Climber's Ligament Injury Mitigation Band

Alexander Nguyen Author, Jubahed Qayum Author, and Joshua Ramos Author

Department of Electrical and Computer Engineering, Carnegie Mellon University

Abstract**—**In this paper, CLIMB, a system capable of monitoring, analyzing, and visualizing forces applied by the hand over time, is proposed. In the form of a glove-like wearable device, CLIMB aims to improve the state of training and rehabilitation in rock-climbing by providing climbers with an unobtrusive, injury-preventative training device. As a result, climbers using CLIMB will avoid overloading their pulley-joints during their workouts, and will identify areas of improvement in their rehabilitation/training via the CLIMB–mobile app.

Index Terms**—**

B: Sensor baseline reading

CLIMB: Climber's Ligament Injury Mitigation band

Γ: Force threshold value

Hang-board: a training tool that helps improve hand and finger strength for climbing and bouldering

Pulley-injury: a tear or rupture of the ligamentous bands that hold the tendons in place in the fingers.

Pulley-joint: a ring-shaped ligament in the finger that holds the flexor tendons in place against the finger bones.

Pulley-tendon: a fibrous band of connective tissue that stabilizes the tendons in the fingers.

I. Introduction

The International Federation of Sport Climbing (IFSC) estimates that as of 2019, there are approximately 44.5 million climbers around the world. Climbing (in all of its forms) is a sport that comes with an inherent risk of physical injury, and studies have found that as much as 81% [1] of climbers have experienced at least one injury over the course of a single year. Of these, finger (pulley) injuries were by far the most common, constituting approximately 61% of all injuries.

A pulley injury occurs when one or multiple pulleys in the fingers, categorized from A1 to A5, ruptures. The pulley ligaments are responsible for holding the finger tendons tight to the bones, so when they are overloaded with weight, they tear, unable to withstand the force of the constricting muscles attached to the tendons. [2] These injuries are painful in both the short term and during the long term recovery, and the most severe pulley injuries (Grade IV) require surgery and months of rehabilitation to recover. Thus, it is crucial for climbers to be able to prevent these types of injuries as best they can.

There are several ways for climbers to mitigate the risk of pulley injuries; the most prominent of these solutions is the PulleyPal, a finger splint which aims to support the pulley by squeezing it against the finger bone. Additionally, climbers can train their fingers to increase pulley strength and increase the maximum load of each finger. However, while careful training and finger splints aim to assist the pulley ligaments through physical support and strengthening, there is currently no solution on the market that can accurately predict and prevent these injuries before they occur in real time. This project aims to fill the gap in pulley injury prevention for climbers.

The primary goal of CLIMB is to mitigate the risk of pulley injuries occurring during a climbing session by equipping climbers with a wearable device that can monitor force readings from the fingers, determine the moment at which a pulley injury is likely to occur, and immediately alert the user in time for them to return to a safe position. The secondary goal is to further assist climbers in their training and preparation by providing the wearer with weight distribution metrics and data analysis after each climbing session through a mobile application.

II. Use-Case Requirements

A. Ergonomics: Comfortable and Unobtrusive

The device will be comfortable and unobtrusive to the climber. It is imperative that the structure of the wearable covers no more than 30% of a climber's finger surface area and is flexible to avoid applying restraint. This is also so that climbers can preserve their grip (surface area contact with the wall).

B. Durability

The device will be robust and ready to tackle any climbing situation. Therefore, the glove and sensors will support 90 degree bends while accurately measuring up to 120 pounds of direct, perpendicular force per joint over numerous uses.

C. Safety: Injury Prevention

The device will actively work to prevent pulley injuries. Thus, it will maintain an on-device haptic alarm system that fires within 200ms whenever a user applies force on a pulley-joint that encroaches the safety boundary. Further, the obtained sensor readings must be accurate and fine-grained with a maximum error of ±2.5%,

D. Economic: Mitigate Healthcare Costs

The device will eliminate the need for surgical intervention due to pulley-injury. By preventing pulley-injuries, this device will relieve climber's of the potential $9,532 price-tag [4] for tendon repair.

E. Social: Feedback-driven

Lastly, the visualization component of the system, which provides user's analytical feedback, will be seamless, user-friendly, and suggestive toward improvement. To achieve this, the force readings from any given workout will be communicated to the mobile application within 10s of workout completion. Further, user's will be able to compare their training progress with others.

III. ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

The architecture of the system is composed of two central modules: hardware and software. Together, these components operate to obtain applied-force readings, raise haptic alarms, process the readings into suggestive metrics, and visualize these metrics as analytics on a mobile application.

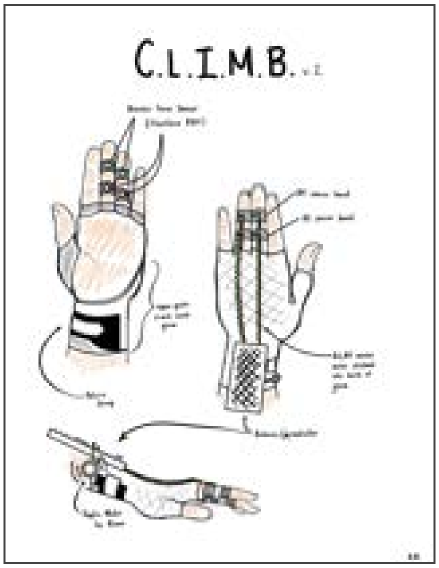


Fig 1. Wearable device design

In Figure 1, the placement of supporting hardware (ie. PCB, Sensors) and the structure of the fabric encompassing the hand is emphasized to depict the flexibility and unobtrusiveness of the device. This component acts mainly as the data-collection layer of the design by procuring force readings over an interval of time and packaging/shipping them to the processing layer. The secondary operation of this component is to drive a haptic alarm upon the encroachment of *dangerous* force-reading levels.

In Figure 2, the mobile application describes the obtained force readings analytically. Prior to displaying these analytics, the readings are processed and stored on the mobile device. Additionally, the haptic-alarm threshold can be set via the mobile application.

In Figure 3, the system and its subsystems are depicted to describe the flow of data. The implementation details of this system will be further discussed in section VI.

