Final Design Report

TEAM 24 – DIGIT

November 30, 2022

Team Members

Bobby Bose	bobbybose@uky.edu
Jewell Catlett	ijca223@uky.edu
Lili Petrowsky	lxpe222@uky.edu
Gwynne Symons Buxton	gesy222@uky.edu

Course Advisors

Dr. Regina Hannemann	Regina.hannemann@uky.edu
Dr. Robert Adams	Robert.adams@uky.edu
Dr. Samson Cheung	Sen-ching.cheung@uky.edu

Teaching Assistants

Seun Alo	seun.alo@uky.edu
Cole Cuddy	coleton.cuddy@uky.edu
Daniel Muchow	daniel.muchow@uky.edu

Project Sponsor

Madison Bates	Madison.bates@uky.edu

Mentors

Dr. Steven Shake	steveshake@aol.com
Dr. Sridhar Sunderman	ssu223@uky.edu

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 ABSTRACT

This paper presents an analysis of the final design of the medical sensory glove designed by Team DIGIT. Team DIGIT designed and constructed a sensory glove for the purpose of tracking real-time motion of the fingers using sensors, to better assist doctors with monitoring and helping patients with hand impairments. The glove is intended to be used in sessions with patients, who will wear the glove and be instructed to perform specific movements using their hand. The glove aims to improve on prior versions, by reducing the form factor of the circuity and maintaining constant connectivity to the glove. The report begins with the problem statements and requirements that were developed at the beginning of the project to guide Team DIGIT in the design and build of the glove. The only change to the requirements was removing the wrist movement from marketing requirement one after talking to our sponsor, as the tracking of wrist movement was important for the previous versions of the glove for basketball shot tracking but is not necessary for the medical sensory glove. Subsequently, the subsystem design, testing and design summary are discussed.

2 PROBLEM STATEMENT

2.1 Need

There is a need to help monitor patients who have impaired motor function in their hands, such as those who have suffered a stroke, spinal cord injuries, or have other ailments that affect hand motor skills. It is important for medical professionals to be able to collect hand motion data to assess a patient's hand motor skills in real-time. By utilizing hand motion data, medical professionals could assess the current condition of a patient or determine the impact of treatments or injuries. Medical professionals need the ability to instruct patients to perform certain hand motor functions and record their movements using means that minimally impact motion while simultaneously monitoring the data in real-time.

2.2 Objective

The goal of project Sensory Glove V4 was to implement this glove as a medical device for doctors to use during therapy sessions for patients struggling with impairment in the hands. The design objectives include tracking motion in the fingers, recording and storing data for comparison in data analysis, displaying data, and prompting commands. Team DIGIT's technical solution for these design objectives were to (1) create a PCB capable to support sensors, (2) record and capture data with the sensors on the PCB then transfer data wirelessly to a secondary device, (3) develop a user-friendly GUI.

2.3 Background

It is estimated that one in four adults over the age of twenty-five will have a stroke in their lifetime [1] with 50% to 70% experiencing movement impairment [2]. These numbers only pertain to hand paresis from strokes and do not account for paresis caused from carpal tunnel surgery and other neurodegenerative disorders such as MS, ALS, and Parkinson's. Physical and occupational therapy is a common practice to help patients try to regain movement, however depending on the severity of the injury/disorder, the increased out of pocket expenses can put patients in financial jeopardy. Depending on if the patient has insurance, each therapy session can be as low as $15 or up to around $250 if uninsured [3]. Additionally, the therapist may recommend the purchase of equipment to aid in recovery at home.

There are currently some therapy gloves in the market geared toward rehabilitation of hand movement such as the MusicGlove [4], the Neofect Smart Glove [5] , and the CyberGlove [6]. These products can be used to improve hand function recovery by fostering motor learning and brain reorganization. While these products can be useful, they can be awfully expensive, which limits their accessibility to the public.

The previous iterations of the project were inherited by Team DIGIT however they were geared to be for basketball players to improve their game. These gloves had the main objective of shot tracking to improve their game and shooting form by comparing a shot made by an individual to a previously set benchmark shot [7]. Team SCREEN in 2021 improved upon the previous iterations developed by teams S.W.I.S.H in 2019 and S.H.O.T. in 2020 by creating a glove that was smaller and less obstructive than the previous iterations and measured data in a more concise manner. Team SCREEN used a fingerless glove, which was a stretchy knit winter glove [8]. Team SCREEN had a wireless glove that was powered with two-coin cell batteries and had a Bluetooth module to transmit data from the sensors to the application.

Team DIGIT's glove is meant for rehabilitation for impairment in the hands and needed to be improved to fit the needs and objectives from above. Our sponsor would like for the glove to eventually be used at the same time as an EEG cap to compare the data from movement and electrical brain activity. Since the glove will be used simultaneously with the EEG cap, the data needs to be recorded in real time to a secondary device to ensure that the two sets of data can be compared with each other. Our sponsor is currently using VRFREE gloves by sensoryx [9] for the current stage of her study. However, the gloves are bulky and hard to put on patients that have very rigid hands. The gloves occasionally stop sending data to the computer, creating lags in the data. The sponsor wants a game to coincide with the glove that provides instructions for movements, feedback based on progress, and goals for patients to reach.

3 REQUIREMENTS SPECIFICATION

3.1 Marketing Requirements

The marketing requirements were created after looking at the previous iterations of the glove and meeting with our sponsor to determine what she wanted in the glove.

1. **Shall** track finger motion
2. **Should** be wireless
3. **Shall** include a goal-based game
4. **Should** send data to 3D modeling software
5. **Shall** display data from sensors in real-time
6. **Shall** be easy to put on and remove
7. **Shall** be cleanable
8. **Shall** not interfere with patient's movement
9. **Should** have a rechargeable battery

3.2 Objective Tree

The objective tree, shown in Figure 3.1, was created from the Marketing Requirements. Weights were assigned to each grouping, using the AHP pairwise comparison tables shown in Appendix A.

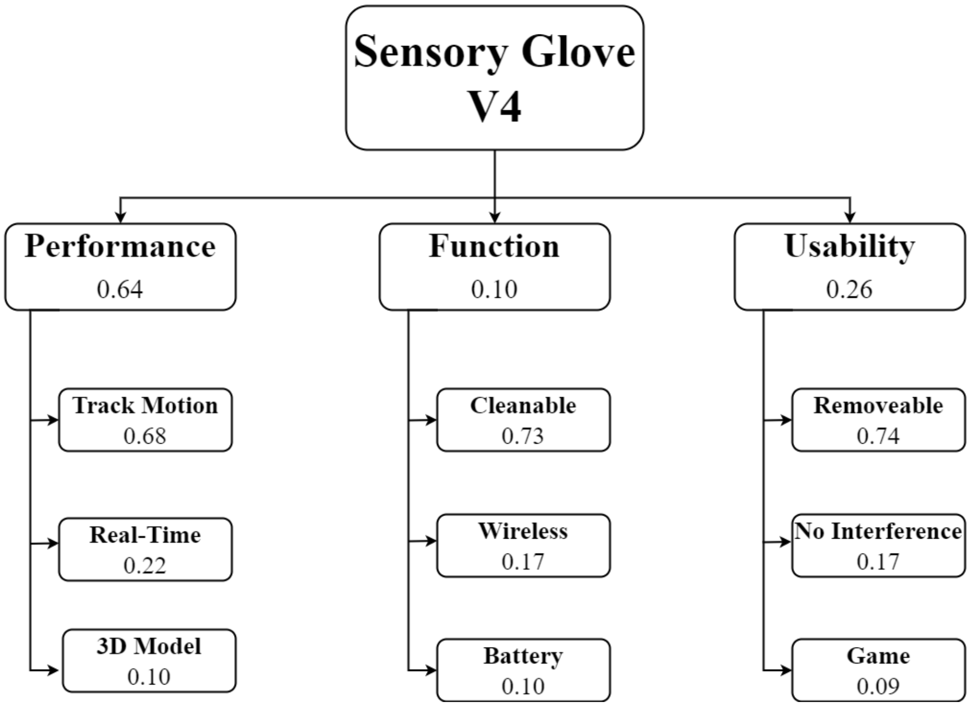


Figure 3.1 Objective Tree

3.3 Engineering Requirements

Table 3.1 defines the Engineering Requirements (ER) related to this project. Each Engineering Requirement represents an attribute of the glove and/or software that is derived from at least one of the Marketing Requirements. The corresponding justifications are provided to clarify the importance of the Engineering Requirements to the overall project.

The Engineering Requirements are highlighted to show the status of each requirement. The rows colored in green indicate the requirement has been tested and verified to work; Yellow rows indicate the requirement as implemented but does not quite meet the specification; Red rows indicate the requirement was not fulfilled. The details of the testing can be found in Section 5 Testing and Validation.

Table 3.1 Engineering Requirements

No.	MR	Engineering Requirements	Justification
1	2, 9	Battery should last 5-8 hours while powering the PCB	For the glove to be wireless, a battery with a long life is required. To ensure that the glove does not shut down during consecutive data collecting sessions that last one hour, the battery life should last 5-8 hours

No.	MR	Engineering Requirements	Justification
2	2, 5	Should monitor battery level	By monitoring the battery level, the user will be able to detect low power in the battery. This is to ensure the glove does not shut down during data collection.
3	3	Shall be programmed in LabVIEW, Python, or MATLAB.	The sponsor required that our code is written LabVIEW, MATLAB, and Python.
4	1, 5	Sensors shall be connected to the ADC pins on the PCB	Sensors hooked up to ADC pins will allow the microchip to detect changing voltage with analog to digital conversion (ADC). The ADC pins allow a range of voltage to be read compared to the digital pins that only read two values.
5	8	PCB shall be smaller than the previous PCB design (1.95” x 2”)	Designing a smaller PCB will make it easier to put on or take off the glove especially if the patient is rigid from stroke or disability.
6	8	PCB shall contain the control system and connections to all sensors on a single PCB	Designing a PCB to contain all the electronics removes extra weight
7	5	Data shall be streamed in real-time	Allows for immediate feedback based on results.
8	4, 5	Should display 3D motion	A visual method to view and discuss results with patients.
9	6, 7	Circuity shall be removable	Helps with cleaning and working with circuitry while the glove is currently not being used.
10	1	Shall contain a minimum of 8 sensors	This allows motion tracking with all 5 fingers and adduction/abduction movement. Our sponsor will be using 5 flex sensors and 3 pressure sensors.
11	4, 5	Shall achieve a sampling rate greater than 10 Hz	The sponsor is currently unable to achieve a sampling rate higher than 10 Hz with her prototype. She has requested that we attempt to achieve higher rates to track hand motion more precisely.
12	2, 9	The glove should use Bluetooth or Wi-Fi	Both are affordable and high-speed data transmission technologies. Either option will allow us to have a high sampling rate, as indicated in ER 11.
13	2, 9	Should be powered by a rechargeable battery	Using a rechargeable battery reduces our overall waste expenditure and removes the need to buy and store disposable batteries. The battery will be removed and charged via a micro-USB charger
14	1, 3	The secondary device shall provide positive feedback when the patient correctly follows commands	Allows patients to receive quick input regarding whether they are following directions correctly.
15	6, 8	The glove shall be made from material that allows flexibility with different hand sizes	Allows the glove to fit over a variety of patients' hands.

No.	MR	Engineering Requirements	Justification
16	1, 4, 5	Data from the glove shall be sent to a secondary device	Allows the user to look at the collected data from the glove while the patient is actively using the glove. This removes interference with additional wires.
17	1, 3	The secondary device shall display commands to instruct the patient to perform specific tasks to test hand movement	Simple visual commands issued through the game will provide an easy method to communicate instructions to the patient during testing.
18	6, 8	The overall weight of the glove shall not be more than 4oz	The sponsor’s glove with the electronics weighs over 8 oz. The goal for this requirement was to cut the weight of the glove and electronics in half. A lightweight glove will allow the user to effectively use the glove and move more freely.
19	6, 7	The circuity shall be encased	Allows for easier removal and reattachment of the circuit to the glove while preventing the patient from scraping or poking themselves on the PCB components.

3.3.1 Verification of Engineering Requirements

Verification ensures the design meets the requirements and specifications. Table 3.2 describes how each Engineering Requirement will be verified for the glove prototype.

Table 3.2 Verification of Engineering Requirements

No.	ER	Verification
1	Battery should last 5-8 hours while powering the PCB	Charge battery to full and then leave the PCB powered for up to 8 hours. Confirm after 5 hours that the PCB is still powered.
2	Should monitor battery level	Confirm that the battery level can be viewed through observation by either software or hardware.
3	Shall be programmed in LabVIEW, MATLAB, or Python	Confirm that all code written is in either LabVIEW, MATLAB, or Python.
4	Sensors shall be connected to the ADC pins on the PCB	Wiring the sensors to ADC pins then read in changing voltage with the software using ADC.
5	PCB shall contain a PCB smaller than the previous PCB design (1.95” x 2”)	Measure both PCB designs and compare both designs to see if the new PCB is smaller than 1.95” x 2.”
6	PCB Shall contain the control system and connections to all sensors on a single PCB	Observing that all connections lead back to the PCB (No additional small PCB breakout boards for sensors).
7	Data shall be streamed in real-time	Move sensors on the glove and see if data shows up on the secondary device simultaneously.
8	Should be displayed in 3D motion	Check that the hand model matches the current position and orientation of the actual hand being tested.
9	Circuity shall be removable	The circuitry will be detached and reattached to the glove. Then the glove will be turned on and tested to see if it is working.

No.	ER	Verification
10	Glove shall contain a minimum of 8 sensors	Test each sensor and be able to read the data.
11	Shall achieve a sampling rate greater than 10 Hz	Check how many data points the glove can receive in one second of testing.
12	Glove should use Bluetooth or Wi-Fi	Check that glove has the capability to transmit data over either technology.
13	Glove should be powered by a rechargeable battery	Deplete the battery completely, and then charge until full again and check the battery level to confirm.
14	Secondary device shall provide positive feedback when the patient correctly follows commands	Run an instance of the game and test that when a command is followed correctly; positive feedback is issued on screen by the game.
15	The glove shall be made from material that allows flexibility with different hand sizes	Test flexibility by putting the glove on various hand sizes and flexing the fingers and rotating the wrist.
16	Data from the glove shall be sent to a secondary device	Test the Bluetooth/or Wi-Fi connection from the primary device to the secondary device.
17	Secondary device shall display commands to instruct the patient to perform specific tasks to test hand movement	Run an instance of the game and check to make sure commands are being issued continuously as the patient completes previous tasks.
18	The overall weight of the glove shall not be more than 4oz	Weigh the glove on a scale with all components attached.
19	The circuitry shall be encased	Check that all circuitry except for sensors and associated connecting wires are packaged in a case

3.4 Design Impact Statements

3.4.1 Environmental

The impact of technology on our environment is a growing issue in the modern world. Electronic waste is a term used to describe electrical devices that are at or near the end of their lifetime. It is possible to dispose of this waste in specific ways to allow it to be recycled and reused in the future. However, most people do not do this. Since this product is intended to be designed to potentially be marketed to the general public for at-home use, it cannot be expected that everyone will comply with electronic disposal procedures. Therefore, to mitigate our electronic waste, we have set requirements to try and help our glove be more eco-friendly. All circuitry on the glove is removable, in order to make sure the glove can be fully cleaned. This gives us the ability to reuse the glove and eliminates the need to buy new gloves after each session and/or for each patient. In addition, we had planned to use a rechargeable battery. This would have eliminated the need to constantly dispose of old batteries and would reduce our waste. Although we did not fulfill this requirement, we instead chose to use the connected computer as our power source for the device. Since this does not require disposing of old power sources, we still reduce our over electronic waste that could be generated.

3.4.2 Manufacturability

The sponsor and mentor have expressed their desire for the glove to be made a product in the market. The following are the constraints for manufacturability:

1. The design will consist of accessible components in the market so that future replications of the device are identical to the prototype. The parts picked for the prototype will allow the sponsor and mentor to choose different sizes and type of connectors. For example, the on-board connectors are a 2mm pitch male connector. Any other 2mm pitch male connector may be replaced with the original connector.
2. The construction process of the design will be documented in detail for the sponsor and the mentor to have the capability to replicate the device. The detailed documentation will include ECAD files, bill of materials (BOM), block diagrams to determine which sensor connects to which pin for the software development, and instructions for assembly of the device.

3.4.3 Social

Although this sensory glove's intended use is for sensing by medical professionals, making this glove easy to use and easily accessible will allow a wide variety of people to take advantage of this technology. In addition to medical professionals using the glove in the office or at home for diagnosis or physical therapy, a commercial market could also be possible. For example, self-monitoring using devices such as watches is becoming very popular for health and fitness applications. If hand motion data can be demonstrated as an accurate early indicator of issues and, if appropriately priced and marketed, individuals could be motivated to collect hand motion data for themselves. While it would not necessarily be appropriate to provide a diagnosis, recommending consulting with a medical professional would be an option. Furthermore, while our glove game application is intended to help with data collection, it could spawn a new application for general gaming.

3.4.4 Standards

Standards ensure that the product works together and follows organization regulations to avoid product failure. The medical sensory glove will follow the standards from the International Electrotechnical Commission (IEC) such as standards IEC 62304 about medical device software [10] and IEC 60601-1-11 for medical electrical equipment [11]. Standards from the Institute of Electrical and Electronics Engineers (IEEE) will also be respected including the IEEE 1233 Guide for Developing System Requirements Specification [12] and IEEE 802.15.1-2005, Standard for Information technology [13]. The glove will also be created keeping in mind the standards from the Association for the Electronic Interconnection Industry (IPC), specifically IPC-6011, the Generic Performance Specification for Printed Boards [14] and IPC-2221A, the IPC Standards for PCBs [15]. Finally, with the intention of eventually getting FDA approval so that the glove can be sold to mass markets and used in therapy centers, the FDA standards for medical devices [16] will also be respected.

3.4.5 Economic

Healthcare equipment can be very expensive, which directly impacts accessibility, and this lack of accessibility can negatively impact patient care. Developing a sensory glove that is inexpensive benefits both medical facilities and patients. Inexpensive equipment is accessible to a greater number of care providers which, in turn, provides more accessibility to a greater number of patients while also reducing overall costs. Because the glove is intended to be used as medical equipment, it is important that it provides accurate and reliable data. Inexpensive sensors and materials need to be thoroughly examined to determine if they will be accurate, repeatable, and reliable. Other aspects of cost are reusability and repairability. If cheaper sensors are accurate but less durable, it may be possible for sensors to need replacing after one or a few uses. Providing a mechanism for component removal and replacement would also allow the glove to be disassembled for cleaning and reuse.

4 DESIGN

4.1 Design Summary

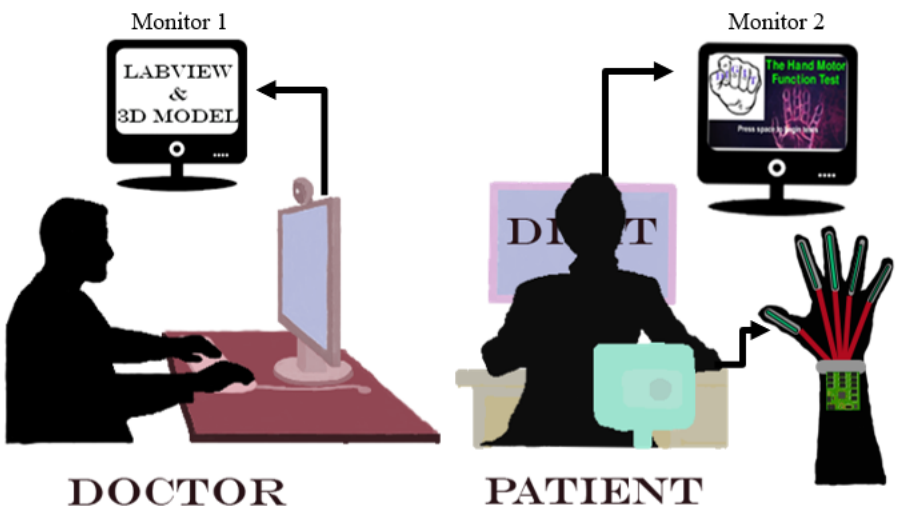


Figure 4.1 Testing Setup

The Medical Sensory Glove is intended to be used in therapy sessions with a patient. During testing, it is interfaced with an external EEG cap that is used to monitor brain activity correlated with hand movement data. The cap is not being created by this team and will be connected later. The secondary device, which is a computer, is required for the sessions. Figure 4.1 illustrates the setup in the doctor's office for a session. Patients will be asked to sit in front of a computer monitor, where the user interface, or game, is available for the patient to interact with during testing. The doctor administering the tests will be seated separately with their own monitor in front of them, where they can view test data in real-time. Both monitors will be connected to the secondary device, which will allow the doctor to interact with the game as well if necessary. Typically, each session lasts 2 hours, but can vary per patient.

Before testing, patients will put on the glove, which is also connected to the secondary device. How long this takes very much depends on the patient and their hand impairment but combined with the time it can take to set up the software, this can be anywhere from 10-20 minutes. Randomly selected tasks will be chosen for the patient to try and complete and will be displayed on the patient's screen. They will have 8 seconds to attempt each task. Data from the sensors on the glove will be continuously streamed in real-time during this period. If the patients are successful in completing a task, they are awarded points by the game. Although the tasks are shown in succession, patients will have an opportunity to rest between test suites. At the end of the test session, the point total will be displayed on screen, to allow the patient to see how they performed overall.

As part of the initial design of the project, we were asked to create a 3D model of the hand using the sensor data, as indicated by Engineering Requirement 8. After initial research and the first half of our development time, we decided not to focus on this anymore, due to the time it would take to learn how to develop it, and the other, more essential requirements that needed to be completed. Despite this, the Engineering Requirement has been kept in order to indicate for future users or development teams, that this is something the sponsor was interested in.

4.2 Functional Decomposition

4.2.1 Level 0

Figure 4.2 Medical Sensory Glove Level 0

Level 0 of the functional decomposition visually describes the overall functionality of the medical sensory glove which is displayed in Figure 4.2. The functional requirements are given in Table 4.1. Finger movements are inputs into the system that are used to determine the bend of the fingers of a patient. The outputs of the system are a 3D model, battery level, movement data, feedback, and the next task. All five outputs are used for immediate feedback to the patient and data analysis to the therapist.

Table 4.1 Medical Sensory Glove Functional Decomposition Level 0

Module	**Medical Sensory Glove**
Inputs	• Finger Movement
Outputs	• 3D Model • Battery Level • Movement Data • Next Task • Patient Feedback
Functionality	Records finger motion which is then stored as a text file for data analysis. Then the data is used for a command-prompt game.

4.2.2 Level 1

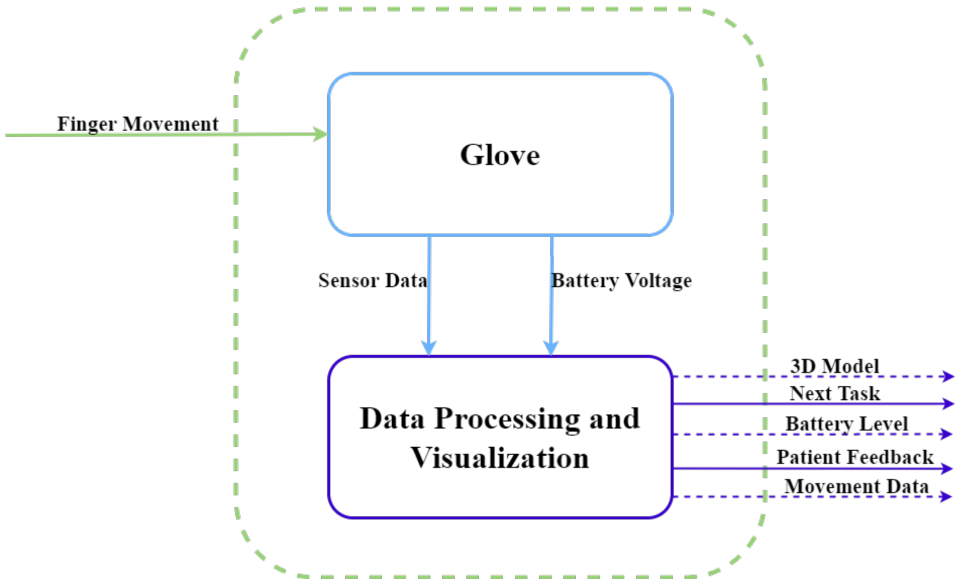


Figure 4.3 Sensory Glove Level 1

Level 1 of the functional decomposition visually describes the overall functionality of the medical sensory glove broken down into two sub systems which are displayed in Figure 4.3. Finger movements are inputs into the glove subsystem that are used to determine the bend of the fingers. Then, battery voltage and sensor data are outputs of the glove system and inputs into the data processing and visualization subsystem. The outputs of this subsystem are a 3D model, battery level, movement data, patient feedback, and the next task. The outputs with the dashed purple line go to the therapist or doctor's computer while the solid purple lines will go to the patient's computer. The functional requirements for Level 1 subsystems of the glove and data processing are given in Table 4.2 and Table 4.3.

Table 4.2 Glove and Accompanying Electronics Functional Decomposition Level 1

Module	Glove and accompanying electronics
Inputs	• Finger Movement
Outputs	• Sensor Data • Battery Voltage
Functionality	Measures finger movement, collects the raw data, and then transmits the data to the secondary device.

Table 4.3 Data Processing and Visualization Functional Decomposition Level 1

Module	Data Processing and Visualization
Inputs	• Sensor Data • Battery Voltage

Module	Data Processing and Visualization
Outputs	• 3D Model • Next Task • Battery Level • Patient Feedback • Movement Data
Functionality	Sensor data is processed in LabVIEW and then sent to the game's code. The game first compares the data to a benchmark, and then displays the results on screen for the patient.

