Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

User's Manual

Brain 4-CE

Submitted to

Prof. Pisano
apisano@bu.edu

by
Team 4
Brain 4-Ce

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: 4/17/2023

Brain 4-Ce User's Manual

Table of Contents

Executive Summary

Life can be increasingly difficult for those who lose motor ability in our society. We seek to provide a solution to those who struggle with motor impairments by constructing an ambulatory electroencephalogram (EEG) powered by thoughts. Our project will give the ability to interact with and move about the world to users.

Packaged in a sleek headset adorned with electrodes, the user's electrical impulses associated with motor imagery tasks will be translated into action. They will be able to maneuver and control computers and, eventually, machinery that will move them in the real world.

This is Brain-4ce. *Where thoughts become motion.*

1 Introduction

The idea of using the mind to control objects and interact with the environment has been fantasized by science fiction for decades. Rooted in the idea that the mind is a powerful tool, this self-defined project aims to tap into the unlimited potential of electroencephalogram (EEG)-based controllers for not just the medical field, but any industry. Applications include, but are not limited to, neural cognitive prosthetics, operating robotics, and virtual realities such as the Metaverse. Brain-4ce was born out of the desire to push limits and explore new and improved methods to create cost-effective, life-altering technology for the everyday user. This manual will explore the pieces that brought this undertaking to life.

Brain-4ce was broken down into 3 separate components: 1) Hardware/Firmware, 2) Neural Network Classification, and 3) Virtual Environment Interaction. Separating the components allowed for easier debugging and updating. The seamless integration of these three components, coupled with the feature of working in real-time, come together to create a user-friendly, enjoyable experience.

The Brain-4ce headset sits comfortably on the head with Cyton and Daisy printed circuit boards resting at the apex of the occipital lobe. These boards, the Cyton specifically, are used for data collection. The user will be prompted with motor-imagery tasks, in which they will imagine taking actions but not physically doing them. If the user desires to interact with the data, they can do so through a graphical user interface; otherwise, they can assume their brain's electrical data is being fed into a convolutional neural network (CNN). This CNN classifies the user's thoughts in real-time into one of four actions: forward, backward, left, or right. Once the classified data exits the neural network, it is given to the virtual environment. The transfer of data across these different components is done through a series of socket programs built into the main functions of each component to create server-client data serialization. Once the processed data reaches its final destination, the virtual environment, the user will see their thoughts turn into action as they navigate an interactive game, collecting stars for points.

The resulting product is a headset capable of facilitating the everyday life of users who need it and of providing a fun time to those that desire it. In the following pages, you will find the appropriate instructions for how to set up and use the system, understand the inner workings of the system, and interact with the user interface.

2 System Overview and Installation

2.1 Overview block diagram

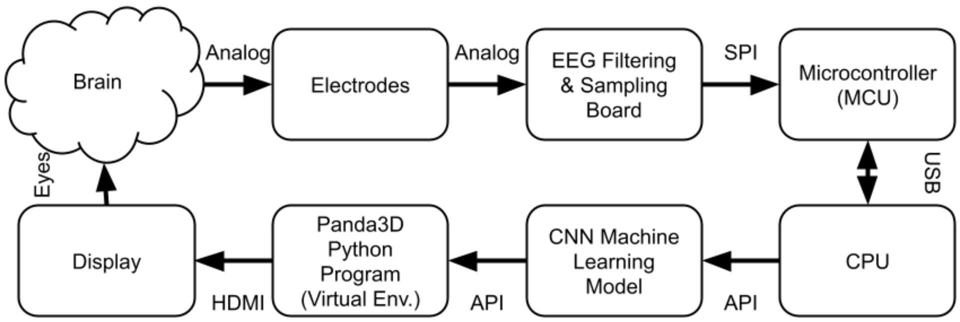

Figure 2.1 - Block diagram overview of the system

The Brain-4ce system begins with a headset containing bias and reference electrodes that are attached to the user's head using electrode conducting gel. The electrodes are connected to the analog-to-digital converter (ADC), which samples the user's brain waves. While the ADC is running it will alert the microcontroller when new samples are available and the microcontroller will request the new sample be sent to pass it on to the computer over a USB connection. The computer may also save or modify recordings as needed using a simple graphical user interface (GUI) interface. Once information has been recorded and interpreted by the computer, a convolutional neural network (CNN) will classify the electroencephalogram (EEG) signal into distinct motor imagery tasks, which in turn directs the orientation and controls of the virtual environment.