Overall, the hardware is maintained by a lightweight, flexible fabric, and the software requires a smart-device to access the mobile application.

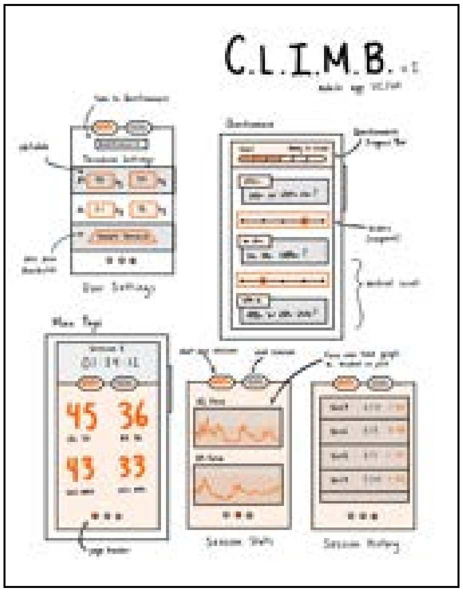
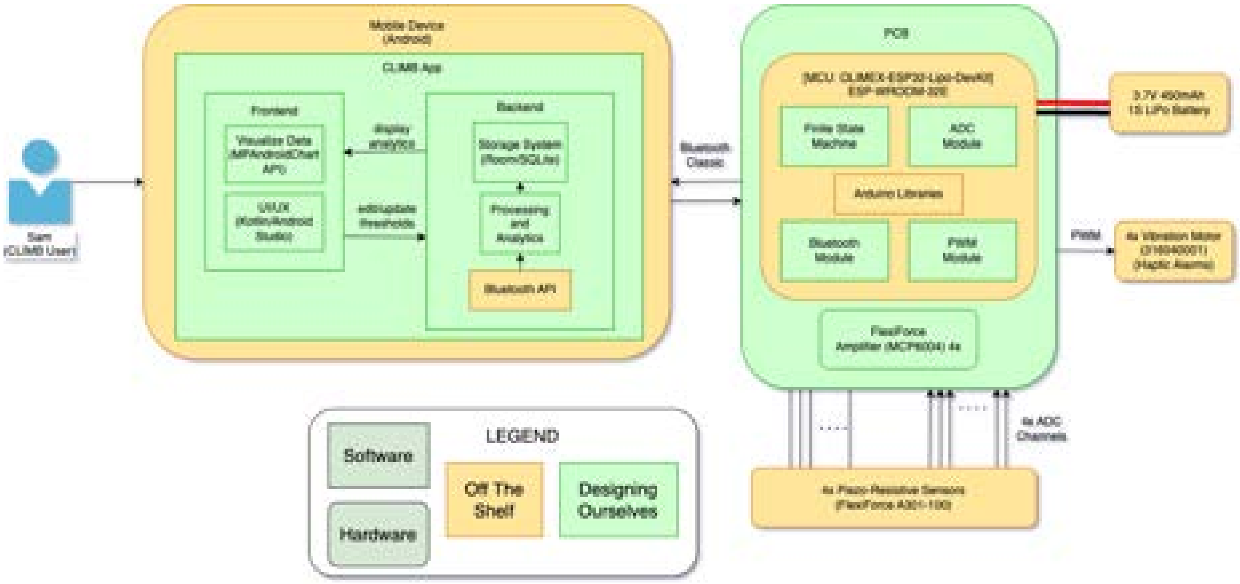


Fig 2. Mobile application design

Fig. 3. System specification block diagram



A. *Principles of Engineering*

Design modularization and skill-based task assignment are critical for achieving project efficiency and collaboration. Breaking complex deliverables into smaller, manageable modules enables parallel development, simplifies debugging, and improves scalability. Assigning tasks based on team members' skills and skill-maturity ensures that work is completed effectively while fostering growth opportunities. This approach allowed our team to work concurrently, manage progress through clear ownership and milestones, and mitigate risks by isolating interdependencies. Well-defined interfaces between modules streamlined collaboration and communication, resulting in faster delivery, higher quality outcomes, and optimized resource utilization.

B. *Principles of Science*

During the initial stages of the project, we conducted biomechanics research to understand and determine the various forces and stressors affecting the finger muscles during rock-climbing. In doing so, we identified that the A2 and A4 pulleys are responsible for the majority of forces endured during rock climbing. This research contributed to our eventual placement of piezoresistive sensors for the most accurate force readings relevant to pulley injuries for each pulley.

While designing and tuning our amplifier circuit, we performed circuit analysis to achieve our desired amplified output range while respecting the physical constraints of our system such as the absolute maximum and minimum ratings (ie. current) of our operational amplifier integrated chips, MCUs, and piezo-resisitive sensors. This involved designing different kinds of amplifier circuits (ie. inverting, non-inverting, feedback) and verifying the physical constraints of each design using component datasheets.

C. *Principles of Mathematics*

We performed many calculations to determine the optimal system parameters of our subsystems.

To decide which Bluetooth paradigm (communication speed), buffer size, and force-reading sampling rate to use in our system, we computed the time it would take to communicate data with Classic vs. Low-Energy speeds to send data of various buffer sizes, which are filled at varying sampling speeds. This allowed us to choose the optimal size, paradigm, and sampling rate given the performance tradeoffs we observed to communicate data within a time that respected our feedback requirements.

To decide the resistance of R3 (Fig. 8), we performed several weight experiments and recorded lines-of-best-fit for V_{out} vs. Force for various values of R3 to determine which resistance value would produce a curve that is most accurate (linear) since this parameter is directly proportional to the gain of the amplifier circuit.

To eliminate noise due to floating pressure on the sensor, we implemented a calibration routine that collected the force readings of the glove for ten seconds, averaged the recordings, and used the average to normalize every recording thereafter.

IV. Design Requirements

A. *Ergonomic Requirements*

The hardware component will be supported by a standard fabric crack climbing glove, which is known for its durability, comfort, and minimal invasiveness. Furthermore, thin and flexible sensors will be used for data-collection. Since the average adult finger is characterized with a length and width of 80mm x 20mm [3], then to achieve a maximum 30% area reduction with 2 sensors per finger:

$$80\,mm \times 20\,mm = 1600\,mm^2$$

$$(1600\,mm^2 \times 0.3)\,/\,2 = 240\,mm^2$$

$$\sqrt{240\,mm^2} = 15.49\,mm$$

The largest size sensor the design will accommodate is 15.49 mm x 15.49 mm.

