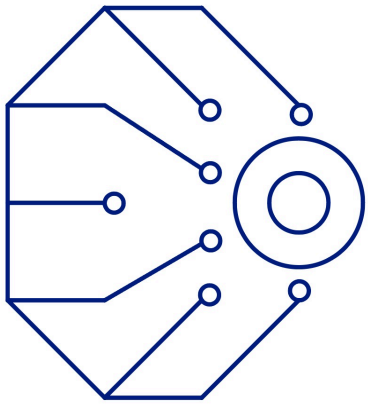
Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

First Prototype Test Report



NeuroToys

by

Team 9
NeuroToys

Team Members

Adam Shaikh adamparz@bu.edu
Andrés Marquez Santacruz afms@bu.edu
Gabriela Porto Machado gmpm@bu.edu
Mete Gumusayak mgumus@bu.edu
Robert Bona rjbona@bu.edu

Materials and Setup

Our prototype for NeuroToys integrates hardware and software to create a brain-controlled RC car. The core hardware includes an ESP-WROOM-32 microcontroller, an L298N motor driver, two 18650 lithium-ion batteries in a 2S holder in series with a 32V 3A glass fuse for safety, a personal computer, and a NeuroSky Mindwave Mobile 2 non-invasive EEG headset. The EEG, powered by AAA batteries, collects brainwave signals and transmits data to the computer via Bluetooth. A Python-based interface processes and classifies the brainwave data and sends commands to the ESP32 via Bluetooth. Preloaded C code on the ESP32 interprets these commands to control the toy's movement.

Figure 1: Control flow diagram

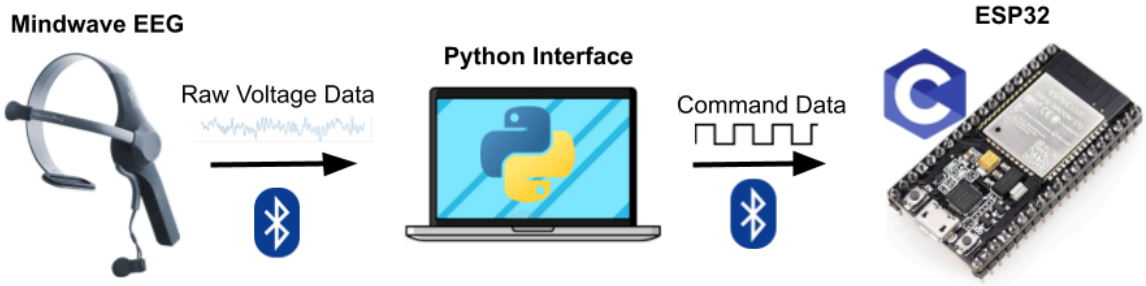
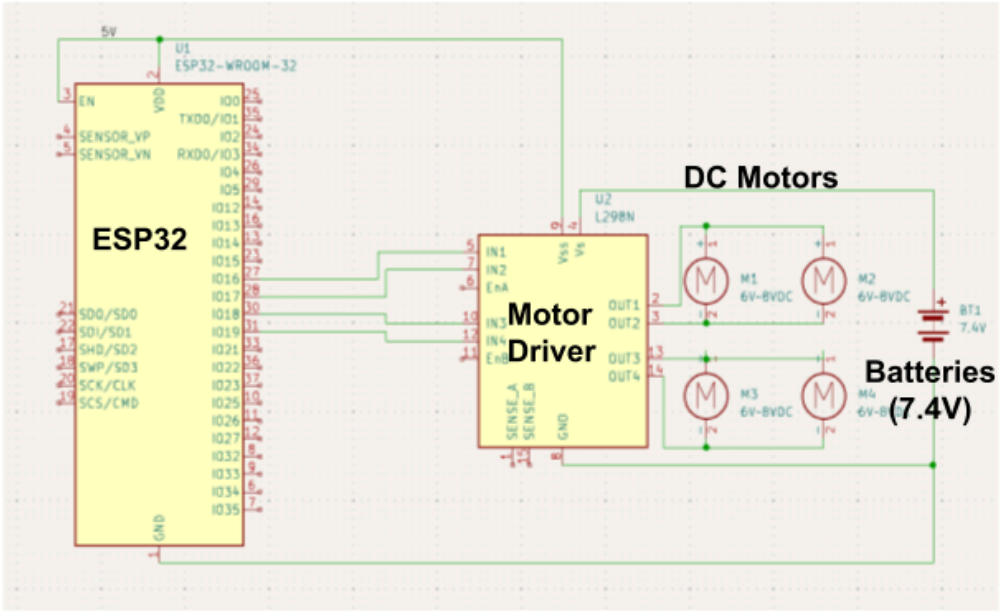


Figure 2: RC car circuit including pinouts



Measurements Taken

The headpiece transmits raw brain voltage data (μV). The Python interface processes this signal by performing a Fourier Transform to isolate the beta frequency band from the EEG signals, which is the range associated with focus. The beta power is then calculated (expressed in μV^2 or dB), representing the user's focus level. A threshold is established to determine whether a command should be sent to the ESP32, which then controls the forward movement of the toy car.

