Boxing Workout Station: “Boxing Buddy”

Sabrina Mead, Nicholas Ebert, Andy Gonzalez, Catherine Sailor

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The Boxing Buddy is an original high-level engineering project that utilizes sensors and app development to create an at home workout station that helps the individual create more efficient training methods. This project senses and records both force data and reaction time for each punch thrown by the user, as well as directing specific workouts for the user to partake in through an app that works in sync with the structure and circuit. Within the app the user can choose between three different types of workouts to train. Data is sent almost instantaneously between the printed circuit board powering the project and the app during the workout process. Individual users are then able to receive live feedback based on their performance when utilizing this product.**

***Index Terms* — Ceramics, coaxial resonators, delay filters, delay-lines, power amplifiers.**

I. Introduction

Exercise is an integral part of life, and finding quick ways to get a good workout in the most efficient manner possible is something that many athletes wish to improve on. There are many forms of exercise to choose from including cardio, weights and other forms of sports. One sport in particular that has many health benefits is boxing. Boxing is traditionally associated with sparring, however, is also offers many overall health benefits as well for its users. Firstly boxing helps to improve cardiovascular fitness because of its high intensity short bursts of movement which improves metabolic function of the cells in your body. The act of throwing punches, jabs, and dodges also increases overall strength for the user which is the more well known result of boxing. However, boxing is also known to result in an improvement in hand and foot coordination and spatial awareness due to the full body movements that are required which stems from having to bring both your brain and body into harmony in order to execute the different movement sequences. The Boxing Buddy was conceived and developed to bridge the gap between this type of exercise and makes the sport more accessible to people, by making it a more affordable option for everyday users.

The Boxing Buddy uses modern technologies to complete a high-level electrical and mechanical engineering design project. The project is implemented on a two layer FR-4 copper planed originally designed printed circuit board with components that create power regulating circuits, USB to UART capabilities, force and reaction time sensing, and an ESP32 chip with Bluetooth and Wi-Fi capabilities. The circuit is built around a mechanical structure design that is able to withstand the force from being repeatedly punched and have enough vibrational capabilities to be able to oscillate when punched to get an accurate force reading. The structure utilizes springs in order to allow accurate force sensor readings. The sensor readings are then communicated over Wifi signals to our own mobile application. The mobile application is then able to capture the data sent from the sensors and display it to the user in an easy to understand way which allows them to monitor their progress in the sport. The resulting output of all of these components working together is a fully functioning training station with live feedback being output directly to the user.

II. Engineering Requirements

After knowing the overall goal and motivation of our project, in the initial stages of the design process for the Boxing Buddy the group set out to meet a number of engineering requirements that would help complete our goal of creating a usable boxing workout station. In the table below the main specifications we set for ourself are set.

Description	**Requirement**
The system should measure the reaction time accurately.	>= 80%
The system should measure the force accurately.	>= 80%
The structure of the Boxing Buddy should be adjustable and reach a tall maximum height.	>= 6' max height and >= 1' of adjustable height

The spring should allow enough movement for the accelerometer to get a reading.	Spring constant <= 30lbs/in
Startup time for application and device will be minimal.	<= 10 seconds
The mobile application should receive and display input data from the esp32 in a short time.	<= 5 seconds
Each strike pad should have LED indicators that flash when prompted by the app with minimal delay.	<= 5 seconds

Fig. 1. Specifications chart

The group has met each of the requirements highlighted with the Boxing Buddy design, which will be demonstrated later in this paper.

III. System Concept

The overall system of the Boxing buddy is based on an interaction between a mechanical structure, electronics, and mobile application. The design of the complete system is summarized to be understood in a flowchart below.

