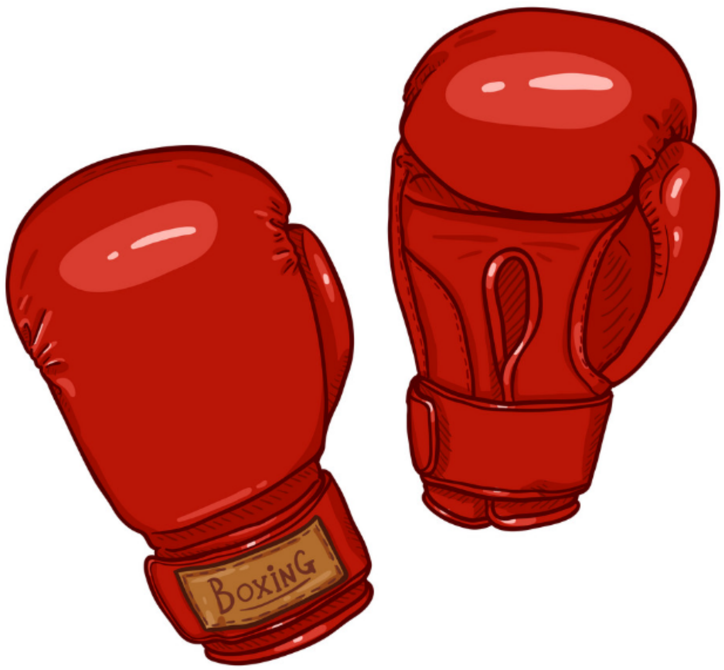
Boxing Buddy



EEL 4914: Senior Design I
University of Central Florida
Department of Electrical Engineering and Computer Science

Group 15

Nicholas Ebert	Electrical Engineering	nicholas.ebert@knights.ucf.edu
Andy Gonzalez	Electrical Engineering	andy.g@knights.ucf.edu
Sabrina Mead	Electrical Engineering	sabrinamead1@knights.ucf.edu
Catherine Sailor	Mechanical Engineering	casailor@knights.ucf.edu

Table of Contents

I. List of Figures

II. List of Tables

1.0 Executive Summary

2.0 Project Description
- 2.1 Project Motivation
- 2.2 Project Goals
- 2.3 Project Objectives/Description
  - 2.3.1 General Structure
  - 2.3.2 Mobile Application
  - 2.3.3 Training Modes
- 2.4 Requirement Specifications
- 2.5 Quality of House Analysis

3.0 Project Hardware and Software Initial Design
- 3.1 Initial Design Architectures and Related Diagrams

4.0 Related Standards and Realistic Design Constraints
- 4.1 Force Sensor Constraints
  - 4.1.1 Constraint of Max Amount of Measurable Force
  - 4.1.2 Dimension Constraint
  - 4.1.3 Cost Constraint
  - 4.1.4 Accuracy Constraint
- 4.2 Software Constraints and Standards
  - 4.2.1 Software Constraints
  - 4.2.2 Software Standards
- 4.3 Design Constraints
  - 4.3.1 Economic and Time Constraints
  - 4.3.2 Manufacturability and Sustainability Constraints
- 4.4 PCB Standards and Constraints
- 4.5 WiFi standards
- 4.6 Power Constraints and Standards
  - 4.6.1 Power Constraints
  - 4.6.2 Power Standards

5.0 Research and Part Selection
- 5.1 Existing Similar Products
- 5.2 Force Sensors

5.2.1 Piezoelectric Sensor
5.2.2 Button Load Cell
5.2.3 S-Beam Load Cell
5.2.4 Through Hole Load Cell
5.2.5 Pancake Load Cell
5.2.6 Miniature Load Cell
5.2.7 Single Point Load Cell
5.2.8 Capacitive Load Cell
5.2.9 NTEP/Waterproof
5.2.10 Accelerometer
5.2.11 Conclusion
5.3 Reaction Time
5.4 Sensor to App Communication
5.5 Microcontroller Chips
5.6 PCB
5.6.1 DC-DC Voltage Regulator
5.7 LEDS
5.8 Expectations of the Mobile Application
5.9 Types of Mobile Applications
5.10 Programming Languages
5.10.1 The C Language
5.10.2 C++
5.10.3 C#
5.10.4 Java
5.10.5 Python
5.10.6 Javascript
5.10.7 Swift
5.10.8 Programming Languages Summary
5.11 Framework Options
5.11.1 Ionic
5.11.2 React Native
5.11.3 Xamarin
5.11.4 Flutter
5.11.5 Aurelia
5.11.6 Angular
5.11.7 Summary of Frameworks
5.12 The Back-end
5.12.1 Amazon Web Services

5.12.2 Azure
5.12.3 Google Cloud Platform
5.12.3 Back-end API Summary
5.13 Power Routing
5.14 Parts Selection Summary

6.0 Mechanical Design
6.1 Strike Pads
6.2 Support Tubing
6.3 Adjustable Height
6.4 Cable Housing

7.0 Project Prototype Testing Plan
7.1 Mechanical Design Testing
7.2 MCU Testing
7.3 WiFi Testing
7.4 Accelerometer Testing

8.0 PCB Design
8.1 Components Included
8.2 DC-DC Voltage Regulator Design:
8.3 PCB Schematic
8.4 PCB Reflow Oven Process
8.4 PCB Manufacturers

9.0 Mobile Application User Interface Design

10.0 Administrative Content
10.1 Milestone Discussion
10.2 Finance Discussion

11.0 Conclusion and Overview

12.0 Appendix
12.1 Bibliography

I. List of Figures

Figure 2.1: House of Quality
Figure 3.1: Block Diagram Legend
Figure 3.2: Hardware Block Diagram
Figure 3.4: Initial Device Structure Design
Figure 4.1: IPC Standard Procedure
Figure 4.2: Hierarchy of IPC Design Specifications
Figure 4.3: Example of Trace Spacing
Figure 4.4: Differential Pair Constraints
Figure 4.5: Standard American 15 Amp Wall Receptacle and NEMA 1-15 (left) and NEMA 5-15 (right) Standard Plugs
Figure 5.1: Nexersys Interactive Boxing Trainer Specifications
Figure 5.2: Piezoelectric Sensor
Figure 5.3: Button Load Cell
Figure 5.4 S-Beam Load Cell
Figure 5.5: Through Hole Load Cell
Figure 5.6: Pancake Load Cell
Figure 5.7: Miniature Load Cell
Figure 5.8: Single Point Load Cell
Figure 5.9: Capacitive Load Cell
Figure 5.10: Advantages of Bluetooth Low Energy
Figure 5.11: Bluetooth Low Energy Chips
Figure 5.12: Bluetooth Module
Figure 5.13: Images of MIcrocontroller chips
Figure 5.14: MSP430FR6989 data sheet
Figure 5.15: Mega 2560 MCU development board
Figure 5.16: Chosen Switching Regulator
Figure 5.17: DIP LED Lights
Figure 5.18: Surface Mounted Diode LED Bulb
Figure 5.19: Chip on Board LED Spotlight Bulb
Figure 5.20: Chosen LED's
Figure 5.21: Amazon Web Services Amplify Diagram
Figure 5.22: Azure Structure Diagram
Figure 5.23: Diagram of Google Cloud Services
Figure 6.1: Overmont Taekwondo Kick and Boxing Karate Pad
Figure 6.2: RDX T15 Noir Focus Pads
Figure 6.3: Tubing with a circular cross section (left) and tubing with a rectangular cross section (right)
Figure 6.4: Telescoping Weld-Together Rails
Figure 6.5: This shows the box to protect electrical equipment that can not go within the tubing.
Figure 7.1: Two views of the Boxing Buddy design in Solidworks with numbers corresponding to parts
Figure 7.2: The Boxing Buddy design in Ansys with a mesh
Figure 7.3: The Boxing Buddy axially oriented

Figure 7.4: WiFi Testing Terminal Window
Figure 7.5: Set-up for testing
Figure 7.6: Accelerometer output from testing
Figure 8.1: DC-DC Voltage Regulation Schematic
Figure 8.2: DC-DC Regulation Summary of Circuit Specifications
Figure 8.3: PCB Schematic (Foundation)
Figure 8.4: Typical solder reflow thermal profile
Figure 9.1: Boxing Buddy Homepage
Figure 9.2: Previous Workouts Page
Figure 9.3: Workout Specific Application Pages

II. List of Tables

Table 2.1 The Boxing Buddy Goals
Table 2.2 System Requirements
Table 4.1 PCB Standards
Table 4.2 PCB Constraints
Table 5.1 Conclusions of Sensor (Cheaper)
Table 5.2 Conclusions of Sensor (Expensive)
Table 5.3: Bluetooth Low Energy Price Comparison
Table 5.4: Comparison of Bluetooth Low Energy and Bluetooth Classic
Table 5.5: WiFi Standards needed for Selecting an Option
Table 5.6: WiFi Chip Price Comparison
Table 5.7: Summary of Top Microcontrollers
Table 5.8: Summary of Texas Instruments Development Board
Table 5.9: Technical Specifications of Mega 2560 microcontroller
Table 5.10: Board Component Summary
Table 5.11: Advantages and Disadvantages of Linear Regulators
Table 5.12: Summary of Buck Switching Regulator IC Specifications
Table 5.13: Comparison of Surface Mounted Diode LEDs
Table 5.14: Summary of Selected LEDs
Table 5.15: Advantages of Selected LEDs
Table 5.16: Advantages and Disadvantages to Different Application Types
Table 5.17 Summary of Front-End Language Advantages and Disadvantages
Table 5.18: Summary table for Frameworks
Table 5.19 Summary of Back-end APIs
Table 5.20: Parts Selection Summary
Table 8.1: PCB Components summary
Table 8.2: PCB Manufacturing Summary
Table 10.1 Fall 2022 Schedule
Table 10.2: Fall 2022 Schedule Continued
Table 10.3 Fall 2022 Schedule Drafts
Table 10.4 Fall 2022 Schedule Drafts Continued
Table 10.5 Spring 2023 Schedule
Table 10.6 Total Financial Cost

1.0 Executive Summary

Exercise is an integral part of life, and finding quick ways to get a good workout in the most efficient manner possible is something 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 it 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 also is 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. Because of these reasons, boxing has many health benefits for its users which is what inspired Boxing Buddy to become a project.

Boxing Buddy hopes to bridge the gap between making the sport more accessible to people while also making it an affordable option for its users. Currently boxing memberships can range anywhere from $100 to $200 a month to train at a gym, Boxing Buddy will instead provide a private training simulation that a user can utilize in the privacy of their own home for only a one-time price that will save them money in the long-run. To do this we will utilize different sensors to track the force and reaction times of the user to track their growth and progression within the sport. As well as create a durable structure with strike pads and an interactive app where the user can choose from pre-programmed workouts. This document has been written to describe the process of the design of this product and the technology specifications required to create the electrical and mechanical systems required. Our study starts with looking at other relevant designs and products that are already being used. Following this initial examination, more research is done in order to choose the rest of the parts that will be used in the system. During this stage of the process design constraints and standards for each technology are discussed and compared for each decision in the design including decisions regarding the systems sensors, microcontroller, and software choices.

The Boxing Buddy allows the user to train against themselves so that as they get better, they can compare to their previous workout information and track their progress. This means people of varying skill levels will find great use of the Boxing Buddy. The Boxing Buddy is a great alternative to traditional boxing training methods that is an affordable option for those that want to increase their health and stamina through boxing in the comfort of their own home.

2.0 Project Description

This section serves to provide the motivation for developing a personal boxing training simulation as well as portray the goals and objectives of the project. The motivation behind creating an easy to use training simulation will be discussed to help the reader understand the why behind our project. Followed by Project objectives of what the project will accomplish and different requirement specifications that will be met.

2.1 Project Motivation

Typically, in order for people to obtain boxing training with feedback they have to go to a gym with a coach. Gym memberships and coaching can often be expensive, and require availability in the schedule of the trainee to align with availability in the coach's training schedule. Additionally, coaches can provide feedback and the coach or trainee could record that feedback themselves, but humans make mistakes, so there may not consistently be recorded feedback from training sessions with a human coach. Although there are already some existing devices that allow for at-home training without a coach that provide automatic feedback to a user interface, the options are limited, they are expensive, and they often have poorly placed or developed user interfaces. Our device will be more affordable and will send rapid feedback directly to an application on the users' mobile devices providing a lower risk of damage compared to a user interface mounted onto the design itself, greater convenience for users, and an overall better experience.

2.2 Project Goals

This section provides a general outline of the different goals that will be achieved throughout this project which are classified as basic goals. As well as giving an overview of advanced goals that should be able to be implemented during this project, and stretch goals that will only be added if time permits. These goals are listed in Table 2.1.

The basic goals encompass the core functionality of the Boxing Buddy project. The system will include two strike pads that can record force data and reaction time attached to an adjustable mechanical frame. A mobile application will be the method of interaction between the user and the Boxing Buddy. The advanced and stretch goals encompass elements of the Boxing Buddy project that can be added should time allow. The stretch goals are more difficult than the advanced goals to implement. These goals include adding handles that the user can grab onto which will record heart rate information of the user. Furthermore, an additional strike pad can be added and the strike pads themselves can be changed to be movable to make striking them more of a challenge.

Goal Type	Description
Basic	System will have 2 strike pads with force and reaction time sensors.
Basic	System will have an interactive app interface.
Basic	Each of the strike pads will have force sensing.
Basic	Reaction time of each punch will be recorded and uploaded to the app interface.
Basic	Structure will be adjustable for different heights.
Advanced	Heart rate monitors for the user to monitor during system use.
Advanced	Creating pre-programmed workouts synced with the app.
Advanced	Adding in an additional strike pad with sensors.
Stretch	A robotic punching arm is added to simulate getting punched.
Stretch	Movable strike pads.
Stretch	Adding in motion sensors for aid in preprogrammed workouts.

Table 2.1 The Boxing Buddy Goals

2.3 Project Objectives/Description

This section provides the objectives of our project and the description. Into three separate sections including general structure, mobile application, and training modes. Each of these sections will have a detailed description explaining the information about them.

2.3.1 General Structure

The device will have a minimum of three strike pads. The strike pads will be attached to a base with adjustable height so it can be suitable for people of various heights or if the user just wants to adjust the angle at which they are striking. Each of the strike pads will have LED lights attached to the top of them that will turn on to assist in measuring reaction time.

The strike pads for the device will be the primary source of data collection. The two strike pads are intended to mimic striking different portions of the body. If there is additional time to modify and test the project after completing the initial design we will add more strike pads and allow them to move to allow for more striking zones and angles. Each of the strike pads will be padded so the users are not injured when they hit them. Within each strike pad, beneath the padding, will be a device to measure the force the user strikes the pad with. The force reading will be sent to an application on the user's mobile device and stored there. In addition to the force being measured, the user's reaction time can be measured based on the time between when they hit the strike pad and the LED lights illuminating on top of the strike pads signaling for the user to strike. An additional feature that may be added if the initial design is completed is the ability for the device to measure and record the user's heart rate. The heart rate data would be sent to the application as well.

In order for the data to be recorded and sent to the user interface, a microcontroller will be connected to the sensors. The microcontroller will also be connected to the lights to turn them on and off and to record the users' reaction time. There will be components on the microcontroller with wireless abilities to allow for the data to be sent to the application.

2.3.2 Mobile Application

One of the aspects of our device that separates it from regular striking pads often used for boxing training, is a user interface that allows for users to view feedback on their performance. The user interface will be an application accessible on the user's handheld mobile device. The data from users' training sessions will be sent to the application and recorded on it for them to review.

In the application, users will be able to select a training mode among three different types of modes. The three different modes are the reaction time mode where the LEDs light up and reaction time as well as force is recorded. The second is just the force workout where only force is recorded. The third is the repetitive hits workout where there is a time limit and the user hits the strike pads as many times as possible. Viewing the force with which they struck the strike pads as well as their reaction time will depend on which training mode they select. The mobile application will also include a historical workouts page that enumerates the previous workouts recorded on the account logged in. If there is time to further develop the application once the initial design has been accomplished, an option to select programmed workouts from the application will be added.

2.3.3 Training Modes

In order to offer various types of feedback and focus for training, there will be two different training modes for the user to choose from on the application. The training modes are described below.

Strike Force
The strike force training mode allows for the users to view how hard they can hit the strike pads. During this training mode the users will receive feedback in the application from the force sensors showing them a numeric measurement of the force with which they struck the pad.

Reaction Time
The reaction time training mode will utilize the LED lights to help measure the user's reaction time. Once the user selects this training mode, at a time unknown to the user, the LED light on top of one of the strike pads will come on indicating that they should hit that strike pad. This will happen multiple times within the session. Each time the light on a strike pad comes on, the time will be measured between when the light came on and when the user struck the pad to measure the user's reaction time.

2.4 Requirement Specifications

The minimum technical requirements for our system are listed below in Table 2. The requirements highlighted in blue are those that will be demonstrated. The Boxing Buddy consists of several different types of requirements. The types of requirements are as follows: performance, software, and structure. In order to be able to achieve these requirements, it is paramount that these requirements are measurable with clear thresholds.

The first type of requirement is the performance requirements. These requirements are centered around efficiency and accuracy. In order for the Boxing Buddy to be an effective exercise tool, the measurement of the force reading as well as the reaction time must be accurate. This will allow the user to correctly value how strong and how fast they are respectfully. Additionally, using the Boxing Buddy must be quick and easy. This will make the product more enticing to the user especially since it will be controlled by the user's phone.

The second type of requirement is the software requirements and these refer to the mobile application. These involve making an efficient, easy to use application. The last type is the structure requirements which refer to the mechanical structure of the Boxing Buddy as well as the electrical sensors.

	Type	Description/ Requirement	Quantitative Requirement
1	Performance	The system should measure the amount of force and reaction time with a high accuracy	> 80%
2	Performance	Each strike pad should have LED indicators that flash when prompted by the mobile app with a minimal delay.	< 5 seconds
3	Performance	Startup time for application and device will be minimal.	< 15 seconds
4	Software	An interactive mobile app should be connected to the Boxing Buddy systems and will receive and display the input data in a short time.	< 5 seconds
5	Software	The mobile app should display previous workouts.	≥ 5 workouts
6	Software	The system should be able to operate autonomously for the 3 pre-programmed training session(s) that can be selected through the app	= 3 training sessions
7	Structure	The structure should have an adjustable height.	< 7 feet
8	Structure	The system will have 2 strike pads with all functionality.	= 2 strike pads
9	Structure	The strike pads should be able to withstand a high punching force.	> 500 lbs

Table 2.2 System Requirements

2.5 Quality of House Analysis

This HOQ consists of the market and engineering requirements and how they correlate. The HOQ is shown in Figure 2.1 below. Marketing requirements are on the left-hand-side. These requirements outline the components of the Boxing Buddy that would make a customer more likely to buy it. The top is the engineering requirements which reflect efficiency, accuracy, and pricing.

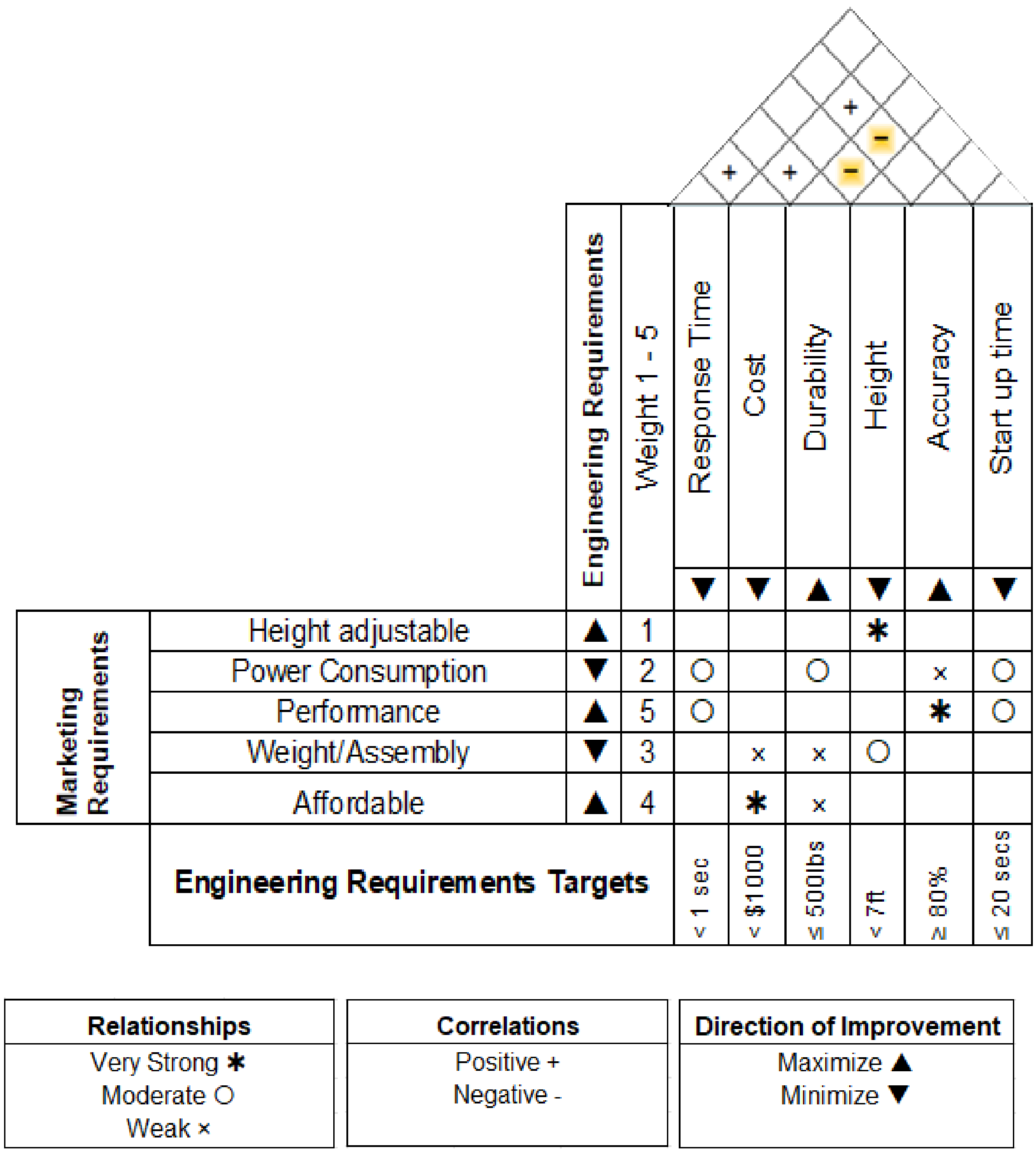


Figure 2.1: House of Quality

3.0 Project Hardware and Software Initial Design

This section of the document will consist of the project hardware. This will be shown in diagrams that will be easy to interpret. Hardware includes the force sensors, micro controller, LED lights, AC/DC convertor, etc.

Not only will it show the hardware but the software as well. In essence, the main controller feeds a wifi module that then sends the data to the package receiver and sends feedback.

3.1 Initial Design Architectures and Related Diagrams

At the beginning of the semester and the creation of the Boxing Buddy Robot design, one of the first tasks was to create an initial design flow diagram as a part of a document called the ‘Divide and Conquer’ document. Within this document our team layed out the known aspects of the design to begin bringing the project together and the rudimentary block diagrams for the initial project are shown below. They include the status of each element along with the member responsible for each element.

The hardware block diagram, in Figure 3.2 below, shows all the elements related to the electrical wiring and mechanical structure. This block diagram includes the metal structure that will hold the strike pads. Each of the strike pads will also have a force sensor each and an indicator light each that will be made from LEDs. The main controller will control the LEDs and gather force data as well as distribute power. Power from the wall is also shown in this diagram.

The software block diagram on the next page, shown in Figure 3.3, includes elements relating to the mobile application and WiFi module in the main controller that will be sending data back and forth.

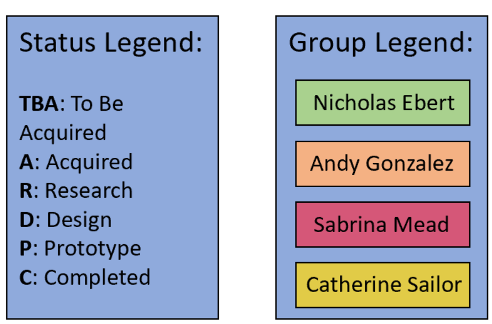


Figure 3.1: Block Diagram Legend

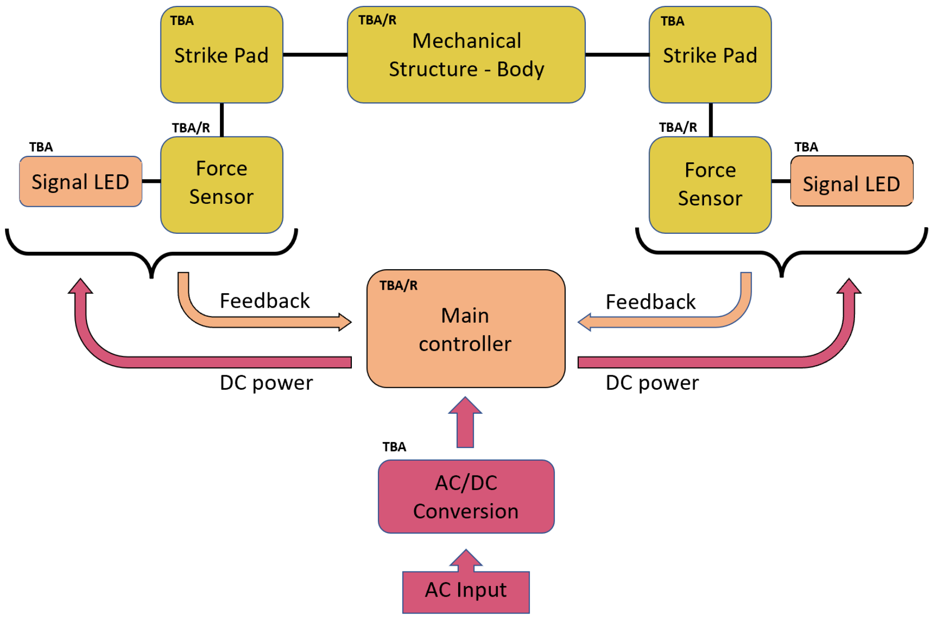


Figure 3.2: Hardware Block Diagram

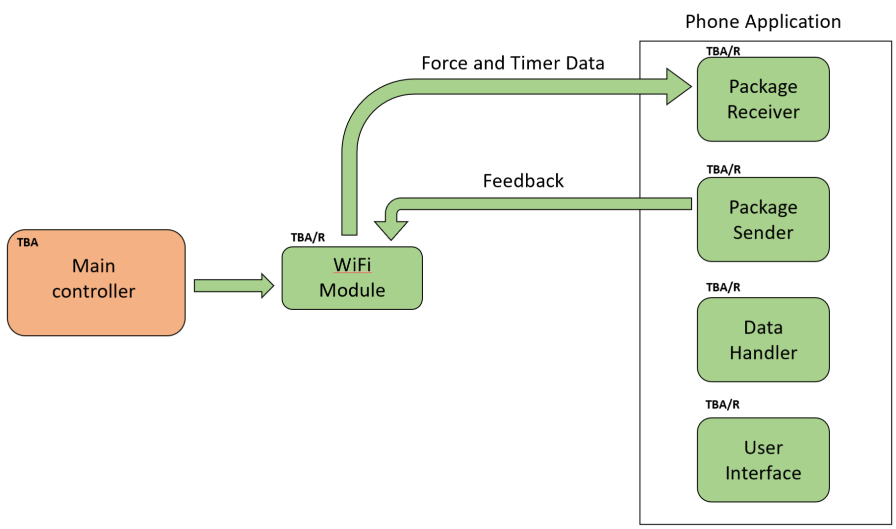


Figure 3.3: Software Block Diagram

A key starting point to the design process of the Boxing Buddy was choosing how many and what type of sensors would be utilized for the design of this project as well as the controller type being used and how to integrate both using a PCB and microcontroller together. The figures above show the initial design which is subject to change for both the hardware and software components of the Boxing Buddy.