4.2.3 Level 2

Level 2 of the functional decomposition separates the glove and the secondary device into their respective subsystems, as depicted in Figure 4.4 and Figure 4.5. The glove block in Figure 4.4 takes in the finger movement from the sensors and sends the sensor signals to the PCB. The PCB is powered from the battery if the glove is wireless or through a USB cable that is connected to the computer if the glove is wired. The PCB then powers the sensors while sending the sensor data and battery level if applicable to the secondary device. The functional requirements for the level 2 glove are in Table 4.4, Table 4.5, and Table 4.6.

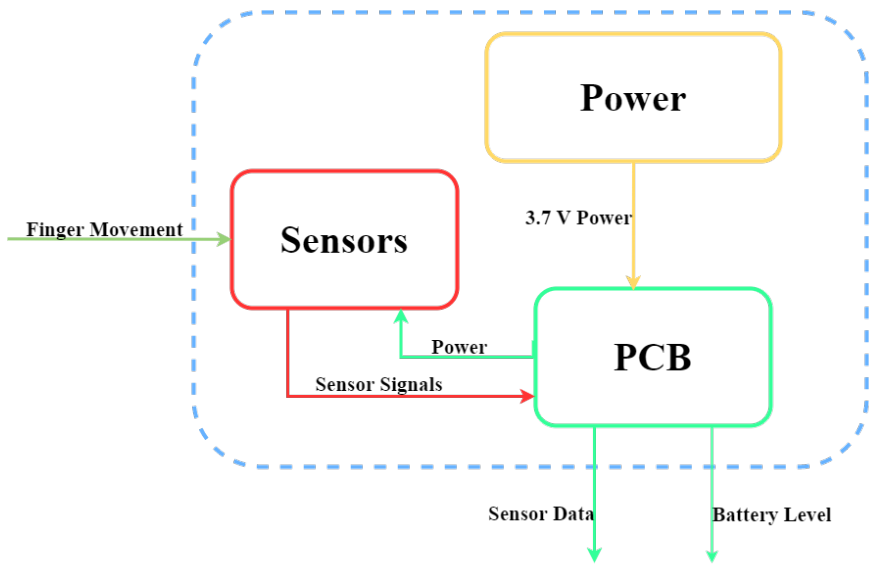


Figure 4.4 Glove Level 2

Table 4.4 Power Functional Decomposition Level 2

Module	Power
Inputs	• N/A
Outputs	• 3.7V Power
Functionality	Provides 3.7 volts of power to the PCB

Table 4.5 PCB Functional Decomposition Level 2

Module	PCB
Inputs	• 3.7V power • Sensor signals
Outputs	• Sensor Power • Sensor Data • Battery Level
Functionality	The PCB collects data from the sensors and sends that data to a secondary device.

Table 4.6 Sensors Functional Decomposition Level 2

Module	Sensors
Inputs	• Finger Movement • Power
Outputs	• Sensor Signals
Functionality	Measures finger movement and outputs voltage signal sensors to microcontroller

The data processing and visualization block in Figure 4.4, takes in the raw data from the sensors and the battery voltage, which is sent wirelessly over Bluetooth or Wi-Fi (depending on which option is ultimately chosen). The data is first processed in the Data Processing block, in LabVIEW, and then sent to the User Interface block, where it is compared to a benchmark for the current task. The results of the current test, as well as the next task to complete, are outputted from this block, through the patient's screen. In addition, the Data Processing block also outputs the current battery level, as well as the 3D model representation of the patient's hand movement data. The functional requirements for the level 2 Data Processing and User Interface are given in Table 4.7 and Table 4.8

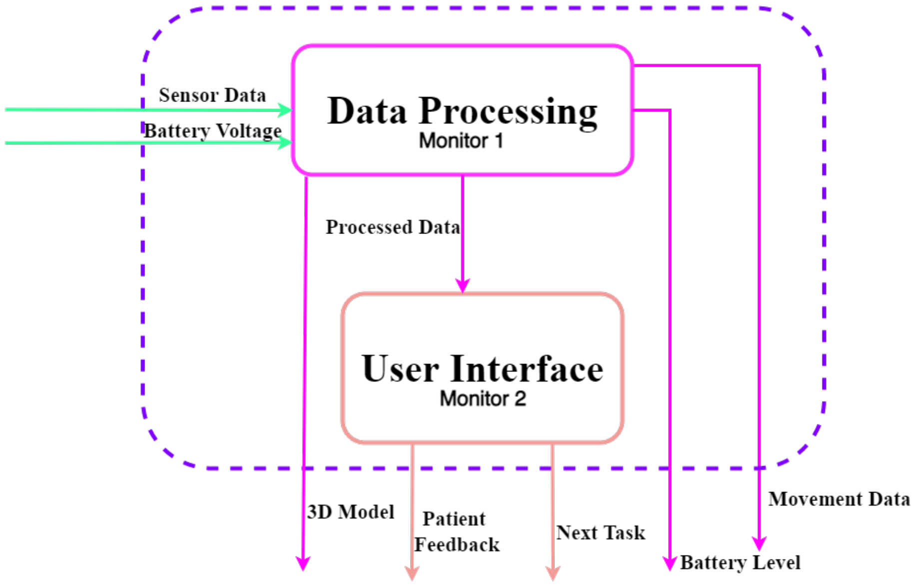


Figure 4.5 Data Processing and Visualization Level 2

Table 4.7 Data Processing Functional Decomposition Level 2

Module	**Data Processing**
Inputs	• Sensor Data • Battery Voltage
Outputs	• Movement Data • 3D Model • Battery Level
Functionality	Takes data from the sensors on the glove then processes and calibrates the data to be used later.

Table 4.8 User Interface Functional Decomposition Level 2

Module	**User Interface**
Inputs	• Processed Data
Outputs	• Patient Feedback • Next Task
Functionality	Handles the comparison of test data to benchmarks, and manages the game used for command issuing

4.3 Software Flow

Figure 4.6 shows the overall flow for patient testing. The code for the game can be found on the team's GitHub [17]. When the session starts, the game begins on the title screen, displaying the game title. Patients and doctors then have the option to either set benchmarks for the test or use preset default benchmarks. If they choose to set their own benchmarks, then they are taken through each test in the game one by one and asked to perform the correct hand motion. Their hand position data is recorded and saved in an external file. After this, the patient then has the option to proceed with the game, or exit.

Once testing starts, a help screen is first displayed, showing information about how testing works, and how to earn points. Once the patient is ready, the first test suite and first task are loaded, and the game moves on to a countdown screen, letting the patient know what task is about to be assessed. Once the countdown finishes, the task is displayed on screen for a predetermined amount of time. During this time, the sensors on the glove are continuously sending data from the PCB to LabVIEW, where it is first written to a file, and then read in from the game's code. Within the code, the sensor data is compared against the benchmark for the task. If the patient is successful in performing the task, they are shown a task success screen, and are awarded points. Bonus points are earned based on how fast a patient is able to complete a task. This scales linearly with the time remaining when a task is completed.

The point system is used to provide positive feedback for the patient and encourage them to keep going. If the patient fails, the game simply moves on without displaying any kind of failure screen. This is in order to align with our goal of avoiding any negative feedback to make sure the patient does not feel discouraged. After all tasks have been completed, patients have the option to either repeat the test suite to try to perform better, move onto the next test suite if there are any remaining, or exit the game. Once all testing is finished, the final screen is displayed, with the patient's total points shown on screen.

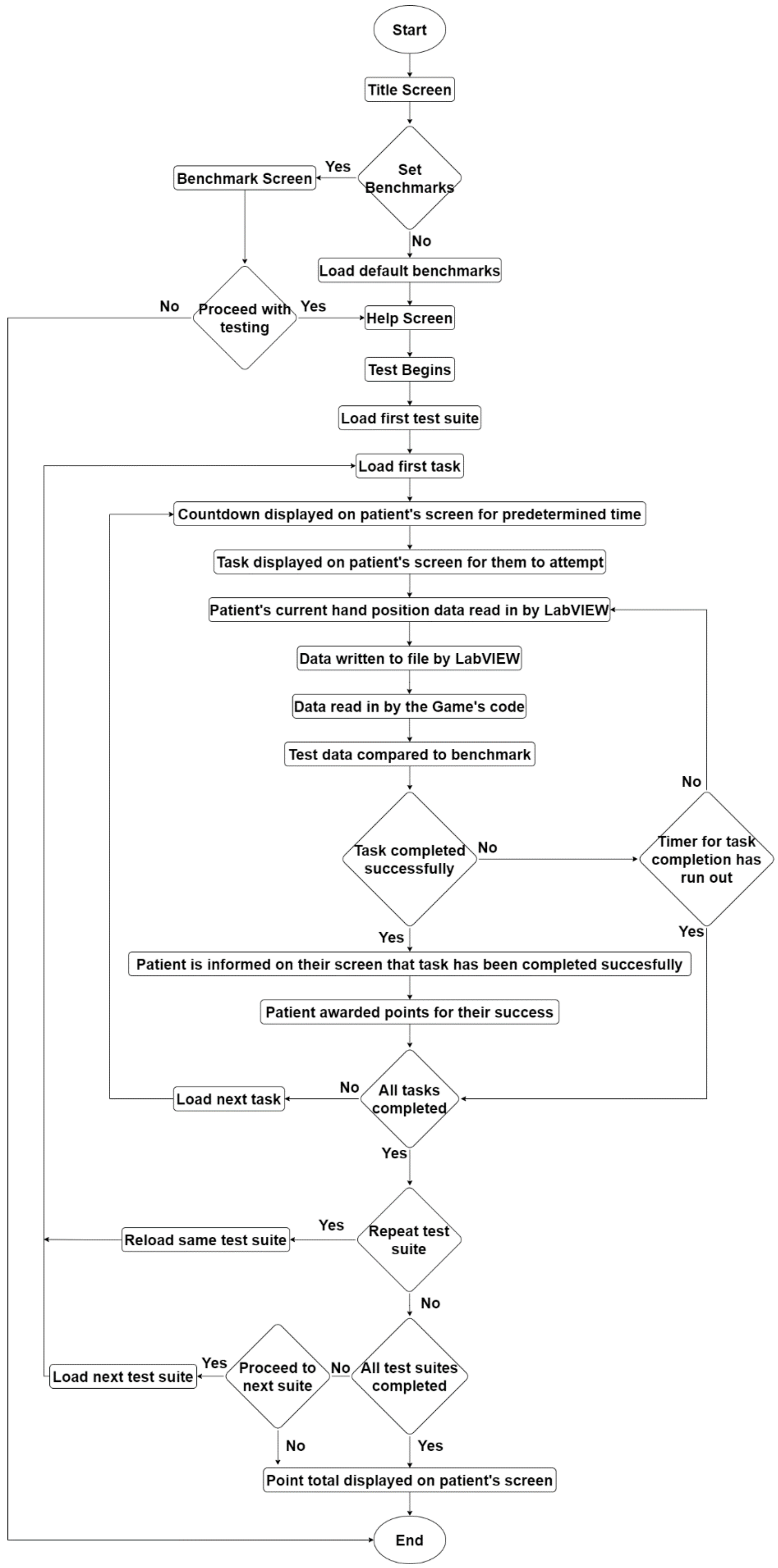


Figure 4.6 Overall Test Flow

4.4 Final Design

The following sections breakdown each part of the glove and describe the final design choices made. Appendix B provides the design alternatives that were researched along with more AHP tables that were used to aid in decision making.

4.4.1 Glove Design

4.4.1.1 Material of Glove

The material of the glove needs to be flexible enough to wear different hand sizes can wear it and it is easy to put on and take off especially for patients with rigid hands. The ranking of importance of the attributes for the material of the glove were done using the AHP pairwise comparison Table 4.9. The main priority is that the glove and its material do not cause any impairment or interference with movements. The next two priorities are the flexibility and smoothness of the glove. The glove should not irritate patients' skin while performing tasks and should fit to their hand size. While the cleanability of the glove is not super important especially if it goes to mass market and people buy it just for themselves, but it is important for therapist and doctors' offices. The cost is the lowest priority compared to the other properties.

Table 4.9 Material of Glove AHP

Rank		Flexibility	Smoothness	Cost	Cleanable	No Impairment	GM	Weight
2	Flexibility	1.00	2.00	8.00	4.00	1.00	2.30	0.314
3	Smoothness	0.50	1.00	7.00	3.00	0.33	1.28	0.175
5	Cost	0.13	0.14	1.00	0.25	0.11	0.22	0.030
4	Cleanable	0.25	0.33	4.00	1.00	0.11	0.52	0.071
1	No Impairment	1.00	3.00	9.00	9.00	1.00	3.00	0.410
Sum		2.88	6.47	29.00	17.25	2.55	7.32	1.00

Shore hardness is used to measure and indicate the flexibility or stiffness of a material [18], with materials being on three different scales depending on the material as seen in Figure 4.7. Shore 00 Hardness Scale is for rubbers and gels that are exceptionally soft like a marshmallow or gummy candy. Shore A Hardness Scale is for flexible mold rubbers that can be anywhere from unbelievably soft and flexible to hard and almost no flexibility, similar to a pencil eraser, rubber bands and a bottle nipple. The final scale is the Shore D Hardness Scale which is the hardness of hard rubbers, plastics, and semi-rigid plastics like PVC pipes and golf balls. Two of the most common filaments for 3D printing are ABS and PLA [19], both of which tend to be on the Shore D scale, which would be too hard for a glove and not flexible enough to do simple hand movements. TPE filament, thermoplastic elastomers filament, is a wider classification for soft materials and the popular flexible filament for 3D printing. TPE can be broken down into six different classes including TPU, TPC, and TPA [20].

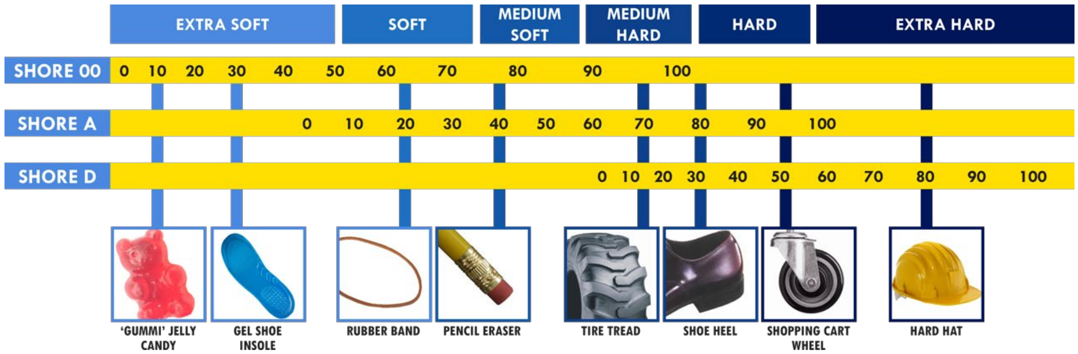


Figure 4.7 Shore Hardness Scales [18]

Before creating the final design of the glove, the material of the glove needed to be chosen. One design was for each material to ensure that the criteria is met with there are no external factors being taken into consideration such as each design being varied sizes or completely different. To test each material, an apple watch band design has been created in Autodesk Fusion 360 [21] as shown in Figure 4.8 following the steps outlined from Learn Adafruit [22] to see how each material wraps and feels around the wrist while doing simple tasks that will be provided during the session in the game. After printing with multiple brands of TPU and flexible PLA+, the team narrowed the type of filament down to the Amazonbasics TPU for the PCB case as it was slightly stiffer and the Polymaker Polyflex TPU for the fingers as it was very smooth and flexible. The 3D printing settings for both filaments can be found in the GitHub 3D-Design Repository [23]. The Polymaker Polyflex TPU95.ini and Amazonbasics TPU.ini can be imported into the slicing software.

Figure 4.8 Apple Watch Band Design Created in Fusion 360 [24]

To meet Engineering Requirement 19, the glove needed to contain a case for the PCB. The case was 3D printed using the Amazon TPU filament. Figure 4.9 illustrates what the PCB case looks like in Fusion 360. The PCB case was custom designed around the PCB design and was created using the YouTube tutorial for a 3D Printable Raspberry Pi Case [25] as a guide. Figures 4.10-4.12 are of the PCB case in Fusion 360 drawing to display the measurements of the case in inches.

Figure 4.9 Fusion 360 Design of PCB Case [23]

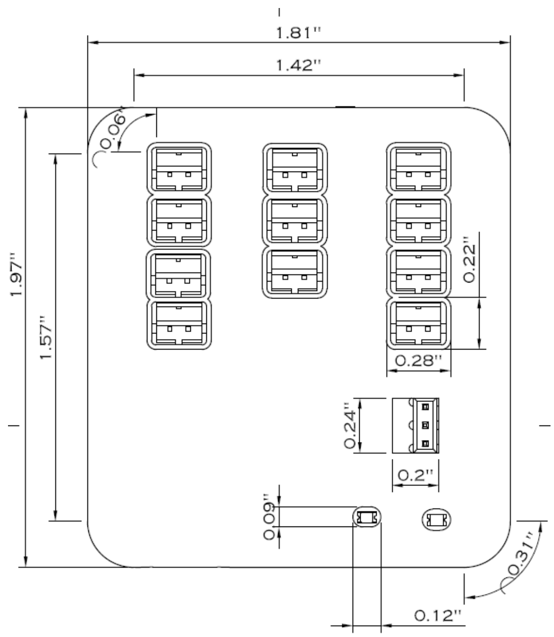


Figure 4.10 PCB Case Drawing Top View [23]

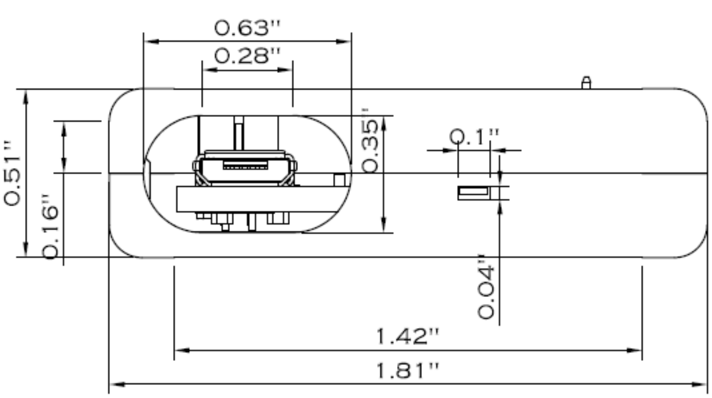


Figure 4.11 PCB Case Drawing Front View [23]

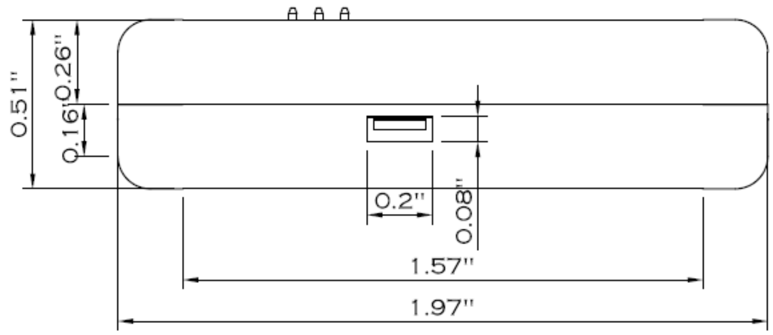


Figure 4.12 PCB Case Drawing Right View [23]

4.4.1.2 Design of Glove

Team DIGIT changed the type of glove being made from a basketball shot tracking glove to a medical sensory glove for stroke and hand impairment rehabilitation. The previous iteration of the glove by Team SCREEN was made with a fingerless glove, which was a stretchy knit winter glove [8]. Our sponsor is currently using VRFREE gloves by sensoryx [9] in Figure 4.13 for the current stage of her study. However, the gloves are bulky and hard to put on patients that have very rigid hands. Depending on the severity of a patient's hand impairment, it can be difficult to put the glove on. After talking with our sponsor, she expressed an interest in having an exoskeleton glove that would just be on top of the hand with straps to hold the glove to the hand around each finger. Table 4.10 is showing the three main characteristics that the glove should have. With our sponsor putting emphasis on how hard it can be to put a glove on a patient that has limited mobility, ease of putting the glove on is weighted the most. Due to every patient having different sized hands, flexibility of the glove to fit multiple sizes and stretch over a hand while not being too tight was the next most important characteristic of the glove. Finally, the glove should be comfortable and not cause chafing or itchiness to the patient while doing the tasks asked of them as they will be wearing the glove for at least two 60-minute sessions.

Figure 4.13 VRFREE Glove by sensoryx [9]

Table 4.10 Design of Glove AHP

	Flexibility	Ease of putting on	Comfort	GM	Weight
Flexibility	1.00	0.25	2.00	0.79	0.218
Ease of putting on	4.00	1.00	3.00	2.29	0.631
Comfort	0.50	0.33	1.00	0.55	0.152
Sum	5.50	1.58	6.00	3.63	1.00

Table 4.11 compares an exoskeleton glove, fingerless glove, and fingered glove based on the characteristics in Table 4.10. For each characteristic, each glove is numbered one through three with one being the best and three being the worst of the three. When the columns are averaged, it shows that given the characteristics Team DIGIT is looking for in a glove, the exoskeleton is the best, followed by the fingerless glove.

Table 4.11 Glove Design Comparison

	Exoskeleton	**Fingerless**	**Fingered**	**Weights**
Flexibility	1	3	3	0.218
Ease of putting on	1	2	3	0.631
Comfort	2	1	1	0.152
Average	1.33	2	2.33	

To start out with designing the glove, different types of existing 3D printed hands/gloves were researched on Thingiverse [26] and Printables [27]. The Flexi Hand [28] shown in Figure 4.14 is a popular design when searching for flexible hands. While this is not a glove it was used to get an idea of how to print something that bends like the fingers on a hand should. The team considered making a thinner version of the flexi hand to sit on top of the hand but struggled with figuring out how to secure it to the hand of the patient. The next thing researched was 3D printed gloves which provided a lot of gauntlet type designs [29] like Figure 4.15 and Figure 4.16 however most of the designs were to worn with the person's hand in a fist sitting in the palm area of the 3D printed glove. The other type of design that results from searching 3D printed gloves is the Iron Man Glove [30] shown in Figure 4.17. This design is intended for a person to be able to put their whole hand inside and move around with no trouble. However, this design would be hard to make so that it fits multiple sizes of hands as it would most likely be too tight or too loose on different patients if printed as one size, but the wrist and palm part of the design provided ideas for ways to encase the PCB while keeping it out of the way of the patient.

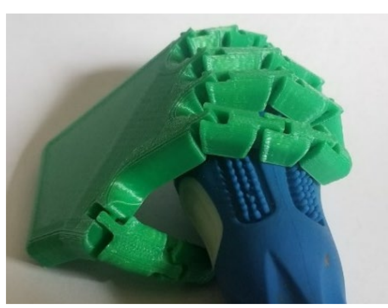


Figure 4.14 Flexi Hand [28]

Figure 4.15 Gauntlet Glove [29]

Figure 4.16 Gauntlet Glove Palm Up [29]

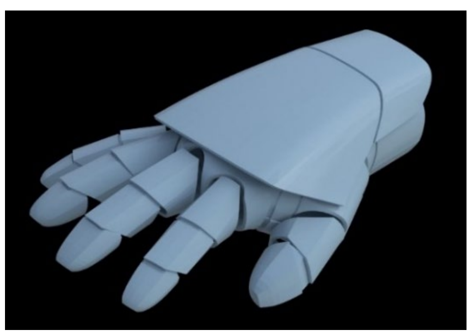


Figure 4.17 Iron Man Glove [30]

Our sponsor, Madison Bates, shared some research articles and publications that she had read about 3D printed gloves that were really useful because they were designed specifically for medical uses like what Team DIGIT is trying to create. The most helpful article that she provided showed that what Team DIGIT was trying to create was possible and helped aid in creating a better design than what Team DIGIT had been creating. The article was from an IEEE Engineering in Medicine & Biology Society Conference [31] which showed the 3D printed exoskeleton glove made from TPU in Figure 4.18. This glove has the sensors and wires embedded into the material as they paused the printer to insert the flex sensors and wires. While this is an amazing concept, they took multiple measurements on one person's hand so the glove could be custom fitted and the wires and sensors could be the exact length that is needed. Team DIGIT was not able to embed the sensors or wires into the 3D print as the glove needs to fit various sized hands, but this article provided ideas of how to hold the sensors to the patient's hand and make it flexible at the same time.

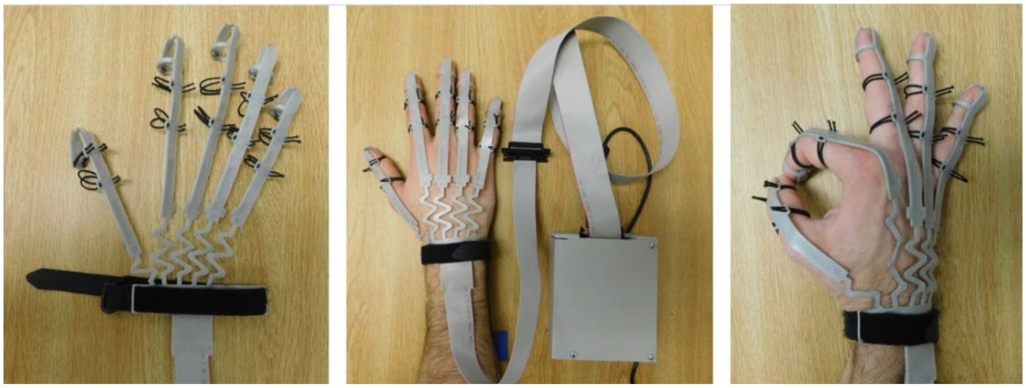


Figure 4.18 3D Printed Exoskeleton Glove [31]

The fingers that Team DIGIT designed were created in Autodesk Inventor [32] with each finger having two medium pegs and one small peg that connect to the top of the finger to create a finger cap as seen in Figure 4.19. As stated in section 4.4.1.1 Polymaker Polyflex TPU was used to 3D print the fingers. Figure 4.20 illustrates all the finger designs imported into Fusion 360 to see them as one design. Figure 4.21through Figure 4.27 are the drawings of each finger and peg to show the dimensions of each part.

