High Level Design Report

EG41430-01 - December 12, 2023

We’re Just Trying to Graduate

Adrienne Miller, Kate Hjorth, Caroline Cho, Addison Azar, Mike Guyette

Table of Contents

1 Introduction

Our proposed athletic trainer pads allow users to improve upon their speed, accuracy, and agility in a very simple drill. We use four identical looking pads that flash LED lights and respond to touch. The purpose is to determine the reflexes of the user based on how much time passes from the pad's LED being lit to the user hitting the pad. There is an app interface that wirelessly takes in this data, stores it, determines trends over time, and alters the drill based on user performance.

2 Problem Statement and Proposed Solution

Various fields of human physical activity test the subject's speed, endurance, reaction time, and accuracy. All sports require the use of speed and agility drills for their tryouts and daily training. Physical therapists and practitioners recuperate their patients by guiding them through various mobility drills to regain their movement capabilities. Military and law enforcement go through rigorous physical training that targets their endurance and reaction time to prepare them for the taxing jobs ahead of them. Children and teenagers in school are regularly tested on speed and endurance through antiquated forms of physical activity. Despite this large emphasis placed on reflex, accuracy, and endurance in the world of physical health, there is no standard for how to collect the data through certain movements or to analyze that data. This wide range of physical activities heavily depend on human direction, timing, setup, and analysis, which inherently leads to human error. Furthermore, the lack of standardization in tracking results and analyzing data creates difficulties in seeing improvement or progression over a period of time. The current training drills are not automatically catered towards the individual's strengths, weaknesses, or general improvement, and do not constantly progress with developments made by the user. Human analysis allows for predicting and aiding in improvement; this is inefficient and prone to human mistake, which costs the subject time and energy.

In all of these applications, the practitioner needs to clearly see what progress is being made based on updated, real-time data to help their client build on this skillset. However, one cannot easily do so from the lack of technology that interfaces between user performance and data analysis. This lack of modernization in these spheres hinders the person's ability to grow; technologically integrating these activities would prove beneficial to help further the human capacity for physical activity.

3 System Requirements

<u>Demo Day Goals</u>

- 4 circuits with addressable LEDs that light up
- Pushbutton in each circuit that detects touch

- ESP Now interfacing between all four EPSps
- App that connects to pads through ESPNow
- App interface allows for user to pick the drill they want to do
- Various types of drills to test reflex, accuracy, speed, endurance
- Real-time data collection between the time of the LED lighting up and the push button being touched
- Data storage with analysis of trends over time
- Machine learning that grow with the user based off of past trends

What capabilities are required of the embedded intelligence?

Four circuits will contain an LED and a pushbutton. The ESP32s will sense when the button is pushed, and will interface with the app via ESPNow to collect the data. The data consists of the time between the LED lighting up and the user pushing the button.

How is the device powered? If it runs on batteries, what kinds of batteries are used, how long should the system be able to run on the batteries, etc? How does the user replace/recharge the batteries?

The ESP32 circuits will be battery powered, and each touchpad will contain a 5V battery (that can be stepped up to drive the push-button) that will drive the LED, push-button, and clock. The battery needs to last for about 12 hours and will be rechargeable. The batteries will be easy to remove and replace on the touch pad design.

There are lots of requirements related to wireless interfaces. How many devices need to be supported? What range is required?

Four separate ESP32s will need to be supported. The app we will design will also need to be supported. ESPNow will connect all of these devices together. The devices will be separated by a few meters, as ESPNow supports a large range. ESPNow operates within the 2.4 GHz band from the built in antennas on ESP32.

What are the user interfaces?

The users will primarily interface with the app, as this is where they will decide what they would like their drill to focus on. The user will also be physically interacting with the touch pads, as this is what they will hit for the pushbutton to be triggered.
app , pushbutton itself

How is the system installed and used?

Each circuit is battery powered, and can be recharged. Individual ESPs in the touchpads will interact with each other through ESPNow to determine which one will be lit up. They will be interfacing with the app through ESPNow as well. The user will use the app to determine the

characteristics of the drill they want to do. The system will have the capability to be mounted to the wall or floor, depending on the desired activity.

If you project involves voltages and or currents that may be dangerous, what are safety requirements associated with your system?