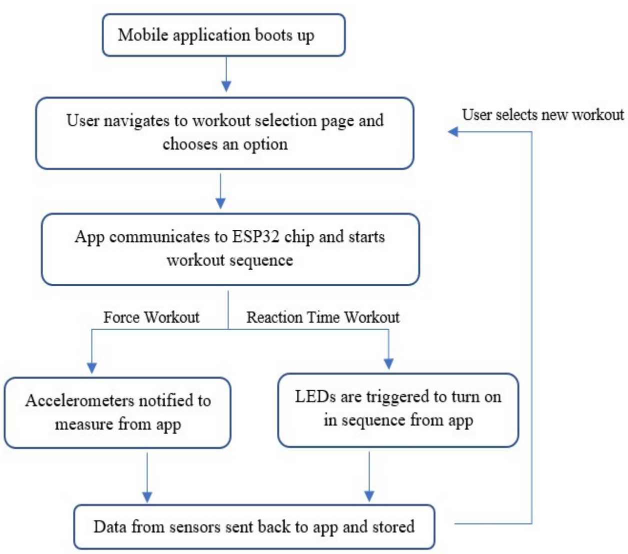


Fig. 2. Complete system flowchart that shows the transitional flow of a user and the different states of the device in sequential order.

As shown in the flowchart, the system works surrounding the mobile application and the communication between the mobile application and the microcontroller on board the PCB. The user goes into the mobile application and by navigating to the different pages they can decide which type of workout they would like to utilize at the time. The application then communicates that information to the circuit board on the structure which notifies either the force sensing to begin to take place or the reaction time sensing to begin. Once the workout is completed in its entirety the data from the previous workouts are then stored inside another portion of the app known as the workout history section.

The system is not entirely able to function autonomously as it does require live feedback and input from the user, making it a good tool to be used in conjunction with other training areas.

A. System Hardware Concept

Now that the system flow has been described, the major hardware interfacing this concept can be split up into 2 parts: a mechanical hardware concept and an electrical hardware concept.

On the mechanical side, the main idea is to create a small and easy to move structure that has the durability to be punched without falling over, while also being small enough to be usable in an at home working environment. As well as strike pads that are supported by spring mechanisms to allow for movement.

The electrical hardware concept is the integration of interfacing sensors that can receive input from the outside world and powering those sensors with varying voltage levels, with an MCU acting as the communication between the two.

IV. System Components

The system is created out of different components and individual parts that were headed by different members of our team that were then interfaced in order to create the final product. This section provides a semi-technical introduction to each of these components.

A. Structure

The Structure of the boxing buddy is created out of parts consisting of various materials and was built to be able to withstand punch forces and still be portable and adjustable.

C. Accelerometers

The accelerometer is an electronic device in our boxing buddy used to read the acceleration which will then be used to calculate the force of the user's punch.

D. LED Indicator Lights

The LEDs are used in the boxing buddy to implement visual representation for the user to strike the right or left pad. The elapsed time between when the LED was on and the pad was struck will result in the reaction time in seconds.

E. Mobile Application

For the Boxing Buddy, the mobile application is built using React Native and Expo Go. The goals of the mobile application were such that the app should be accessible on both Apple and Android devices as well as be easy to view and use for the user.

F. Communication between application and MCU

The communication between the mobile application and the MCU happens via the WebSocket API. This allows for smooth communication between the server and client. JavaScript is used as the main language.

G. Microcontroller

The microcontroller for this projects objective was to be able to communicate wirelessly with our mobile application. Due to this requirement the ESP32-WROOM-32E microcontroller was chosen.

H. PCB

The main functionalities that are needed on the PCB are to be able to regulate DC to DC voltages, Convert Analog to Digital Input, USB to Serial Capabilities, Wifi communication, Sensor/LED Connectors, and intake Power.

V. Mechanical Structure

This section will cover the mechanical structure in

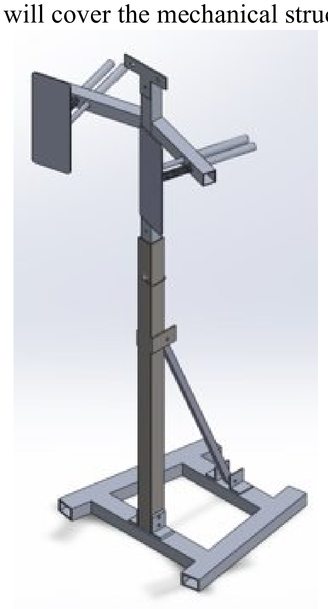


Fig. 3. Complete system mechanical structure

detail and all of the parts that work together to create the boxing buddy. Although the primary aspect of this device that separates it from most existing boxing training products available to consumers are the components that will provide the user with feedback, the mechanical structure that withstands the users' strikes is incredibly important as well. The following section discusses the elements of the overall structure of the product including the electrical housing components.