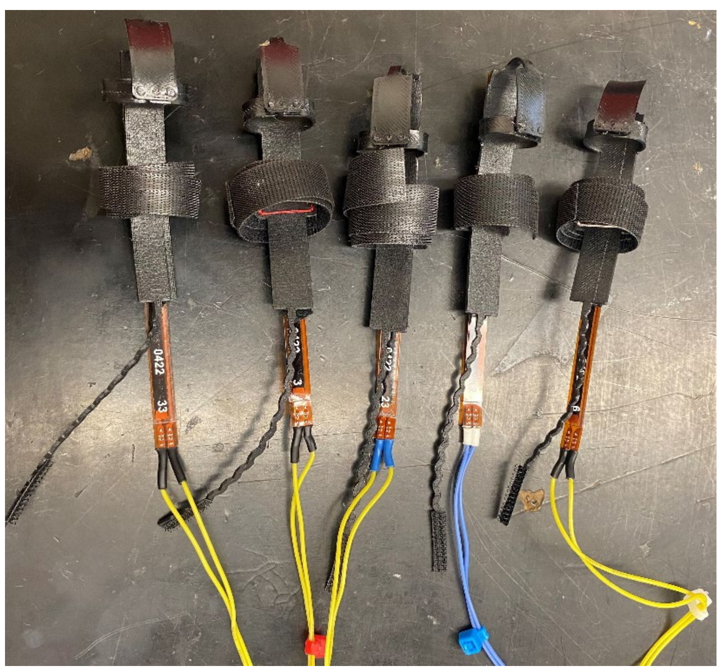


Figure 4.19 Physical 3D printed fingers with finger caps connected from left to right: Thumb, Pointer, Middle, Ring, Pinky

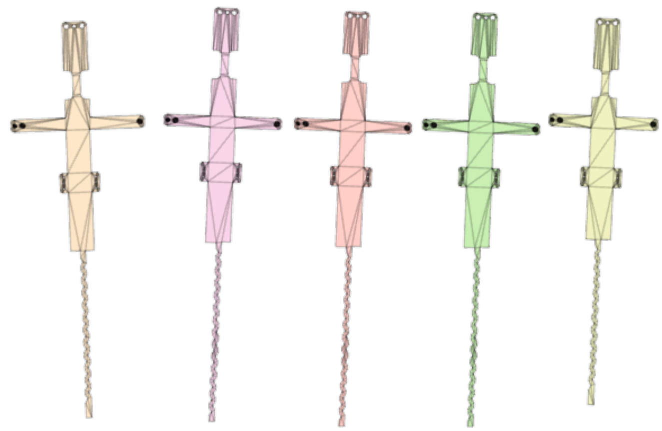


Figure 4.20 From Left to Right: Thumb, Pointer, Middle, Ring, Pinky [23]

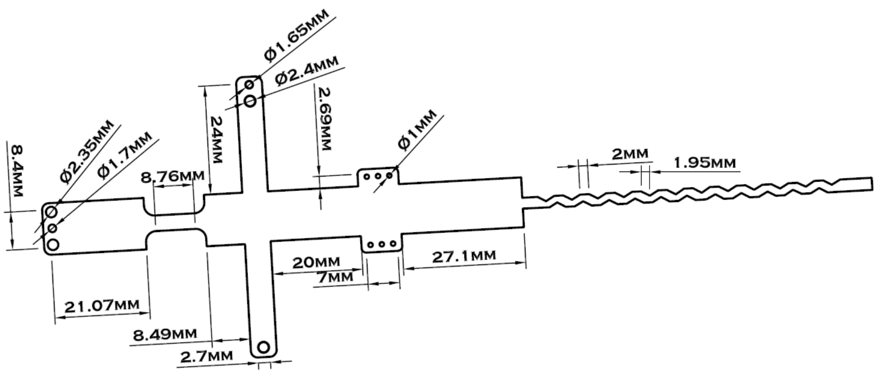


Figure 4.21 Thumb Finger Drawing [23]

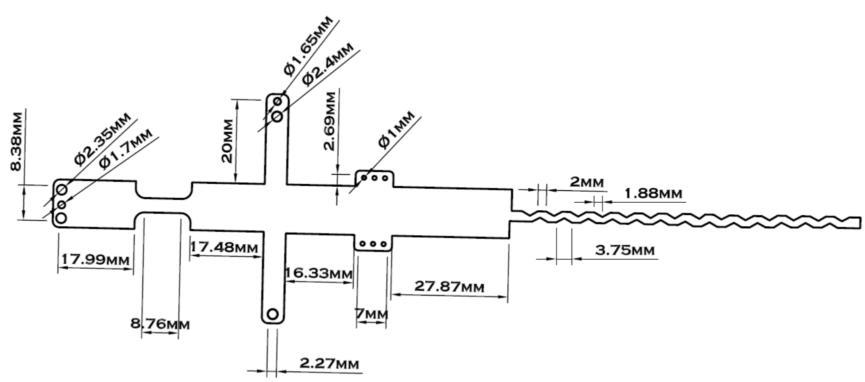


Figure 4.22 Pointer Finger Drawing [23]

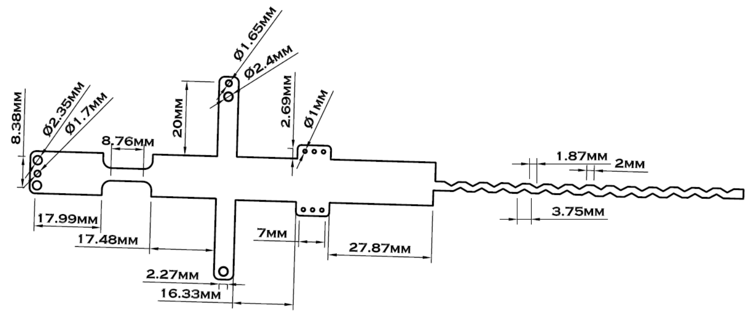


Figure 4.23 Middle Finger Drawing [23]

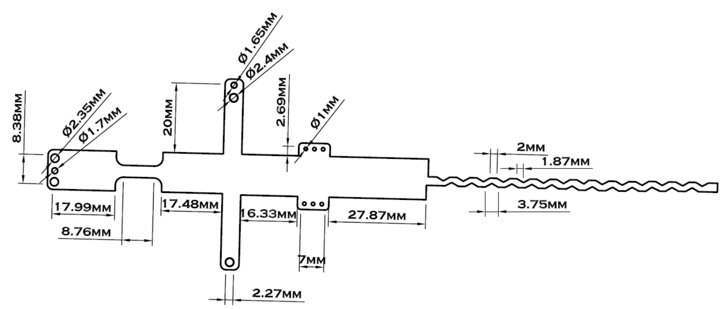


Figure 4.24 Ring Finger Drawing [23]

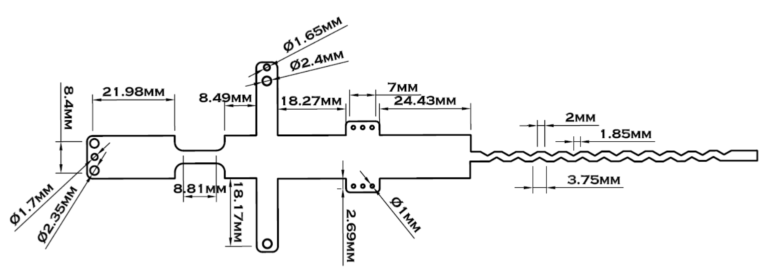


Figure 4.25 Pinky Finger Drawing [23]

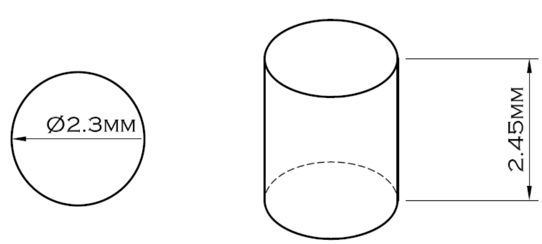


Figure 4.26 Medium Peg Drawing [23]

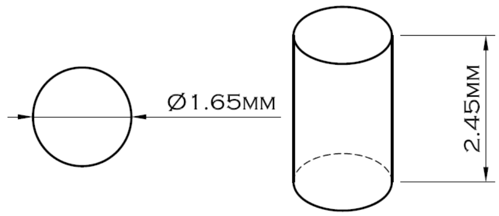


Figure 4.27 Small Peg Drawing [23]

Due to time constraints, Team DIGIT was unable to fully 3D-print an exoskeleton glove. As a result, gloves and wrist wraps that athletes and those that need wrist support use were researched. The Neo G Active Wrist Support glove seen in Figure 4.29 was easy to attach all the components on with Velcro however since the medical sensory glove will be used for stroke patients that have hand impairment, Team DIGIT decided it may be too cumbersome to put on and take off. Figure 4.28 and Figure 4.30 are both a similar design where the thumb is inserted into a hole and then the wrap is wrapped around the wrist which are ideal as they can fit various sized hands. The McDavid Sport Wrist Wrap in Figure 4.28 had a softer and smoother texture that will not cause any irritation so that is the wrap Team DIGIT chose to use. Utilizing a hand sewing kit, Velcro was sewn onto the wrap in the approximate locations that each finger and sensor would be placed to act as a strap so the wires and fingers would be attached to the glove. This makes it easier for the glove to be put on and stored as there are not multiple loose pieces that could get lost. Figure 4.31 shows the wrap laid out flat with all the components attached in their proper place while Figure 4.32 illustrates what the glove looks like on a person's hand.

Figure 4.28 McDavid Sport Wrist Wrap

Figure 4.29 Neo G Active Wrist Support

Figure 4.30 Futuro Wrist Performance Comfort Support

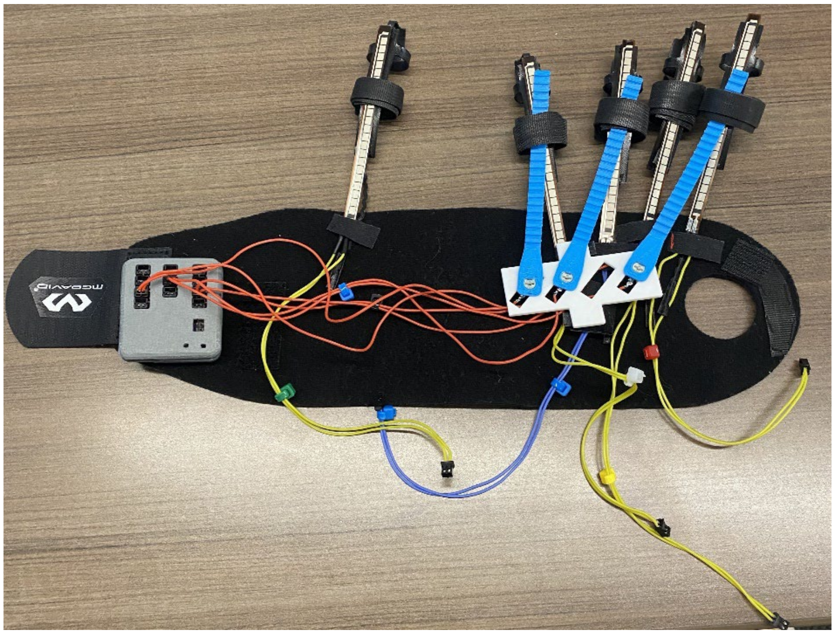


Figure 4.31 Wrist wrap laid flat with PCB case, sensors and 3D printed pieces connected

Figure 4.32 Glove on a team member

4.4.2 PCB

The PCB design targets to meet Engineering Requirements 3, 4, 5, 6, 9, 10, and 18. Since Engineering Requirements 1 battery should last 5-8 hours and Engineering Requirement 2 should monitor battery level are shoulds, this design is focused on producing a stable and cost-efficient custom PCB that satisfies the shalls of the Engineering Requirements. Therefore, the PCB will be a wired connection and therefore will be powered through a USB. The goal was to design and create a small-scale light-weight custom PCB that LabVIEW [33] can target. The sponsor of team DIGIT, Madison Bates, used National Instrument's USB-6211 [34], which is a Multifunction I/O Device, that weighs a total of 7.2 oz. This device was far from meeting the engineering requirement 18. However, a LabVIEW add-on was discovered which allows LabVIEW to target Arduino boards. Additionally, Arduino offers Autodesk Eagle [35] CAD files to the community so that Arduino based custom PCBs can be developed. There are many different types of Arduino microcontrollers to consider. A few worth noting is Arduino Micro [36], Arduino Nano [37], Arduino Uno Rev 3 [25], and Arduino Nano 33 BLE [38].

As mentioned earlier, this design focused on the shalls of the Engineering Requirements. Engineering Requirements 1 and 2 are shoulds that implement a wireless battery powered PCB. Since the patient using the medical sensory glove will be seated and wired up to an EEG cap, a wireless PCB wouldn't make a huge difference. Moreover, the wireless PCB needed to be operated at 3.3V to keep the weight of the battery relatively low so that it is not intrusive to the patient. As an example, a 9V battery weighs about 1.2 oz while a 3.7V 500 mAh battery weighs 0.7 oz. Implementing a 9V battery into the design adds weight to the glove. In addition, a 9V battery would result in less power efficiency of the PCB since the voltage would need to drop from 9V to 5V as the ATMega32U4 operates from a range of 2.7V to 5V. Operating the PCB at 3.3V would be more efficient for power efficiency. Furthermore, there is concern for the battery life and stability of the clock speed of the microchip as the battery depletes. For example, according to the ATMega32U4 datasheet, the operating voltage of a 16Mhz crystal requires a clean 4.15V minimum. Therefore, the clock speed of the microchip would need to be downgraded to a much lower crystal. The next standard external crystal is 8MHz. The minimum operating voltage for this clock speed is 2.4V (keep in mind the ATMega32U4 requires a minimum of 2.7V). The last concern for this design approach is the battery life. To calculate the battery life, the total current consumption would need to be calculated. A Bluetooth module to connect to the software to send data readings. The Bluetooth module that was considered for this design was the Adafruit Bluefruit SPI Friend [39] that consumes around 15mA. The flex sensors from SpectraSymbol have a flat resistance of 9k ohms have a current draw of 0.17368 mA as seen from the Multisim [40] simulation in Figure 4.33. Lastly, the ATmega32U4 can consume up to 200 mA from the power source. Therefore, the total current draw would be $I_{total} = 11\ x\ 0.17368 + 15\ mA + 200\ mA = 216.91\ mA$. Equation 1 is used to calculate the total battery life with the total load.

$$Battery\ Life = \frac{Total\ rated\ mAh}{Current\ Load\ (mA)} \qquad (1)$$

Using the battery life equation, Equation 1, the total battery life of a 500mAh would be $Battery\ Life = \frac{500mAh}{216.91mA} = 2.3\ Hours$ until the battery starts to deplete. While this would enough for the ATMega32U4 and 8 MHz crystal, the concerning part is whether the 3.3V LDO regulator will have enough power. The maximum dropout voltage (the voltage needed to produce a clean 3.3V output voltage) for an LDO regulator such as a MIC5225-3.3 is 450mV. If the design used this regulator, the regulator would need an input voltage of 3.75V (3.3V + 0.45V = 3.75V) from the battery to have a clean voltage output. Therefore, a larger battery voltage would be required to have a wireless PCB, and that means adding more weight to the glove. Furthermore, there are LDO regulators with lower maximum dropout voltages, but these regulators are more costly. Therefore, the entire design would be more expensive. So, the medical sensory glove consists of a wired PCB to satisfy the engineering requirements.

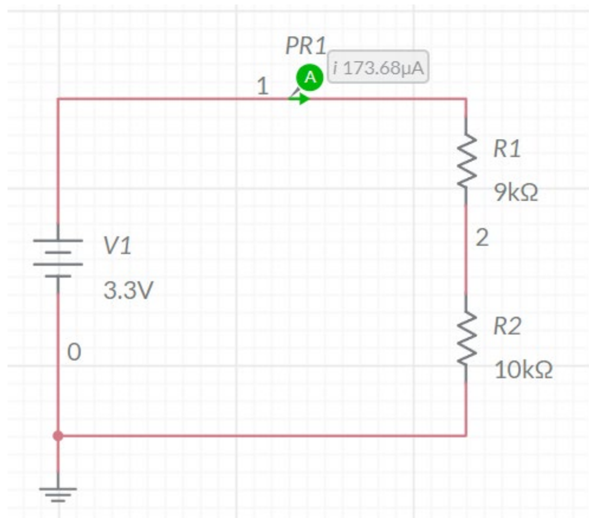


Figure 4.33 Multisim Flex Sensor Simulation

4.4.2.1 Microcontroller

The brain of a PCB is the microchip. When designing a custom PCB to fit the needs of the requirements and the sponsor, a few things had to be considered. First, the number of ADC pins were taken into consideration. The sponsor requested a minimum of 8 sensors but plans to upgrade to 10 sensors to record and store more data from the hands. Since the PCB is wired to a computer via USB, the operating voltage range is not a big factor in choosing a microchip since the USB provides a consistent 5V to the system. Next, the microchip must be LabVIEW compatible (able to be targeted by the software). Lastly, the microchip must have a built-in UART to save space and lower the cost for buying a separate UART chip. Table 4.12 summarizes the different specifications for each microchip along with the Arduino controllers that uses it. Given the table below, the microchip that satisfies these requirements is the ATMega32U4. The AHP table, Table B.10 in Appendix B shows the importance of each feature.

Table 4.12 Processor Chip Comparison

Processor	Operating Voltage	ADC Pins	LabVIEW Compatibility	Built-In UART
ATMega32U4	2.7V - 5.5V	12	Yes	Yes
nRF52840	1.7V - 5.5V	8	Yes	Yes
ESP32-PICO-D4	2.7V - 3.6V	18	No	No
ATMega328p	1.8V - 5.5V	6	Yes	Yes
ATMega328	1.8V - 5.5V	8	Yes	Yes

The custom ATMega32U4 TQFP 44 PCB was designed to contain and measure 11 resistive sensors. The overall design of the PCB consisted of components selected to operate the medical sensory glove efficiently and safely. The following is a detailed description of each component selected.

Each sensor for this project has a 2-pin, 2mm pitch male, wire-to-board connector. This connector is in a voltage divider circuit to calculate the resistance of the sensor. The fixed resistor in the voltage divider circuit is a surface mount 0603 resistor. Given the component size selected is a market standard size, this gives the sponsor the ability to select different fixed resistor values for the voltage divider circuit. The sensors for the medical sensory glove are flex sensors and potentiometers. This design includes a 3x2 header pins for AVR boot-loading in the case that the PCB malfunctions. To enable this feature, the pins SCK, MISO, and MOSI must be accessible. The pin AVCC must be connected to a low pass filter to enable the supply voltage pin for all A/D converter channels on the ATMega32U4. The PCB has two 0603 LED chips used as indicators for debugging purposes. The on-board USB reciprocal is connected a 2A fuse in the case of a surge from the PCB to protect the computer. Additionally, two varistors are connected to the data pins of the USB as ESD protection to protect the device from current surges. Varistors are voltage-dependent resistors. When the voltage across the varistor exceeds a certain value, the resistance falls to a low value and allows a current to flow, thus providing over-voltage protection to the device. Furthermore, the PCB contains a 16 MHz external crystal. An external crystal is what determines the clock speed of the microchip. This crystal requires an operating voltage of at least 4.15V to stabilize the external crystal. Additionally, this circuitry also has two decoupling capacitors on the crystals to keep the power to the crystal stable. Figure 4.34 is the Eagle schematic [41]. Figure 4.35 is the Eagle board layout, and Figure 4.36 is the 3D model of this PCB.

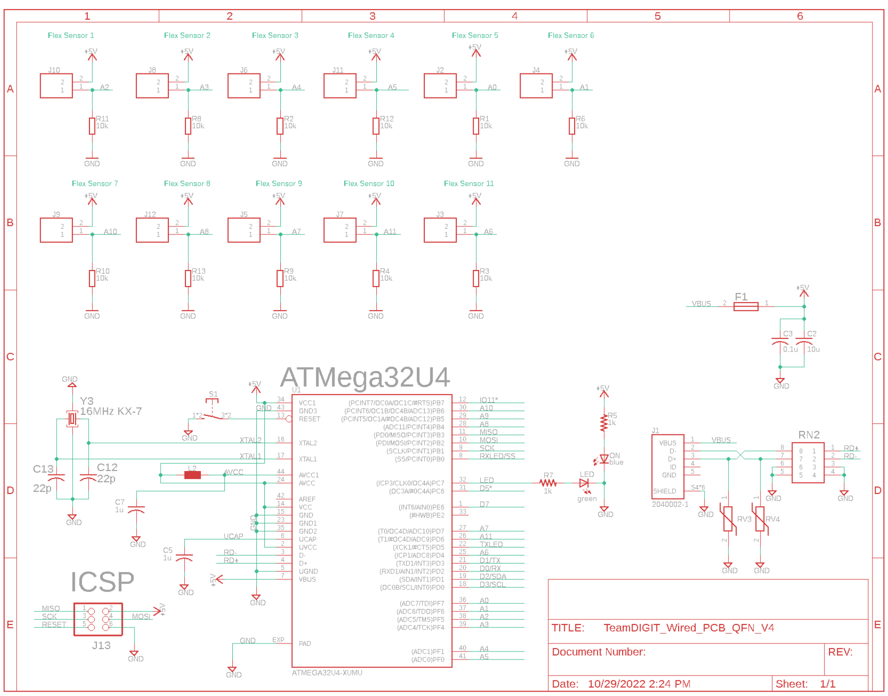


Figure 4.34 Eagle Schematic [41]

The PCB layout [41] of this design is a 4-layer board with dimensions 1.4” x 1.7”. The goal was to design a board that is within the dimensions of 1.95” x 2.0” since that was the previous version’s dimensions. To achieve this, the board needed to be 4 layers.

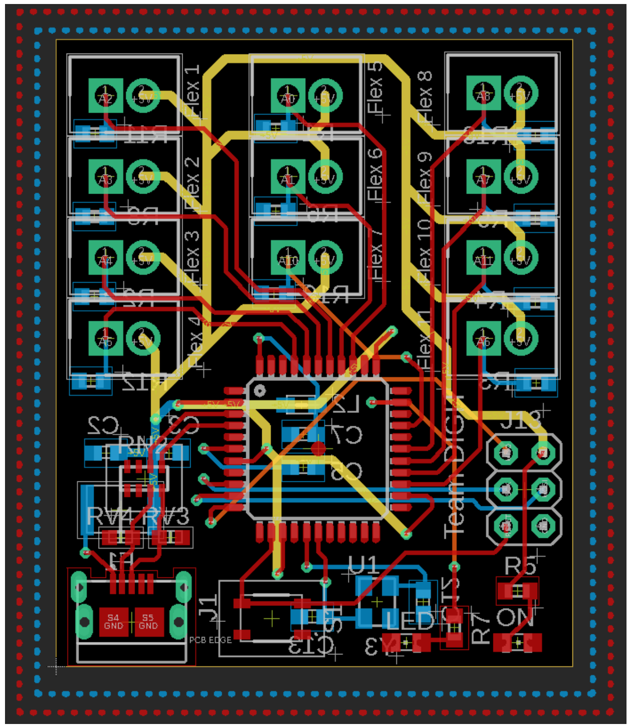


Figure 4.35 Eagle Board Layout [41]

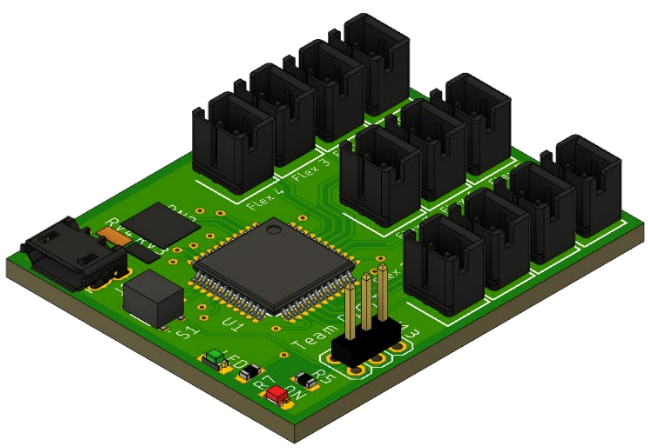


Figure 4.36 3D Model of PCB [41]

4.4.3 Power

For the power supply, it needed to be able to supply enough voltage to power the components on the PCB and all the sensors for consecutive sessions which each last around one hour. The ranking of importance of the attributes for the power supply, were done using the AHP pairwise comparison in Table 4.13 below.

Table 4.13 Power Supply Criteria AHP

Rank		Weight	Rechargeable	Cost	Voltage	Capacity	GM	Weight
4	Weight	1.00	0.33	7.00	0.33	0.25	0.72	0.108
1	Rechargeable	3.00	1.00	6.00	2.00	3.00	2.55	0.383
5	Cost	0.14	0.17	1.00	0.14	0.14	0.22	0.033
3	Voltage	3.00	0.50	7.00	1.00	1.00	1.60	0.241
2	Capacity	4.00	0.33	7.00	1.00	1.00	1.56	0.235
Sum		11.14	2.33	28.00	4.47	5.39	6.65	1.00

For the overall project, an environmental design impact was discussed on how we hoped to minimize our electronic waste by using rechargeable batteries if the PCB was going to be wireless. For this reason, the rechargeability of a battery is a number one priority. The nominal voltage and capacity that a battery has are almost tied in weight of importance with capacity being slightly higher due to the fact that the glove needs to last consecutive 60-minute sessions. The weight of the battery should not dramatically increase the weight of the glove once everything is connected which is why is has a lower ranking. Our sponsor did not indicate how much an individual glove shall cost to create, so the cost of the battery is the least important. In addition, since the battery needs to be rechargeable, it will cost more since rechargeable have a higher price than non-rechargeable ones and a charge is needed in order to recharge the battery which adds to the price.