Each circuit will have a protective, waterproof casing around it to withstand being hit by sweaty users. The battery used is a standard voltage. The LEDs run the risk of getting hot, so figuring out how long they can last on high power will be tested.

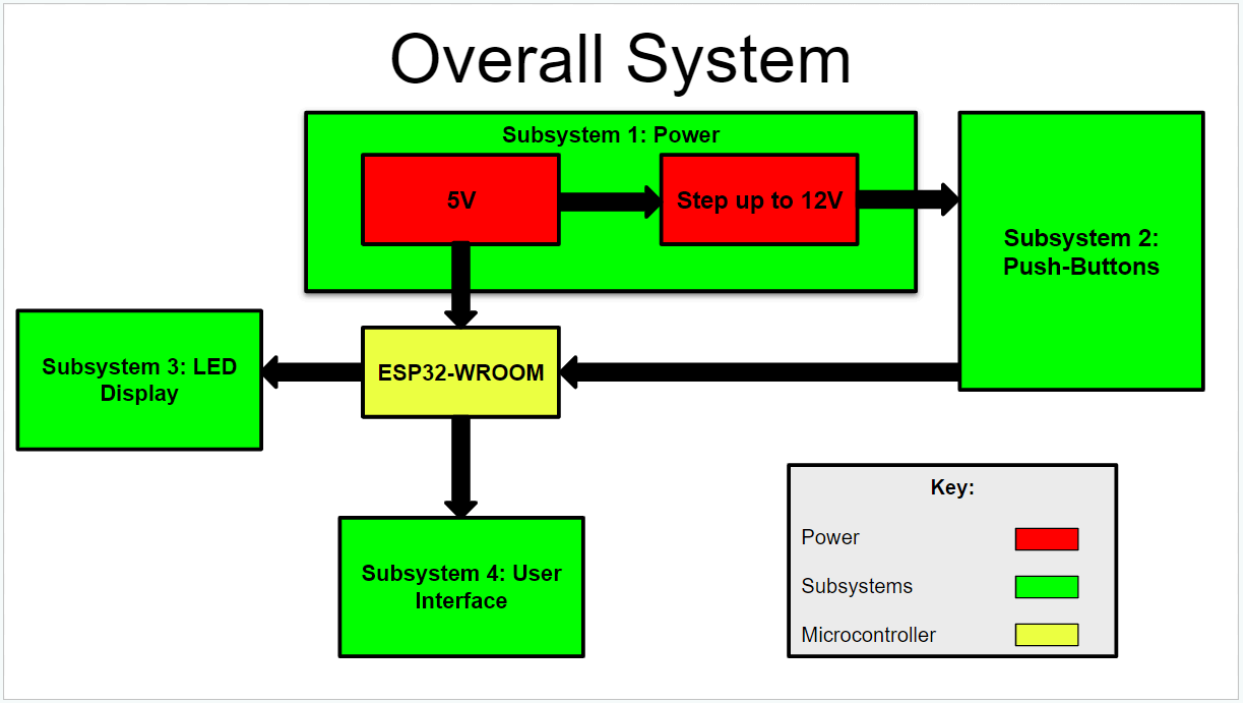
What are the mechanical requirements, such as weight, size, etc.

The current push buttons that serve as the main way to turn the devices on/off, as well as pair with the app, are 12mm. The LEDs in question contain seven 5mm x 5mm RGBW LEDs, which are mounted on a ~25 mm pad. The push buttons that the user are hitting are 12mm. The protective casing will be self-manufactured using 3D printing, and making sure the filament used is head and water-resistant. Weight and size will be determined as we begin to assemble the system.