A. Strike Pads

Considering the time, we had to complete this project as well as what would be best for users, rather than fully designing the strike pads ourselves we thought it would be best to invest in strike pads specifically designed for boxing that are already on the market. The strike pads needed to provide a surface large enough for most users to comfortably hit, with firm enough padding and covering so the user does not get injured while striking the pads. In order to avoid having to alter the pads for our use, we sought out pads with straps to allow for easy, yet secure, attachment and removal. The pads that we ultimately chose were the Seisso Kick Pads. The Seisso pads were offered at a decently low cost compared to other strike pads and had good reviews. They also met our needs with a large surface area and straps on the back.

B. Support Tubing

The Boxing Buddy needed tubing to make up the structure that would be able to prevent the Boxing Buddy from tipping over when struck and that would not be deformed or broken off when the strike pads are struck by the user. We found that steel generally has both a higher tensile yield strength and modulus of elasticity than aluminum. This means that the steel will be more resistant to deformation and damage under the stress acting upon it. Due to financial and time constraints, we ended up using both aluminum and steel tubing that we were able to get for no cost. The use of the aluminum made the structure lighter, causing it to move when struck. We were able to address this issue with the use of sandbags to weigh the structure down, and the aluminum was still sturdy enough to withstand the force of the punches. Any of the parts that had to be welded had to be made of the same material, so we adjusted the design of the structure based on the amount of material we had. The inability to weld the different materials did not cause any issues because we also wanted the structure to disassemble into multiple parts to allow for easier transportation, so we just used the

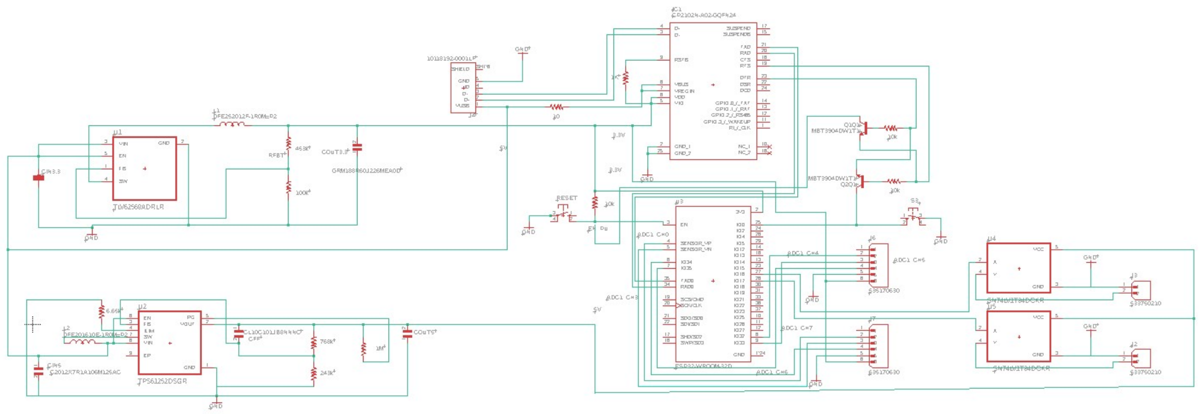


Fig. 4. Complete Final Schematic for PCB

different materials to define where the bolted connections would be made in the assembly.

C. Adjustable Height

For the Boxing Buddy to be appealing for a wider audience, we wanted to make the height adjustable. We spoke to an experienced welder to understand some of the possible materials that could be used for the project, and he suggested telescoping tubes. Telescoping tubes are two hollow tubes used together allowing for adjustable length. One of the tubes has a slightly smaller cross-section than the other allowing it to fit inside the larger tube so it can slide inside it to the desired length. Two of the free materials we were able to obtain were a steel tube with a 2.5 x 2.5 cross section and an aluminum tube with 2 x 2 in cross section. We were able to use these tubes as the telescoping tubing.

