Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

First Prototype Testing Plan

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

Required Materials

Hardware:

- ESP-WROOM-32 with Onboard LED, Breakout Board
- L298N Motor Driver
- 18650 3.7V Battery (2)
- 2S 18650 Battery Holder
- Muse 2 Non-invasive EEG headset and USB charger
- Personal computer with relevant project files installed
- Assembled RC car

Software:

- Python
  - Acquires EEG headset connection
  - Processes and classifies brainwave data
  - Bluetooth interface between EEG and ESP32, sending command signals
- C
  - Pre-uploaded to ESP32 to accept commands via Bluetooth

Set Up Summary

There are three primary components to our system: the Muse 2 EEG headpiece, a computer which runs the Python interface, and the ESP32. 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 data, which is associated with focus. The beta power is then calculated (expressed in μV^2), 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. A similar signal processing technique is used for right and left control, where the AF7 and AF8 electrodes on either side of the forehead go through a band pass filter separately to isolate the gamma frequency band from the raw EEG data. Peak detection algorithms are then used to classify between right and left eye blinks, which in turn send the right and left turn commands to the car.

Figure 1: Control flow diagram

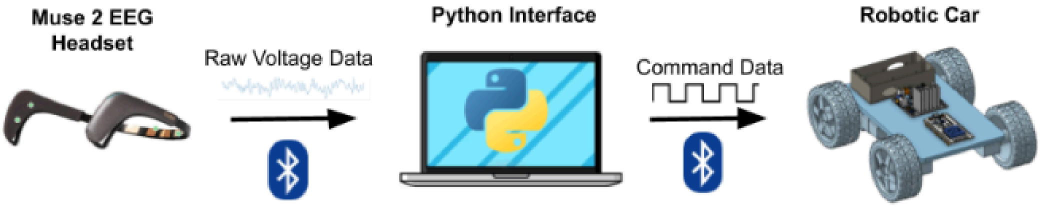
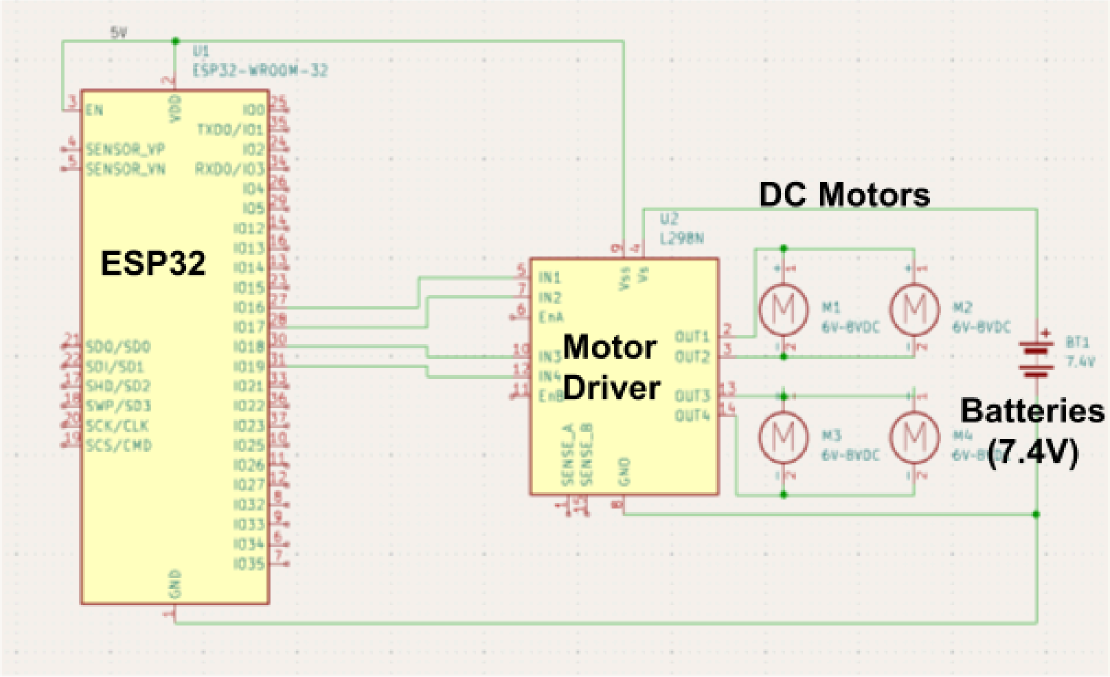


Figure 2: RC car circuit including pinouts



Pre-testing Setup Procedure

1. Make sure the headgear is fully charged, muse repository is downloaded, and all packages are installed
2. Supply power to the toy via on/off switch
3. Turn on headset
4. On Terminal, run: `muselsl stream`
5. Wait for connection to be established to begin testing procedure

6. Place the headset along the middle of your forehead with the rubber ear sensors resting behind your ears. The earpieces should sit behind your ears like a pair of glasses. Adjust both sides simultaneously to tighten it back up for a snug fit that feels comfortable.
7. On a new terminal window, run: `python main_gui.py` to open the application

Testing Procedure

Focus

1. Member 1 wears the headset
2. Member 2 starts a 1 minute timer
3. Member 1 should attempt to focus and unfocus at 10 second intervals. These intervals are announced by Member 2 at the start and each switch.
4. Member 2 marks down errors as they occur at the start of each interval.
5. Forward movement of the toy is enabled at above the dynamic beta power (focus level) threshold, and disabled when below.

Blinking

1. Member 1 wears the headset
2. Member 2 starts a 1 minute timer
3. Member 1 should attempt to blink their right eye for 10 seconds at a frequency of one blink per second, repeat with the left eye, and take a 5 second break. These times are announced by Member 2 at the start and each switch.
4. Member 2 marks down errors as they occur at the start of each interval.
5. Right and Left eye blink detections command the car to turn in the direction

Combined Control

1. Member 3 places the stop
2. Member 2 starts the stopwatch
3. Member 1 attempts to make the car move forward, turn in the clockwise direction, and return to its original position

4. Member 2 pauses the stopwatch and records the time
5. Member 3 replaces the car in the starting position
6. Member 2 restarts the stopwatch
7. Member 1 attempts to make the car move forward, turn in the counterclockwise direction, and return to its original position
8. Member 2 pauses the stopwatch and records the time

<u>Measurable Criteria</u>

Focus

I. Movement of the toy is enabled *at some point* within 5 seconds after a focus period begins and disabled *at some point* within 5 seconds after a focus period ends.

II. Errors are only considered during the 5 second period between focused and unfocused described above.

III. Maximum error rate of 2 out of 6 total intervals required to pass.

IV. Multiple trials are permitted.

Blinking

I. Turning commands are enabled in at least 5 out of the 10 blinks

II. Maximum of 1-2 blinks detected during the break period (noise)

III. Multiple trials are permitted.

Combined Control

I. Car makes it across the circuit in less than 60 seconds

Score Sheet

Interval	Error (Y/N)
Focus	
Unfocus	
Focus	
Unfocus	
Focus	
Unfocus	

Interval	# Errors
Right Blink	
Left Blink	
Break	
Right Blink	
Left Blink	
Break	

Turn Direction	Time
Clockwise	
Counterclockwise	