B. *Durability Requirements*

The sensor must be capable of bending 90 degrees while undergoing 534N (54 kg) of constant, direct force without tearing. Over the course of 100 uses, the sensor must not lose more than 2.5% of sensitivity and the wearable fabric must not tear or lose formation.

C. *Safety Requirements*

The sensors will be calibrated upon initialization, and the force readings will be measured accurately with fine-grain by amplifying the sensor reading range to support up to 54 kg. Further, using on-device peripherals, changes of at least 5mV/kg will be identified to ensure sensor precision. Moreover, readings within 20% of the safety threshold will signal the alarm subsystem within 100ms to drive the first haptic. Upon threshold excession, a second signal will be fired within 100ms to drive the first haptic again and begin driving a second haptic.

D. *Economic Requirements*

The economic requirements are directly dependent on the safety requirements. By preventing injuries, worst-case healthcare costs are mitigated.

E. *Social / Feedback Requirements*

To ensure data transport within 10s, there must be a sufficient communication speed. Given a workout of 10 minutes, at a 10 Hz sampling rate, with 32-bit force reading values:

$$10\,min \times 60\,sec/min \times 10/sec \times 32\,bits = 192\,Kb$$

$$4\,sensors \times 192\,Kb\,/\,10\,sec = 76.8\,Kbps$$

The system needs to maintain a minimum data transfer speed of 77 Kbps.

V. Design Trade Studies

A. *MCU Platform*

Four criteria for selecting an MCU were considered: computational capability, power consumption, size, and connectivity. As described in Table I, the Raspberry Pi is certainly the most computationally capable of the controllers and supports more connection methods, but it consumes significantly more power and is much larger. While it does contain Bluetooth capabilities, it lacks built-in peripherals ADC, so its use in this system would require additional equipment, which would add more size. The STM32F4E platform has the lowest power consumption of the three, but it would require additional modules to support Bluetooth, increasing its overall size factor and power consumption.

Table I. Comparison of Microcontroller Platforms

Attributes	**MCU Platform**			
	STM32F4E	***HiLetgo ESP-WROOM 32***	***Olimex ESP-WROOM 32E DevKit LiPo***	***RPi***
Clock	160 MHz	240 MHz	240 MHz	Up to 1.8 GHz
Capabilities	ADC, GPIO, UART, I2C, (Bluetooth with HC-05)	ADC, GPIO, UART, I2C, Bluetooth, WiFI	ADC, GPIO, UART, I2C, Bluetooth, WiFI, LiPo-Power	GPIO, WiFi, Bluetooth, HDMI, USB, Ethernet, PCIe
Power Consumption	0.3 W	0.6 W	0.6 W	<= 6W
Built-in Exposed ADCs	9	15	16	0
Max Storage	4 GB	4 GB	4 GB	8 GB
RAM	256 KB	320 KB	320 KB	1 GB
Size	68 mm x 53 mm x 13 mm	51 mm x 25 mm x 13 mm	48 mm x 28 mm x 13 mm	85 mm x 56 mm x 17 mm
Familiarity	Very	None	None	None

While the ESP32 consumes slightly more power than the STM32, the miniature size, built-in Bluetooth capabilities, peripherals providing extensibility, and more capable CPU contribute to its selection. Furthermore, although the system developers lack familiarity with the ESP32, it is supported by numerous Arduino libraries and documentation, allowing for quick understanding, iteration, and development. Between the HiLetgo and Olimex breakout board models of the ESP-WROOM32, there is a minute, key difference: LiPo-power support. The Olimex ESP32 DevKit LiPo board uses a low dropout regulator (LDO), which steps down our LiPo battery's voltage (3.7V nominal) to 3.3V, supplying power to the ESP32 and peripherals. This aspect of the Olimex model, along with the mentioned ESP32 trade-offs, is what guided our decision to utilize this board for CLIMB.

B. *Force Sensors*

To support this system, durable, thin, and flexible force-sensitive resistors were investigated to determine a suitable sensor for obtaining accurate force readings. Beyond the sensors considered in this trade-study, there are a number of force-sensitive devices for obtaining force readings, however, to abide by the system's design requirements, piezo-resistive sensors were selected due to their lightweight and simple nature. Of the contenders, the Tekscan A301-100 was selected.

Table II. Comparison of Sensors

Attributes	**Force Sensitive Resistors (FSRs)**		
	Interlink FSR 402	***Tekscan A301-100***	***ALPHA MF01A-N-221-A 06***
Diameter	18.28mm	14.00 mm	14.7 mm
Thickness	0.2 – 1.25mm	0.203 mm	0.205 mm
Flexibility	yes	yes	yes
Repeatability Accuracy	±2%	±2.5%	±2%
Durability (# actuations)	≥ 10 million	≥ 3 million	> 10 million
Range	0.1N – 100N	0N – 4,448N	0.3N – 9.8N
Response Time	< 3µs	< 5µs	< 5 ms

In comparison, the Tekscan has comparable thickness, flexibility, and response time. In contrast, the Tekscan is less durable and slightly less accurate, however, it provides a significantly larger force-range and smaller diameter.

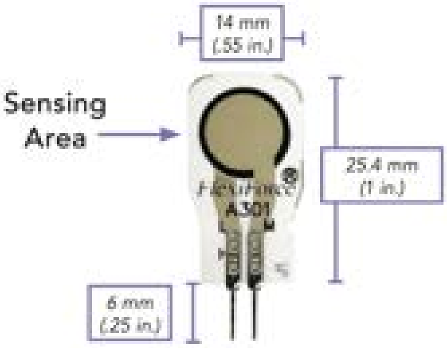


Fig 4. Tekscan FlexiForce A301-100 Sensor [7]

Overall, the deciding factors were the A301's range and diameter. To satisfy the design requirements, the sensor must read up to 54 kg (534N) and obtrude up to 30% of the finger. The Interlink option fails to support the area requirements, while both fail to satisfy the force-reading requirements; thus, the Tekscan is the sensor for this application. [5] [6]

C. *Communication Paradigm*

The process by which force-readings will be transported to the processing layer will follow one of two schemes: (1) send the force readings in real-time in parallel with measurement, or (2) collect the force readings, and send them all at the end of a session.