For the rails to remain at a height after they have slid out, we decided that the large rail would have a hole in it and the smaller rail would have multiple holes in it of the same size. To secure the tubes at each height, we chose to use a wire-lock pin. The holes in the tubes were made based on the diameter of the pin we chose. The pin we chose fits tightly around the tubing to securely fasten in place, but it also has a tab to allow for easy unlatching. The pin is made of steel so it can support the weight of the smaller telescoping tube and the vibration from each strike without breaking.

D. Springs

The springs were selected based on the spring constant. They needed to be sturdy enough to not break, but also needed a low enough spring constant to allow for movement for the accelerometer to get a reading. The dimensions of the springs also had to fit rods going through them that attach the strike pads to the rest of the structure. We were able to find 3 inch long springs with a diameter large enough to fit the rods in the center, and a spring constant of about 22 lbs/in. These springs allow for the accelerometers to move enough to get a reading and reduce the movement of other components. The closed and ground ends also allow the springs to sit flat up against the parts they are between.

In order to hold the springs in place and allow for the strike pads to move when struck, the springs were placed between the strike pad holders and the arms of the smaller telescoping tube. The rods going through the center of the springs were fastened to the strike pad holders on one end and fastened to washers on the other end on the back side of the arms of the telescoping tube. The washers are larger than the holes the rods go through, so it helps keep the pad holder attached when it is struck and the spring in place while still allowing for movement as it is not fastened to the arm of the smaller telescoping tubing, it just rests against it until the pads are struck and returns to the original position when the springs return to their original position. A second rod is attached to each of the pad holders that go through other holes in the arm of the telescoping tube next to the holes the rods that run through the springs go through. These second rods prevent the pad holders from rotating when struck.

VI. PCB

The PCB for this project has a couple of main parts that will be described in detail in this section. The final schematic that brings all the described parts together is shown in figure 4.

A. Microcontroller

The heart of the project is an ESP32-WROOM-32E-N16 microcontroller. The ESP32 family is chosen for its Bluetooth and Wifi connectivity as well as its high level of integration capabilities. The ESP chip has built in antenna switches and analog to digital converters which helped to simplify the PCB design needed for the project since we could utilize these instead of designing our own. The ESP32 can also perform completely as a standalone system which reduced the need for our design to have both a microcontroller chip and a Wifi chip into just needing one singular device for both purposes. The ESP32 can interface with other systems easily to provide Wi-fi functionality through its SPI/SDIO and I2C/UART interfaces which highly contributed to it being chosen in the final design. The ESP32 chip receives the data from the sensors directly through wires and then transmits that data over Wi-fi to the mobile application.

B. Power Intake

Instead of having a battery pack or some other form of charging device, due to the nature of the system being pretty stationary our team opted to have a wall plug-in as our power source. We chose to use a wall plug that had a built in Analog to Digital conversion so that we would not have to create our own ADC on board the schematic. Using this method and a 6ft USB to USB micro-B chord we routed 5 V of unregulated DC power from the wall onto the board.

C. Voltage Regulation

The Boxing Buddy requires two different voltage levels on board in order to function properly. The board initially receives 5 V from the DC wall plug into a micro-b USB connector. From this micro-b connector, the 5V is then routed through to two different voltage regulators in the circuit.

First the 5V is brought down to 3.3 V using a TLV62568ADRLR regulator which has an operating voltage of 2.5V to 5.5 V and an output current of up to 2 A. The inductor that we used in this circuit, the DFE252012F-1R0M=P2, also played an important part. During the design process for our final board, we made.
sure to calculate all saturation current possibilities for regulator and then chose an inductor that was rated 20 to 30% higher than those calculations to ensure that it would not reach saturation current accidentally. The 3.3 V regulator is the most crucial power source on the board in order to allow everything to function as intended which is why such care was taken into choosing the inductor.

The second regulator brings the 5V to a regulated 5V using the TPS62152RGTR voltage regulator that has an operating voltage of 3 to 16 V and an output current of up to 1 A, which falls within the requirements needed for this component. For the 5V regulator similar care was taken to choose the inductor for the circuit which was DFE252012F-1R0M=P2.