2.2 User Interface

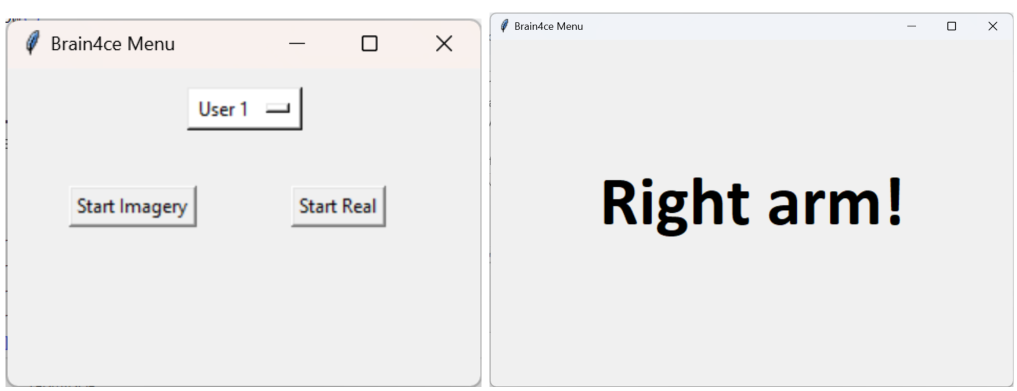

Figure 2.2 - GUI for training data capture

Our EEG-based brain-computer interface (BCI) contains a simple and intuitive GUI that is used for recording a user's thoughts and actions. The menu can store up to 5 distinct users as well as their saved recordings. On the first screen, a user may select either "Start Imagery" or "Start Real" to begin a recording session. The "Start Imagery" option will assume and require that the user does not actually perform a physical/operative action, but only imagine it. This is known as a motor-imagery task and is a common technique used among BCI systems. On the other hand, "Start Real" will require that the user actually perform physical actions involving their arms, feet, and hands.

The second menu, the task window, presents the user with commands regarding the action they must act upon. Each task is displayed for 5 seconds and then disappears momentarily for 2 seconds. During the duration of each task, a Python-based BrainFlow program will begin recording the user's electrical brain activity. Once all tasks have been completed, the user will be asked whether they wish to save their recordings.

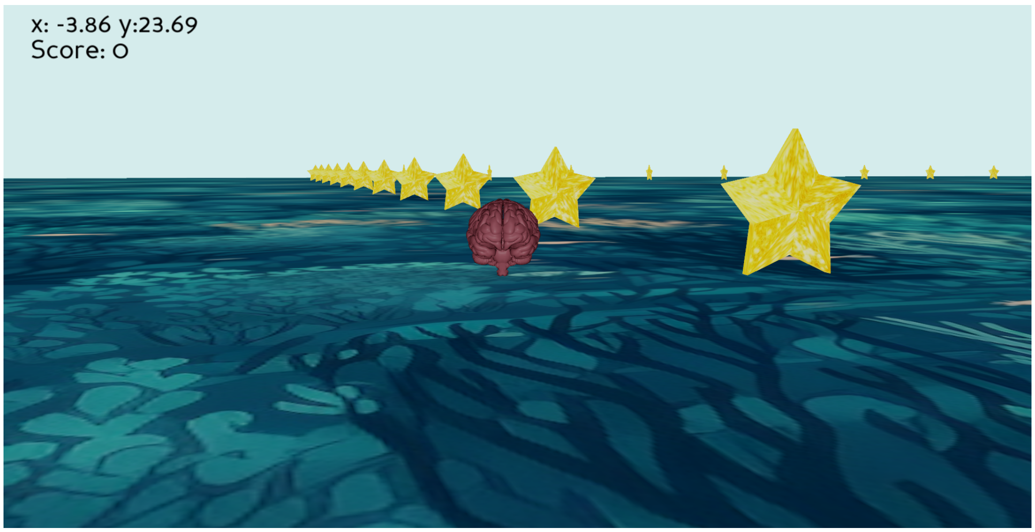

Figure 2.3 - 3D Virtual Environment

Once a user has finished recording both imaginative motor-imagery tasks and physical movements, they will be instructed to play a simple game based inside a 3D virtual environment. This game features simple control mechanics that allow a user to operate in 4 directions: right, left, backward, and forward. In this game, their challenge is to collect as many stars as possible using only their thought patterns. Each thought pattern is associated with a given direction such as right or left. The score is also monitored at the top-left corner of the screen which is used to keep track of the current tally of collected stars.

2.3 Physical Description

The hardware for this project includes the headset, spring-actuated dry electrodes, ear clips, a mounted PCB case, and a PCB. The following sections will describe each in detail, and how they interact with each other.

2.3.1 Headset Body

The headset body is the main structure of the headset that contains 33 total positions to hold the EEG electrodes following the 10-20 EEG system. The body is 3D printed using a Creality Ender S1-Pro 3D printer with PLA filament. Mounting holes are located periodically throughout the design for the PCB case. The headset is slightly wider on one half, indicating where the back of the head should be. This is the side where we mount the PCB case.

