

# ***Neuromancer***

*A wireless EEG machine designed for easy BCI implementation*

## **Final Design Report**

EEL 4924 – Electrical Engineering Design II

April 18, 2011

Chris Carter, Michael Carnevale, Kenny Orkis

## **Abstract**

Our project consists of designing a wireless EEG machine that will be ready for easy BCI integration. Brain signals will be captured with electrodes connected to buffer amplifiers. The signals will be passed from the buffer to an analog amplifying circuit. The amplified signal will be passed to a digital device that samples the data and transmits it via serial communication. The serial data will be transmitted wirelessly to a PC that will read in the data and display it through various methods for easy data analysis. The data will be displayed in a GUI created with MATLAB. The GUI will also identify actions using a BCI algorithm and output the data via an emulated serial port. The active electrode amplifier circuit, digital sampling circuit, wireless transmission system, and PC software will be designed by our group. The analog amplifier will be purchased. The BCI algorithm will be taken from other researchers.

# Introduction

The neuromancer is a wireless EEG device. The headset will require minimal set up time, no assembly, and easy to use software. The whole package will allow for the analysis of individual brainwave channels as well as the isolation of key frequency bands for more in depth signal analysis. The project will require a firm understanding of both hardware and software. The hardware will involve both analog and digital components. The software will involve serial data communication, wireless transmission, digital signal processing, computational learning algorithms. The analog amplifier will be purchased. The BCI algorithm will be taken from other researchers.

# Features

The primary focus in feature selection during the design of the neuromancer was BCI integration and application. Its features consist of:

- Active Electrodes with pin array contacts to allow for through-hair penetration and signal detection.
- Wireless communication with PC which gives mobility to the user while simultaneously reduces noise by removing long cables which indirectly function as noise amplifying antennae.
- User friendly software with GUI designed to enhance the experience of first time users. GUI will allow for an extracted features output for BCI applications.
- Low profile package design reduces bulk and weight to allow for greater mobility.
- Li-Po batteries, battery charger, and cascaded voltage regulation for operating life of approximately 1.5 hours.

## Competition

There are several on market devices similar to the neuromancer that span a wide range of functionality and prices. On the one side of the spectrum is NeuroSky and Mattel's MindFlex game which argues that it's users can direct a levitating ball through an obstacle course by focusing the alpha brain waves of the frontal lobe. The device retails for \$60.00, has no software for signal analysis or verification, and has created several skeptics of the devices true BCI technology. On the other side is the Epop by Emotiv which offers 14 channels driven by wet active electrodes on a wireless helmet. Depending on the users financial constraints and applications, the user can select various software packages that allow for signal analysis, BCI interaction with Emotiv specific applications and games, or a development kit for designers and researchers; prices range from \$30 0.00 for the barebones to \$7,500.00 for the Enterprise Edition.

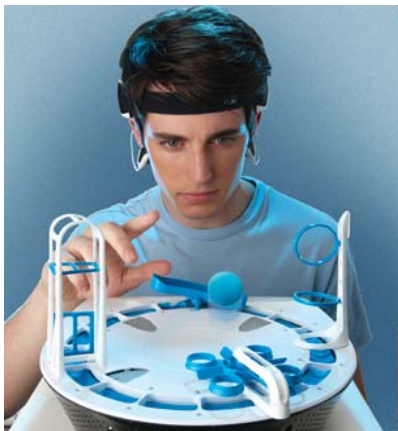


Figure 1 – MindFlex by NeuroSky

Figure 2 – Epop by Emotiv

## Technical Components

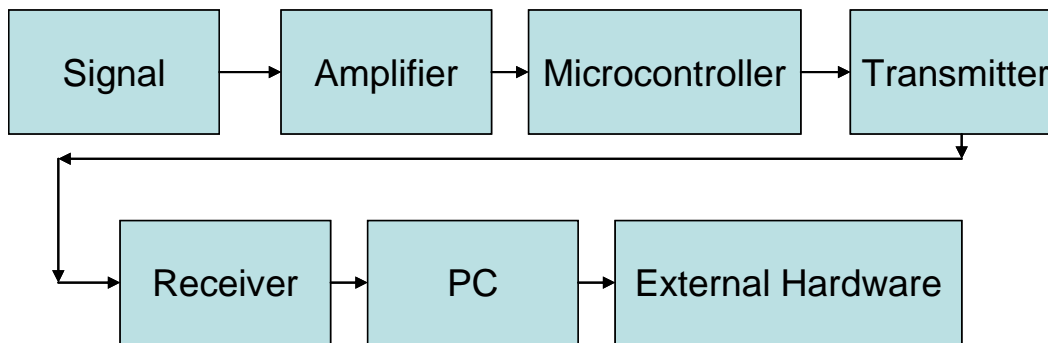


Figure 3 - Flowchart of Neuromancer Project

Pictured above is a flowchart describing the technical aspects surrounding neuromancer. An input brain signal is passed through an amplifier to ADC port into a microprocessor (the atmega 324p for our project.) The signal is then transmitted wirelessly to a computer with software that will analyze and separate the sampled frequency in real-time and allow for easy BCI incorporation. The technical aspects characteristic are as follows:

- Buffered pre-amplifier circuit to improve the detection of small signals (10~250  $\mu\text{V}$ ) in the presence of noise ( $\sim 10$  mV).
- Amplify the small signals while simultaneously attenuating the noise using common mode rejection, high gain amplification ( $G \sim 4000$ ), and low pass filters to prevent alias during ADC sampling.
- Digitally sample the signal ( $f_{\text{max}} = 60$  Hz) and transmit data to PC using both wireless Xbee modules as well as wired communication via an RS232 serial to USB converter chip.
- Computer software capable of doing high speed calculations on large quantities of data for real time frequency identification as well as exportation of the data for operation of BCI components also in real time.
- The external hardware is any BCI device the user might add in that can communicate with the PC's serial port.

## Division of Labor

Michael Carnevale is responsible for:

- Design circuit, schematic, and PCB layout for digital board which includes:
  - Multiple ADCs
  - Interrupt inputs
  - FTDI Chip for serial communication

- Programming the microprocessor
- Setting up the Emulator and interfacing it with MATLAB
- Help Ken design Active Electrode
- Help Ken redesign Amplifier Board

Chris Carter is responsible for:

- Design and build power circuit which includes:
  - Multiple Voltage Regulators
  - Rechargeable Li-Ion battery
  - Battery voltage level monitors
  - Recharging circuitry
- Design and build the helmet
- Help Michael program the microprocessor

Ken Orkis is responsible for:

- Design Active Electrode and cables
- Create MATLAB GUI
- Create MATLAB functions for reading in digital board data, time and frequency signal analysis, polycoherence analysis, BCI algorithm
- Design wireless transmitters

# Gantt Chart

Project Neuromancer Spring 2011  
Chris (C), Michael (C), Kenny (O)

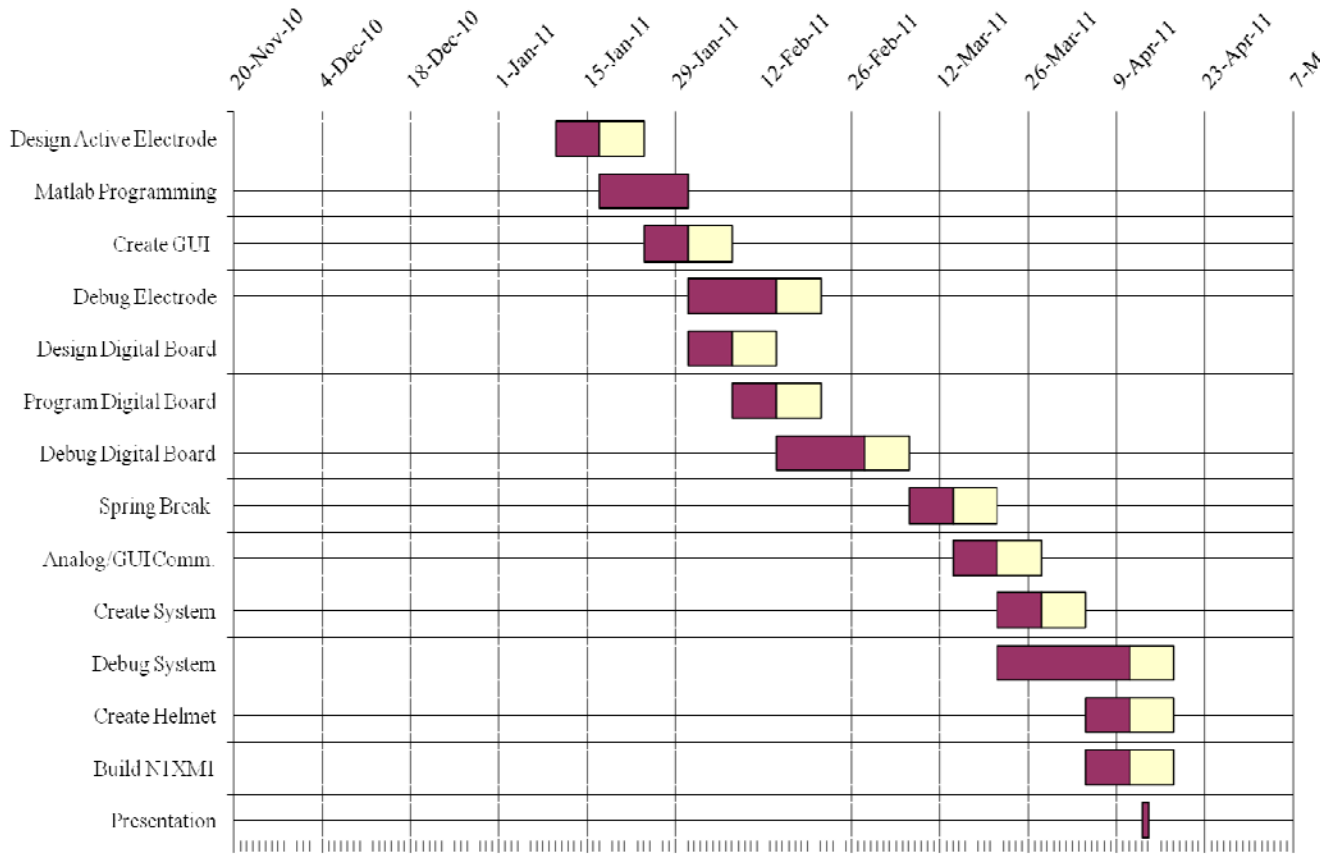


Figure 4 - Gantt Chart

# Active Electrodes

## Purpose:

The pre-amplifier electrode circuit has many benefits in the EEG system. EEG signals are measured differentially from two points on the scalp. This can be done in several different ways but considering the high noise to signal ratio it has been found to best be measured using a common mode rejection amplifier across the two points. The common mode rejection amplifier eliminates ambient noises and DC biases that can accumulate on the skin from EMG signals (DC voltages), mains hums (60 Hz), fluorescent lighting (120 Hz), etc. The signals in question are approximately 250  $\mu\text{Vpp}$ . Noise can be as high as 10 mV.

Ideally the CMR amplifier would be placed as close as possible to the points in considerations, in practice this is not so easy to accomplish as cables must be extended from the points of interest to the CMR amplifier. These cables, depending on their length can act as antennae for extra noise. The cables can also act as phase shifters if their lengths are not identical. They can also act as attenuators.

