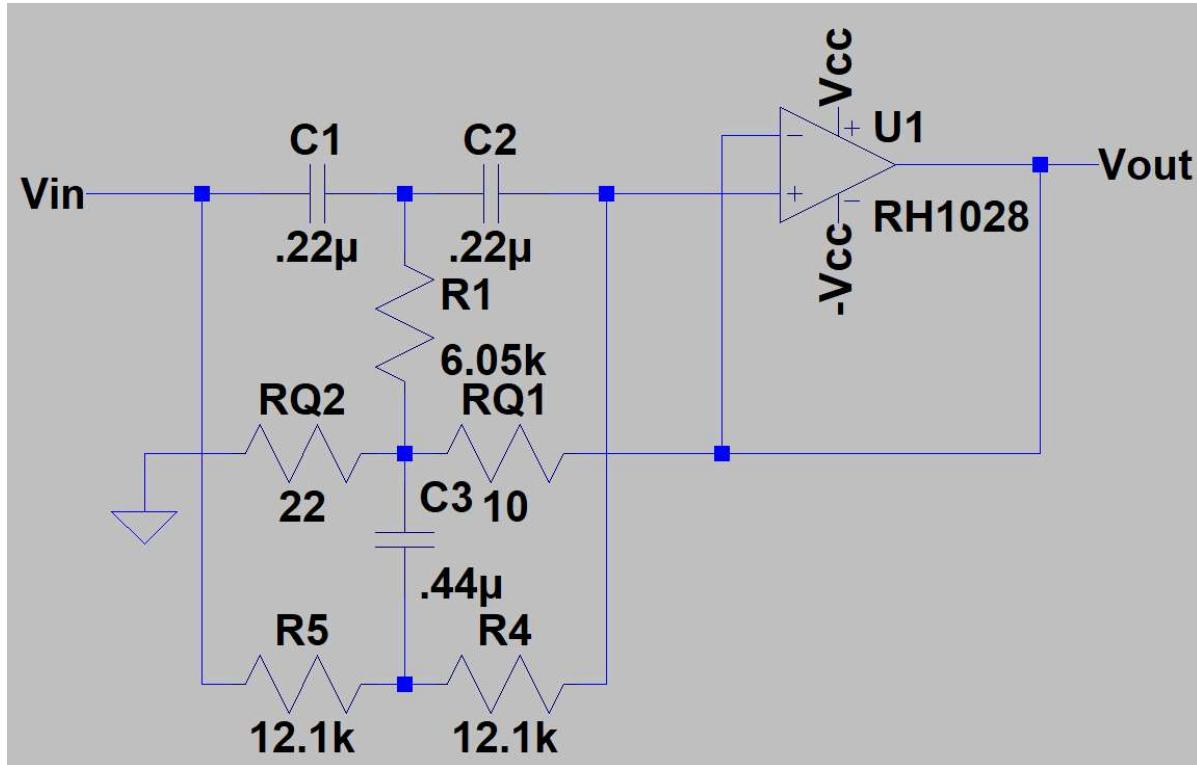


Figure 10: Active Notch Filter Design

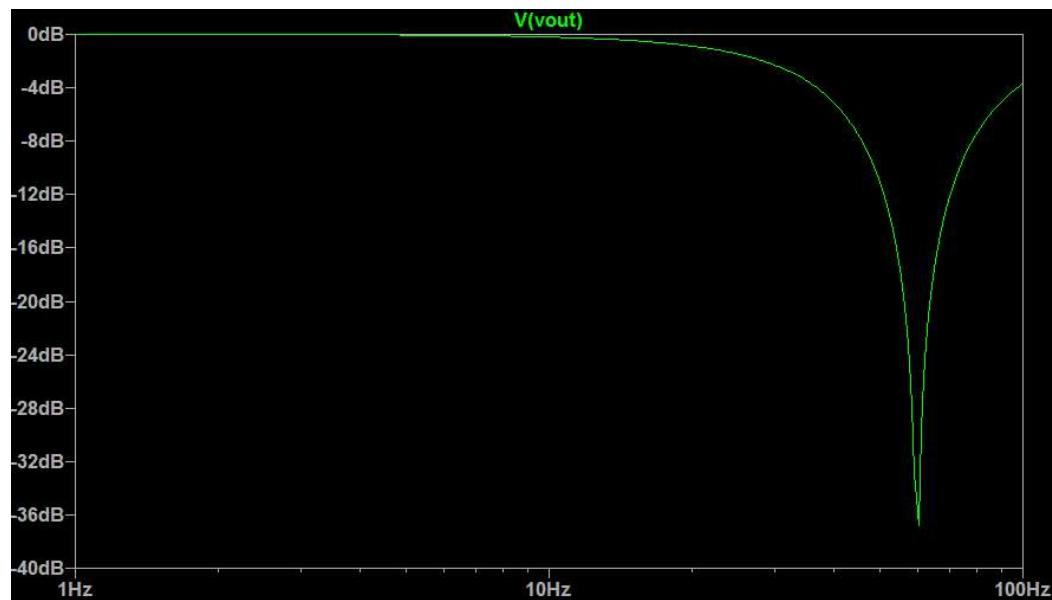


Figure 11: Active Filter frequency response

## 5.2. Low-Pass

Since the frequency of the useful brainwaves in this study are below 30 Hz, a filter is needed to attenuate all unhelpful artifacts above that threshold. To accomplish that, a third order Butterworth low-pass filter was designed, as shown in Figure 12. The cut-off frequency was placed at 35 Hz so as to avoid attenuation of any EEG signals of interest. The filter was designed to provide unity gain with a quality factor of 0.7 so as to not also increase the amplitude of the frequencies removed by the high pass filter. The filter was also designed to use easily available inexpensive components while still meeting all of the requirements. The frequency response of this filter is shown in Figure 13.

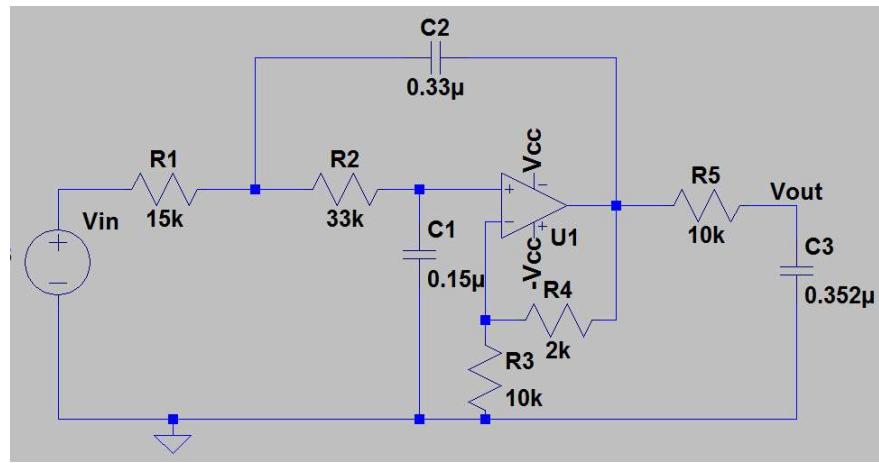


Figure 12: 3rd Order Low-pass Filter Design

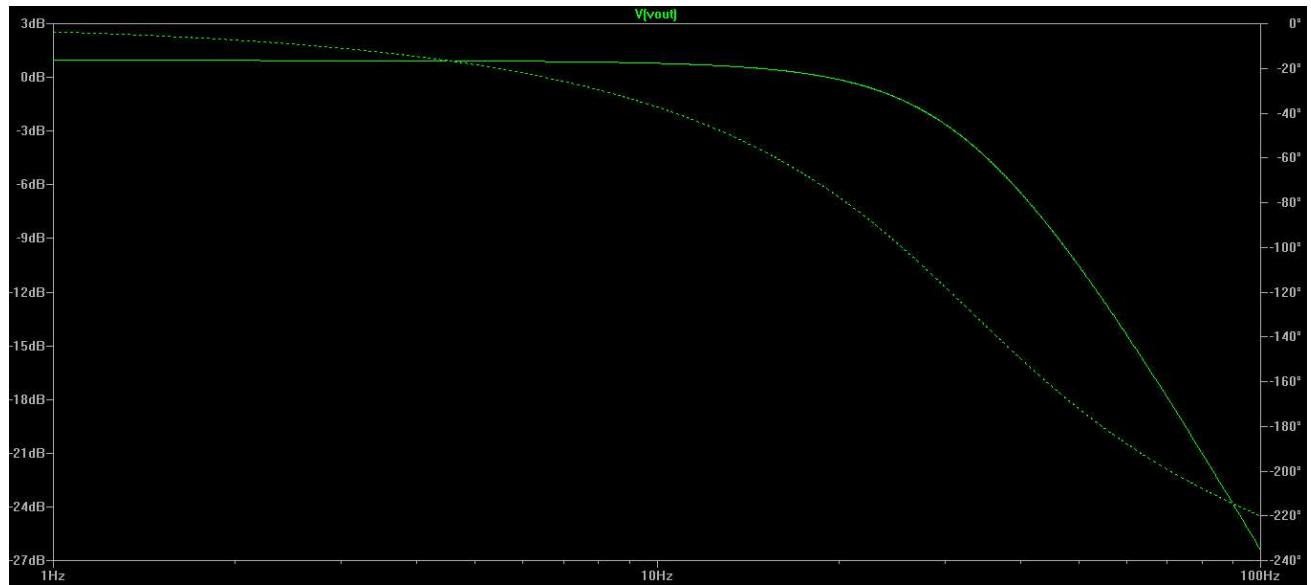


Figure 13: Low-pass Filter Frequency Response

### 5.3. High-Pass

A similar design was used to make a third order Butterworth filter with cutoff frequency at 7 Hz, unity gain in the pass band, and a quality factor as close to 0.7 as possible, as shown in Figures 14 and 15.