4 System Block Diagram

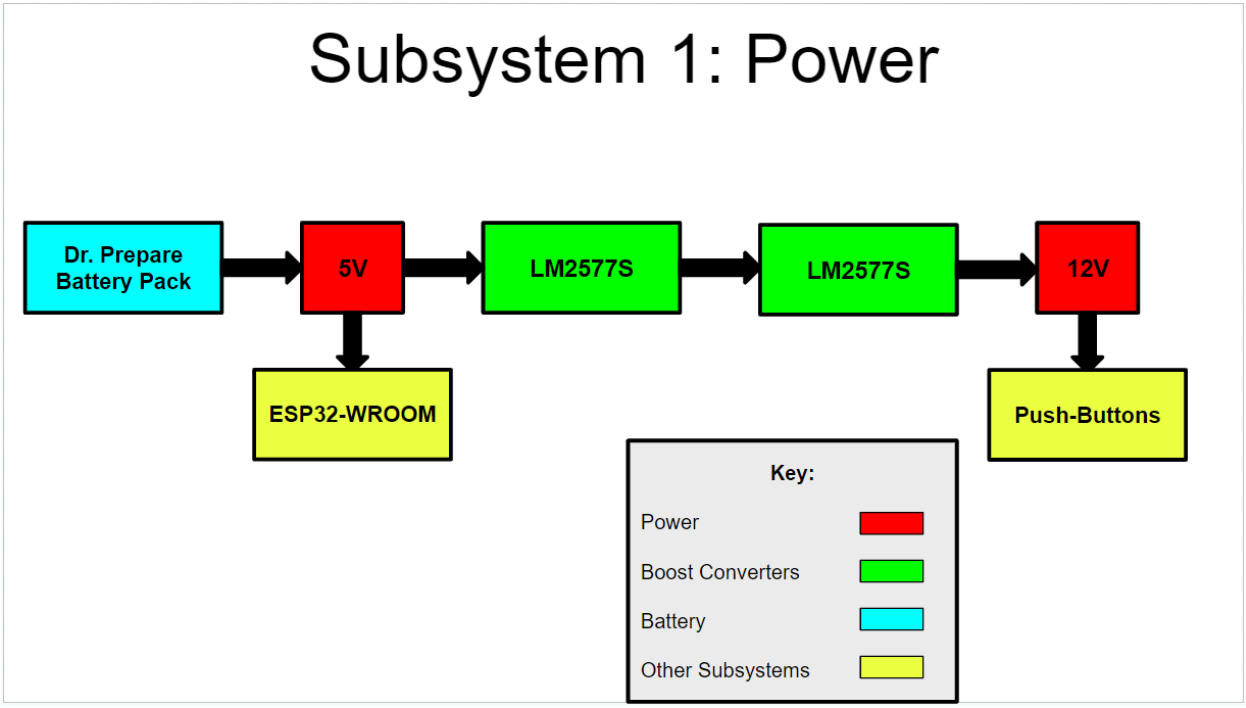
Link for All Block Diagrams

4.1 Overall System:



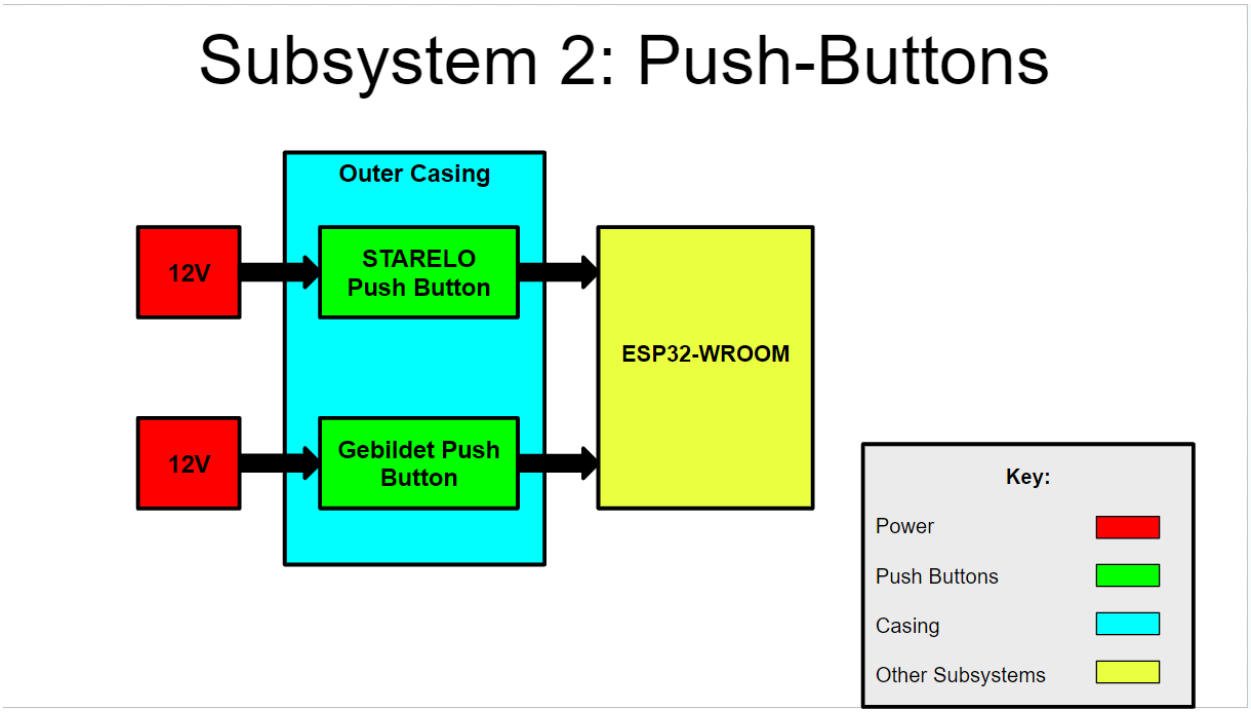
The overall system contains four subsystems. Subsystem 1 contains the power supplies. 5V directly powers the ESP32-Wroom; this 5V is also stepped up to 12V which powers Subsystem 2, which contains both types of push buttons. Subsystem 2 also directly connects to the ESP32-Wroom; this ESP32 connects to both Subsystem 3 and Subsystem 4. Subsystem 3 contains the LED display. Subsystem 4 contains the user interface.