While there a plenty of rechargeable batteries available, the battery needs to be able to provide the voltage and capacity needed while adding as minimal weight as possible and not interfering with users' movements. The Table B.4 compares different power supply options in approximate weight (g), approximate size, cost of individual, nominal voltage (V), nominal capacity (mAh), and if it is rechargeable. When it comes to small rechargeable batteries, the main two options are lithium-ion batteries or coin cell batteries. Both of these have maximum nominal voltages around 3.7 volts.

Initially, the PCB was going to need at least seven volts of power which would have required at least two batteries in order to reach seven volts or need a voltage regulator module to increase the output power. However, it was discovered that voltage regulators drain the battery faster, especially for a small battery like a 3.7 volt, so in order to be effective, at least nine volts is recommended for a voltage regulator. The teammate working on the PCB, was able to redesign a wireless PCB that ~~will~~ only need around 3.3 volts of power, so that only one battery is needed, and no voltage regulator is needed. The five options from Table B.4 have a nominal voltage of around 3.6 to 3.7 volts; however, their weight and nominal capacity are widely varied. As discussed in section 4.4.2, Team DIGIT has determined it is not reasonable to make the glove wireless since the patient is already sitting down in front of the computer and will be connected with multiple wires for the EEG cap.

4.4.4 Sensors

Various sensors were investigated to determine if they would be good for measuring the bend of fingers and/or the separation of fingers. The sensors examined for finger bending motion included flex sensors which were used on the previous version of the glove and a Conductive Rubber Cord Stretch Sensor from Adafruit [42]. For finger separation, the sensors that were investigated were two range finding sensors, the Adafruit Time of Flight Distance Sensor [43] and the Adafruit Proximity and Lux Sensor [44], and a potentiometer-based sensor. Flex sensors have been chosen to be used on the glove to measure the finger bending motion.

The flex sensors were chosen to measure finger bend. Flex sensors were the easiest to test and implement into the glove. They have repeatable measurements for relative motion. Five flex sensors, one for each finger, were tested to verify their ability to measure the movement of each finger. These five flex sensors were set up and connected to the Arduino micro. Then, as seen in Figure 4.37, five traces were displayed of the voltage level of each sensor. This testing was performed using MATLAB. The code for testing the sensors can be found in the GitHub repository, Sensor Conditioning, named “arduinoplot2.m” [45]. Each sensor trace has a spike that correlates to when the flex sensor is bent. It should be noted that the sensor voltage levels and offsets are at different levels. This is not unexpected, and the actual levels are irrelevant because the voltage changes are the main measurement criteria. The flex sensors were chosen as the sensor to measure finger bending based on the criteria in the AHP Table B.1 in Appendix B. The flex sensor compared to the other sensors can be seen in Table 4.14. One consideration was the size of the sensor. The flex sensors are very thin and lightweight, which allows it to measure finger bend without interfering with the patient’s movement. The flex sensors also functioned well and were easy to connect to the Arduino micro allowing finger bend data to be brought into MATLAB and LabVIEW and easily displayed on a plot in real time. Although flex sensors are not the cheapest sensor, they do not cost an extravagant amount. Because these sensors are pretty durable this helps justify the slightly more expensive cost. This sensor durability, along with the fact that the physical connections to the flex sensors themselves were also more durable, indicates that they won’t need to be fixed or replaced as often. In contrast to this, the rubber stretch sensors were very difficult to connect leads to and they ended up having a lot of connectivity issues which would manifest as a reliability problem especially compared to the flex sensors. Since the flex sensors were tested while wired, the amount of power needed for them was not observed. This is not an observation that is currently needed since this sensory glove does not need to be wireless and the other criteria were more important.

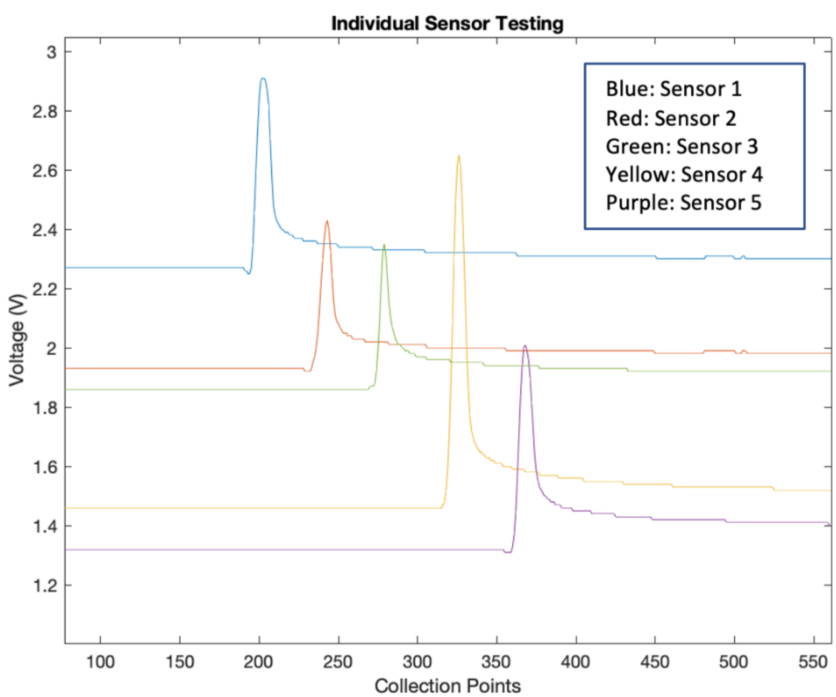


Figure 4.37 Testing 5 flex sensors, one for each finger, for measurement for finger bend

Table 4.14 Sensor Comparison for Measuring Finger Bend

	Flex Sensors	**Rubber Stretch Sensors**
Functionality	Good performance and connectivity to PCB	Difficult and poor connectivity to PCB
Size	Small, lightweight, and flat, flat shape is easier to put on glove	Small, lightweight, and cylindrical, rounded shape harder to put on glove
Cost	Slightly more expensive ($12.95)	Slightly cheaper ($9.95)
Power	Not currently applicable	Not currently applicable

The potentiometer-based sensors were chosen to measure finger separation. The potentiometer-based sensor, which can be seen in Figure 4.38, is composed of a potentiometer with a 3D printed structure (white piece) designed to attach to the glove and the sensor in order to hold the potentiometer in place while another flexible 3D printed torque arm (blue piece) is attached to the top part of the potentiometer. This torque arm, which can be seen in Figure 4.39, is designed to be flexible, so the patient is still able to bend their finger with as little interference as possible. The 3D drawing of this torque arm, ”PotentiometerTorqueArm.stl”, can be found in the GitHub repository labeled Sensor Conditioning [45]. The stabilizing piece, seen in Figure 4.40, is used to hold to potentiometer in place so the top can accurately rotate. The 3D drawing of this stabilizing piece, ”PotentiometerStabilizingPiece.stl”, can be found in the GitHub repository labeled Sensor Conditioning [45]. This sensor is a self-designed sensor since sensors of a similar design that could measure finger separation with many different movements were not found. The torque arm was designed to be longer and have ridges across the length of the piece. These ridges were designed so that the torque arm is even more flexible along the length of the finger. These ridges act as a joint allowing the piece to bend more easily. The ridges are cylinder shaped, rather than triangular, to evenly distribute the stress from bending the piece so that the piece does not break. The piece can now bend completely in half very easily. The structure of this piece was tested while attached to the potentiometer on a finger, the torque arm was still able to move the potentiometer when the finger moved. Potentiometers between the ranges of 10-100kΩ were tested to determine which value provided the best response on the plot with minimal movement since finger separation is a relatively small angle movement. A 50kΩ potentiometer was selected for this sensor to detect finger separation. The voltage value using the 50kΩ potentiometer was displayed towards the center of the voltage range and, while moving the torque arm to simulate figure separation, the value varied about 0.25 volts. which was visualized on the chart. The torque arm was attached to the potentiometer using epoxy and wire to connect the sensor to the PCB were soldering onto 2 of the potentiometer pins.

Figure 4.38 Potentiometer-based sensor with torque arm (blue piece) and stabilizing piece (white piece) attached to potentiometer [45]

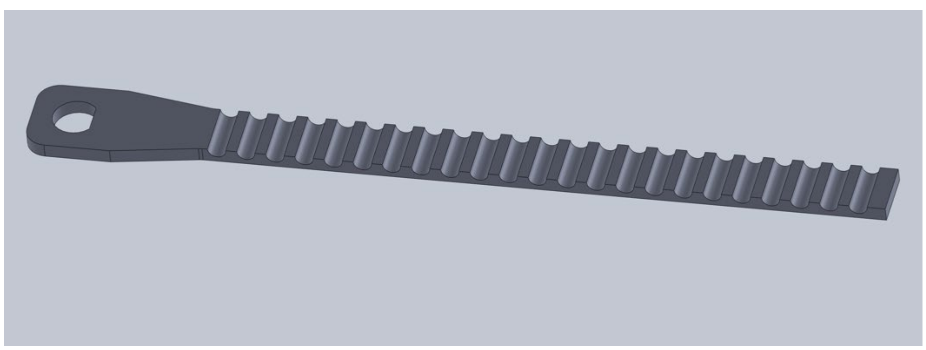


Figure 4.39 3D drawing of torque arm attached to top of potentiometer [45]

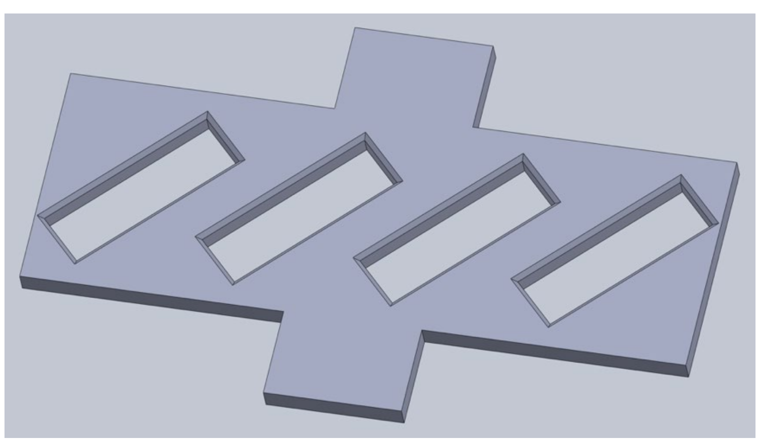


Figure 4.40 3D drawing of stabilizing piece for holding the potentiometers [45]

The potentiometer-based sensors were tested using the Arduino Micro connected to MATLAB. Various torque arms with different lengths, widths, and fit to the potentiometer were 3D printed for the sensor out of flexible filament. The best torque arm length was 118mm since that was the length that allowed the sensor to reach furthest down the finger, providing the most stability on the glove, while also allowing it to be easily rotated when attached to a finger. The potentiometer-based sensor was connected to the Arduino Micro and then the custom PCB, and the torque arm was moved back and forth, and that movement was plotted in LabVIEW. To test the actual finger movement with the sensor, the glove was worn with the sensor taped to the glove and the attached finger was moved side to side while watching the value of the plot in LabVIEW change simultaneously. The LabVIEW plot of this finger movement can be seen in Figure 4.41, where the plot goes up and down based on finger separation. This sensor seemed to have consistent movement and measurement each time it was moved.

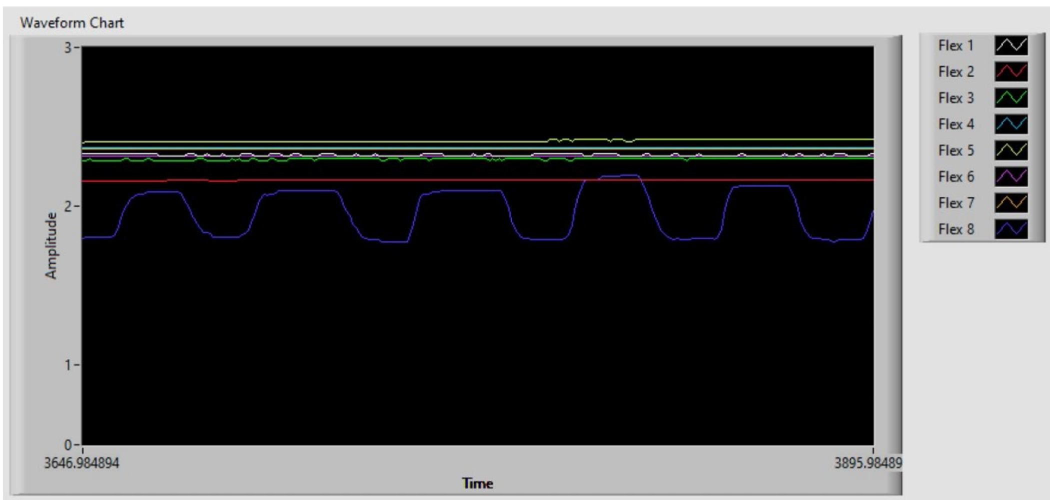


Figure 4.41 50kΩ potentiometer-based sensor tested for measuring finger separation

The potentiometer-based sensor was chosen out of the flex sensors and the time-of-flight sensors based on the criteria in the AHP Table B.1 in Appendix B. The potentiometer-based sensor compared to the other sensors can be seen in Table 4.15. The potentiometer-based sensor can measure finger separation even when the finger it is attached to is bent. It can also accurately measure figure separation even when the finger is bent without the bending adding additional resistance and impacting the data accuracy which would then show up on the plot. It is important to ensure that the correct resistance is being plotted based only on finger separation and not anything else. The potentiometer on this sensor is very small and very lightweight. The parts attached to the potentiometer are also fairly small and light. The torque arm is very thin to allow the finger which it is attached to bend. The cost of the potentiometers is very low, and they come in a large pack. They also seemed very durable while testing, so they should be less likely to break. If a replacement is needed and because a lot of them come in a package for a low cost, they could be replaced inexpensively. Since these sensors were tested while wired, the amount of power needed for them was not observed. This is not an observation that is currently needed since this sensory glove does not need to be wireless and the other criteria were more important.

Table 4.15 Sensor Comparison for Measuring Finger Separation

	Potentiometer-based sensors	Flex Sensors	Time-of -flight sensors
Functionality	Easy connection, can measure no matter the finger position	Easy connection, can still measure finger separation inaccurately in some positions	Harder to connect, goes out of range and can't measure finger separation in some positions
Size	Small, lightweight, can be placed flat on the glove	Small, lightweight, needs to be balanced	Larger sensor, needs to be balanced
Cost	Potentiometers very cheap and come in large package ($0.08)	A more expensive sensor ($12.98)	Can be less expensive ($5.95)
Power	Not currently applicable	Not currently applicable	Not currently applicable

4.4.4.1 Signal Processing

A VI has been created in LabVIEW containing a Front Panel and Block Diagram, which can be seen in Figure 4.42 and Figure 4.43. This VI, "ArduinoSensorFileWriteTest.vi", can also be found in the GitHub repository labeled Sensor Conditioning [45]. This LabVIEW VI was designed to bring in serial data from the PCB and then display that data on the front panel. This data is plotted in the waveform chart as a voltage level. This VI also writes the data from the sensors into a CSV file to be stored so it can be analyzed at a later time. This CSV file contains a time stamp column in milliseconds, starting when the LabVIEW program begins, along with the data from each channel, one column per channel. After reading in 5 sensor channels for 20 minutes, the CSV file was approximately 1.3 MB. At this rate, the CSV file for 8 sensors after 20 minutes will be approximately 3 MB and after an hour, approximately 6 MB. The CSV file is also simultaneously read by the game while data is being continuously written to the file. This allows the game to compare the incoming data to the benchmark data. The game is also able to be launched from LabVIEW's front panel by clicking the "Launch Python" button which was placed inside a case event function inside LabVIEW. Different components were placed and connected in the Block Diagram which allowed a time stamp to be generated and for data to be read from the Arduino via a serial port. Then LabVIEW waits for a CR/LF as the indicator that one full line of data has been received. Once the data is received, the data is sent to a CSV file write function as well as to a waveform graph. The waveform graph, CSV file write, and serial read functions were placed inside a while loop so that the waveform only updates when all 8 input channels have been read. This LabVIEW VI is designed to read data from 8 channels, which is the minimum amount required for the sensors on the glove, however, more channels can be easily added in the front panel by dragging the channel listing down and adding more channels. While traces will automatically be added to the waveform chart, the user can add channel legend names by dragging the legend list on the front panel and adding channel names.

In order to transfer the data to a secondary device for visualization, game play and recording the microcontroller and the secondary device must use compatible protocols [46]. Both wired and wireless transfer mechanisms as well as protocols such as serial, TCP/IP (Transmission Control Protocol/Internet Protocol), etc. were considered. A serial protocol is being used because it is simple, efficient (low power), widely available on most platforms and can be used over both wired and wireless connections. Serial communication is compatible with MATLAB, LabVIEW, and

Python which could be beneficial for future development of this project. All three software platforms can all utilize serial connections regardless of the transmission mechanism. Furthermore, for the glove's required data sample rate, serial data can easily be sent in real-time and does not require a complicated communication software stack. These design choices were made based on the AHP Table B.3 in Appendix B.1

Sampling rates from 10Hz to 50Hz per channel were tested and it was found that around 30Hz provided smooth data for visualization in real-time without much delay. Processing to convert A-to-D binary values to voltage levels along with a 3-point moving average filter have been implemented directly on the PCB to be used for real-time data transmission.

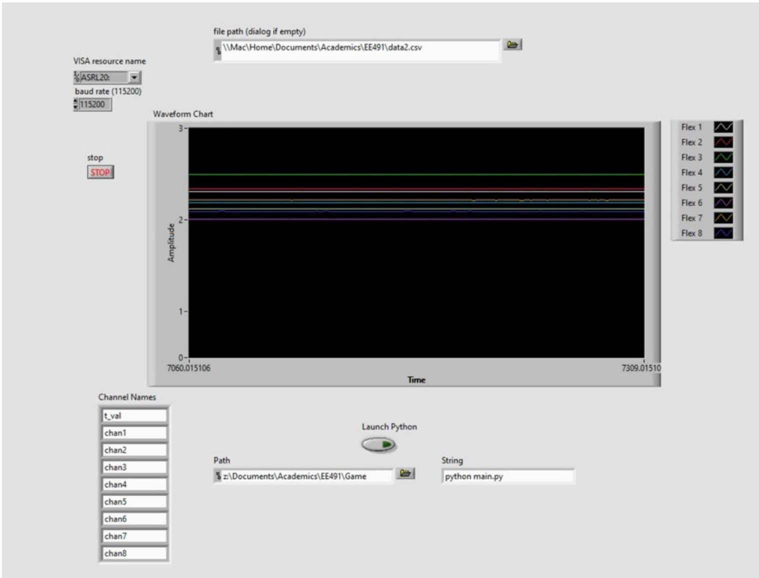


Figure 4.42 Front Panel in LabVIEW to see data plotting, connect to COM port, and change path names

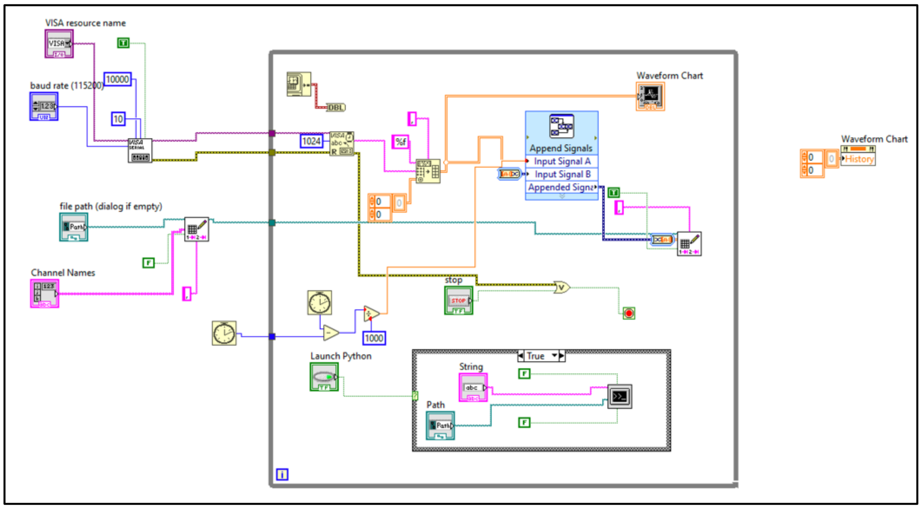


Figure 4.43 Block Diagram in LabVIEW to plot data from the serial port, write to CSV file, and launch game in Python

In order to obtain a proper comparison between different filtering methods, a MATLAB routine was written to store approximately one minute of data for 5 finger sensors (used flex sensors). This MATLAB routine, “arduinoplot2.m”, can be found in the GitHub repository labeled Sensor Conditioning [45]. A 5-point moving average filter (5-Pt Avg), 3-point moving average filter (3-Pt Avg), a low pass filter (lpfx1) and a double low pass filter (lpfx2) were implemented, tested, and observed in MATLAB. These 4 filters were compared using the raw data collected by the sensors as can be observed in Figure 4.44. Since the filters were applied after the data was collected and for the sensory glove they would need to be applied in real time, they were shifted according to how they would appear in real time. From the graph, it can be seen that the low pass filtered lines were shifted further in time and the double low pass filter would take longer than the others to filter. From this testing the 3-point moving average is being used to filter the incoming data from the sensors. This is because the moving average filters have the least amount of delay, while also providing adequate smoothing. The 3-point moving average filter was implemented in the Arduino code and is being used to filter the data before being sent to LabVIEW.

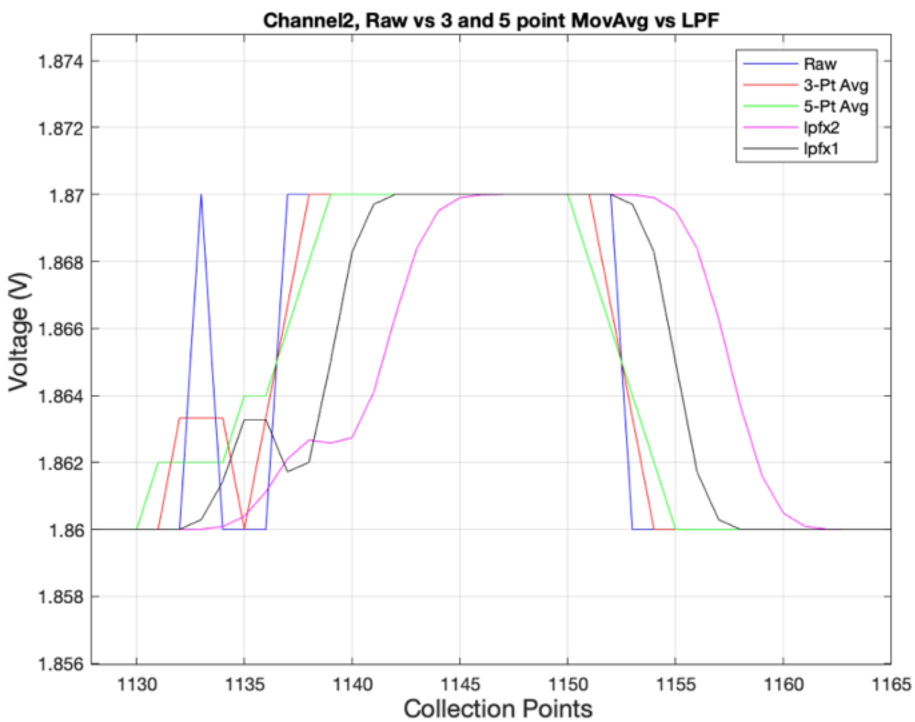


Figure 4.44 Comparison of different filters: 3-point Moving Average, 5-Point Moving Average, Lowpass Filter, and Double Lowpass Filter

The 3-point moving average filter was chosen based on the criteria shown the AHP Table B.2 in Appendix B.1. The accuracy for all of the different low pass filters is about the same. The 3-point moving average filter was the best for real time implementation especially when compared to the longer low pass filter and double low pass filter designed used in MATLAB. The moving average filters had less of a delay when plotting which resulted in a better in real time visualization experience for users.

4.4.5 Game

The game acts as the user interface for the patient to interact with during the testing procedures. It is displayed on the patient's screen in the overall setup, as seen in Figure 4.1. Python was chosen as the language to develop the game in due to the various game development packages available [47]. Several options were considered, including Pygame, Pyglet, and PyOpenGL. After applying the AHP process, PyGame was chosen as the package best fit for our purposes. The AHP process used to make this decision can be seen in Appendix B.3

In order to use the data from the sensors, we needed a method to transfer data from our chosen signal processing software, LabVIEW, to the game's code. Originally, we were planning to use the Python Integration Toolkit for LabVIEW, which is advertised as an easy way to integrate LabVIEW with external Python code [48]. However, support for this toolkit ended over the summer, so we were forced to find an alternative method. After considering our other options, we decided to simply transfer the sensor data by writing to a file from LabVIEW and then reading in from the game's code. This method is simple, and due to the lack of needing an external software or toolkit, is less prone to errors.