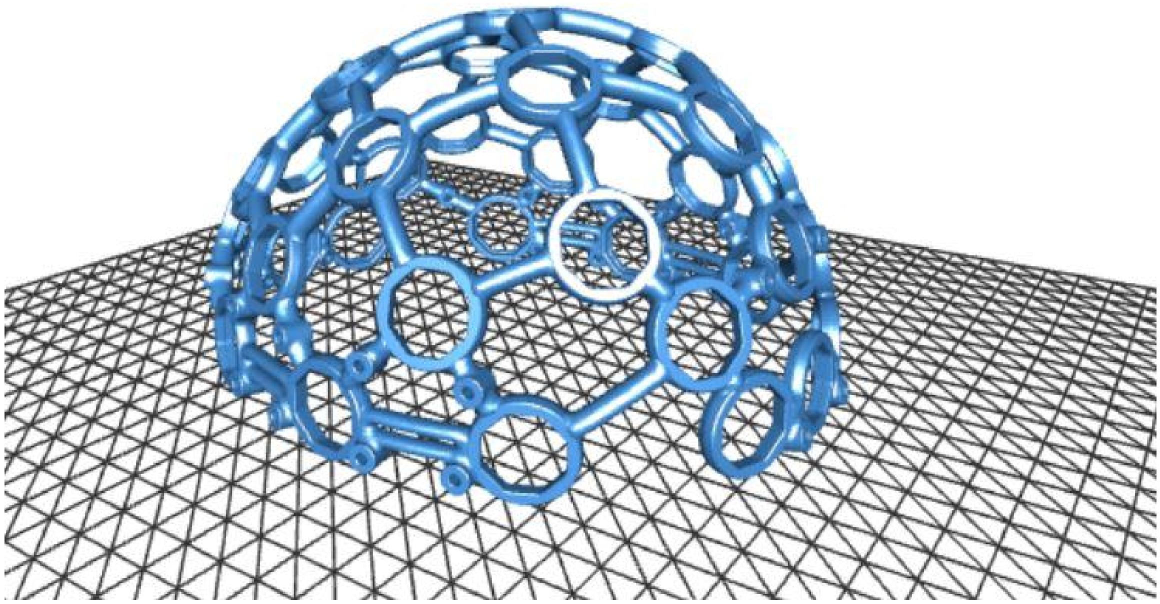

Figure 2.4 - 3D Render of Headset Body design

2.3.2 EEG Electrodes

Each electrode is spring actuated, with the casing being 3D printed using the same printer and filament as stated in section 2.3.1. The outer casing has an octagonal shape that fits into one of the 33 inserts in the body. Springs are placed within this outer casing followed by the inner electrode casing. This inner casing contains the electrode itself, this is constructed using Ag-Cl dry electrodes, ¼" long screws and accompanying hex nuts. The wire exits through the opposite end of the electrode.

Figure 2.5 - Fully assembled electrode

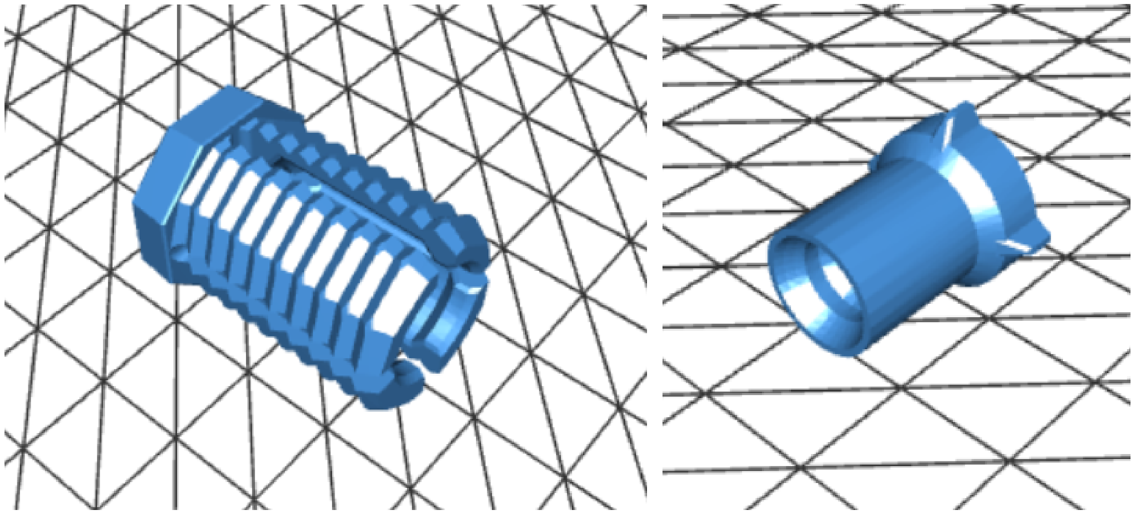

Figure 2.6 - 3D render of electrode housing

16 of these electrodes are constructed and are connected to the pin headers on the cyton and daisy board. Positions for these electrodes were determined using a principal component analysis (PCA) on 64-channel motor imagery EEG data from publicly available datasets. Results from this PCA are outlined in Figure 2.7 and indicate that electrode positions in the back of the head account for the majority of variance in motor imagery data.