4.2 Subsystem1 and Interface Requirements:



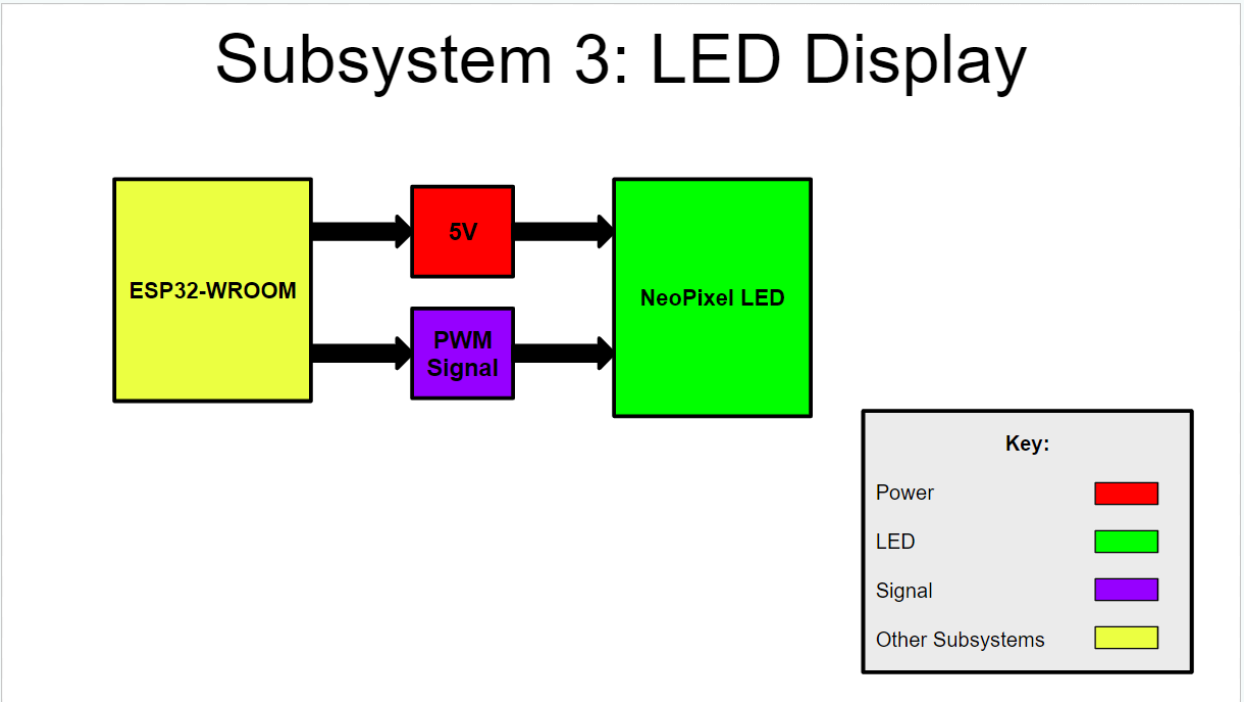
Subsystem 1 represents the power requirements necessary. The power is supplied by the Dr. Prepare battery packs located in each pushbutton. These battery packs will provide 5VDC. This 5VDC will be used to power the ESP32-Wroom and the pushbuttons. The push buttons require 12V of power and, therefore, require the use of two LM2577s voltage regulators to step-up the power. The battery packs will be rechargeable by the user.