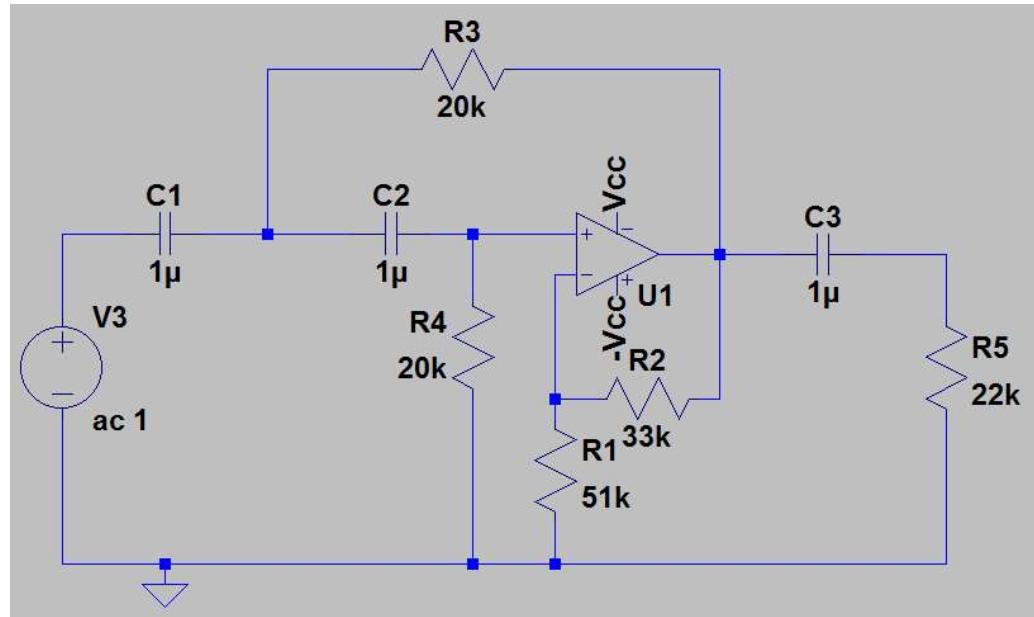


Figure 14: High-pass Filter Design

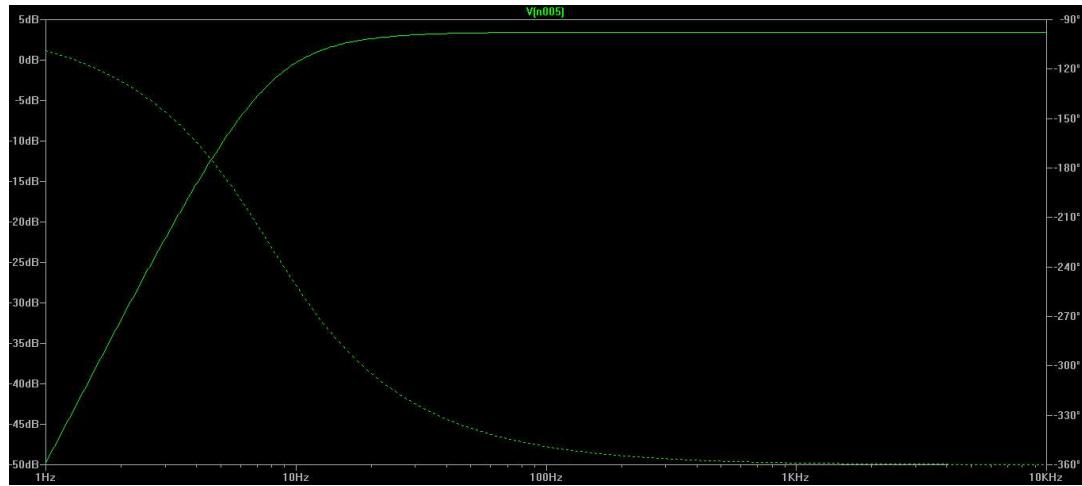


Figure 15: High pass Filter Frequency Response

## 6. Instrumentation Amplifier

Brain waves are read at such a low amplitude (microvolts) that they need to be amplified to prevent them from not being picked up by the microcontroller. Precise instrumentation amplifiers, with high gain, low offset voltage, high input impedance, and near-zero output impedance, are available. Such instrumentation amplifiers can allow for the microvolt signal from the electrode to easily be amplified to the single volt range. However, concerns arise from the high amplitudes of artifacts, such as signals from muscles as well as 60 Hz line noise. Once amplified, these artifacts could potentially cause for the microcontroller to saturate and rail, so they must be removed for proper amplification.

### 6.1. Implementation

The instrumentation amplifier used for this stage of the project is the AD620 amplifier because it was specifically designed to be used with EEG equipments. The gain of the amplifier is controlled by the resistance between two of its pins following the equation:

$$G = 1 + 49,400 / R_G$$

So with a 560 ohm resistor the amplifier would have a gain of about 90. The AD620 also has an input impedance of  $10\text{ G}\Omega$  with a negligible output impedance which makes it an effective impedance bridging device. This quality allows the active sensor to maximize the signal to noise ratio of the brain waves.

## 7. Combined Active Sensor

The active sensor consisting of filters and instrumentation amplifiers was initially designed on a breadboard to verify base accuracy. After confirming that the anticipated functionality was achieved, the filters and amplifiers were soldered onto a perforated board for the purpose of reducing resistive and capacitive error from the breadboard. The perforated board also allows for construction over a much smaller space while both retaining and improving accuracy. Unfortunately, errors on the perforated board caused by issues such as parasitic capacitance led to unfavorable results. A printed circuit board, or PCB, helps avoid this issue with its copper traces.

Anima's PCB was designed using the EAGLE software program and sent to a OSH Park, a community-run PCB fabrication shop. After PCBs were delivered, the necessary components for the circuit were soldered on and the accuracy of the design on the PCB was once again verified. The PCB layout is shown in Figure 16, and the completed PCB is shown in Figure 17.

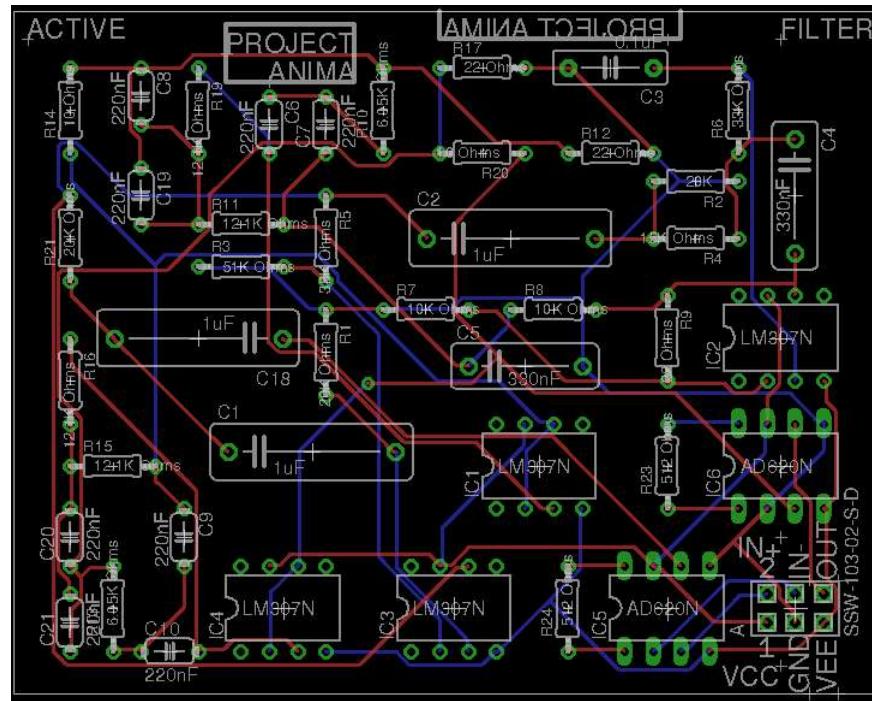


Figure 16: Printed Circuit Board Design

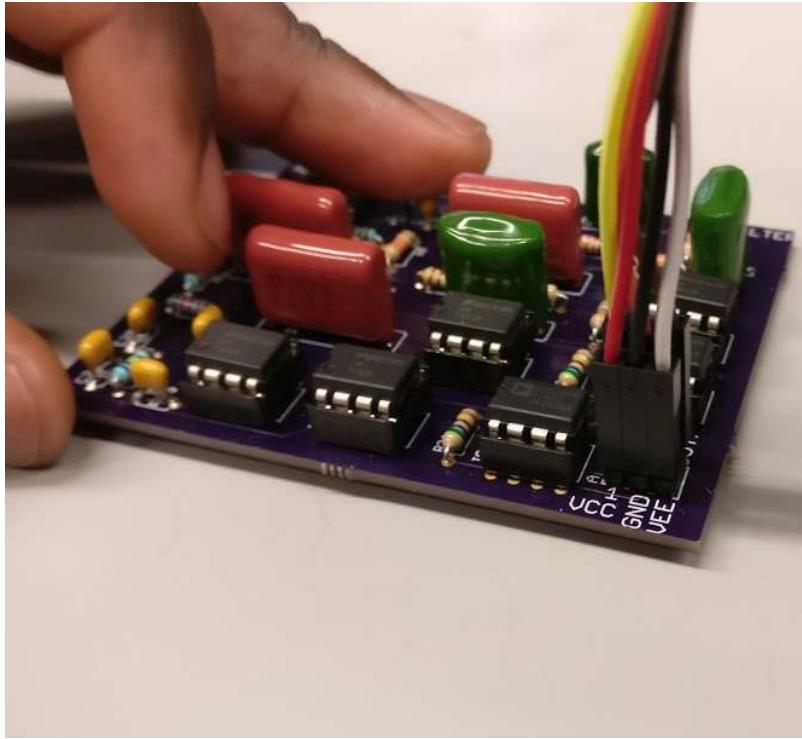


Figure 17: Printed Circuit Board

## 8. Microcontroller

### 8.1 Microcontroller Selection

The Arduino Mega was chosen due to exceeding all of these requirements at a minimal cost. The Arduino Mega has a 16 channel, 10-bit analog-to-digital converter that operates on an input voltage range of 0V to 5V and sampling frequency of 10 kHz [13]. It is powered by a connected laptop's batteries in order to isolate the EEG circuit from 60 Hz line noise as well as possible power spikes, and conveniently transmits data to the computer through use of the popular Universal Serial Bus (USB) and Asynchronous Receiver/Transmitter (UART) protocols. The Arduino Mega also has sufficient processing power and speed to perform another layer of signal cleaning if necessary, such that the execution of such a task would seem as if it were operating in real time.

As the EEG signals of interest reside between 8 and 30 Hz, it is important to have a minimum sampling rate of 60 Hz, the Nyquist frequency of the EEG signals of interest. Anima's design consisting of several flex electrodes and two neutral ground reference wires requires at least eight such channels, or differential inputs, are required for any such analog-to-digital converter. Additionally, in order to minimize quantization error, it would be preferable to have as many bits as possible of resolution. After testing with raw data, this was determined to be the preferred microcontroller.

After analog-to-digital conversion, all EEG data is transmitted from the Arduino towards the computer, and input into MATLAB for feature extraction.

## 9. Digital Signal Processing

After receiving serial data from the microcontroller, the computer still cannot make accurate decisions on cursor movement from only a signal that is an aggregate of all brain waves. The first step to addressing this problem is to map this signal into representations of brain waves. The output needs to represent brain waves that are both accurate and separable, so that the following step, machine learning, has accurate data with which to make decisions regarding cursor movement.