Option (1) has the benefit of instant, real-time analytics. A user can see the data propagating throughout the system during their climb. However, this approach implies that a climber needs to keep their phone nearby to sustain connectivity while climbing, potentially increasing the risk of damage or injury if their phone is hanging around in their pocket or if the climber falls on it.

Furthermore, option (1) introduces more overhead, whereas option (2) relieves the system of real-time complexities such as scheduling and interrupt-handling, providing a more simplistic approach. Moreover, because climber's are focused on their climbing, it is unnecessary to provide real-time analytics as it provides no value to the climber, and fails to address any design requirements.

While it may require larger memory utilization on-device, the risk of injury or disconnection is mitigated, and overhead and complexity is minimized with option (2).

D. *Android App Development Platform*

When choosing an android app development platform, both Android Studio/Kotlin and the Flutter/Dart framework were considered. Developing apps in Flutter has several benefits, including cross platform capabilities, a diverse set of widgets, and the ability to use a single codebase for multiple platforms.

In contrast, app development in Kotlin has the built in advantage of native performance, which comes as a result of Kotlin compiling to native code (which typically results in smoother execution when compared to Flutter's rendering engine). Additionally, when combined with Android Studio, developing in Kotlin gives the developer access to the full breadth of available Android APIs making it easy to integrate with native features, services, and libraries. Android Studio also provides a rich array of tools for android development, including an Android phone emulator which is incredibly useful for debugging and testing.

Each system has many unique benefits, however in the end Android Studio/Kotlin was selected for frontend and backend development for its advantages specifically in native Android development.

VI. SYSTEM IMPLEMENTATION

The following implementation describes the architecture for a one-handed device. To support two hands, simply double the system architecture with two PCBs, two MCU, etc. Though there will be slight modifications in the reception/transmission of user-data. This is described in the *Dual-Wielding* section.

A. *Printed Circuit Board*

At the base of this system is a printed circuit board maintaining all of the electrical components. In particular, it allows for the mounting of an MCU (Fig. 7) via female header pins to allow for flexible device testing. It routes the MCU's ADC units to the amplified sensor reading outputs for data collection. It routes the MCU GPIO pins to four haptic vibration motors (see Fig. 9) which comprises the alarm system. It routes the amplifier circuit depicted in Fig. 6. Lastly, it serves as a strong, durable base for the device. The PCB was designed using the Fusion CAD software and manufactured by OSH PARK.

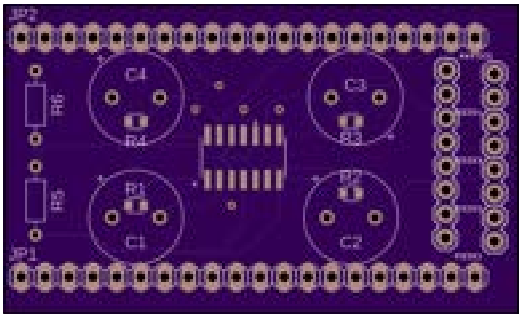


Fig 5. CLIMB Printed Circuit Board

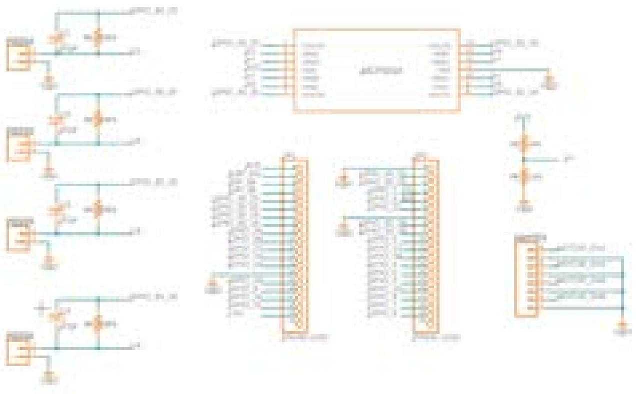


Fig 6. CLIMB PCB Schematic

B. *Processing Unit: Olimex ESP32-DevKit-LiPo*

Aboard the ESP32, the Arduino ESP32 libraries were used to program the controller's peripherals to interface with the surrounding electrical components. Furthermore, a Bluetooth (Classic) procedure was implemented to allow for serial communication with the mobile device. Upon initialization, the MCU waits for a start signal from the mobile device via Bluetooth. This signal will be followed by the user-set force threshold values for each A2 and A4 pulleys, which are compared to the real-time force-converted ADC readings.

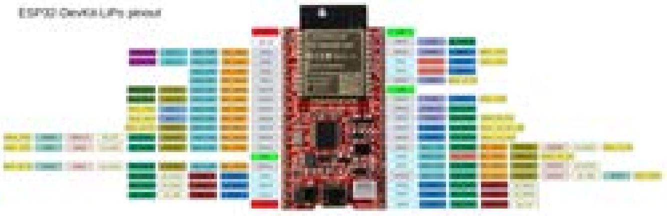


Fig 7. ESP32 Layout

The ADC units are programmed to read from 4 channels at a 10 Hz frequency with a 12-bit resolution such that miniscule changes in force readings can be detected rapidly. The GPIO units are programmed to interface the system haptics when the force-converted ADC readings exceed the user-set force threshold. After receiving the start signal, the MCU begins processing the readings, constantly monitoring the threshold and recording the readings in a buffer. Upon receiving a stop signal from the mobile device, the MCU stops collection and begins data transmission to the mobile device. It then clears the buffer, and idles until the next session starts.