Within PyGame, each section of the game, or in other words, the different physical screens that the patient sees on their display, are modeled within the code as screen objects. Therefore, in order to make the flow of the game and code smoother, each section of the game was stubbed out into its own subroutine. The list of subroutines with short descriptions can be seen in Figure 4.46. The overall game is controlled through the main function, which calls different subroutines as needed. The flow follows the overall testing flow seen in Figure 4.6. The game's code can be viewed in our teams GitHub, under the Game repository [17].

The tasks used in the game are read in from external files located under the /Tasks folder. Each file is turned into a different test suite with the tasks in the files being grouped together for that suite. This was intentionally set up this way to allow the doctor to easily be able to edit and add new tasks and test suites, without needing to look through the game's code itself. The game reads in these files in alphabetical order, and the order of test suites is determined the same way. During testing, tasks are randomly selected from the current test suite. Tasks do not repeat within a test suite unless the patient wants to reattempt that suite. Then the test suite is reloaded with all original tasks.

Some tasks also include a corresponding image to help the patient get a clearer understanding of what the task is asking them to do. These are located under the /images folder. Implementing new images does require an addition to a Python dictionary object in the code. Specific instructions on how to do this is included in the README file [17] bundled with the game. The sources for all images currently in the game, including the background images, are included in a file packaged with the game called image_sources.txt.

Benchmarks are used in the game to compare sensor data against during testing. On the title screen of the game, doctors and patients have the option to either set new benchmarks, or use a default file, which should have been preemptively set up. Ideally, new benchmarks only need to be set once, during the patient's first test session. Afterwards, this file can be set as the new default by just changing the name of the file. More specific instructions are included in the README file. When setting new benchmarks, patients are taken one by one through all tasks available in the game, asked to perform the correct hand motion, and then asked to press SPACE to record the sensor data. Each task's benchmark is a list/array of eight values, corresponding to the eight sensors currently on the glove. The game is written to support the removal or addition of sensors without needing to edit anything in the code. During test comparison, data is read in from the data file LabVIEW writes to, and each sensor's current value is compared against the benchmark's value. A constant tolerance percentage value is set within the game, and for a task to be successfully completed, each sensor's value must fall within the tolerance.

Aside from the benchmark setting screens, all relevant screens for testing can be seen in Figure 4.45. The game first starts at the title screen where they have the option to set benchmarks or immediately proceed with testing. If they choose to proceed with testing, they are first taken to two help screens, the first of which shows the labels used for different fingers for tasks. The second help screen displays information about the general procedure of the game and how to earn points. Afterwards, the game moves onto a countdown screen for the duration of the countdown timer. Then, a randomly selected task is displayed on screen for the patient to see. If the patient is able to successfully complete this task in the required time, the game moves onto the task success screen. Either after this, or if the patient could not complete a task in time, the game moves onto the next countdown. We chose to purposefully not implement a task failure screen, as this could induce discouragement in patients. Once all tests in the current suite are finished, the game moves onto the test suite complete screen. The patient then has the option to reattempt that test suite to try to perform better, move onto the next suite if there are any remaining, or quit testing. After testing has finished, the final screen is displayed with the patients point total.

Figure 4.45 Eight different screens displayed during testing. From left to right and top to bottom: title screen, help screen 1, help screen 2, task countdown, task display, suite complete, and final screen

```python
# Description: Main function that controls the overall game flow
def main(): ...


# Given: Game screen
# Returns: Whether to full quit game?, Whether to set benchmarks?
# Description: Title screen of game which user sees first
def title_screen(screen): ...


# Given: Game screen
# Returns: Whether to full quit game?, Whether to continue with game?
# Description: Title screen of game which user sees first
def benchmark_screen(screen): ...


# Given: Game screen
# Returns: Whether to full quit game?
# Description: Shows the user a diagram with the finger label names used in the task descriptions
def finger_name_screen(screen): ...


# Given: None
# Returns: None
# Description: Imports tasks from external files and creates the test suites
def import_tasks(): ...


# Given: Game screen
# Returns: Whether to full quit game?
# Description: Help screen that displays useful instructions about game procedure
def help_screen(screen): ...


# Given: Game screen, tasks for test suite, player's current points
# Returns: Whether to full quit game, player's total points
# Description: Handles the running of a full test suite
def run_test_suite(screen, tests, points): ...


# Given: Game screen, which number current task is, task to be completed
# Returns: Whether to full quit game?
# Description: Displays a countdown before player completes a task
def countdown_screen(screen, task_number, task): ...


# Given: Game screen, which number current task is, task to be completed
# Returns: Whether task was completed successfully, time remaining when task was completed, whether to full quit game?
# Description: Displays task for patient to complete and a countdown to complete in
def task_screen(screen, task_number, task): ...


# Given: None
# Returns: Current hand position data
# Description: Reads in the current hand position data and formats it
def read_data(): ...


# Given: Task to be completed
# Returns: Whether task was completed successfully
# Description: Compares current hand position data to task's benchmark
def benchmark_compare(task): ...


# Given: Game screen, current task number, player's total points, bonus points for completing task
# Returns: Whether to full quit game
# Description: Informs user that task was completed successfully, and how many points were earnt
def task_success(screen, task_num, points, bonus_points): ...


# Given: Game screen, whether there are more suites to complete
# Returns: Whether to move on to next suite, Whether to repeat test suite, Whether to full quit game
# Description: Informs user that suite was completed successfully, and how many points they have
def suite_complete_screen(screen, more_suites): ...


# Given: Game screen, player's points
# Returns: Whether to full quit game?
# Description: Final screen of the game that displays point total for player
def final_screen(screen, points): ...
```

Figure 4.46 Different subroutines within the game's code [17]

5 TESTING AND VALIDATION

For the following subsections, the Engineering Requirements related to each component are provided in Tables 5.1 – 5.5. The Engineering Requirements are highlighted to show the status of each requirement. The rows colored in green indicate the requirement has been tested and verified to work; Yellow rows indicate the requirement as implemented but does not quite meet the specification; Red rows indicate the requirement was not fulfilled. Below each table are descriptions of how each Engineering Requirement was tested and verified.

5.1 Glove Design

The following tests have been performed on the final glove design and accompanying components.

Table 5.1 Glove Design Engineering Requirements

ER	MR	Engineering Requirement	Verification
9	6, 7	Circuitry **shall** be removable	The circuitry will be detached from the glove and then reattached. The glove will then be turned on and tested to see if it is working
15	6, 8	**Shall** be made from material that allows flexibility with different hand sizes	Test flexibility by putting the glove on various hand sizes and flexing the fingers and rotating the wrist
18	6, 8	**Shall** not weigh more than 4oz.	Weigh the glove on a scale with all the components attached
19	6, 7	Circuitry **shall** be encased	Check that all circuitry except for sensors and associated connecting wires are packaged in a case

Below is a list of the engineering requirements corresponding to the glove design with descriptions of how each one has or has not been verified.

Engineering Requirement 9: To satisfy this requirement, straps of Velcro were sewn onto the wrist wrap in the appropriate places to connect the PCB case, the sensors, fingers, and the wires. Each piece was removed from the wrist wrap and then reattached in the correct position. A team member then put the glove on and went through the benchmark test to verify the glove was still working.

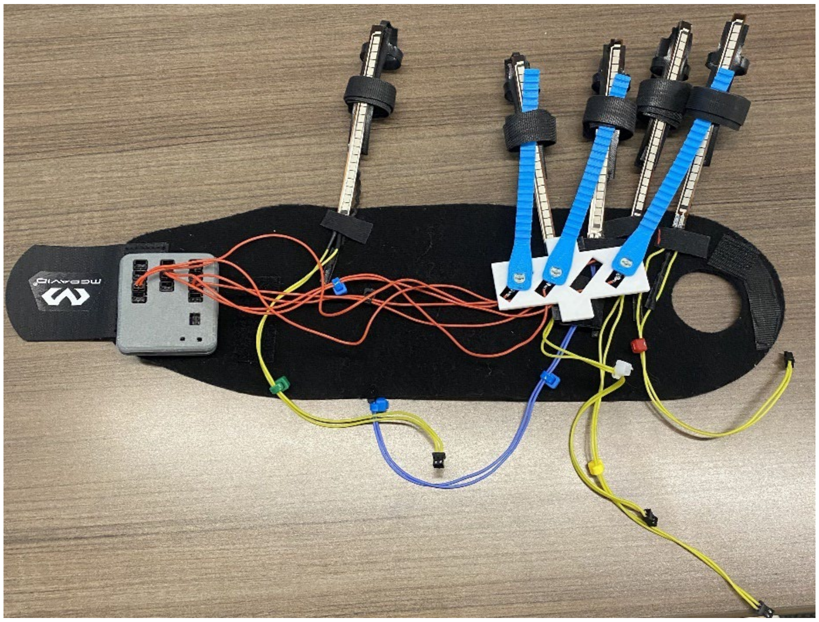


Figure 5.1 Glove laid out flat with all components attached

Engineering Requirement 15: One of the instructors, each team member and a few peers tried on the glove. The glove fit every person that tried it on, however the Velcro and each piece were not always in the correct position depending on a person's hand size.

Engineering Requirement 18: The wrist wrap, PCB, PCB case, USB cord, and sensors were all placed on a kitchen scale as seen in Figure 5.2. The total weight was 3.5 oz.

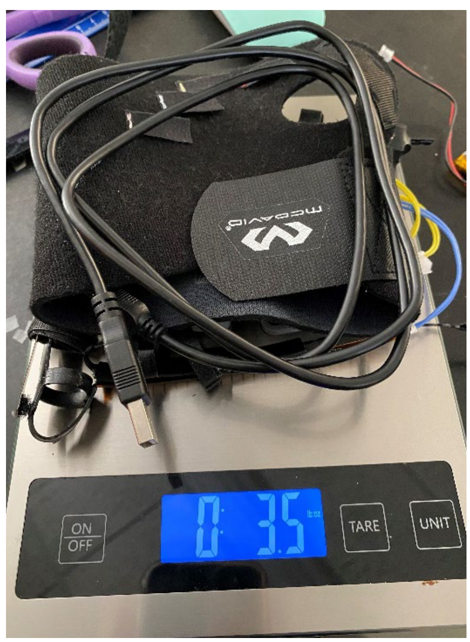


Figure 5.2 Weighing glove and accompanying parts on kitchen scale

Engineering Requirement 19: The PCB case was custom designed around the PCB in Fusion 360. The PCB is able to be placed in the case with the sensors, USB, and associated wires attached and then taken out of the case.

Figure 5.3 PCB outside of case

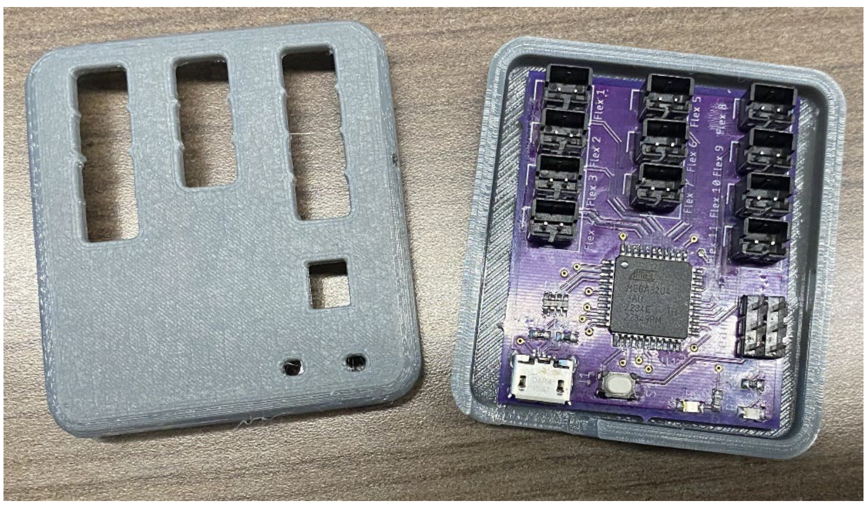


Figure 5.4 PCB inside of case with top off

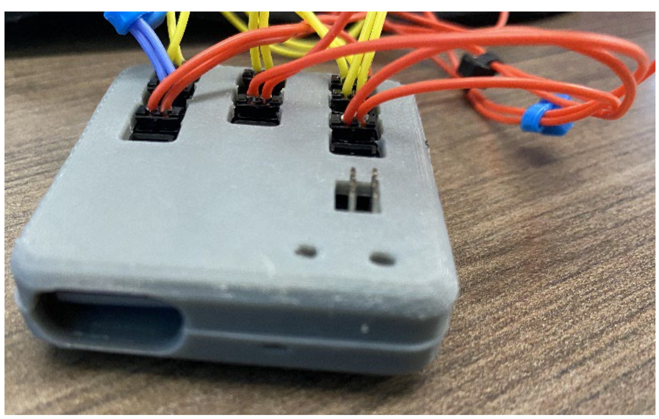


Figure 5.5 PCB inside case with sensor wires connected

5.2 PCB

The following tests have been performed on the first prototype of the PCB. These tests satisfy the engineering requirements highlighted in green.

Table 5.2 PCB Engineering Requirements

ER	MR	Engineering Requirement	Verification
4	1, 5	Sensors **shall** be connected to the ADC pins	Wire the sensors to the ADC pins and then read in changing voltage with the software using ADC
5	8	PCB **shall** be smaller than the previous team's PCB design	Measure both PCB designs and compare both designs to see if the new PCB is smaller than 1.95 x 2"
6	8	PCB **shall** contain the control system and connections to all sensors on a single PCB	Observing that all connections lead back to one PCB with no additional small PCB breakout boards for sensors
10	1	Glove **shall** contain a minimum of 8 sensors	Test each sensor and be able to read the data (ADC measurement)
12	2	Glove **should** use Bluetooth or Wi-Fi	Check that the glove has the capabilities to transmit data over either technology

Engineering Requirement 4: This requirement requires the sensors to be connected to the ADC pins on the ATMega32U4. This allows the user to read in ADC values that can be converted to voltage and then to resistance. This requirement was verified by connecting the sensors to the connector pins that was routed to the ADC pins on the PCB as seen in Figure 5.6. Then, an Arduino program called One_Sense.ino [49] was flashed onto the PCB to read in the ADC value of each flex sensor which was then converted into resistance. In Figure 5.7 the serial monitor displays the resistance values of flex sensors 1 through 5. The remaining flex sensors are disconnected hence the serial monitor outputs "inf". In Figure 5.8, the serial monitor displays the resistance values of flex sensors 5 through 11. Therefore, this requirement was fulfilled.

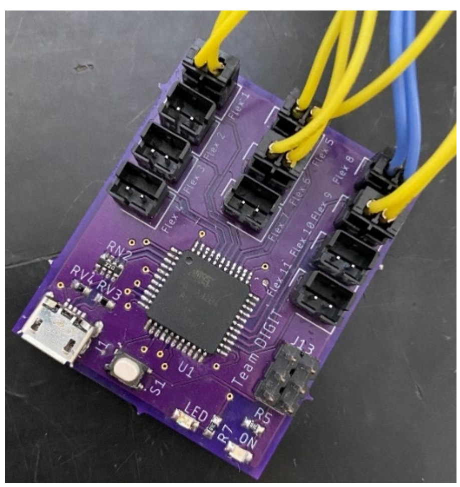


Figure 5.6 Sensor connector pins inserted into ADC pins on PCB

COM13

```
Flex Sensor 6:  inf
Flex Sensor 7:  inf
Flex Sensor 8:  inf
Flex Sensor 9:  inf
Flex Sensor 10: inf
Flex Sensor 11: inf
Hit any key to receive a new reading

Reading all flex sensors...

Flex Sensor 1:  11401.67
Flex Sensor 2:  13571.43
Flex Sensor 3:  12937.22
Flex Sensor 4:  11765.96
Flex Sensor 5:  19738.37
Flex Sensor 6:  inf
Flex Sensor 7:  inf
Flex Sensor 8:  inf
Flex Sensor 9:  inf
Flex Sensor 10: inf
Flex Sensor 11: inf
Hit any key to receive a new reading

Reading all flex sensors...

Flex Sensor 1:  11401.67
Flex Sensor 2:  13517.24
Flex Sensor 3:  12834.82
Flex Sensor 4:  11719.75
Flex Sensor 5:  10297.62
Flex Sensor 6:  inf
Flex Sensor 7:  inf
Flex Sensor 8:  inf
Flex Sensor 9:  inf
Flex Sensor 10: inf
Flex Sensor 11: inf
Hit any key to receive a new reading
```

Figure 5.7 Flex Sensor Testing 1

```
COM13

Flex Sensor 4:  inf
Flex Sensor 5:  inf
Flex Sensor 6:  10257.43
Flex Sensor 7:  inf
Flex Sensor 8:  11401.67
Flex Sensor 9:  13625.87
Flex Sensor 10: 12834.82
Flex Sensor 11: 11765.96
Hit any key to receive a new reading

Reading all flex sensors...

Flex Sensor 1:  inf
Flex Sensor 2:  inf
Flex Sensor 3:  inf
Flex Sensor 4:  inf
Flex Sensor 5:  inf
Flex Sensor 6:  10257.43
Flex Sensor 7:  inf
Flex Sensor 8:  11401.67
Flex Sensor 9:  13625.87
Flex Sensor 10: 12834.82
Flex Sensor 11: 11765.96
Hit any key to receive a new reading

Reading all flex sensors...

Flex Sensor 1:  inf
Flex Sensor 2:  inf
Flex Sensor 3:  inf
Flex Sensor 4:  inf
Flex Sensor 5:  inf
Flex Sensor 6:  inf
Flex Sensor 7:  10257.43
Flex Sensor 8:  11401.67
Flex Sensor 9:  13571.43
Flex Sensor 10: 12834.82
Flex Sensor 11: 11765.96
Hit any key to receive a new reading
```

Figure 5.8 Flex Sensor Testing 2

Engineering Requirement 5: For this requirement, the PCB must be smaller than 1.95” x 2” which was the dimensions of the previous team’s PCB. To verify that team DIGIT’s PCB was smaller, the dimension tool from Autodesk Eagle was used measure the dimensions for both PCBs. Unfortunately, the PCB from the previous team is in the possession of the sponsor for that team, so there was no other way of measuring the PCB other than using the ECAD files. In Figure 5.9, the PCB from the previous team [50] measured to be 1.95” x 2”. In Figure 5.10, team DIGIT’s PCB [41] measured to be 1.4” x 1.7”. Therefore, this requirement was fulfilled.

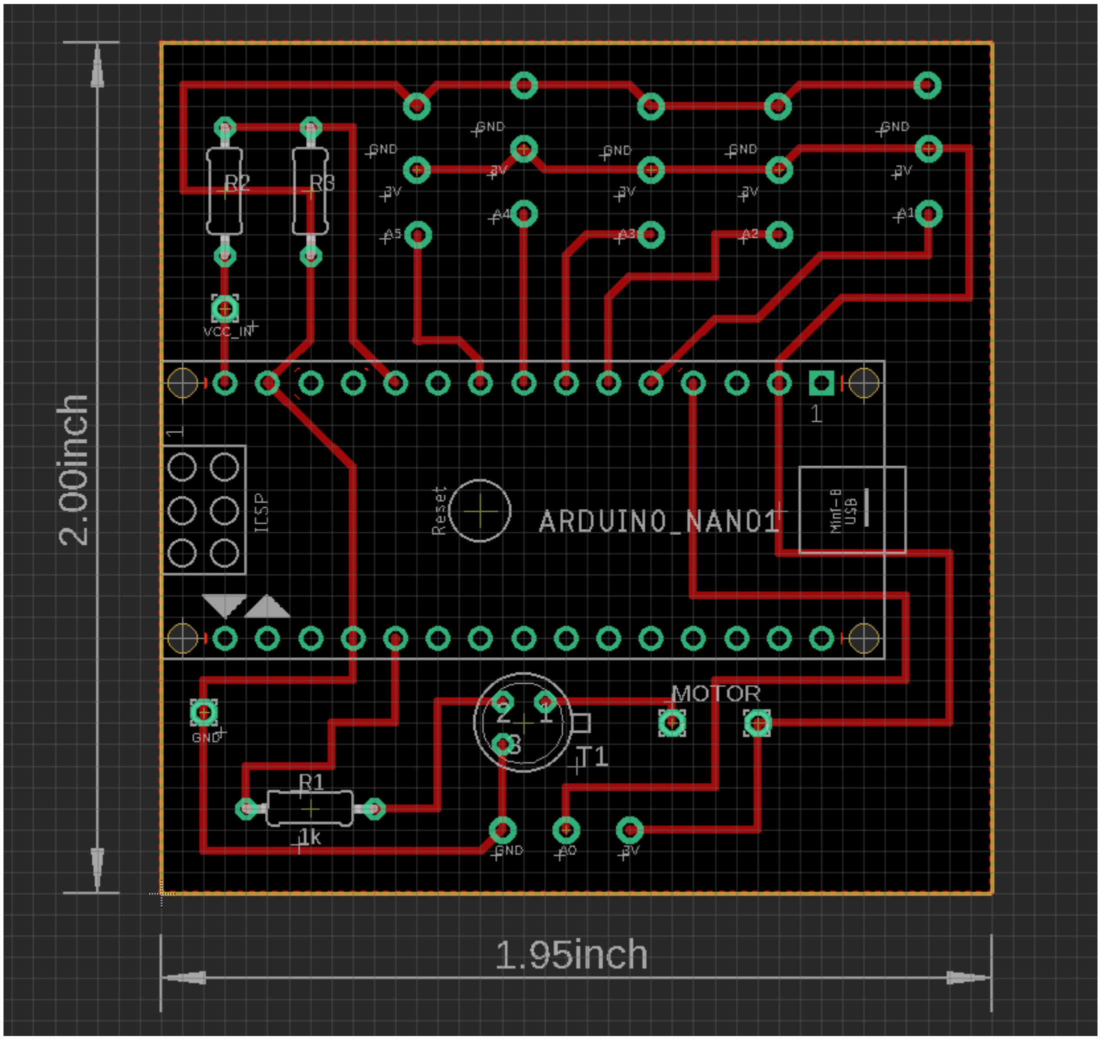


Figure 5.9 Previous Team's PCB [50]

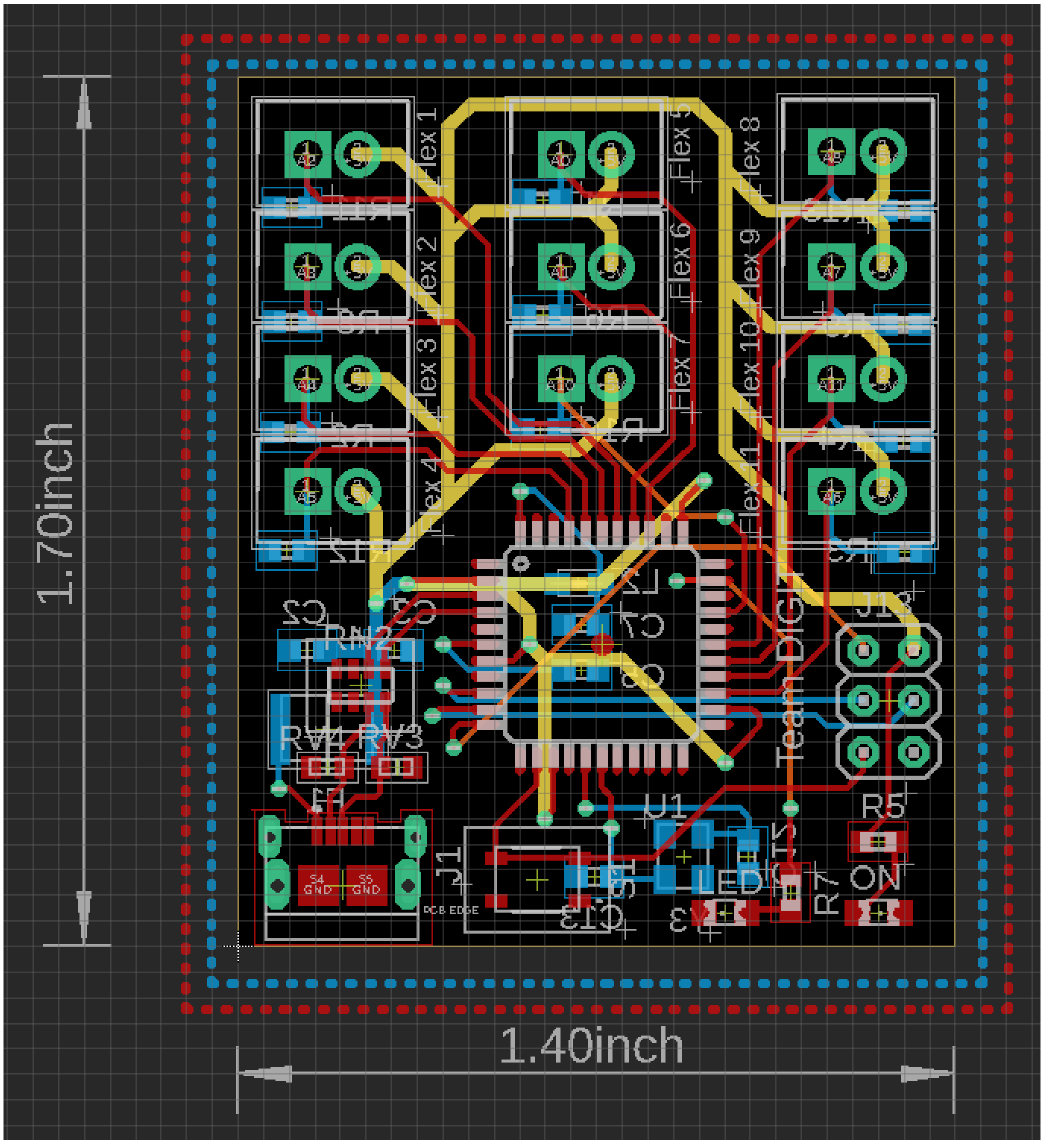


Figure 5.10 Team DIGIT's PCB **[41]**

Engineering Requirement 6: This requirement is fulfilled through observation of the entire glove. All the controls and sensor connectors are located on the PCB as seen in Figure 5.11.