### 9.1. Signal Separation

The EEG signals sent serially from the Arduino are aggregate signals representing all brain waves and unfiltered artifacts across all frequency bands. Isolating the brain waves of interest, mu waves (8-12 Hz), by frequency requires transforming the signals from the time domain to the frequency domain. After acquiring EEG data from the microcontroller, the computer performs the Fast Fourier Transform (FFT) to transform time bins of 100 ms (deemed appropriate after extensive testing) each into the frequency domain. Since brain waves reside in specific frequency bands and in different parts of the brain, each brain wave is identified by its amplitude during that time bin, within both the frequency range and on the specific electrode or electrodes. Distinction between brain waves by amplitude and source separates the Beta waves from 16 Hz - 31 Hz, the Theta waves from 4 Hz to 7 Hz, and the Mu waves between 8 Hz and 12 Hz.

### 9.2. Localization

Of all the different brainwaves previously mentioned, the Alpha and Mu waves are the only ones residing on the same frequency bands, and need to be isolated from one another. Since Alpha and Mu waves originate from different parts of the human brain and are highly localized, they are more easily measured from the location on the scalp closest to their point of emission. Specifically, Alpha waves originate from the parietal and occipital lobes at the back of the head, while Mu waves originate from the motor cortex located in the region of the brain between the two ears. Therefore, localizing each of these two waves is relatively simple in theory. It is important to have at least one electrode on the occipital or parietal lobes, where Alpha waves are stronger, and another over the motor cortex for the Mu waves. Afterwards, Alpha waves can be estimated as the difference in frequency-domain amplitude between the parietal/occipital lobe electrode(s), and the motor cortex electrode(s) and vice-versa for the mu waves.

### 9.3. Feature Extraction

The neural network expects as input a series of values correlated with the user mental processes. However, after isolating the brain waves of interest, the output would only consist of frequency domain signals which barely represent the intentions of the user. Some additional processing is necessary to extract features which are strongly correlated to user intents or voluntary mental processes. Choices of interest include, of all amplitudes within each frequency range, the average amplitude over the range of mu waves (left vs right side of the brain), minimum amplitude in the mu range, range of amplitudes and average quadratic amplitude (where amplitudes are squared and averaged). Determination of the proper input criterion has been found via experimentation, and it is notable that the neural network can expand to allow for multiple choices of input, although that would be undesirable due to increased storage and computation requirements.

## 10. Machine Learning - Artificial Neural Network

An artificial neural network (ANN) is a computational approach to learning patterns modelled on the neurons of the human brain, and how it solves problems using artificial neurons. Artificial neural networks offer an excellent means of classification based on nonlinear signals with large data sets. In this case, the data set is large due to continuous time training over long periods of time, necessitated by the low frequency of human brain waves [14]. Artificial neural networks have been used effectively in detecting abnormalities in medical patients' brain signals with high accuracy [15]. Similarly, neural networks have been demonstrated to be capable of reading brain waves of healthy humans in various EEG products [5]. Given that this method of learning is relatively well explored (gaining ground in 1975 with the backpropagation algorithm, discussed later), understanding how to create successful types of artificial neural networks is necessary to get the most out of the data that is being extracted.

### 10.1. Artificial Neural Network Background

A feed-forward ANN consists of many nodes called neurons organized by layers [16]. The topology of a feed-forward ANN is that of a directed acyclic graph (see Figure 19). Each neuron consists of an input and output, as well as an activation function and set of weights. Every neuron has inputs and/or outputs linked with every other neuron in the next or previous layer by what can be referred to as links. Some ANNs use as an activation function a threshold function such as a sigmoid function, but all are used to determine how the node will act. In a feed-forward neural network, neurons are separated into layers that specify the flow of the network. The first and last layers are known as the input and output layers and the data going into and out of the neural network deal with these layers only. The rest of the layers are known as hidden layers in part because the information about the values of these nodes never needs to be known by the user. Whereas the input and output layers are mandatory, hidden layers are technically not mandatory for an artificial neural network. Input neurons are unique in that they only have one input, and output neurons are unique in that their output doesn't go forward to any more neurons, as they are the "final" neuron in the decision-making process.

Weights in neural networks are constantly being modified in training to self-adjust such that when training is finished testing should produce correct results with high accuracy (see the section

on Backpropagation for the weight adjustment formula). Each neuron unit or node consists of inputs with associated weights that are summed and tested to see that they meet a specific threshold. Neurons may have many input values, but only one output value, even if that output value may affect many different neurons in later layers. The activation function is usually a step or sigmoid function where the neuron “fires” just like an actual neuron if a specific threshold is reached (a nonzero value is output on firing). This function determines the strength of the output given the input, and the weights determine the effect of each input on the value of the generalized input of a neuron. The threshold of the activation function is specified by a bias weight that is a defining characteristic of each neuron. Along with this bias weight is an input of -1 so that when all the weighted sums are calculated the activation function gives a nonzero value (one in a sigmoid function) when the weighted inputs sum to a value above the specified threshold in the bias weight. Essentially, all inputs are weighted and summed together to be input into each neuron. The unit-step function could be used to demonstrate the threshold; however, the step function does not apply for the purposes of this project because it needs to be differentiable, as is shown later. Instead, Anima uses the sigmoid function, a popular differentiable function used in neural networks. Figure 18 shows a comparison of the two activation functions.

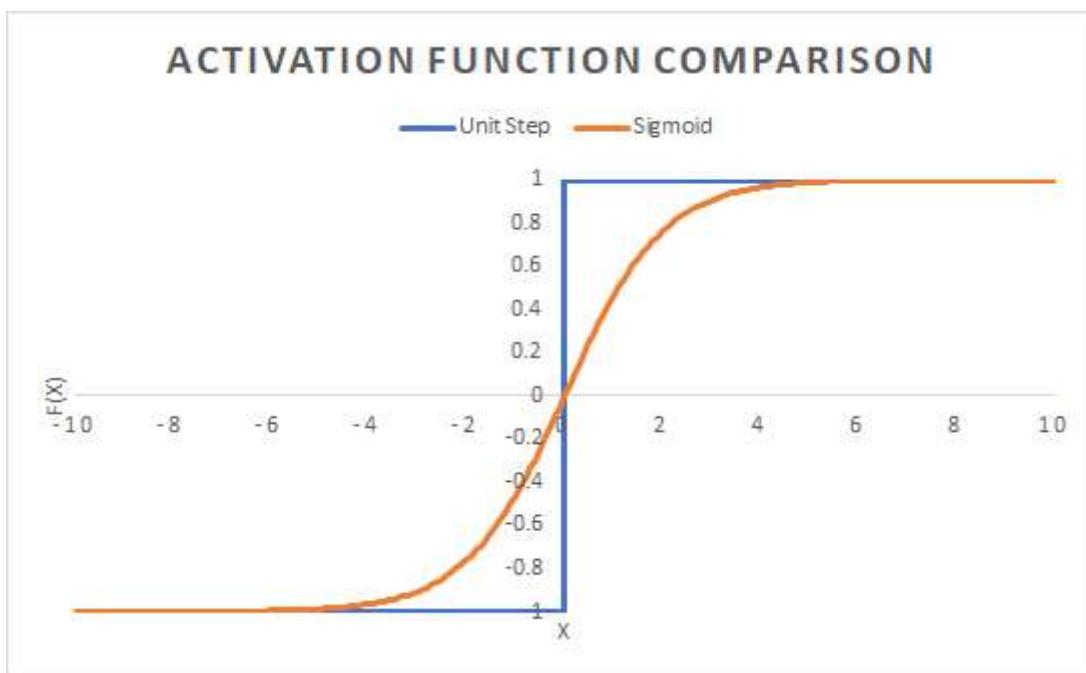


Figure 18: Comparison of the Unit Step and Sigmoid Functions

Feed-forward neural networks are trained with backpropagation learning in a relatively low noise environment to minimize the effect of noise and artefacts (unwanted signals from the subject and machines). In backpropagation learning, a multilayer neural network is trained from output to input by adjusting the weights of input parameters repeatedly to learn how to classify them based on patterns in the data [17]. For this project, there are three outputs to cover the two directions the cursor can move in (and a rest state) and there should be many inputs for information about every relevant value regarding mu waves. Figure 19 below represents a possible network topology for a neural network. The addition of a hidden layer for the training and testing of neural networks has been demonstrated to expand the possible applications for this type of learning. As an example, just adding one hidden layer to a network can allow for representing any continuous function by that network and two or more hidden layers can represent even functions discontinuous in their derivatives [16].

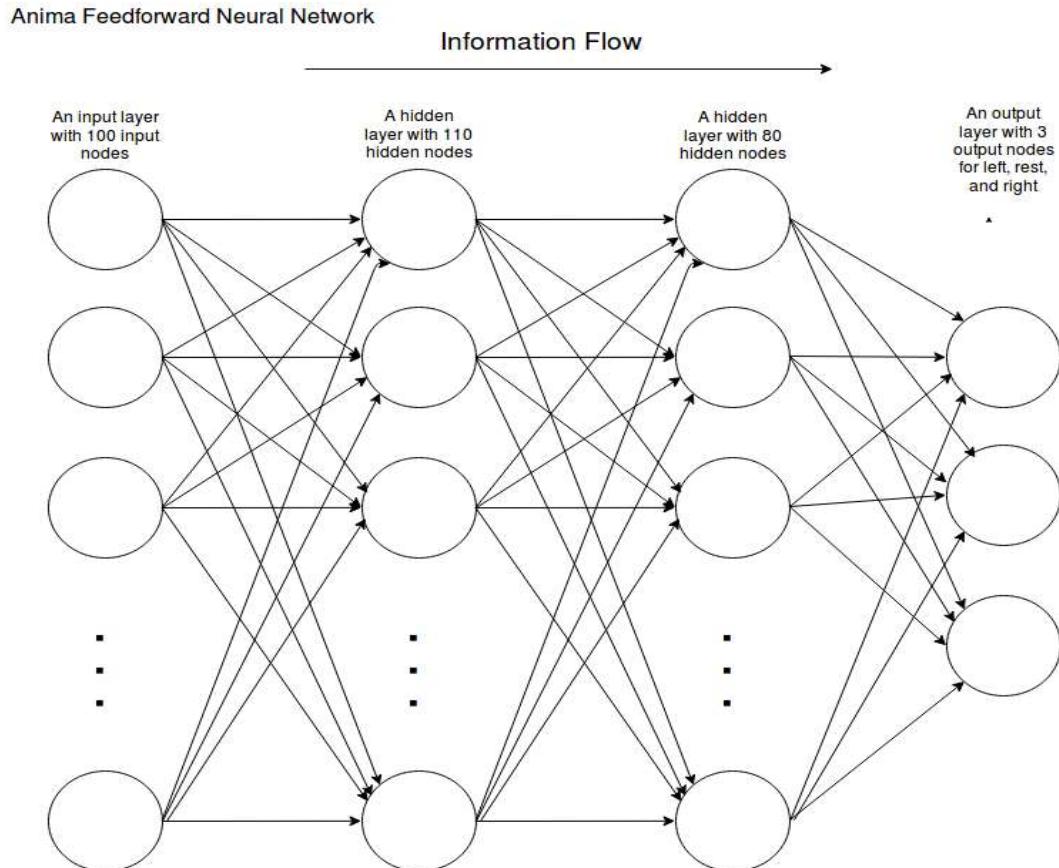


Figure 19: Neural Network Topology

## 10.2. Hidden Layer Size Determination

In order to determine optimal characteristics for the neural network, classification accuracy was tested given different parameters for the hidden layers. Classification accuracy is the ratio of correctly classified instances over the ratio of incorrectly classified instances given a dataset. In Anima's case, inputs consisted of all relevant brain data, whereas outputs consisted of three classes: left, rest, and right. A correctly classified instance could be when the neural network guesses that the user was attempting to move the cursor left and the user was trying to move left; an incorrect classified instance could be when the neural network guesses that the user was attempting to move left and the user was not trying to move left. In Figure 21, a confusion of 45.5% is shown on the bottom right square of the bottom left chart.