4.3 *Subsystem2 and Interface Requirements:*



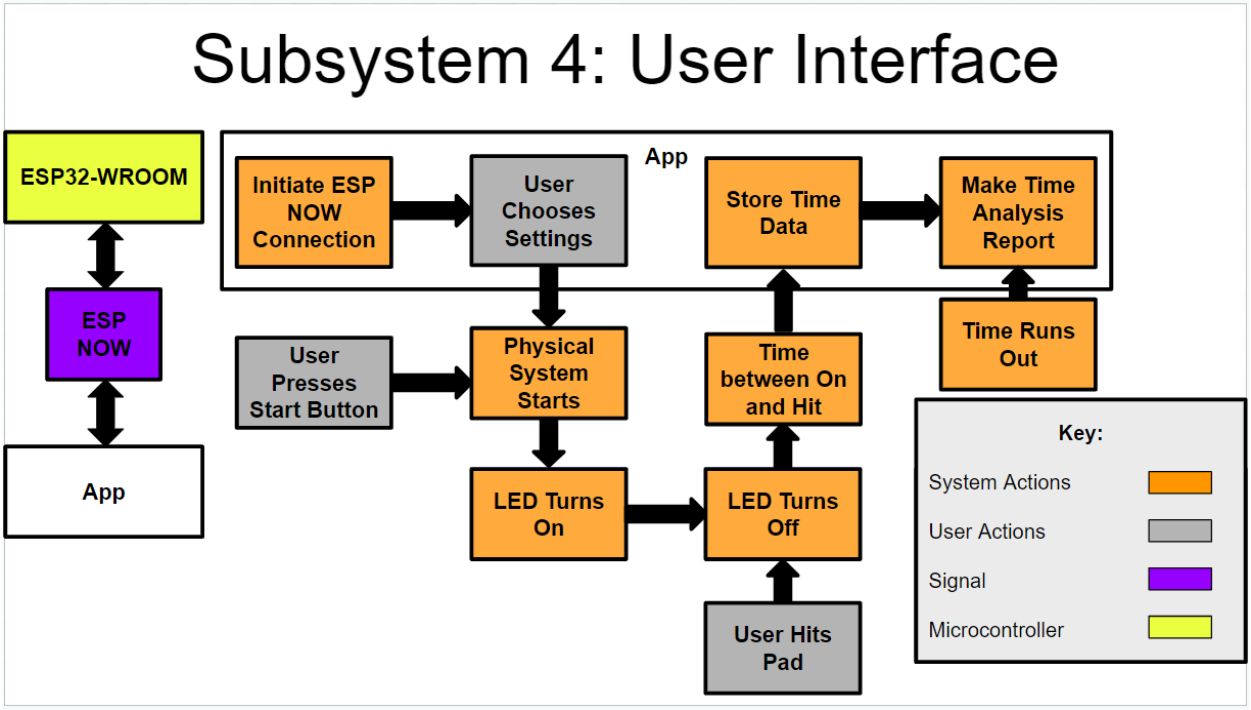
Subsystem 2 contains the pushbuttons. 12V will power the STARELO pushbutton, which serves as our main way to turn the devices on / off, as well as set up the pairing with the app. 12V also powers the Gebildet pushbutton, which serves as the main button that detects touch from the person using the system. Both pushbuttons are covered by a protective, outer casing. Both of these pushbuttons will likewise directly interface with the ESP32-Wroom.

4.4 Subsystem3 and Interface Requirements



The ESP32-Wroom interfaces with the NeoPixel LED through three, physical wires. One wire is power (5V), one is ground, and the other is for the PWM signal to determine the color being displayed.