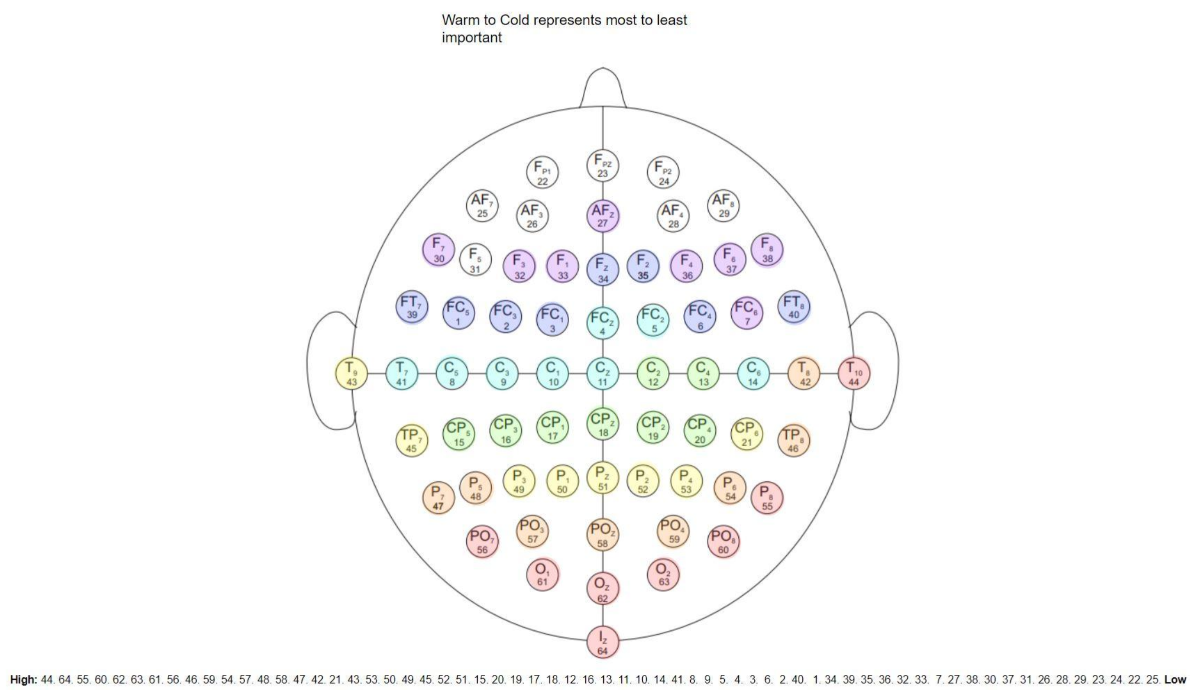

Figure 2.7 - PCA results indicating optimal electrode placement

2.3.3 Ear Clips

2 Ear clips containing gold cup electrodes are also wired to the cyton board. These are necessary as a reference for the 16 active electrodes contained in the headset. These clips are printed in the same fashion as noted previously. Using the gold cup electrodes

purchased from the OpenBCI shop, these clips are comfortably placed on the user’s earlobes. Each electrode is then wired to the Cyton board. This is still under development, but our prototype is pictured below.

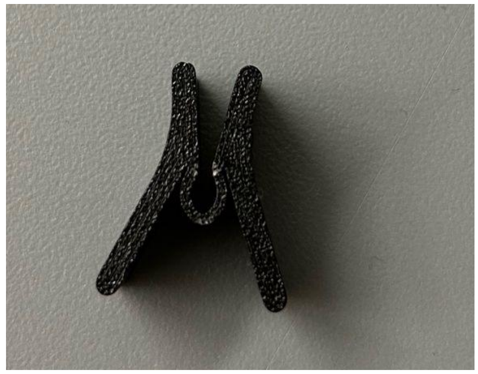

Figure 2.8 - 3D printed ear clip for measuring bias and reference voltages

2.3.4 Cyton and Daisy PCBs

The heart of this project is our programmable circuit boards that collect data from EEG electrodes and relays it to the user’s computer via USB. The ADC bio-sampling chip is responsible for sampling the active and reference electrodes. The board is equipped with low-pass filters that are responsible to prevent aliasing in the frequency domain. A microcontroller contains the device firmware and is responsible for directing communication between components. Finally, a USB to UART microcontroller transforms the data so that it conforms with the USB standard and can be transmitted to the computer. 26 pin outs are located on each of the Cyton and Daisy boards arranged in a 2x13 array. If we reference the pins numerically from 1-26 starting from the top left and ending on the bottom right (aerial view), we connect 8 of the active electrodes to positions 2-9 (inclusive) on both the Cyton and Daisy board. The two reference electrodes are connected to positions 1 and 13 on the Cyton only.

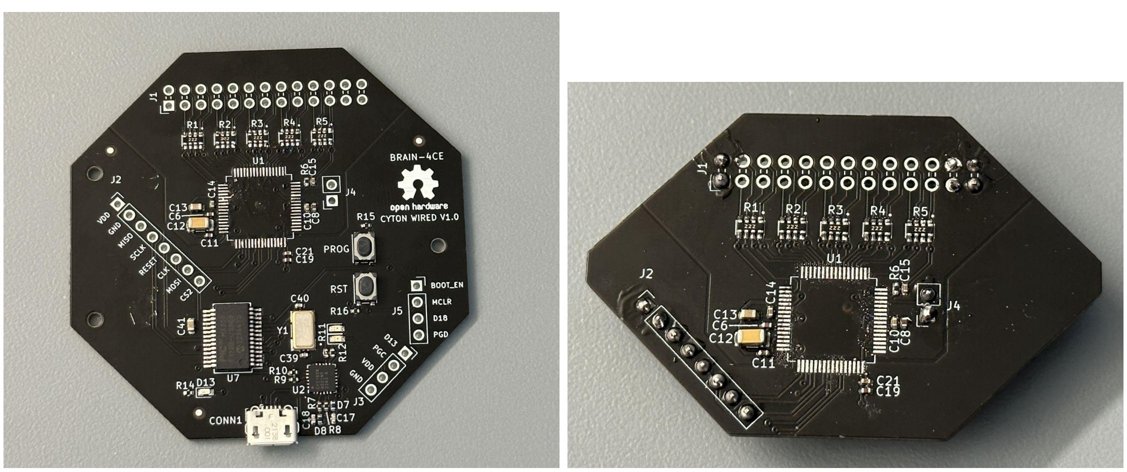

Figure 2.9 - Partially assembled Cyton Wired and Daisy PCBs

2.3.5 PCB Case

The casing for our PCBs is two parts: the frame and the cover. The frame is mounted to the headset body using stainless steel screws. The Cyton board rests atop the frame with the pins on the side of the opening. The cover is then fit over the extruding components on the daisy and snaps into place on the frame.