Cross-entropy is defined as:

$$H(p, q) = - \sum_x p(x) * \log(q(x))$$

$P(x)$  is the “true” distribution of expected output values and  $q(x)$  is the “unnatural” distribution of values output by the neural network. Essentially, cross-entropy measures how well the neural network is doing and how accurate its guesses are, as well as how confident it is when correct and incorrect. As cross-entropy is essentially a measure of error, lower values represent a better neural network, and higher values represent a worse one.

A neural network’s data is split into three datasets: a training dataset (70% of total), a validation dataset (15% of total), and a testing dataset (15% of total). The training dataset is used while updating weights given the backpropagation algorithm, and while testing, a performance measure, like cross-entropy, is computed periodically for the neural network. As the neural network trains, performance, as measured by cross entropy, is measured every epoch, or every time the neural network goes through the dataset once, and performance values are measured for all of the 3 datasets. Training performance is expected to decrease after every epoch, as the training dataset is used for changing of the neural network’s weights. Validation and testing datasets are only observers in this initial stage, as weights are not changed for them. However, performance is indeed measured, and at first it decreases as well. After some time, the neural network will begin to overfit, which is

detected by the validation performance increasing once more. This is due to the validation dataset having no say in the weights of the neural network, and the neural network beginning to become too specialized for understanding its training dataset only, so that it can no longer generalize to other unseen datasets, such as the validation and testing datasets. Therefore, once validation performance begins to increase once more, the neural network will terminate training, and choose the network with the lowest/best validation performance. Once this is done, the validation dataset will have had some impact on the training of the neural network. Therefore, testing performance is calculated as well for insight on how the neural network would perform on completely new and unseen data. A graph of training, validation, and test performance while a neural network trains is shown in Figure 20, and the confusion matrices which result in the trained neural network is shown in Figure 21.

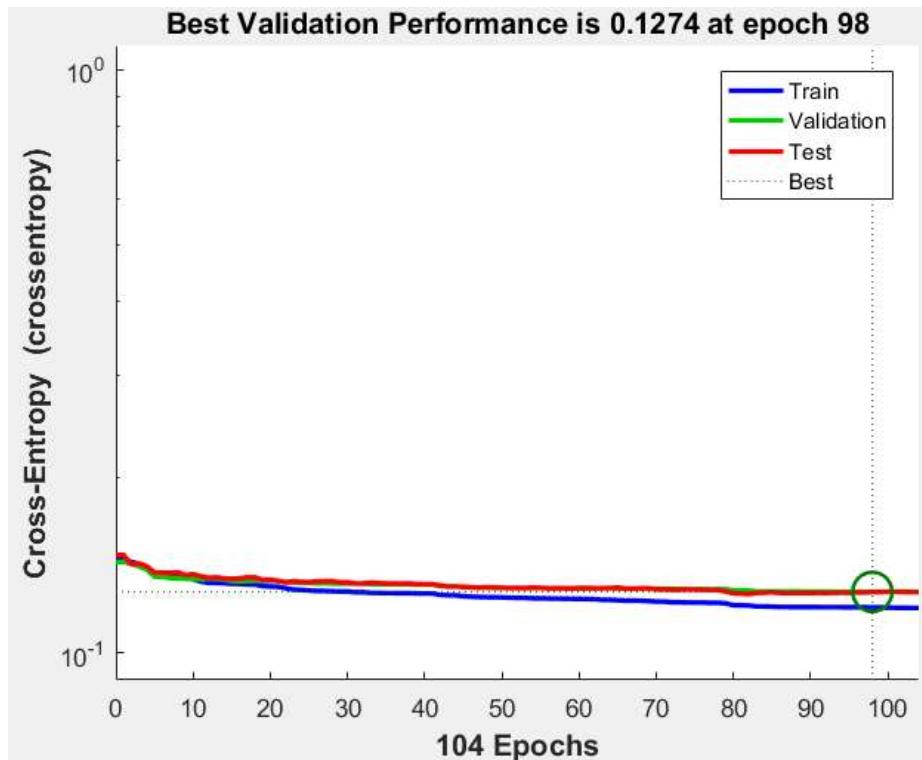


Figure 20: Training Performance Graph

Confusion matrices can also be used to understand how a neural network has performed. A confusion matrix represents on one axis, the output classes, or the output values of the neural network, whether left, calm, or right. On the other axis, the confusion matrix shows target classes, or whether the correct answer was left, calm, or right. The confusion matrix then forms a grid, whose

values then represent the accuracy at each value. Therefore, values of target class x and output class y represent how many times the neural network guessed class y, given that the correct class was x. This is also represented as a percentage, or the percentage of the time the neural network classified the input data into the correct output group given input data representing that output group. There are then extra columns given to represent averages. For example, after the 3rd output class is a row that represents the percent of the time that the neural network classifies the input into the correct output group, given that the data represented target class x. Similarly, the 3rd column after the 3rd target class represents the percent of the time the neural network would classify input data into the correct output class, given that they guessed an output class y. Confusion matrices can be computed given only the training dataset, validation dataset, or test dataset, as well as for all datasets. An example confusion matrix is included below, as Figure 21. For reference, guessing accuracy on three classes would be 33%. Therefore, accuracies of 47.2%, 41.1%, 41.9%, and 45.5%, representing the training, validation, test, and overall dataset classification accuracies, were all much higher than guessing accuracy. This shows that the neural network was indeed capable of finding a pattern in the brain waves representing left, calm, and right, the three classes.



Figure 21: Confusion Matrices

In order to determine both the amount of hidden layers to use, as well as the amount of hidden nodes to have in each hidden layer, first one had to decrease the search size. Research suggested that the optimal amount of hidden nodes generally lies approximately between the amount of input nodes (100) and the amount of output nodes (3). Additionally, as previously noted, two hidden layers can allow a neural network to represent most functions. However, the addition of more hidden layers means more layers through which the backpropagation algorithm would have to send the gradient, and therefore a longer training time and possible loss of potential training. To conclude, only one and two hidden layers were tested.

To find optimal parameters for the neural network, one and two hidden layers were attempted, as well as from 1 to 100 hidden nodes per layer. For each such parameter, 10 neural networks were trained, and the average of their classification accuracy was taken. Data representing

a two layer neural network is represented in Figure 22, where the x axis represents the amount of nodes in the first layer, the y axis represents the amount of nodes in the second layer, and the z axis represents the average classification accuracy after 10 neural networks using that parameter. This graph showed that classification accuracy generally increases with more nodes per hidden layer, but moreso on the first layer than on the second. Figure 23 shows the average classification of 2nd layer hidden nodes varied from 1 to 100 for  $10 \times 100 = 1,000$  neural networks, with 1st layer hidden nodes varied from 1 to 100 on the x axis, and classification accuracy on the y axis. Additionally, Figure 24 shows the average classification of 1st layer hidden nodes varied from 1 to 100 for  $10 \times 100 = 1,000$  neural networks, with 2nd layer hidden nodes varied from 1 to 100 on the x axis, and classification accuracy on the y axis. These neural networks were tested on an hour of data collected on one individual, and determined to scale on other datasets and people as well.