C. Sensing and Amplification

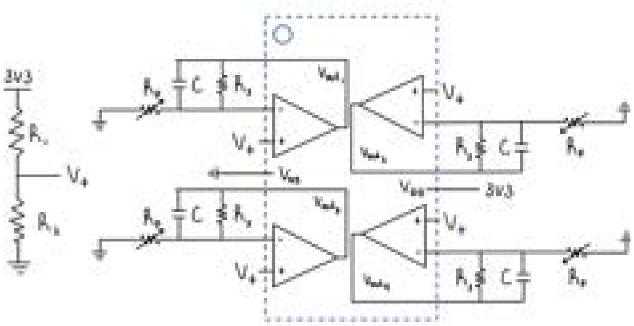


Fig 8. Sensor Amplifier Circuit

The A301-100 sensors (Fig. 4) are wired as described in Fig. 6 to a non-inverting operational amplifier, resulting in the following gain representation and output:

$$V_{out} = V_{+}(1 + \frac{R_P}{R_3})$$

$$R_1 = 47\ K\Omega,\ R_2 = 2.2K\Omega, R_3 = 837K\Omega,\ V_S = 3.3V$$

$$R_P = [\sim 100\ K\Omega,\ \sim 3M\Omega],\ C = 47pF,$$

$$V_{+} = 0.14V,\ I_{+} = 67\mu A$$

$$V_{out} \in [0.14V,\ 3.3V]$$

This model provides a 3.16V reading range for the ADC, and hence force-detection. Further, this range respects the ESP32's ADC voltage range (constraint) of 0V to 3.3V, utilizing ~95.8% of it. The R1 and R2 values were chosen to respect our amplifier's (MCP6004) operational requirements [13], which specify an absolute maximum current rating at the analog input pins (V+/-) of ±2mA. Hence, we chose a pair of resistors that would respect this constraint, while maximizing the voltage range of V_{out}.

Furthermore, the sensors were attached onto the wearable fabric while having their prongs directed towards the back of the hand, where they were wired to the MCU's ADCs via an unobtrusive wire.

D. Haptic Engagement

The on-device haptics will consist of four vibration motors shown in Fig. 9 [8]. These motors are driven by pulse-width-modulating (PWM) signals (4x) that each increase in pulse-width as the force readings of their corresponding pulley encroach their safety threshold. This provides the user with more gradual feedback.

Fig 9. Haptic Vibration Motor [9]

The magnitude of the PWM signal is dependent on the error between the force-threshold of a pulley and its real-time force readings, updating every 100ms, such that if the error is within 30%, 1V is driven, within 20% 2V are driven, and within 10% 3V are driven, resulting in incrementally stronger haptic alarms.

E. Android Application Implementation

The user can access all of the weight distribution data collected during their session through the CLIMB mobile app on their Android device. This app was built in Android Studio and the frontend was written using a combination of Kotlin and XML, while the backend was written entirely in Kotlin. There are four main screens that the user can interact with: the home screen, the session data screen, the session history screen, and the threshold settings screen.

Upon opening the app, the user sees the home screen, where key metrics and data points from the last recorded session are displayed. This includes the A2 threshold and A4 threshold (in Kgs) as well as the maximum singular force recorded by sensors on each of the A2 and A4 pulleys during the previous session. The session statistics screen displays graphs of raw force data over time for each pulley (A2 Middle, A2 Ring, A4 Middle, A4 Ring) that has been monitored. Each of these graphs is fully interactive, and the user can zoom in and scroll along both the X and Y axes. The session history screen can be accessed via a button at the bottom of the session data screen. This screen displays a list of all past climbing sessions and will allow the user to view the session data from that session. Finally, the user settings screen gives the user the ability to manually change the alarm thresholds for each individual pulley sensor on the wearable device. In order to update these thresholds in the wearable device, the wearable device must be paired and connected via Bluetooth with the user's mobile device. As such, once the user adjusts the thresholds and presses the "Update Thresholds" button, the new threshold values for each sensor will be stored within the app. When the next session is started by the user, the app will confirm that the device is connected to the app via Bluetooth (which is always done as part of the start sequence) and sends the new thresholds to the device's MCU as part of the start signal message in the following format:

startMsg = "START[A2T1][A2T2][A4T1][A4T2]",

where A2T1, A2T2, A4T1, and A4T2 are passed as two-digit numbers representing the respective pulley sensor thresholds set in the app. These thresholds are then parsed from the bluetooth start message within the ESP32 and updated accordingly.

There are two components that are always visible to the user no matter which screen they are viewing: the session utility bar which is situated at the top of the app layout, and the navigation bar which is located at the bottom of the app layout (see Fig. 10). The navigation bar contains three icons that the user can press to quickly switch between the app's three main screens. The session utility bar serves several important functions which we will elaborate upon further.

Fig 10. Always Present Components

To start and end a climbing (force tracking) session, there is a start/stop session button located within the session utility bar. It has three functions, depending on the current state of the app (see Fig 11.):

1) START → default state, while no session has begun and no data transfer is occurring. Pressing the button in this state will start a session and launch the session start sequence.
2) STOP → during an active session, pressing the button will stop the session and begin the transfer of session data from the device to the app.
3) LOADING → immediately following a session (stop button press) the button will display the text "LOADING". While session data transfers to the app, the button is not pressable to avoid unexpected app behaviors.

Fig 11. Start/Stop Button States

Additionally, the session utility bar contains information relevant to each session, namely the session ID, current date (month/day/year), and session duration (which updates each second during an active session). This information is later used by the session history tab to identify past workouts.

Once a session has been completed and the session data has been received by the app via Bluetooth, the data is processed and analyzed. All of the app's screens are then populated with the most recent session's data accordingly.

F. Sensor Calibration Implementation

The piezoresistive sensors are attached to the fingers through elastic bands wrapped around both the sensor and the finger at each pulley location. Because of this, there is some force applied to the sensors by the elastic band which needs to be accounted for during the session. Adjusted alarm thresholds will be calculated by a calibration routine which will run before each session begins.

When the user starts a session, the app makes sure that the wearable device is paired and connected to the mobile device. Once this is confirmed, the app and wearable device commence the sensor calibration routine. First, the app begins the routine by sending a start signal, along with the alarm thresholds for each pulley sensor, to the wearable device. When the wearable device receives this start signal, it takes sample force readings over a period of 10 seconds (approximately 100 readings in total). The user will also need to be wearing the wearable device during calibration or the calibration will not succeed, so if the returned readings are abnormally low (close to zero kgs), the calibration fails and requires the user to start a new session. If the calibration fails, the wearable device sends a NACK signal back to the app, which lets the app know that the calibration has failed. After receiving the NACK signal, the app will terminate and delete the session. As a safeguard, there is an indicator on the app screen during calibration telling the user not to disconnect or take off the wearable device while calibrating.