If different signals collect on the two lines then they will not be removed by the CMR, using short cables and amplifying the signals before they are transmitted will reduce the SNR and reduce the noise effects from the cables. The danger in pre-amplifying is that the signals must be equally amplified else a mismatch will cause the CMR amplifier to allow more noise through. To account for these issues, **resistors with a 0.1% tolerance were used, as were short cables.**

## Cable Length:

The cable acts as a high pass filtered antennae for miscellaneous signals. Longer cables allow, signals with longer wavelengths and lower frequencies to add noise to the system. The cables we used were 2 meters and 14 inches long with HPF cutoff frequencies of 164 MHz and 844 MHz respectively.

$$\lambda = \frac{v}{f},$$

$$V = 3 \times 10^8 \text{ m/s}$$

$$F_{\text{max}} = 10 \text{ MHz}$$

$$\text{Lambda} \sim 3 \text{ m}$$

## Pre-Amplifier Circuit:

The pre-amplifier circuit chosen was a non inverting amplifier with a gain of 11. This design was settled upon after testing and tweaking various circuit designs. The original circuit chosen was a unity gain buffered circuit with a bootstrapped positive feedback loop. The schematic was designed by Jeorg Hansmann and noted as a semi-reliable option that some members of the OpenEEG community have tested and had positive results with. The schematic is below in figure 1.

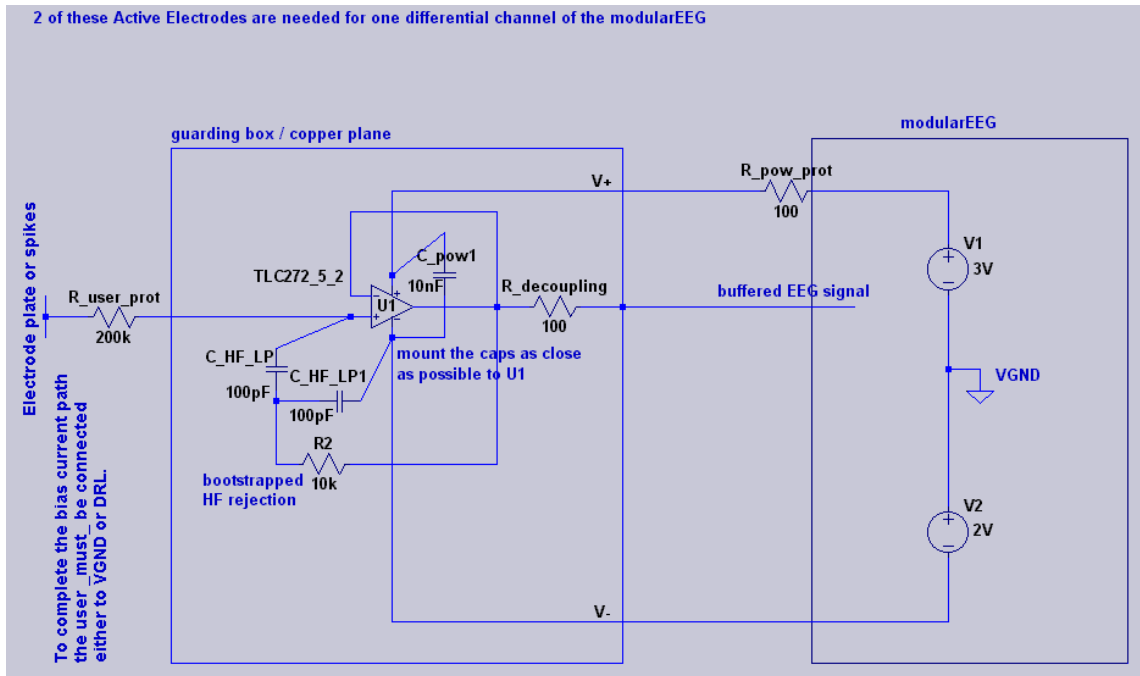
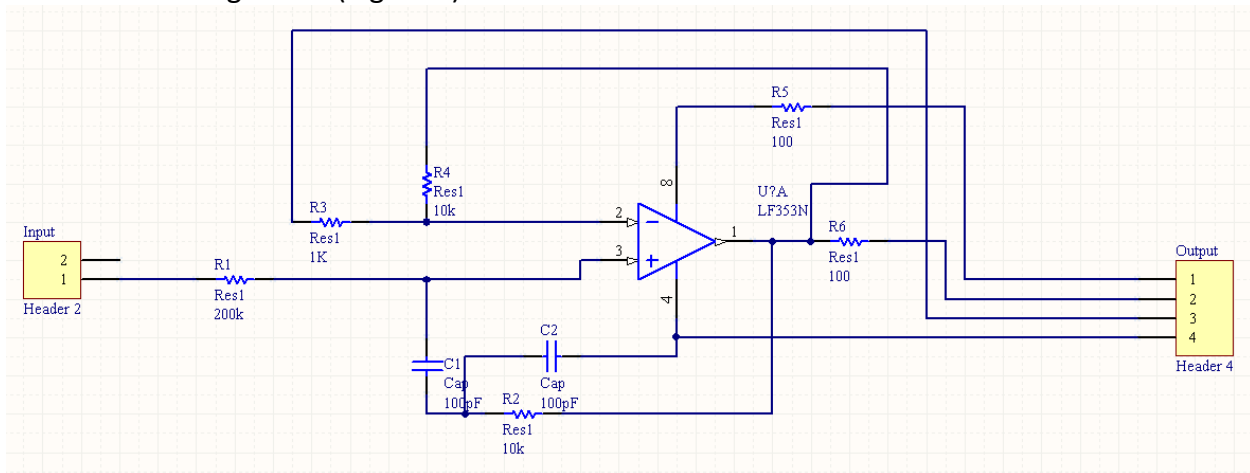


Figure 1 – Original Circuit created by Jeorg Hansmann (AEV1)

After much testing this amplifier appeared to be unreliable as too much noise was passed through the CMR amp and into the high gain amplifier. Testing showed that a 250  $\mu\text{V}$  signal could not be detected but a 2.5 mV signal could. Considering it was impossible to increase the brain's output signal, I changed the shorted negative feedback loop to a resistor chain so that the unity gain buffer turned into a non-inverting amplifier. I left the positive feedback as it was.

This yielded the following circuit (Figure 2).



This circuit immediately produced problems as a 30mV, 14 Hz test input signal created a rail-to-rail 5V 3.9 kHz output signal. This signal was determined to be the results of an unstable positive feedback loop created by adding in the negative feedback loop, an unexpected outcome.

Removing the positive feedback loop eliminated the oscillator and reverted the circuit back to its intended state, a non-inverting amp with a gain of 11. The 250  $\mu\text{V}$ , 14 Hz PWM test signal was then sent through the



electrode and was successfully identified in the frequency spectrum of the signal. The signal was then passed through two parallel active electrode pre-amplifier circuits, through the amplifier board to be identified at the output of the high gain amplifier.

Two of the electrodes were then connected to the hair pin arrays, epoxied and tested on the scalp. No brain waves were recognized at the output. The amplifier failed to send signals through from the signal amplifier as well. Discussions with Mike Stapleton led me to believe that the epoxy broke some of the board traces, short on time, I abandoned testing them further and focused on the other two non-epoxied board that were still functional. The other boards still passed signals from the function generator, they were connected to the pin array and tested for brain waves yet nothing was conclusive. The impedance between contacts was 300k which was expected and appropriate for the high impedance electrodes. Further test might have yielded a conclusion as to why the electrodes did not work but limited knowledge and time forced me to change focus and clean the software up for final demonstrations.

Below are some simulations from LT-spice which show how two sets of non-inverting amplifiers can be used to reference a voltage on the body without referencing a common ground. The first circuit schematic (figure 3) resembles the final electrode design. Notice how the signal is amplified by a factor of 11 when both the input and the output are measured with respect to ground. The second set of images (figure 4) shows a voltage difference between two inputs and the outputs being measured with respect to ground. In my design, the output signals are fed into the input terminals of a instrumentation amplifier which is used for CMR differential amplification of the signal to eliminate noise before being passed to the high gain amplifier. Notice how neither signal is referenced to ground, only to each other, yet the outputs are clearly visible.

