Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

Final Prototype Testing Plan

EEG-based Brain-Computer Interface

Submitted to:

Prof. Alan Pisano
apisano@bu.edu

by

Team 04
Brain Force

Team Members:

Alexander Johnson atjohnso@bu.edu
Brendan Shortall shortall@bu.edu
Dayanna De La Torres dayannat@bu.edu
Jonathan Mikalov jmikalov@bu.edu
Mitchell Gilmore mgilm0re@bu.edu

Submitted: 4/06/2023

Required Materials

Hardware:

- EEG Electrode Headset
  - 12 electrode probes with housing
  - Cyton board case and cover
- Cyton PCB
  - USB-A to Micro-USB cable
  - PIC32MX250F128B Microcontroller
  - ADS1299 Bio-sampling chip

Software:

- OpenBCI GUI
- 3D Virtual environment:
  - main.py: run simulation mechanics and control features
- BrainFlow Streaming script
- ML test model
- Socket programs
  - Data transfer between BrainFlow streaming scripts into ML model
  - Data transfer between ML model and virtual environment

Set Up

There are two main goals to this prototype testing. The first goal is to prove the functionality of the hardware. This will be composed of simultaneously testing the hardware and software that is responsible for receiving the data stream from the device. The goal of this software is to allow us to easily begin and end the data stream with the click of a button, and automatically store this data for later use. We will also be able to view, in real time, brain waves from our 8-channels on a monitor. The main difference between this test and our previous is that we hope to get output data that has distinguishable features rather than strictly noise.

The second goal is to successfully have socket connectivity between the different aspects of the project (the headset, neural network, and virtual environment). While the neural network is not completed, it has been simulated through a faux classification script. The goal of these socket programs will be to prove we can catch data from the headset in real time and that we can transfer data from the “neural network” to the virtual environment to emulate a stream of actions.

Set Up Procedure

Set up for testing our hardware will begin by connecting the Cyton board to the OpenBCI GUI to ensure that the hardware and firmware are working properly. We will also be testing the efficiency of the headset-board system by placing the headset on someone's head and viewing the FFT plot of the channels. We hope to get values in the microvolt range that have some discrete artifacts.

Once we ensure the board is working properly, we will then connect it to our BrainFlow code. This code should allow the board to be plug and play, automatically detecting the board and COM port it is connected to. Once a connection has been established, we will then test the functionality of the BrainFlow code by starting a data stream.

We then want to test the entire system. This will consist of plugging in the headset, streaming data into the ML model, and sending the ML model output into the virtual environment. We should be able to see that in real time, the virtual environment will update based on the classification of the ML model.

Testing Procedure

STAGE 1: The testing procedure for the PCB includes:

1. Begin by ensuring that power is being sent to the board indicated by the blue LED
2. Open the OpenBCI GUI and connect the board
3. Observe the FFT plot before the headset is connected
4. Connect the headset, and observe the FFT plot once more.
5. Compare differences, look for artifacts.

STAGE 2: Once we verify that this system is working correctly.

1. Disconnect the Cyton, Close the OpenBCI GUI
2. Open our streaming software, connect the device.
3. Ensure that the automatic board detection system works properly
4. Begin a data stream
5. Wait some time and end the stream
6. Check that the output data is as expected

STAGE 3: Now we begin to test the entire system

1. Boot up the socket code, the ML model, and the virtual environment
2. Plug in the headset and begin a stream
3. View each stage to ensure that the data is coming in/out as expected
4. View the virtual environment to see if it updates as expected

Measurable Criteria

Stage 1:

- Does the board receive power?
- Do we connect to the OpenBCI GUI?
- Can we view the FFT plot in real time, as well as the individual channels?
- What is the peak to peak magnitude of the signals?
- Once the headset is connected, can we notice a difference between the two?

Stage 2:

- Does the script automatically detect the board is connected?
- Does it detect the correct COM port? (Try multiple ports)
- Does the stream begin when instructed?
- Does it end when instructed?
- Is the data received reasonable?

Stage 3:

- Does the BrainFlow code communicate with the ML model?
- Does the ML model communicate with the virtual environment?
- Can we see movement in the virtual environment?

Score Sheet

Stage 1:

Test:	Results:	Comments:
Power received?	(Y/N)	
Connected to GUI?	(Y/N)	
Peak-to-Peak magnitude (Disconnected from headset)		
Peak-to-Peak magnitude (Connected to headset)		
Notice artifacts?	(Y/N)	

Stage 2:

Test:	Results:	Comments:
Automatic board detection		
Test every COM port		
Begin stream		
End stream		
Data reasonable?		

Stage 3:

Test:	Results:	Comments:
BrainFlow to Model		
Model to V.E.		
V.E. updates?		