Once non-zero (legitimate) force readings have been returned, the MCU calculates the baseline force measurement for the session for each individual sensor. This is calculated individually for each sensor by taking the mean of all force readings from each sensor taken over the ten second calibration period using the mean equation:

$$B_i = (\Sigma\, x_1, x_2, x_3, x_4, \ldots, x_n) / n \,,$$

for $i = 1,2,3,4$ and n = total number of readings

After the baseline force reading for the A2 pulley sensor on the index and middle fingers (B_1, B_2 respectively) and the baseline force reading for the A4 pulley sensor on the index and middle fingers (B_1, B_2 respectively) have been calculated, the adjusted alarm thresholds must be calculated accordingly, before being sent to the wearable device to be updated. In order to do so, the raw thresholds (Γ_{raw}) that were sent from the app must be converted from grams (kg) to a converted threshold (Γ_{conv}) in ADC readings:

$$\Gamma_{conv,i} = -41.2 + 83.6(\Gamma_{raw,i}) - 0.302(\Gamma_{raw,i})^2 \,,$$

for i = 1,2,3,4, where Γ_{conv} are ADC readings and Γ_{raw} is in kg.

Once the raw thresholds from the app have been converted to millivolts, the adjusted alarm threshold (Γ_{adj}) for each pulley sensor can be calculated:

$$\Gamma_{adj,i} = \Gamma_{conv,i} - B_i \,,\ \text{for i} = 1,2,3,4$$

where Γ_{conv} = corresponding threshold set in the app

Finally, when the calibration routine has successfully completed (and the updated adjusted thresholds have been sent to the wearable device), the wearable device sends an ACK signal back to the app to indicate successful calibration and the session will start.

G. Data Visualization & Analysis

When the session ends, the app attempts to pair and reconnect with the wearable via Bluetooth. Once the wearable is connected to the app, it sends a signal to the wearable device indicating the end of the session. Once the wearable receives this signal, it stops collecting force data and sends all data from the session to the app via Bluetooth. During this period, it is important to note that the start session button is disabled, preventing a user from starting a new session while data from the previous session is being sent to the app. When all the data has been sent to the app, the wearable sends a signal to the app indicating that all data has been transmitted. This data is stored in a database (further elaborated upon in the next section).

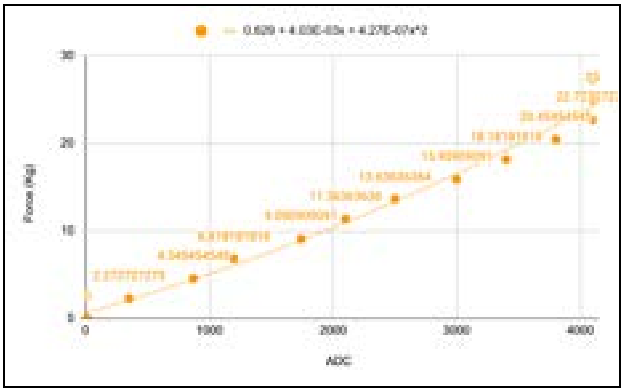


Fig 12. Force vs. ADC (Pin-point)

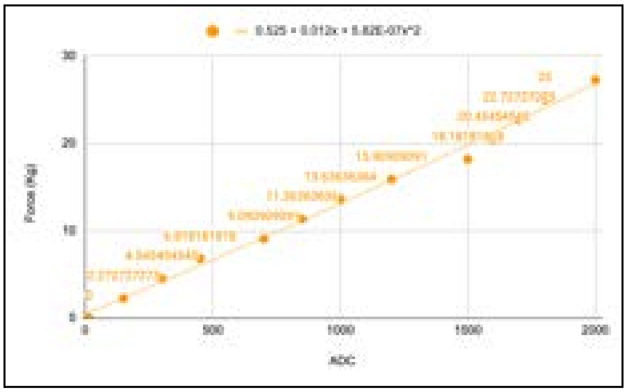


Fig 13. Force vs. ADC (Finger-flush)

When analyzing the data from pin-point-on-sensor and finger-flush-to-sensor experiments, the force readings received from the wearable are converted from ADC readings to pounds. This conversion to kg occurs on the MCU using the following formulation of the finger-flush model (Fig. 13), and then shipped to the mobile app.

$$f_{kg} = 0.525 + 0.012 * f_{adc} + 5.82 f_{adc}^{\,2} * 10^{-7}$$

where f_{kg} is the force reading in kilograms and f_{adc} is the force reading from the ADC

Then, key data points (session maximum force, high risk percentage) are extracted from the converted force readings from each sensor. Using the MPAndroidChart open-source API, the converted session data is displayed as a line graph with a y-axis of force (in kilograms-Force) and x-axis of time (where each discrete point is a measure over a delta of 100 milliseconds).

Fig 14. Session Data Visualization

H. Database Implementation

The database plays an integral role in the functioning of the app. It is the central, connecting point between the hardware platform and the app frontend. At its core, it is based on SQLite, an embedded open-source database management system. The largest benefit of using SQLite for the database is that it is lightweight and universally compatible: in other words, it allows the app (even if running on a moderately weak mobile device) to store and query data in a completely standalone manner, without needing to interact with external factors. Many more complex database management systems are not designed for this workload and would typically be hosted via cloud services. Because of this, the entire platform can run without the need for an internet connection.

Nonetheless, the database schema comprises of one table:

workout_data

time	force_reading	workout_id	finger
INT_32	INT_32	INT_32	INT_32

Fig 15. Database Schema

The table 'workout_data' stores force readings received after a workout ends. For each force reading *fr* received, a new row is created on this table. The time column is occupied by a time-delta (where one unit increase represents a change over 100 milliseconds) from the beginning of the workout. The force_reading column is occupied by the value *fr*. The workout_id column is occupied by a unique, non-null integer generated at the start of each new workout. Lastly, the finger column is a value distinct value within the set {0,1,2,3}, where 0 represents the A2 Pulley section of the middle finger, 1 represents the A2 Pulley section of the ring finger, 2 represents the A4 Pulley section of the middle finger, and 3 represents the A4 Pulley section of the ring finger. When a user aims to retrieve data from the database via graph visualizations, this workout_id column is what will be queried to differentiate between workouts, while the finger column differentiates the individual reading at said finger section within that workout.