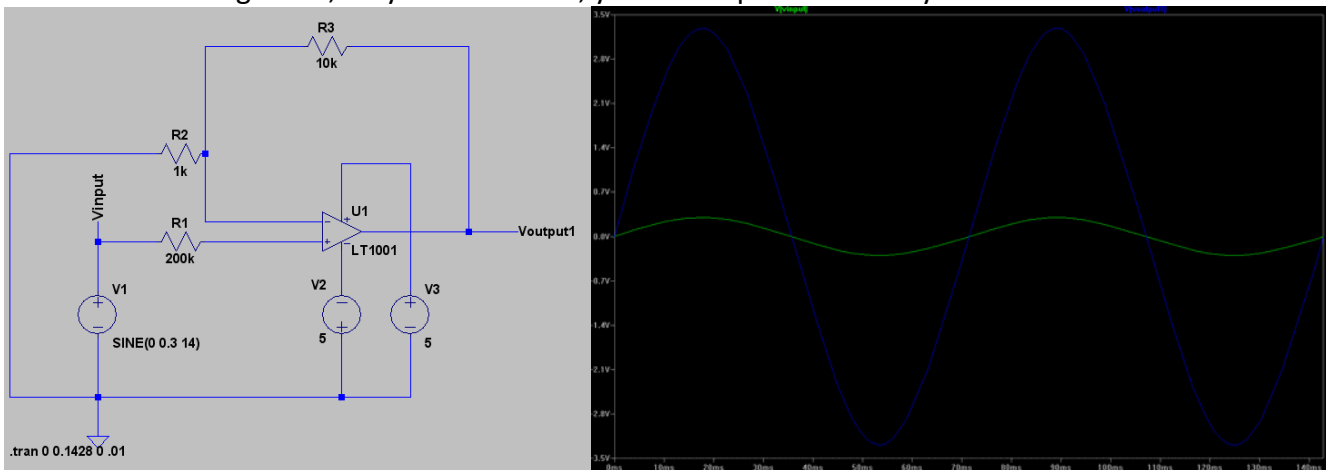


Figure 3 – Single non-inverting amplifier and signals at input/output

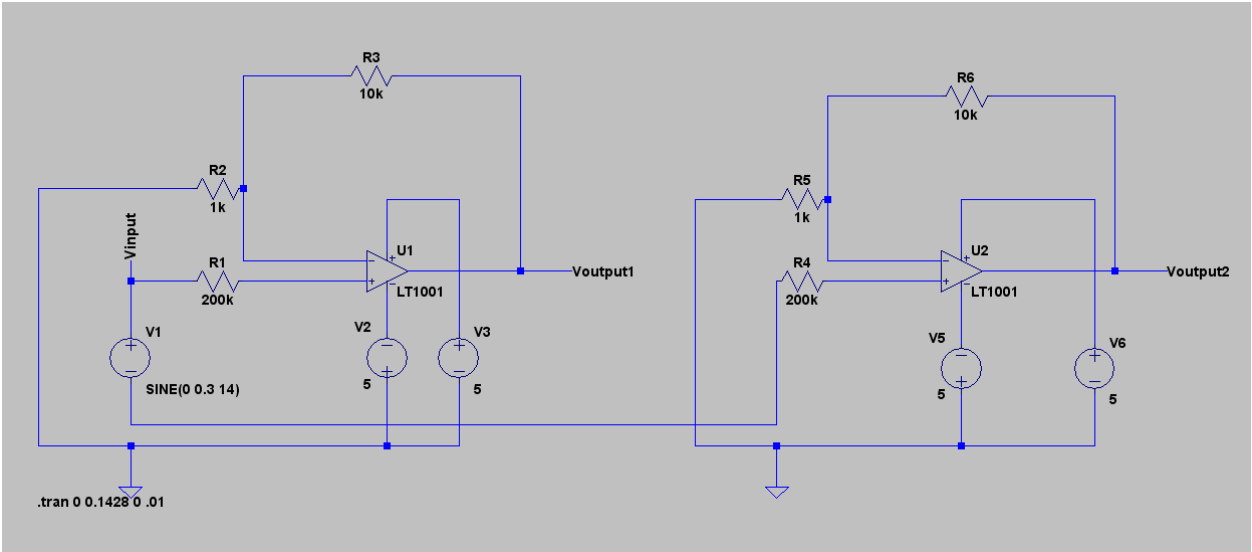


Figure 4a – Dual Non-inverting amplifiers and signals at input/output

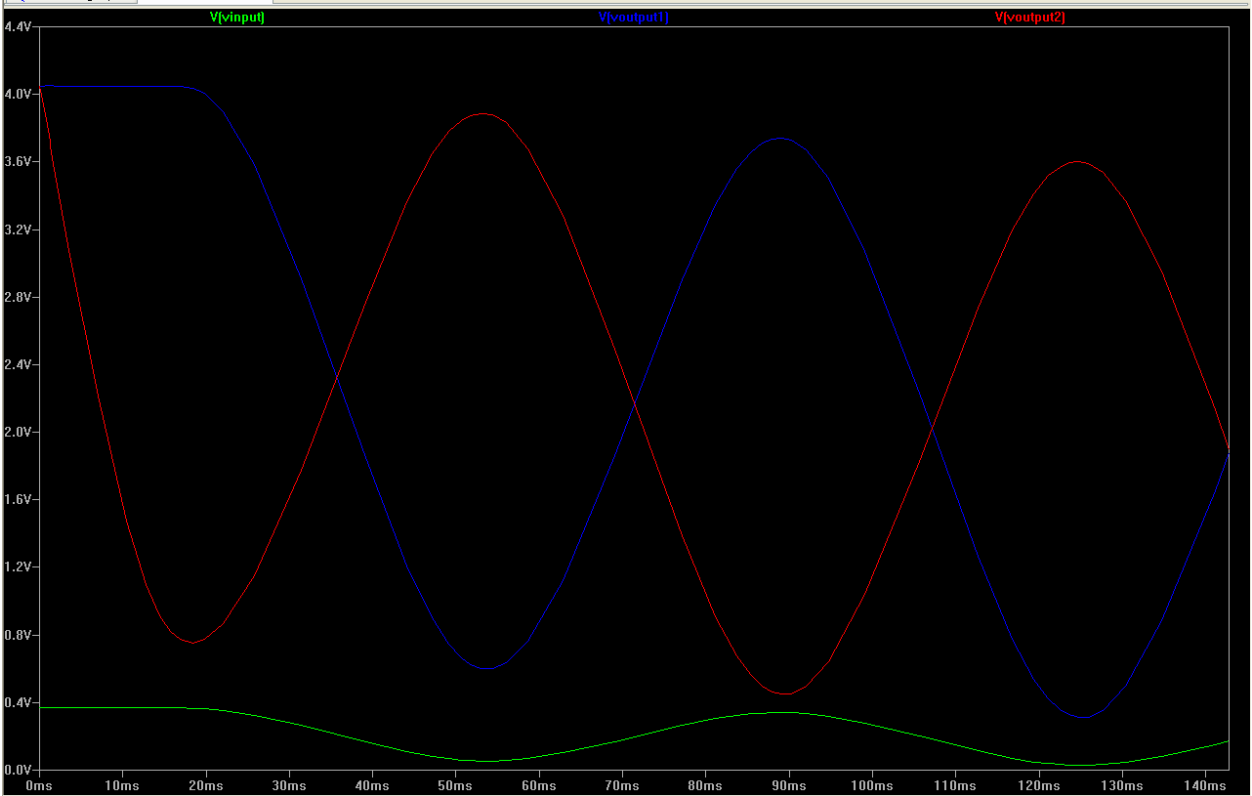


Figure 4b – Dual Non-inverting amplifiers and signals at input/output

# Amplifier Board

The amplifier board was designed by a collective group of individuals under the name of the OpenEEG project. The board is part of a device known as ModularEEG. The board can be purchased from Olimex populated or unpopulated. We purchased it populated. The board contains a three stage high gain amplifier with that uses common mode rejection to isolate the biosignals from the overwhelming noise the signal is enveloped in. Below is a schematic of the board as well as pictures of a 16 Hz square wave signal captured at the input stage, after the second stage, and after it has been passed through the final stage and a 60 Hz lowpass filter.