On the more mechanical/structure side during the initial part of this divide and conquer document our group also made a project schematic which acts as a visual representation of what the initial thought for the structure of the device is depicted below in Figure 3.4.

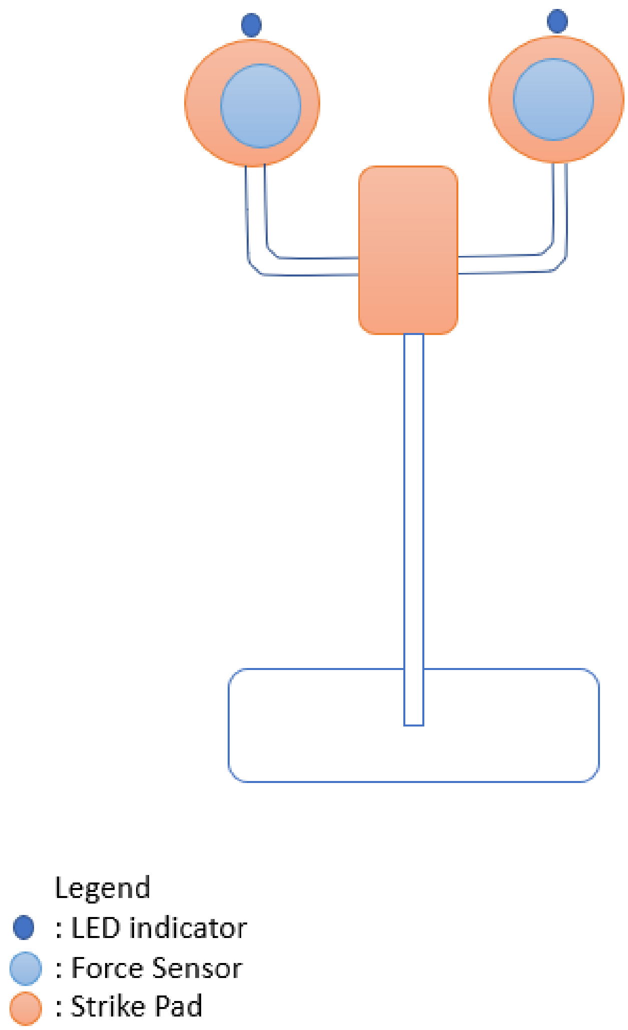


Figure 3.4: Initial Device Structure Design

From the initial schematic the design is subject to change as well similarly to our first rudimentary block diagrams from above. However the basic idea is shown within the schematic that highlights where the force pads and LED's will be located.

4.0 Related Standards and Realistic Design Constraints

This section discusses standards that are applicable to the design of the Boxing Buddy structure, electrical components, and app development standards. The engineering standards come from national organizations that establish a norm or requirement ensuring interoperability between different products in the same market.

This section will also cover design constraints related to the Boxing Buddy project. A constraint is any factor that will limit design in some way. This can be anything from cost and time to size of a particular part. Both Constraints and standards will cause certain parts and development paths to be chosen that adhere to the national and market standards as well as parts and development paths that do not violate the given constraints.

4.1 Force Sensor Constraints

The constraints and standards of the force sensors are summarized into different categories. The constraints of the force sensors include the amount of maximum force the sensor can endure and the dimensions of the force sensor.

4.1.1 Constraint of Max Amount of Measurable Force

The constraint for the maximum force the sensor can endure comes from the fact that every force sensor can read force. The amount of force it can endure and read is different amongst all of them. It has some kind of correlation with the expense as well. The more expensive the more force the force sensor can read. However, when it gets more expensive it also gets bigger which may be the reason why they can measure more. While these bigger sensors do measure more they are also used more oftenly for industrial applications. We had a handful of possible force sensors to choose but the constraint of the amount of force it can measure turned them away. That is why some force sensors, although they can measure higher forces and stay the same dimensions, get more expensive for a valid reason.

This constraint is measured in pounds per square inch (PSI) or newtons (N). In PSI the untrained person can reach up to 150 PSI. While the elite boxer athlete can reach up to 800 PSI. We want to have a project that can measure between these numbers or even higher because we don't know who our customer may be.

It may be an elite boxer just wanting to try out our project or a novice just picking up boxing. Having this information, we implement this into our project design in our force sensor. This is when the constraint comes in. Since 800 PSI is what an elite boxer can produce we are aiming for a force sensor that can measure around that variable. Looking at force sensors from now and on we look at the datasheets or information provided to see the max amount of force it can endure/read.

For example, some force sensors we were looking at were about $15 for a handful of them. Amazing deal, but, the max amount of force we found on the datasheet that those sensors can endure were little to nothing and if we are trying to measure the force of the user's punch then the range of 0-30 kgs for the sensor would be nothing compared to it. These were the piezoelectric force sensors and were a good candidate until that constraint was revealed.

The button load cell was an excellent choice as well. The maximum amount of force it can measure was well over 1000 lbs and that is more than enough for our project. However, the expense was around $100 which is ok for a force sensor but we may have two or three or four. It adds up.

4.1.2 Dimension Constraint

The other constraint is the dimensions of the force sensor. For where the location of the force sensor will be placed it must be small or medium. It will be going near the padding of the arm where the user will punch so it can not be too large. If it is too big then it can interfere with the user's punch and possibly affect the output. Too big can also add more weight than needed for the overall project where we can easily replace one smaller. The more weight can also affect the springs that are between the body and padding of the project. The weight may just damage the spring.

The thickness of the sensor can affect the places it is put as well. Too thick may make it not fit in the tightest spots. The shape that force sensors take can be a constraint too. We are looking for one that is just a normal shape like a circle or square or rectangle. Some force sensors take a weird shape which will make it hard to implement and use for a force sensor to measure the force of the user. But, force sensors that take that weird shape are usually used for other applications while still having a slight chance of being used for a reason like ours for measuring a user's punch but at that point we can just use another force sensor that has more promising deliverables.

We are trying to minimize the weight of the overall product. Large force sensors have seen to be more for industrial applications and uses and they cost more. For our project the constraint of dimensions on the sensor must be that it is small or medium. In specific, no more than 3 inches at most overall. It must be easily fitted anywhere within the project and be durable.

4.1.3 Cost Constraint

Each force sensor is entitled to a price. Now that price may not look pretty but it is what it is. We have no sponsor and all the expenses are coming from the project members pockets. As the prices of these sensors go up they have many reasons to do so. One of the reasons being that it has a higher tolerance for the force being measured. This higher tolerance for the force being measured and while keeping its petite size will make it more expensive. Some force sensors go for about $500 each. Which is a ginormous amount. When looking at force sensors we will look for one that is a good price but that can also fit into our other constraints of size and force tolerance.

4.1.4 Accuracy Constraint

Accuracy can be a constraint due to the fact that we need a force sensor that can measure the force of the user's punch accurately. Force sensors all have a tolerance of accuracy in their datasheets. The more expensive the better the accuracy will get no matter what type of force sensor it is. While looking at the constraints we will find a good middle in all of these categories in order to come with the best decision.

We want a good accuracy because that is the main point of our project. When we may have a budget we will choose to spend the most money in our force sensor category. The force sensor must be good because that is what will make the most impact on the quality of our project. Regarding quality the house of quality goes over a few items that are listed and then can be shown the importance of it in regards to the roof or the side of the house.

Each force sensor can either be analog or digital. These titles help divert them from how they output what the sensor is sensing. The types of outputs can be a constraint for the force sensor. However, it must give out an output. Outputs come in digital or analog. Digital outputs give out signals that are a discrete voltage or a digital measurement. The output comes out as digital binaries in ones and zeroes. These outputs are quicker to come out but less accurate than those of the analog. The price of digital output devices also comes in a cheaper price than those of analogs. Now analog outputs are more on the continuous and offer a more accurate output. Analog signals is a continuous signal that represents a quantity, like speed. Our constraint for our force sensor is that it must give out an analog or digital signal.

Although analog is more accurate it can be more expensive. Analog too might have the chance of us needing to convert it to digital just to get a binary output to put out into our app so it can display to our user the amount of force their punch was. With how digital is coming up where it is getting more advanced it can still be a valid option for us.

4.2 Software Constraints and Standards

The constraints related to the software development are centered around how long it will take to develop the mobile application as well as how much the application development will cost. The standards related to the Boxing Buddy mobile application involve adhering to Apple Store and Play Store guidelines as well as native user interface design standards. Ultimately, this application must be built on time while not being overly expensive and simple for the user to navigate.

4.2.1 Software Constraints

The two most important constraints in regards to software development are the time of completion and cost of development. These constraints will significantly limit how complex the mobile application can be as well as which development paths are available to the developers.

After the design is completed, there will be a maximum of four months to complete the entire construction of the Boxing Buddy, including everything from the mechanical frame to the mobile application. One primary factor that can lead to a long development time is the amount of learning of new material by the developers to create the mobile application. None of the developers have any experience building a mobile application, so the process in general will take time to learn. This means that any path that can minimize the learning time will be highly sought after. Furthermore, there are some programming languages that are more familiar to the developers than others. The developers are familiar with Python, C, Java, some Javascript, and HTML. Given that most mobile application development will use Javascript, HTML, C#, Swift, Java, or Dart, this constraint will limit which frameworks are chosen and which languages are chosen.

Another constraint related to time of development is support. Different frameworks and languages have differing levels of support when it comes to using them to develop hybrid mobile applications. Given the lack of experience of the developers when it comes to creating a hybrid mobile application, a development path with a lot of support would be the most ideal. In terms of software development, support refers to various factors.

First, support refers to the user base. The number of people who use a language or framework for the type of hybrid mobile application to be developed will influence the decision on whether to use that language or framework. The more people who have used the framework or language for this purpose, the more resources there are for debugging. Inevitably, bugs will occur and obstacles will arise that slow development. If there are plenty of resources available where people are using the same language and development and run into the same issues, the time it takes to overcome these obstacles and debug will be greatly shortened.

Support also refers to the developers of the framework and language updating the framework and language consistently to be up to date and remove any bugs. Additionally, this also includes any libraries that are created by the framework and language developers that can aid in the hybrid mobile application development process. If a framework is specifically designed to create hybrid mobile applications, then the libraries it includes will be more geared to the needs of the Boxing Buddy mobile application and thus reduce development time. This paired with a large user base will mean that there are more people making helpful libraries for hybrid mobile application development. If the language and framework is open-source, especially paired with a large user base, then there will be even more libraries created to aid in development that can be more niche and specific to the Boxing Buddy mobile application needs.

Open-source languages and frameworks will likely have a lot more packages and libraries to use. This is due to the nature of something that is open-source. If something is open-source, that means there are many developers. Virtually anyone can become a developer and create packages and libraries. These are checked and verified by the developers who run the language or framework. If the language or framework is not open-source, then there are a limited number of developers who must be paid by the company to create packages and libraries. This structure means there are fewer sources of packages and developers.

Code reuse is an excellent way to reduce the development time. When there are available packages and libraries already made for a specific task, then only implementation is needed. Creating this code from scratch can be very difficult and be prone to errors and bugs. It is always a good idea to implement packages that have been used successfully before. The drawback to using existing packages and libraries is security. Some malicious people may create packages that harm the operating system and steal information. This means it is crucial to carefully vet the packages that are going to be used to make sure they are safe and secure and have been used successfully before by many people.

The second major constraint is cost of development. There are several reasons why cost is a major constraint. First, one of the goals of the Boxing Buddy is to be affordable. At least more so than the other products on the market. The goal is that a customer would buy this product to workout in the comfort of their home.

It can also be bought by a gym to allow for more diverse workouts but primarily, the Boxing Buddy should be affordable by the individual.

Most programming languages are free to download and use. The frameworks on the other hand, whether back end or front end, can sometimes cost money. While this type of cost would not increase the price of the product for the user directly, it will increase the cost of development. The Boxing Buddy is being created without any sponsors or monetary assistance from the university so all funding must come out-of-pocket from the project members. Many frameworks are open-source and are free to use. Other options have different tiers of payment with the first being free. It is important that the mobile application functions as expected first and foremost. If there are frameworks available that allow the group members to develop at no cost without compromising the functionality of the mobile application, then these development paths will be sought first.

4.2.2 Software Standards

Because one of the goals of the Boxing Buddy is accessibility, the mobile application is intended to be a hybrid mobile application. This means that it will be available for download on both the Google Play Store and the Apple App Store. Publishing the mobile application on these platforms means there are requirements that must be adhered to. The regulations coming from the Google Play Store and the Apple App Store are referred to as industry standards.

The Google Play Store enumerates some requirements on their Android developers website. Mobile applications cannot be deceptive, malicious, or intend to harm the user first and foremost. This means that the mobile application for the Boxing Buddy must do precisely what it is described as doing. Because the Boxing Buddy application could store data, there is also the responsibility of securely storing this data. The Google Play store requires that the handling of data be fully disclosed to the user. This means the Boxing Buddy must include a privacy statement that the user agrees to. Another standard on deception is that the application cannot impersonate another person or company.

For safety and security, Google also requires the application to be up-to-date and continue to be updated frequently. The standard states that the mobile application must target an Android API level within one year of the most recent up-to-date version that Android has released.

In general, the Google Play Store prohibits any inappropriate content which includes anything to do with illegal substances, sexual situations, real-money gambling and contests, and hate speech.

The Apple App Store also enumerates their standards on their website. Many of these are very similar, if not identical, to the standards from the Google Play Store.These include restricted content like no pornography, illegal substances, illegal activities, defamation of a person or group, impersonation of a person or company, and hate speech among others.

Additional Apple standards talk about metadata. This refers to advertising and promoting. All descriptions and screenshots of the application must be accurate and submitted to Apple for review. The mobile application must not use excessive battery, and it must not encourage the user to violate security protocols in place. One place where the Apple Store standards differ from the Google Play Store standards in a significant way is the API standards. Apple requires the developed mobile application to use the current API operating system. This ensures the highest level of functionality and, more importantly, security for the user.

Another set of standards that the Boxing Buddy will adhere to are user interface standards. When creating a mobile application, there are certain design industry standards common to the Android interface and the Apple interface. For example, the location of a menu button or how to navigate between pages. The Android operating system does this in a specific way. As does the Apple operating system. These standards will be difficult to meet because the Boxing Buddy will be an application designed for both operating systems. The most user-friendly method of building an application is to model it after the native environment. While this won't be possible for the Boxing Buddy, the goal of user friendliness will be a top priority whenever possible.

4.3 Design Constraints

The constraints that directed the choices for the overall design of the Boxing Buddy are separated and broken down below. These constraints are examined together with the requirements for the project, the intended use of the product, the available or attainable resources, and various other considerations. Although the constraints limit the design possibilities, they ultimately lead to a better design and more effective decision making. The constraints focused on for this design consists of economic, time, manufacturability, and sustainability constraints.

4.3.1 Economic and Time Constraints

With no financial sponsorship, this project was fully funded by the four members of the group. We set a project budget based on the estimated cost of each of the components of the project as well as the amount that each of us would be willing and able to contribute to the project. After discussing these considerations, we settled on a budget that would not exceed $650. Although remaining below this budget limits the quality of parts we are able to purchase, we were still able to find parts that were of an acceptable quality for the intended use and environment.

In addition to the cost being based on our personal spending capabilities and realistic costs of components, we also kept in mind what the device would cost to users if it were to be on the market. One of our goals for the project was to create a device that provided a better option for users than what is on the market in multiple areas, one of which was the cost. Current similar products on the market cost around $2,500, so we wanted the Boxing Buddy to cost less than that while maintaining a desirable quality. When selling a product, the cost that the product sells for has to be greater than the cost it took to construct the product in order for the seller to profit off the project. With a budget of $650 to construct the Boxing Buddy, if we were to sell the product, we could still sell it for a price greater than $650 that would allow us to profit off of the sale while maintaining a market price lower than that of existing products.

Maintaining an acceptable quality for the project within the financial constraints is important to reduce the need for repeated spending by the users. While there are other training devices intended for use in the home available at a lower cost than the boxing Buddy, aside from the fact that they do not provide feedback like the Boxing Buddy, they are also available at a lower cost because they are made of lower quality materials. Reducing the cost by purchasing lower quality materials ultimately can create increased spending in the end when the products need to be replaced, as the life spans of the materials will not be as long as better quality materials.

The time constraints for this project are set by the courses this project is a part of. The start of the project is the beginning of the Senior Design 1 course and the ending of the project is close to the end of the Senior Design 2 course, around April 2023. Based on those dates, there is about an 8 month time period for the project to be completed within. Within these 8 months the project is to be thought of, designed, tested, constructed, and fully functional to meet the engineering requirements and the requirements for the Senior Design courses. For the Boxing Buddy, this means that by April 2023 it must be able to withstand a user striking the two strike pads many times while providing the user with an adjustable height for the device, accurate performance feedback, and an easy to use mobile application for the user to set training modes and view their performance within. To meet the aforementioned requirements within this time constraint, in the Milestone Discussion of this paper, the timeline of this project is laid out in the table with specific dates for various components of the project to be completed by.

4.3.2 Manufacturability and Sustainability Constraints

The manufacturability of the device is the extent to which the product can successfully be constructed while meeting the financial, design, and distribution requirements set for the product. When manufacturing a product, the creator has to keep in mind how the product will be constructed, the assembly requirements of the product, how the product will be tested, how the product would be shipped, and any repairs that the product may eventually need after use. For this project, we wanted to make it so that the amount of space the product takes up could be condensed for shipping and other transportation while maintaining few assembly requirements for the user within the home.

In the Mechanical Design section of this paper, the components of the mechanical design of the product are broken down in detail. The overall constructed design of the product can be seen in the Mechanical Design Testing section of this document. For this project, the components will be purchased separately and constructed by both the members of our group as well as a fabrication shop. The components will be purchased from companies that have experience and resources to create the products that exceed our own, thus providing better quality products for our project and further allowing for the project to be completed within the aforementioned economic and time constraints. Although the parts of the product that are purchased from the companies will require some time and money for shipping, this outweighs the amount of time it would take for us to manufacture each of the components ourselves while also providing better quality products. The metal components that need to be fabricated into the design of the product will be given to a fabrication shop with the design. These components will be given to the fabrication shop for construction rather than being constructed by us because the skills required to fabricate the design with the metal components is outside of our skill set, and it would require too much time for any of the group members to obtain the necessary skills at an acceptable level for this product.

Another constraint for this project is the sustainability of the product, or its ability to function the way it was intended dependably. The product’s sustainability helps determine whether or not it is a product that will last and continue to be desirable to consumers. The sustainability of the product contributes to the design of the product and purchase of materials, as lower priced material options or a poorly constructed product could reduce the life span of the product, and thus, the quality.

In order to maintain sustainability, the materials were purchased from multiple companies that were able to provide a higher quality product than we would be able to produce ourselves, so the product can function the way it is intended to and can last longer. The sustainability will also be maintained by protecting fragile components of the product with sturdier components to prevent damage.

The material for the frame of the Boxing Buddy was also selected to withstand the impact of the users' strikes over an extended time period.

4.4 PCB Standards and Constraints

Complying to standards and constraints are an important part of making safety a priority in whatever type of printed circuit board that is being designed. One of the most important standards that regulate printed circuit boards today come from the Institute of Printed Circuit known as IPC that was established in 1957. IPC is a trade association for the electronic interconnection industry. IPC produces and maintains standards that serve as a common guideline for PCB design, fabrication, assembly test, and other areas of concern as well as training, market research and public policy advocacy. The American National Standards Institute (ANSI) accredited IPC as a standard developing organization, and as a result IPC has the most widely recognized acceptability codes in the electronics industry. IPC publishes standards for nearly every phase of the electronic product development cycle, including design, purchasing, assembly, packaging, and more. Choosing to comply with IPC standards can differentiate a product from either succeeding or failing. IPC standards matter because adhering to these standards throughout the design/manufacturing process of creating printed circuit boards helps produce safe, reliable, high performing products that require persistent attention to detail.

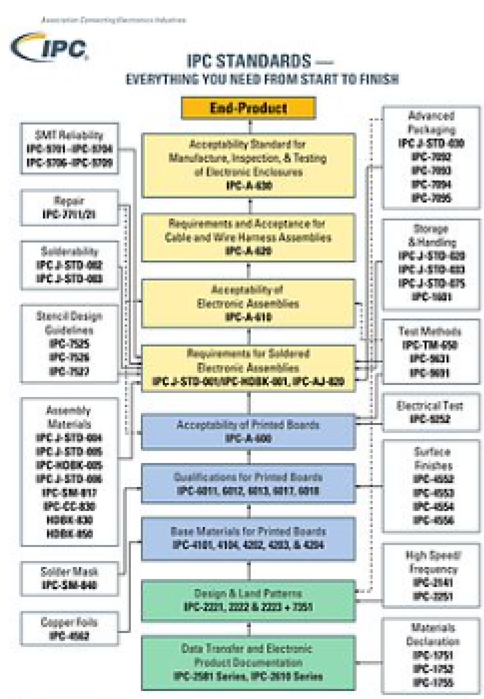


Figure 4.1: IPC Standard Procedure

Essentially IPC standards matter because they help ensure the end product printed circuit board is high quality. Complying with IPC standards can help a company improve its processes and products in numerous ways. In compliance with that our group will also be following and taking note of the IPC standards. Most namely the following are some of the most important IPC standards to take note of:

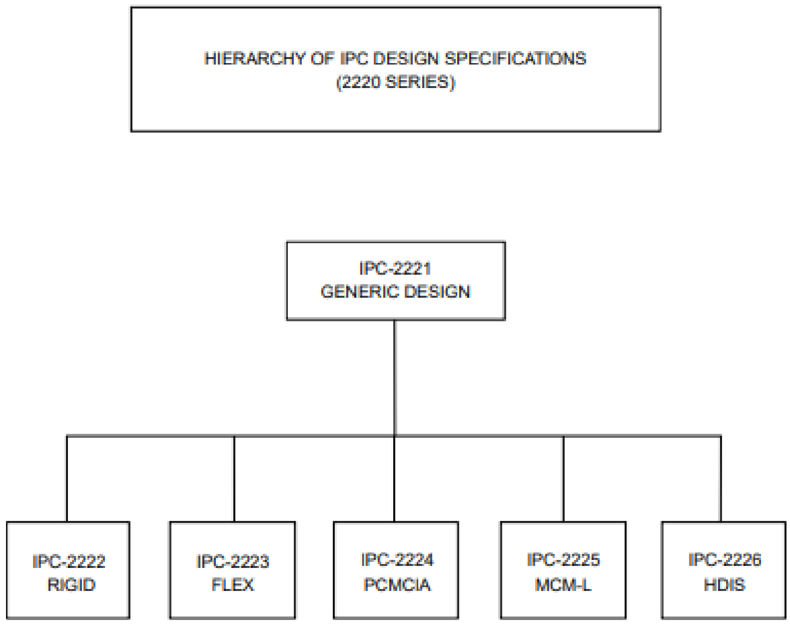


Figure 4.2: Hierarchy of IPC Design Specifications

<u>IPC-2221A Generic Standard on Printed Board Design:</u>

The IPC-2221A standard is intended to provide information on the generic requirements for organic printed board design. The standard addresses all aspects and details of the design requirement depending on the material being used to provide the structure for mounting and interconnecting electronic, electromechanical, and mechanical components. This standard covers how electrical considerations such as PDN bus layout, conductor clearance, and impedance control should be implemented on a PCB, as well as how thermal design should be considered. SEction 9 of this standard also covers holes and interconnection standards, which contain guidance on drill size, land pattern, and conductor spacing on a PCB. Our group will refer to this when needed throughout the designing process of the printed circuit board for our project.

<u>IPC-2223 Sectional Design Standard for Flexible/Rigid-Flexible Printed Boards</u>:

The IPC-2223 standard establishes the specific requirements for the design of flexible and rigid-flexible printed board applications and its forms of component mounting and interconnecting structures. It provides guidance on what flexible materials in the structures should be composed of, if they should be reinforced or not, and if there should be a dielectric in conjunction with metallic materials. Depending on the type of board there may be multiple different configurations including being a single layer board, double layer board, multilayer board, or multiple conductive layers.