D. USB to UART Connection

On board in order to upload code from the computer onto the ESP32 chip we used the CP2102N-A02-GQFN24 chip which is an IC USB to UART bridge chip. The chip has an operating voltage between 3V and 3.6V and has a normal operation power consumption of 9.5mA to 13.7 mA. Using the transmitter and receiver connection points between the CP2102N-A02-GQFN24 and the ESP32 chip the serial connection is able to be made and data transmitted correctly.

E. Accelerometer Connectors

The accelerometer connectors are Molex 6 pin connectors, with one connection for the X, Y, Z, ST, GND, and 3.3V pins. The accelerometers receive all data in analog form so in order to make sense of the data we are inputting; the board utilized an ADC1 Channel on pins 32 and 34 of the ESP32 chip in order to convert the numbers into a usable form.

E. LED Indicator Connectors

The LED connectors consist of 2 pin Molex connectors. The LED’s required connections to both the 3.3V power source coming from the pins within the ESP32 chip, as well as the 5V regulated power coming out of the 5V regulator because of the operating voltage of the LED’s being used. In order to allow the digital signal to be received and the right voltage to be received the SN74LV1T34DBVR voltage shifter was used which is a Buffer Non-Inverting 1 Element Push-Pull Output SOT-23-5 with an operating voltage range of 1.65 V to 5.5 V. This allows one pin of the LED to receive the digital signal from the code, and then one pin to receive 5V which allows it to turn on.

VII. Electrical Hardware

The main pieces of electrical hardware that have not been covered in technical detail yet are the accelerometers for force sensing, and the LED indicator Lights to calculate reaction time.

A. Accelerometers

The accelerometer is a small 3-axis electronic sensor with signal conditioned voltage outputs and 4 mounting holes. The ADXL377 can withstand -200 to +200 g's which measures acceleration from motion, shock, or vibration. The accelerometer has 6 pins that are GND, 3.3V, X, Y, Z, and ST. Each of these pins has a role. Where GND is for the connection to ground, 3.3V is the power that the accelerometer is receiving, X, Y, and Z are the outputs of the accelerometer, and the ST pin is for self-testing. The force sensing capabilities of the system are calculated using the basic ideas of newtons second law of motion; force is equal to mass times acceleration. By utilizing an accelerometer connected to the force strike pad and measuring how much the strike pad moves, the program is then able to calculate the force by taking the weight of that strike pad and multiplying it by the output of the accelerometer. The accelerometer chosen for this purpose is the ADXL377. The ADXL377 is powered by 3.3V and has an X, Y, and Z axis plane, however for the purposes of this project we only need the reading from the Z axis. The output of the sensors is analog and must be converted to digital within the system.

B. LED Indicator Lights

Reaction Time Sensing is measured using a code interface between the microcontroller and the mobile application. Once the reaction time workout is chosen on the mobile application the program utilizes the LED hardware to signal to the user to strike the pad. The code begins by randomly sending a signal at a random time to turn on one of the two LED indicator lights. Once the signal is sent the time elapsed between the strike pad being hit and the LED turning on is recorded. The program then keeps a database of these times for the workout so the user can see how their reaction time changes and alters throughout the workout. The LEDs themselves are blue indicator lights that require 5V and 16 mA to turn the LED on.

VIII. Mobile Application Detail

The Boxing Buddy team chose to make a mobile application. Based on other similar products there were several options for a user interface. One was a screen attached to the structure itself that displays the necessary information and operates the workouts, the other is a website, and the last option is a mobile application. Due to the boxing nature of the project, we thought it best not to have a screen display attached to the structure so that no one hits the screen by accident. The website is not ideal either because it is unrealistic for a user to have to open their computer and go to a website in order to see their data and operate the Boxing Buddy. This made the mobile application the logical choice. The mobile application allows the user to see the data from their workouts expressly on their mobile device and allows the user to operate the Boxing Buddy. This choice prevents the user from accidentally damaging the equipment and reduces the hassle of trying to operate the Boxing Buddy. This decision is summarized in the Table below.