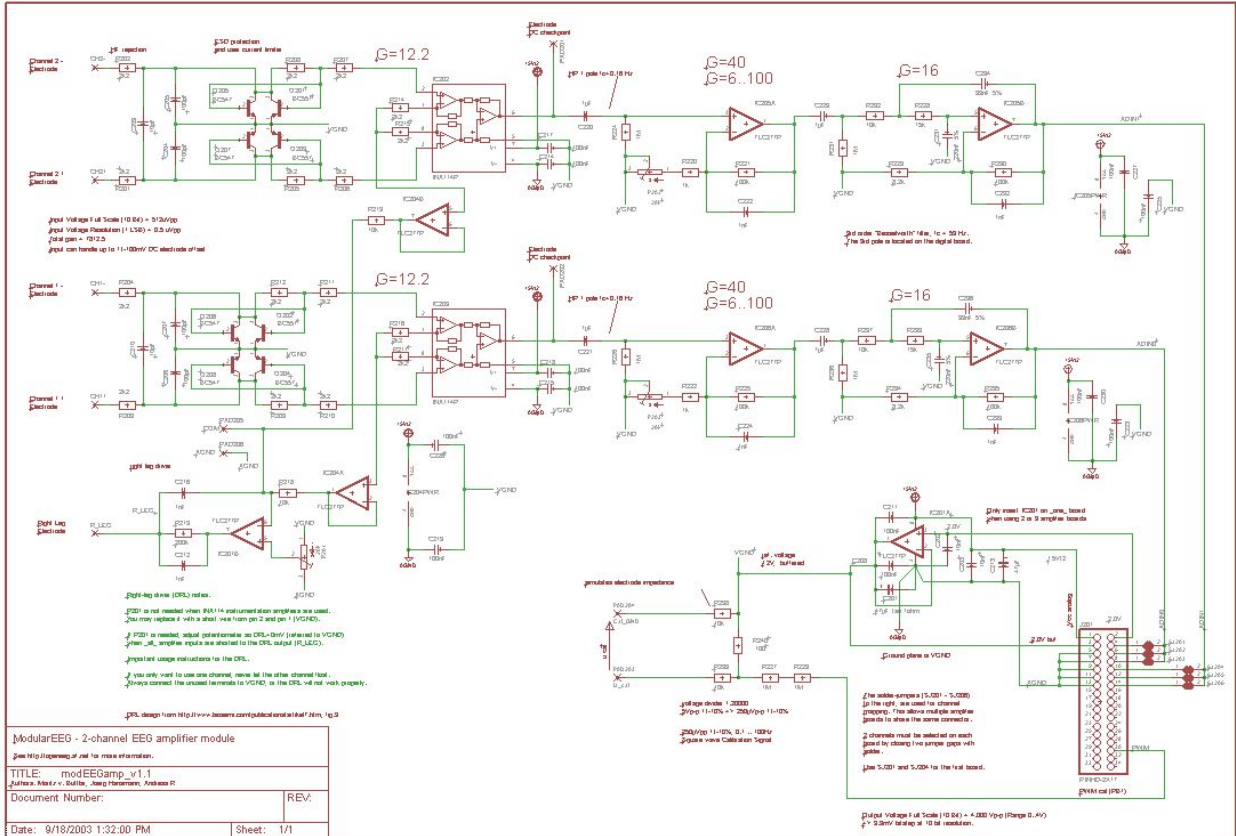


Figure 9 – ModularEEG Analog Board Amplifier Schematic

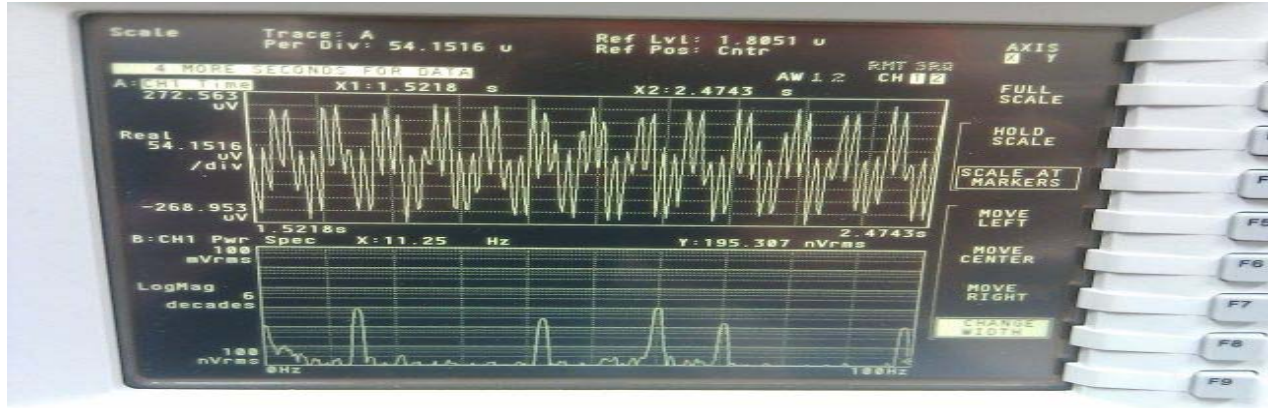


Figure 10 – 14 Hz Square wave at Analog Board input

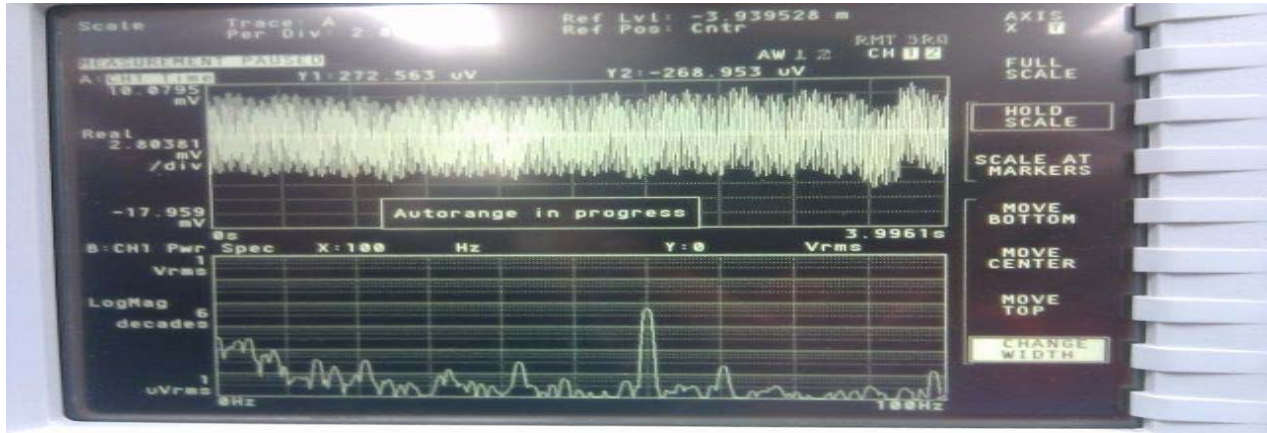


Figure 11 – 14 Hz Square wave after 2<sup>nd</sup> stage

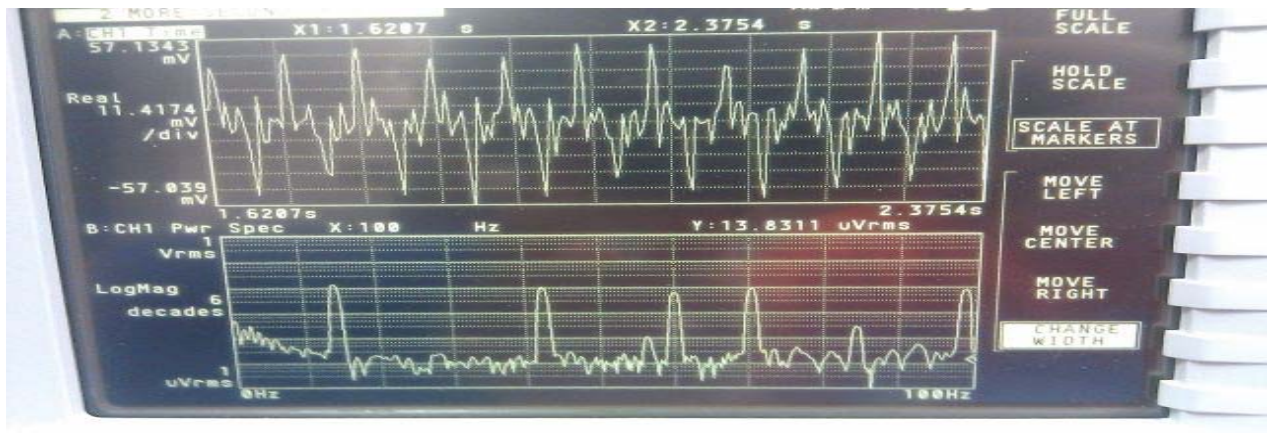


Figure 12 – 14 Hz Square wave after 60 Hz lowpass filter

## Analog Amplifier Board Breakdown

The analog board is used to read EEG input signals, amplify them, and eliminate common mode noise before passing it to the digital board for sampling, transmission, and data analysis. The board consists of two channels for measuring signals, a DRL feedback loop, a test signal generator, and an output header. Figure 1 shows an overview of the open source schematic of the board. Figure 2 highlights the major parts of the board for easy identification. The rest of the figures are used to individually isolate the specific part as it is described in greater detail.

## Analog Amplifier – Buy vs. Design



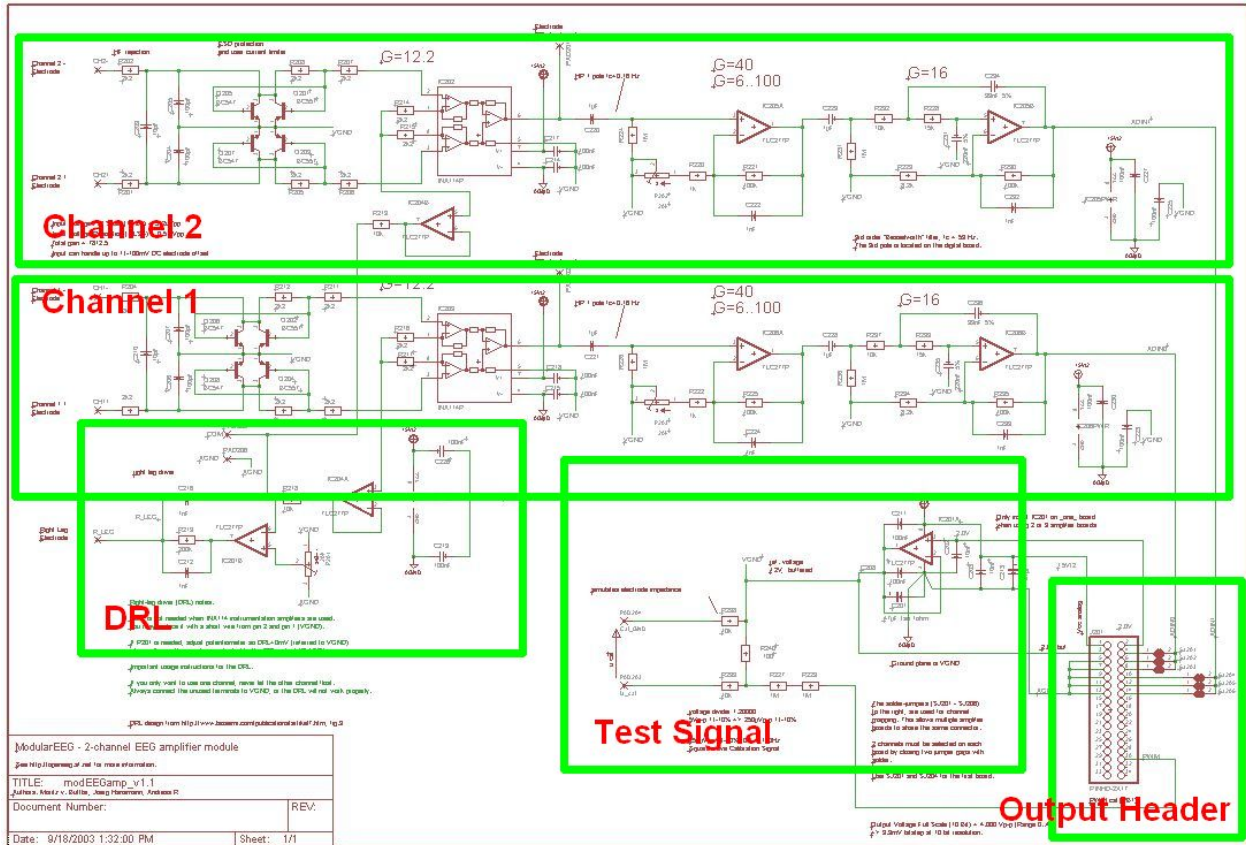


Figure 2 – Analog Board Schematic (Segmented)

## Analog Board – Output Header

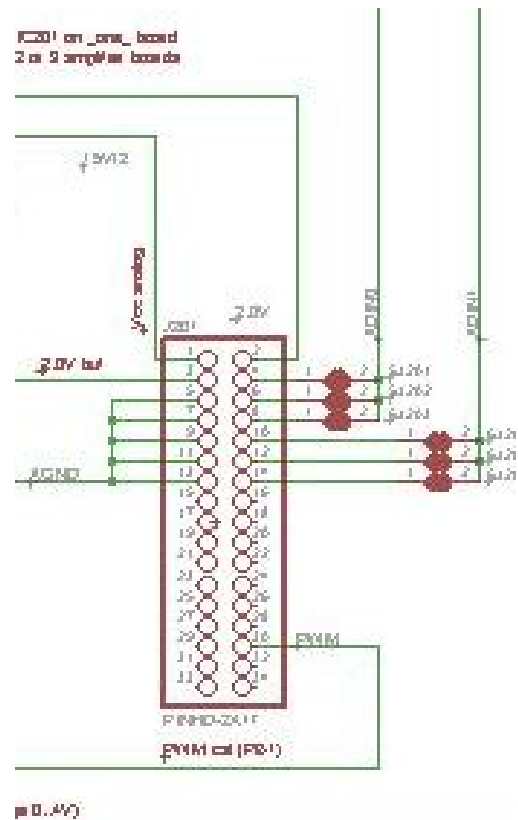


Figure 3 – Analog Board Schematic (Output Header)

The analog board receives signals from electrodes and cables connected to the test subject. The board receives power (+5V, +2V, GND) from the digital board which was designed with to a highly noise sensitive system. The board also receives a square wave from the digital board for the test signal function generator. The board sends the cleaned up signals to the digital board.

The Pinout for the header is as follows:

- Pin 1 – +5 V DC
- Pin 2 – +2 V DC
- Pin 3 – +2 V DC (Buffered)
- Pin 4 – Channel 1 Output
- Pin 5 – GND
- Pin 7 – GND
- Pin 9 – GND
- Pin 10 – Channel 2 Output
- Pin 11 – GND
- Pin 13 – GND
- Pin 30 – Square Wave for Test Signal Function Generator

## Analog Board – Test Signal Generator

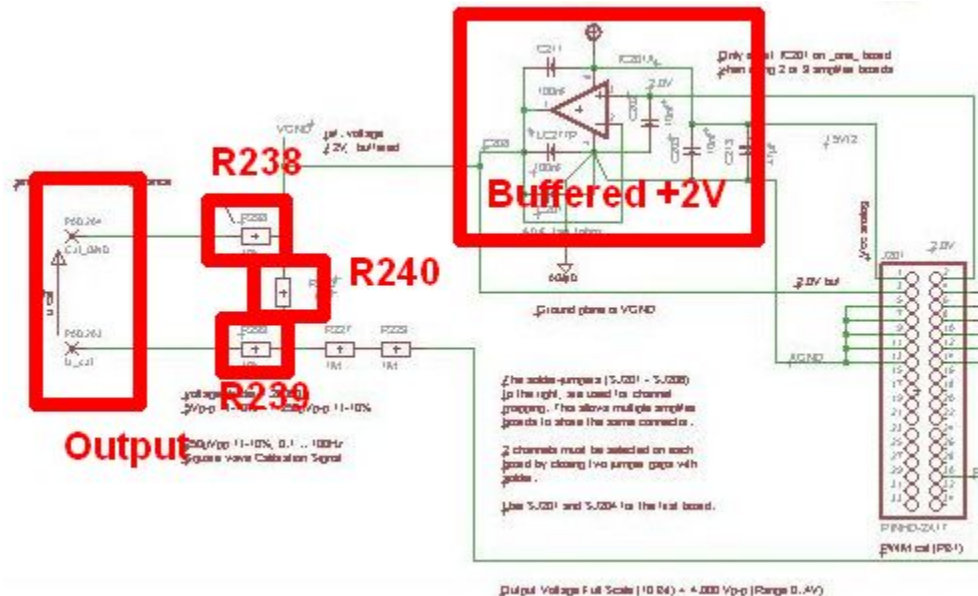


Figure 4 – Analog Board Schematic (Test Signal Generator Circuit)

As mentioned earlier, the EEG signals are very small and measured through common mode rejection amplifiers. Therefore the system power and all circuitry before the differential amplifier must be powered by noise free circuit (ideally). Each active component (IC) on the board needs power supplied from the digital board. The Vcc+ is powered by a +5V signal that is buffered on the digital board. The Vcc- is transmitted through the output header, on pin 2, it is +2V. The 2V signal is then buffered with the non-inverting amplifier (Sole IC in Figure 4) and reclassified as VGND.

The test signal generator takes a zero to 5V square wave signal created by the microprocessor and greatly attenuates it across a resistor divider. The output is a 250uV signal measured across R240. The signal is fed to a header for easy referencing. The nodes for measurement (the nodes connected to the header) are not directly connected across the resistor but passed through R238 and R239 first. Passing the signal through these resistors gives 20kΩ of output impedance (the same as properly prepared and abraded skin).

