RAISE: Building an Arduino-Based Brain-Computer Interface and Writing a Real-Time Signal Plotting Program

Chanha Kim (Computer Science, 2022)

Abstract

With the recent progress of startups like Neuralink and Neurable, *brain-computer interfaces* are emerging as 'the next big thing' in the consumer tech market. A brain-computer interface or BCI is a device that can read and interpret neural signals, such as electroencephalography (EEG). BCIs are interdisciplinary, combining neuroscience, circuits, signal processing and machine learning, and their immense complexity often requires a team of experts to create a novel system. This project aimed to understand basic neuroscience and circuit concepts and resulted in creating an Arduino-based BCI and a real-time signal plotting program (mBCI Lab). Specifically, an Arduino Due and an EEG circuit design from an existing BCI project were used to create the BCI circuit, and the real-time signal plotting program was based upon a Python-based, real-time graphing project by Sepúlveda et al. (2014). In the future, I will delve into more advanced neuroscience and circuits topics—to further deepen my understanding of BCI technology—and learn how to apply machine learning techniques to interpreting neural signals.

Introduction

Guiding Question

How do brain-computer interfaces work and how do I build one?

Project Goals

Learn how brain-computer interfaces work and build
a minimal BCI circuit + a real-time signal plotting program.

Background

BCIs are systems that read and interpret signals from the nervous system to enhance the interaction between humans and their environments. Numerous types of BCIs exist and are differentiated mainly based on the signal that they read (e.g. EEG vs MEG) and whether the technology is invasive(e.g. invasive vs non-invasive EEG). The non-invasive, EEG BCIs, which are explored in this project, read electroencephalography (i.e. the electrical activity of the brain) by attaching electrodes to the scalp without entering the skin. Non-invasive EEG BCIs are implemented as electronic circuits and require modules that amplify and filter the brain's electrical signal (measured in volts). Leveraging BCI technology enables biomedical engineers to develop neurologically seamless prosthetic limbs, helps researchers study the physiological processes of cognition, and allows consumers to interact with their environments on a more intimate level. [1]