A. Mobile Application Type

While determining which type of mobile application we wanted to create, we kept our essential goals for the app in mind. These goals are two-fold, user-friendliness, accessibility. Accessibility refers to the app being available on apple or Android devices. The first type of app listed is the Native app. This is specific to either android or an apple device. The native app offers the most customization, user friendliness, and overall functionality, but it violates our goal of accessibility. Hybrid apps are available on both apple and android devices and are customizable. They tend to be slower than native apps and are a bit worse in terms of user friendliness. "Drag and drop" apps use app builder tools and are nice for quick development but they lack the ability to perform some crucial tasks that our app needs to be able to perform. Web applications are also quicker to develop and have more functionality than drag and drop apps, but they tend to have poor user friendliness.

We chose to create a hybrid mobile application. There are lots of libraries established for IoT communication, which is essential to our project and most importantly, it's accessible to both apple and android devices. The table below summarizes this decision.

Type of Application	**Main Advantage**	**Main Disadvantage**
Native	More user friendly and gives increased access to OS and platform specific tools on a device.	Lower accessibility
Hybrid	Much more accessible. Android and Apple users both can use it.	Slower and more limited access to OS and platform specific tools on a device.
"Drag and Drop"	Speed of development and easy to learn.	Poor customization and limited uses. Not very complex.
Web Application	Quicker to develop than Native and Hybrid and very adaptable to changes.	Limited functionality and nonnative feel, that is, poor user friendliness.

Fig. 5. Type of Application chart

B. Mobile Application Front-End

When choosing which framework to build the app from, it was important to keep the essential goals for the app in mind. These goals are as follows: user friendliness, free to program, able to be run on iOS or Android, and not too complicated to program. We chose to react native because it is a well-documented, free-to-use mobile app framework. React Native is often used to make hybrid applications. There are many libraries that allow the developer to make a user-friendly native-looking application. The table below summarizes why react native was used over other frameworks. Other frameworks are either meant for native development or they use languages that are not ideal for our project, see "Mobile Application Back-End."

Also, Expo Go is used to run the application. Expo Go allows the mobile application to be run on a mobile device without the app needing to be published on the Google Play Store or Apple Appstore. The combination of React Native as a development tool and Expo Go as a quick and efficient debug and app launching tool allowed development to be smooth and debugging to be efficient. Using Expo Go also allowed launching the application to be quick such that it meets the specification of launching the app in under 10 seconds.

Framework	**Languages**	**Main use and Benefits**
Ionic	JavaScript, CSS, HTML	Primarily used for web-based applications that exist both on mobile devices and on the internet.
React Native	JavaScript	Hybrid mobile application development with great user friendliness from native tools.
Flutter	Dart	Hybrid mobile application with higher degrees of control and precision.
Xamarin	C#, .NET, and Swift	Hybrid mobile application development.

Fig. 6. Framework Chart

C. Mobile Application Back-End

The back end of the mobile application uses JavaScript as its main language. Having JavaScript as the main language in which react native is built on is an important feature to keep in mind because JavaScript is the main Internet of Things language, and our main communication happens via the WebSocket API. Originally, we sought to use a database to store the user's workout data and information, so they could keep this data on the app. Due to time constraints we decided against this course. Instead, information is stored on the app directly.

There were several back-end APIs that were considered and attempted for the Boxing Buddy. The first is the HTTP standard communication and the other is the WebSocket communication standard. HTTP communication was used first, but this proved to be inefficient. Each time the ESP32 and the mobile application communicated, they needed to respond with a webpage. The WebSocket API allows the two ends to communicate seamlessly and much quicker with very fast updates. This allows us to meet our communication specification of having data display on the application in less than 5 seconds. Communication happens much quicker at a approximately 0.1 seconds.

X. Workout Design Detail

The workouts were designed partially using the ESP32 code to control GPIOs and on the app where things are displayed.