This standard provides guidance on how to comprise a board of these different types and if it is better to use fully flex material or a combination of both flex and grid material in a certain scenario. IPC-2223 covered important design considerations such as pad/via placement, bend radius, coverlays, and structural specifications for flex PCBs. This standard is helpful to use in conjunction with IPC-2221 as mentioned before and our group will refer to this document during the PCB design process.

<u>IPC-2222 Sectional Design Standard for Rigid Organic Printed Boards:</u>

The IPC-2222 standard establishes the specific requirements for the design of rigid organic printed boards. Printed circuit board designers need to address many perspectives to be successful with their design, and IPS-2222 offers guidance on the following specifications:

- Layout Solvability - Complex Packaging Skillset
- Electrical Integrity - Signal and Power Performance on all Layers
- Manufacturability - Design for Excellence Considerations for High Yield and Lower Cost
- Application Considerations - Environmental, Performance, Shelf life, etc.

This standard results in optimal component placement, routing density and electrical performance to achieve an efficient design with high yield and defect-free manufacturability. Also goes into detail on factors that enhance the manufacturability of a rigid PCB, including material selection, hole size, interconnects, and mechanical properties. IPC-2222 is most commonly used in conjunction with IPC-2221 and both will be taken into account during the design of the boxing buddy PCB.

<u>IPC-7801 Reflow Oven Process Control Standard:</u>

The IPC-7801 Standard provides process control for solder reflow ovens by baseline and periodic verifications of oven profiles. The standard is intended to verify the operating parameters of the reflow oven, and for users of reflow equipment to baseline performance and periodically verify and demonstrate acceptable oven performance repeatability.

Summary of IPC Constraints Being Used:

Standard Name:	Purpose:
IPC-2221	Generic Standard on Printed Board Design
IPC-2222	Sectional Design Standard for Rigid Organic Printed Boards
IPC-2223	Sectional Design Standard for Flexible/Rigid-Flexible Printed Boards
IPC-7801	Reflow Oven Process Control Standard

Table 4.1 PCB Standards

<u>Constraints:</u>

In addition to standards there are also many constraints that have to be taken into consideration when designing and creating a printed circuit board. Constraints from board fabrication and assembly requirements drive PCB design. If a PCB designer has one incorrect or missing constraint during the late stages of layout design that could start a costly redesign cycle as even having routing changes for one signal could easily introduce problems with neighboring signals. Incorrect routing can easily turn into a large amount of rework causing delays and additional costs to the overall design of the printed circuit board and system as a whole. In order to stop this from happening it is important to start a list of signal integrity requirements in the form of constraints early in the PCB design cycle. To ensure throughout the design that proper electrical performance of the board will be completed. SOme of the common design constraints for PCBs are covered below:

<u>Length constraints:</u>

Length constraints impose limits on routing lengths for nets and net branches. These also include minimum length and maximum length, and matched length constraints. Using this approach of limiting establishes a restriction on relative lengths of group members. Our group will utilize this technique throughout the design process to ensure that too much space is not taken up by different parts that will cause issues in the long run.

<u>Routing Constraints:</u>

Printed Circuit Board routing is an important issue to address. Routing constraints are defined at the net-class level and are applied to all nets in the net class. Routing constraints impose restrictions on routing layers usage, on via type allowed for routing, and on the allowed trace-width range. There are different rules that are used in the designing and production of a PCB that help a designer bridge various components present in the circuit. For instance certain parts can only be routed in certain ways to ensure functionality. One thing in particular that is restricted is Trace spacing. Trace spacing must not be too far apart or too close to allow the circuit to operate, especially when there are varying amounts of power present on the one board varying trace spacing might be needed.

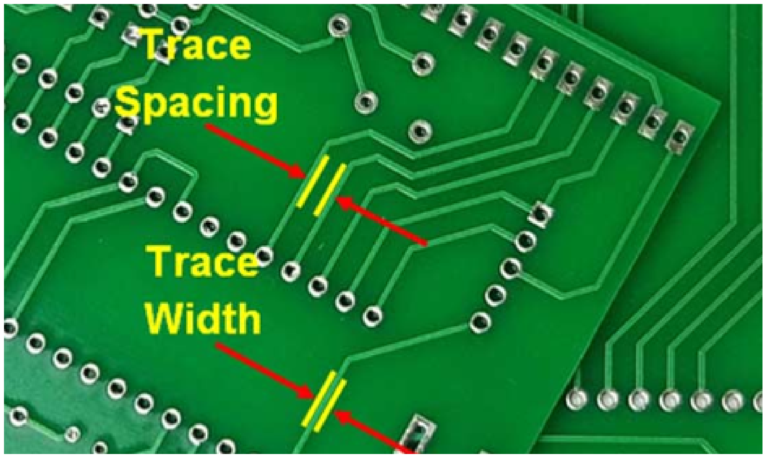


Figure 4.3: Example of Trace Spacing

Clearance Constraints:

Since the PCB is meant to be made up of multiple small components on a small board, the amount of space to mount the different parts is another constraint to take into consideration when creating the circuitry design. These issues can be resolved by making sure that all components have an absolute clearance from other components of 0.25 mm to allow proper routing and trace width around each of the individual components. Defining a minimum clearance between edges of two routing objects on a particular layer helps to define the clearance constraints within a two step process:

1. Create a set of named clearance rules and specify the clearance values for each rule
2. Specify how to apply those rules to particular nets referencing rules by their names

In the Boxing Buddy our group as stated before will use a clearance of 0.25 mm as a defining factor within the design of the printed circuit board.

Differential Pair Constraints:

Differential Pair Constraints include trade width, differential spacing, and maximum separation distance. The maximum separation distance specifies how long traces can run parallel while violating differential spacing value, and is layer independent.

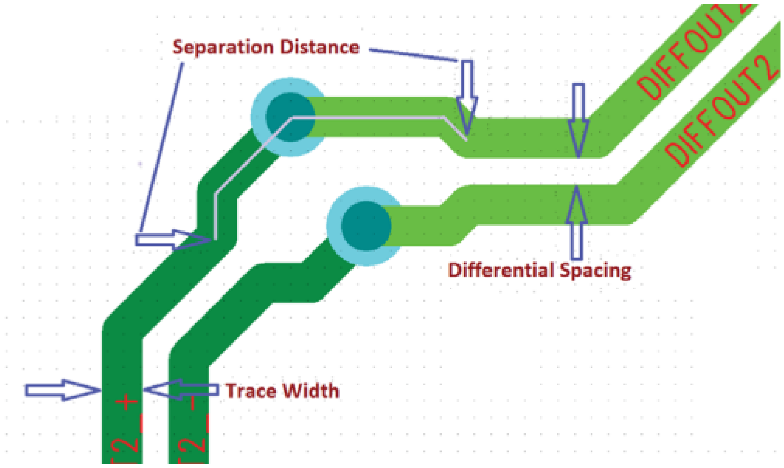


Figure 4.4: Differential Pair Constraints

Summary of PCB design constraints:

Constraint	Course of Action
Length constraints	Don’t make legs longer than necessary
Routing Constraints	Look up routing rules to ensure they are being followed
Clearance Constraints	Keep a clearance of at least 0.25 mm between all parts
Differential Pair Constraints	Set maximum separation distances

Table 4.2 PCB Constraints

From this section it is apparent that following the above standards and constraints is crucial to creating a good design for the system's printed circuit board.

4.5 WiFi standards

The common WiFi standards are created by IEEE. IEEE 802.11n has been the standard for WiFi usage across the web and mobile devices for at least the past ten years. This became an ANSI approved standard in 2010 and has been used since then. The IEEE 802.11 standard defines modifications to the physical layer and the Medium Access Control layer to achieve higher communication throughput in a WiFi connection. The maximum throughput is at least 100 Mb/s as defined by this standard.

There is a more modern standard from IEEE designated at IEEE 802.11n. This standard is colloquially referred to as 5G. It is meant to replace the previous WiFi standard because it increases capabilities. 5G, or IEEE 802.11n, exploits Orthogonal Frequency-Division Multiplexing to modulate the signal across several different channels to reduce interference. This standard is capable of any operation below 6 GHz. 5G can also operate in high bands as well as above the 24 GHz threshold. This increased bandwidth allows for higher capabilities, multi-Gbps throughput, and short delays. Being able to access these bandwidths allows for more coverage thus reducing traffic and minimizing interference. This will provide the user with a smooth interface that allows all functionality to be achieved without any lagging.

4.6 Power Constraints and Standards

The Boxing Buddy system is being designed such that the power supply comes from the 120 Vac wall outlet and is then converted to DC and routed to all other parts. The goal for power routing is to be as convenient as possible. Since the Boxing Buddy is designed to be used at home primarily, the power source that most people are comfortable with would either be the wall outlet or a battery. Batteries must be recharged or replaced which would either still use the wall outlet or be thrown away and generate more garbage.

4.6.1 Power Constraints

Several constraints may restrict the Boxing Buddy group members when selecting the desired power routing method. Because the Boxing Buddy is to be powered from the wall outlet, there are length constraints that must be observed. In this case, a longer length is desired for maximum functionality by the user. The Boxing Buddy system will need to be placed near an outlet by the user and the longer the cable length, the more flexibility the user has as to where to put the structure. Of course, extension cords can be used whenever needed but one major goal of the Boxing Buddy is affordability. The goal of affordability guides the group members to design the structure such that minimal external or third party parts will be needed to use the Boxing Buddy. This ensures a minimal cost for the user.

Another constraint to be aware of in regards to the power routing is the voltage level. The PCB, microcontroller, and accelerometer will all need power supplies. These electronics operate at a much lower voltage compared to the wall outlet. Not to mention DC power as opposed to AC. Obtaining an AC to DC adapter that converts the wall 120 Vac voltage into a low voltage like 5 Vdc will be sought after to reduce the burden on the PCB. Having too high a current can damage the conductors on the board. Current must be limited to 1A throughout the PCB whenever possible. A voltage regulator can be used to change the voltage level but the current must be maintained to be 1A or less to ensure no damage occurs to the conductors.

4.6.2 Power Standards

The main standard associated with power routing for the Boxing Buddy is the wall receptacle. There are several different types of receptacles seen in American homes. There are 15 and 20 amp varieties as well as receptacles that can be used to prevent damaging fault conditions.

Figure 4.5: Standard American 15 Amp Wall Receptacle and NEMA 1-15 (left) and NEMA 5-15 (right) Standard Plugs

It is important to follow these standards for safety of fault conditions that can damage electronics or cause the house to lose power. It is also important for accessibility. All American homes have these, thus increasing accessibility.

Most plugs and wall warts can be used in any of the ANSI standard wall outlets. It is less important to specify which outlet is used and much more important to specify a cable that is compatible with any of the outlet types. Compatibility is equally as important as accessibility. The ANSI standard wall outlet is 120 Vac if it is connected to single phase power like the outlet shown above. The cable used in the Boxing Buddy will connect to this outlet and output a DC voltage that is much lower to be picked up by the main controller. There are two types of standard plugs associated with these outlets. Type I is referred to as the NEMA 1-15 standard plug. This plug has two flat blades and no grounding pin. The flat blades will plug into the two slits at the top of the outlet. The other type of plug is called the NEMA 5-15 plug. This plug has the same two flat blades but also has a cylindrical grounding pin that plugs into the bottom of the outlet. The images above show a comparison of the two standard NEMA plugs.

5.0 Research and Part Selection

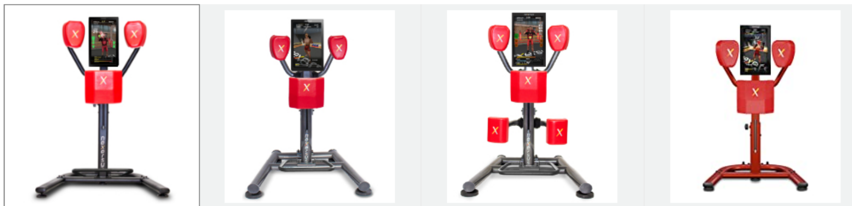
This section involves the research of our main electrical and coding components and similar products. Force sensors, reaction time, microcontroller chips, PCB, LED, programming languages, etc. Within each section there will be comparisons and chosen devices for our project with a reason why. Similar products will be listed and shown what they provide and their cost.

5.1 Existing Similar Products

As with all products and technologies, the best way to create the best design possible is to improve on past/competing technologies. There are already existing products that have a similar motivation and goals that the Boxing Buddy does, and this section will highlight some of them and what features Boxing Buddy will recreate.

One existing product that helps train boxers is the Nexersys N3 Interactive Boxing Trainer & Sparring Partner. This product delivers an interactive boxing experience through personalized training. The goal of the Nexersys N3 is to be a personal coach and sparring partner with personalized training. Depending on the model used of the product which is summarized in table 3.1 there are a varying of 3 to 5 strike pads per machine and the product is able to record metrics including score, power, accuracy, speed, and strike count.

The Nexersys N3 product line offers great benefits to its user within the training area, however, it is a very costly piece of equipment for an individual to invest in for themselves being priced at $2,495 dollars. Because of that, Boxing Buddy will take the ideas that are being executed within this system and try to duplicate them for a lesser cost to the average user. Another feature of the Nexersys is the large, digital user interface attached to the structure that can be seen in table 3.1. Having the screen attached to the structure creates a greater risk for damage to the screen and contributes to the weight of the entire product. For these reasons we chose to create a user interface that could be used on the consumers personal mobile devices instead. Having a screen on the system may be considered more user friendly. A major goal of the Boxing Buddy is to be very user friendly. Therefore, the team will strive to make the mobile application as clear and simple as possible to avoid adding any confusion by not having the mounted screen.



	N3 Elite	N3 Pro	N3 Commercial	N3 Youth
Subscription Required	✗	✗	✗	✗
Ideal Use	Home	Commercial & Home	Commercial	Home
Tablet Size	18.5″	22″	22″	18.5"
Number of Strike Pads	3	3	5	3
Net Weight	190 lb	205 lb	235 lb	170 lb

Figure 5.1: Nexersys Interactive Boxing Trainer Specifications

Another similar product to this is traditional boxing training done with a trainer at a gym. Obviously this alternative does not compare on a technological side, however we still thought it was important to touch on for cost reasons. A monthly gym membership can easily run anywhere from $100 - $180 a month for classes that might not work with the user's schedule. Because of that Boxing buddy will try to create an alternative training experience that can be done at any time that is convenient for equal to or less than the price of a year of traditional boxing training ($1200 - $2160)

In addition to the aforementioned products, another similar product is standard boxing training devices that are intended to be used within the home. Like a gym, they do not typically have the technological aspect and do not provide feedback; however, the prices of these products tend to be significantly lower than the Nexersys N3 or a gym membership. These devices typically range from around $100 to $500, and only require a one time cost. In order to compete with the low price of these products, the Boxing Buddy is constructed in a way that allows for the price to be lower while offering the added benefit of the technological interface to provide users with feedback.

5.2 Force Sensors

Force sensors convert weight, tension, pressure, force, etc from a mechanical input. It converts it into an electrical output signal that can be seen. So in our case a force we are producing is from the user's punch. That force will be converted into an electrical output that can be read on an interface of our choosing. There are many different types of force sensors in the world and we go over a handful of them. We go over how they work and what they measure and how good of a fit the sensor would be for our project. The best candidate for a sensor would have to have the characteristics of being low cost, compatible with microcontrollers, small in size, and durability.

When choosing our force sensor as mentioned before the low cost will be a factor. Many of these sensors are expensive. Not to mention that we only need two and that when buying in bulk they will come in minimum packs of 5. Prices will be shown on every sensor mentioned and it will help with the decision of which one to pick.

Another factor is the size. In other words, the dimensions of the device. To our surprise, many of the sensors are quite big. The fact that in our project we will need a sensor that can either fit between the padding or in a small tube towards the body of the structure is restricting. So in choosing our sensor we will need to check the dimensions of each and every one of them to make sure we are not getting a big sensor that wouldn't fit.

Durability is just as important a factor. Due to the fact that the sensor will be close to the action. All these punches, elbow grease, sweat, etc. will be near the sensor. We will need to find a sensor that is durable as well. Durability will also be affected by where we decide to locate it. Not to mention we will be more inclined to find one that is water-proof or atleast water-resistant.

Measurable force threshold is key. We are looking at the datasheets and specifications of every force sensor we come across to make sure that whatever punch we throw at it will be measurable. Imagine throwing a punch and the sensor can not even read the amount of force the user puts out due to the fact that the sensor is not capable enough.

Some sensors have a correlation in price and measurable force. As the price gets higher the more force they can withstand. There are some outliers though. The maximum measurable force we are trying to get is around 800 pounds per square inch (PSI). This is the average PSI of an elite boxer. If we can grab a hold of a sensor with this type of measurable force then we are good since most average people wont get around that numerical value. Anything that can be near 800 pounds of force is more than enough for our project. The average human can punch with around 360-450 pounds of force. It is important the Boxing Buddy can not only withstand that amount of force, but significantly surpass it. This will ensure several things. First, it will ensure the force readings are accurate. Second, it will ensure that no components get damaged from punching too hard. Third, it will ensure that a very strong user will still be able to utilize the Boxing Buddy to its fullest extent.

5.2.1 Piezoelectric Sensor

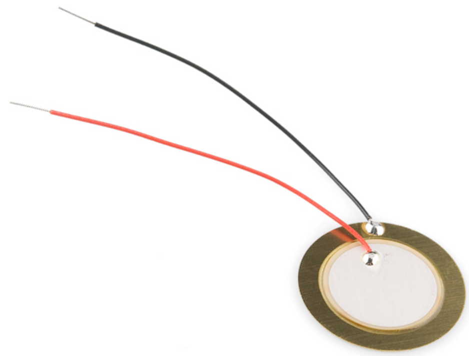


Figure 5.2: Piezoelectric Sensor

This type of sensor is called the piezoelectric sensor. It works by measuring the voltage across an element generated by the pressure applied. It can measure the changes in temperature, strain, pressure, or force. For our occasion we would use it for force in our project. From this, the voltage can be read and displayed perfectly suiting our requirements.

Not to mention the price of these go about ~$15 a pop. This is comparatively low compared to all these other force sensors where just one can go for ~$500. But to my understanding it might just be because those types of sensors may be more for the industrial side of things in the world.

The size of the piezoelectric sensor is small. Averages around 30 mm in diameter. This will easily fit anywhere on the structure of our project. The dimensions are 2.36 x 1.18 x 0.23 inches. This includes the tail they all come with.

The durability is high since some come waterproof while others do not. In our case of the project we are doing a waterproof one will be more favorable due to the fact that sweat might just enter into the sensor. If it is bound to happen we have to look out for those possibilities even if the chances of it happening is less than one percent.

The measurable force can be around 30kg. Which in reality would not be enough for the average or elite boxer. Because of this the piezoelectric sensor will still be considered but this is a strong negative.

Therefore, for our project we believe this piezoelectric sensor is a good candidate. It perfectly aligns with the use of a microcontroller to display the force provided. When pressure is applied on the sensor that pressure can then be converted to a voltage that can be read on a multimeter. This value of a voltage can then be converted to force to show the user the amount of force their punch did.

5.2.2 Button Load Cell

Figure 5.3: Button Load Cell

A button load cell is a small sensor that is based on strain gauges that are used in measuring force. This can be compression or push force. It converts mechanical force into numerical digital values that the user can read. The button cell ranges in size as well. From just a couple of millimeters to several inches in diameter.

Regarding our project this type of load cell is a consideration for use in our padding to measure the forces of the users punches. However, the fact that the force is only measured when pressing the button is a limitation to measuring the force because what happens if the user punches it on the side of the button? It may not correctly measure the force of the punch. A load cell that can do what the button load cell can do but has a pancake button that covers the entire surface would be a perfect match for it.

A solution to the problem of the button being the only sensitive part can be adding a plate that is behind the padding. This is connected straight to the button and whenever the user hits the padding anywhere it will apply at least a small amount of force that will be able to push the button and read the force. However, the downfall to this is the force reading won't be accurate if hit on the edge of the plate compared to right in the center.

The cost for one of these sensors is all over the place. It really depends where you get them from. For now we will be looking at digikey. One of these button sensors is around ~$166 for one. Where a pack of 5 will be ~$779.

The button load cell comes in all kinds of sizes. From 10 mm in diameter to 82 mm. The fact that it is sold so small as well can be held as a benefit for a candidate in our project.

Durability on one of these sensors is good as they are built for industrial purposes as well. On the data sheet it says a long term stability (1 year). It also has a long cycle life expectancy.

The range for the forces is dependent on the size of the button load cell. Where the smaller you get the smaller the ranges get. It is a strong correlation. We would use the 8 mm diameter one if we had to choose and this one has a range of max 500 lbs.

5.2.3 S-Beam Load Cell

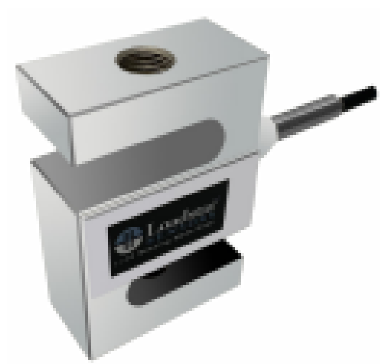


Figure 5.4 S-Beam Load Cell

The S-Beam load cell is a load cell that is used to measure tension(pull) or push forces. These load cells have a spring inside of it and placing force/weight on it causes an elastic deformation. The S-Beam also has strain gauges that measure the length of the deformation right in the middle of the device.

Regarding our project this type of load cell is not what we are looking for. This load cell when used in force applications must be calibrated and have a lot of trouble calibrating they are not mounted. If used in our project it would be placed upon the padding which may not be mounted securely and upon the punches may unalign and misplace the sensor.

The cost would be around ~$300 for the one we choose. The fact that we need two would make it $600. For it being that much money we might as well make sure it is the one we want. However, one of its downsides is its size/shape.

The size is quite small being around 3 inches in height with 2 inches in length and 0.75 inches in width. But in order for it to work in our project we will need to orient it a specific way and that way would make the 3 inches in height the width.

The durability of the s-beam force sensor is good enough and on average says it can be used 1 million times.

The range of the maximum force depends on the type of s-beam load cell we acquire. The one that fits our project would have a 1000 lb force maximum. There are differences between the compression forces and the tension forces but for our project the compression is what we are mainly focusing on here.

5.2.4 Through Hole Load Cell

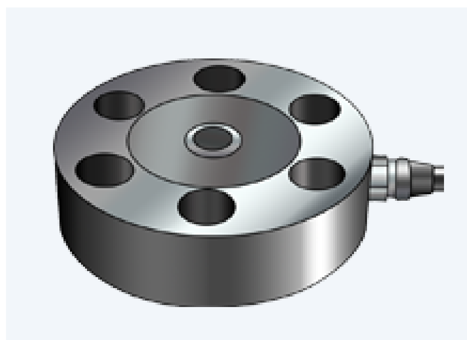


Figure 5.5: Through Hole Load Cell

The through hole load cell is used to measure force applications but where the structure has to go through the load cell. In most cases this type of load cell is used in industrial applications such as bolt force measurements, clamping forces, and rolling mill overload monitoring systems.

Regarding our project this type of load cell is of no use as nothing has to go through the load cell. The punches being thrown won't be needing to go through the hole.

The cost of one of these through hole load cells is ~$500 or ~$1000 for two. Not only is that a lot of money but there are other reasons why this is not a good fit for our project and for the one reason it won't be worth it to get something expensive and have it not fit all our needs.

The size of this load cell is at a minimum of 75 mm which is alright for our project. The fact that it comes so big is because these types of load cells are used so much in the industrial side of the world. Majority of industrial applications are big in size so it makes sense for the sensors being used there are big as well.

Durability is up to par and good.

The force range in these load cells are far greater than any of the ones previously mentioned. They start at atleast 550 pounds and go up to 11,000 pounds depending on the type that is being looked at.

There is a different type of through hole load cell that will be discussed right under this one and it is called a pancake load cell. The difference would be the fact that the pancake load cell is used in higher capacity.

5.2.5 Pancake Load Cell

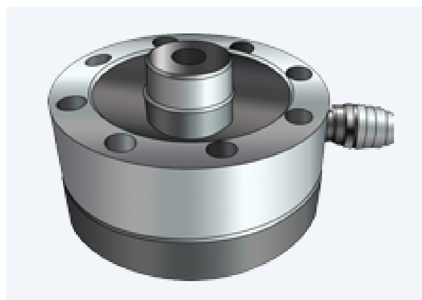


Figure 5.6: Pancake Load Cell

The pancake load cell is similar to the through hole. So for the same reason this won’t be a suitable load cell for our project. This one is just a variation of the through hole load cell but I decided to add it on here to see the differences. The differences are the capacities that it has available. The capacity in other words is the range of the force that it comes with.

Regarding our project this type of load cell is of no use as nothing has to go through the load cell. The punches being thrown won't be needing to go through the hole. Just like the through hole.

5.2.6 Miniature Load Cell

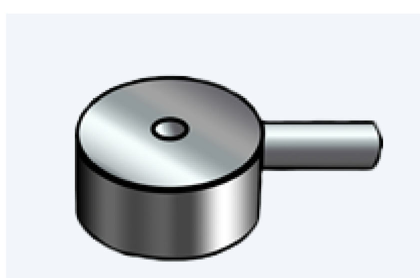


Figure 5.7: Miniature Load Cell

These load cells can have push or pull mechanics/applications depending on the one you purchase. However, these are used for smaller applications. For example, if the place you need to measure the force is a small confined space where a normal sized load cell would not fit.

For our project this is not a sensor we are looking for since we have a good amount of space. However, if deciding to use the force sensor in the pipe where punches will be pushed to the sensor then this may be a load cell to consider.

The cost of a miniature load cell is around $500. For the size it makes sense as smaller things tend to get more expensive since the designers and engineers do not have a lot of space to work with and are harder to manufacture.