4.5 Subsystem4 and Interface Requirements



The created app will interface with the ESP32-WROOM microcontroller via ESP NOW, a wireless communication protocol designed for Espressif systems. Within the app, the user must initiate the ESPNow connection and choose the settings for the desired program. Once the user chooses those settings and presses the external start push button, the physical system will begin. LEDs will turn on, and as the user hits the pads, those LEDs will turn off. The time between the LED turning on and the user hitting it will be measured and stored within the app. Once time runs out and all the data is collected, the app will create an analysis report for the user to view.

4.6 Future Enhancement Requirements

Possible additions we would like to make in the future are a power saver mode that ends up reducing the load off the battery between reps and an addition to the outer casing that allows for the devices to be mounted on vertical surfaces.

5 High Level Design Decisions

Power

The power supplied by the batteries in each sensor must be equivalent to 5VDC. This amount of voltage should be sufficient to provide adequate power to each component of the board, including the LEDs, push-button, and clock. We will utilize LM2577S voltage regulators to step up the power from 5V to the required 12V needed for the push-buttons. We will utilize eight of these voltage regulators in total, two for each PCB in our sensors. The LEDs only

require 5V and will not need voltage regulators to function. The integrated drivers that they come with will also protect LEDs from any voltage or current fluctuations that are unwanted. The batteries will also be utilized to power the microcontroller on the PCB. We decided to use one ESP32-C3-WROOM per PCB which will be responsible for the push-button subsystem, cue and feedback system, and communication with the ESPNow. The ESP32-C3-WROOM was chosen as our microcontroller as it possesses the capabilities to be powered by 5V, connect to the pushbuttons and ESPNow, and drive the LEDs. It, also, fits within our cost requirements. The specific batteries we have chosen to fulfill each of these functions and requirements are Dr. Prepare power banks with two USB ports rated at 5VDC 3A. One power bank will be located in the hardware of each pushbutton, will provide 10 hours of runtime, and are rechargeable.

Push-Buttons

The push-buttons will be connected to an ESP32 board and will be located under a 3D-printed pad. The main push button must be able to detect a user touch upon the user hitting the pad, the other will be used to turn the device's power and connectivity functions on. Because of its application, the buttons must be sturdy enough to withstand a forceful push, and should be resistant to heat, water, and other elements. The main button also needs to register a “push” when the user touches anywhere on the surface outer protective casing that we are using to house all of the electronics. For this reason, the Gebildet Stainless Steel Waterproof Push Button will be used to register touch by the user and the STARELO Push Button will be used to change power and connectivity settings. The Gebildet push button has two states: on and off, and the design will use the change of state to register a “push”. The Gebildet push button is made of stainless steel, therefore is sturdy enough to withstand repeated forceful touches. The push buttons are both waterproof and therefore will not break down when in contact with sweat, rain, or other water. The 12 mm surface area makes it easier to instigate the change of state when the user touches anywhere on the pad or tries to turn the device on and off.

LED Display

In order to obtain the functionality we want out of our project, we needed to incorporate an LED of some sort into each device. After doing our research, we ended up choosing the NeoPixel Jewel RGBW LED. We had three main reasons for choosing this specific part:

1. The device has seven 5050 (5mm x 5mm) individual LEDs built in. This size and number of individual LEDs is perfect for what we want as our research suggested that 5050 LEDs are bright enough to have sufficient visibility even if the user wants to operate the device outside in sunlight.
2. The device can be controlled using only one microcontroller pin, each LED is addressable as the driver chip is inside the LED allowing for multiple colors to be displayed to match our desires, and each LED requires only ~18mA constant current drive so the color will be very consistent.
3. The device requires 5VDC and is only $6.95 so they are easy to fit into our system and if we blow any of them up they are cheap to replace.

User Interface

Physically, the user will start the system with a central pushbutton, and start the mobile app on their phone. The user will be hitting four separate pushbuttons.