To test the efficiency of our prototype, these focus levels were measured and plotted, as well as a visual discernment of whether the toy moved once the user was intentionally focusing.

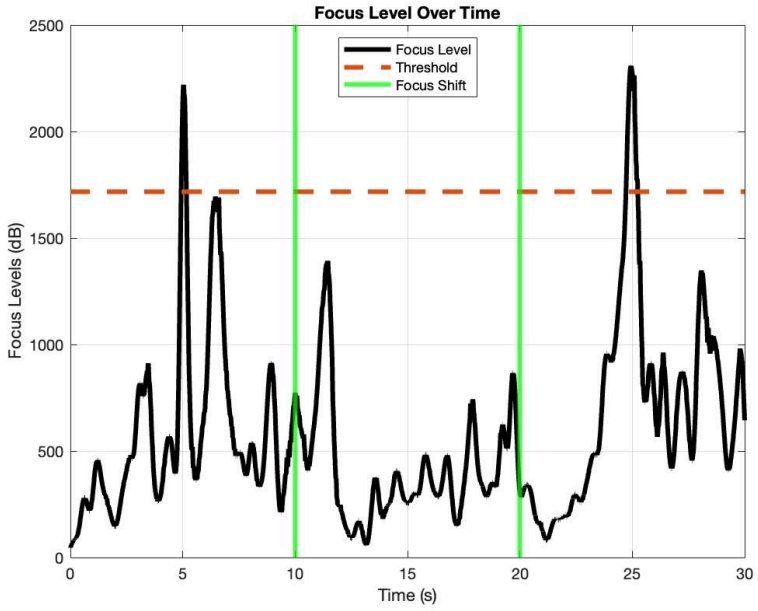
Procedure

1. Member 1 wears the headset.
2. Power is supplied to the toy via on/off switch.
3. The Python script is run. A Bluetooth connection will be established before continuing.
4. Member 2 starts a 40 second timer.
5. Member 1 should attempt to unfocus in the first 20 seconds, then Member 2 announces the focus shift, where member 1 attempts to focus and command the car to move
6. Movement of the toy is enabled at above the dynamic focus level threshold, and disabled when below. Spikes in beta power indicate intense focus, which should coincide with each announced focus period.

Results

Initial Test Data

Figure 3: Single Pre-Trial Illustrating Focus Level vs. Time

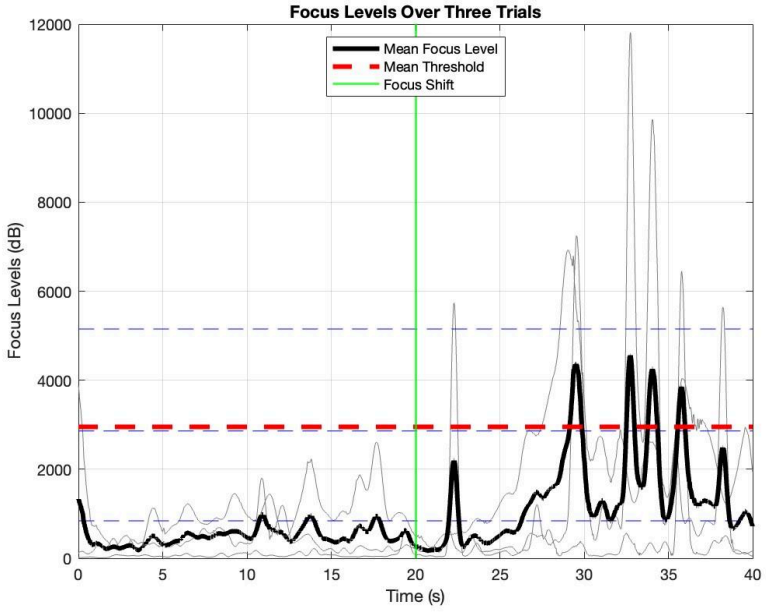


The above figure illustrates one of our initial trials, recorded during the troubleshooting phase of the prototype, before it was connected to the toy. It shows evidence that our system can correctly classify focused and unfocused brain signals.

The next step was to connect to the ESP32, using the data so that a received signal above the focus threshold (red dashed line) enables the toy to move forward, and to stay still otherwise.

Complete Test Data

Figure 4: Full Testing Illustrating Focus Level vs. Time



The above figure illustrates the test data recorded after the ESP32 connection and commands were established and using the procedure outlined previously. When the threshold was reached, successful forward movement of the car occurred.

From this, we can conclude that the Brain Computer Interface was successfully implemented in the initial prototype. Moving forward, we aim to have the movement occur more smoothly, without sudden bursts of action. In addition, the group aims to introduce a more complex classification modeling when sending these commands, allowing for front, back, right, and left distinctions.