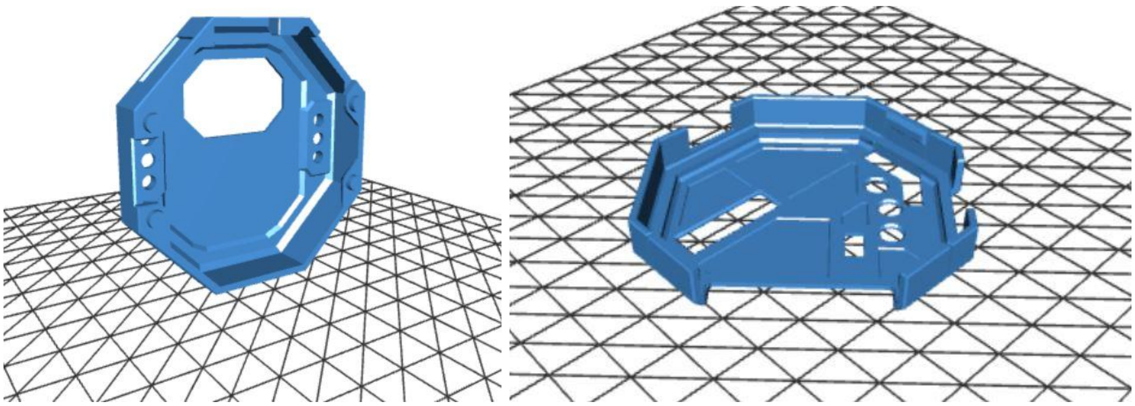

Figure 2.10 - 3D rendering of PCB case

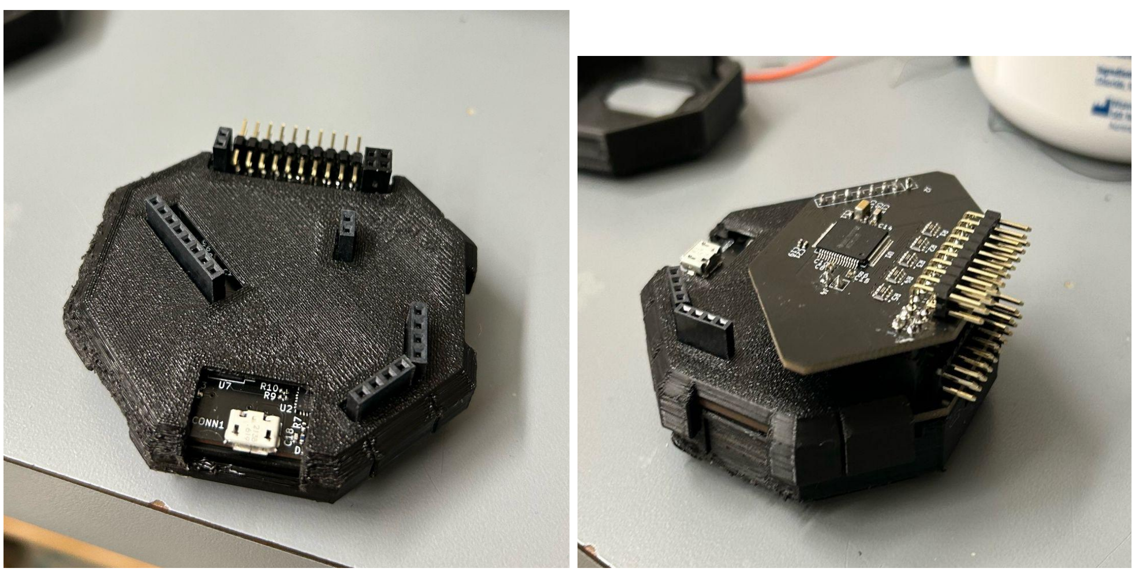

Figure 2.11 - Fully assembled case with PCBs

2.4 Installation, Setup, and Support

2.4.1 Hardware

The headset and electrode probes come assembled out of the box with 16 electrodes placed in the rear of the headset body. The PCB is mounted in its casing on the back of the headset as well, with the probes attached to their correct pins.

- Begin by attaching the included USB to the USB port that is visible on the Cyton board.
- Plug the opposite end into any USB port on your PC. Ensure power is being sent to the board indicated by a blue LED on the top side of the Cyton. Once this LED

is on, the hardware is ready to boot up software. If you would like to change the electrode positions, gently remove the probe by squeezing the sides of the outer electrode casing and pulling outwards. You should never change where the electrodes are connected to the boards, but if you choose to do so refer to the layout described in section 2.3.4. Once you are ready to begin, connect an ear clip to each of your earlobes and place the headset on your head with the board on the back. Make sure that each of the probes is making contact with the scalp.

2.4.2 Software

All of the software runs on Python3 with necessary libraries outlined in the 'requirements.txt' file contained in our Github repository. Clone our repository either via SSH or HTTP to download the necessary scripts. Running the command 'pip install -r requirements.txt' will download each of the required libraries. To begin searching for a board, execute the boardStreamer script. Optional command-line arguments include -p to specify the data port (default=800), -m to specify the mode (either 8 or 16 channels, default=COM) and -w to specify the number of samples per packet (default=10). These arguments are not necessary, and the program should automatically identify a board once it is connected to the computer's USB port. Boot up the virtual environment GUI by executing the 'main.py' file contained in the Virtual Environment directory. Once this is running, ensure that an active connection is successful to our boardStreamer code.

3 Operation of the Project

The Brain 4-Ce has two operable modes. Both modes require the same device setup prior to operation and are outlined in section 3.1. Following the device set up the end user can either start up the OpenBCI GUI to visualize the EEG in real time (section 3.2) or the user can begin the demo we created in order to play a game to showcase real-time motor imagery classification (section 3.3).