The user interface must be able to let the user start the system, choose a workout, store the workout, and develop new workouts based on data received from the four ESP32 boards. For connection, we choose to use ESPNow because it is a wireless transfer of data between many devices. ESPNow allows us to connect all the boards to the app at the same time without using WIFI or Bluetooth. The lack of reliance on WIFI and bluetooth gives the user freedom to use the system anywhere at any time. Furthermore, the protocol uses wirelessly-received signal strength indication (RSSI) to determine the distance between each sensor, and authorizes those distances, removing the chance of interference from external devices. ESPNow has low-power consumption, a quick response time (milliseconds), and can cover a large distance. The app will be on the user's mobile device. Software will be developed so that the app stores the data from the push buttons and analyzes the data (IE accuracy of pushing the correct button, time taken from LED lighting up and the user hitting the button, specific color being lit and hitting the correct color in the correct order). The user can pick between two modes in the app: a standardized, baseline workout, and an adaptive mode that will actively try to improve weak areas that the system identifies. All users will start with the baseline mode to gather initial data. In the adaptive mode, mew workouts will be created based on the performance in the previous workout. The app will allow the user to view their workouts, and an analysis of their performance.

6 Open Questions

As we progress with the design of our project, there are several design considerations we will have to reevaluate and reassess. In terms of power, we will have to determine how we want to recharge the battery pack, whether via a computer, a battery pack, or simply through a power block connected to an outlet. Similarly, we will need to establish how long the LEDs can function on high before they burn out, ensuring they fit the requirements necessary to run a session. We will further need to determine whether a temperature regulator will be needed to ensure that the LEDs do not reach too high of a temperature throughout their use. There are, also, several hardware specifications in regards to the protective outer casing of the touchpads that we will have to determine in the future. The necessary weight and size of each of these touch pads cannot be predetermined and will be something we assess after we have built and assembled the other components. The size requirements of the sensor pads must allow for the PCB board, batteries, attached LEDs, and push-button to fit within the protective casing. Along the same lines, we are unclear the most effective way to mount the touchpads to allow for their use in various environments, including outdoors, on walls, and on hardwood floors. We are currently considering several possible methods such as suction cups, a 3D printed holder, Velcro, and potentially a combination of a few mounting methods. Finally, we will need to make several determinations in the future in regards to our app interface. This will include conducting further research in respect to the ESPNow and how to code it onto the ESP32 as well as further research into app development to ensure it can achieve the functions outlined by our project.

7 Major Component Costs

Component	Unit Cost	Quantity	Total Cost
Dr. Prepare Battery Pack	$21.99	4	$87.96
ESP32-C3-WROOM-02-H4	$2.00	4	$8.00
STARELO Stainless Steel Waterproof Push Button	$12.49	1 (1 unit includes 5 buttons)	$12.49
LM2577S voltage regulator	$8.76	8	$70.08
NeoPixel Jewel - 7 x 5050 RGBW LED	$6.95	4	$27.80
Gebildet Stainless Steel Waterproof Push Button	$7.99	2 (1 unit has 2 buttons, need 4 in total)	$15.98
Total Cost of Entire Project			$222.31

8 Conclusions

https://www.mouser.com/ProductDetail/926-LM2577S-ADJ-NOPB

https://www.mouser.com/ProductDetail/356-ESP32C3WROOM02H4

https://www.amazon.com/Prepare-10000mAh-Portable-Ultra-Compact-External/dp/B09L4T9YBX/ref=sr_1_6?crid=11VWIZ267JWAB&keywords=5+volt+battery&qid=1702232231&sprefix=5+volt+%2Caps%2C165&sr=8-6

https://www.amazon.com/Momentary-pre-Wiring-Waterproof-Stainless-Normally/dp/B09BKWMNJ9/ref=asc_df_B09BKWMNJ9/?tag=hyprod-20&linkCode=df0&hvadid=533377416454&hvpos=&hvnetw=g&hvrand=8417526223970594339&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9016280&hvtargid=pla-1457503465824&psc=1&mcid=f13d2a25b39a39a9bfa976d94684eb60&gclid=CjwKCAiAmZGrBhAnEiwAo9qHic45gnobjbMYiTGm9_kir2L3CcGjVdvMGvcJrnhNdiJPI_dgu151fxoC6CwQAvD_BwE

https://www.adafruit.com/product/2860

https://www.amazon.com/Gebildet-Waterproof-Button-Momentary-Stainless/dp/B0811QKG1R/ref=sr_1_1_sspa?keywords=Gebildet&qid=1701136073&sr=8-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&th=1