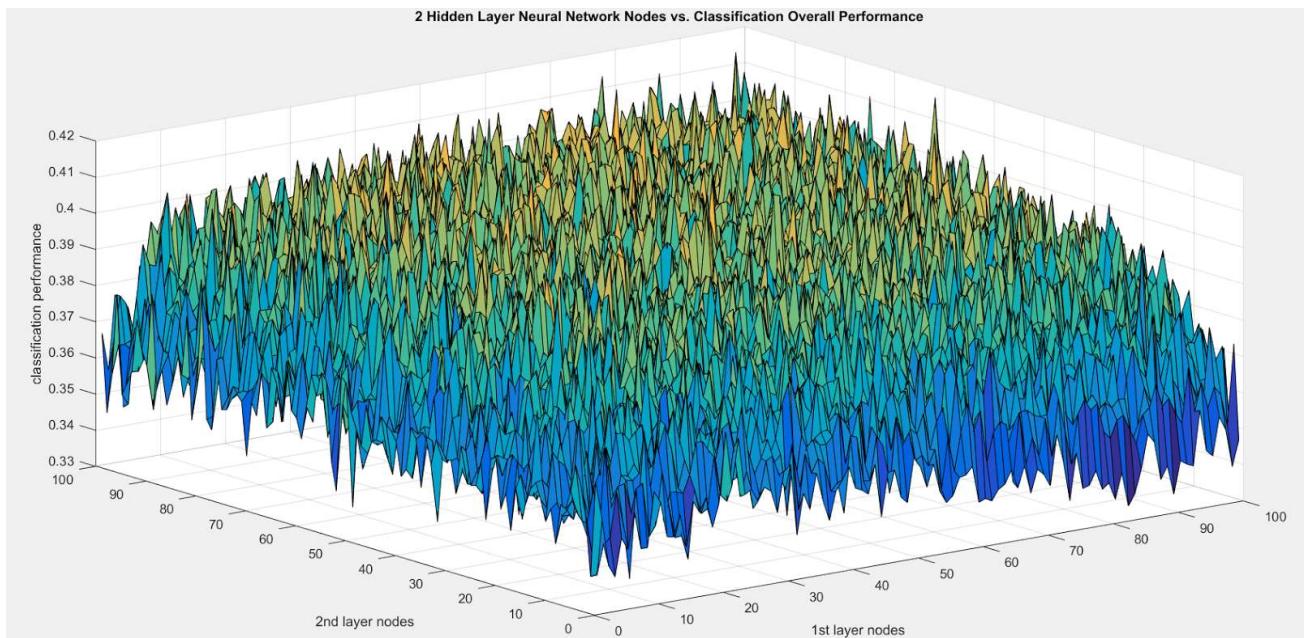


Figure 22: 2 Hidden Layer Neural Network Classification Performance

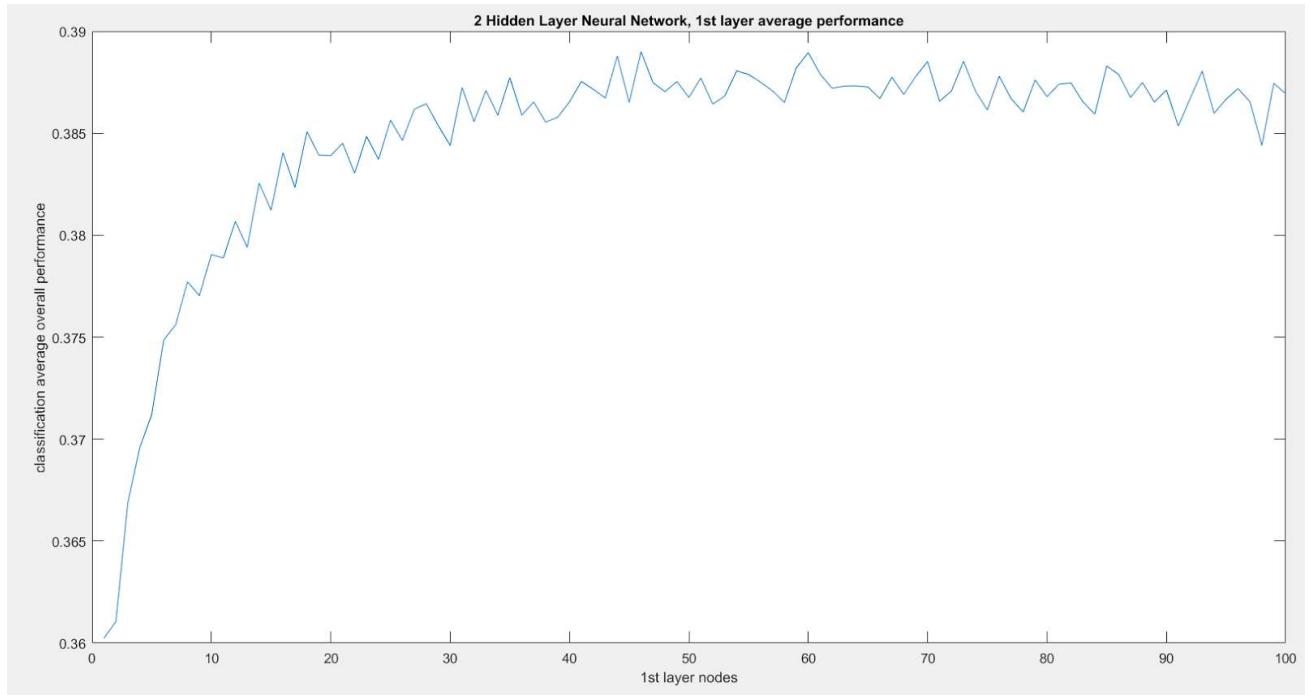


Figure 23, 1st layer average performance

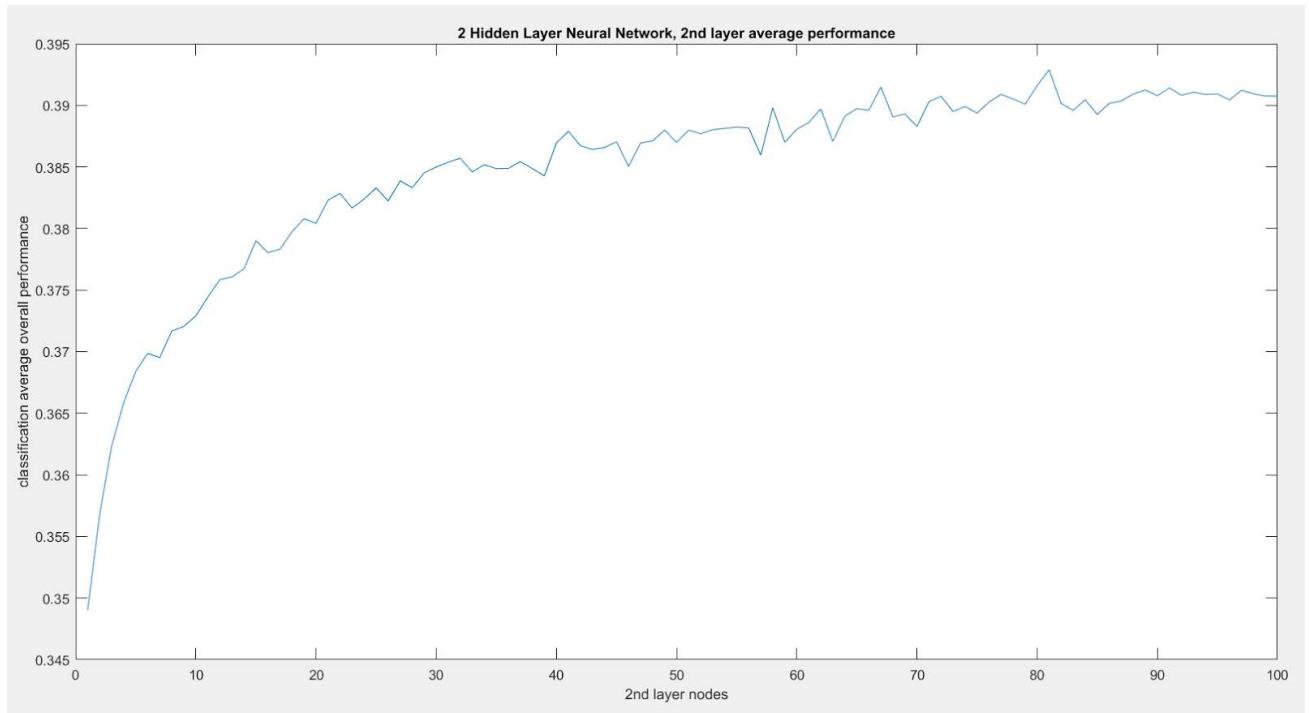


Figure 24: 2nd layer average performance

Looking at the data from testing as described in the previous paragraph and shown in Figure 22, the maximum classification accuracy (average of 10 neural network instances) was 41.9%, exhibited with two hidden layers, with 80 hidden nodes in the first layer and 50 hidden nodes in the 2nd layer. However, after analyzing averages on Figures 23 and 24, it was noticed that the noisiness of the data led to many unexpected spikes, so the trend of our data was analyzed instead. After looking at the trends, it was instead determined to have two hidden layers, with 110 hidden nodes in the first layer and 80 hidden nodes in the second one. This configuration, shown in Figure 25, was able to provide Anima's neural network with >42% accuracy, which was more than that of the previous configuration. Therefore, Anima chose to use two hidden layers, with 110 hidden nodes in the first layer and 80 hidden nodes in the second layer. This configuration's classification accuracy is shown in Figure 21, to be 45.5% overall. This value was much higher than any found during our simulation, and was satisfactory for Anima.

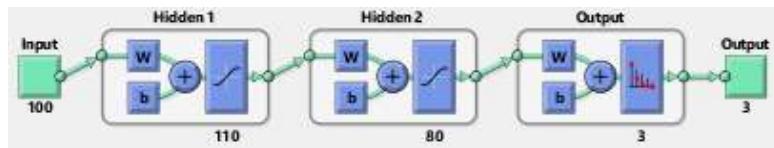


Figure 25: Final Neural Network Topology

### 10.3. Backpropagation

Backpropagation learning works to determine whether a pattern in data is useful by taking the link between nodes where the pattern is and increasing the value of its weight [17]. As was mentioned earlier a feed-forward ANN works based on weights applied to every neuron. The outputs for these neurons in the input layer are just a vector of inputs with no activation function on them. In training, the desired output is also given as a vector since the purpose of training is to adjust the weights by error between expected and actual output so they can produce desired output when tested. In perceptron (one-layer network) learning, adjusting the weights is relatively trivial since each output sums up the weighted input vector, but for multilayer networks the existence of hidden nodes (neurons) with their own weights makes the problem more complicated. In backpropagation learning, input and output example vectors are used to find the error between the current output and the expected output. Next, the error at the output is propagated back to each previous hidden layer,

adjusting the weights at each previous layer to better fit the output. This helps toward the process of reaching the optimal network weight structure.

The backpropagation algorithm is selected so that any errors while training will propagate back towards the front, changing weights and in effect “training” the neural network to learn the patterns. The backpropagation algorithm works in effect by using training data, and calculating an expected output given training input data. From there, the output nodes will have training output data, which will be compared with the output values predicted by the neural network. The difference between the training output data and the predicted output data is labelled as the error, and it is this error that is propagated back towards earlier nodes in the network to constitute the “learning” in the artificial neural network. The gradient, or partial derivative, of the error is then found for each weight preceding the current node, to represent the amount that the current node’s error would decrease given a change in the previous node’s weight. This is then acted upon by changing the weight between the current and previous node through use of that gradient and error. From there, the previous node does the same to its preceding weights, until the input node is reached. Through use of the backpropagation algorithm, a neural network can come to “learn” how to predict outputs in the training set given inputs. Given that the input and output data relationships existing in the training set are representative of the input and output data relationships existing in the real world, the neural network, after use of backpropagation, characterizes a function to predict a set of output values for this function given any set of input values.

The process of updating the weights can be illustrated by the following equation from [16]:

$$W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times Err_i \times g'(a_i)$$

The weight term  $w$  is defined as the weight from node  $j$  in the previous layer to node  $i$  in the current layer. The update is a product of the learning rate, the activation of node  $j$ , the error at  $i$ , and the derivative of the activation function. The learning rate is user defined and is chosen based on a trade-off between speed and accuracy in learning (it can also be described by how much the weight is adjusted by toward the ideal value). Backpropagation learning runs over epochs (or the number of times looping training), so the number of epochs should be sufficient to compensate for small

learning rates. The activation of node  $j$  is calculated by passing the weighted sum of the inputs to the node to the activation function. The error at node  $i$  is the back propagated error or, if  $i$  is in the output layer, it is the difference between the training example's output and the output calculated by the current neural network with the training example's input. The last component, the gradient of the input to node  $i$ , is used in conjunction with the error at node  $i$  to determine how much node  $j$  is contributing to the overall error of node  $i$ . This process of updating the network is what this project uses for training.

There are many training functions, each of which employs a different algorithm to compute the error in order to update weights. After testing and comparing classification accuracy of different backpropagation algorithms, including traditional gradient descent, bayesian regularization, and levenberg-marquardt, we chose to use scaled conjugate gradient to train our neural network. The scaled conjugate gradient method is a second order technique (in contrast to gradient descent, which is first order), which generally results in convergence of the neural network weights at a lower (local) minima than that of first order methods, albeit at higher computational cost [18]. The scaled conjugate gradient method is also a conjugate gradient method, meaning that it will move weights towards the conjugate of the directions in previous steps, as opposed to directly down the gradient as traditional gradient descent would do. By moving in the conjugate of directions in previous steps, previous weight changes will not be undone by future ones as is common with other methods. The scaled conjugate gradient method varies from other conjugate gradient methods by taking an approximation of the Hessian matrix of the error function as opposed to manually computing the Hessian as other conjugate gradient methods do. In short, the scaled conjugate gradient method can compute more accurate neural network representations of functions more quickly than other conjugate gradient methods, though at a higher computational cost.