## Analog Board – DRL Circuit<sup>1</sup>



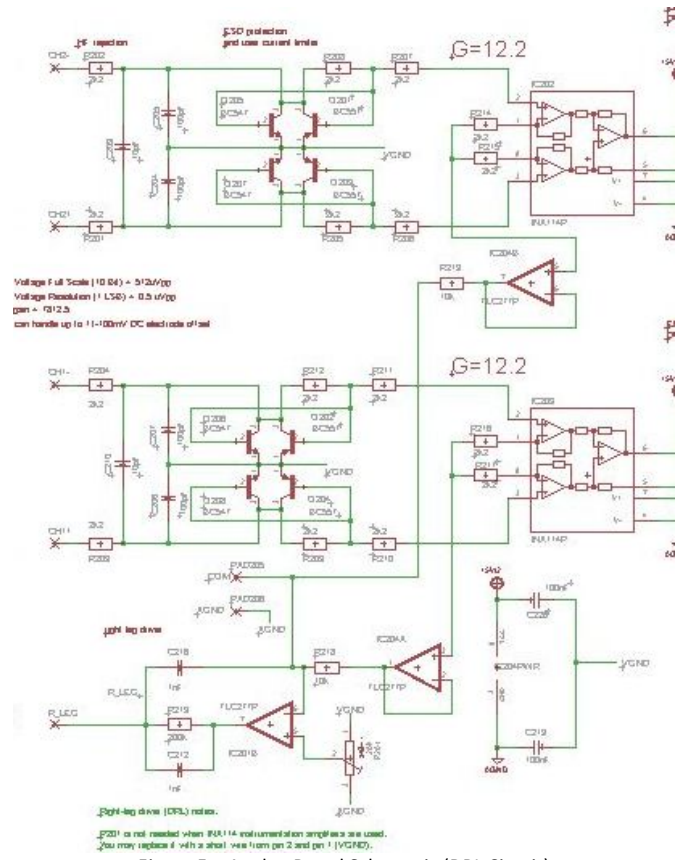


Figure 5 – Analog Board Schematic (DRL Circuit)

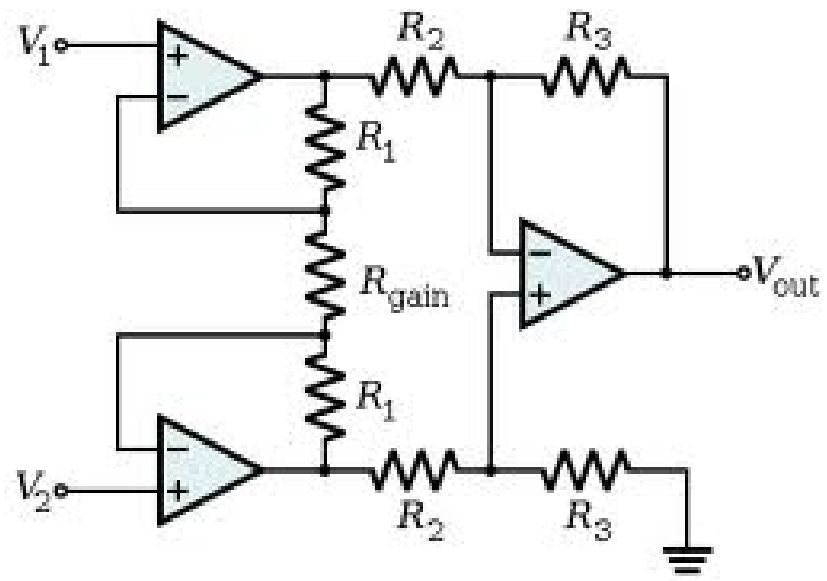


Figure 6 – Common Instrumentation Amplifier Circuit<sup>2</sup>

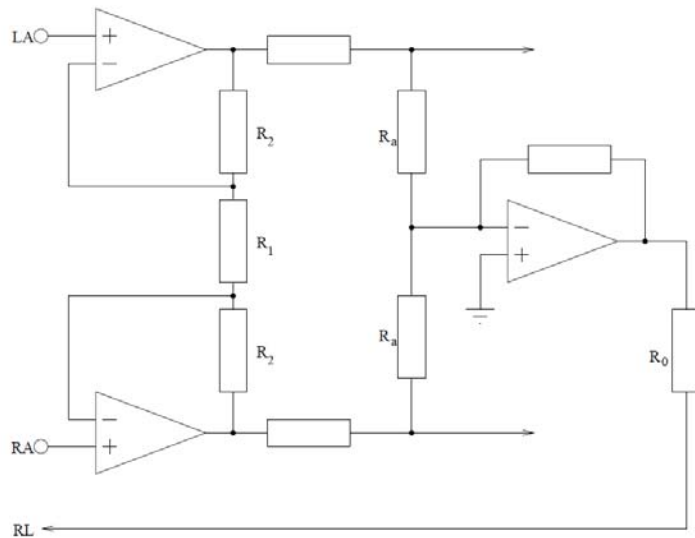


Figure 7 – Instrumentation Amplifier as DRL<sup>1</sup>

The DRL circuit is an important component of the common mode rejection amplifier. The DRL (driven right leg) is often used in systems that measure biological systems as it neutralizes external noise that collects on an individual's skin. In standard instrumentation amplifier design, the gain and relative differential ground comes from the first resistor in the circuit ( $R_{gain}$  in figure 6). By breaking  $R_{gain}$  into two resistors and center tapping the resistors with the DRL, the DRL is essentially the grounded voltage for both halves of the CMR amplifier. By buffering this signal and passing it back into the body, the body has a relative ground and simultaneously charged like that of a faraday cage for noise elimination. The DRL also improves user safety by limiting the current returned to the body. The resistance between the DRL and the input electrodes is several mega ohms. In figure 7,  $R_o$  refers to the skin. In this diagram it can be seen how an increase in the amplifier voltage would cause an increased voltage at the  $V_n$  terminal with respect to system ground. Therefore the output voltage cannot spike and exceed the supply rails, therefore the current would safely dissipate over the high resistance of the skin.

### Analog Board – Channel 1 (&2) Circuit

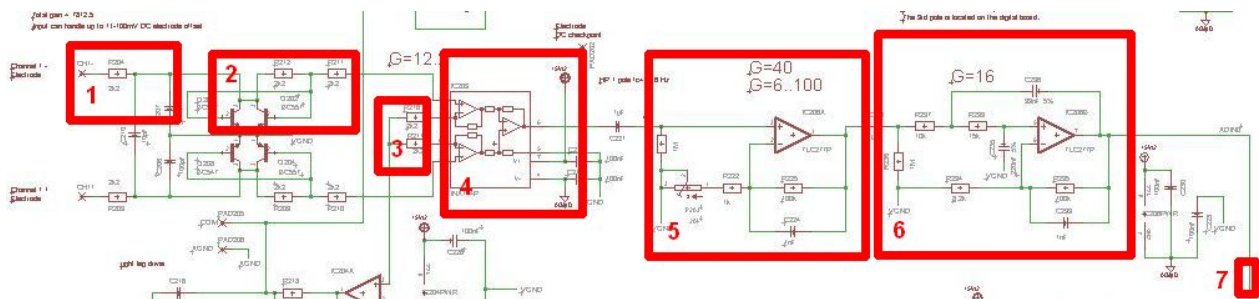


Figure 8 – Analog Board Schematic (Channel 1 Circuit)

The primary function of the Analog board is to receive small noisy signals from the skin, eliminate noise, amplify the signal, and pass it to the Digital board. Above (figure 8) is a picture of the primary amplifier path with the key components highlighted and labeled as numbers 1-7.

Part 1 – Part 1 is a lowpass filter to eliminate general high frequency, the cutoff is 2.3 kHz, much higher than the BW under inspection (0-60 Hz)

Part 2 – ESD protection and user current limiter, this circuitry is used to protect the user and hardware from large voltage spikes between the system and ground. Large positive or negative voltages will activate either the NPN or PNP transistor and short the signal to ground. Small signals ignore the transistors and go into stage 4.

Part 3 – DRL connection

Part 4 – Instrumentation Amplifier (CMR amplifier), the high and low reference signals are fed from the electrodes, into the amplifier. The amplifier isolates common signals such as DC offsets and noise that might appear on the skin such as 120 Hz noise from fluorescent lights. If there is a mismatch in the pre-amplifier gains, common modes will be uneven and not eliminated by this amplifier.

Part 5 – High Gain Amplifier, this stage amplifies the signal based on the resistor ratios of R221 and R202 which is a pot. The gain equation is  $G = 1 + (R221/(R220 + R201))$ . R221 is the 100k negative feedback resistor, R220 is the 1k resistor between the pot and the negative terminal. Therefore the gain is maximized when the pot is set to zero leaving  $G = 1 + (100k/1k) \sim 100$ .

Part 6 – Besselworth Filter, this stage is a 3<sup>rd</sup> order lowpass filter that, with its flat passband and steep cutoff, resembles both Butterworth filters and Bessel filters<sup>3</sup>. This filter was designed with a cutoff frequency of 59 Hz that further eliminates noise and any chance of aliasing before reaching the ADC.

Part 7 – Path terminates at Output header pin 4 (pin 10 for channel 2)

## Digital Board

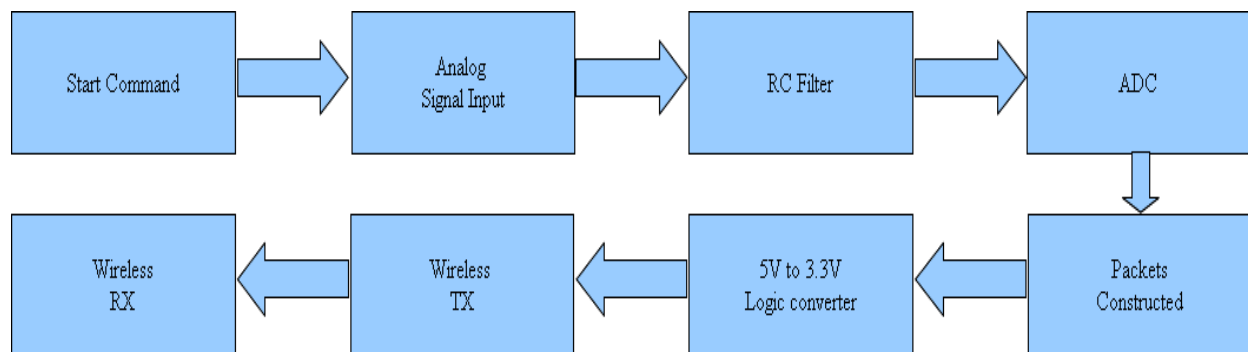


Figure 13 – Digital Board Flowchart

**Start Command**– Matlab sent bits to the Digital Board to signal that it is ready.

**Multi-Channel Analysis** – analyze signals on any number of channels between 1 and 8.

**Single Channel Analysis** – analyze one signal in frequency domain, polychorrence, time domain and split into individual frequency bands.