Figure 5.11 Entire glove on a team member’s hand

Engineering Requirement 10: This requirement is fulfilled. Engineering Requirement 10 requires at least 8 sensors for the glove. Team DIGIT’s PCB was designed to support 11 sensors. This was verified through observation of the PCB in Figure 5.12 and Figure 4.29 in section 4.4.2.1 Microcontroller.

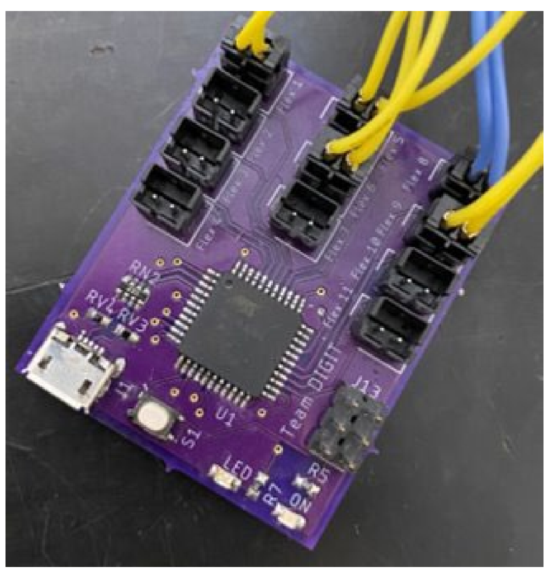


Figure 5.12 PCB with 11 sensors

Engineering Requirement 12: This requirement was not fulfilled for concerns regarding the design being wireless as explained earlier in section 4.4.2 PCB. To summarize, the issue with a wireless design was the increase in cost due to the need for a lower maximum dropout voltage on a low dropout voltage regulator and a Bluetooth or Wi-Fi module. Additionally, there was no extra time to explore the wireless PCB option. So, the PCB was designed to be a wired connection where the PCB was powered through USB. Therefore, this requirement was not fulfilled.

5.3 Power

As stated in sections 4.4.2 and 4.4.3, Team DIGIT has decided to go with a wired PCB and glove so the power will be supplied by the computer or secondary device. Therefore, no testing has been performed to verify that the power supply meets the Engineering Requirements. If a wireless glove were to be designed, the team would connect the battery to an empty circuit board and use a voltmeter to read the voltage the battery provides. The battery would then be connected to the PCB where each sensor would be tested to see if enough power is being supplied. Finally, the team would run through the entire game to simulate one 60-minute session. After the session is complete, the battery level would be read to determine if the battery would last for another session without needing to be recharged. Once the battery has been depleted, the battery would be plugged into the charge to charge until full while the team times how long it takes.

Table 5.3 Power Engineering Requirements

ER	MR	Engineering Requirement	Verification
1	2, 9	Battery **should** last 5-8 hours	Charge the battery to full and then leave the PCB powered up for 8 hours. Confirm after 5 hours that the PCB is still powered
2	2, 4	**Should** monitor battery level	Confirm that the battery level can be viewed through observation by either software or hardware
13	2	**Should** be powered by a rechargeable battery	Deplete the battery completely, and then charge until full again and check battery level to confirm

Below is a list of the engineering requirements corresponding to the power engineering requirements. As stated previously, this glove and PCB are wired so they are powered through a USB cable that connects to the computer. Due to this none of these requirements were fulfilled but below is the testing and verification that would have taken place had we designed a wireless glove.

Engineering Requirement 1: Due to the fact, that we decided to go with a wired glove, there is no way to say how long the battery would last and no testing was done. Had we gone wireless, testing similar to what is listed in the verification for Table 5.3 row 1 would have been done. Using Equation 1, we are able to calculate the approximate battery life for the 3.7 V 2000mAh and 3.7 V 500mAh batteries using the current load of 216.91 mA mentioned in section 4.4.2 however this does not take into account how long the battery life would be while the glove is being used for continuous sessions.

$$Battery\ Life = \frac{Total\ rated\ mAh}{Current\ Load\ (mA)} \qquad (1)$$

The total battery life of a 500mAh would have a battery life of 2.3 hours while the total battery life of a 2000mAh would have a battery life of 9.22 hours.

Engineering Requirement 2: To fulfill this requirement, code would have had to been written to be loaded onto the PCB so that the PCB would be able to read the battery level and send it to the doctor's computer. Another option would be to program an LED to turn on when the battery reaches a certain power level.

Engineering Requirement 13: All batteries that were researched and bought were rechargeable. The 3.7 V 2000mAh and 3.7 V 500mAh batteries both connected to a small charger that can be plugged into a USB port. The coin cell batteries came with a charging unit where they can be placed in to charge.

5.4 Sensors

Table 5.4lists the tests performed to verify that the sensor design decisions meet the Engineering Requirements.

Table 5.4 Sensor Engineering Requirements

ER	MR	Engineering Requirement	Verification
3	3, 4	**Shall** be MATLAB, LabVIEW, or Python	Confirm that all code written is in either MATLAB, LabVIEW, or Python
7	5	Data **shall** be streamed in real-time	Move sensors on the glove and see if data shows up on the secondary device simultaneously
10	1	Glove **shall** contain a minimum of 8 sensors	Test each sensor and be able to read the data (ADC measurement)
11	4, 5	**Shall** achieve a sampling rate greater than 10Hz	Check how many datapoints the glove can receive in one second of testing
16	1, 4, 5	Data from the glove **shall** be sent to a secondary device	Test the connection from the primary device to the secondary device

Engineering Requirement 3: To fulfill the requirement that MATLAB, LabVIEW, or Python is used. LabVIEW was used to read in voltage data from the PCB and then simultaneously plot that data. LabVIEW also launched the game written in Python.

Engineering Requirement 7: To test whether data was streamed in real time to the secondary device (LabVIEW), the flex sensors and potentiometer-based sensors were moved on the glove while plotting on the graph at the same time.

Engineering Requirement 10: The glove contains a total of 8 sensors, consisting of 5 flex sensors and 3 potentiometer-based sensors. With all 8 sensors, all the data can be read in and plotted simultaneously. All 8 traces from the sensors can be seen in Figure 5.13. Up to 11 sensors can be added by uncommenting the additional lines in the Arduino code and adding channels to the LabVIEW front panel.

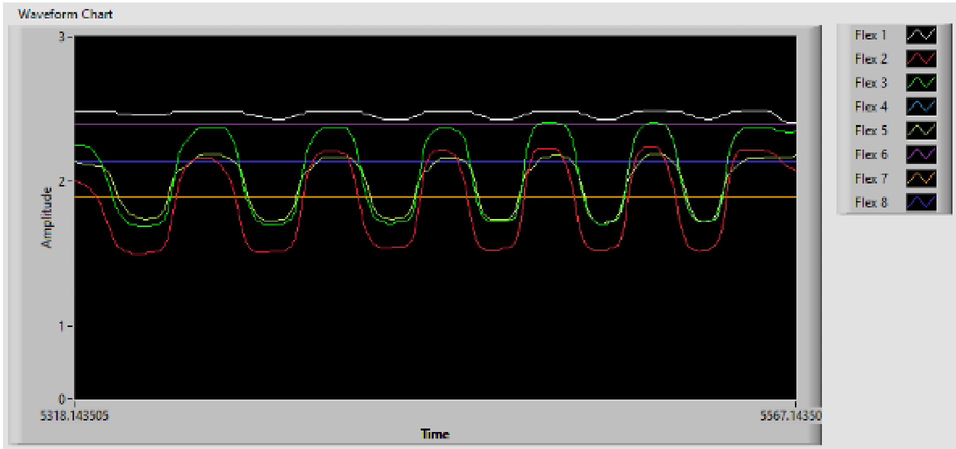
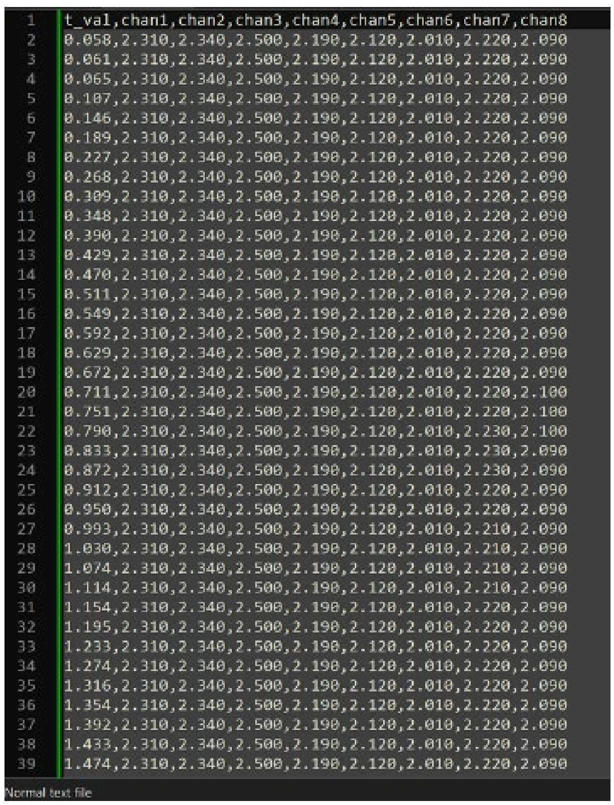


Figure 5.13 LabVIEW waveform chart with 8 traces plotting simultaneously

Engineering Requirement 11: A sampling rate of at least 10Hz was verified by looking in the CSV file which contains a time stamp. Reading the time stamps on the file confirmed that about 26 lines of code were read into the file per second with each line containing one reading per channel. The initial 26 lines can be seen in Figure 5.14. This equates to at least 26 samples per second or 26Hz for each channel which is about 208 total samples per second.



```
t_val,chan1,chan2,chan3,chan4,chan5,chan6,chan7,chan8
0.058,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.061,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.065,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.107,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.146,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.189,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.227,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.268,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.309,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.348,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.390,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.429,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.470,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.511,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.549,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.592,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.629,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.672,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.711,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.100
0.751,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.100
0.790,2.310,2.340,2.500,2.190,2.120,2.010,2.230,2.100
0.833,2.310,2.340,2.500,2.190,2.120,2.010,2.230,2.090
0.872,2.310,2.340,2.500,2.190,2.120,2.010,2.230,2.090
0.912,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.950,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
0.993,2.310,2.340,2.500,2.190,2.120,2.010,2.210,2.090
1.030,2.310,2.340,2.500,2.190,2.120,2.010,2.210,2.090
1.074,2.310,2.340,2.500,2.190,2.120,2.010,2.210,2.090
1.114,2.310,2.340,2.500,2.190,2.120,2.010,2.210,2.090
1.154,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
1.195,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
1.233,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
1.274,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
1.316,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
1.354,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
1.392,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
1.433,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
1.474,2.310,2.340,2.500,2.190,2.120,2.010,2.220,2.090
```
Normal text file

Figure 5.14 CSV file with time stamp and 8 channels of data

Engineering Requirement 16: The data was verified to be sent to the secondary device by plotting the sensor data. The sensors were moved, and the voltage value changed consistently based on this movement. For example, on the plot in LabVIEW, the voltage value was decreased as the bend was increased.

5.5 Game

Table 5.5 lists all the Engineering Requirements related to the game, along with the verification to show it's satisfied. An explanation of how each Engineering Requirement related to the game was or was not satisfied is provided as well.

Table 5.5 Game Engineering Requirements

ER	MR	Engineering Requirement	Verification
3	3, 4	**Shall** be MATLAB, LabVIEW, python	Confirm that all code written is in either MATLAB, LabVIEW, or Python
8	8	**Should** be displayed in 3D motion	Check that hand model matches the current position and orientation of the actual hand being tested.
14	1, 3	**Shall** provide positive feedback when patient correctly follows commands	Run an instance of the game and test that when a command is correctly followed, some form of feedback is issued on screen by the game.
17	1, 3	Secondary device **shall** display commands to instruct patient to perform specific tasks to test hand movement	Run an instance of the game and check to make sure commands are being issued continuously as the patient completes previous tasks.

An explanation of how each Engineering Requirement related to the game was or was not satisfied is provided here:

Engineering Requirement 3: Engineering Requirement 3 says that everything should be programmed in MATLAB, LabVIEW, and Python. The game is entirely coded in Python, thereby satisfying this requirement.

Engineering Requirement 8: Engineering Requirement 8 says that the input sensor data should be displayed in 3D motion. This requirement was not fulfilled by us. After initial research, we could not find a simple, open-source method to displaying the input hand motion data in real-time in a 3D format. Due to the additional time, it would take to research and learn how to develop a model, as well as the other, more essential requirements that needed to be completed, we decided not to focus on this requirement.

Engineering Requirement 14: Engineering Requirement 14 says that the game shall provide positive feedback when patients correctly follow commands. This is intended to help keep the patient positively encouraged and to keep progressing with the tests. This requirement is satisfied by the game and an example can be seen in the task success screen in Figure 5.15, which is displayed only when a patient successfully completes a task. The game informs the patient that they have completed a task successfully and how many points they are awarded.

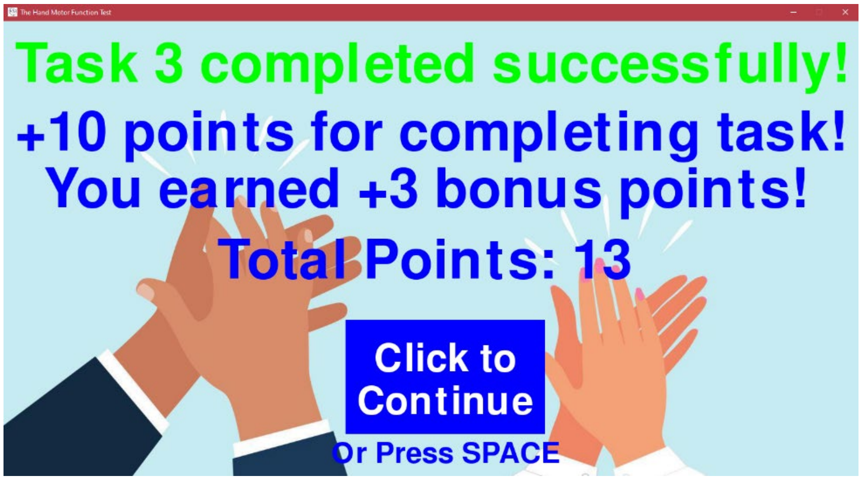


Figure 5.15 Task success screen shown upon successful completion of a task

Engineering Requirement 17: Engineering Requirement 17 says that the secondary device shall display commands to instruct patients to perform specific tasks to test hand movements. The game is currently able to do this, as can be seen in the task display screen in Figure 5.16. In addition, it has been shown that as patients complete tasks, new tasks are able to be displayed continuously for the required duration. The suite complete screen, shown again in Figure 5.17 can only be accessed after displaying the set five tasks per suite.

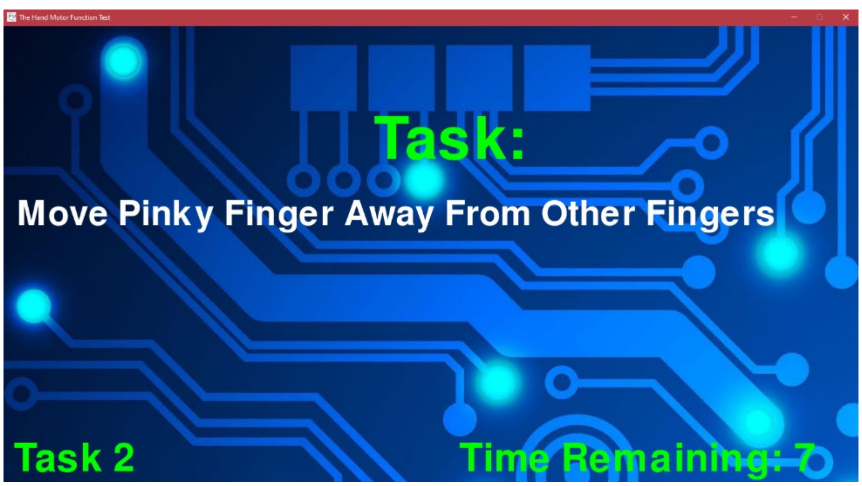


Figure 5.16 Task display screen showing current task to be completed

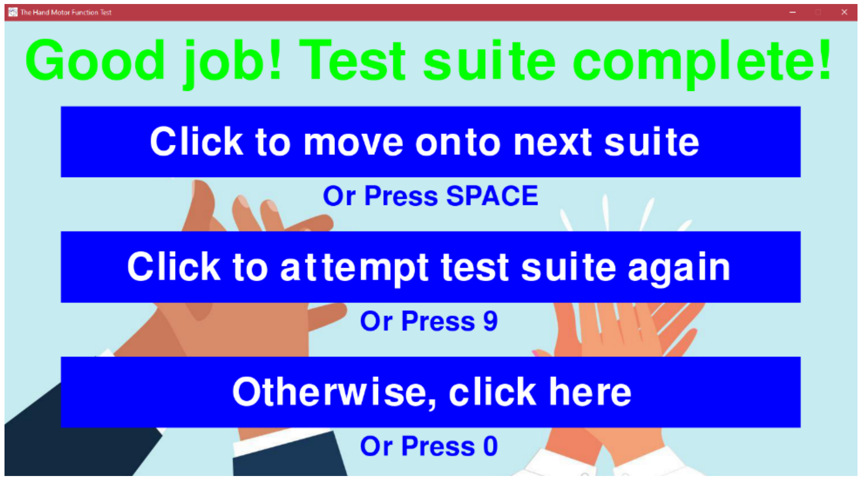


Figure 5.17 Suite complete screen shown upon completion of a test suite

6 FINAL SYSTEM INTEGRATION

Figure 6.1 Glove Integration

The glove was composed of different materials sewn and attached together with 3D printed & Velcro pieces. The flex sensor was attached to the flexible 3D fingers with small Velcro pieces. The 3D fingers were then attached to the finger by clipping the 3D pieces together and sewing the 3D finger to a Velcro wrap that adjusts around the finger. Then, the white 3D piece called the potentiometer stabilizer that holds the potentiometers in place. The stabilizer was attached to the adjustable wristband by sewing the Velcro piece to the wristband and attaching the other Velcro piece to the stabilizer. Then, the blue torque arm was glued to the knob of the potentiometer and attached to the finger using the adjustable Velcro wrap. Finally, the PCB case was attached to the wristband with a bigger Velcro piece.

The sensors are connected to the PCB via wire harnesses. The wires are soldered onto the sensor pins on one end of the wire and the other end of the wire has female sockets crimped onto it so that it was inserted into a 2-pin female connector. Then, that connector was connected to its on-board mating part.

The custom AtMega32U4 PCB has been flashed with an Arduino Leonardo bootloader using an AVR pocket programmer purchased from sparkfun.com. There was a hookup guide made available by sparkfun that has step by step instructions to burning the bootloader onto the microchip.

Once the PCB became programmable, the PCB was placed into a case that sits on the wrist of the glove. Then, the PCB was flashed with an Arduino program to read the sensors data and send that data to the serial port. Then, the LabVIEW program was loaded and the COM port for the PCB was selected to receive the data from the PCB through the serial port.

LabVIEW and the Python code for the game are connected through an external csv datafile. LabVIEW writes the sensor data to the file continuously in real-time. The game checks the latest data, or in other words the last line in the file, whenever it needs input data. This was either during the initial setting of benchmarks, or during actual testing, in which data is read in from the csv around every 10ms.

7 PROJECT PLAN

7.1 Work Breakdown

Table 7.1 is the workflow breakdown for the project. It provides the activities that were completed to be done along with a description and deliverables. Each activity has who was assigned and if there were any activities that were predecessors.

Table 7.1 Workflow Breakdown

ID	Activity	Description	Deliverables	Time	People	Resources	Predecessors
1	Research						
1.1	Design of Glove	Select a material for glove	-Identify types and costs -Choose a material	7	Lili	Internet	N/A
1.2	Power Supply	Select a power supply	-Identify types and costs -Choose a power supply	7	Lili	Internet	N/A
1.3	PCB	Design the PCB	-Identify types and costs -Choose a microcontroller Choose a PCB	7	Jewell	Internet	N/A

ID	Activity	Description	Deliverables	Time	People	Resources	Predecessors
1.4	Sensors	Select sensors	-Identify types and costs -Choose two types of sensors	7	Gwynne	Internet	N/A
1.5	User Interface	Select a software package	-Identify different packages and benefits of each -Choose a software package	7	Bobby	Internet	N/A
2	Interface Circuity						
2.1	Purchase parts	Order and receive parts	-Place order for components -Receive parts	21	All	Internet	1.1-1.4
2.2	Design a PCB	Use a PCB design software to create a PCB that incorporates all components for this project	-Create schematic -Create board layout	60	Jewell	Autodesk Eagle	1.3
2.3	Purchase and integrate PCB	Purchase the PCB and integrate it to the glove	-Order and receive PCB -Assemble PCB	90	Jewell	Oshpark	1.3
2.4	Choose a processor chip	Select a processor chip that will satisfy the engineering requirements	-Select a processor chip	14	Jewell	Internet	1.3
3	Secondary Device						
3.1	Create an application	Design and code an onscreen game for the patient to interact with during testing	-Design overall flow of the game -Code overall flow -Implement each individual screen within the game	90	Bobby	Pygame	N/A

ID	Activity	Description	Deliverables	Time	People	Resources	Predecessors
3.2	Integrate game application with data processing application	Create a connection between the game's code and the data processing application to transfer processed data	-Establish connection between game code and data processing software -Implement the transfer of data between applications	7	Bobby Gwynne	Python Integration Toolkit [48]	N/A

7.2 Team Member Contributions

Table 7.2 are the team member contributions. It provides the primary and secondary contributions of each team member along with an explanation of their tasks.

Table 7.2 Team Member contributions

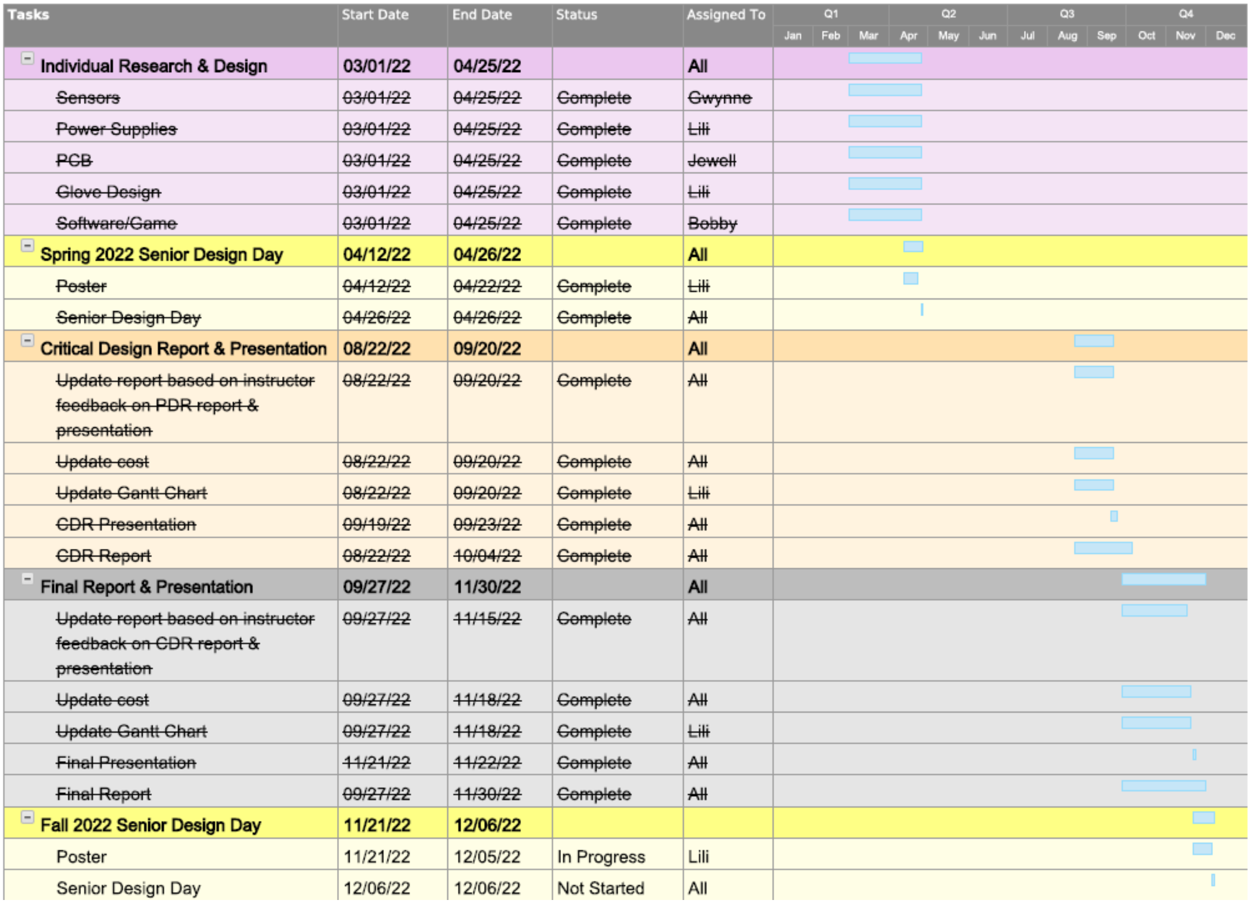
Team Member	Primary Responsibilities	Secondary Responsibilities	Explanation of Task
Bobby Bose	User Interface	Data Processing	Designing and coding the user interface to be used with patient tests.
Jewell Catlett	PCB Design	Wireless Connection and Power	Designing a custom PCB to read sensor data and transfer sensor data via Bluetooth to secondary device.
Lili Petrowsky	Glove Design	Wireless Connection and Power	Design and 3D print an exoskeleton glove
Gwynne Symons Buxton	Sensors	Data Processing	Identification of sensors and appropriate data rates and signal conditioning for real-time analysis.