3.1 Device Setup

The device comes pre-assembled with the probes placed and electrically connected; the user simply needs to place the headset on their head. For demonstration, Professor Osama allowed us to utilize him as an example. In Figure 3.1 he demonstrated proper headset orientation.

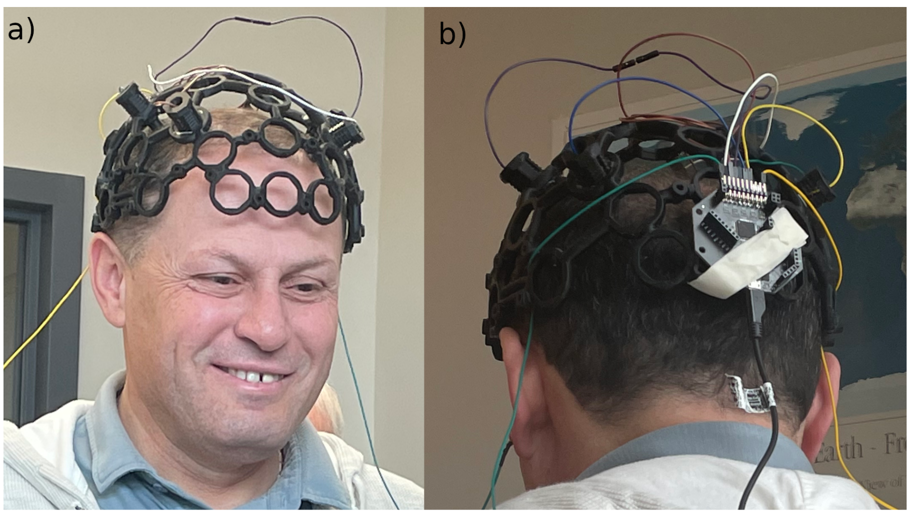

Figure 3.1 - a) Professor Osama wearing the headset from the front. b) Professor Osama wearing the headset from the back

After the Device is placed on the head, there will be two dangling cup-shaped probes that need to be placed on the user’s lower earlobe in order to get clear signals from the probes. Professor Osama again shows what this looks like in Figure 3.2.

Figure 3.2 - The bias and reference nodes used for noise rejection

After the device is secure the end user simply needs to plug the device into a computer in order to begin operation. After the device is plugged in, the blue LED shown in Figure 3.3 will light up, indicating an operational state.

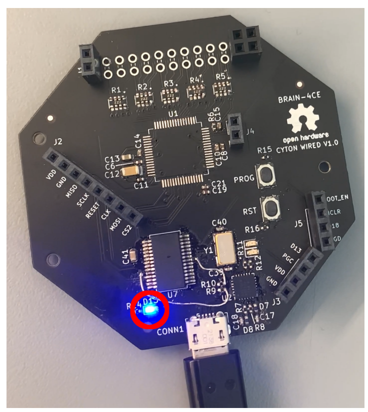

Figure 3.3 - The blue power indicator LED

3.2 Operating Mode 1: OpenBCI GUI

After following the installation procedure for the OpenBCI GUI. Launch the GUI by building it using the previously outlined procedure. Connect the device to the user as explained in section 3.1. Then connect the board to the computer to power it on. After this is completed the remainder of the procedure is equivalent to that outlined by OpenBCI in their documentation found here.

3.3 Operating Mode 2: Motor Imagery Demonstration

In order to run the motor-imagery[1] demo you simply need to go into the root of the project's repository and run the board streamer script. In order to find the valid options for the script simply run `python boardStreamer -h` in order to get the help menu. Figure 3.4 gives an example of the help menu.

[1] Motor-imagery refers to the brain activity that precedes actual bodily motion

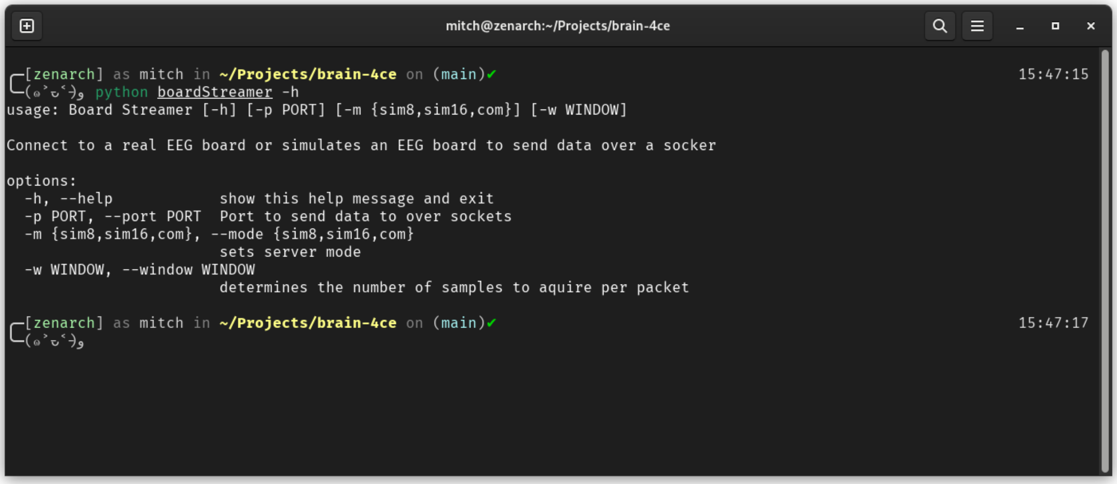

Figure 3.4 - Help menu for the demo program

After running the script (no input arguments required) python boardStreamer then you should receive output saying the board was connected to and the virtual environment should load. See Figure 3.5 as a reference.