**RC Filter**- Filters out signal that have a frequency of 60 Hz or more

**Packets**- Packets consist of both channels and a number of bits that help to keep synchronization of the sent signal.

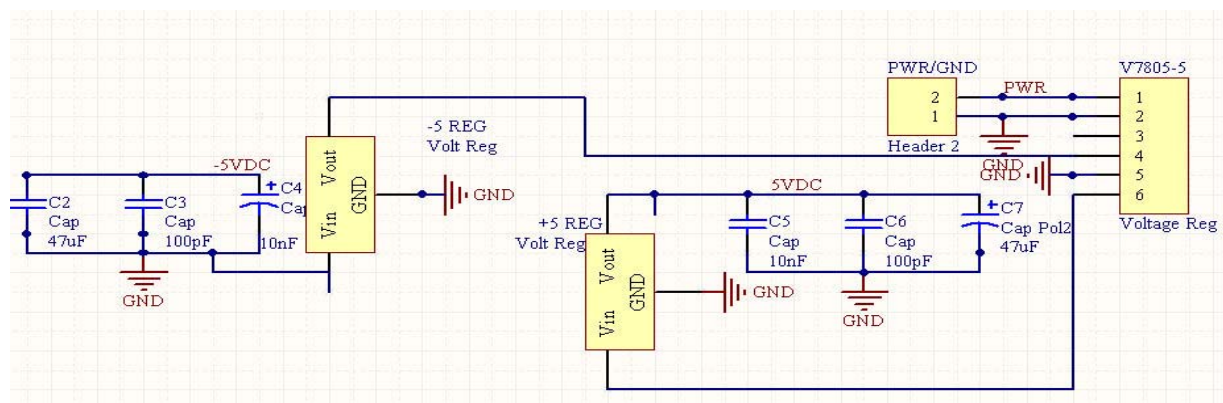
**Logic Converter**- Converts signals that leave the microprocessor from 5V to 3.3V and signals that leave the XBEE from 3.3V to 5V.

**Transmission**- Signal is then wirelessly transmitted to Matlab where it it decoded and analyzed.

Figure 14 – Digital Board Schematic

This board serves as the core of our project as it is the middle man between the brain, and the computer. It provides many important functions, and had its own problems associated with it. This board has high sensitive power rails that supply the power to the ADCs and the analog board. There are also power rails that supply voltage to the active electrodes, in order to properly bias the op amps. This board contains the XBEE module for wireless transmission. Since test signals in the range that the analog board takes as a inputs are not able to be generated using normal equipment, a pulse width modulated signal is generated at a frequency of alpha waves (8-14 Hz). This signal is then attenuated using a voltage divider which can be run through either channel at anytime.

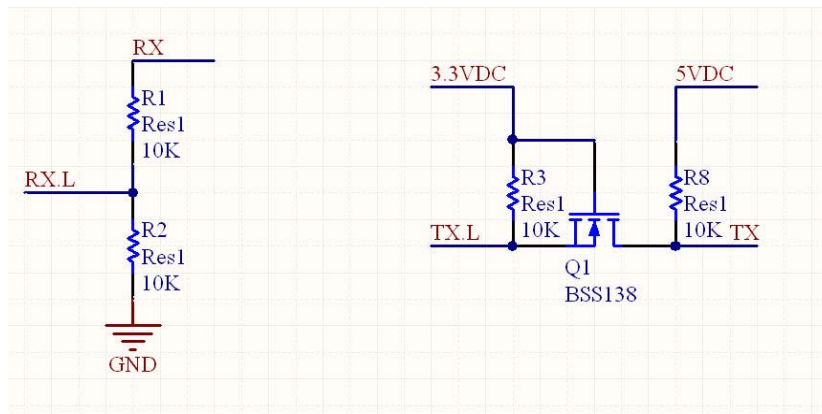
This board contains six different voltage levels all of which perform a specific function. Since the electrode biasing has to contain both positive and negative 5V rails these voltages were established first. This presented one of the main problems in the voltage regulation circuitry. Since one battery was being used, center tapping to create both positive and negative rails was not an option. To solve this problem the battery was fed into a V7805 DC to DC converter. This takes the 7.2V lithium ion batteries and makes a positive and negative 7.2V rail with a common ground. Since  $\pm 7.2V$  is not required this circuitry is then voltage regulated using a LM2940-5 and a LT4905, bringing the voltage to plus and minus 5V as shown in the figure.



Both of these rails have 3 capacitors that help eliminate noise at different frequencies as noise is the biggest problem on low strength signals. It is common knowledge that three capacitors in

parallel cause one equivalent capacitance. However different size capacitors respond at different rates thus each eliminates a difference type of noise (light, thermal/sound, antenna effect). The 5V rail is then voltage divided into a 2V to supply the dc offset for the test signal. The 5V rails are also used to power the AREF pin. Since fluctuation on this pin can cause major changes to the output, this pin is set using a potentiometer. This potentiometer is used to form an RC circuit that begins attenuating all frequencies above 40 hertz. This helps to eliminate any noise generated by the processor and/or other sources on the AREF pin specifically. Finally the 5V is regulated to 3.3V to supply power to the XBEE. The minus 5V rail is specifically used to provide power to the bottom rails of the op amps on the active electrodes.

Wireless transmission was handled using two XBEE pro series rf modules. The first of which was connected to the digital board and the second was connected to a computer using a breakout board containing an RS232 chip. Several uncommon problems occurred when using this hardware setup. The first of which was the difference in voltage levels between the Atmega324p and the XBEE. Since the processor is running at 5V, high signals could potentially damage the XBEE. To solve this problem a voltage divider was used. This voltage divider takes signals coming from the processor and steps them down to a level that the XBEE can accept



without harm. Communications the other way need the opposite effect which is solved using a BSS138 Nchannel mode field effect transistor shown below.

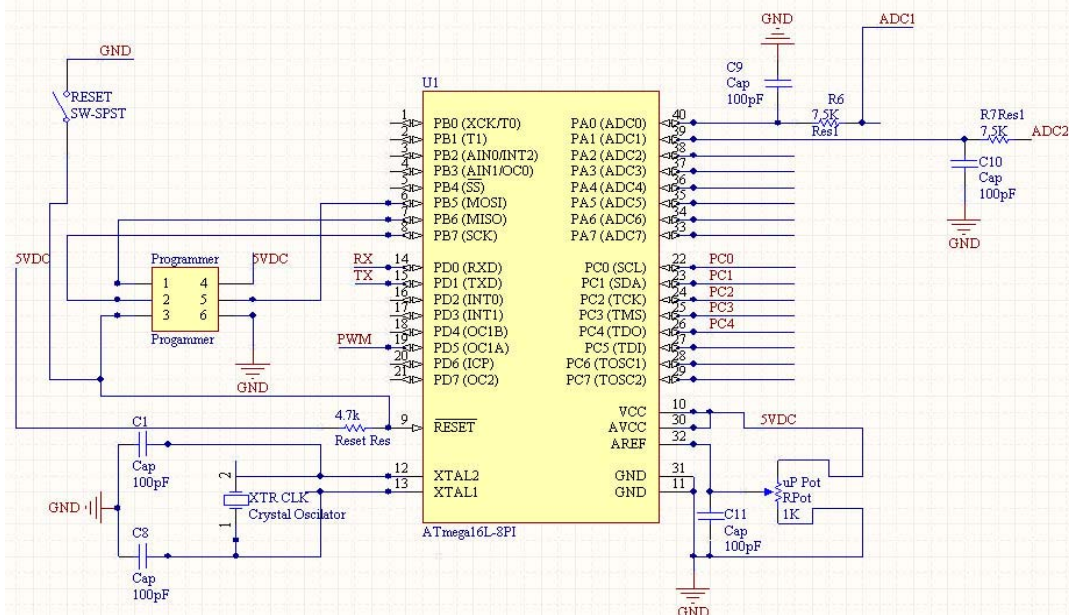
After the logic level conversion had taken place the other main problem was syncing. The digital board waits for Matlab to send a signal telling the processor that it is ready for data transmission. A previously determined amount of data is then sent which in this case is a 20 character header followed by 420 packets of 6 samples. The processor then waits for Matlab to finish its computations. When another signal is received another transmission occurs. The problem here comes down to the clock speed of the computer running Matlab, and the amount of time it takes to execute the instructions to send then receive data. In both cases (Matlab and the Atmega324p) the two lines of code occur one after another. Since Matlab runs at different speeds on every computer the amount of time it takes to execute these instructions changes. This causes Matlab to take too long to switch cycles meaning packets can be sent before Matlab has time to receive them or Matlab attempts to read packets before they are sent. Either one causes a reshaping error thus making the program crash. To solve this problem Matlab functions were used to calculate flops of instruction sets, and the overall time it takes to execute each instruction. This was then compared to an equivalent calculation on the processor's side. This also includes any and all processes the computer is running other than Matlab as this causes changes in the execution of these steps. The difference was then used to generated a delay on the processor's code resolving

the sync issues. Since each computer is unique, a number was calculated so that it contained enough robustness that system operates on most any setup. To help with this debugging process a RS232 header was also included so that wireless can at anytime be bypassed and a hard wire connection can be used. This also allows for two computers to see the same data at the same time as both wireless and hardwired transmission can occur simultaneously.

This particular board is setup to run two adc's both of which have low pass RC circuits to help eliminate noise at high frequencies ( $> 60$  Hz). As previously stated noise reduction is the most important thing when it comes to transmission of small signals. These RC filters are designed to eliminate noise that is introduced on the electrode cables. Since anything can act as an antenna for any and all airborne signals it is important to eliminate noise caused by these cables. This particular board is limited by the analog board which allows for two channels to operate at the same time. Using another analog board (one with more channels) it is possible to digitize and packet up to 6 channels at a time.

The packets prepared by the digital board contain relevant information in the transmission other than the data itself. The first of which is the packet number that is being transmitted this value counts from 1-255 then resets. This helps to insure that all packets are successfully received. The packets also contain a header which tells then number of active channels being transmitted. Since 6 channels are sent regardless, channels that are not currently in use are sent as a 0.

Since this board is designed to be used by other users several other pieces of hardware have been included on the board. The first of which is a power light. This light glows when the processor is receiving enough power to run the board providing the user with an indication that the board is on. Next a slot for an off board clock is provided. Since the baud rate and number of channels can be specified users may choose to change the clock originally provided. The programming header is also pre-wired into the board to allow access to the microprocessor's fuse bits, code, and other ports. Since sync issues can be a problem in many new setups (whether it be Matlab, channel number or baud rate) a reset switch was provided to reset the microprocessor while still providing power to the rest of the board. This insures that the XBEEs stay connected and the voltage regulation circuitry is still functioning during a reset. Also several other ports have also been mapped to the main header to allow access for other applications and/or debugging. A picture of the microprocessor and some of the user friendly features is shown on the beginning of the next page.



Although at first glance this digital board appears to be a typical board with many of the usual parts, much time was taken to calculate the specific resistor and capacitor values to insure that any and all noise introduced to the system would be taken back out. The system is also designed to be robust enough so simply changing computers doesn't cause the system to stop functioning. The regulators used in this system also provide stability as the lithium ion batteries that supply the power drain from their peak supply of 8.4V down to their depletion at 6V.

## Signal Analysis Software Breakdown

The signal analysis software for the neuromancer was created using MATLAB. It offers a user friendly GUI that is fully functional on multiple platforms and has an easy configuration menu for declared serial ports for both the neuromancer wireless dongle AND any BCI hardware the user might want to connect. The software also has several settings for signal analysis that break the signal into time domain, frequency domain, and EEG band signals. The system also has a data recording feature that allows for recording of both channels as well as user/trial specific

information so that the user can easily select old recordings for further signal processing. The program can be split into five dominant parts: Data Read/Analysis Cycle, primary GUI, Dual Signal Analysis, Channel 1 (or 2) Analysis, BCI GUI.