Approach

Building the Brain-Computer Interface

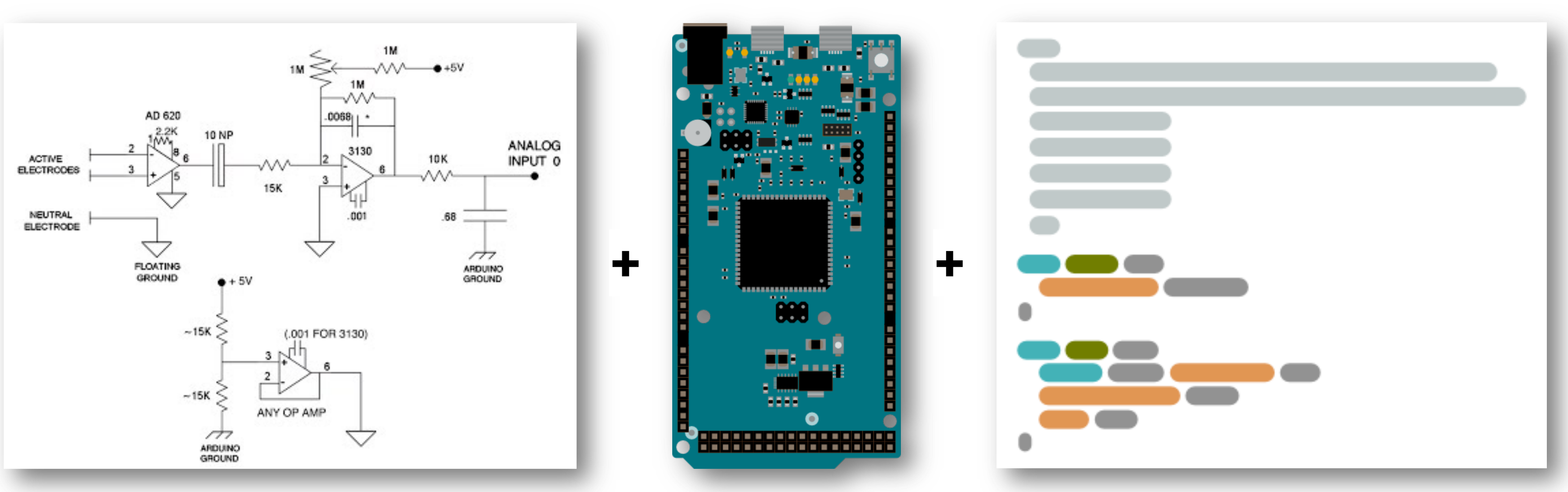

Figure 1. A diagram of BCI components. The three components are the circuit design (left) borrowed from chipstein's "EEG With an Arduino Project" [2], the Arduino Due (middle) [3], and the Arduino firmware code sampling at 100 Hz (right).

Building the Real-Time Signal Plotting Program

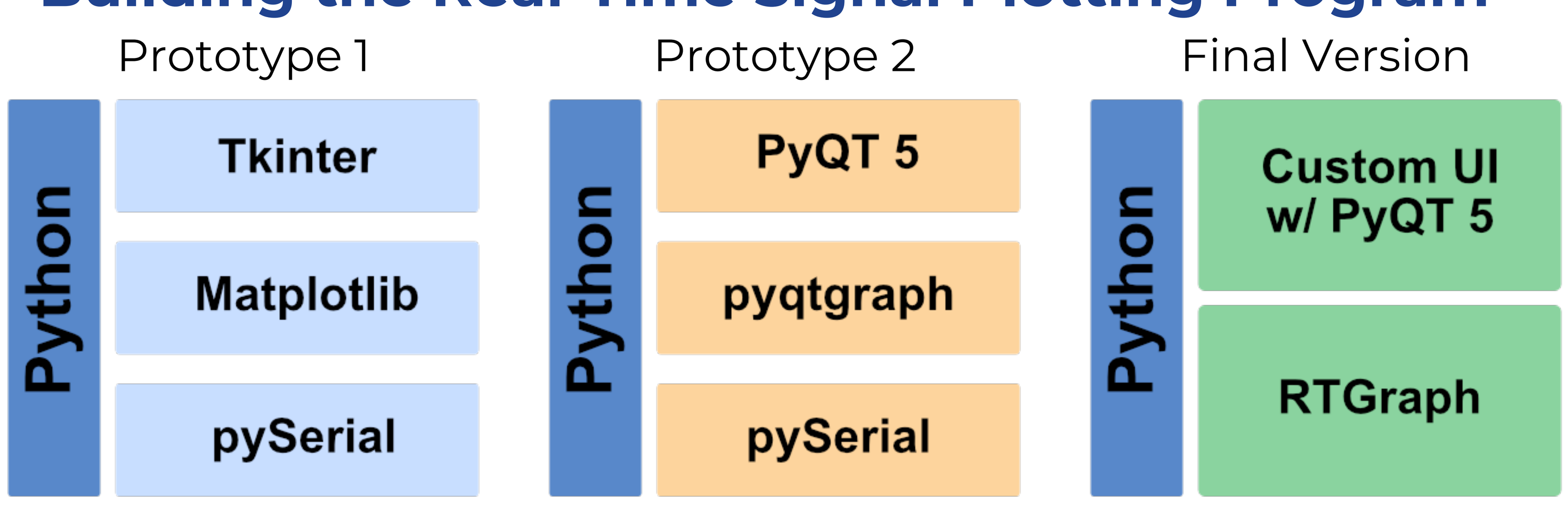

Figure 2. A diagram of UI tech stacks. A tech stack is the set of technologies a program uses. Each tech stack corresponds to a prototype real-time signal plotting program created in this project. The final version of the program uses RTGraph, a real-time signal plotting program implemented in Python [4].