7.3 Gantt Chart

Figure 7.1 and Figure 7.2 are the Gantt chart for the entire project and were created using Smartsheet [51]. Tasks that are crossed out, have been completed. The Gantt chart starts out with the course deliverables/the main assignments for each semester. From there, the assignments have been broken down into their respective parts as in Figure 7.3.

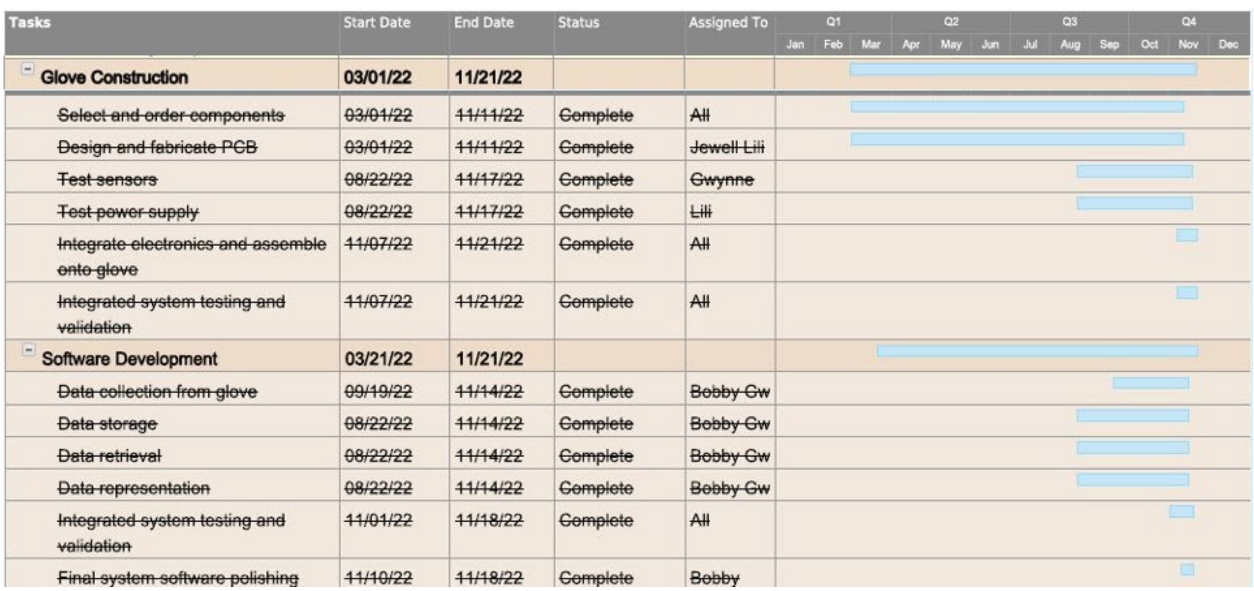
Tasks	Start Date	End Date	Status	Assigned To	Q1			Q2			Q3			Q4		
					Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Course Deliverables	**01/18/22**	**12/07/22**														
Requirements Report & Presentation	**01/18/22**	**02/22/22**		**All**												
~~Marketing Requirements~~	~~01/18/22~~	~~02/15/22~~	~~Complete~~	~~All~~												
~~Objective Tree~~	~~01/18/22~~	~~02/15/22~~	~~Complete~~	~~All~~												
~~Engineering Requirements~~	~~01/18/22~~	~~02/15/22~~	~~Complete~~	~~All~~												
~~Rationale~~	~~01/18/22~~	~~02/15/22~~	~~Complete~~	~~All~~												
~~Justification~~	~~01/18/22~~	~~02/15/22~~	~~Complete~~	~~All~~												
~~Testability of Engineering Requirements~~	~~01/18/22~~	~~02/15/22~~	~~Complete~~	~~All~~												
~~Verification of Engineering Requirements~~	~~01/18/22~~	~~02/15/22~~	~~Complete~~	~~All~~												
~~Impact Statements~~	~~01/18/22~~	~~02/15/22~~	~~Complete~~	~~All~~												
~~References~~	~~01/18/22~~	~~02/15/22~~	~~Complete~~	~~All~~												
~~Requirements Presentation~~	~~01/18/22~~	~~02/22/22~~	~~Complete~~	~~All~~												
~~Requirements Report~~	~~01/18/22~~	~~02/22/22~~	~~Complete~~	~~All~~												
Preliminary Design and Research Report & Presentation	**03/03/22**	**04/14/22**		**All**												
~~Update requirements based on instructor feedback on requirements report~~	~~02/28/22~~	~~04/04/22~~	~~Complete~~	~~All~~												
~~Work Breakdown~~	~~01/18/22~~	~~04/28/22~~	~~Complete~~	~~All~~												
~~Gantt Chart~~	~~01/18/22~~	~~04/28/22~~	~~Complete~~	~~Lili~~												
~~Functional Decomposition~~	~~03/01/22~~	~~03/10/22~~	~~Complete~~	~~All~~												
~~Cost~~	~~03/28/22~~	~~04/25/22~~	~~Complete~~	~~All~~												
~~Design Alternative~~	~~03/28/22~~	~~04/25/22~~	~~Complete~~	~~All~~												
~~PDR Presentation~~	~~03/04/22~~	~~04/07/22~~	~~Complete~~	~~All~~												
~~PDR Report~~	~~03/04/22~~	~~04/14/22~~	~~Complete~~	~~All~~												

Figure 7.1 Gantt Chart Part 1



Tasks	Start Date	End Date	Status	Assigned To
Individual Research & Design	03/01/22	04/25/22		All
~~Sensors~~	~~03/01/22~~	~~04/25/22~~	~~Complete~~	~~Gwynne~~
~~Power Supplies~~	~~03/01/22~~	~~04/25/22~~	~~Complete~~	~~Lili~~
~~PCB~~	~~03/01/22~~	~~04/25/22~~	~~Complete~~	~~Jewell~~
~~Glove Design~~	~~03/01/22~~	~~04/25/22~~	~~Complete~~	~~Lili~~
~~Software/Game~~	~~03/01/22~~	~~04/25/22~~	~~Complete~~	~~Bobby~~
Spring 2022 Senior Design Day	04/12/22	04/26/22		All
~~Poster~~	~~04/12/22~~	~~04/22/22~~	~~Complete~~	~~Lili~~
~~Senior Design Day~~	~~04/26/22~~	~~04/26/22~~	~~Complete~~	~~All~~
Critical Design Report & Presentation	08/22/22	09/20/22		All
~~Update report based on instructor feedback on PDR report & presentation~~	~~08/22/22~~	~~09/20/22~~	~~Complete~~	~~All~~
~~Update cost~~	~~08/22/22~~	~~09/20/22~~	~~Complete~~	~~All~~
~~Update Gantt Chart~~	~~08/22/22~~	~~09/20/22~~	~~Complete~~	~~Lili~~
~~CDR Presentation~~	~~09/19/22~~	~~09/23/22~~	~~Complete~~	~~All~~
~~CDR Report~~	~~08/22/22~~	~~10/04/22~~	~~Complete~~	~~All~~
Final Report & Presentation	09/27/22	11/30/22		All
~~Update report based on instructor feedback on CDR report & presentation~~	~~09/27/22~~	~~11/15/22~~	~~Complete~~	~~All~~
~~Update cost~~	~~09/27/22~~	~~11/18/22~~	~~Complete~~	~~All~~
~~Update Gantt Chart~~	~~09/27/22~~	~~11/18/22~~	~~Complete~~	~~Lili~~
~~Final Presentation~~	~~11/21/22~~	~~11/22/22~~	~~Complete~~	~~All~~
~~Final Report~~	~~09/27/22~~	~~11/30/22~~	~~Complete~~	~~All~~
Fall 2022 Senior Design Day	11/21/22	12/06/22		
Poster	11/21/22	12/05/22	In Progress	Lili
Senior Design Day	12/06/22	12/06/22	Not Started	All

Figure 7.2 Gantt Chart Part 2



Tasks	Start Date	End Date	Status	Assigned To
Glove Construction	03/01/22	11/21/22		
~~Select and order components~~	~~03/01/22~~	~~11/11/22~~	~~Complete~~	~~All~~
~~Design and fabricate PCB~~	~~03/01/22~~	~~11/11/22~~	~~Complete~~	~~Jewell Lili~~
~~Test sensors~~	~~08/22/22~~	~~11/17/22~~	~~Complete~~	~~Gwynne~~
~~Test power supply~~	~~08/22/22~~	~~11/17/22~~	~~Complete~~	~~Lili~~
~~Integrate electronics and assemble onto glove~~	~~11/07/22~~	~~11/21/22~~	~~Complete~~	~~All~~
~~Integrated system testing and validation~~	~~11/07/22~~	~~11/21/22~~	~~Complete~~	~~All~~
Software Development	03/21/22	11/21/22		
~~Data collection from glove~~	~~09/19/22~~	~~11/14/22~~	~~Complete~~	~~Bobby Gw~~
~~Data storage~~	~~08/22/22~~	~~11/14/22~~	~~Complete~~	~~Bobby Gw~~
~~Data retrieval~~	~~08/22/22~~	~~11/14/22~~	~~Complete~~	~~Bobby Gw~~
~~Data representation~~	~~08/22/22~~	~~11/14/22~~	~~Complete~~	~~Bobby Gw~~
~~Integrated system testing and validation~~	~~11/01/22~~	~~11/18/22~~	~~Complete~~	~~All~~
~~Final system software polishing~~	~~11/10/22~~	~~11/18/22~~	~~Complete~~	~~Bobby~~

Figure 7.3 Gantt Chart Part 3

7.4 Bill of Materials

Table 7.3 is a list of all materials that were bought by Team DIGIT along with the materials Team DIGIT inherited from Team SCREEN. The Excel Spreadsheet in the GitHub Repository Bill-of-Materials [52] named Total Expenditures.xlsx includes the link to where each item was bought and the Previous Team Items.xlsx provides an inventory of all materials Team DIGIT inherited from Team SCREEN.

Table 7.3 Bill of Materials

Item	Qty	Vendor
SainSmart TPU	1	Amazon
ATARAXIA ART PLA+	1	Amazon
Lithium Ion Battery 3.7V 2000mAh	5	Adafruit
Lithium Ion Battery 3.7V 500mAh	5	Adafruit
Adafruit micro Lip – USB LiIon/LiPoly Charger	2	Adafruit
Adafruit VCLN4040 F	2	Adafruit
Adafruit VL53L4CD Time of Flight Distance Sensor	2	Adafruit
Conductive Rubber Cord Stretch Sensor	1	Adafruit
Stainless Medium Conductive Thread	1	Adafruit
Watch strap connector (42mm/44mm/45mm)	2	Amazon
Watch strap connector (38mm/40mm/41mm)	2	Amazon
100nF capacitor	20	Digikey
10k resistor	30	Mouser
10 µF capacitor	10	Mouser
16MHz crystal	10	Mouser
Diode	10	Mouser
1k resistor	20	Mouser
1 µF capacitor	20	Mouser
22pF capacitor	20	Digikey
3x2 header	6	Mouser
2-pin male connectors	30	Mouser
ATMega32U4 TQFP44	3	Mouser
ATMega 32U4 QFN44	3	Mouser
22 Resistor Array	10	Mouser
CG0603MLC varistor	10	Mouser
Low power op amp	20	Mouser
500mA fuse	10	Mouser
EMI suppression beads	10	Digikey
Push button	10	Mouser
USB micro	6	Mouser
Green LED	20	Mouser
CT Energy Lithium Ion 2032 Battery Charger	1	Amazon
LIR 2032 Lithium Button Battery Charger	1	Amazon
AmazonBasics TPU	1	Amazon
SpiderMaker SpiderFlex	1	Amazon
PolyFlex TPU	1	Amazon
Ninjaflex TPE	1	Amazon
Long Flex Sensor	20	Adafruit

Item	Qty	Vendor
Short Flex Sensor	10	Adafruit
Pocket AVR Programmer	1	Sparkfun
AVR Programming Cable	1	Sparkfun
A000093	2	Digikey
A000005	2	Digikey
MCP6004-I/P	2	Digikey
MCP004-I/SL	6	Digikey
Fermerry 28AWG Silicone Electric Wire	1	Amazon
Tactile Switches	5	Mouser
USB micro B	5	Digikey
SanDisk 32GB Ultra Memory Card	1	Amazon
Female Pin	100	Mouser
Female Connector	30	Mouser
Mini micro Open Barrel Crimping Tools	1	Amazon
Switch	5	Mouser
Neo G Active Wrist Support	2	CVS Pharmacy
Futuro Wrist Performance Comfort Support	1	CVS Pharmacy
McDavid Sport Wrist Wrap	1	Walmart
Velcro One-Wrap Thin Roll	1	Walmart
Velcro Sticky Back	1	Walmart
Singer Hand Sewing Thread Spools Kit	1	Walmart
TeamDIGIT_Wired_PCB_TQFP_V3	3	OshPark
TeamDIGIT_Wired_PCB_QFN_V4	3	OshPark
ITSYBITSY_SENSORY_V2	3	OshPark
NANO_SENSORY_2022_09_06	3	OshPark
TeamDIGIT_WIRED_PCB_QFN_V2	3	OshPark
TeamDIGIT_Wired_PCB_TQFP_2022_09__06	3	OshPark
ATMEGA32U4	3	DigiKey
10k Resistor	30	Mouser
16MHz Crystal	10	Mouser
0603 10 µF Capacitor	10	Mouser
2-pin male connector	30	Mouser
Items from previous Teams		
Fingerless Glove	6	Team SCREEN
DC-DC Step-up	3	Team SCREEN
Coin Cell Holder	2	Team SCREEN
Breadboard wires		Team SCREEN
RGB 5m LED	100	Team SCREEN
Ion Motor	6	Team SCREEN
Breakaway Connector for Arduino Shield	10	Team SCREEN
LEDS	12	Team SCREEN
MB12A05	2	Team SCREEN
Vibration Sensor	1	Team SCREEN
10 µF 50V	2	Team SCREEN
100 µF 50V	2	Team SCREEN
Button buzzer	2	Team SCREEN
22 ceramic disc capacitor	5	Team SCREEN
104 ceramic disc capacitor	5	Team SCREEN

Item	Qty	Vendor
IN4007	5	Team SCREEN
SN74HC595N	1	Team SCREEN
SH538	1	Team SCREEN
PN2222	5	Team SCREEN
A92B331	3	Team SCREEN
S8050D331	5	Team SCREEN
Digital Battery Charger	1	Team SCREEN
Adafruit Small Enclosed Piezo w/ wires	2	Team SCREEN
Development kit	3	Team SCREEN
Arduino Nano 33 BLE Board	2	Team SCREEN
Flex Sensor w/ blue	30	Team SCREEN
Spectrasymbol	2	Team SCREEN
Spectrasymbol	1	Team SCREEN
Reachargeable 9V battery	1	Team SCREEN
9V battery	2	Team SCREEN
1 in flexpoint sensor system	28	Team SCREEN
3D black slits to hold flex sensors	17	Team SCREEN
Mini Digital Video Battery	2	Team SCREEN

7.5 Cost Analysis

Table 7.4 lists all the materials bought using the Reese Terry Funds along with the individual cost, quantity bought, where materials were from, and total cost spent. The Excel Spreadsheet in the GitHub Repository Bill-of-Materials [52] named Total Expenditures.xlsx includes part/manufacturer numbers along with the link to where each item was bought.

Table 7.4 Total Expenditures

Item	Individual Cost	Qty	Vendor	Cost
SainSmart TPU	$ 15.99	1	Amazon	$15.99
ATARAXIA ART PLA+	$ 39.99	1	Amazon	$39.99
Lithium Ion Battery 3.7V 2000mAh	$ 12.50	5	Adafruit	$62.50
Lithium Ion Battery 3.7V 500mAh	$ 7.95	5	Adafruit	$39.75
Adafruit micro Lip – USB LiIon/LiPoly Charger	$ 5.95	2	Adafruit	$11.90
Adafruit VCLN4040 F	$ 5.95	2	Adafruit	$11.90
Adafruit VL53L4CD Time of Flight Distance Sensor	$ 14.95	2	Adafruit	$29.90
Conductive Rubber Cord Stretch Sensor	$ 9.95	1	Adafruit	$9.95
Stainless Medium Conductive Thread	$ 9.95	1	Adafruit	$9.95
Watch strap connector (42mm/44mm/45mm)	$ 14.95	2	Amazon	$29.90
Watch strap connector (38mm/40mm/41mm)	$ 14.95	2	Amazon	$29.90
100nF capacitor	$ 0.10	20	Digikey	$2.00
10k resistor	$ 0.09	30	Mouser	$2.70
10 µF capacitor	$ 0.78	10	Mouser	$7.80
16MHz crystal	$ 0.43	10	Mouser	$4.30
Diode	$ 0.13	10	Mouser	$1.30

Item	Individual Cost	Qty	Vendor	Cost
1k resistor	$ 0.10	20	Mouser	$2.00
1 µF capacitor	$ 0.12	20	Mouser	$2.40
22pF capacitor	$ 0.18	20	Digikey	$3.60
3x2 header	$ 0.46	6	Mouser	$2.76
2-pin male connectors	$ 0.04	30	Mouser	$1.20
22 Resistor Array	$ 0.07	10	Mouser	$0.70
CG0603MLC varistor	$ 0.37	10	Mouser	$3.70
Low power op amp	$ 0.32	20	Mouser	$6.40
500mA fuse	$ 0.26	10	Mouser	$2.60
EMI suppression beads	$ 0.08	10	Digikey	$0.80
Push button	$ 0.46	10	Mouser	$4.60
USB micro	$ 1.97	6	Mouser	$11.82
Green LED	$ 0.17	20	Mouser	$3.40
CT Energy Lithium Ion 2032 Battery Charger	$ 23.99	1	Amazon	$23.99
LIR 2032 Lithium Button Battery Charger	$ 15.99	1	Amazon	$15.99
AmazonBasics TPU	$ 29.71	1	Amazon	$29.71
SpiderMaker SpiderFlex	$ 36.00	1	Amazon	$36.00
PolyFlex TPU	$ 29.99	1	Amazon	$29.99
Ninjaflex TPE	$ 33.70	1	Amazon	$33.70
Long Flex Sensor	$ 12.95	20	Adafruit	$259.00
Short Flex Sensor	$ 11.95	10	Adafruit	$119.50
Pocket AVR Programmer	$ 18.50	1	Sparkfun	$18.50
AVR Programming Cable	$ 2.10	1	Sparkfun	$2.10
A000093	$ 22.10	2	Digikey	$44.20
A000005	$ 29.12	2	Digikey	$58.24
MCP6004-I/P	$ 0.63	2	Digikey	$1.26
MCP004-I/SL	$ 0.57	6	Digikey	$3.42
Fermerry 28AWG Silicone Electric Wire	$ 10.99	1	Amazon	$10.99
Tactile Switches	$ 0.55	5	Mouser	$2.75
USB micro B	$ 2.68	5	Digikey	$13.40
SanDisk 32GB Ultra Memory Card	$ 8.99	1	Amazon	$8.99
Female pin	$ 0.034	100	Mouser	$3.400
Female Connector	$ 0.045	30	Mouser	$1.350
Mini micro Open Barrel Crimping Tools	$ 19.99	1	Amazon	$19.99
Switch	$ 1.29	5	Mouser	$6.45
Neo G Active Wrist Support	$19.49	2	CVS Pharmacy	$38.98
Futuro Wrist Performance Comfort Support	$11.79	1	CVS Pharmacy	$11.79
McDavid Sport Wrist Wrap	$11.99	1	Walmart	$11.99
Velcro One-Wrap Thin Roll	$5.97	1	Walmart	$5.97
Velcro Sticky Back	$7.47	1	Walmart	$7.47
Singer Hand Sewing Thread Spools Kit	$2.00	1	Walmart	$2.00
TeamDIGIT_Wired_PCB_TQFP_V3	$7.97	3	OshPark	$23.91
TeamDIGIT_Wired_PCB_QFN_V4	$7.97	3	OshPark	$23.91
ITSYBITSY_SENSORY_V2	$3.90	3	OshPark	$11.70
NANO_SENSORY_2022_09_06	$3.40	3	OshPark	$10.20

Item	Individual Cost	Qty	Vendor	Cost
TeamDIGIT_WIRED_PCB_QFN_V2	$5.97	3	OshPark	$17.91
TeamDIGIT_Wired_PCB_TQFP_2022_09_06	$6.77	3	OshPark	$20.31
ATMEGA32U4	$5.80	3	DigiKey	$17.40
10k Resistor	$0.09	30	Mouser	$2.70
16MHz Crystal	$0.43	10	Mouser	$4.30
0603 10 µF Capacitor	$0.36	10	Mouser	$3.60
2-pin male connector	$0.04	30	Mouser	$1.20
Osh Park Overnight Shipping	$24.00	1	OshPark	$24.00
DigiKey Overnight Shipping	$12.99	1	DigiKey	$12.99
Mouser Overnight Shipping	$30.00	1	Mouser	$30.00
Total Spent			$1,380.96	

7.5.1 Total Cost

Table 7.5 outlines the total cost to produce one glove. The total cost is on the lower side as it is assuming that a roll of Polyflex TPU and Amazon TPU are already in possession. The cost for each roll of filament is around $29.99 which will result in the total cost jumping up to $191.80. The Excel Spreadsheet in the GitHub Repository Bill-of-Materials [52] named Cost for one Glove.xlsx includes the each item used for one glove, the quantity of each, and the total cost for the item.

Table 7.5 Total Cost for one glove

Item	Qty	Total Cost
TeamDIGIT_Wired_PCB_TQFP_V3	1	$ 7.97
3x2 Header	1	$0.46
22pF Capacitor	2	$0.36
0.1 µF Capacitor	1	$0.10
1 µF Capacitor	2	$0.24
10 µF capacitor	1	$0.36
10k Resistor	11	$0.99
1k Resistor	2	$0.20
16MHz Crystal	1	$0.43
ATMega32U4	1	$5.80
2-pin male connector	11	$0.44
2-pin female connector	11	$0.50
Female pins	22	$0.75
SMD Switch	1	$1.29
LED	2	$0.34
USB	1	$2.68
CAY16-220J4LF	1	$0.07
CG0603MLC-05E	2	$0.74
MF-MSMF050-2	1	$0.26
MH2029-300Y	1	$0.08
Pocket AVR Programmer	1	$18.50
Potentiometers	3	$0.81
Long Sensors	5	$64.75

Item	Qty	Total Cost
Material of fingers	10.9 grams	$0.44
Material of PCB case	13.38 grams	$0.40
McDavid Sport Wrist Wrap	1	$11.99
Velcro	1	$10.00
Thread	1	$2.00
Total	90	$132.94

8 SUMMARY

8.1 Future Recommendations

There are many aspects of the project that could be improved upon further by future teams. Marketing Requirement 2 said that our glove should be wireless. However, due to our sponsor not being too concerned about this, and the time constraints we already had, we did not accomplish this. Despite this, it would still be a valuable improvement to implement wireless capabilities as it could help with making the glove less awkward to work with during testing, and to put on and take off. It would also be beneficial to improve on the design and build of the glove in some ways. Two big changes would be 3D printing the entire glove and finding a way to embed sensors into the 3D print itself. These would again help with making the glove less unwieldly to work with, as well as making it lighter and easier to put on and remove.

Organizing the wires would also help with making the device less cumbersome. Currently, there are a lot of wires together in a small area, and they can get tangled. There are many options to organize them better including embedding them into the print, as mentioned before. Other solutions could be zip tying wires together, or even simply making them shorter. Lastly, adding another layer of complexity to the game could be beneficial in increasing patient's retention. One of the main goals of the game is to make sure patients are encouraged to keep progressing through the test, and currently there are implementations in the game to help with that, such as the point system and the ability to earn bonus points from completing tasks faster. Adding another layer to this would help even more with keeping players attention. However, future additions would have to be careful to not increase player discouragement in any way, from a lack of being able to accomplish something.

8.2 Reflection

There are a few things we wish we could have done differently during the design and development of the device. The first is obtaining earlier access to LabVIEW. Since LabVIEW is unique in design and operation from other programming software's, we had to learn how to use it from scratch. Obtaining access earlier would have helped with making this less stressful, as well as with finding and testing options to integrate it with Python code. We also wish we could have experimented with wireless PCBs. This was a Marketing Requirement we were unable to fulfill, and although it was not a priority for our sponsor, it would have been nice to have at least tested with some wireless options and compared them with our wired PCB.

Another improvement we wish we could have made is reducing the PCB layers from 4-layer to 2-layer. This would have lowered the cost for one PCB, as well as reduced the time it takes to fabricate. Lastly, we wish we could have started integrating our subprojects together earlier. We did not really start final integration until the last couple weeks, and although it worked out for us, it would have been helpful to start earlier and even experiment with different integration options. Overall, we are satisfied with our product and its ability to fulfill most of our requirements.

8.3 Conclusion

Our overall goal with this project was to help patients who have hand motor function impairments. To do this, we designed a medical sensory glove that can be used to track hand motion in real-time. In addition, we designed a game that can be used with the device to facilitate testing to help patients improve their hand motor functions. Our Marketing Requirements, Engineering Requirements, and design impacts helped shape the design, construction, and testing of our glove. For the individual parts of the device, we chose to use a TPU filament for the material of the glove, flex sensors and potentiometers-based sensors to measure hand movement, LabVIEW to handle signal processing, a custom ATMega32U4 PCB for our primary device on the glove, and the Python-based package, Pygame, to develop our game in.