I. Wearable Device Fabrication

The base of the wearable device is an off the shelf Black Diamond crack climbing glove. This particular glove was chosen for three major reasons: durability, comfort, and usability. Since the Black Diamond glove was originally designed for crack climbing, it fulfills many of the design requirements we previously set. The fingers are minimally covered, preserving the climber's grip and the glove's construction is durable, but comfortably soft, made of a combination of rubber and suede leather. A velcro strap around the wrist serves to tighten the glove and ensure that it does not move while in use. Additionally, the glove does not cover the climber's palm while simultaneously providing ample space on the glove mounting electronics.

The electronic components of the wearable device were attached and secured to the base (glove) in several ways. First, the haptic sensors were secured to the back of the glove such that their location on the back of the glove remained static during use, as well as corresponded to their respective pulley sensors. Next, the piezoresistive sensors, which are connected to the ESP32 via insulated wires, were similarly secured to the back of the glove (albeit looser to account for flexing of the fingers). Finally, the microcontroller, amplifier PCB, and battery are attached to the user via a housing (elaborated upon further in the next section) attached to the user's wrist via an adjustable velcro strap.

J. Encapsulation

To protect the user from accidental electrocution or harm due to system malfunction, a capsule composed of PLA plastic was designed using the Fusion CAD software. This capsule acts as a bed for the ESP32 to sit snug-fit, and a pocket to store the battery away from the user, since it is placed under the PCB/ESP. Additionally, electrical tape is wrapped along the top of the ESP32 and bottom of the PCB to protect the device from foreign substances, or static electricity from the user.

Fig 16. Device Capsule

K. *Overall Production*

The CLIMB glove acts mainly as the data-collection layer of the design by procuring force readings over an interval of time and shipping them to the processing layer. The secondary operation of this component is to drive a haptic alarm upon the encroachment of dangerous force-reading levels on a particular pulley. The mobile application describes the obtained force readings analytically. Users can set pulley force thresholds, begin a workout, and analyze previous workout sessions. To track data, a lightweight, reliable database management system, SQLite, is used to store and query user data.

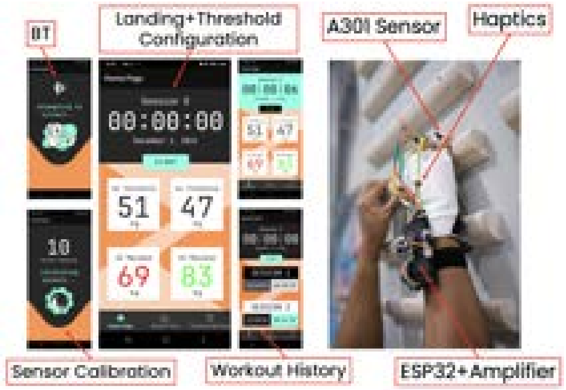


Fig. 17. Final product

L. *Dual Wielding*

To support dual-wielded CLIMB, the only change that is necessary occurs on the mobile-device-end. This is because the hardware system is naturally modular, therefore creating a second identical hardware device (under a different Bluetooth tagname) is enough in terms of hardware, which we accomplished.

The mobile application's Bluetooth interfacing software would have to be modified to open two dual input/output buffers with the left and right CLIMB hardware systems (as opposed to a single dual input/output buffer). Then, on the app frontend, all functionality, charts, and visualizations would have to be doubled, differentiating between the left and right hardware system. Then, when a user attempts to begin a workout, a start signal is sent to *both* hardware systems, where each device would calibrate independently. When a user attempts to stop a workout, each respective input Bluetooth buffer allocated for the two devices would be read, and independently parsed into the database. The database would include a new column, *glove*, where each data point could be queried more specifically to match a specific left or right hardware system.

Ultimately, a second identical hardware device was assembled. However, the full functionality required on the mobile-application could not be finished in time. Duplicating front-end elements while maintaining high user-experience standards would have required creating new mockups and pages, ultimately repeating large and time-consuming portions of the design process. Not to mention, given the asynchronous nature of data collection used in the CLIMB system, where input/output buffers on the mobile application are opened and closed dynamically when communication is required/ not required, balancing this process over two buffer groups showed many bugs. One common issue experienced when two ESP32 systems were introduced was refusal to reconnect on the ESP32 side, where every session after the first would no longer work as intended.

VII. TEST, VERIFICATION AND VALIDATION

A. *Ergonomic Validation*

To validate the system's ergonomic requirements, we surveyed 20 randomly selected climbers. Each participant provided the shape characteristics of their fingers. With this information, we tested the system sensor size requirements to verify that it only imposed up to 30% surface area coverage for at least 90% of participants' fingers.

We met the system's ergonomic use-case requirements by providing a comfortable and unobtrusive wearable device that covered no more than 25% of a user's finger on average.

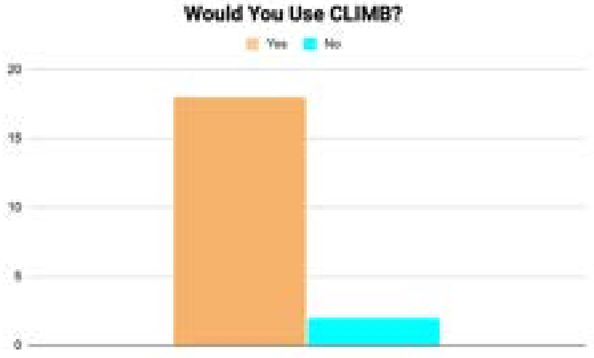


Fig 18. CLIMB Usability Survey

We asked survey participants if they would consider using our product. 18/20 participants said they would. One climber at the Iron City Bouldering Gym said "Yeah I'd use that as long as my fingertips and palm are free, it'd definitely save people from getting pulleys." We used this feedback to iterate on our sensor placement design.