## 11. Graphical User Interface

### 11.1. Implementation

The GUI portion of the project work primarily to interpret and display direction instructions from the neural network by controlling cursor movement on a computer application. There is almost no filtering done in this part of the process and the main job of the GUI is to translate instructions from the neural network output into cursor movement on the screen. The specific cursor begins by moving a few pixels per second and stops if the rest state is determined or the other direction is desired. If the other direction is desired, the cursor will move in that direction with initial slowness until a more consistent instruction in that direction is received (this helps deal with occasional inaccuracy in instruction, and generally makes it easier for the user to move the mouse cursor in the direction desired. There are several programming languages that can be used for this aspect of the project, but it the same language, MATLAB, has been used for the machine learning and the GUI parts of the project.

## 12. Experimental Procedure and Results

To use Anima, one must first set up the electronics. The neutral electrode is to be placed approximately 1 cm above the user's eyebrow. Afterwards, two differential electrodes should be placed each approximately 2 cm above the user's left and right ears. Finally, the third differential electrode will be placed around the center of the user's head looking down, along a line formed by the user's two ears and two differential electrodes. These electrodes must be plugged into the active electrode PCBs, which must then be plugged into the Arduino. Finally, the Arduino and electrodes should be powered by a power supply or battery pack, and the computer should connect to the Arduino via USB. This experimental test setup is seen in Figure 26, and more detailed connections are shown in Figure 27.



Figure 26: Example Test Setup of Anima. 3 Active electrodes connect three differential electrodes and a neutral electrode to the arduino, which then sends such data onwards towards the computer for further processing.

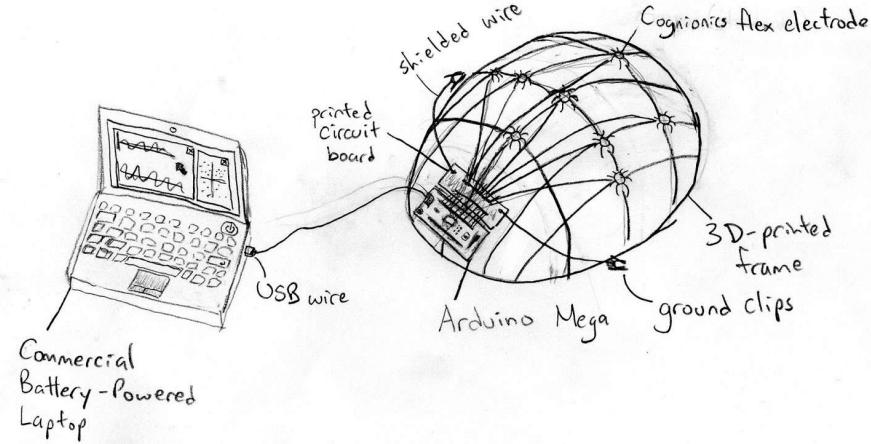


Figure 27: A sketch of Anima (early design) and its connections

Afterwards, one must first collect data, to be used to train the neural network. This data is collected in 6.5 minute rounds, each of which collects 6 minutes of data. Each round starts with 30 seconds in which the user is requested to relax, so that their brain waves can stabilize. After this, they are requested to think about moving left for two minutes, then resting for two minutes, and finally thinking right for two minutes. This 6.5 minute training session must be repeated at least ten times, for a total training time of 65 minutes and 1 hour of data. This data will be saved on disk between every 6.5 minute round in case of power failure or other loss of data.

After collecting at least an hour of data, a neural network must be trained using that data. After training the neural network, the state of the neural network will be saved for future use. Finally, one may use Anima to move a mouse by running a final program, which uses the trained neural network in order to move the mouse.

The neural network was trained using an hour's worth of collected brainwaves. Using pre-recorded data and the best configuration from the tests previously mentioned in section 10.2 to test the neural network a maximum classification accuracy of 41.9% (Figure 21) was achieved. Using real time data Anima was able to achieve much higher accuracies (usually above 60%). Given a well rested and alert user, Anima's classification accuracy attains 75.5% under relatively good conditions. The fact that real time data returned better results than the static data shows how much providing the user with direct visual feedback is important in maximizing Anima's accuracy.

## Conclusion

A fully functional EEG has been completed and verified to move a computer cursor via use of a multilayer neural network. The EEG's active sensor circuit was fully designed, laid out on a PCB design, soldered, and fully tested. Frequency response of the constructed active electrodes was verified and met all requirements. Artifacts were also digitally filtered further, and resulting signals of interest were sent through a classifier, which successfully moved a computer cursor. Therefore, Anima successfully demonstrated movement of a computer cursor in one dimension through the use of reading and analyzing brain waves.

## Works Cited

- [1] Q. Wang, O. Sourina, and M.K. Nguyen, “EEG-Based ‘Serious’ Games Design for Medical Applications”, IEEE 2010 International Conference on Cyberworlds, October 2010. Available: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5656236](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5656236), <http://www3.ntu.edu.sg/home/eosourina/Papers/PID1397361.pdf>.
- [2] B.H. Kim, M. Kim, S. Jo, “Quadcopter Flight Control using a Low-Cost Hybrid Interface with EEG-based Classification and Eye Tracking”, Computers in Biology and Medicine, vol. 51, pp. 82-92, August 2014. Available: [http://www.computersinbiologyandmedicine.com/article/S0010-4825\(14\)00106-1/abstract](http://www.computersinbiologyandmedicine.com/article/S0010-4825(14)00106-1/abstract), <http://isn1.kaist.ac.kr/mediawiki/paper/cbm14.pdf>.
- [3] K. Jokela, L. Puranen, O.P. Gandhi, “Radio Frequency Currents Induced in the Human Body for Medium-Frequency/High-Frequency Broadcast Antennas”, Health Phys, vol. 51, no. 3, pp. 237-244. Available: <https://www.ncbi.nlm.nih.gov/pubmed/8106240>, Accessed on: May 1, 2017.
- [4] Lobes of the human brain Image [Online]. Available: <http://www.mayoclinic.org/brain-lobes/img-20008887>, Accessed on: May 1, 2017.
- [5] Khare, V., Santhosh, J., Anand, S., & Bhatia, M., “Classification of five mental tasks from EEG data using neural networks based on Principal Component Analysis,” *IUP J Sci Technol*, vol. 5, no. 4, pp. 31-38, 2009.
- [6] Annetta, F., Sean D., Jonah G., Oluwadamilola O., “Cerebro: A Practical Brain-Computer Interface,” unpublished.
- [7] M. A. Lopez-Gordo, D. Sanchez-Morillo, F. Pelayo Valle. “Dry EEG Electrodes”, *sensors*, 2014.
- [8] Dry Electrode [Online]. Available: [http://www.neuroelectrics.com/products/electrodes/drytrode/&sa=D&ust=1492232862813000&usg=AFQjCNEmkhJzdwKLJ-Yuz-GL\\_8S\\_TS9\\_ZQ](http://www.neuroelectrics.com/products/electrodes/drytrode/&sa=D&ust=1492232862813000&usg=AFQjCNEmkhJzdwKLJ-Yuz-GL_8S_TS9_ZQ), Accessed on: May 1, 2017.
- [9] Wet electrode [Online]. Available: [http://www.neuroelectrics.com/products/electrodes/geltrode/&sa=D&ust=1494308994817000&usg=AFQjCNGO-w0MKqJXe2oazuJTgiZI1-\\_Lg](http://www.neuroelectrics.com/products/electrodes/geltrode/&sa=D&ust=1494308994817000&usg=AFQjCNGO-w0MKqJXe2oazuJTgiZI1-_Lg), Accessed on: May 1, 2017.

[10] Gel electrodes [Online]. Available:

<http://www.neuroelectrics.com/products/electrodes/solid-gel-consumable/&sa=D&ust=1494310954137000&usg=AFQjCNGFF1bVob5C-qUlfCLUotJMRI1KQ>, Accessed on: May 1, 2017.

[11] Cognionics, “Flexible Dry EEG Electrode”, Cognionics Flexible Dry EEG Electrode Datasheet, [Online], Available: [http://www.cognionics.com/images/Cognionics/spec\\_flex.pdf](http://www.cognionics.com/images/Cognionics/spec_flex.pdf), Accessed on: May 1, 2017.

[12] Flex Sensors (for hair) [Online]. Available:

<http://www.cognionics.com/images/Cognionics/flex%20sensors.jpg>, Accessed on: May 1, 2017.

[13] AnalogRead(), *Arduino* (2016), [Online]. Available:

<https://www.arduino.cc/en/Reference/AnalogRead>, Accessed on: Oct 15, 2016.

[14] Karungaru, S., et al. "Monotonous Tasks And Alcohol Consumption Effects On The Brain By Eeg Analysis Using Neural Networks." *Int J Comput Intell Appl* 11.3 (2012): -1. Applied Science & Technology Source. Web. 13 Oct. 2016.

[15] Sinha, R., Yogender A., and Barda D. "Backpropagation Artificial Neural Network Detects Changes In Electro-Encephalogram Power Spectra Of Syncopic Patients." *J Med Syst* 31.1 (2007): 63-68. Engineering Source. Web. 14 Oct. 2016.

[16] Russell, S., and Peter N., *Artificial Intelligence: A Modern Approach*, New Jersey 1995

[17] Turnip, A., and Demi S. “The Performance Of EEG-P300 Classification Using Backpropagation Neural Networks.” *Mechatronics, Electrical Power & Vehicular Technology* 4.2 (2013): 81-88. Applied Science & Technology Source. Web. 12 Oct. 2016.

[18] Martin M., “A scaled conjugate gradient algorithm for fast supervised learning.” *Neural Networks*, 6:525--533, 1993.

## APPENDIX

### Specification sheets

Instrumentation amplifier (AD620):

AD620							
-------	--	--	--	--	--	--	--

### SPECIFICATIONS

Typical @ 25°C,  $V_s = \pm 15$  V, and  $R_L = 2\text{ k}\Omega$ , unless otherwise noted.

Table 2.