REFERENCES

[1] "World Stroke Organization," [Online]. Available: https://www.world-stroke.org/world-stroke-day-campaign/why-stroke-matters/learn-about-stroke. [Accessed February 2022].

[2] "National Institute of Aging," [Online]. Available: https://www.nia.nih.gov/research/osbr/nia-small-business-showcase-flint-rehabilitation-devices. [Accessed February 2022].

[3] B. Sears, "verywell health," Dotdash, 25 August 2021. [Online]. Available: https://www.verywellhealth.com/physical-therapy-cost-5194917. [Accessed February 2022].

[4] "MusicGlove Hand Therapy for PC/MAC," [Online]. Available: https://www.flintrehab.com/product/musicglove-hand-therapy/. [Accessed February 2022].

[5] "Smart Glove," [Online]. Available: https://www.neofect.com/us/smart-glove. [Accessed February 2022].

[6] "CyberGlove Systems," [Online]. Available: http://www.cyberglovesystems.com/. [Accessed February 2022].

[7] B. Shoyat, C. Collinsworth, S. McElhannon and B. McDaniel, "Basketball Sensory Glove Final Design Report," 2021.

[8] "Satinior 2 Pair Black Half Finger Gloves Unisex Winter Stretchy Knit Fingerless Gloves, Common Size at Amazon Men’s Clothing store," SATINIOR Store, [Online]. Available: https://www.amazon.com/Satinior-Unisex-Finger-Stretchy-Fingerless/dp/B0796M3DL9/ref=sr_1_1_sspa?keywords=fingerless+glove&qid=1649209950&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExMkRENjJYS1BWOUFQJmVuY3J5cHRlZElkPUEwODExNDA1MkVZV1lBVUtaWVcwQSZlbm.

[9] "VRFree Technology," Sensoryx, [Online]. Available: https://www.sensoryx.com/company/technology/.

[10] "IEC 62304:2006," May 2006. [Online]. Available: https://www.iso.org/standard/38421.html.

[11] "IEC 60601-1-11:2015," January 2015. [Online]. Available: https://www.iso.org/standard/65529.html.

[12] "IEEE Guide for Developing System Requirements Specifications," *IEEE Std 1233, 1998 Edition,* pp. 1-36, December 1998.

[13] "IEEE 802.15.1-2005," July 2011. [Online]. Available: https://standards.ieee.org/ieee/802.15.1/3513/.

[14] "IPC-6011," July 1996. [Online]. Available: https://www.ipc.org/TOC/IPC-6011.pdf.

[15] IPC-2221 Task Group (D-31b) of the Rigid Printed Board Committe (D-30) of IPC, "IPC-2221A," February 1998. [Online]. Available: https://www.ipc.org/TOC/IPC-2221A.pdf.

[16] "Code of Federal Regulations," March 2018. [Online]. Available: https://www.fda.gov/medical-devices/overview-device-regulation/code-federal-regulations-cfr.

[17] T. DIGIT, "teamdigit24/Game," 29 November 2022. [Online]. Available: https://github.com/teamdigit24/Game.

[18] "Durometer Shore Hardness Scale," [Online]. Available: https://www.smooth-on.com/page/durometer-shore-hardness-scale/.

[19] A. P, "3D Printing Materials Guide: Plastics," 3D Natives, 08 June 2020. [Online]. Available: https://www.3dnatives.com/en/plastics-used-3d-printing110420174/#:~:text=ABS,flexible%2C%20and%20resistant%20to%20shocks..

[20] "What is TPE | Thermoplastic Elastomer Materials | HEXPOL TPE," 2021. [Online]. Available: https://www.hexpol.com/tpe/resources/tpe-academy/what-is-tpe/.

[21] "Fusion 360 | 3D CAD, CAM, CAE & PCB Cloud-Based Software | Autodesk," 8 January 2021. [Online]. Available: https://www.autodesk.com/products/fusion-360/overview.

[22] R. Brothers, "3D Printed Band for Apple Watch," 08 March 2017. [Online]. Available: https://learn.adafruit.com/3d-printed-band-for-apple-watch. [Accessed March 2022].

[23] T. DIGIT, "teamdigit24/3D-design," 28 November 2022. [Online]. Available: https://github.com/teamdigit24/3D-design/blob/main/PCB%20Case%20drawings%20inches%20no%20box.pdf.

[24] T. DIGIT, "3D-design/watch band1 v2.f3z at main · teamdigit24/3D-design," 29 November 2022. [Online]. Available: https://github.com/teamdigit24/3D-design/blob/main/watch%20band1%20v2.f3z.

[25] "Fusion 360 Snap Fit Cases | 3D-Printable Raspberry Pi Case," 16 August 2019. [Online]. Available: https://www.youtube.com/watch?v=E0NVC8xhf3I.

[26] "Thingiverse," [Online]. Available: https://www.thingiverse.com/.

[27] "Printables by Josef Prusa," [Online]. Available: https://www.printables.com/.

[28] "Flexi Hand by faah," 23 May 2018. [Online]. Available: https://www.thingiverse.com/thing:2907710. [Accessed 2022].

[29] "Flexy-Hand 2 by Gyrobot," 06 July 2014. [Online]. Available: https://www.thingiverse.com/thing:380665.

[30] "Iron Man Glove by Zefy," 21 March 2019. [Online]. Available: https://www.thingiverse.com/thing:3507482. [Accessed 2022].

[31] M. Burns, R. Rosa, Z. Akmal, J. Conway, D. Pei, E. King, N. Banerjee and R. Vinjamuri, "Design and Implementation of an Instrumented Data Glove that measures Kinematics and Dynamics of Human Hand," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2021.

[32] "Autodesk Inventor Software," 8 January 2021. [Online]. Available: https://www.autodesk.com/products/inventor/.

[33] "Product," [Online]. Available: https://www.ni.com/en-us/shop/software/products/labview.html.

[34] "USB-6211," [Online]. Available: https://www.ni.com/en-us/support/model.usb-6211.html.

[35] "Download EAGLE | Free Download | Autodesk," 29 January 2021. [Online]. Available:

https://www.autodesk.com/products/eagle/free-download.
[36] "Arduino Micro," [Online]. Available: https://store-usa.arduino.cc/products/arduino-micro.
[37] "Arduino Nano," [Online]. Available: https://store-usa.arduino.cc/products/arduino-nano/.
[38] "Arduino Nano 33 BLE," [Online]. Available: https://store-usa.arduino.cc/products/arduino-nano-33-ble.
[39] "Adafruit Bluefruit LE SPI Friend - Bluetooth Low Energy (BLE)," 2015. [Online]. Available: https://www.adafruit.com/product/2633?fbclid=IwAR2hGyffnSiUyYkkdMbeaxaKybCM0yzHwUBxYe5GSg-ygx0lTvVTlpOPlBs&gclid=CjwKCAjws--ZBhAXEiwAv-RNL__NXXB6kcx5m3e1tNx1DB37fyPJne5IOlVR91aQ5mC8zaAgipNbBBoCrKMQAvD_BwE.
[40] "Multisim Live Online Circuit Simulator," [Online]. Available: https://www.multisim.com/.
[41] T. DIGIT, "teamdigit24/PCB," 29 November 2022. [Online]. Available: https://github.com/teamdigit24/PCB.
[42] A. Industries, "Conductive Rubber Cord Stretch Sensor + extras!," [Online]. Available: https://www.adafruit.com/product/519.
[43] A. Industries, "Adafruit VL53L4CD Time of Flight Distance Sensor - ~1 to 1300mm," [Online]. Available: https://www.adafruit.com/product/5396.
[44] A. Industries, "Adafruit VCNL4040 Proximity and Lux Sensor - STEMMA QT / Qwiic," [Online]. Available: https://www.adafruit.com/product/4161.
[45] T. DIGIT, "teamdigit24/SensorConditioning," 29 November 2022. [Online]. Available: https://github.com/teamdigit24/SensorConditioning.
[46] "Serial Communication - learn.sparkfun.com," [Online]. Available: https://learn.sparkfun.com/tutorials/serial-communication/all.
[47] "PythonGameLibraries," [Online]. Available: https://wiki.python.org/moin/PythonGameLibraries.
[48] "Python Integration Toolkit for LabVIEW Download," [Online]. Available: https://www.ni.com/en-us/support/downloads/tools-network/download.python-integration-toolkit-for-labview.html#372372.
[49] T. DIGIT, "Team DIGIT 24 PCB - One_Sense," 29 November 2022. [Online]. Available: https://github.com/teamdigit24/PCB/tree/main/One_Sense.
[50] T. SCREEN, "TeamSCREEN/Eagle PCB Design v3.zip," 2 December 2021. [Online]. Available: https://github.com/SensoryBasketballGlove/TeamSCREEN/blob/main/Eagle%20PCB%20Design%20v3.zip.
[51] "Smartsheet," [Online]. Available: https://app.smartsheet.com/folders/personal.
[52] T. DIGIT, "teamdigit24/Bill-of-Materials," 29 November 2022. [Online]. Available: https://github.com/teamdigit24/Bill-of-Materials.
[53] R. M. Ford and C. S. Coulston, Design for Electrical and Computer Engineers, McGraw-Hill, 2008.
[54] "Difference Between Rechargeable and Non Rechargeable Batteries," 06 December 2015. [Online]. Available: https://pediaa.com/difference-between-rechargeable-and-non-rechargeable-batteries/#:~:text=Cost,cheaper%20compared%20to%20rechargeable%20ones..

Appendix A Objective Tree AHP Tables

This section explains how our group obtained the weights for each branch of the Objective Tree (Figure 3.1). Using the analytical hierarchy process (AHP) from Appendix B in the back of the book [53], we took our Marketing Requirements and categorized them into three objectives and ranked each one according to its relative importance. The geometric mean for each row is computed and totaled using the formula $GM = \sqrt[n]{\alpha_1 \alpha_2 \dots \alpha_n}$. The weight for each row is then computed by dividing the individual geometric mean by the sum of the means. For the weighted score, the weight from each row was multiplied by the category weight.

Table A.1 Sensory Glove AHP

	Functionality	Performance	Usability	GM	Weight
Functionality	1.00	0.20	0.33	0.40	0.10
Performance	5.00	1.00	3.00	2.47	0.64
Usability	3.00	0.33	1.00	1.00	0.26
Sum	9.00	1.53	4.33	3.87	1.00

Performance of the glove is most important because the overall goal of the glove is the track and record hand movements. Without performance, the project would be an ordinary glove that does not track hand movement. The second most important category is usability. Usability comes after performance because being able to use the glove with ease, no interference, and with a goal-based game is what makes the glove a product. The last category, functionality, comes after the other two because battery, wireless, and cleanable are not completely necessary for the system to operate.

Table A.2 Functionality AHP

	Wireless	Cleanable	Battery	GM	Weight	Weighted Score
Wireless	1.00	0.14	3.00	0.75	0.17	0.02
Cleanable	7.00	1.00	5.00	3.27	0.74	0.07
Battery	0.33	0.20	1.00	0.40	0.09	0.01
Sum	8.33	1.34	9.00	4.42	1.00	0.10

The cleanability of the glove is essential to ensure that our product can remain hygienic. This is especially important in a doctor/therapy setting, where the glove is planned to be primarily used. Therefore, cleanable is weighted with at least five times the importance of wireless and battery. Both designing the glove to be entirely wireless and using a rechargeable battery are goals we aim to accomplish to minimize our environmental impact. We prioritize wireless over battery due to wireless also contributing to the glove's requirement to not impair hand motion during testing. Furthermore, if the glove is not wireless, then a battery is not needed as the glove would be powered from the computer. Therefore, we weigh wireless as three times as important as battery.

Table A.3 Performance AHP

	Track Motion	3D Modeling	Real-time	GM	Weight	Weighted Score
Track Motion	1.00	3.00	7.00	2.76	0.68	0.44
3D Modeling	0.33	1.00	0.20	0.40	0.10	0.06
Real-time	0.14	5.00	1.00	0.89	0.22	0.14
Sum	1.47	9.00	8.20	4.05	1.00	0.64

The main objective of the glove is that it tracks the motion of the hand and wrist movements. Without this, the glove is nothing more than something the patient is wearing since it would not be providing data and feedback. For this reason, tracking motion is weighted three and six times higher than real time and 3D modeling respectively. The sponsor will be using the glove simultaneously with an EEG cap, so data needs to be streamed to a secondary device in real time with little to no lag. Hence, real time is weighted about twice as much as 3D modeling. Interest was expressed by the sponsor for the movements to be streamed onto a secondary device as a 3D model so that patients and therapists can see what is happening. However, this is not necessary for the glove to work as intended so it has a weight less than 10 percent.

Table A.4 Usability AHP

	Game	Easy	No Interference	GM	Weight	Weighted Score
Game	1.00	0.20	0.33	0.40	0.09	0.02
Easy	5.00	1.00	7.00	3.27	0.74	0.19
No Interference	3.00	0.14	1.00	0.75	0.17	0.04
Sum	9.00	1.34	8.33	4.42	1.00	0.25

When meeting with our sponsor, she emphasized that a lot of patients that use the glove, have rigid hands and such little movement that the glove needs to be easy to put on. Therefore, the category of easy/removeable is weighted about four times higher than the no interference and ten times higher than the inclusion of a game. If the circuity cannot be removed and the glove is not flexible, it can be very hard to put on and take off. In addition, the glove and its components should not get in the way of patient's movements and therapy session otherwise the data may have some error and it can make the session harder than it needs to be on the patient. Eventually it is desired that the glove will be driven by a game that is on the secondary device that is goal based and provides feedback to patients based on their progress. This is something that does not have to happen, but it would be nice if we are able to achieve this, hence why it has the lowest weight out of the three categories.

Appendix B Design Alternative

B.1 Sensors

In Table B.1 below, the weighted criteria on determining the best sensors to use for each type of measurement can be seen. Functionality of the sensors was weighted as the most important since data cannot be collected from the glove without a properly working sensor. The size was weighted as the next most important part of sensor criteria since the weight and size of the glove could impact a patient's hand movement and therefore interfere with the integrity of the data. Power consumption was weighted as the third most important criteria. While power consumption is important, since the glove does not have to be wireless it was less important than the other factors. Cost was weighted the lowest since it is medical equipment and the cost is not as critical as working, reliable sensors.

Table B.1 Sensor Selection AHP

	Functionality	Cost	Size	Power	GM	Weight
Functionality	1.00	7.00	3.00	5.00	3.20	0.564
Cost	0.14	1.00	0.20	0.33	0.31	0.055
Size	0.33	5.00	1.00	3.00	1.49	0.263
Power	0.20	3.00	0.33	1.00	0.67	0.118
Sum	1.67	16.00	4.53	9.33	5.67	1.00

In Table B.2 below, the weighted criteria on determining the best methods of processing the data on the glove can be seen. Accuracy and Real time processing were equally weighted since these factors are both critical to the main purpose of this sensory glove, which is to accurately read in and plot data in real time while the patient's hand is moving. Visualization was weighted as the second most important criteria since the hand motion data needs to be assessed and easily visualized to determine certain movements. Memory was weighted as the least important criteria because memory requirements for the microcontroller ~~will~~ be minimal.

Table B.2 Processing Technique Selection AHP

	Accuracy	Real-Time	Visualization	Memory	GM	Weight
Accuracy	1.00	1.00	5.00	7.00	2.43	0.424
Real-Time	1.00	1.00	5.00	7.00	2.43	0.424
Visualization	0.20	0.20	1.00	3.00	0.59	0.103
Memory	0.14	0.14	0.33	1.00	0.28	0.049
Sum	2.34	2.34	11.33	18.00	5.73	1.00

In Table B.3 below, the weighted criteria on determining the best methods of collecting the data from the sensors on the glove can be seen. Compatibility and real time transfer were weighted equally as the most important since they both are involved in sending data to the secondary device. Power was weighted as the least important since the glove does not need to be wireless.

Table B.3 Collection Technique Selection AHP

	Compatibility	Real-Time	Power	GM	Weight
Compatibility	1.00	1.00	5.00	1.71	0.455
Real-Time	1.00	1.00	5.00	1.71	0.455
Power	0.20	0.20	1.00	0.34	0.090
Sum	2.20	2.20	11.00	3.76	1.00

B.2 Power

The primary feature of the power supply is that it needs to be rechargeable, to minimize electronic waste. The power supply should provide at least 3.3 volts of power and last consecutive 60-minute sessions. The cost is not a major deciding factor when it comes to having the glove be wireless and selecting a power supply due to the fact that rechargeable batteries cost more to buy initially [54] and require a charger specifically for that battery. Table B.4 compares five different batteries in approximate weight (g), approximate size, cost of individual, nominal voltage (V), nominal capacity (mAh), and if it is rechargeable.

Table B.4 Battery Comparison

Name	Rechargeable	Brand	Cost	Approx. Weight (g)	Approx. Size	Nominal Voltage (V)	Nominal Capacity (mAh)
Lithium-Ion Battery	Yes	Adafruit	$12.50	34	4 x 1.4 x 0.3”	3.7	2000
Lithium-Ion Battery	Yes	Adafruit	$7.95	10.5	1.15 x 1.4 x 0.19”	3.7	500
EEMB LIR2450 Rechargeable Battery	Yes	EEMB Store	$18.99	5.3	Diameter: 24.5mm Height: 5.0mm	3.7	120
Rechargeable LIR2032 3.6V Button Cell FBA	Yes	Gernal	$7.19	20.13	0.79 x 0.79 x 0.13”	3.6	Not Provided
Lithium Ion 2032 Battery Charger for Coin Cell Rechargeable Batteries LIR20322	Yes	CT Energy	$23.99	2.8	Diameter: 20.0mm Height: 3.2mm	3.6	40

B.3 Game

This section explains how the AHP process was used to determine the best Python game development package to use for this project. The criteria used to measure the options against each other were package simplicity, speed, and community support available. As seen in Table B.5, community support was determined to be the most important factor. This is because widely available community support results in the overall design and coding process being easier. The simplicity of the package is also an important factor. With good support however, simplicity can be less important, which is why it is ranked below it. Speed was determined to be the least important criteria. This is because although speed is important, the game we plan to design ~~will~~ be simple in concept and should realistically not result in any latency issues.

Table B.5 Package Criteria AHP

	Simplicity	Speed	Support	GM	Weight
Simplicity	1.00	3.00	0.33	1.00	0.258
Speed	0.33	1.00	0.20	0.40	0.103
Support	3.00	5.00	1.00	2.47	0.638
Sum	4.33	9.00	1.53	3.87	1.00

The package options are compared below in Table B.6 in regard to simplicity. Pygame is by far the simplest, due to being built on a previously created C library. This makes using Pygame easy to work with, since it is entirely based on just function calls. Pyglet is not as simple as Pygame, since it requires a deeper level of knowledge about what is happening behind the scenes. However, it is still easier to understand than PyOpenGL, which requires a prior knowledge of OpenGL software.

Table B.6 Simplicity AHP

	Pygame	Pyglet	PyOpenGL	GM	Weight
Pygame	1.00	5.00	7.00	3.27	0.732
Pyglet	0.20	1.00	3.00	0.84	0.188
PyOpenGL	0.14	0.33	1.00	0.36	0.081
Sum	1.34	6.33	11.00	4.47	1.00

Table B.7 shows the comparison between the package options when looking at speed. Pyglet is known for its speed and is the fastest option available. This is followed by PyOpenGL, which itself is an enhanced version of Pygame. Pygame is the slowest option available, which is understandable due to its age.

Table B.7 Speed AHP

	Pygame	Pyglet	PyOpenGL	GM	Weight
Pygame	1.00	0.14	0.33	0.36	0.087
Pyglet	7.00	1.00	3.00	2.76	0.670
PyOpenGL	3.00	0.33	1.00	1.00	0.243
Sum	11.00	1.47	4.33	4.12	1.00

Table B.8 shows the comparison between package options with regard to available community support. Pygame is scored the highest, due to having an extensive collection of documentation and tutorials. Both Pyglet and PyOpenGL do not have a large community following and are in general, much less popular than Pygame. This has resulted in them not having as much documentation easily available. PyOpenGL does benefit from being built on Pygame however, which results in it scoring higher than Pyglet.

Table B.8 Support AHP

	Pygame	Pyglet	PyOpenGL	GM	Weight
Pygame	1.00	7.00	3.00	2.76	0.651
Pyglet	0.14	1.00	0.20	0.30	0.071
PyOpenGL	0.33	5.00	1.00	1.18	0.278
Sum	1.47	13	4.2	4.24	1.00

Table B.9 shows the final package decision based on the AHP tables. Based on each option's scores, Pygame is determined to be the best package option to use to develop the game with. Its simplicity and wide availability of community support are the main contributors in this decision. In addition, although Pygame suffers from being the slowest option available, the game we plan to develop should not require very fast software, due to its low complexity. Overall, these factors all contribute to Pygame being the best option.

Table B.9 Final Package AHP

	Simplicity	Speed	Support	GM	Weight	Weighted Score
Pygame	0.732	0.087	0.651	0.35	0.473	0.122
Pyglet	0.188	0.670	0.071	0.21	0.284	0.029
PyOpenGL	0.081	0.243	0.278	0.18	0.243	0.155
Sum	1.00	1.00	1.00	0.74	1.00	0.31

B.4 PCB/Microcontroller

In the Marking Requirements 2 and 9, Should be wireless and Should have a rechargeable battery, this schematic is the wireless PCB version as shown in the below. The design is the same as the wired version with additions to make it wireless and battery powered. Realistically the battery wouldn't be connected to the PCB if the USB was connected to the PCB, but a PMOS is put in place as a safety measure (the 5V from the USB wouldn't back-feed voltage into the battery). And an LDO (low-drop-out) 3.3V voltage regulator was added to supply the circuitry with a clean 3.3V voltage source. A few other additions are a connector for the battery along with a voltage divider circuitry to measure the voltage of the battery, and finally a 6-pin connector for the Bluetooth module. This connector may change if a different Bluetooth module is chosen. The downside to this version is that the ATMega32U4 can only run at an 8MHz clock speed due to the input voltage being less than 4.15V.

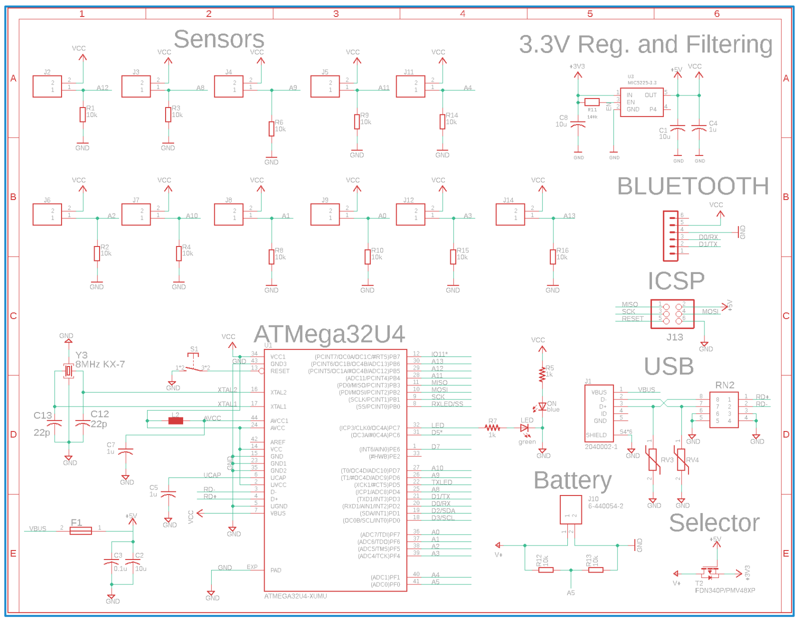


Figure B.1 Eagle Wireless Schematic

Table B.10 shows the AHP the processor chip decision based on the AHP table and based on the processor chip comparison found Table 4.14. In Table B.10 the deciding factors of the processor chip are operating voltage, ADC pins, LabVIEW compatibility, and built-in UART. Each deciding factor was weighed in order of importance with the most weight being the most important deciding factor.

Table B.10 Processor Chip AHP

	Operating Voltage	ADC Pins	LabVIEW Compatibility	Built-In UART	GM	Weight
Operating Voltage	1.00	0.33	0.14	0.20	0.31	0.055
ADC Pins	3.00	1.00	0.33	0.20	0.67	0.118
LabVIEW Compatibility	7.00	3.00	1.00	5.00	3.20	0.563
Built-In UART	5.00	5.00	0.20	1.00	1.50	0.264
Sum	16.00	9.33	1.67	6.40	5.68	1.00

B.5 Glove Material

The material of the glove is critical to the project because it needs to be flexible enough that it can it on all hand sizes and easy to put on hands that are rigid from stroke, ALS, Parkinson's, and other disorders and disabilities that cause hand impairment. The material should not cause irritation to the while the user is wearing it and performing the tasks on prompted during the game. If the glove is used in a therapist's office or in a setting where multiple people use the same glove, it needs to be easy to clean the glove. The main attribute of the glove and the material of the glove is that it should not interfere with the patients' movements or hinder them in any way. The material of the glove should not be too heavy or bulky. Table B.11 compares four different flexible filaments in cost, diameter, and shore hardness.

Table B.11 Material of Glove Comparison

Name	Brand	Cost	Diameter	Shore Hardness
TPU	SainSmart	$15.99	1.75mm	95A
Flexible PLA+	ATARAXIA Art	$39.99	1.75mm	89A
TPC	Digi-Key	$39.99	1.75mm	95A
TPE	Digi-Key	$29.99	3.00mm	Not Provided