The size is just like the name says: miniature. We are working with a small space but nothing that a regular button cell can't handle.

The durability is to be good but we forget that it is smaller so it may have a less life expectancy than a normal button load cell.

The range of the force is great after all although it is miniature. There are many different options that have different types of ranges for the force. But a great difference is that it can measure super lower force levels if chosen the smallest one. It can be used if measuring something really light and in a tight space. However, in our case that is not what we need in our project.

5.2.7 Single Point Load Cell

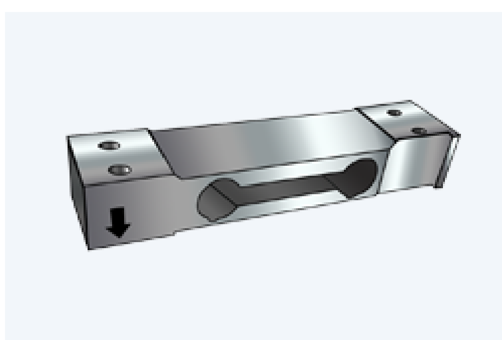


Figure 5.8: Single Point Load Cell

The single point load cell is used for compression or tension force. Load cells like these are accurate and are for using weight scales.

For our project the single point load cell is not suitable for our needs as we need the whole surface to be able to measure the impact of the punch. However, buying two and making a flat plate surface on it and that is where the impact of the punch can be feasible.

The price for one of these single point load cells is ~$300. This is on the better side of things for cost.

For the size the length is 2 inches while the length and width are just under 1 inch. The size is at a pretty good standing for our project and can be something we work with. However, those are the ones where the force range is significantly lower than the larger ones.

As always the durability for these types of load cells are excellent.

The force range can be diverse depending on the size of the single point load cell. For our purpose we would use the smaller ones which have a lower force range which is a big no.

5.2.8 Capacitive Load Cell

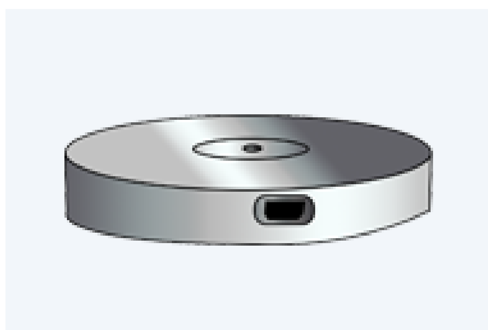


Figure 5.9: Capacitive Load Cell

Capacitive load cell work by the change in capacitance due by the force that is being applied. The change in the gap between the two capacitive plates generates a change in the value of the capacitance. Then measured as change in frequency that is sent to UART/USB interface devices that give out the force output and everything.

For our project this may be a good candidate due to the fact that it already shows the force in the USB interface devices. The only downside would be for how it gets tied up with the app. It already has an interface so if somehow we can get it to where the interface can be shown in the app then it would be better.

The cost is about $500 and everything about it pretty much does not line up with the important requirements.

The size is alright with the capacitance diameter sitting at a 3 inch diameter. The durability for this good. The force ranges to max 500 lbs.

5.2.9 NTEP/Waterproof

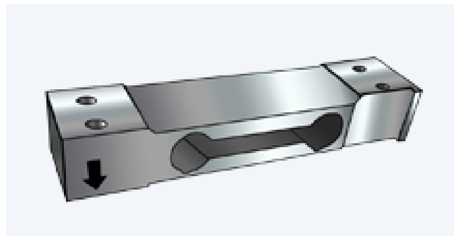


Figure 5.9 NTEP/Waterproof

This is rather a title or description for the load cell. These types of load cells are for very niche conditions. For example, where there might be varying temperatures, these load cells with the title can withstand the difference in temperatures. Steaming hot or blazing cold.

For our project we will not need these types of load cells because the worst conditions our load cell will be under is room temperature or even maybe some sweat getting onto it but we will make it where the sweat won't be able to go to the heart of the electronics.

However, what would be a good benefit would be whatever sensor we are using to be waterproof or at least water resistant.

Due to the fact that flying sweat particles or the sweat getting on the gloves if using gloves are being in contact with the padding which is relatively close to the sensors. We do not want to take any chances but either way we are making sure that the sensors are in a secured location within the structure of our project.

5.2.10 Accelerometer

The accelerometer as it is between the normal strain gauge type of sensors VS accelerometer to measure the force of a punch. An Accelerometer is an electromechanical product that measures acceleration. However, accelerometers respond to vibrations but there are different types. While the modern way to measure the force of punches or kicks among competitive athletes is using force sensors that have strain gauges. This relevant technology rather uses acceleration, hence the name. This accelerometer is long cylinder shaped and tends to be masked with padding. Or could just be a square. In reality they come in all shapes and sizes to fit the product they are meant to be in. The measurements of the acceleration are combined with a low-pass filter with a cutoff frequency that will make it easier to read. This as well helps with the error of the machine measuring the force of the user’s strike. Making it less than 1 percent.

There are different kinds of accelerometers. First comes the piezoelectric accelerometer, this is a common accelerometer that uses lead zirconate titanate or PZT. This element produces a charge when acceleration is detected and then measures the vibration. In our project the measurement of vibration is not ideal since we are trying to find the force of a left hook or right hook or punches in general.

The next type of accelerometer is the piezoresistive accelerometer. This is different from the one before as this one measures stress. It acts mostly like the common strain gauge sensors. This type of accelerometer has a piezoresistive material inside of it. When stress is recognized the material will be deformed/stretched out and this change in resistance is calculated and output to be shown. This may be a good ideal for our project, however, measuring stress VS measuring the force are two completely different things. We may be able to convert stress to force interchangeably in the code when it is read and displayed on the interface.

Capacitive Micro-Electric-Mechanical Systems (MEMS) accelerometer is the next common accelerometer. This type of accelerometer is used to manufacture both capacitive and piezoresistive accelerometers. However, it is also used to measure the changes in capacitance within the masses among the acceleration.

5.2.11 Conclusion

The conclusion will include a table with the sensor, cost, and maximum amount of force it can measure with a conclusion about that potential sensor in our project.

Table 2. Conclusion of Sensor (Cheaper) depicts all the choices of the sensors and concludes the potential match in our project with the cost and maximum force it can measure.

Table 3. Conclusion of Sensor (Expensive) covers the same of table 2, however, does it in the sense of the more expensive sensors.

Sensor	**Cost ($)**	**Maximum amount of force**	**Conclusion**
Piezoelectric sensor	~$15 for a pack	30 kg	This would be a good candidate but the constraint with the maximum amount of force is limiting.
Button Load Cell	~$166	500 LBS	This is a good candidate and this might be the one we choose.
Accelerometer	~$30	N/A this sensor measures acceleration	The accelerometer is a viable choice since F=MA and if we know the mass of the object then we can easily find the force applied.

Table 5.1 Conclusions of Sensor (Cheaper)

Sensor	Cost ($)	Maximum amount of force	Conclusion
S-Beam Load Cell	~$300	1000 LBS	This is a good candidate but for the price we can get a better one that suffices many other needs.
Through Hole / Pancake Load Cell	~$500	11,000 LBS	These types of load cells are used in industrial applications. Meaning the threshold is way above what we need and therefore make the price expensive. We could get a better deal for our needs.
Miniature Load Cell	~$500	220 LBS	These miniature load cells go for quite a price. But for a reasonable reason they are smaller.
Single Point Load Cell	~$300	110 LBS	The single point would not be a good candidate because of the constraint on the force threshold. It could go higher but that would make the dimensions of the single point load cell bigger.
Capacitive Load Cell	~$500	500 LBS	The capacitive load cell consists of using the difference in capacitance to measure force. It has a USB/UART interface that can be useful.

Table 5.2 Conclusions of Sensor (Expensive)

5.3 Reaction Time

The way reaction time will be measured in our project is by visual representation. There are many forms of reaction time, there is audio, touch, hearing, and visual reaction time. In this case An LED light will go off on the padding indicating for the user to strike. More specifically right on top of the padding. From there the user will strike and the calculation for the reaction time will come on the app. The next light will come up and so forth the user will punch continuously. The way it is visual reaction time is at the time the user will click a button and therefore will start the program. A short instant after pressing start the LED lights will start and the user can go along with the training. The procedure will follow these steps:

- Press start
- LED lights
- User plays
- User checks the interface for their results

Testing the Reaction Time

The way we will test reaction time before completely implementing it is by using an arduino and a breadboard. Alongside jumper wires, resistors, a button, and LEDs. The arduino will consist of the code that will be able to tell the reaction time within when the LEDs get lit up and the piezoelectric sensor gets activated. The button will work as a start button. The resistors work so the voltage wont overload the arduino board and fry it.

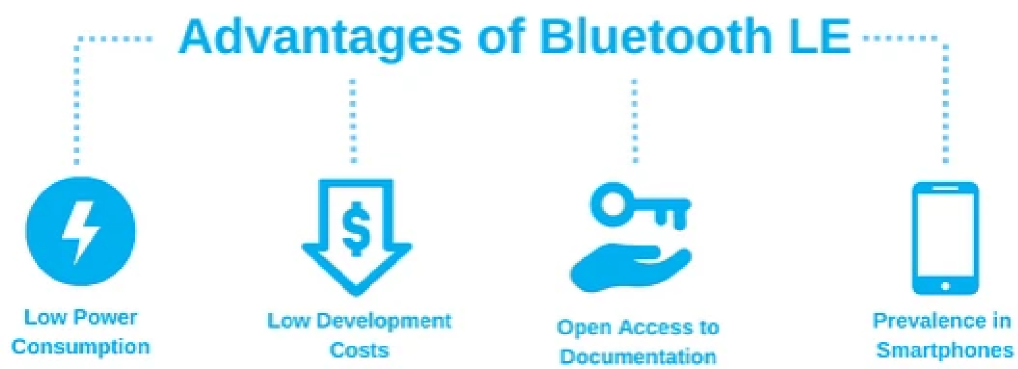
The LEDs can have an effect on the outcome of the reaction time values. Depending on the quality of the LEDs. Not only will the quality but as well as the color of the LEDs.

5.4 Sensor to App Communication

When beginning this project, a main topic that had to be researched was how the data from the sensors we chose would be able to connect and transmit to the microcontroller chip chosen and subsequently to the app interface to be displayed for the user. Below are some of the different options that our group looked into for this purpose:

Bluetooth Low Energy:
Using a bluetooth module/chip was one of the first ideas we as a group researched in order to communicate between the app and the controller. One option was Bluetooth Low Energy technology which is a wireless personal area network technology designed to provide communication between multiple

devices using a reduced amount of energy as compared to classic bluetooth methods. Bluetooth Low Energy technology has become one of the most popular technologies for new electronic hardware products due to it consuming little amounts of power and being affordable. There are two main ways to incorporate Bluetooth Low Energy into a system, either by using a module or using a System-on-a-Chip. The System-on-a-Chip bluetooth module can act as a microcontroller itself, however in the use of the Boxing Buddy we would have a different master controller connected to our customized PCB with the Bluetooth Low Energy System-on-a-Chip acting as a peripheral device. By Mesh Networking with the Bluetooth Low Energy Technology it is possible to communicate with multiple devices inside the network. The way the mesh networking works within the bluetooth module is fully decentralized using a controlled flooding method wherein one bluetooth device receives the message and then propagates it out to the rest of the devices in the network. Each device then receives the message in sequence so that the message is not repeated to a device that already previously received the message. This flooding technique is more scalable and easier to use on larger products than a traditional routing method which would be beneficial in the use of the boxing buddy system to allow communication of data straight from the sensors to the app interface.



Advantages of Bluetooth Low Energy

Figure 5.10: Advantages of Bluetooth Low Energy

There are many advantages of Bluetooth LE some of which are summarized in the image above. Most notably its prevalence in smartphone technology and connection as well as having a low cost make it an attractive option to use in the Boxing Buddy development. However Bluetooth Low Energy does have some disadvantages as well.

Disadvantages: Firstly the data throughput of Bluetooth Low Energy is limited based on the physical radio rate that transmits the data. The rate can vary depending on which type of bluetooth you get ranging from 1 Mbps to 2 Mbps. BLE was also designed for short-range applications.

Bluetooth Low Energy technology is popularly used in fitness tracking for smart watches and fitbits to monitor things such as heart rate and step tracking and works well with short bursts of data at close range, which leads our group to believe that it would be a good option to use to transfer data from sensors to the app from small bursts of workouts.

Bluetooth Low Energy Chips Price Comparison

Chip	Price
Nordic Semiconductor - nRF52840	$7.42
Nordic Semiconductor - nRF52832	$5.48
Nordic Semiconductor – nRF51822	$6.92

Table 5.3: Bluetooth Low Energy Price Comparison

Figure 5.11: Bluetooth Low Energy Chips

Bluetooth Classic:

Another bluetooth option is the traditional Bluetooth classic method of communication. Bluetooth classic is designed to handle more data throughput than bluetooth low energy, however it also consumes much more power quickly. Bluetooth compatible sensors are ideal for applications that require short range connectivity between 50 to 100 m, low power consumption, and data transfer rates up to 1 Mbps.

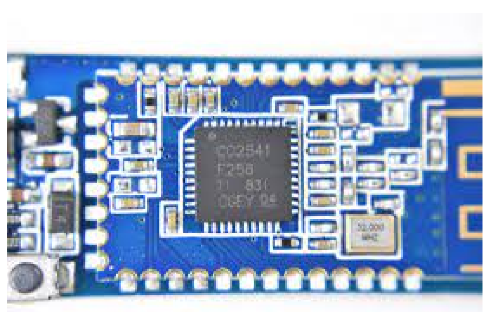


Figure 5.12: Bluetooth Module

Comparison of Bluetooth Low Energy vs. Bluetooth Classic:

	Bluetooth Low Energy	Bluetooth Classic
Protocol	Bluetooth SPecification	Bluetooth Specification
Operating Frequency	2.4 GHz ISM band	2.4 Ghz ISM band
Power consumption	Low; good battery life	High
Data amount	Low	High

Table 5.4: Comparison of Bluetooth Low Energy and Bluetooth Classic

Wifi:

Wifi chips are based on the IEEE 802.11 standard and are used in fixed broadband wireless access networks that use point-to-multipoint architecture. Wifi wireless chips are used in products where users have access to networked services without standard cables, which is what we are trying to accomplish as a form of communication protocol with the Boxing Buddy.

Adaptive Channel expansion within wifi chips has allowed an exponential growth in bandwidth rates for WiFi chips in general. These Wifi chips generally use non overlapping channels in the range of 2.5 megahertz which can transmit at 120 megabytes per second using Wi-Fi protected access which is a form of advanced encryption standard.

There are two main hardwares used for wifi modules including a wifi system on chip similar to the bluetooth low energy model and an application host processor. One of these modules comes with multiple inputs and output devices such as analog to digital convertors, digital to analog convertors, crystal oscillators, serial communication interfaces, and other applications on the hardware side. When it comes to software the wifi module includes a device driver, and an integrated 802.11 security layer, and a management and monitoring utility. Advantages of using a Wifi module as a form of communication is that it greatly simplifies the connectivity design process.

When choosing which wifi module in particular to use in our project certain parameters are important to look at to make sure they will meet the specifications needed by our sensors and controllers and set standards in general. In the table below some of the standards needed to be considered when choosing a Wifi module to use are summarized:

Standard	
IoT architecture	‘Single’ or ‘host processor + Wifi Module’ solution
Wifi protocol support	Data rate, range, and power requirements from the IEEE 802.11 wireless LAN standards
Operating Frequency	IEEE 802.11 standardizes the different specified frequency ranges and channels that certain countries and industries can operate in
Secure WiFi Support	Must make sure WiFi supports the required security standards such as WPA, WPA2, WPA3, WPS
Hardware Interfaces	WiFi modules have many I/O’s and peripheral interfaces
Certification	IoT devices need to comply with regulations in specific countries

Table 5.5: WiFi Standards needed for Selecting an Option

Taking these into consideration here is another table comparing some of the most popular Wifi modules to compare for use within this project.

Chip	Price
ESP32 by Espressif	$4.2
CC3200 by Texas Instruments	$73.15
SAMW25 by Microchip	$53.89

Table 5.6: WiFi Chip Price Comparison

Overall, sensor to app communication requires one of three things. These three things consist of wireless technologies that will make the sensor communicate with the application. Wifi and Bluetooth are for wireless connections. Other options are negated because those would make it wired. Starting off with the bluetooth module. You will need to set up a connection. While this type of connection is universal with the wifi module and xbee modules these are the steps taken for connecting it to the microcontroller of use. While these modules will be connected with resistors to not burn out the module they will be sending out information back through an interface in the middle of the transaction of data.

After looking at all of the data and chips available our group decided to use the ESP32 series from the table shown.

5.5 Microcontroller Chips

A Microcontroller is an essential part of an embedded system and is a compact integrated circuit designed to operate for a specific function within a system. Microcontrollers consist of different elements which all need to be taken into consideration when choosing which microcontroller to use for a task.

The main elements of a MCU unit are the processor (CPU), memory, and I/O peripherals. The processor of a microcontroller is the brain of the unit and processes and responds to various instructions to direct the controller's functions. The memory of a MCU is used to store data received by the processor and respond to instructions that have been programmed into it.

There are two main types of memory for a microcontroller, program memory and data memory. Program memory stores long term information and is non-volatile so power is not needed in order to hold information. This type of memory is also often referred to as FRAM on data sheets for different microcontrollers. Data memory on the other hand is required for temporary data storage while a certain instruction is being executed. Data memory is volatile and loses its memory when it is disconnected from a power source. Data memory is also known as SRAM on microcontroller data sheets.

The last main element are the I/O peripherals which are the input and output pins that allow the interface between the outside world and the processor. The processor receives the data from these I/O peripherals and sends the necessary instructions to output devices that execute the tasks needed in other peripherals.

There are also other supporting elements that are in a microcontroller that are important to take note of and will aid in the Boxing Buddy Robot. Firstly, microcontrollers have the ability to include analog to digital converters (ADC) which are circuits inside the integrated chip that convert analog signals to digital signals. This is important because it allows the processor within the microcontroller to interface with external analog devices, which will be a critical function for the Boxing Buddy since almost all of the data that we need the processor to be able to access will be input into the system through sensors.

Another peripheral supported on microcontrollers are the opposite, digital to analog converters (DAC), which allows the processor to communicate outgoing signals out to external analog components such as sensors. Within the Boxing Buddy there is less of a need to communicate out to the sensors in this way.

The role of the microcontroller in the Boxing Buddy will be to act as a central control unit along with a customized PCB in order to control the force sensors, LED lights, reaction time measurements, and IoT communication interface between the controller and the mobile application. When selecting a microcontroller there are many things to consider in order to get the best performance and economy efficiency from the decision.

The most important factors that the design should consider are core size, peripherals required, speed, power consumption required, flash memory, cost of ownership, and support ecosystems. When selecting the best microcontroller it is also important to optimize the power consumption by looking at low-power states and which peripherals and interrupts being used are active.

These are the steps used in order to select the microcontroller for the Boxing Buddy:

1. Listing all external interfaces the microcontroller will need to support
    a. In this case it will be AC/DC Conversion, Feedback from two Force Sensors, Feedback from two signal LED's, DC power, and either a Wifi module or bluetooth module.

2. Examine Software Architecture
    a. Looking specifically at requirements needed in the software based on how heavy the processing will be, and how long and how often the task will need to be run for the sensors in the system. In the case of the Boxing Buddy the software architecture will need to be able to support short bursts of a lot of input data coming in when the sensors are being used during an hour-long workout period. This in turn will indicate the amount of computing power needed for the architecture which is the biggest requirement to take into consideration.

3. Figure out the number of bit architecture needed
    a. The options for this choice typically range from 8, 16, or 32 bits needed across the different microcontrollers.

4. Identify Memory Needs
    a. In order to do this we look at the software architecture and communication peripherals involved in the application to estimate how much flash and RAM will be required for the application. A good rule of thumb within this step is to always leave space inside a bigger memory so there is room to grow the application if needed.

Below is a summary of some of the top microcontrollers being used today that :

Comparison of Microcontroller chips:

Chip Name	PIC16F886-I/SO	PIC32MX250F128D-I/PT	ATMEGA328P-MU	STM32F103C86T5
Manufacturer	Microchip	Microchip	Atmel	ST Microelectronics
Bit Size	8-bit	32-bit	8-bit	32-bit
Memory	256 bytes EEPROM	128 kB addressable program memory size, 3kB auxiliary flash, 32769 bytes data memory size	32 KB ISP flash memory, 1024B EEPROM, 2KB SRAM	Flash memory up to 128 Kbytes and SRAM up to 20 Kbytes
Architecture Type	PIC	-	AVR RISC	RISC
Comparators	2	3	-	-
Number of Channels	11-channel 10 bit A/D	13-channel 10 bit A/D	8- channel 10 bit A/D	2 12-bit ADCs
Synchronous Serial Port Communication	3-wire SPI, 2 wire I2C bus, EUSART	-	Byte-oriented 2-wire serial interface;SPI serial port	Up to 2 I2Cs and SPIs, 3 USARTs, USB, and a CAN
Price	$4.87	$3.78	$2.41	$4.07
Additional information:	-	Max frequency: 50mHz; operating voltage 2.3-3.6 V	-	Operating at 72 MHz frequency

Table 5.7: Summary of Top Microcontrollers


Figure 5.13: Images of Microcontroller chips

Analyzing the above microcontroller chips, our team decided to look more closely into specifically Texas Instrument components because of choices within other parts and trying to keep the producer consistent throughout the project. Two chips in particular that took our interest were the MSP430FR6989 and the TMS320F2800. Summaries of both parts are included in the table below:

Chip Name	MSP430FR6989	TMS320F2800
Memory - FRAM	128 KB	24 KB
Memory - SRAM	2 KB	-
ADC	12 bit 12 external input channels	2 12 bit
I/O	63 pins	-
Timers	Timer_A, Timer_B	-

Table 5.8: Summary of Texas Instruments Development Board

After looking at the components on both boards, our team initially thought we would move forward with the MSP430FR6989 chip because it has the capabilities needed and in addition our team has worked with the unit in the past for different projects giving us a better understanding of how to implement the features to our advantage. The choice of the MSP430FR6989 will also be helpful during testing of the component because it has a specially designed development kit that will allow our team to test quickly and efficiently before completely creating our printed circuit board, more details of this will be covered in the testing section.

Table 3-1 and Table 3-2 summarize the available family members.

Table 3-1. Device Comparison (With UART BSL)(1) (2)

DEVICE	FRAM (KB)	SRAM (KB)	CLOCK SYSTEM	Timer_A (3)	Timer_B (4)	eUSCI A(5)	eUSCI B(6)	AES	ADC12_B	LCD_C	I/O	PACKAGE
MSP430FR6989	128	2	DCO HFXT LFXT	3, 3 (7) 2, 5 (8)	7	2	2	yes	12 ext 16 ext	240 seg 320 seg	63 83	80 PN 100 PZ

Figure 5.14: MSP430FR6989 data sheet

However, during testing due to compatibility and inabilities to find other TI parts that worked well with the MSP430FR6989 due to extremely long lead times for various other components including wifi ship modules, our group decided to instead move forward with the Mega 2560 microcontroller chip instead. Pictured below is a picture of the Arduino Mega development board that will be utilized in the initial testing phase of the microcontroller.

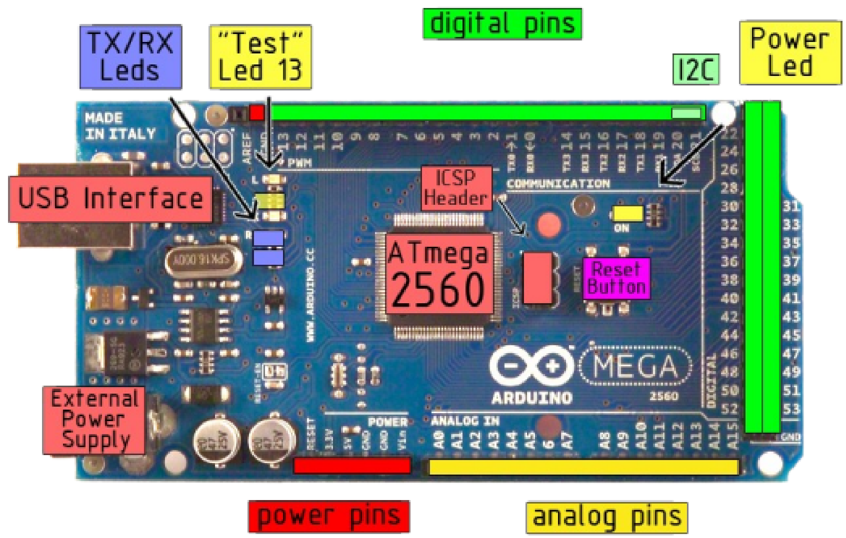


Figure 5.15: Mega 2560 MCU development board

The table below also summarizes the technical specifications of the Mega 2560 chip and how we will utilize the parts within our own design:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Table 5.9: Technical Specifications of Mega 2560 microcontroller

5.6 PCB

A printed circuit board mechanically supports and electrically connects components using conductive tracks, pads, and other features etched from copper sheets laminated onto a non-conductive substrate. In order to meet the requirements for Senior Design the printed circuit board designed for this project must have a significant amount of design work from the students. In order to be considered a significant amount of design work the printed circuit board for this project must contain an MCU chip on the board, that was tested through a development board, as well as be integrated with power, sensor inputs, and output controls. Due to these requirements the printed circuit board for the Boxing Buddy robot will contain the following parts:

Component:	Purpose on Board
MSP430FR6989 MCU Chip	- Process data input into the system and output to the user - Act as an ADC converter between analog input from sensors to processor
DC-DC Voltage Regulator	- Step down 5V DC voltage from MIcrocontroller unit to 3.3V needed for the sensors/LED input/output peripherals
Power Supply Connection to Outlet	- Convert 60 Hz AC current from wall outlet to a DC current that can be input into the microcontroller processor
IoT Module	- Act as a Wifi chip to send data from input/output peripherals to the cloud that will then be accessed through the app interface
Sensor Input/Output	- GPIO pins will act as a direct connect transfer of the force sensors, LED lights, and other modules on the board other than the microcontroller chip

Table 5.10: Board Component Summary

an MSP430FR6989 Microcontroller ChipThe PCB design for the Boxing Buddy will feature a microcontroller chip mentioned in the above microcontroller chip section as well as an IoT communication module of some sort either being a bluetooth classic, bluetooth low energy, or wifi module. Currently the exact design of the PCB is still under progress, however it will be laid out and routed on EAGLE once more design decisions are shown.