A. Workout Logic - Filtering

The ESP32 is connected to two accelerometers, one for each strike pad. Each accelerometer's "z" pin is connected to the analog to digital converter of the esp32. Because the accelerometers need to withstand a high acceleration, we chose a +/- 200 g accelerometer. This means that noise has a greater affect when trying to read small changes in acceleration. Our project relies on being able to quickly sense a change in acceleration to register a punch. This led to implementing an Exponentially Weighted Moving Average Filter. This filter works by making a weighted average of the previously captured data and the raw data coming in. We found that it was best to give higher weight to the previous data. Having a filter allowed the readings from the accelerometer to be steady around 0. This enabled us to implement a threshold where the app registers a punch only when the weighted value exceeds a certain amount. Then, that value goes through an unfiltering function and is converted into pounds of force and displayed on the app.

B. Workout Logic – Force Sense Workout

The first workout is the "Force Sense Workout". In this workout, the user uses the app to select a strike pad to use. After selecting the pad, the corresponding LED turns on and the user can hit the strike pad. This causes the filtered data from the accelerometer to pass the threshold which registers as a hit. Then, the force of the hit, in pounds, is displayed to the user on the app. The user can continue to strike the same pad or select the other pad and strike that one.

C. Workout Logic – Reaction Time Workout

The second workout available is the reaction time workout. In this workout, the user hits the start button on the app and after a random delay no more than 8 seconds, one of the indicator lights will turn on at random prompting the user to strike the corresponding pad. The time, in seconds, between when indicator light turned on and when the user hit the strike pad is displayed on the app.

D. Workout History

The workout history page allows the user to log their workout data. This is user input only. The specification was to allow the user to log at least 5 workouts. The current mobile application allows for 20 slots of the user logging their workout data.

X. Conclusion

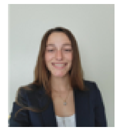
In conclusion our group was able to successfully create a boxing workout station with the capability to interface with a mobile application and accurately read the users force and reaction times. Throughout this process our group learned many technical skills that will be able to be brought into our careers as we enter the industry as well as giving us experience in problem solving, pivoting when errors occur, the engineering design process, how to work in a group, and how to project manage.

Acknowledgement

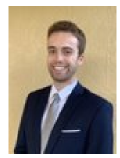
The authors wish to acknowledge the assistance and support of the of the Senior Design Instructors Lei Wei, and Dr. Richie. We would also like to give a big thanks to James (Jim) Essad for providing support on the mechanical structure and build, Dr Kim, and Dr. Yuksel for giving aid in the electrical design of the system.

The authors would also like to acknowledge and thank the reviewers of this project Dr. Sun, Dr QZ Sun, and Dr. Kauffman.

Biography



Sabrina Mead is an Electrical Engineering student at the University of Central Florida. Following graduation, she will be working at General Electric in a rotational program.



Nicholas Ebert is an Electrical Engineering student at the University of Central Florida. Following graduation, he will be seeking a job in the power systems industry.

Andy Gonzalez is an Electrical Engineering student at the University of Central Florida. After graduating he plans on figuring out what discipline he will follow in the Electrical Engineering field.

Catherine Sailor is a Mechanical Engineering student at the University of Central Florida. Following graduation, she will be working for Orlando Utilities Commission as a Project Engineer at Stanton Energy Center.

References

[1] Espressif. (n.d.). ESP32WROOM32E ESP32WROOM32UE - Espressif. Retrieved April 18, 2023, from https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

[2] #267551, M., #1604081, M., #1595581, M., #1611653, M., #1667778, M., #559059, M., #816211, M., #142865, M., Onestone, & #1510525, M. (n.d.). Sparkfun Thing Plus - Esp32 wroom (micro-B). WRL-15663 - SparkFun Electronics. Retrieved April 18, 2023, from https://www.sparkfun.com/products/15663.

[3] Application - s3-us-west-2.amazonaws.com. (n.d.). Retrieved April 18, 2023, from https://s3-us-west-2.amazonaws.com/catsy.557/Dialight_PMI_data_557_Apr2020.pdf

[4] React native · learn once, write anywhere. React Native RSS. (n.d.). Retrieved April 18, 2023, from https://reactnative.dev/

[5] #641951, M., & #767148, M. (n.d.). Sparkfun Triple Axis Accelerometer Breakout - ADXL377. SEN-12803 - SparkFun Electronics. Retrieved April 18, 2023, from https://www.sparkfun.com/products/12803