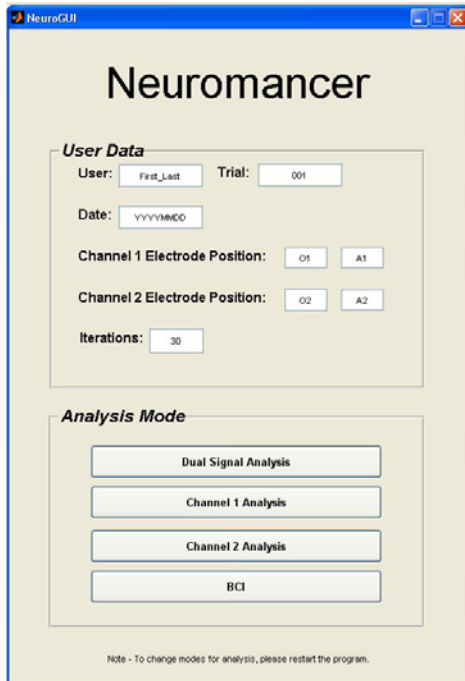


Figure 1 – Primary GUI

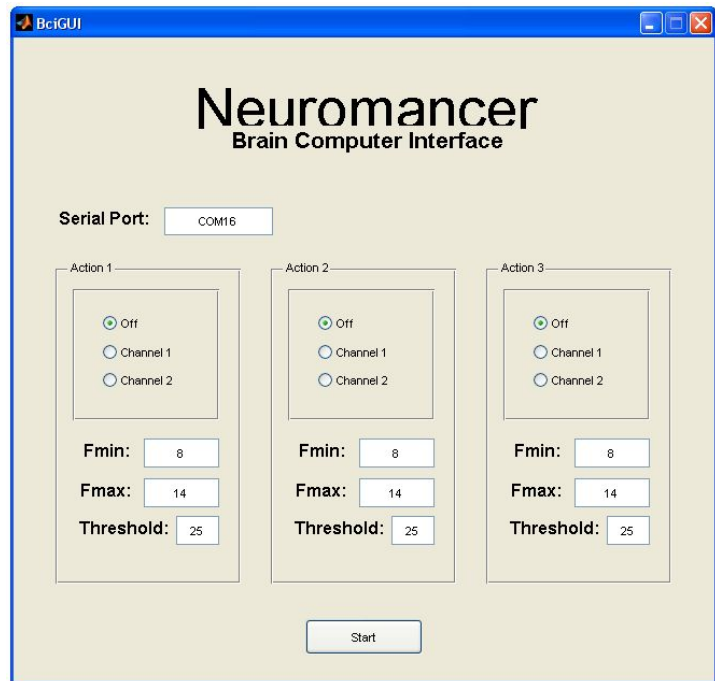


Figure 2 – BCI Gui

## Data Read/Analysis Cycle

The program functions continuously in real time and therefore has a primary loop that it uses to read data in. The data is sampled at 660 Hz (ten times the cutoff frequency) allow for higher signal resolution. The microprocessor transmits data to the PC when it receives the transmit command. The PC sends the transmit command (Hex 48). The microprocessor then sends 70 packets. Each packet contains the packet value 4 times, then 1 sample from channel 1, then 1 sample from channel 2. The program then isolates each channel sample before passing it on to be analyzed and plotted. The program runs faster by reading and processing larger numbers of samples, therefore the program reads in 245 samples at a time before analyzing them. The read cycle takes approx 1/3 of a second. Depending on the analysis mode the data is then processed and plotted accordingly. The signal processing will be greater defined in the Dual Signal Analysis and the Channel 1 (or 2) Analysis sections.

## Primary GUI

The primary GUI starts the initializations of the program, stores inputs about the user, and offers different modes of analysis. Each trial is limited to the number of read cycles the program takes



in from the microprocessor. Once the has looped through the specified iterations quantity, the analysis mode is complete, the data is saved with TrialWriter, and the user is redirected back to the primary gui.

## Dual Signal Analysis

The dual signal analysis section allows the user to simultaneously view both channels in the frequency and time domain. The program reads in 490 samples of each channel, computes the FFT and plots both signals in the time and frequency domain. The number of samples and sampling frequency control the signal resolution. Because of the values we selected, matlab outputs an FFT that spans twice the nyquist frequency, therefore only a few points from the FFT are selected and plotted. The code is given below:

FFT Code:

```
1     disp('Reading Data...');
2     fwrite(s, x);
3     Header = fread(s,20);
4     Signal = [fread(s,420); fread(s,420); fread(s,420); fread(s,420);
fread(s,420); fread(s,420); fread(s,420)];
5     %disp('Isolating Channels...');
6     X = reshape(Signal,6,490);
7     chan1 = X(5,:);
8     chan2 = X(6,:);

9     chan1FFT=abs(fft(chan1))/N;
10    chan1FFT = chan1FFT(1:cutOff);
11    chan1FFT(1)=0;
12    loc1=find(chan1FFT == max(chan1FFT));
13    chan1PF = freq(loc1);
14    totel = sum(chan1FFT);
```

Line 3 shows the synchronization header being read in.

Line 4 shows MATLAB reading the specified number of samples.

Line 6 shows MATLAB reshaping the signal for channel isolation.

Line 7 shows channel 1 being isolated into its own vector.

Line 9 – channel 1 vector is converted into the frequency domain, absolved, and divided by N (signal length) to normalize the output

Line 10 – key frequency features are isolated.

Line 11 – DC components are neutralized.

Line 12/13 – identify the frequency corresponding to the maximum point in the signal FFT.

Line 14 – Find the channel signal strength by summing the signals.

## Channel 1 (or 2) Analysis

The single channel analysis options allow the user to break down the signal into its primary EEG bands and view them in real time. Due to typical PC limitations it was not realistic to plot two channels under heavy analysis in real time. To break the signal into its respective bands, the signal had to be read in, the channel had to be isolated from the transmitted signal, and then passed through butterworth filters for each individual band. Matlab is very efficient at computing math on large data sets and therefore a 100<sup>th</sup> order butterworth filter was selected for band extraction. The code for isolating the alpha band is given below:

```

1      %%%% Alpha BandPass Filter Initializations %%%%
2      Fs = 606; % Sampling Frequency
3      N      = 100; % Order
4      Fstop1 = 6; % First Stopband Frequency
5      Fpass1 = 8; % First Passband Frequency
6      Fpass2 = 12; % Second Passband Frequency
7      Fstop2 = 14; % Second Stopband Frequency
8      Wstop1 = 1; % First Stopband Weight
9      Wpass  = 1; % Passband Weight
10     Wstop2 = 1; % Second Stopband Weight
11     dens   = 20; % Density Factor
12     b = firpm(N, [0 Fstop1 Fpass1 Fpass2 Fstop2 Fs/2]/(Fs/2), [0 0 1
1 0 0], [Wstop1 Wpass Wstop2], {dens});
13     HdAlpha = dfilt.dffir(b);

...
14     y3= filter(HdAlpha,chan1); y3 = y3 + 1.5;

...
15     subplot(6,1,5); plot(y3,'r'); xlim([0 490]); ylim([0 3]); hleg1 =
legend('Alpha Band Wave'); set(gca,'XTickLabel',[]);
set(gca,'YTickLabel',[]);

```

## BCI GUI

The BCI GUI gives the user the ability to test their mental control by interacting with hardware they have configured for BCI interaction. The GUI requires the COM port for the external hardware to communicate with as well as action specifications. The gui can output up to three actions based on the users specifications. The outputs are based on identifying dominant frequencies in specific channels. The program reads in data, computes the FFT, computes the spectral power, and then compares the signal to the maximum and minimum frequency thresholds as well as the magnitude before deciding with the action is true or false. The program then outputs a 0, 1, 2, or 3, depending on what action is true.

# Battery Charging Circuit

## Batteries

One of the main goals of the Neuromancer project was to run all of the components wirelessly and off of batteries. In order to save cost we chose to use rechargeable batteries. The Neuromancer uses a 2 cell 7.4V/730mAh Thunder Power LiPo (Lithium-Polymer) RC battery (shown in the picture below) to run the digital board and helmet circuitry.



Figure 16: Thunder Power Rechargeable Battery

## Charging LiPo Batteries

Lithium polymer batteries are quite volatile and extreme care should be taken when charging them. Below is a list of a few of the key precautions that should be taken when charging these batteries.

- *Make sure that there is at least a 3V per cell charge on the battery pack before placing on the charger. Charging below this level could result in damage and possibly an explosion of the battery.*
- *If each cell is discharged to 2.6 – 2.9V per cell a trickle charge must be used to charge each cell up to 3V. Failure to do so will result in damage and possible explosion of the battery pack.*
- *Never charge a pack with more current than the rated mAh (i.e. Never charge a 450mAh battery with more than 450mA of charging current). Failure to do so may result in explosion of the battery pack.*
- *In a typical cycle, each cell is charged to around 4.2V (8.4V total). Any further charging will result in explosion of the battery pack.*

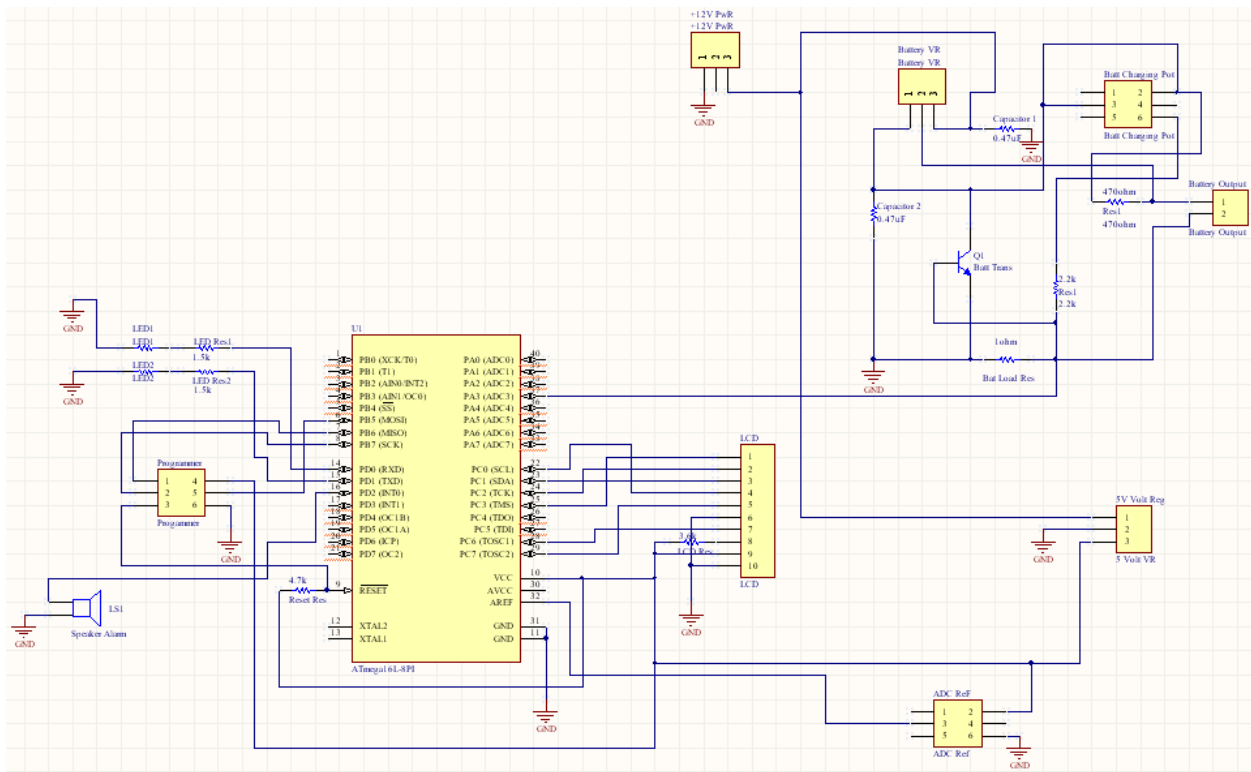


Figure 17 – Charging Circuit Schematic

Taking all of the considerations mentioned above a design for the charging circuit was selected. The above circuit was designed (with help from SHdesigns) to safely charge LiPo or Lithium Ion batteries. It includes the following features:

- *Over-Charge Protection*
- *Trickle Charge*
- *Adjustable Output charging voltage*
- *Voltage monitoring via microprocessor and LCD screen*

### Charging Data

Once constructed, several test runs were conducted to verify the hardware was working properly. Shown below is a model of the current and voltage over an hour charging period after the batteries had been charged to about 50% charge. A graph of a typical charging is also included for comparison.