Potential Vendors to get the PCB printed will be discussed below instead.

5.6.1 DC-DC Voltage Regulator

The voltage of the power supply or even two different components connected on a printed circuit board rarely all operate at the same DC voltage needed for the different circuits. Because of that, within this project we will be utilizing voltage regulators. Utilizing regulators not only helps step up and step down the voltage, but also helps to regulate the load and line of the system to ensure that if there is a fluctuation in the amount of current being drawn in the circuit or a change in the input voltage of the power supply the output voltage will not change.

The two major types of regulators that have these functions are linear regulators and switching regulators.

Linear regulators work by operating like a transistor in the linear region through using a variable resistor. The circuit then senses the output voltage that is being produced and automatically changes the variable resistors value in order to make the output voltage remain constant.

Advantages:	Disadvantages:
Cheap & Easy	Very Inefficient
Quiet	Low Current Only
Small	Limited Vin and Vout ranges

Table 5.11: Advantages and Disadvantages of Linear Regulators

Switching Regulators operate as power converters by turning the power supply on and off to get the desired output voltage instead of using the variable resistor as is used in a linear regulator. Because of that and the disadvantages of linear regulators such as being inefficient our team decided to use a DC DC Switching Regulator in our design.

Chosen Regulator: TPS6286920CRQYR

	Buck Switching Regulator IC
Manufacturer:	Texas Instruments
Function:	Step-Down
Topology:	Buck
Voltage - Input (Min - Max)	2.4 V - 5.5 V
Voltage - Output (Min-Max)	08 V - 3.35 V
Frequency - Switching	2.4 MHz
Mounting Type	Surface Mount
Price	$2.81

Table 5.12: Summary of Buck Switching Regulator IC Specifications

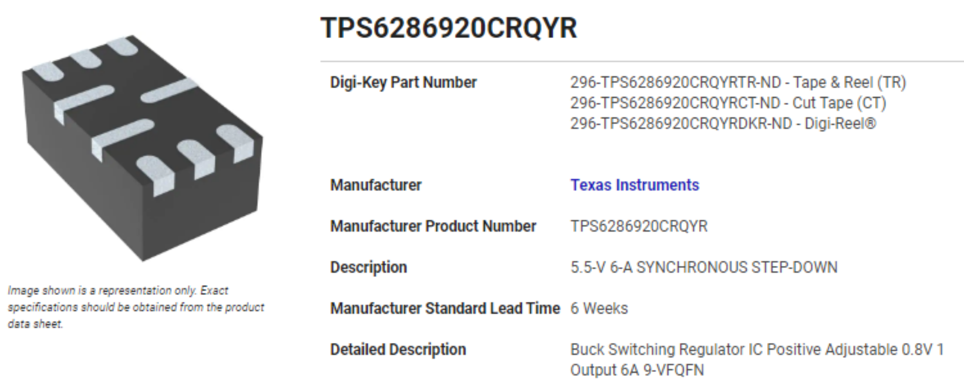


Figure 5.16: Chosen Switching Regulator

5.7 LEDS

The main role of the LED’s in the Boxing Buddy will be to act as an indicator above the strike pads that will turn off and on when sent a signal from the mobile application. LED’s are at the most basic form semiconductor devices that turn electricity into light. There are three main types of LED technology that are used in LED lighting to compare when looking at which type to choose.

First there are Dual In-Line Package LEDs which are traditionally small lights that are normally built into electronics and different boards due to their small size.

Dual In-Line package LEDs are not very powerful and can only emit limited amounts of brightness.

Figure 5.17: DIP LED Lights

Next there are Surface Mounted Diode LEDS which are the most common type of LED Chip available. They are normally soldered onto the circuit board themselves and are much brighter that the traditional DIP LED light while still being small in size. THey are more versatile in their size and shape and can be put into different encasings which is beneficial to using them in different applications. It is also possible to put up to three different colored diodes on one Surface Mounted Diode LED at a time allowing for a range of colors.

Common SMD Chip	Size
SMD 3528	3.5 mm wide
SMD 5050	5 mm wide

Table 5.13: Comparison of Surface Mounted Diode LEDs

Figure 5.18: Surface Mounted Diode LED Bulb

The last main type of LED are the Chip on Board LEDs which are the newest development in LED technology. Chip on Board LEDs are the brightest LEDs created and have the ability to put up to nine diodes within one singular chip allowing even greater variety in color changes than the surface mounted diode LEDs.

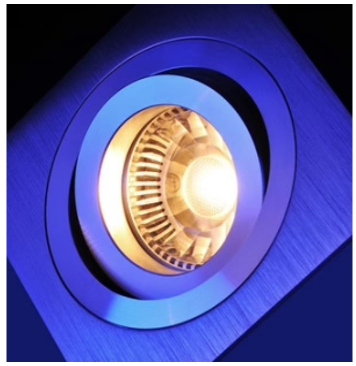


Figure 5.19: Chip on Board LED Spotlight Bulb

Looking at the three types of LEDs available, our team decided to choose a type of DIP LEDs as the indicators above our strike pads with built in resistors installed into the LED's already. This decision was made because they are the most cost efficient option while also providing variety in color, brightness, and size. Although the traditional DIP LED lights are not very large, our plan is to use multiple of the DIP LED's and enclose them in a casing with a cap that will allow a bright light to capture the attention of the user of the boxing robot simulation.

Summary of Parts Bought	
Size:	3 mm
Voltage:	3-5 V
Wattage:	20 W

Table 5.14: Summary of Selected LEDs

Advantages:	Low power consumption, stable performance, high performance, 20 cm long flexible cable, built in resistor
Number included in pack:	12
Price of pack:	$9.99
Price per LED:	$0.83

Table 5.15: Advantages of Selected LEDs

Figure 5.20: Chosen LED’s

5.8 Expectations of the Mobile Application

In its most basic form, the mobile application must obtain force data and reaction time data from the sensors and microcontroller unit and display them to the user. The application will also have the ability to store previous records and show the user their progress. In addition, the app must control the starting and stopping of different workouts. For example, the app will indicate that the workout is starting and then the Boxing Buddy setup will have to start a countdown for the LEDs and reaction time data. In short, the app acts as a log for the user as well as a remote control for the physical system.

When practicing boxing at home, the user should have options of several different practice modes. There may be one instance where the user wants to focus on how quickly they can react to their environment. This would be useful for when they are in a match and need to react quickly to their opponent. The user may also want to instead focus on hitting harder. This will allow each hit to be more effective in a real boxing match. Additionally, the user may instead want to focus on speed in terms of striking as many times as possible. This will allow the boxer to get more strikes on their opponent than the amount of strikes their opponent can get on them. When using the Boxing Buddy, it is important that the user is able to practice these three elements and hone these skills. Thus, there will be three different types of workouts available for selection by the user.

The first type of workout is the Reaction Time workout. When this workout is selected, a delay of five seconds will occur, then after a random amount of time, an LED on one of the strike pads will light up indicating the area that needs to be hit. This will also be when the timer starts. Once the LED is on, the user will strike the indicated strike pad and the timer will end. The elapsed time and the force read from the strike will be sent to the mobile app using a WiFi module.

Another type of workout that is selectable is the Force workout. In this workout option, the user will indicate in the app which strike pad to use. Once indicated, the LED for that strike pad will light up. At this time, the user will strike the pad and the force sensor will send that data to the app. In this mode, only force is being read and transmitted and no timer data will be collected.

The third type of workout is the Repetitive Hits workout. For this workout selection, the user will select the Repetitive Hits option and a duration.This duration is user definable. The app will send that information to the microcontroller unit along with which pad(s): left, right, or both will be used in the workout. After a five second delay upon selection, the LEDs corresponding to the selected strike pads will light up. That will indicate the start of the timer of length equal to the duration defined by the user. Within that time, the user will strike the pad or pads as many times as possible before the timer ends. The number of strikes by the user will be sent to the app and recorded.

An essential part of a fitness regimen is to be able to track progress. According to the American Psychological Association, the more progress is monitored, the greater the likelihood of success. In order for the Boxing Buddy to be a successful fitness app that helps people improve their boxing skills, it is crucial for the mobile app to include a log section that displays previous workouts that the user can easily view and track. This will help the user keep their goals in mind and have more drive to surpass their previous records.

On the homepage of the mobile application, there will be a button for “Previous Workouts.” Upon selection of that option, the user will be navigated to a page with a list of their previous workouts on that account. The list will be in order of most recent workouts to oldest workouts. Each workout will indicate which type of workout and the most relevant information from that workout. For the Reaction Time workout, the information displayed will be the type of workout, the date of the workout, the force with which the strike pad was hit, and most importantly, the elapsed time between the LED indication and the strike (i.e. the reaction time). For the Force workout, the information displayed will be the type of workout, the date of the workout, and the force with which the strike pad was hit. For the Repetitive Hits workout, the information displayed will be the type of workout, the date of the workout, and the number of times the strike pad(s) was(were) hit in the duration defined by the user, and the duration chosen by the user.

The idea is to make the Boxing Buddy an effective tool for making the user a better boxer. The three different workouts and the historical workouts page are the foundation the user needs in order to become a more successful boxer.

5.9 Types of Mobile Applications

There are four different paths available when building a mobile application from scratch. The first option is to build a native app. This means that the app will only work in one type of operating system, that is, ios or Android operating systems. Another option is to create a hybrid app which allows the app to be used cross-platform on different operating systems. A simple option is another type of hybrid application commonly referred to as “drag and drop” which allows the developer to create an app using templates and very little programming. Each of these options has advantages and disadvantages. The last type of mobile application available is a web application. Web based mobile applications are accessed via the browser and do not need to be downloaded to be accessed.

The disadvantages to a native app are clear. They are less flexible and less accessible than hybrid apps. Only one sector of the market may use the app if only one native app operating system is chosen. If the desired outcome is an app that runs on Android and ios, then two different code banks are necessary.

Native apps, however, do have a few key advantages. First and foremost, they tend to be much more user friendly. The interface for a native app will match the usual interface for applications on that operating system making it much easier for a user to learn and use the app. Furthermore, the performance of a native app is much better than a hybrid app. They run, load, and execute faster due to not needing a middle layer between the app and the platform. In addition to user friendliness and performance, native apps also have better security. Native apps use code that is compiled into an assembly language which is generally hidden. Hybrid apps tend to use HTML which can more easily be exploited by attackers. Native apps will also have access to more features since the operating system is designed for that phone. For example, apps that require the camera or microphone would have a much easier time creating a native app.

Hybrid apps have several advantages over native apps as well. They are much more accessible because Android and iPhone users will be able to use this app. Another advantage is that they use one codebase with well known languages like HTML, CSS, and JavaScript. This makes development much faster because they do not have native operating system requirements.

Unfortunately, the tradeoff is performance. They can be slower and require internet connection. There is an extra layer between the app and the operating system that needs to translate the code into the proper format for the operating system. This slows things down. Also, hybrid apps do not have access to native features of an operating system like the camera or face ID.

The third option with very little programming is the “drag and drop” method. This makes them very limited. Most drag and drop app development environments are created for specific business niches. Some features may not be available in the development environment which can be very limiting. The major benefit is the fact that there is no coding, and they are quick to build. The tradeoff of significant limitations may be too steep for some projects.

The final type of mobile application is the web application. These are viewed in the browser instead of downloaded from the app store. One advantage to this is that the application will be accessible to more people like a hybrid mobile application. Any device will be able to access them. There are additional advantages to a web application though. Since a progressive web application is only on the internet and does not use the app store, the developers would not have to have anything approved by the Google Play store or the Apple App store. This can save time during development. Also, web applications can be updated easily and can be very flexible when accommodating different devices and screens.

One major drawback to the web application is that it loses a native feel. Since it is not a traditional mobile application from the app store, users are less likely to enjoy using it and will most likely not be as familiar with how to operate it. Furthermore, within the app, the layout and structure will be different from the native applications and more closely resemble a website. Also, not having the Google Play Store and Apple App Store standards can be a disadvantage as well. These standards uphold a secure application platform that is designed to be efficient and secure for the Android and Apple users respectively. While it may decrease the development time because of no need for approval, it lacks user friendliness and engagement with the user. Progressive web applications show a lot of promise, but they are still in development. For example, Apple does not allow certain functionality in progressive web applications like the Face ID, bluetooth, or sending notifications. It is possible that the Boxing Buddy application may need to use bluetooth or be able to send notifications.

There are advantages and disadvantages to each of these mobile application types. Each type can be applicable to different use cases. For the Boxing Buddy, it is necessary for the team to choose a mobile application type that matches the specified goals and expectations.

There are several goals and requirements of the Boxing Buddy that are necessary to keep in mind when selecting a mobile application type. When it comes to the application, user friendliness is a primary goal. The ease of use of the application must be considered in the application type. Additionally, speed of development is a constraint that will limit which type of application is chosen. The entire project must be completed in four months, so within that time, a mobile application must be completed. Another requirement to consider is speed and efficiency of the application. If the application takes too long to start up or load different pages, then it may be very inconvenient for the customer to use.

Type of Application	Main Advantage	Main Disadvantage
Native Application	More user friendly and gives increased access to OS and platform specific tools on a device.	Lower accessibility
Hybrid Application	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 non native feel.

Table 5.16: Advantages and Disadvantages to Different Application Types

Based on the information summarized above in Table 5.16, the “Drag and Drop” mobile application type can quickly be ruled out. This is due to the very limited complexity and customization needed for the Boxing Buddy mobile application. This application will require an API and database to store information from the physical system. The main debate is between the native application or the hybrid application. Because of the goal of accessibility, the hybrid application is very enticing. If the mobile application for the Boxing Buddy required tools such as Face ID, the Camera, or microphone for example, then the native application would be a better fit because of the access to operating system specific tools.The Boxing Buddy will not need these tools so the best option moving forward is to create a hybrid application or a web application. There are certain frameworks when paired with creating a hybrid application that can make the mobile application feel more native to the user. This is an advantage to the web application which will never really be that user friendly. The chosen mobile application type is the hybrid application. It has the benefit of being pretty user friendly only being beaten by the native application and a high level of functionality when paired with an appropriate framework. The next step in development is to choose a language to program in and a platform to use for development. Before those are chosen, it is necessary to outline what exactly the mobile application will need to do.

5.10 Programming Languages

The Boxing Buddy mobile application will not need to access any advanced features specific to a phone like the camera, microphone, Face ID, Touch ID, etc. This means it is not completely necessary to create a native app because native apps specific to an operating system have an easier time accessing operating system specific tools. Furthermore, in order for the application to be accessible to the widest range of people, a hybrid app should be used. That way, people with the android operating system can access this app via the Google Play Store and those with ios can access this app with the Apple App Store. This aligns most closely with the goal of accessibility. There are several different platforms and frameworks for creating a hybrid app each with their own structure. Each method will often use different programming languages.

There are many programming languages that exist, each with their own advantages and disadvantages. For the Boxing Buddy product, it is necessary to choose a language that will perform well in a mobile app setting. Given the skill level of the programmers, the language or languages chosen should be fairly simple to use and understand. The languages should also be efficient and quick to make the creation of the mobile as streamlined as possible while keeping the performance of the product as high as possible. It is entirely possible that multiple languages will be needed for the Boxing Buddy mobile application. This is likely to be true because in mobile app development and web development, there is such a thing as back-end development and front-end development. Front-end development is also called client-side development. This refers to the user interface mainly. How the user interacts with the mobile app from its visuals to ease of navigation fall under this category. Back-end development is also called server-side development.This refers to data and database management in an application. This is necessary for almost all applications. For the Boxing Buddy project, the mobile application will need to acquire information from the WiFi module and store this somewhere. The information to be gathered is the force and reaction time data and all historical workout data including time and date data. The following programming languages are discussed in terms of usefulness for the front-end development, that is, the main user interface and framework.

5.10.1 The C Language

One language that could be chosen is the C language. C is widely considered the father of modern programming languages. The reason for this is that most programming languages like Python, C++, C#, and Java are all based on C structure. It is older than these languages. This means that there is a lot of documentation and aid to be found for using C to do varying tasks.

Many libraries exist for C meaning most common tasks have been completed by someone else and are often shared publicly. This would speed up the process of development due to fewer pieces of the mobile app needing to be programmed from scratch. Even though C has many libraries, it is still rather old which means many functions that are built into other languages like Java and Python must be done manually by the programmer. This can slow down the development time.

C is also not a language designed for a specific task, so it has some versatility to it. Since its development, C has become a language that is used for a more specific reason. More advanced and versatile languages exist now, so it has become a language often used as a bridge between hardware and firmware. In other words, it is primarily used to control hardware functionality like for a microcontroller. Due to C being rather simple, it uses very few instructions which means its execution speed is fairly high. Furthermore, the C compiler generates and processes machine code very fast which also increases its speed. C may have a quick processing speed, but its development speed is slow due to creating many functions from scratch. Lastly, the most significant drawback to the C language is the fact that it is not object oriented.This means that the principles of abstraction, encapsulation, inheritance, and polymorphism present in all object oriented languages cannot be found in C. Making a mobile app will be much more streamlined and clear with the use of objects and classes. There are several object oriented languages that have developed directly from C which offer other advantages that C cannot offer.

5.10.2 C++

C++ is one of the updated, object oriented renditions of the C language. C++ offers many similar advantages to C because they are very similar languages, with the added advantage of being object oriented, so it gains access to the object oriented principles.

C++ is still often used as a bridge between hardware and firmware. A major difference is that classes are made first then combined to form more complicated programs. As opposed to C which is limited to user made functions. In addition to classes, C++ is newer and offers more built in functions than C. This means a quicker development time. C++ is also written very similarly to C which means it does not take much time to learn if the programmer already knows C. C++ shares the advantages of C’s speed in compiling and processing while also speeding up the time for development with the added built-in functionality. Even with all these advantages, C++ is not generally used for websites or mobile applications. This means finding an environment that uses C++ to make a hybrid app could prove very difficult. Another option is to write in C++ and have a program convert the code automatically. This takes a long time and is prone to bugs in the conversion process.

5.10.3 C#

C# is the other object oriented version of the C language. Because it is directly derived from C, it is simple to learn and shares many advantages. These include the speed associated with the compiler and processing as described in the section about the C language as well as the increased number of built in functions like discussed in the section about C++. C# also has access to the creation of classes and objects like C++ due to it being an object oriented language.

Microsoft developed C# as their approach to a multipurpose programming language for their .NET ecosystem. The idea was to create a language designed for web based applications. C# works very well for this use especially for Windows Web applications. Developers have branched out and used C# for other reasons like game development, server side programming, and mobile applications.

Because C# can be used for mobile app development, there are environments that use C# like Xamarin. Xamarin is a cross platform framework that can be used to create hybrid mobile applications accessible on Android or ios. Developers use their PC or Mac to write Xamarin applications. These applications are written in .NET with C#. The developers then compile them into native form of the application, either .apk for Android or .jpa ios applications. In order for C# to be this versatile, it relies on other supporting frameworks like Xamarin and the microsoft .NET ecosystem for web development. This further complicates the development environment making it more difficult to learn to program in C# as well as slowing down the execution of programs written in C#. On its own, C# is pretty narrowly focused on web development for Microsoft.

Its lack of versatility is a major handicap that keeps it from being the most efficient choice. Additional framework other than .NET in order for C# to be used on Linux and MacOS. C# can be compared to Java because both are primarily used for web development. C# is slower than the other C based languages aforementioned as well as slower than Java. It is also significantly less versatile and widely applicable than Java.

5.10.4 Java

Java is another language that uses a structure similar to C. It is object oriented and has been around for a long time. It has been the language of choice for network applications by many top companies and developers for its versatility and built in capabilities. Java is relatively easy to learn because it is based on C, has fantastic documentation, and has many libraries and built in functions that make development smoother and quicker.

Java is a secure and robust language that is primarily used for web applications. It is also rather verbose, requiring many lines of code for a seemingly simple task. Compared to C and C++, it is much slower because Java needs an interpreter to convert the high level language into machine code. Consequently, this means Java has a high memory consumption. This is similar to how C# works. Java can outperform C# and its use and versatility more than make up for its lacking speed. When developing mobile applications, Java is often used for Android development.The Boxing Buddy will ideally be cross platform. Like other languages, a framework is necessary when creating a hybrid application. Frameworks for hybrid mobile applications that use Java are not that common. Most use one or more of the following: C, C++, C#, or Javascript.

5.10.5 Python

Python is a very beginner friendly programming language that offers very simple and easy to use syntax. It is not C based, so if C is the only experience a developer has, then Python will take some time to get used to.

Python is an open sourced language that is pretty new. It offers a huge range of libraries and built-in functions for a variety of uses. Because of its versatility, Python can pretty much be used for any type of development. In more recent times, Python has found a particular affinity for machine learning and AI applications.

On the negative side, Python is not compiled like other languages, so it tends to be slower than compiled languages which all the aforementioned languages are. Additionally, unlike Java, Python is not nearly as secure. Additionally measures must be taken to ensure security which add to the complexity and development time. Furthermore, as with the other examples, a framework is necessary to develop a mobile application in Python. There are many framework options for Python but not too many good options to develop hybrid applications. There are options like Kivy which is for hybrid applications but only supports Python 2.7 for ios applications. The other framework option is BeeWare. This means the developer is completely dependent on BeeWare to update its repository frequently in order for the software to be up to date and compatible with the Google Play Store and Apple Store.

5.10.6 Javascript

Javascript is the world's most used language for front-end web development. This includes webpage interactions, pop-ups, alerts, events, etc. Javascript is the language that performs more complex functions on a webpage while HTML is usually meant for the user interface.

Javascript is also very common in hybrid mobile application development. There are a variety of frameworks that use Javascript as the main development language. In addition to its wide use, Javascript also boasts being able to be used for front-end and back-end development. In many other cases, a different language would need to be learned and used for the interface portion, the front-end, and the data storage and package handler portion, the back-end.

Javascript is written pretty differently than the other languages discussed above. Without prior knowledge, this may present a steep learning curve. Another disadvantage is that javascript only allows single inheritance. Inheritance is one of the object oriented programming principles. It is when one class inherits attributes from another class. In Javascript, a class can only inherit from one other class while in other languages like Java, a class can inherit from multiple classes effectively expanding code reuse. This can pose a potential problem for certain cases in development and may result in a longer development time when compared to other object oriented languages..

5.10.7 Swift

Swift is a relatively new language specifically designed for mobile apps. It is a very powerful and efficient language that is general purpose, meaning it can be used for a variety of application needs.

Swift was created by Apple in 2014 and was meant to replace Objective-C. It is compatible with Objective-C. The main drawback to this language is that it is only used for the ios devices. The Android operating system can use Java as aforementioned. Swift is a powerful ios tool that offers great customization and complexity, but will increase development time because a separate front-end will need to be created for both the Android operating system and the ios devices.

5.10.8 Programming Languages Summary

There are a few different ways to summarize the information in this section. The clearest summarization would be to enumerate the intended use for each language or primary user for each language. This will show what tools are most readily available for that language. Additionally, for each language, there should be a main advantage and main disadvantage shown to summarize the detailed descriptions above.

Language	Main Advantage	Main Disadvantage	Intent for Use
C	Easy to learn and execution speed	Not object oriented and not supported by frameworks for mobile app development	Firmware
C++	Easy to learn and object oriented version of C	Not supported by frameworks for mobile app development	Firmware
C#	Object oriented and supported by frameworks for mobile app development	Lack of versatility and need for other products (.NET)	Server-side programming and web development
Java	Ease of use, security, and versatility	Not much support for hybrid mobile app development	Web development
Python	Very easy to use	Not much support for hybrid mobile app development	General/All purpose
Javascript	Lots of support for hybrid mobile application development	Difficult to learn and single inheritance	Web development
Swift	Can be used to create very customizable ios mobile applications	Can only be used for ios devices so a different front-end would be necessary for Android OS.	ios mobile app development

Table 5.17 Summary of Front-End Language Advantages and Disadvantages

Based on the information summarized in Table 5.17, Swift can easily be ruled out. One of the major constraints for this project is time. There is not enough time in two semesters to develop separate mobile applications for ios devices and Android devices. Since Swift is only able to be used for iOS devices, a separate application will need to be made for Android devices if the goal of accessibility is to be maintained. There are several other languages discussed for the front-end development that have this same pitfall. These languages are Java and Python. While both of those languages offer enticing advantages, the drawback of little support for hybrid mobile application development is too great. The major constraint here is complexity. It would be too hard to create a hybrid mobile application with Java or Python because it will take too much time to learn how to do so and require too much code from scratch which will further lengthen the development time.

C and C++ are enticing because they are fast in execution time and are easy to learn with experience in C. For the Boxing Buddy mobile application, they are less than ideal. This is because they are primarily used for firmware and control. This means there are not any frameworks that support C and C++. The constraint here is again time. It will take too much time to learn how to create a hybrid application using these languages. They will also require too much code from scratch by the developer. This is prone to errors and will lengthen the development time with excess debugging.

The last two languages are C# and Javascript for the front-end development. C# is significantly less versatile than Javascript and requires more additional framework which can increase the price of development when buying licenses. Javascript has lots of supported frameworks for hybrid mobile application development and can even be used for back-end development in some cases. This can save a lot of time because fewer frameworks will be needed to learn. This can also save money because fewer products will need to be purchased for the app development. In conclusion, the best language for front-end development after weighing the advantages and disadvantages summarized in Table 5.17 is Javascript. Javascript has the most support and fits all of the needs of the Boxing Buddy mobile application without violating any constraints.