Parameter	Conditions	AD620A			AD620B			AD620S <sup>1</sup>			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
GAIN	$G = 1 + (49.4 \text{ k}\Omega/R_G)$										
Gain Range		1		10,000	1		10,000	1		10,000	
Gain Error <sup>2</sup>	$V_{out} = \pm 10$ V										
$G = 1$		0.03	0.10		0.01	0.02		0.03	0.10		%
$G = 10$		0.15	0.30		0.10	0.15		0.15	0.30		%
$G = 100$		0.15	0.30		0.10	0.15		0.15	0.30		%
$G = 1000$		0.40	0.70		0.35	0.50		0.40	0.70		%
Nonlinearity	$V_{out} = -10$ V to +10 V										
$G = 1\text{--}1000$	$R_L = 10 \text{ k}\Omega$	10	40		10	40		10	40		ppm
$G = 1\text{--}100$	$R_L = 2 \text{ k}\Omega$	10	95		10	95		10	95		ppm
Gain vs. Temperature	$G = 1$		10			10			10		ppm/°C
	$Gain > 1^2$		-50			-50			-50		ppm/°C
<hr/>											
VOLTAGE OFFSET											
Input Offset, $V_{osi}$	$V_s = \pm 5$ V to $\pm 15$ V	30	125		15	50		30	125		µV
Overtemperature	$V_s = \pm 5$ V to $\pm 15$ V		185			85			225		µV
Average TC	$V_s = \pm 5$ V to $\pm 15$ V	0.3	1.0		0.1	0.6		0.3	1.0		µV/°C
Output Offset, $V_{oso}$	$V_s = \pm 15$ V	400	1000		200	500		400	1000		µV
	$V_s = \pm 5$ V		1500			750			1500		µV
Overtemperature	$V_s = \pm 5$ V to $\pm 15$ V		2000			1000			2000		µV
Average TC	$V_s = \pm 5$ V to $\pm 15$ V	5.0	15		2.5	7.0		5.0	15		µV/°C
Offset Referred to the Input vs. Supply (PSR)	$V_s = \pm 2.3$ V to $\pm 18$ V										
$G = 1$	80	100		80	100		80	100			dB
$G = 10$	95	120		100	120		95	120			dB
$G = 100$	110	140		120	140		110	140			dB
$G = 1000$	110	140		120	140		110	140			dB
<hr/>											
INPUT CURRENT											
Input Bias Current		0.5	2.0		0.5	1.0		0.5	2		nA
Overtemperature			2.5			1.5			4		nA
Average TC		3.0			3.0			8.0			pA/°C
Input Offset Current		0.3	1.0		0.3	0.5		0.3	1.0		nA
Overtemperature		1.5			0.75			2.0			nA
Average TC		1.5			1.5			8.0			pA/°C
<hr/>											
INPUT											
Input Impedance											
Differential Common-Mode			10  2		10  2			10  2			GΩ_pF
Input Voltage Range <sup>3</sup>	$V_s = \pm 2.3$ V to $\pm 5$ V	$-V_s + 1.9$	$+V_s - 1.2$	$-V_s + 1.9$	$+V_s - 1.2$	$-V_s + 1.9$	$+V_s - 1.2$	$10  2$	$10  2$		GΩ_pF
Overtemperature		$-V_s + 2.1$	$+V_s - 1.3$	$-V_s + 2.1$	$+V_s - 1.3$	$-V_s + 2.1$	$+V_s - 1.3$	$+V_s - 1.3$	$+V_s - 1.3$		V
	$V_s = \pm 5$ V to $\pm 18$ V	$-V_s + 1.9$	$+V_s - 1.4$	$-V_s + 1.9$	$+V_s - 1.4$	$-V_s + 1.9$	$+V_s - 1.4$	$+V_s - 1.9$	$+V_s - 1.4$		V
Overtemperature		$-V_s + 2.1$	$+V_s - 1.4$	$-V_s + 2.1$	$+V_s - 2.1$	$-V_s + 2.3$	$+V_s - 1.4$	$+V_s + 2.3$	$+V_s - 1.4$		V

## AD620

<b>Parameter</b>	<b>Conditions</b>	AD620A			AD620B			AD620S <sup>1</sup>			<b>Unit</b>
		<b>Min</b>	<b>Typ</b>	<b>Max</b>	<b>Min</b>	<b>Typ</b>	<b>Max</b>	<b>Min</b>	<b>Typ</b>	<b>Max</b>	
<b>Common-Mode Rejection</b>											
Ratio DC to 60 Hz with 1 kΩ Source Imbalance	$V_{CM} = 0 \text{ V}$ to $\pm 10 \text{ V}$										
G = 1		73	90		80	90		73	90		dB
G = 10		93	110		100	110		93	110		dB
G = 100		110	130		120	130		110	130		dB
G = 1000		110	130		120	130		110	130		dB
<b>OUTPUT</b>											
Output Swing	$R_L = 10 \text{ k}\Omega$ $V_S = \pm 2.3 \text{ V}$ to $\pm 5 \text{ V}$	$-V_S + 1.1$	$+V_S - 1.2$	$-V_S + 1.1$	$+V_S - 1.2$	$-V_S + 1.1$	$+V_S - 1.2$				V
Overtemperature	$V_S = \pm 5 \text{ V}$ to $\pm 18 \text{ V}$	$-V_S + 1.4$ $-V_S + 1.2$	$+V_S - 1.3$ $+V_S - 1.4$	$-V_S + 1.4$ $-V_S + 1.2$	$+V_S - 1.3$ $+V_S - 1.4$	$-V_S + 1.6$ $-V_S + 1.2$	$+V_S - 1.3$ $+V_S - 1.4$				V
Overtemperature		$-V_S + 1.6$	$+V_S - 1.5$	$-V_S + 1.6$	$+V_S - 1.5$	$-V_S + 2.3$	$+V_S - 1.5$				V
Short Circuit Current			$\pm 18$		$\pm 18$		$\pm 18$				mA
<b>DYNAMIC RESPONSE</b>											
Small Signal –3 dB Bandwidth											
G = 1			1000			1000			1000		kHz
G = 10			800			800			800		kHz
G = 100			120			120			120		kHz
G = 1000			12			12			12		kHz
Slew Rate		0.75	1.2		0.75	1.2		0.75	1.2		V/μs
Settling Time to 0.01%	10 V Step										
G = 1–100			15			15			15		μs
G = 1000			150			150			150		μs
<b>NOISE</b>											
Voltage Noise, 1 kHz											
Total RTI Noise = $\sqrt{(e_{ni}^2) + (e_{no}^2/G)^2}$											
Input, Voltage Noise, $e_{ni}$		9	13		9	13		9	13		nV/√Hz
Output, Voltage Noise, $e_{no}$		72	100		72	100		72	100		nV/√Hz
RTI, 0.1 Hz to 10 Hz											
G = 1		3.0			3.0	6.0		3.0	6.0		μV p-p
G = 10		0.55			0.55	0.8		0.55	0.8		μV p-p
G = 100–1000		0.28			0.28	0.4		0.28	0.4		μV p-p
Current Noise	f = 1 kHz	100			100			100			fA/√Hz
0.1 Hz to 10 Hz		10			10			10			pA p-p
<b>REFERENCE INPUT</b>											
$R_{IN}$		20			20			20			kΩ
$I_{IN}$	$V_{IN+}, V_{REF} = 0$	50	60		50	60		50	60		μA
Voltage Range		$-V_S + 1.6$	$+V_S - 1.6$		$-V_S + 1.6$	$+V_S - 1.6$		$-V_S + 1.6$	$+V_S - 1.6$		V
Gain to Output		$1 \pm 0.0001$		$1 \pm 0.0001$			$1 \pm 0.0001$				
<b>POWER SUPPLY</b>											
Operating Range <sup>4</sup>		$\pm 2.3$	$\pm 18$		$\pm 2.3$	$\pm 18$		$\pm 2.3$	$\pm 18$		V
Quiescent Current	$V_S = \pm 2.3 \text{ V}$ to $\pm 18 \text{ V}$	0.9	1.3		0.9	1.3		0.9	1.3		mA
Overtemperature		1.1	1.6		1.1	1.6		1.1	1.6		mA
TEMPERATURE RANGE											
For Specified Performance		–40 to +85			–40 to +85			–55 to +125			°C

<sup>1</sup> See Analog Devices military data sheet for 883B tested specifications.<sup>2</sup> Does not include effects of external resistor  $R_G$ .<sup>3</sup> One input grounded, G = 1.<sup>4</sup> This is defined as the same supply range that is used to specify PSR.

## Operational amplifiers (LM741):

### LM741

SNOSC25D – MAY 1998–REVISED OCTOBER 2015



[www.ti.com](http://www.ti.com)

## 6 Specifications

### 6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)<sup>(1)(2)(3)</sup>

		MIN	MAX	UNIT
Supply voltage	LM741, LM741A	±22		V
	LM741C	±18		
Power dissipation <sup>(4)</sup>		500	mW	
Differential input voltage		±30	V	
Input voltage <sup>(5)</sup>		±15	V	
Output short circuit duration		Continuous		
Operating temperature	LM741, LM741A	-50	125	°C
	LM741C	0	70	
Junction temperature	LM741, LM741A		150	°C
	LM741C		100	
Soldering information	PDIP package (10 seconds)		260	°C
	CDIP or TO-99 package (10 seconds)		300	
Storage temperature, T <sub>stg</sub>		-65	150	°C

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) For military specifications see RETS741X for LM741 and RETS741AX for LM741A.
- (3) If Military/Aerospace specified devices are required, please contact the TI Sales Office/Distributors for availability and specifications.
- (4) For operation at elevated temperatures, these devices must be derated based on thermal resistance, and T<sub>j</sub> max. (listed under "Absolute Maximum Ratings"). T<sub>j</sub> = T<sub>A</sub> + (θ<sub>JA</sub> P<sub>D</sub>).
- (5) For supply voltages less than ±15 V, the absolute maximum input voltage is equal to the supply voltage.

### 6.2 ESD Ratings

		VALUE	UNIT
V <sub>(ESD)</sub>	Electrostatic discharge	Human body model (HBM), per ANSI/ESDA/JEDEC JS-001 <sup>(1)</sup>	±400 V

- (1) Level listed above is the passing level per ANSI, ESDA, and JEDEC JS-001. JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

### 6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
Supply voltage (VDD-GND)	LM741, LM741A	±10	±15	±22	V
	LM741C	±10	±15	±18	
Temperature	LM741, LM741A	-55		125	°C
	LM741C	0		70	

### 6.4 Thermal Information

THERMAL METRIC <sup>(1)</sup>	LM741			UNIT	
	LMC (TO-99)	NAB (CDIP)	P (PDIP)		
	8 PINS	8 PINS	8 PINS		
R <sub>θJA</sub>	Junction-to-ambient thermal resistance	170	100	100	°C/W
R <sub>θJC(top)</sub>	Junction-to-case (top) thermal resistance	25	—	—	°C/W

- (1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, [SPRA953](#).