Figure 18 - Model of the Current during the charging process

Figure 19 - Model of the Voltage during the charging process

# Charge Characteristics

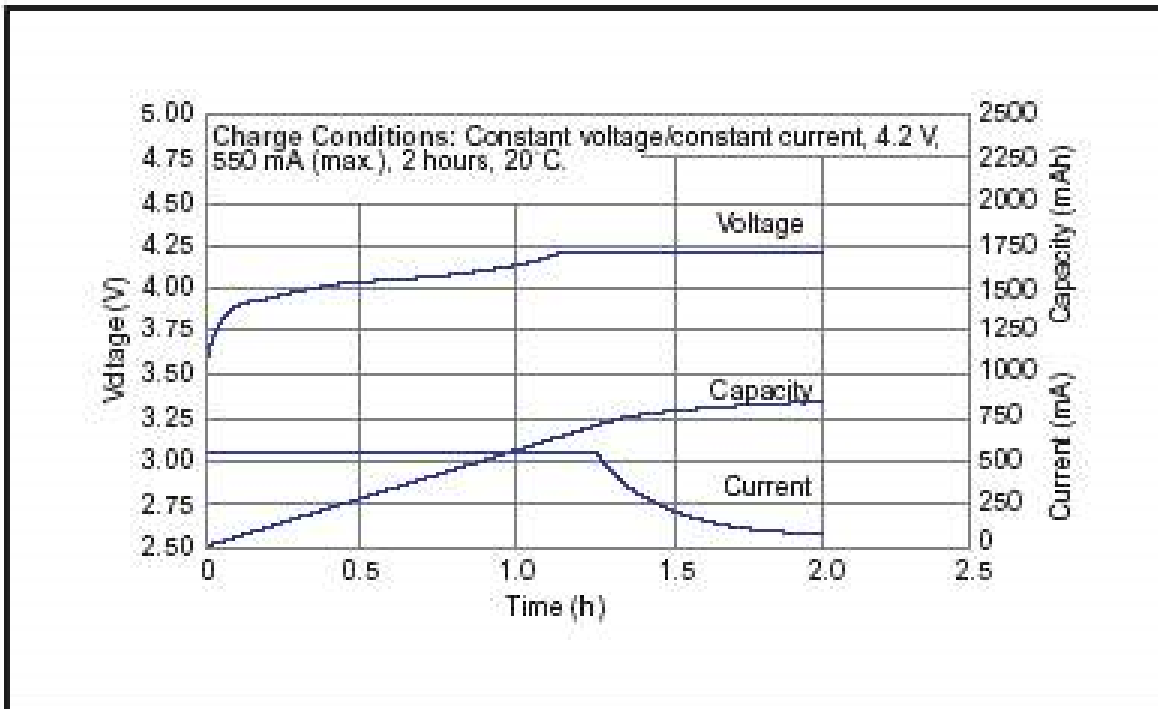


Figure 20 - Typical charging characteristics

## Appendix A - Digital Code

```
#include <stdint.h>
#include <util/delay.h>
#include <avr/io.h>
#include <avr32_adc.h>
#include <util/delay.h>

#define F_CPU 20000000 //Define CPU Speed
#define USART_BAUD 38400 //Define Baud Rate
#define USART_UBBR_VALUE ((F_CPU/(USART_BAUD*16))-1)
double i,i2;
uint8_t p=0;
uint8_t p2=0;

void SetupPorts(void) {

    DDRD = 0x30; // OCR0A+B (4,3)
}

void SetupTimers(void) {
```



```

}
USART_vSendByte(u8Data);          //Echo other received Characters
}

for(;;)
{

here: USART_vSendByte(u8Data);      //Packet Header
USART_vSendByte(u8Data2);
USART_vSendByte(u8Data);
USART_vSendByte(u8Data);
//USART_vSendByte(u8Data2);
ADC=ADC_START(0);
i=(ADC/1023.0)*255.0;
p=i+0x00;                          //Read Convert and Send Cancel 1
USART_vSendByte(p);
//USART_vSendByte(u8Data2);
ADC=ADC_START(1);
i2=(ADC/1023.0)*255.0;
p2=i2+0x00;                         //Read Convert and Send Cancel 2
USART_vSendByte(p2);
//USART_vSendByte(u8Data2);
}
}

```

## Appendix B - Battery Charger Code

```

//Chris Carter
//Rechargeable Battery Circuit

//*****
//*****      Code Initialization      *****
//*****

#include <avr/io.h>
#include <UF_LCD.h>
#include <UF_DAC.h>
#include <avr32_adc.h>

void main(void){

double volt;
unsigned int num1,num2,num3,x;      //Variables for program

DDRD = 0x03;                       //Setting Port D as output ports

lcd_init();
ADC_INIT();

//*****
//*****      Battery Charging Circuit      *****
//*****

while (1){

    ADC_START(3);

    volt = (ADC/110.5)+.05;          //Analog to Digital conversion
    volt = volt/10;

//*****

```

```

//***** No Batteries Present *****
while (volt <= 0.019)
{
  lcd_char('I');           //Writing message to the LCD.
  lcd_char('n');
  lcd_char('s');
  lcd_char('e');
  lcd_char('r');
  lcd_char('t');
  lcd_char(' ');
  lcd_char('B');
  lcd_char('a');
  lcd_char('t');
  lcd_char('t');
  lcd_char('e');
  lcd_char('r');
  lcd_char('i');
  lcd_char('e');
  lcd_char('s');

  lcd_command(0xC0);       //Moving to second line of LCD

  for (x = 21; x > 0; x--) //Clearing blocks on voltage bar
    lcd_char(' ');

  ADC_START(3);

  volt = (ADC/110.5)+.05; //Analog to Digital conversion
  volt = volt/10;

  lcd_command(0x80);      //set cursor back to top line
}

//***** Batteries Charging *****
while (volt < .65 && volt >= 0.05)
{
  lcd_char('C');           //Writing message to the LCD.
  lcd_char('h');
  lcd_char('a');
  lcd_char('r');
  lcd_char('g');
  lcd_char('i');
  lcd_char('n');
  lcd_char('g');

  for(x = 11; x>0; x--)
    lcd_char(' ');

  lcd_command(0xC0);       //Moving to second line of LCD

  for (x = 21 - volt * 30.7692; x > 0; x--) //Creating blocks on voltage bar
    lcd_char(0xFF);

  for (x = volt * 30.7692; x > 0; x--) //Clearing blocks for voltage bar
    lcd_char(' ');

  PORTD = 0x01;           //Turning on LED
  _delay_ms(55);
  PORTD = 0x00;
  _delay_ms(55);
  PORTD = 0x02;
}

```



```

    _delay_ms(55);
    PORTD = 0x00;
    _delay_ms(55);
    PORTD = 0x01;
    _delay_ms(55);
    PORTD = 0x00;
    _delay_ms(55);
    PORTD = 0x02;
    _delay_ms(55);
    PORTD = 0x00;
    _delay_ms(55);
    PORTD = 0x01;
    _delay_ms(55);
    PORTD = 0x00;
    _delay_ms(55);
    PORTD = 0x02;
    _delay_ms(55);
    PORTD = 0x00;

    lcd_command(0x80);           //set cursor back to top line

    ADC_START(3);

    volt = (ADC/110.5)+.05;      //Analog to Digital conversion
    volt = volt/10;

}

//*****
//***** Charge Complete *****
while (volt < 0.49 && volt > 0.020)
{
    lcd_char('C');           //Writing message to the LCD.
    lcd_char('h');
    lcd_char('a');
    lcd_char('r');
    lcd_char('g');
    lcd_char('e');
    lcd_char(' ');
    lcd_char('C');
    lcd_char('o');
    lcd_char('m');
    lcd_char('p');
    lcd_char('l');
    lcd_char('e');
    lcd_char('t');
    lcd_char('e');
    lcd_char('!');

    lcd_command(0xC0);       //Moving to second line of LCD

    for (x = 21; x > 0; x--) //Clearing blocks on voltage bar
        lcd_char(' ');

    PORTD = 0x01;           //Turning on LEDs
    _delay_ms(55);
    PORTD = 0x00;
    _delay_ms(55);
    PORTD = 0x02;
    _delay_ms(55);
    PORTD = 0x00;
    _delay_ms(55);
    PORTD = 0x01;
    _delay_ms(55);

```

```

PORTD = 0x00;
_delay_ms(55);
PORTD = 0x02;
_delay_ms(55);
PORTD = 0x00;
_delay_ms(55);
PORTD = 0x01;
_delay_ms(55);
PORTD = 0x00;
_delay_ms(55);
PORTD = 0x02;
_delay_ms(55);
PORTD = 0x00;

PORTD = 0x04;           //Turning on Speaker
_delay_ms(55);
PORTD = 0x04;
_delay_ms(55);
PORTD = 0x04;
_delay_ms(55);
PORTD = 0x04;

ADC_START(3);

volt = (ADC/110.5)+.05;   //Analog to Digital conversion
volt = volt/10;

lcd_command(0x80);      //set cursor back to top line
}
}
}

```

## Appendix C – References

1 - *Signal Processing & Neural Networks Group, Dept. of Engineering Science, University of Oxford, 17 Parks Road, Oxford, OX1 3PJ, UK.*

[http://www.robots.ox.ac.uk/~neil/teaching/lectures/med\\_elec/notes2.pdf](http://www.robots.ox.ac.uk/~neil/teaching/lectures/med_elec/notes2.pdf)

2 - Instrumentation Amplifier

[http://t1.gstatic.com/images?q=tbn:ANd9GcQ4b5-sxe\\_PWdBC75bjpDOAg9QreYS4dvN9G2QNMycb-fikrmbkKw&t=1](http://t1.gstatic.com/images?q=tbn:ANd9GcQ4b5-sxe_PWdBC75bjpDOAg9QreYS4dvN9G2QNMycb-fikrmbkKw&t=1)

3 - Besselworth Filter

<http://www.mina.ubc.ca/files/474/2005/Team%205%20Final%20Report.pdf>

4 – ModularEEG Design Files

<http://openeeg.sourceforge.net/doc/modeeg/modEEGamp-v1.0.png>