5.11 Framework Options

A framework is a programming tool that provides a basic structure and some built-in, useful mechanisms for a specific type of use. There are frameworks for web development, native application development, hybrid application development, machine learning, and many more. The Boxing Buddy mobile application requires a framework made for hybrid mobile application development that is not too complicated to use or learn and a framework that uses one of the better programming languages for hybrid mobile application development discussed in the previous section. The following sections will contain details on the different advantages and disadvantages of several different frameworks designed for hybrid mobile application development.

5.11.1 Ionic

Ionic is an open-sourced, web-based framework created in 2013. Ionic is a framework which uses web based technology to develop a user interface. Ionic incorporates HTML, CSS, and Javascript. All these components make Ionic what is called a Progressive Web Application. By definition, a progressive web application uses web services to create a mobile app that is cross platform for both Android and ios devices.

Due to Ionic being a progressive web application, it is often used as a framework for applications that exist in the online web space as well as on mobile devices. Some examples of mobile applications considered to be progressive web applications are the Uber app, the Starbucks App, and Spotify. Many more exist as well, and most have websites associated with them.

The advantages and disadvantages of the Ionic framework center around the fact that this framework is web-based. Being a web-based framework gives the developer access to a very wide range of libraries that can be extremely useful in application development and can provide exceptional customization tools. The drawback to being a web-based framework is that the mobile application can lose a native feel to it. The web-based libraries and tools will not be able to access native platform tools that make the user interface more natural to the user. Another disadvantage of a web-based framework like Ionic is that it tends to be slower than other options.

5.11.2 React Native

React Native is an open-sourced framework released in 2015 by Facebook for the purpose of creating hybrid mobile applications. React Native is not a web-based framework and instead uses React as a tool to control native elements directly. This means that hybrid mobile applications created with React Native will have a feel and look to them that more closely resembles a native application.

React Native uses primarily Javascript to develop the front-end. This Javascript is translated to native code and can achieve a frame rate of sixty frames per second. An additional benefit to having a framework that is not web-based is that in some instances, performance can be better. In a study that compared Ionic to React Native, there was not very much difference in load time for a cold and hot start of an Ionic application and a React Native Application. A cold start is when nothing is cached and the application is opened and loaded. A hot start is when the application had previously been opened and loaded and the application was opened subsequently. More complicated features may prove to be executed faster on a React Native application.

5.11.3 Xamarin

Xamarin is an open-source framework created in 2016 by Microsoft. The purpose of Xamarin is to create hybrid mobile applications. This framework support the .NET framework and uses the C# language for development. The native libraries Xamarin uses are from the .NET framework

Xamarin acts as a buffer layer between the shared code and the platform code. The shared code is the code written in this case in C# that can be used by both the Android operating system and the ios devices. Platform code is the code that is specific to the Android operating system or the ios devices. Due to this structure, Xamarin requires some knowledge of the platform specific code. More likely the developer will need to know Swift when developing because the ios devices are much more different than the Android and Microsoft devices which have similar operating systems. Thus, Xamarin requires more learning than the aforementioned frameworks leading to an increase in the development time. Additionally, using Xamarin may not give as much of a native feel for the user. One of the main goals is user friendliness.

This framework does, however, have the advantage of being a bit newer and having the Xamarin Component Store which has a lot of useful libraries and tools. While the main Xamarin framework is open-sourced, free in other words, the majority of the useful libraries and tools are trapped in the Xamarin Component Store which requires a paid subscription. The free version has application size restrictions and does not allow the use of third party libraries. This could prove to be a major restriction for the Boxing Buddy application. The first tier of subscription to the Xamarin Component Store releases these specific restrictions but comes in at $25 per month.

5.11.4 Flutter

Flutter is a framework owned by Google that was created in 2017. Flutter is a great framework to use when development time needs to be short. The Boxing Buddy has the constraint of needing to be completed by April of 2023. For experienced application developers, Flutter can be a very efficient framework with lots of builder tools and mechanisms that allow you to directly change the application when fixing bugs. However, Flutter uses the Dart language which has become a popular language to learn, but it is unique to Flutter. None of the developers on the Boxing Buddy team know Dart or are even familiar with it which can make development slower. The speed and efficiency of the Flutter framework is counteracted by the rate at which Dart will need to be learned.

There are additional, more unique advantages to using Flutter however. While its newness may mean there is not a large resource base and a lot of code will need to be written from scratch, it also allows some newer technology that can increase performance. This is due to the fact that Flutter does not need a Javascript bridge. Flutter also lets you control very specific details about the application down to the individual pixel level. This means Flutter has high customization capabilities that can be used to create rather complex user interfaces. The downside to this is that Flutter mobile applications tend to use a lot of memory which many users may find unappealing.

5.11.5 Aurelia

The Aurelia framework was created in 2015. It uses Javascript and HTML for development which means not a lot of learning of new languages needs to take place. This framework is open source as well but it has rather limited documentation. Since the Boxing Buddy team has not ever built an application, this could be a major drawback that makes debugging difficult. Aurelia does, however, have specific tools that are specialized to help in the testing process. There are widgets that make unit testing and integration testing much more streamlined than other options.

5.11.6 Angular

Angular is another open-source framework made by Google. It is a Javascript based framework that uses HTML templates and has support for AJAX. This is appealing as these are languages that the developers have experience in and are very well documented. This means the learning curve is reduced and there will likely be fewer bugs to catch than some other more unique or complex options. Angular is a high-performance framework that tends to work better for larger, more complex mobile applications at the expense of simplicity. The benefit of simplicity here is two fold. First, it makes the application more user friendly which is more enticing to potential customers. Second, it makes testing and debugging easier for the developer. These two areas are lacking in Angular when compared to frameworks such as Flutter and React Native.

5.11.7 Summary of Frameworks

There are a few different ways to summarize the information in this section. The clearest summarization would be to enumerate the intended programming languages for each framework. This will show what tools are most readily available for that framework. Additionally, for each framework, there should be a main advantage and main usage shown to summarize the detailed descriptions above.

Framework	Main languages used	Cost	Main Use and Benefits
Ionic	Javascript, CSS, HTML	$50 per month	Primarily used for web-based applications that exist both on mobile devices and on the internet
React Native	Javascript	Free	Hybrid mobile application development with great user friendliness from native tools
Xamarin	C#, .NET, and Swift	$25 per month	Hybrid mobile application development
Flutter	Dart	Free	Hybrid mobile application with higher degrees of control and precision
Aurelia	Javascript and HTML	Free	Hybrid mobile application development with high security.
Angular	Javascript and HTML	Free	Larger more complex hybrid mobile applications

Table 5.18: Summary table for Frameworks

Based on the attribute rankings of the frameworks summarized in Table 5.18, the best framework for the Boxing Buddy can be deduced. The constraint of time to finish the project means choosing a framework that has a low learning curve is paramount. Frameworks that use unique languages like Xamarin and Flutter which use Swift and Dart respectively will take too long to learn for the developers and thus not be chosen. Another important constraint is cost. There are free, open-source options as well as subscription options that can be used for the Boxing Buddy mobile application. Because the cost of the Boxing Buddy is being paid out-of-pocket by students, the free options are much more appealing. Since Xamarin was already ruled out, this constraint only rules out Ionic.

The options remaining include React Native, Aurelia, and Angular. The easiest to eliminate out of these is Aurelia because it is designed for larger more complex mobile applications which the Boxing Buddy application will not be. Moreover, it has a smaller user base which means support in terms of usable libraries and debugging help is more limited. This again could violate the time constraint if debugging takes longer and more code has to be written from scratch. The latter reason of a smaller user base is also the reason to eliminate Angular. In comparison to React Native, there is far less support and available libraries. In addition to the ample support, React Native also prioritizes user friendliness. This is a major goal of the mobile application. Users are much more likely to have a better reaction to an application that's easy to use. React Native uses libraries designed to manipulate the native looking environment. When building a hybrid mobile application, React Native is the closest a framework can be to a native application.

5.12 The Back-end

In today's world, mobile applications use a lot of data. That data has to be managed properly to preserve the best functionality and user experience. The Boxing Buddy mobile application will need to store at least 5 previous workouts. In addition to this, the mobile application will need the current workout information to display on the home screen including time, elapsed time, and calculated force. The best way to manage this data is to use a cloud computing service.

There are several benefits to using a cloud computing service. First, using a cloud computing service means the data is stored on the internet which means less space is required from the microcontroller unit. Moreover, this means the application is very scalable. Cloud computing allows for the processing of very large amounts of data so if the group members decide to scale the mobile application, the framework will already be in place. In general, the benefits of a cloud computing service center around its flexibility. Cloud computing services can be cheaper than traditional databases, but they are by no means free. Aside from a free trial, they all cost some amount of money that will need to come out-of-pocket for the group members. This means a major constraint for this service is price. The constraint of time of development is also paramount in the selection of the backend APIs. All of which will require quite a high learning curve so any way to reduce development time will be prioritized.

5.12.1 Amazon Web Services

Amazon Web Services is a publicly available, open-source cloud computing service. It started out in 2006 and is currently the most widely used service of its kind. The fact that Amazon Web Services is open-source means there is more support for this service. More tools that do not have to be created by the group members means a shorter development time due to code that doesn't have to be written from scratch as well as decreased debug time.

Amazon Web Services offers the AWS Amplify which is a development platform that can be used to create hybrid mobile applications. The beauty of AWS Amplify is that it acts as a tool to help create and manage the services of Amazon Web Services. A diagram of how Amplify acts as a manager for the backend API services is shown below in Figure 5.21.

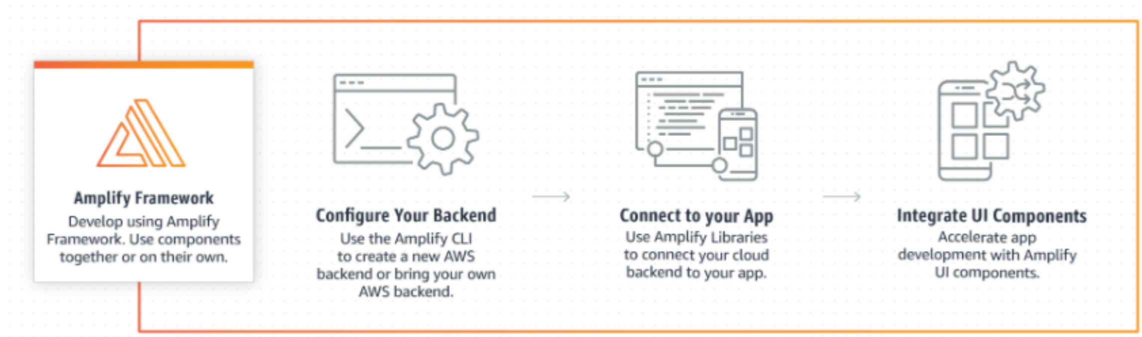


Figure 5.21: Amazon Web Services Amplify Diagram

The right side of this diagram in Figure 5.21 shows the backend connecting to the mobile application and user interface. Amazon Web Services actually recommends using React Native as a framework to build a cross-platform mobile application. This is a notable benefit because in the previous section (section 5.11), React Native was declared the most ideal framework to use.

The whole point in using Amazon Web Services is to manage data. The data will be stored in Amazon DynamoDB which can also be managed by AWS Amplify. Amazon DynamoDB is a NoSQL database. NoSQL database refers to a database whose contents are not relational. While a relational database is more common, the NoSQL database is faster because no “joins” are required of the related components like in a relational database. This results in faster queries leading to a faster response time.

5.12.2 Azure

Azure is a cloud computing service created by Microsoft that started in 2010. It is not yet open-source though it is progressing that way. Although Azure is geared toward being used for Microsoft specific services, it is still widely used with many other services. It has a large user base, though not as large as Amazon Web Service, but large enough to provide quite a bit of support.

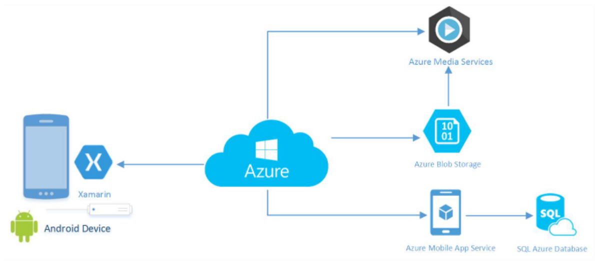


Figure 5.22: Azure Structure Diagram

Figure 5.22 Above shows how Azure acts as a back-end API for hybrid mobile application development. The figure was taken from the Microsoft website. Clearly Microsoft suggests making Android applications and usingXamarin because these are Microsoft services. Azure can be used to make hybrid mobile applications but it is not as well supported.

Microsoft recommends using primarily the MySQL database for the backend. NoSQL databases are available but there appears to be more microsoft support for using Azure with MySQL. MySQL is a relational database. This means there are often at least two sets of information needed for a query that need to be joined together to retrieve useful information. This results in a slightly longer query time.

5.12.3 Google Cloud Platform

Google Cloud Platform was created in 2010 and is the smallest of the main three cloud computing services. Google Cloud Platform does not appear to recommend any specific framework which is a benefit in terms of flexibility. Flexibility is always a benefit because when it comes time to build the mobile application, obstacles may arise where a change of course is necessary. If the backend allows for this flexibility, this can reduce the development time which is one of the two main constraints for the software development of the Boxing Buddy. Flexibility also allows for changes to be made later on more easily. Changes will likely be necessary in the late stage due to inevitable bugs. Having a flexible system will prove very helpful when trying to debug.

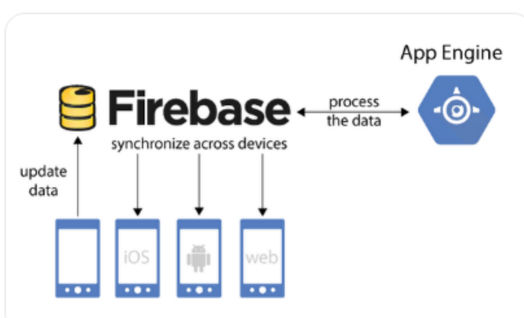


Figure 5.23: Diagram of Google Cloud Services

Figure 5.23 above shows the structure of how Google Cloud Services is used for hybrid mobile application development. Firebase is the data storage element. Firebase is the google framework for creating the mobile application. The Google App Engine is the backend API that gives the developer the ability to process and act on the data. Google uses MySQL primarily like Azure which is slightly slower than NoSQL. There are NoSQL options for Google Cloud Computing Platform, but they are not as widely supported.

5.12.3 Back-end API Summary

Table 5.19 above summarizes the main advantages and disadvantages of the three main back-end API options. All of the three main cloud computing platforms above will require a subscription service to be paid by the group members.

Selection	Primarily used with	Primary Database Type
Amazon Web Services	React Native	NoSQL
Azure	Xamarin	MySQL
Google Cloud Platform	N/A	MySQL

Table 5.19 Summary of Back-end APIs

Each option is roughly the same price so the constraint of cost is not swaying the decision in this case. The type of database preferred is NoSQL. The Boxing Buddy mobile application does not require a MySQL database for any particular reason so the NoSQL database is preferred. This is due to the query speed of the NoSQL database. Additionally, the front-end framework selected for its own benefits was React Native which Amazon Web services recommends is used for hybrid applications built using that cloud service. Azure recommends Xamarin which would result in an additional subscription service. The constraint of cost limits the group members and rules out that option. This leaves the clear choice as Amazon Web Services as the back-end API selection. This adheres to the constraints of being the most time efficient and cost effective cloud computing service.

5.13 Power Routing

The power routing for our project will be straight to the outlet in a wall. However, this routing will have to be connected to our project in one shape or another. The way it can be connected can be determined by these two products: USB mini or Barrel connector.

The USB mini is a wall adapter that can have a voltage input of 90 ~ 254 VAC and an output of 5V. Along with that it has 1 A of max current output and 5 W of power. This is an AC/DC adapter.

The barrel connector on the other hand has an input voltage of a steady 120 Volts (AC), 5 watts or power, 1 A of current output, and 5 output voltage (DC).

5.14 Parts Selection Summary

This section will summarize the final selection choices within all mechanical, software, and electrical components in a chart. This section will not include any duplicate parts purchased for testing. There will also be some components not yet purchased that will be included here, so the total cost will not yet be known.

Category	**Type**	**Selection**
Front-end framework	Software	React Native
Back-end API	Software	Amazon Web Services
Main Programming Language	Software	Javascript
MCU	Electrical	ATmega2560
LEDs	Electrical	3 mm LED pre wired
Force sensor	Electrical	Accelerometer
WiFi module	Electrical	ESP32-DOWDQ6
AC/DC adapter	Electrical	5V 1A Wall Charger Power Adapter with Plug 5.5 x 2.5mm / 5.5 x 2.1mm
Voltage Regulator	Electrical	TPS6286920CRQYR

Table 5.20: Parts Selection Summary

6.0 Mechanical Design

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

6.1 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 and to adapt them to our own use. Boxing strike pads need 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 adapt the strike pads for our purposes we needed to adjust them to be able to connect to the adjusting height rail that would allow slight movement when they were struck to mimic training with a partner. We chose the strike pads based on their size, shape, price, and reviews.

One of the products we considered was the Overmont Taekwondo Kick and Boxing Karate Pad which can be seen in the image below. These pads are fairly reasonably priced at around $20 each, so $40 total for our purposes. The pads also had primarily positive reviews. These strike pads weigh 9 oz and are 15.7" x 7.9" x 3.9". Due to the shape and size of these strike pads, the Boxing Buddy would best be designed with an arm on either side of the adjustable rail that would attach to the back of the strike pad for stability. In order for these strike pads to move slightly when the users hit them, the rail would need to be able to attach to a compression spring. While this design is possible, it would likely be more costly due to the additional beams that would make up the arm and a spring sturdy enough to hold the weight of both the steel arm and the strike pad. They would also require more fabrication than other pad options with the addition of the arm that would require a directional change to attach to the pads.

Figure 6.1: Overmont Taekwondo Kick and Boxing Karate Pad

Another pad that we considered was the RDX T15 Noir Focus Pads shown in the image below. These pads come in a pack of two and are also around $40 for both. Compared to other pads of a similar style and shape, these pads had better reviews, stating that they could be used by experienced adults unlike similar items that consumers stated seemed to be primarily for children or beginners. The pads are 10.16” x 7.99” x 8.74” and around 25 oz. To attach this style of strike pad to the Boxing Buddy, at the bottom end of the strike pads we would secure a piece that would allow the strike pads to directly attach to the compression springs without an arm. Due to this easier attachment and need for less fabrication, we decided to use the RDX strike pads over the Overmont strike pads.

Figure 6.2: RDX T15 Noir Focus Pads

6.2 Support Tubing

The Boxing Buddy needed beams 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. One of the primary concerns for the support tubing was the material. When selecting a material, the two materials that we were selecting between were aluminum and steel, as those are what we found most other similar products were made of, and there was a wide variety of materials within our price range available in those materials. 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. There was also minimal difference in the price of the aluminum products and the steel products, so we decided to go forward with the steel beams.

Another aspect of the support tubing that we focused on is the shape that they would be. The shapes that we were deciding between were tubing with circular cross sections and tubing with rectangular cross sections which can both be seen in the image below. Our decision for which shape depended once again on the ability of the tubing to resist deformation from the stress acting on the tubing when users are training with the Boxing Buddy. We found that the moment of inertia of tubing with a rectangular cross section is higher than tubing with a circular cross section, so bending stress is lower for tubing with a rectangular cross section than tubing with a circular cross section. For this reason we chose to go with the rectangular tubing.

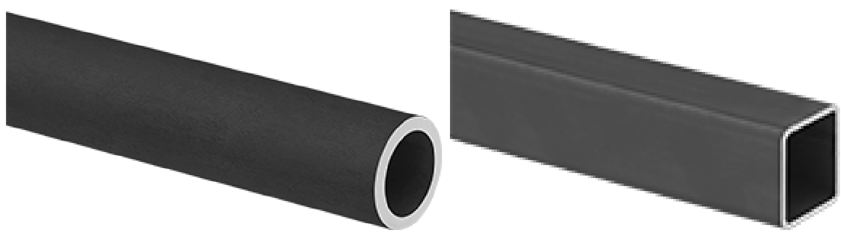


Figure 6.3: Tubing with a circular cross section (left) and tubing with a rectangular cross section (right)

6.3 Adjustable Height

In order 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 rails. Telescoping rails are two hollow rails used together as shown in the image below. One of the rails is slightly smaller than the other, typically ¼” smaller in height or diameter, and it fits inside the other rail so it can slide inside it to the desired length. In order for the rails to remain at a height after they have been 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. The holes would be large enough for a metal pin to go through and keep the inner rail in place at each of the height options. Like the support beams, the rails used for the adjustable height also needed to be sturdy, so we chose steel rails with rectangular cross sections.

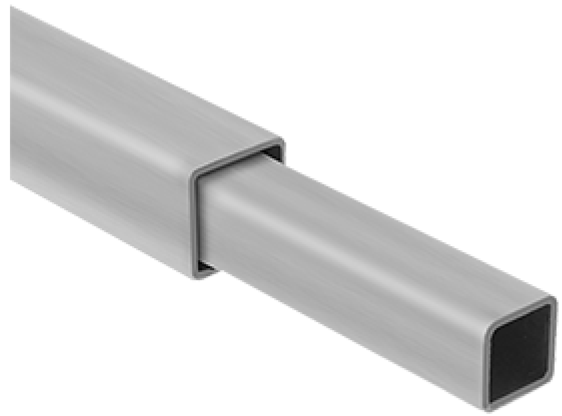


Figure 6.4: Telescoping Weld-Together Rails

6.4 Cable Housing

For the electrical components of the Boxing Buddy, cables were used to connect components, thus creating a need to consider housing the cables. One of the primary reasons for considering cable housing is for aesthetic purposes. People often think a product does not appear professional or complete if the cables are exposed and unorganized. In addition to the cosmetic aspect of cable housing, it is also important to help prevent damage to the cables. If the cables are not within housing, they are exposed to external forces and at a higher risk of damage. Users could unintentionally snag the cables causing them to break or disconnect. Another reason to include cable housing is to prevent the cables from posing as a safety hazard. If the cables are exposed pets could easily have access to them to chew on them and injure themselves, or people or animals could get caught in the cables.

Based on the above reasons we ultimately decided to include cable housing in our design. One of the ideas we initially considered was running the cables along the sides of the railing and tubing and creating a cover to attach to the outside. One of the main issues we found with that design was with the adjustable height railing. If the cables ran along the outside, they would interfere with the capability to easily adjust the height. Additionally, running the cables along the outside would require additional fabrication or purchase of the housing for the cables and could potentially look less aesthetically pleasing. We decided the aforementioned design would not be ideal for the Boxing Buddy and chose to go a different route. The design that we chose to include instead required no additional fabrication aside from an extra hole in the large rail of the telescoping rails. To house the cables we decided to use the railing and tubing as the housing, keeping the cables inside of them. The cables enter through the top opening of the smaller rail for adjusting the height and feed down through the telescoping rails to a hole near the bottom of the larger rail, shown below. This design revoked the need for additional fabrication while providing protective and aesthetic housing for the cables.

Other electrical components that will not fit within the tubing, such as the MCU, also need protection, which warranted the design of a protective box. The protective box was designed out of plastic to be lightweight, but a plastic that handles vibration well was chosen. The design can be seen below. There is a notch at the top of the design to allow for any connecting cables to connect to components inside of the box.

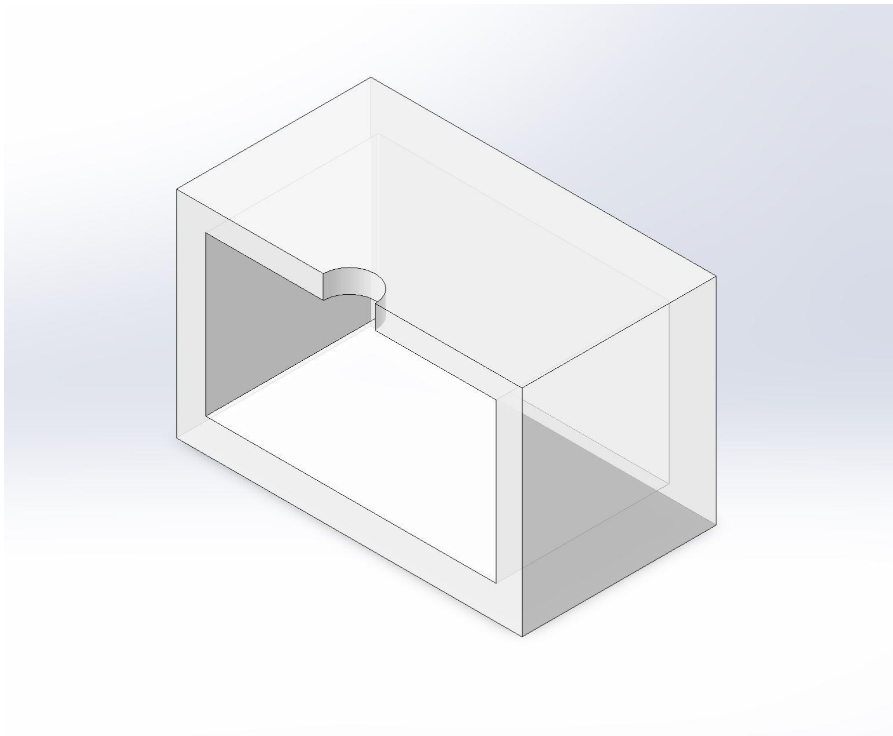


Figure 6.5: This shows the box to protect electrical equipment that can not go within the tubing.

7.0 Project Prototype Testing Plan

Project prototype testing plan includes the testing of mechanical design. More specifically the force analysis of the mechanical structure. After all, the structure is meant to withstand great amounts of force if given by a novice boxer or elite boxer. The force analysis will help us realize where in the structure will need to be more sturdy or if the material we are using will be needing changes. Along with maybe changing the materials we can also change the length of a few things because as things get further away from the center of mass the more unbalanced and unsturdy it gets.