Results

mBCI (a minimal brain-computer interface)

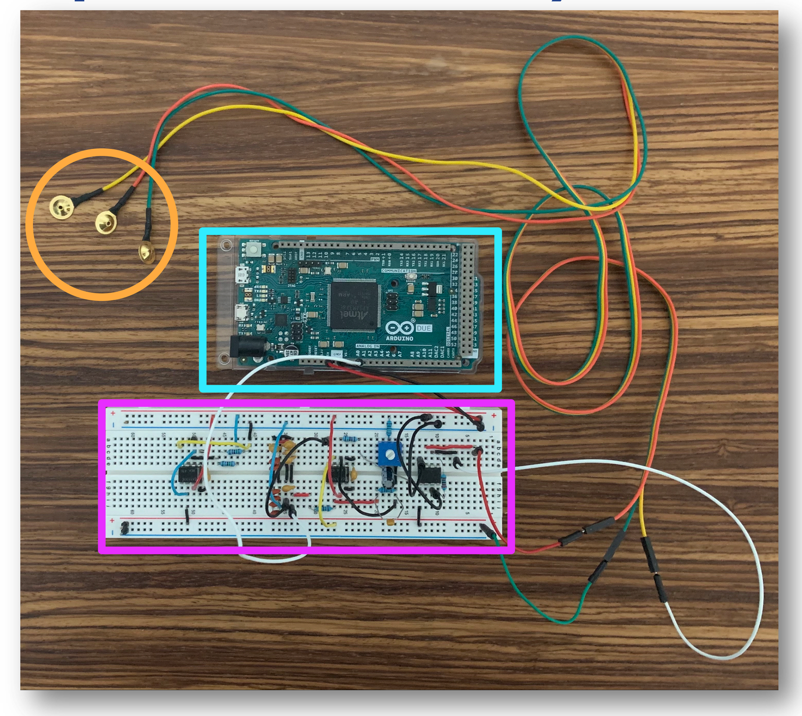

Figure 3. An image of a minimal BCI. The circuit uses an Arduino Due (boxed in cyan) and connects to a computer via a micro-USB to USB cable (not pictured). The electrodes (circled in orange) connect to the locations on the scalp associated with the desired neural signal, and the amplification/filter module (boxed in light purple) amplifies and filters the neural signal. Amplification increases the strength of the signal, and filtering removes the unwanted noise in the signal. This BCI circuit is minimal in the sense that it reads neural signal from a single location and utilizes a relatively basic amplifier-and-filter module.

mBCI Lab (a real-time signal plotting program)