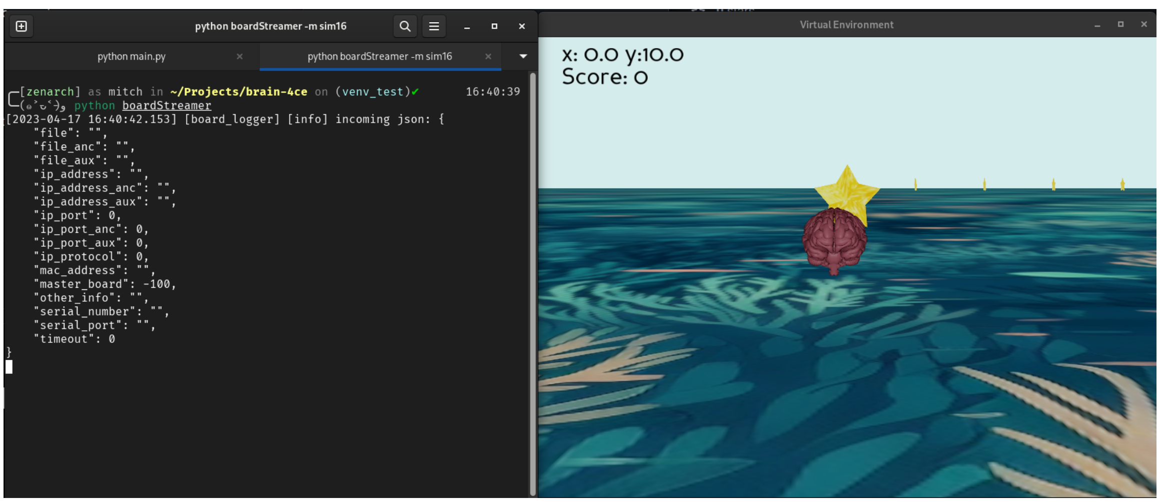

Figure 3.5 - Demo running with board logging (left) and the virtual environment (right)

4 Technical Background

4.1 Hardware Component

The hardware of the system is built on OpenBCI's previous work on the Cyton and Daisy printed circuit boards (PCBs), which are capable of electroencephalogram (EEG) data collection, as well as OpenBCI's Ultracortex Mark IV headset. The Cyton functions as the data collection hub of the system. It contains an ADS1299 analog front-end chip and a PIC32MX250F128B microcontroller, which together handle converting the raw analog EEG data obtained from the Ultracortex headset into a digital representation that can be interpreted by a computer. Additionally, the Cyton contains two push buttons for resetting and programming the microcontroller over USB. The micro-USB connector connects to the microcontroller through a CP2102N USB-toUART bridge controller. This controller converts USB data to UART protocol data and vice versa, enabling a wired communication channel between the microcontroller and a computer. The Daisy functions as an extension of the Cyton board, doubling the number of data channels from 8 to 16. This is accomplished through a second ADS1299 chip attached to the Daisy and wired to the Cyton via through-hole connectors. Both the Cyton and the Daisy have angled header pin connectors, which serve as the connection point for each of the 16 electrodes on the Ultracortex headset. These angled connectors are wired to the ADS1299s through a series of resistor arrays to limit the current flowing into the chips.

The Ultracortex Mark IV headset consists of a 3D-printed frame, a 3D-printed housing for the Cyton and Daisy PCB system, and 16 electrode probes with 3D printed housings, soft jumper wires, and dry electrodes with spiky tips. The electrode probes are configured in a specific orientation on the head as determined by principal component analysis of EEG datasets. This determined the optimal electrode placement to obtain the best data for machine learning classification. The probes are spring-loaded for better comfort on the head. Each probe terminates in a female soft jumper wire, which slots into the angled connectors on the Cyton and Daisy boards.

4.2 Machine Learning Component

The machine learning component of the system uses a Convolutional Neural Network (CNN) to classify motor-imagery, which involves several steps. Firstly, data obtained from OpenBCI's Cyton and Daisy EEG headset is collected, preprocessed, and labeled. The data is preprocessed by filtering, segmenting, and extracting features from the EEG signals. Then, it is labeled with the respective motor-imagery task, such as right-hand or left-hand movement. This labeled dataset is then used to train the CNN model. The CNN model is a type of deep learning model that is particularly well-suited for image recognition tasks, such as recognizing patterns in EEG signals. The training process involves feeding the labeled data into the CNN, which learns to recognize patterns in the EEG signals associated with each motor-imagery task.

After the CNN model is trained, it is used to classify motor-imagery tasks in real-time while someone is wearing the headset. The EEG signals are continuously recorded and processed in real-time using a Python library called BrainFlow. The CNN model is then used to classify each segment of the EEG signal into one of the motor-imagery tasks. The classification results are then passed to a Python program that displays the commands as motion of a 3D object. This Python program can be customized to display the commands in any way desired, such as controlling a robotic arm or a video game character. The system's real-time classification capability makes it suitable for a wide range of applications, such as brain-computer interfaces (BCIs) and assistive technologies for people with disabilities.