7.1 Mechanical Design Testing

To test the structure of the product prior to purchasing the materials and constructing it, we used two softwares to model the device and conduct stress analysis. The product was designed using Solidworks. Within Solidworks, the specific material of a part can be selected and each of the parts can be mated together to create an assembly. This allowed us to make an accurate model of the Boxing Buddy to conduct testing before constructing the Boxing Buddy. It additionally provided us with a detailed model to give to the shop to fabricate parts of our project outside of our range of skills. The assembly that was designed in Solidworks is shown below. All of the components of the boxing buddy were designed as individual parts in Solidworks and then they were meshed together in an assembly. Creating the components as individual parts provides a more accurate representation of how the Boxing Buddy is intended to be in real life, as it will be in multiple parts that can easily be assembled to allow for smaller shipping and general transportation of the product. Each component being a separate part in Solidworks also allowed for more testing of variations in the design. If the entire structure had been designed as one part, then it would be more difficult to make adjustments to various sections of the product than it is when each of the components is an individual part that can just be swapped out. It is also particularly helpful and important to have the components as individual parts in order to incorporate the adjustable height. With the Solidworks assembly of the Boxing Buddy, the height can easily be adjusted to conduct a force analysis at various height settings of the Boxing Buddy.

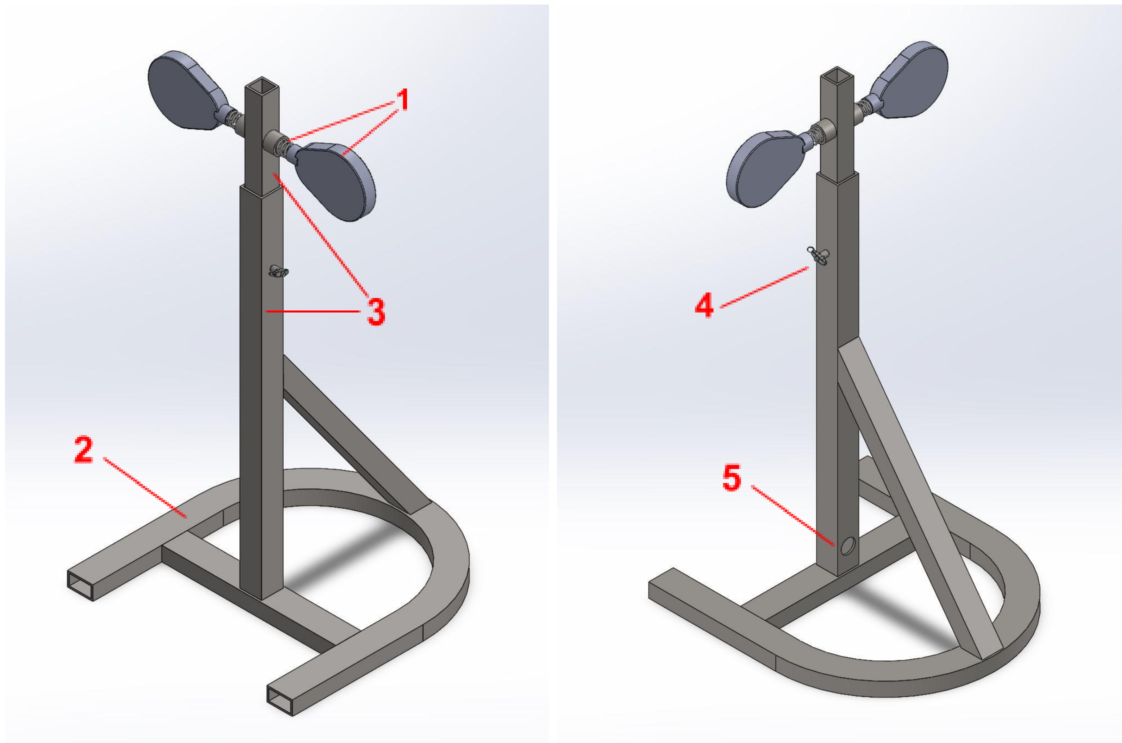


Figure 7.1: Two views of the Boxing Buddy design in Solidworks with numbers corresponding to parts

The design above does not include the electrical components, but all of the structural components can be seen. The red lines and numbers on the images correspond to the following part descriptions:

1: A strike pad and the compression spring it attaches to that connects the pads to the rest of the structure and allows for movement

2: Tubing with rectangular cross sections used to for the supportive base of the Boxing Buddy

3: Telescoping rails allowing for height adjustment

4: Pin used to secure the telescoping rails at each of the height options

5: Hole allowing for cables housed inside the telescoping rails to exit out the bottom of the larger rail

Although Solidworks is a reliable tool for design, within the industry it is not trusted for structural analysis. Since Solidworks is not trusted for structural analysis, to conduct stress analysis on the Boxing Buddy design, we used Ansys. Ansys is used more in the industry as it is more trustworthy and accurate than Solidworks for structural and fluids analyses.

In order to test the structure designed in Solidworks in Ansys, there were multiple steps that had to be taken. The Solidworks file had to be saved as an IGS file type instead of the default Solidworks Assembly Document file type in order to be imported into Ansys. Once imported into Ansys, a mesh had to be created around the part. A mesh creates nodes all over an object that connect to create geometries that make up the object. If a mesh is made finer, meaning it includes more nodes and geometries, it will better represent the object and produce more accurate results. Ansys allows for the fineness of the mesh to be adjusted for both the object as a whole or individual sections or surfaces where there is greater concern for the effects of the force. The design of the Boxing Buddy with a mesh is shown below.

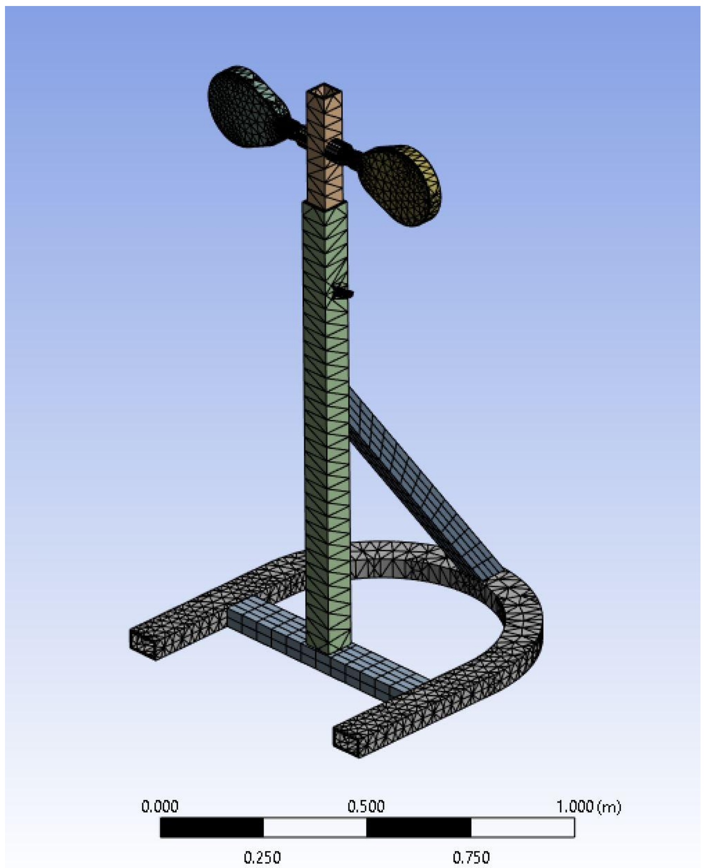


Figure 7.2: The Boxing Buddy design in Ansys with a mesh

After the mesh is created on the object, the forces acting on the object must be added. The axial directions for the Boxing Buddy are shown below for reference. The fact that the Boxing Buddy will be resting on the ground is represented by the normal force from the ground, acting in the +y direction, opposite to the equivalent downward force from the weight of the Boxing Buddy acting in the -y direction. The force from the users strike then has to be added. It is not likely that a user will strike both strike pads at once, so the force will only be analyzed at one strike pad at a time. The force of the user's strike is in the -z direction onto a strike pad.

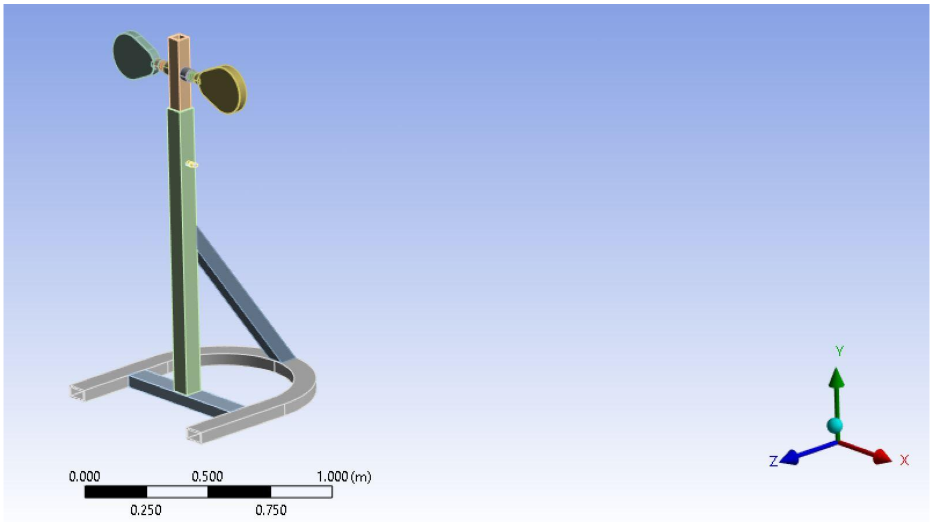


Figure 7.3: The Boxing Buddy axially oriented.

After the force is added, the analysis can be conducted to demonstrate any effects the force will have on the object. For the purposes of our project, we are primarily concerned with any lasting deformation or breaking that the force could cause, or if it would cause the entire structure to be displaced. We began with the average force of a skilled boxer's punch which is around 2,400N and adjusted the force for other trials. We conducted this analysis process for various height settings and forces to represent a variety of users. With the materials and structure we chose and created, our design was able to properly withstand the forces used in the analyses.

7.2 MCU Testing

The selected microcontroller unit is the ATmega2560 chip. This chip is made by Atwell and can be found on the Arduino Mega 2560 development board. The chip and the board can be purchased separately and the Eagle file is available for free from the Arduino website.

The testing of the ATmega2560 microcontroller which will be used in the PCB design for the Boxing Buddy will need to be tested using the development board with the same chip which in this case is the Arduino Mega 2560 Rev3. The Boxing Buddy will require the use of GPIO pins as well as the analog to digital converter pins. Each accelerometer needs three position analog pins corresponding to the X, Y, and Z coordinates of its location. This means six analog to digital convert channels are required for the Boxing Buddy. These channels will be tested during the accelerometer testing described below. The

MCU will also be tested with WiFi capabilities when it is connected to the chosen WiFi module.

7.3 WiFi Testing

The WiFi chip selected is the ESP32 DOWDQ6. This WiFi chip can be found on the ESP32 development board and each can be purchased separately. The Eagle File is available for the ESP32 development board, so the schematic can be added to the schematic of the ATmega2560.

The ESP32 development board will be used to test the WiFi capabilities of the system. The ESP32 can use Arduino IDE which mainly why it was chosen. In order to test, the ESP32 Rx and Tx pins must be connected to the Tx and Rx pins of the Arduino Mega 2560 respectively. Other essential pins, like the power routing pins, must be connected on the Arduino Mega 2560 and the ESP32 development board. Then, simple code is uploaded to establish a network connection.

The test must be kept simple in this case. Only a proof of concept is necessary. The code uploaded just enters the network credentials and submits a “Wifi.begin” command. The terminal window displays the connection status.

```
rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0x00
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:
mode:DIO, clock div:1
load:0x3fff0008,len:8
load:0x3fff0010,len:2036
load:0x40078000,len:9988
load:0x40080000,len:252
entry 0x40080034
Connecting to WiFi..
Connecting to WiFi..
Connected to the WiFi network
```

Figure 7.4: WiFi Testing Terminal Window

Figure 7.4 shows a terminal window example where the WiFi connection status becomes “connected to the WiFi network.” This is the desired output.

7.4 Accelerometer Testing

An accelerometer can be tested many different ways. However, in our instance we will be testing it with a microcontroller and breadboard. The microcontroller will be the MSP430 by texas instruments. An accelerometer can measure the acceleration in two dimensions or three dimensions. For our instance we got one that can measure in three dimensions since the punch will make the thing move in three axes.

Given the accelerometer it will be giving out an analog input. To get the accelerometer to work with the microcontroller we need to connect it with jumper wire and solder. The accelerometer has six different holes to connect jumper wires and solder them. Three of those holes include the 3 axes: x, y, z. Every one these holes will need to be connected on the analog in section on the microcontroller. From this the microcontroller will know where to retrieve the sensor data from the accelerometer. In other words, the acceleration of the device.

One of the other holes that a jumper wire can go through and solder is the GND hole. The GND hole is ground. This is connected to the GND on the microcontroller for any excess power it may receive. In other words, it's where the voltage is zero. GND is also used for all electrical devices that are being connected together to be met and complete the circuit.

The other hole is the Vcc hole. The jumper wire will be soldered onto the hole and then plugged in the microcontroller in the power section. Depending on the amount of power the device needs depends on where it is plugged in. But in the case of an accelerometer it will be connected in the 3v3 power section. The 3v3 power section means that the voltage is at 3.3. The reason that chips and smaller modules work with 3.3 Volts is because any more voltage may fry or burn the chip. There is another voltage supply that is higher voltage if needed to connect a more demanding voltage device.

After all this is done and the accelerometer is connected the microcontroller will be able to read the accelerometer and give out the values it is reading. The readings will come out in variables of x, y, and z. After giving these values in the code we will divide by the measurement of gravity which is 9.8 m/s^2 because this is an active force acting upon everything on earth. This will only work with the code that will make the microcontroller output these readings onto lets say an application or LED screen or on the screen of the computer.

After the testing is done and the accelerometer is working perfectly fine this is when we will start to displace the accelerometer in different parts of an object and start experimenting measuring the force. What is meant by displacing is setting the accelerometer in different parts of an object and moving the object and determining what is the best way of catching the acceleration of the object that it is attached to.

The way we will get force from this is from Newton's second law of motion, F=MA. From the accelerometer we can get the acceleration of the object. The object has a mass as well. From this we can use the equation from Newton's second law of motion and get the force.

All the conversions will be done in the code of the microcontroller and be given out as one final output. The image below is the result of all the testing procedures being completed. The wires got soldered onto the accelerometer and then connected to the microcontroller using jumper wires

There must be different codes depending on the specific accelerometer. We used a code for a slightly different accelerometer and it did not work. When finding the accelerometer (ADXL335) we searched up examples and found one to use to test ours respectively. After downloading code and using it it worked perfectly fine and gave us the outputs of the x, y, and z values.

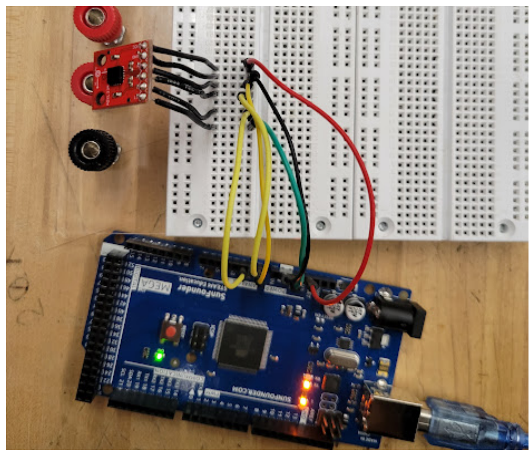


Figure 7.5: Set-up for testing

The figure below shows the result of the values of x, y, and z in the accelerometer. These are the values that will be used to determine the force of the user. These are the values that will be converted over to force with F=ma.

```
15:41:44.092 -> accelerations are x, y, z: 352 373 410
15:41:44.186 -> accelerations are x, y, z: 352 373 409
15:41:44.278 -> accelerations are x, y, z: 351 372 409
15:41:44.416 -> accelerations are x, y, z: 351 371 408
15:41:44.509 -> accelerations are x, y, z: 350 371 408
15:41:44.604 -> accelerations are x, y, z: 350 371 408
15:41:44.698 -> accelerations are x, y, z: 351 371 407
15:41:44.792 -> accelerations are x, y, z: 349 370 407
15:41:44.884 -> accelerations are x, y, z: 350 371 408
15:41:45.023 -> accelerations are x, y, z: 350 371 407
15:41:45.115 -> accelerations are x, y, z: 350 371 408
15:41:45.208 -> accelerations are x, y, z: 350 370 408
15:41:45.303 -> accelerations are x, y, z: 351 371 409
15:41:45.395 -> accelerations are x, y, z: 352 373 410
15:41:45.488 -> accelerations are x, y, z: 352 373 409
15:41:45.627 -> accelerations are x, y, z: 351 372 408
15:41:45.720 -> accelerations are x, y, z: 351 373 409
15:41:45.815 -> accelerations are x, y, z: 351 372 409
```

Figure 7.6: Accelerometer output from testing

The values of the x, y, and z shown in the figure above are the values of when the accelerometer is at rest.

8.0 PCB Design

This section will provide information on the PCB Design and what it will entail. The job of the PCB in our project and how the construction of the PCB will be followed. The PCB in the Boxing Buddy has several functions. The first and foremost function is to route power to the different components. The second function is to acquire data via timers and the analog to digital converters attached to the accelerometers and communicate this information to the mobile application.

8.1 Components Included

Within the design process of creating our Printed Circuit Board there were certain components needed in order to ensure that we had a significant PCB design. According to documents found within the senior design canvas webpages, a significant PCB design must have a microcontroller chip on board that does not include being a development board snap on. The MCU chip must be fully integrated into the system along with any other power, sensor inputs, or output controls that may be needed in order for the overall system to function properly.

The microcontroller is the ATmega2560 which is the same microchip on the Arduino Mega 2560. The Arduino Mega 2560 is the development board that is used for testing only. The ESP32-DOWDQ6 is the WiFi chip selected which will also be on the PCB. The ESP32 development board contains the ESP32-DOWDQ6 but the ESP32 development board is only used for testing. The eagle schematics from each of these will be combined to form the schematic used in the Boxing Buddy.

The Arduino takes power using an Aux power cable that can connect to the wall. The PCB will use this same power input and have 5V and 1A coming into it. In order to control the voltage level to each component that may require different voltages, a voltage regulator is used. The voltage regulator must be added to the Boxing Buddy schematic to adjust voltage level for the accelerometer and the microcontroller.

In order to meet these criteria the following components will be included in the printed circuit board. The following table enumerates the different components and their purpose.

It is possible that after building the Boxing Buddy, a filter will be needed for the analog to digital converter to make the readings more precise and clear. This will be found later. A filter is a basic component using resistors and capacitors and does not need to be included in the table or the parts selection summary where price is calculated.

Component	Purpose
Power Supply	- Receive Power from plug source and convert to digital as needed - Power source for entire system
Microcontroller (MCU) Chip	- Gather sensor information, process the information, and output a particular action based on the information gathered
Wifi Chip Module	- Communicate between microcontroller and a cloud based storage system to integrate with our mobile application
Analog to Digital Converter	- Used to convert Analog input from sensors to Digital input to integrate within the microcontroller
Sensors Input connection	- Provide connection between sensor input data and microcontroller to process the data
Voltage Regulator	- Voltage regulators needed to provide a DC voltage drop between 5V which will come out of the microcontroller on board to 3.3V which is needed by the sensor inputs within the system

Table 8.1: PCB Components summary

With these components within our eagle schematic that would signify an acceptable significant PCB design for the project.

8.2 DC-DC Voltage Regulator Design:

The voltage of the power supply feeding a PCB is seldom at the correct level needed for the circuitry involved, which is exactly the case that occurs in the Boxing Buddy Robot. As mentioned in the table above on the printed circuit board there will have to be a DC-DC conversion between the voltage brought onto the board from the wall outlet through the power supply and subsequently get dropped down to the microcontroller voltage of 5V and the sensor voltage of 3.3V. In order to design this section of the PCB our team utilized Webench Power Design.

In the design pictured below, we created a buck convertor for a 5V input from the microcontroller system and are converting it to 3.3 V for the accelerometer measurement that we are using to collect our force data. In this design we are utilizing the TPS6286920CRQYR buck switching regulator in the circuit design shown below.

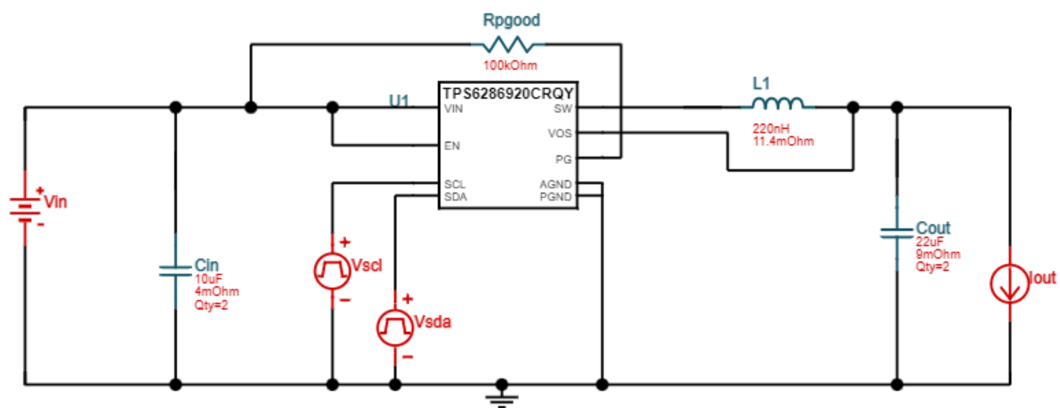


Figure 8.1: DC-DC Voltage Regulation Schematic

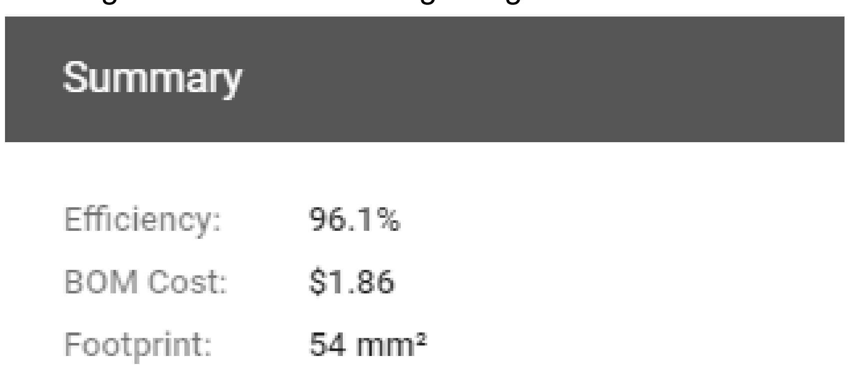


Figure 8.2: DC-DC Regulation Summary of Circuit Specifications

The above circuit design was chosen by our group due to its high efficiency. It is important that the DC voltage levels are as accurate as possible to ensure the proper operation of the different components. The circuit will be put into effect within the printed circuit board itself.

8.3 PCB Schematic

When making the PCB schematic we decided to use the Eagle files of the Arduino Mega 2560. After implementing these files and opening them in Eagle we decided to use it as a foundation. After this we will remove the components not needed on the arduino mega 2560 and use this as the PCB schematic. The only components needed from a microcontroller are the analog inputs, the power input, GND, 3V3, Wifi chip, etc.

The wifi chip we used was the ESP32 WiFi Module. From this we can also grab the Eagle files from the datasheets and use it to input onto the current PCB we have. So, the essence of this was the Eagle files that the arduino mega 2560 had will be used hand in hand with the WiFi module Eagle files.

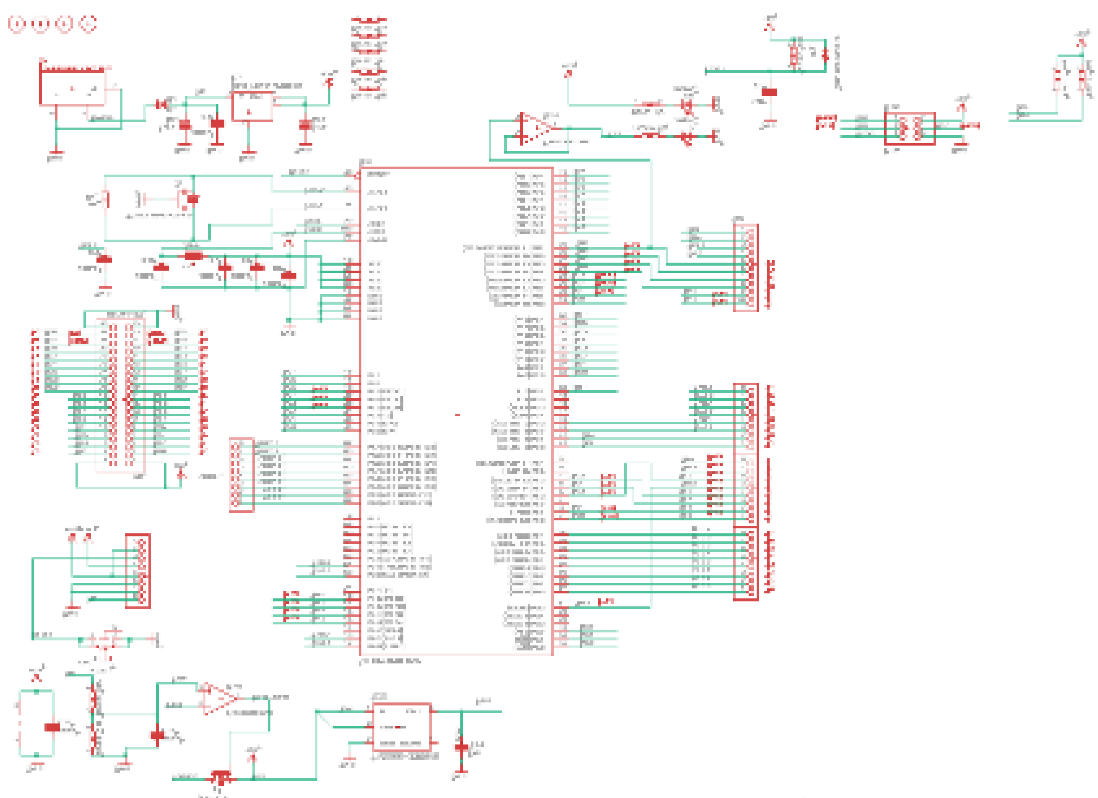


Figure 8.3: PCB Schematic (Foundation)

8.4 PCB Reflow Oven Process

Due to how small the parts are that are being used in the design our group will utilize a reflow oven when actually attaching these components onto the board.