LM741

www.ti.com

SNOSC25D – MAY 1998 – REVISED OCTOBER 2015

### 6.5 Electrical Characteristics, LM741<sup>(1)</sup>

PARAMETER	TEST CONDITIONS		MIN	TYP	MAX	UNIT
Input offset voltage	$R_S \leq 10 \text{ k}\Omega$	$T_A = 25^\circ\text{C}$ $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		1	5	mV
					6	mV
Input offset voltage adjustment range		$T_A = 25^\circ\text{C}, V_S = \pm 20 \text{ V}$		$\pm 15$		mV
Input offset current		$T_A = 25^\circ\text{C}$ $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		20	200	nA
				85	500	
Input bias current		$T_A = 25^\circ\text{C}$ $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		80	500	nA
					1.5	$\mu\text{A}$
Input resistance		$T_A = 25^\circ\text{C}, V_S = \pm 20 \text{ V}$	0.3	2		$\text{M}\Omega$
Input voltage range		$T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$	$\pm 12$	$\pm 13$		V
Large signal voltage gain	$V_S = \pm 15 \text{ V}, V_O = \pm 10 \text{ V}, R_L \geq 2 \text{ k}\Omega$	$T_A = 25^\circ\text{C}$ $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$	50	200		V/mV
			25			
Output voltage swing	$V_S = \pm 15 \text{ V}$	$R_L \geq 10 \text{ k}\Omega$ $R_L \geq 2 \text{ k}\Omega$	$\pm 12$	$\pm 14$		V
			$\pm 10$	$\pm 13$		
Output short circuit current		$T_A = 25^\circ\text{C}$		25		mA
Common-mode rejection ratio	$R_S \leq 10 \Omega, V_{CM} = \pm 12 \text{ V}, T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		80	95		dB
Supply voltage rejection ratio	$V_S = \pm 20 \text{ V} \text{ to } V_S = \pm 5 \text{ V}, R_S \leq 10 \Omega, T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		86	96		dB
Transient response	Rise time	$T_A = 25^\circ\text{C}$ , unity gain		0.3		$\mu\text{s}$
	Overshoot			5%		
Slew rate		$T_A = 25^\circ\text{C}$ , unity gain		0.5		V/ $\mu\text{s}$
Supply current		$T_A = 25^\circ\text{C}$		1.7	2.8	mA
Power consumption	$V_S = \pm 15 \text{ V}$	$T_A = 25^\circ\text{C}$ $T_A = T_{A\text{MIN}}$ $T_A = T_{A\text{MAX}}$	50	85		
			60	100		mW
			45	75		

(1) Unless otherwise specified, these specifications apply for  $V_S = \pm 15 \text{ V}, -55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$  (LM741/LM741A). For the LM741C/LM741E, these specifications are limited to  $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ .

### 6.6 Electrical Characteristics, LM741A<sup>(1)</sup>

PARAMETER	TEST CONDITIONS		MIN	TYP	MAX	UNIT
Input offset voltage	$R_S \leq 50 \Omega$	$T_A = 25^\circ\text{C}$ $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		0.8	3	mV
					4	mV
Average input offset voltage drift				15		$\mu\text{V}/^\circ\text{C}$
Input offset voltage adjustment range		$T_A = 25^\circ\text{C}, V_S = \pm 20 \text{ V}$	$\pm 10$			mV
Input offset current		$T_A = 25^\circ\text{C}$ $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		3	30	nA
					70	
Average input offset current drift				0.5		$\text{nA}/^\circ\text{C}$
Input bias current		$T_A = 25^\circ\text{C}$ $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		30	80	nA
					0.21	$\mu\text{A}$
Input resistance		$T_A = 25^\circ\text{C}, V_S = \pm 20 \text{ V}$ $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}, V_S = \pm 20 \text{ V}$	1	6		$\text{M}\Omega$
				0.5		
Large signal voltage gain	$V_S = \pm 20 \text{ V}, V_O = \pm 15 \text{ V}, R_L \geq 2 \text{ k}\Omega$	$T_A = 25^\circ\text{C}$ $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$	50			V/mV
	$V_S = \pm 5 \text{ V}, V_O = \pm 2 \text{ V}, R_L \geq 2 \text{ k}\Omega, T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		32			
			10			

(1) Unless otherwise specified, these specifications apply for  $V_S = \pm 15 \text{ V}, -55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$  (LM741/LM741A). For the LM741C/LM741E, these specifications are limited to  $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ .

**LM741**

SNOSC25D –MAY 1998–REVISED OCTOBER 2015

www.ti.com

**Electrical Characteristics, LM741A<sup>(1)</sup> (continued)**

PARAMETER	TEST CONDITIONS		MIN	TYP	MAX	UNIT
Output voltage swing	$V_S = \pm 20 \text{ V}$	$R_L \geq 10 \text{ k}\Omega$	$\pm 16$		$\pm 15$	V
		$R_L \geq 2 \text{ k}\Omega$	$\pm 15$			
Output short circuit current	$T_A = 25^\circ\text{C}$			10	25	35
		$T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		10		40
Common-mode rejection ratio	$R_S \leq 50 \Omega$ , $V_{CM} = \pm 12 \text{ V}$ , $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$			80	95	dB
Supply voltage rejection ratio	$V_S = \pm 20 \text{ V}$ to $V_S = \pm 5 \text{ V}$ , $R_S \leq 50 \Omega$ , $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$			86	96	dB
Transient response	Rise time Overshoot	$T_A = 25^\circ\text{C}$ , unity gain			0.25	0.8
					6%	20%
Bandwidth <sup>(2)</sup>	$T_A = 25^\circ\text{C}$			0.437	1.5	MHz
Slew rate	$T_A = 25^\circ\text{C}$ , unity gain			0.3	0.7	V/μs
Power consumption	$V_S = \pm 20 \text{ V}$	$T_A = 25^\circ\text{C}$			80	150
		$T_A = T_{A\text{MIN}}$			165	mW
		$T_A = T_{A\text{MAX}}$			135	

(2) Calculated value from: BW (MHz) = 0.35/Rise Time (μs).

**6.7 Electrical Characteristics, LM741C<sup>(1)</sup>**

PARAMETER	TEST CONDITIONS		MIN	TYP	MAX	UNIT		
Input offset voltage	$R_S \leq 10 \text{ k}\Omega$	$T_A = 25^\circ\text{C}$	2		6	mV		
		$T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$			7.5	mV		
Input offset voltage adjustment range	$T_A = 25^\circ\text{C}$ , $V_S = \pm 20 \text{ V}$		$\pm 15$		200	mV		
Input offset current	$T_A = 25^\circ\text{C}$			20		nA		
		$T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$			300	nA		
Input bias current	$T_A = 25^\circ\text{C}$			80	500	nA		
		$T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$			0.8			
Input resistance	$T_A = 25^\circ\text{C}$ , $V_S = \pm 20 \text{ V}$		0.3		2	MΩ		
Input voltage range	$T_A = 25^\circ\text{C}$		$\pm 12$		$\pm 13$	V		
Large signal voltage gain	$V_S = \pm 15 \text{ V}$ , $V_O = \pm 10 \text{ V}$ , $R_L \geq 2 \text{ k}\Omega$	$T_A = 25^\circ\text{C}$	20		200	V/mV		
		$T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$	15					
Output voltage swing	$V_S = \pm 15 \text{ V}$	$R_L \geq 10 \text{ k}\Omega$	$\pm 12$		$\pm 14$	V		
		$R_L \geq 2 \text{ k}\Omega$	$\pm 10$		$\pm 13$			
Output short circuit current	$T_A = 25^\circ\text{C}$		25		mA			
Common-mode rejection ratio	$R_S \leq 10 \text{ k}\Omega$ , $V_{CM} = \pm 12 \text{ V}$ , $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		70			dB		
Supply voltage rejection ratio	$V_S = \pm 20 \text{ V}$ to $V_S = \pm 5 \text{ V}$ , $R_S \leq 10 \Omega$ , $T_{A\text{MIN}} \leq T_A \leq T_{A\text{MAX}}$		77		96	dB		
Transient response	Rise time Overshoot	$T_A = 25^\circ\text{C}$ , Unity Gain			0.3	μs		
					5%			
Slew rate	$T_A = 25^\circ\text{C}$ , Unity Gain		0.5		2.8	V/μs		
Supply current	$T_A = 25^\circ\text{C}$		1.7			mA		
Power consumption	$V_S = \pm 15 \text{ V}$ , $T_A = 25^\circ\text{C}$		50		85	mW		

(1) Unless otherwise specified, these specifications apply for  $V_S = \pm 15 \text{ V}$ ,  $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$  (LM741/LM741A). For the LM741C/LM741E, these specifications are limited to  $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ .

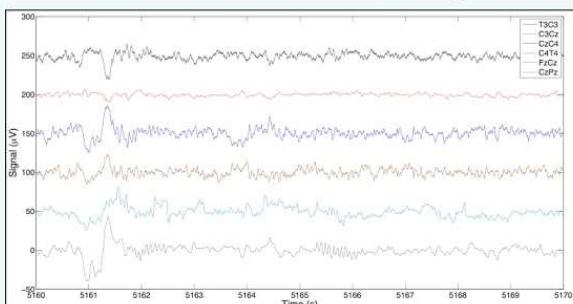
Sensors (Cognionics flex):

## COGNIONICS FLEXIBLE DRY EEG ELECTRODE

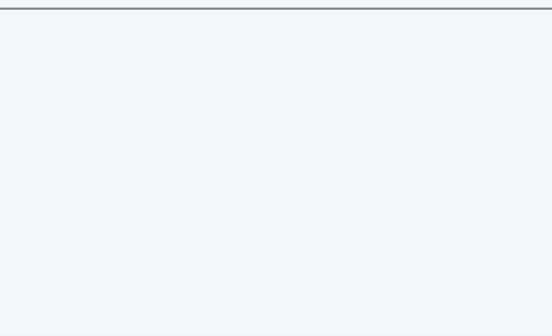


- Designed to work through hair with no gels, liquids or clean up
- Flexible legs slide under hair strands for scalp contact
- Flattens under hard pressure for maximum user comfort
- Optimized materials and coatings to ensure low-noise signals
- Designed for use with Cognionics mobile headsets, sensor amplifiers and wireless data acquisition systems
- Patent-pending technology

**Low Noise EEG Recordings**

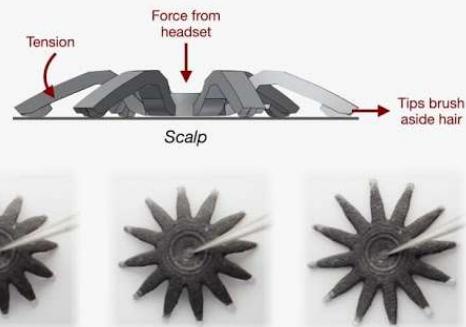


Sample EEGs from sleeping subject



Sample EEGs from a 10-20 array on an awake subject (eyes closed)

**Advanced Sensor Construction**



- Cognionics has developed an innovative EEG sensor specifically designed to enable a practical dry EEG system:
  - ✓ Novel shape optimized for hair penetration and scalp contact
  - ✓ Special silver coating imparts optimal conductivity to the sensor's surface for the best possible signal transduction
  - ✓ Flexible material safely flattens under pressure - no hard metal pins
  - ✓ Fits with Cognionics EEG headsets for rapid setup and reliable, long-term recordings

Impedance	100 - 2000 KΩ on unprepared skin
Material	Conductive Elastomer
Contact	Ag/AgCl
Connector	Cognionics Mini-snap

**For Information, Help and Pricing:**

**Cognionics**  
info@cognionics.com

**BRAIN VISION LLC**  
Solutions for neurophysiological research  
sales@brainvision.com

Cognionics sensors are for research and evaluation only. Signals should not be used for medical diagnosis.