4.3 3D Virtual Environment Component

The 3D virtual environment component is controlled using the output of the CNN, which is mapped to specific motor-imagery tasks, as mentioned previously. The environment is built using the Panda3D library in Python. It is designed as a game where the user's objective is to move a 3D object around a flat plane to collect stars. The motion of the 3D object is controlled entirely by the user's motor-imagery thoughts, which enable them to navigate through the game environment hands-free. The virtual environment includes visual and auditory feedback for users whereby the stars disappear once collided with and a score counter is incremented. Overall, this technical approach offers an immersive and interactive method for users to utilize their motor-imagery skills while playing a stimulating and challenging game.

5 Relevant Engineering Standards

IEEE/ISO/IEC 11073-20701-2020/IEEE 11073-20702-2016

Standard 20701 enables communication between medical devices and external computer systems. It provides automatic and detailed electronic data capture of patient vital signs information and device operational data. The primary goals are to: Provide real-time plug-and-play interoperability for medical devices and facilitate the efficient exchange of vital signs and medical device data.

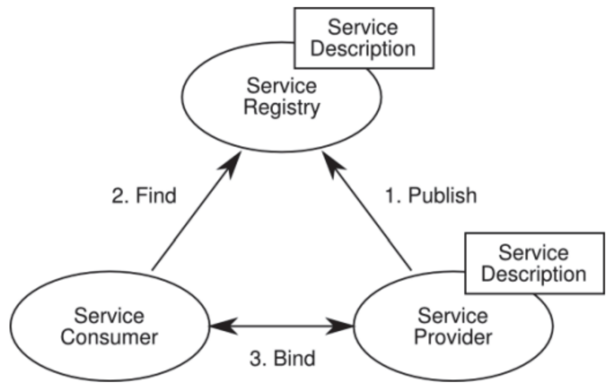

Figure 5.1 - Visual representation of service-oriented medical device exchange architecture

By following a service-oriented medical device exchange architecture, a Service Provider announces itself with an intended operation/task (contract) to a Service Registry. Hence, the Service Registry is a Service Provider that provides contracts to a Service Consumer and thus facilitates the discovery of tasks/functional capabilities. These services are as defined in standard 20702, which uses the announcement protocol to enable data streaming over UDP/TCP.

Using these standards, we are able to create a functional approach to the socket programming method of sharing data across the pieces of the project. It is a clear outline of how to create a closed clinical environment in which to collect, share, and use data as needed.

IEEE P2731

IEEE P2731 provides a standard that establishes terminologies and definitions used in the description of Brain-Computer Interfaces. This standard is multi-pronged. In terms of terminology, it creates a BCI Glossary to provide a clear explanation to all stakeholders of the BCI terminology which is sometimes misused, ambiguous or difficult to comprehend by some stakeholders (e.g. users vs. neurologists vs. engineers). It also aims to define a Functional Model of BCIs, so that all BCIs could be described according to a singular baseline model.

6 Cost Breakdown

Item	Quantity	Description	Unit Cost	Extended Cost
PCB	1	Manufactured with OpenBCI schematic	$1.40	N/A
PCB Hat	1	Manufactured with OpenBCI schematic	$1.40	N/A
ADS1299	2	Low-Noise, 8-Channel, 24-Bit Analog-to-Digital Converter	$61.97	$123.94
PIC32	1	32-Bit Single-Core 40MHz 128KB	$5.30	N/A
USB to UART Microcontroller	1	USB 2.0 UART Interface 24-QFN (4x4)	$2.16	N/A
Micro USB Connector	1	Receptacle Connector 5 Position Surface Mount, Right Angle; Through Hole	$2.02	N/A
18-8 Stainless Steel Screw Pack	1	Number 4 Size, ¼" Long, Packs of 100	$2.31	N/A
18-8 Stainless Steel Hex Nut Pack	1	2-56 Thread Size, ¾" LOng, Packs of 100	$3.90	N/A
Springs	17	302/304 Stainless Steel with 5.5 total coils and Closed ends	$2.44	$41.48
Gold Cup Electrodes	1	1 meter	$44.99	N/A
Conductive Paste	1	8oz Jar	$20	N/A
			Beta Version-Total Cost	$248.90

The main component of the EEG-Based BCI is the printed circuit board which can be manufactured from a 3rd party vendor. Specific components (resistors, capacitors, LEDs, etc.) for the PCB may be found under the OpenBCI documentation. A MicroUSB and UART converter are needed to send information to the CPU. Springs, electrodes, and conductive paste are used for the 3D-printed headset.

7 Appendices

7.1 Appendix A - Specifications

Our final product includes a lightweight, comfortable headset containing 16 EEG electrodes. Self-assembled PCBs act as the heart of the hardware, containing highly sensitive analog to digital converters and USB compatibility. A plug-and-play interface allows anyone to use the headset with ease and offers multiple modes for users to explore.

7.2 Appendix B – Team Information

From left to right: Brendan Shortall, Mitchell Gilmore, Dayanna De La Torres, Jonathan Mikalov, Alexander Johnson

Our team is composed of 4 Computer Engineers and 1 Electrical Engineer who have put countless hours into this project. Without the hard work and dedication of this team, bringing this project to completion would not have been possible.

Mitchell plans to further his education in pursuit of a PHD from Boston University in Electrical Engineering.

Dayanna has landed a well-deserved position at a highly accredited IT company, Accenture, where she will begin full-time employment following graduation.

Jonathan plans to work full-time at Watts Water Technologies as an embedded systems engineer. After completing an internship there last Summer, he was offered a full-time position following graduation.

Brendan and Alex are still in pursuit of a full-time position following this semester, with a plethora of applications and interviews scheduled in the near future.