Reflow soldering can help to quickly assemble printed circuit boards that may pose manual hand soldering challenges. We will follow the following procedure when soldering our parts onto the board in order to ensure the greatest amount of functionality and efficiency possible:

1. Create a stencil for the fabricated PCB
2. Create a solder paste mixture containing both solder mask and flux to be applied with a stencil to the fabricated PCB
3. Place the components in position and hold them in place with the adhesive nature of the solder paste
4. Place the PCB inside the convection reflow oven to begin the process

The convection reflow oven process is based on convection heating. The temperature of the hot air in the reflow oven creates a thermal profile that is the optimum heat and cooling rate for the solder paste and components within the board and stencil allowing the pieces to adhere to each other cleanly even when the parts are much too small to be properly soldered by hand. The four main steps within this thermal profiling process for the board are preheating, soaking, reflow, and cooling. Putting the circuit board through these temperatures allows the solder paste to melt, however the temperature should not be too high to where the components on the board become damaged. Pictured below is a typical solder reflow thermal profile that should be taken into consideration when actually creating the printed circuit board through the reflow oven.

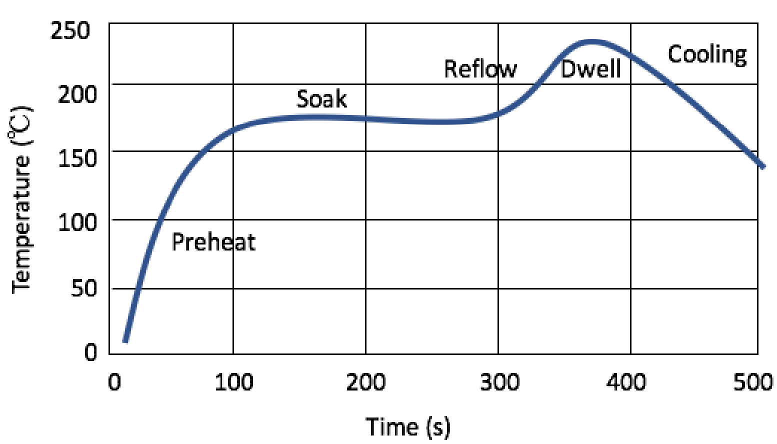


Figure 8.4: Typical solder reflow thermal profile

Depending on how big of a board that is being created and how updated the software is needed can cause different reflow ovens to be better suited for different purposes. Temperature precision and sense are also important considerations to look into when choosing which reflow oven is best suited for a particular circuit board use, however due to access abilities, our group will be utilizing the reflow oven that is available in the Senior Design lab in order to attach components on our circuit board.

Utilizing the reflow oven will not only make our connections more secure and accurate, but will also make the soldering done to the board quicker and more efficient especially when compared to hand soldering.

8.4 PCB Manufacturers

Many factors come into mind when choosing a PCB manufacturer for your construction of a PCB. The main factors, however, are the industry expertise/ capabilities, quality/certifications, the amount of time it takes for the delivery, and of course the expense. The PCB is a crucial part to making the performance between all electronics devices communicate with each other great. Choosing a PCB manufacturer is an important decision. Therefore, we will be comparing the options we have and see which would best benefit us in our situation.

All PCB manufacturers differ from the services they offer. Which is basically their capabilities. We will have to choose a manufacturer that provides us with the type of PCB we require. The difference in PCB comes from single-sided, double-sided, low layer count, etc.

Quality is an important factor as well since you do not want a cheap quality PCB. Each manufacturer has a quality management system (QMS). What a quality management system is is a system that documents the processes, procedures, and responsibilities for achieving the objectives. This makes it easier for the customer to have a clear vision as to how the PCB is being made and if it is up to the customers standards. Manufacturing yield percentages come under the category of the quality of the product. It should be shown and easily found under the information of the manufacturer.

Delivery is an important factor. Tracking nowadays should be a given but sometimes it is not provided and when deciding on a manufacturer you would want to be able to track your product and see where it is at. Alongside the tracking, they should add a time estimate so the customer can expect it by a time or range of time.

Lastly, the cost of the product is a big factor. Everything has a reason. The cost for the manufacturer can be low but see where it comes from. It may come from a far place which may lead to shipping costs or longer delivery time. The cost can be affected as well by the technology being used in the PCB. Using a higher end can lead to a higher cost and vice versa. So we need to find the sweet spot and find something that is not too high tech but can still satisfy our needs in the ways we intend to use.

As of right now the nearest PCB manufacturer in Orlando is Quality Manufacturing Services, INC, JRT Manufacturing LLC, and Circuitronics LLC.

PCB Manufacturing	Certifications
Quality Manufacturing Services, INC	ISO 9001:2015 and AS9100:2016
JRT Manufacturing LLC	N/A
Circuitronics LLC	ISO 9001:2008 compliance company

Table 8.2: PCB Manufacturing Summary

The table above will compare the nearest PCB manufacturer in Orlando and compare their certification for their credibility. Quality Manufacturing Services takes the lead since they have both ISO and AS certifications.

9.0 Mobile Application User Interface Design

The following section will depict the design for the user interface of the Boxing Buddy mobile application. Existing similar products like the home boxing trainer made by Nexersys mentioned in a previous section usually have a mounted screen that displays the data and controls the boxing system. The Boxing Buddy is unique in that the user's mobile device will act as a remote control that will control the boxing system as well as display the data from the workouts including previous workouts.

Because the user will need to download a mobile application in order to use the Boxing Buddy to its fullest potential, it is extremely important that this mobile application is user friendly. Being simple and easy to use will make the user more likely to want to use the mobile application and thus the Boxing Buddy.

The figure below shows a prototype for the home screen of the mobile application. Buttons are shown as rounded shapes. There are two buttons at the top that will allow the user to navigate pages. One is the "Home" button which is the starting page. The other is the "Previous Workouts" page that will take the user to a new page that will display at least five, most likely more, previous workouts.

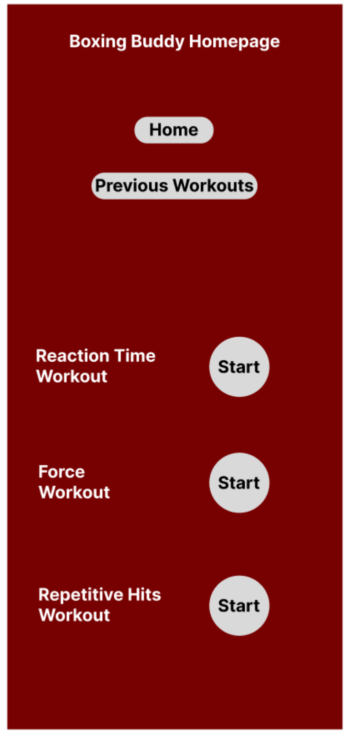


Figure 9.1: Boxing Buddy Homepage

The figure below shows the interface for the "Previous Workouts Page." After clicking on the "Previous Workouts" button the user is navigated to this page.This page will display at least five previous workouts in chart form.

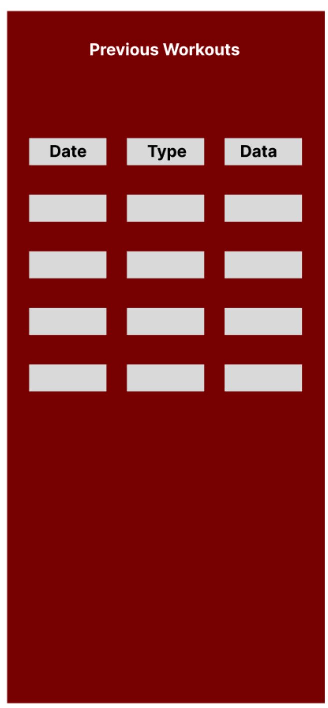


Figure 9.2: Previous Workouts Page

The left most column will display the date of completion of the workout. The workouts on this page will be organized by date with the most recent workout at the top. The second column will display the type of workout completed, either the reaction time workout, the force workout, or the repetitive hits workout. If the workout type is the reaction time workout, then the data section in the third column will display the reaction time calculated as well as the accelerometer reading converted into a force. If the workout type is the force workout, then only the force reading is displayed. If the workout type is the repetitive hits workout, then the duration of the workout as well as the hit count is displayed.

There are three other buttons on the home page shown in figure 9.1. They are all start buttons that correspond to the three workout types. Pressing the “Start” button for each of these workout types will route the user to a new page that is specific to that workout type. Once navigated to the specific workout page, there will be a display specific to the selected workout and a couple other buttons to choose when on the page.

The figures below show the pages that the user is navigated to when they select the corresponding start button.

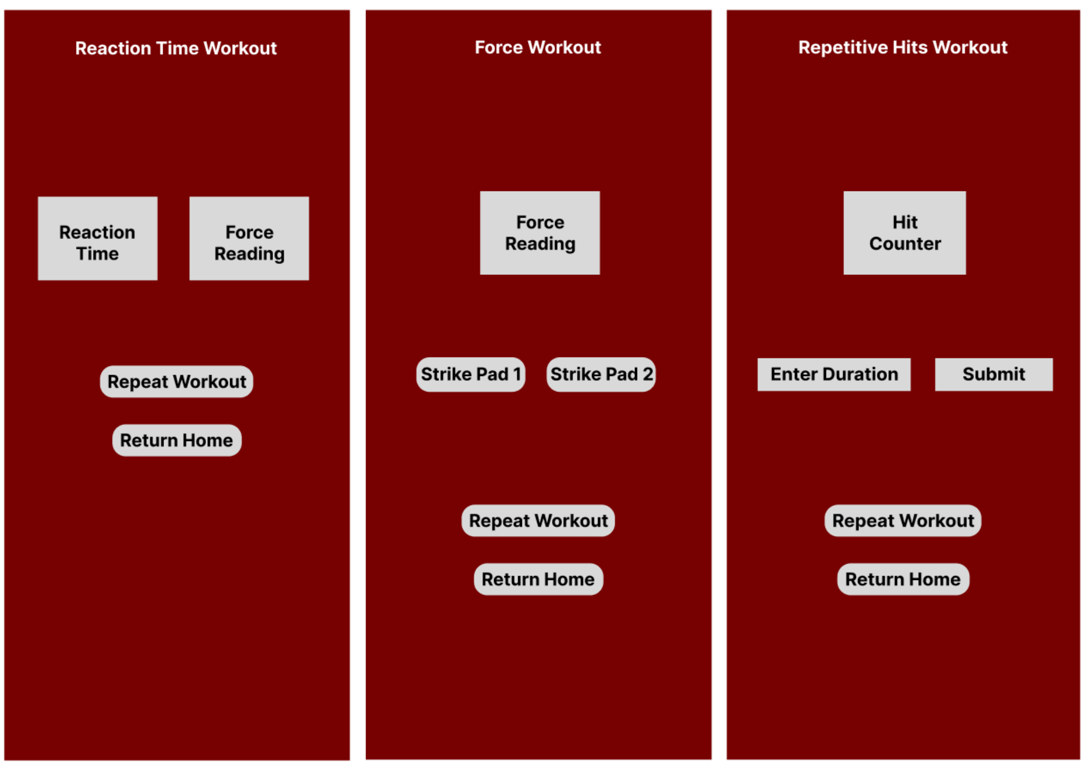


Figure 9.3: Workout Specific Application Pages

The reaction time page, shown on the left in Figure 9.3, has two display features. The left display box shows the reaction time of the user between when the indicator light above a random strike pad was illuminated and when the user hit the indicated strike pad. Note that there is a five second delay to allow the user to prepare after selecting start. The second display box is for the force reading. This display box shows the force reading taken by the accelerometer during that strike. The force reading is taken from the accelerometer and converted into a force. Lastly, there are two buttons available on this screen. The first button is the “Repeat Workout” button. This allows the user to stay on the same page and complete another of the same type of workout. Each time the workout is complete, it will be saved to the previous workouts page. The other button allows the user to return to the home screen. All of the workout specific pages will have these same two button options of “Repeat Workout” and “Return Home.”

The force workout page, shown in the middle, has one display box and two additional buttons. The two additional buttons are for the strike pad selection (one for each strike pad). The strike pad selection must be made prior to doing the workout. After selecting, the display box will display the force reading from the strike pad selected.

The repetitive hits workout, shown on the right in Figure 9.3, has one display box and additional buttons. Here, the user must enter a duration for the workout and press submit. Then, the display shows the number of hits during that duration.

10.0 Administrative Content

This section of administrative content consists of group affiliated activities. This is the method of where we hold each other accountable and make discussions for our future endeavors.

10.1 Milestone Discussion

This following section includes the milestones and agenda of the group for the remainder of the Fall 2022 semester.

Table 10.1 and 10.2, Fall 2022 Schedule, includes the assignments that are not drafts but still significant in our document and given tasks with a completion date.

Milestone	**Assigned to**	**Task**	**Complete by**
Attend All ABET Lectures			
1	All	Form Group	08/25/2022
2	All	Initial Idea	09/09/2022

Table 10.1: Fall 2022 Schedule

Milestone	Assigned to	Task	Complete by
Divide and Conquer Phase			
3	All	Initial Project Document (Divide and Conquer)	09/16/2022
4	All	Divide and Conquer Meeting with Professor	09/20/2022
Divide and Conquer Phase 2.0			
5	All	Update Divide and Conquer Document	09/23/2022
6	All	Discuss what task everyone has	9/23/2022

Table 10.2: Fall 2022 Schedule Continued

Table 10.3 and 10.4, Fall 2022 Schedule Drafts, includes the draft pages that are assigned throughout the semester. This includes the jobs and everything to be done by the completion date.

Milestone	Assigned to	Task	Completion Date
60 Page Draft Phase			
7	All	Start writing Project Report	09/30/2022
8	All	Make a detail tasks list	09/30/2022
9	All	Order Test Components	10/14/2022
10	All	Start Testing Project	11/04/2022
11	All	Test MCU	11/08/2022

Table 10.3: Fall 2022 Schedule Drafts

Milestone	Assigned to	Task	Complete by
100 Page Draft Phase			
12	Catherine Sailor	Test Mechanical Aspects of the Project	11/17/2022
13	All	Finalize and peer review the paper	11/30/2022
14	All	Final Report	12/06/2022

Table 10.4: Fall 2022 Schedule Drafts Continued

Table 10.5, Spring 2023 Schedule, consists of very broad milestones for the spring 2023 semester.

Milestone	Assigned to	Task	Completion Date
1	All	Finish Testing Project	February 2023
2	All	Get mobile website/app working	March 2023
3	All	Test Final Product	April 2023
4	All	Final Presentation	End of SD2
5	All	Final Report	End of SD2

Table 10.5: Spring 2023 Schedule

10.2 Finance Discussion

Table 10.6, Total Financial Cost, is the financial situation given at the present moment. This will later on be divided by the number of members in the group and be paid.

Item	Quantity	Price	Unit Price	Received
Accelerometer	2	$42.30	$21.15	Yes
LEDs	6	$5.34	$0.89	Yes
MCU	1	$9.73	$9.73	Yes
WiFi module	1	$3.26	$3.26	Yes
AC/DC adapter	1	$8.59	$8.59	No
Voltage Regulator	1	$2.80	$2.80	Yes
Steel Tubing	TBD	TBD	TBD	No
RDX T15 Noir Focus Pads	2	$42.99	$21.49	No
Compression Spring, 4" Long, 1.687" OD, 1.303" ID	2	$23.20	$11.60	No
Telescoping Rails	TBD	TBD	TBD	No
T-Handle Locking Quick-Release Pin	1	$48.54	$48.54	No
Total	TBD	TBD	TBD	TBD

Table 10.6: Total Financial Cost

11.0 Conclusion and Overview

The Boxing Buddy offers people a simple way to practice and hone their boxing skills from the comfort of their home. The Boxing Buddy is designed to be easy to use and effective at helping the user track their skills and get better.

Controlling the Boxing Buddy is done from the mobile application which will be found on the Google Play Store as well the Apple App Store. The mobile application will serve as a remote control of the Boxing Buddy System as well as the interface with which the user views their workout data, whether that is current workout data or historical workout data.

The mobile application design and how it works is further detailed in section 9.0. However, in the mobile application, there are three workout types to choose from. The first workout type is the “Reaction Time” workout. Once this workout is selected, a five second delay will occur. After the 5 second delay, an indicator light will light up on a randomly selected strike pad (one out of the two available) after a random amount of time between 1 and 10 seconds after the initial 5 second delay. Once the indicator light lights up, the user should strike the indicated pad as quickly as possible. The reaction time is the amount of time between the activation of the indicator light and when the user hits the strike pad. In this workout type, the force data will also be shown on the application.

Force data is acquired using accelerometers. There are two accelerometers, one on each strike pad. Each accelerometer has three analog outputs which connect to the analog to digital converter of the ATmega2560 microcontroller. There is an X, Y, and Z output for each accelerometer. Acceleration is found by how fast these coordinates change. The mass of the strike pad is known and is used to convert the acceleration into force. The strike force is equal to the mass of the strike pad multiplied by the acceleration from the accelerometer. Accelerometer will be measured in m/s/s and the force will be measured in Newtons.

The second workout type is the “Force Workout.” Here, after the user selects this workout, they must indicate which strike pad they will use. After indicating, the indicator light will signify the Boxing Buddy is ready for the strike. Then, the user strikes the pad and force is calculated and displayed as stated above.

The last workout type is the “Repetitive Hits” workout. In this workout, the user first inputs a duration in seconds. After the user hits the “Submit” button, a 5 second delay will occur. After the 5 second delay, the user hits the strike pads as many times as possible during the specified duration and the amount of hits is displayed after the duration ends.

The Boxing Buddy is a versatile tool for people who want to box with varying skill levels.

12.0 Appendix

12.1 Bibliography

1. American Psychological Association. (n.d.). *Frequently monitoring progress toward goals increases chance of success*. American Psychological Association. Retrieved December 6, 2022, from https://www.apa.org/news/press/releases/2015/10/progress-goals#:~:text=WASHINGTON%20%E2%80%94%20If%20you%20are%20trying,by%20the%20American%20Psychological%20Association
2. Zhong xin chu ban she. (2019). *Ran Dian: Zheng Zai Fa Sheng de Chuang Ye Shi = startups*. Amazon. Retrieved December 6, 2022, from https://aws.amazon.com/startups/start-building/how-to-build-a-mobile-app/
3. *Competitions, quizzes, hackathons, scholarships and internships for students and corporates*. Unstop. (n.d.). Retrieved December 6, 2022, from https://unstop.com/blog/advantages-and-disadvantages-of-c-programming-language
4. *Top programming languages and their uses*. KDnuggets. (n.d.). Retrieved December 6, 2022, from https://www.kdnuggets.com/2021/05/top-programming-languages.html
5. Kamienski, N. (2022, October 27). *React native vs xamarin: Pros and cons (2022 edition)*. Pagepro. Retrieved December 6, 2022, from https://pagepro.co/blog/react-native-vs-xamarin-pros-and-cons/
6. Strategist, A. G. (2019, November 19). *The 10 best hybrid app frameworks in 2022*. MobileAppAaily. Retrieved December 6, 2022, from https://www.mobileappdaily.com/best-hybrid-app-frameworks
7. Ionicframework. (n.d.). *The cross-platform App Development leader*. Ionic Framework. Retrieved December 6, 2022, from https://ionicframework.com/
8. *Simple.powerful.unobtrusive.* Aurelia. (n.d.). Retrieved December 6, 2022, from https://aurelia.io/
9. Simmons, C. (2022, April 6). *Ionic vs. react native: Performance comparison*. Ionic Blog. Retrieved December 6, 2022, from https://ionic.io/blog/ionic-vs-react-native-performance-comparison
10. *Progressive web apps (pwas)*. MDN. (n.d.). Retrieved December 6, 2022, from https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps
11. Google. (n.d.). *Developer policy center*. Google. Retrieved December 6, 2022, from https://play.google.com/about/developer-content-policy/
12. *What is nosql? NoSQL databases explained*. MongoDB. (n.d.). Retrieved December 6, 2022, from https://www.mongodb.com/nosql-explained

13. F., M., D., G., S., J., Gustavo, & Name *. (2022, November 30). N3 pro. Nexersys. Retrieved December 6, 2022, from https://nexersys.com/product/n3-pro/
14. *ESP32-D0WDQ6 ESPRESSIF systems: Mouser*. Mouser Electronics. (n.d.). Retrieved December 6, 2022, from https://www.mouser.com/ProductDetail/Espressif-Systems/ESP32-D0WDQ6?qs=chTDxNqvsykWgzfXx0gR%252BQ%3D%3D
15. 13, P. J., & 13, D. J. (2022, January 18). *ESP32: Connecting to a WIFI network*. techtutorialsx. Retrieved December 6, 2022, from https://techtutorialsx.com/2017/04/24/esp32-connecting-to-a-wifi-network/
16. Atmell. (n.d.). *Microchip technology*. Mircochip. Retrieved December 6, 2022, from https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf
17. Admin. (2022, May 10). Moment of inertia - formulas, moi of objects [solved examples]. BYJUS. Retrieved December 6, 2022, from https://byjus.com/jee/moment-of-inertia/
18. *Bending stress - definition, formula, assumptions, diagram*. byjusexamprep_img. (n.d.). Retrieved December 6, 2022, from https://byjusexamprep.com/bending-stress-i
19. *Power Plug & Outlet types A & B*. WorldStandards. (2022, January 24). Retrieved December 6, 2022, from https://www.worldstandards.eu/electricity/plugs-and-sockets/ab/
20. Google. (n.d.). *The Collaborative Interface Design Tool.* Figma. Retrieved December 6, 2022, from https://www.figma.com/
21. *Arduino Mega 2560 REV3*. Arduino Official Store. (n.d.). Retrieved December 6, 2022, from https://store.arduino.cc/products/arduino-mega-2560-rev3
22. *VEL05US050-US-MB: Digi-key electronics*. Digi. (n.d.). Retrieved December 6, 2022, from https://www.digikey.com/en/products/detail/xp-power/VEL05US050-US-MB/5023711
23. *Civil Engineering - strength of materials - discussion*. IndiaBIX.Com. (n.d.). Retrieved December 6, 2022, from https://www.indiabix.com/civil-engineering/strength-of-materials/discussion-2397
24. *Applying IPC standards to PCB layout design*. Applying IPC Standards to PCB Layout Design. Retrieved December 6, 2022, from https://resources.pcb.cadence.com/blog/2021-applying-ipc-standards-to-pcb-layout-design
25. *The top 10 microcontrollers for PCB design*. SnapEDA Blog. (n.d.). Retrieved December 6, 2022, from https://blog.snapeda.com/2016/08/04/the-top-10-microcontrollers-for-pcb-designers/
26. Media, O. S. (n.d.). *8 sensor protocols for your next IOT project*. Embedded Computing Design. Retrieved December 6, 2022, from

https://embeddedcomputing.com/technology/iot/8-sensor-protocols-for-your-next-iot-project
27. Teel, J. (2022, December 1). *Review of bluetooth low energy (BLE) solutions (updated 2022)*. PREDICTABLE DESIGNS. Retrieved December 6, 2022, from https://predictabledesigns.com/review-of-bluetooth-low-energy-ble-solutions/
28. *WIFI and WiMAX wireless chips information*. WiFi and WiMAX Wireless Chips Selection Guide: Types, Features, Applications | Engineering360. (n.d.). Retrieved December 6, 2022, from https://www.globalspec.com/learnmore/semiconductors/communications_rf_wireless_chips/wifi_802.16_chips
29. Rodriguez, L. (2017, April 25). *Create an app that collects sensor data*. Creative Bloq. Retrieved December 6, 2022, from https://www.creativebloq.com/how-to/create-an-app-that-collects-sensor-data
30. *Analog | Embedded Processing | Semiconductor Company | ti.com*. (n.d.). Retrieved December 6, 2022, from https://www.ti.com/lit/ds/symlink/msp430fr6989.pdf
31. *Load Cells*. Loadstar Sensors. (n.d.). Retrieved December 6, 2022, from https://www.loadstarsensors.com/products/sensors/load-cells.html#specialty
32. It's better than sinder! (n.d.). Retrieved December 6, 2022, from http://www.circuitronics.org/certifications/
33. Electronics, V. (2021, March 8). *4 things to consider when choosing a PCB manufacturer*. Versa Electronics - Electronic Manufacturing Services. Retrieved December 6, 2022, from https://versae.com/choosing-a-pcb-manufacturer/
34. *What is a Quality Management System (QMS)?* ASQ. (n.d.). Retrieved December 6, 2022, from https://asq.org/quality-resources/quality-management-system
35. *Measuring the force of punches and kicks among combat sport athletes using a modified punching bag with an embedded accelerometer*. measuring the force of punches and kicks among combat sport. (n.d.). Retrieved December 6, 2022, from https://www.researchgate.net/publication/301922720_Measuring_the_force_of_punches_and_kicks_among_combat_sport_athletes_using_a_modified_punching_bag_with_an_embedded_accelerometer
36. Senpai RoySenpai Roy 3111 gold badge11 silver badge33 bronze badges, CR DrostCR Drost 35.9k22 gold badges3737 silver badges104104 bronze badges, Mitch AlexMitch Alex 2122 bronze badges, & HenryHenry 1. (1962, April 1). *How can I determine the pound-force behind a punch?* Physics Stack Exchange. Retrieved December 6, 2022, from https://physics.stackexchange.com/questions/169522/how-can-i-determine-the-pound-force-behind-a-punch