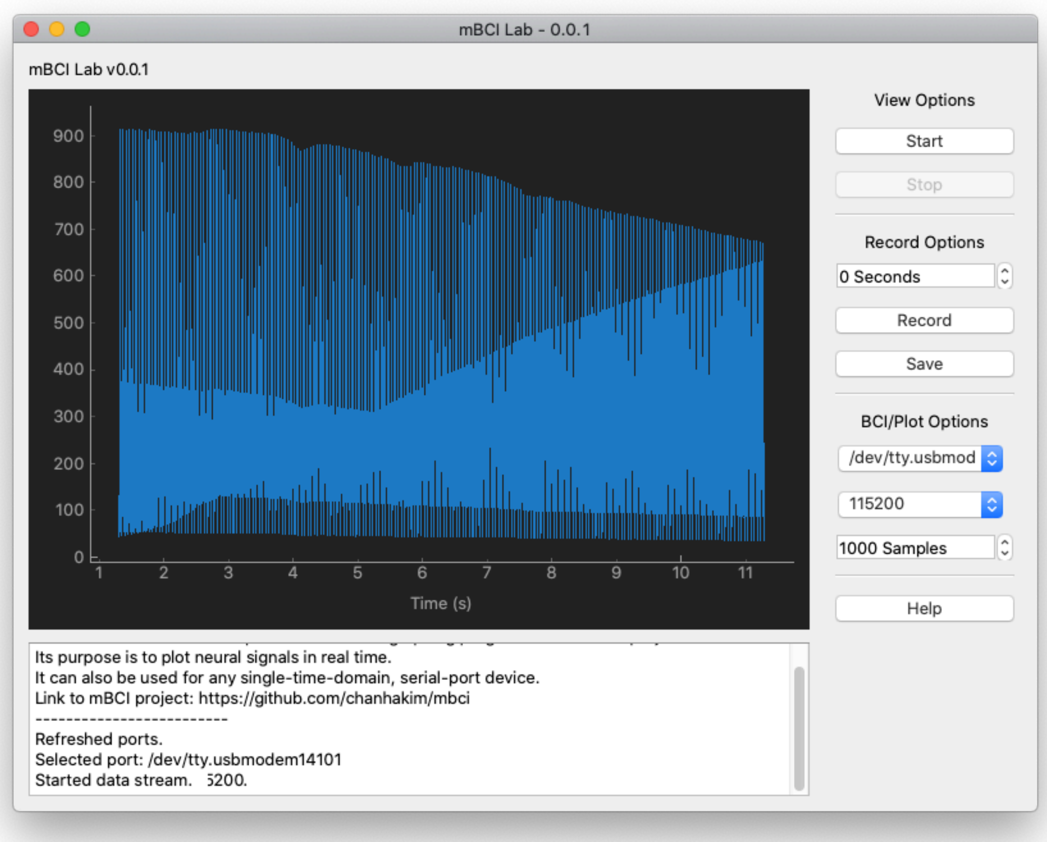

Figure 4. A screenshot of mBCI Lab. mBCI Lab is based on Sepúlveda et al.'s RTGraph project, a real-time signal plotting program written in Python. Specifically, mBCI Lab utilizes the backend processing of RTGraph and introduces a new graphical user interface for interacting with the mBCI. The former is written as a fork (or copy) of the latter. *The signal present in this screenshot is random noise from a BCI that is not connected to a subject.

Key Takeaways & Next Steps

Key Takeaways

- BCIs are systems that read and interpret neural signal. mBCI can read single-channel EEG at 100 or 200 Hz.
- In the context of non-invasive EEG BCIs, neural signal is electro-encephalography (measured in volts). EEG arises from populations of neurons that release electrical signal towards the skin surface.
- Circuits are collections of wires, capacitors, resistors, and other electronic components that direct the flow of electricity.
- Differential amplifiers can read the electrical differences between two scalp locations and pass the measured neural activity along to a computer as digital values.

Next Steps

- Review circuit fundamentals and learn more advanced circuits topics.
- Collect neural data based on signal acquisition techniques found in academic literature.
- Analyze neural data using neural signal analysis.
- Leverage machine learning to calibrate different frequencies of neural signals to actionable tasks (e.g. moving a cursor on a screen).

Project Links

mBCI: www.github.com/chanhakim/mbci

mBCI Lab v0.0.1: www.github.com/chanhakim/mbci_lab

References

1. Wolpaw, J.R. & Wolpaw, E.W. (Eds.). (2011). *Brain-Computer Interfaces: Principles and Practice*. Oxford.
2. Chipstein. (N.d.). *EEG With an Arduino*. chipstein. https://sites.google.com/site/chipstein/home-page/eeg-with-an-arduino
3. Arduino. (N.d.). *Arduino Due*. Arduino Store. https://store.arduino.cc/usa/due
4. Sepúlveda, S. (2015, January 1). *RTGraph*. GitHub. https://github.com/ssepulveda/RTGraph

Acknowledgements

I 'd like to thank Professor Osborn for his mentorship, the Fletcher Jones Foundation for funding my project, and my friend Manu Patil for answering my questions about circuits.