Fig 19. CLIMB Training Survey

We asked another 20 survey participants if they would consider using our product on either static or dynamic climbs. All 20 participants said they would use our product during static training, while only 4 of them said they would use it during dynamic climbs. Another climber at the Iron City Bouldering Gym said "You're using climbing tape? As long it has the same feeling as climbing tape, it shouldn't feel any different than a taped up finger." We used this feedback to iterate on our sensor fabric choice.

B. Durability Verification and Validation

To verify the system's durability requirements, the device underwent 20 uses, each for 5 minutes with a constant 10kg applied at a 90 degree angle. This was repeated for 84kg as well. After each, the sensitivity loss of the device was verified to be within 2.5%, being approximately 2%, and the wearable fabric was examined to have no tears or disfigurations.

To validate durability, the device underwent dynamic weight applied by a user at various angles for 1-hour, 10 times. After each, the sensitivity loss of the device was validated to be within 3%, being approximately 2.5%, and the wearable fabric was examined for any tears or disfigurations, having none.

This meets the system's durability use-case requirements by tackling even the most extreme cases (84 kg/sensor) and verifying that the system remains robust.

C. Safety Verification

To verify that the system initializes properly, the system was booted up and calibrated 100 times and verified to succeed at least 95% of the time. The system successfully calibrated 100% of the time.

To verify sensor precision, incremental weights were placed atop the sensor (1kg, 2kg, 3kg, etc.) and a voltmeter was used to monitor the voltage changes per gram. The system reflected a 0.1mV/g voltage change up to 30 kg.

To verify that an alarm signal is triggered within 100ms when the force reading is within 30% of the safety threshold, an instant force exceeding the threshold was applied after initialization, and an external (software) timer recorded the delay. The system fired its alarm within 100ms 100% of the time.

To verify the second haptic's functionality, the first haptic was disconnected. Immediate force was applied until the threshold was reached. The system fired its second alarm within 200ms of threshold excession with a 100% success rate.

This met the system's safety use-case requirements by demonstrating that the system actively worked to prevent pulley injuries by remaining with a maximum error of ±2.5 and firing alarms within 200ms.

D. Economic Verification

The economic use-case requirements are directly dependent on the safety use-case requirements. By preventing injuries, worst-case healthcare costs are mitigated. In the previous section, injury prevention is verified.

E. Social / Feedback Verification

To verify the feedback requirements of the system, tests were performed using mocked force readings to verify the minimum data transfer. To do this, a mocked buffer input of 48 KB was generated and communicated to the mobile device. This test was timed, recorded, and repeated 50 times to verify that the shipment completed in 10s with a 95% success rate. The shipment took approximately 5 seconds.

To verify the user-experience related requirements, 48KB of data was mocked on the mobile device and passed to the database handler to ensure that information is stored properly and can be pulled to display on the application charts. This test was repeated 50 times to verify that the flow of data on the mobile device can reach the user's screen with a 95% success rate. The user data reaches the user interface within 0.05s of data being shipped 100% of the time.

These tests meet the use-case requirements by verifying that the system is providing users with a seamless experience in accessing analytical feedback that is suggestive towards improvement and can be used to compare progress with others.

F. End-to-end Verification and Validation

To verify end-to-end integration, the system underwent 50 dynamic climb tests where readings workouts were started, data collected, shipped, and formatted on the app to determine if the data is successfully displayed analytically 95% of the time. The data successfully displayed 98% of the time, the failures occurring due to accidentally battery disconnection.

VIII. PROJECT MANAGEMENT

A. Schedule

The project schedule breaks tasks down into five main categories: Project Deadlines, Wearable Device Design, Embedded Implementation, Mobile App Design and Implementation, and Integration and Unit Testing. Each task has a team member who is primarily responsible for its completion, although there is overlap and collaboration on every task is expected. The GANTT chart used to visualize the schedule can be found at the end of the document as Table I. Several additions were made to the schedule throughout the design process, with the main changes being made towards the end of the schedule. Integration was a significant challenge that required all three team members to work both together and interchangeably on various aspects of the project that were previously assigned to single team members, and this is reflected in the GANTT chart.

B. Team Member Responsibilities

Alexander is primarily responsible for developing the Android mobile application, including the front end UX/UI, overall app architecture, and functionalities where the app interacts with the wearable device (programming ESP32 sensor calibration, bluetooth pairing, etc.) and database (data visualization and analysis). He is also responsible for design and construction of the wearable device housing (glove).

Jubahed is primarily responsible for end-to-end Bluetooth connectivity setup, as well as the application backend. This includes processing incoming serial data, keeping relevant information through filtering and processing, and inputting into the SQLite database. He is also responsible for allowing frontend elements to interface with the database safely and securely.

Joshua is primarily responsible for the development of the hardware platform. This includes designing the PCB, programming the MCU to interface peripherals and communicate via Bluetooth, testing and integrating sensors and haptics, and designing circuit schematics.

C. Bill of Materials and Budget

Our final total cost for our project, including testing, demo, and development is $385.84. The full bill of materials and budget can be found at the end of the document as Table IV.

D. Risk Mitigation

At project conception, we identified 4 major risks, sensor inaccuracy/viability, alarm failure, battery/MCU malfunction, and task management.

Our largest risk of sensor failure stemmed from the physical properties of most small-scale force sensors (piezo-resistive sensors). Typically, these sensors are unstable and experience a non-linear change in resistance as force is applied. However, to mitigate sensor failure, we began research on the day of project conception to find a piezo-resistive sensor and an approach we could apply to bridge these gaps. After identifying 3 different sensors, we called the distributors to inquire more details, and placed orders immediately to begin unit testing. In our unit tests, following the manufacturer's recommendations, we used an amplifier to amplify the sensor readings while providing stability via a phase lead compensation capacitor to prevent the response from oscillating, essentially performing a low-pass filter effect with frequency response shaping. Finally, we performed weight tests and tuned amplifier gains while respecting hardware constraints. In doing so, we determined an approach and sensor that could support our product.

The second largest risk was alarm failure. Should the alarm fail to fire due to malfunction, our user could experience severe injury. Thus, we incorporated redundancy in our system to increase the reliability of our system. We introduced a second haptic on the same finger that fired along with the first. Although this did not fully mitigate the risk of both motors failing, however, it reduced the probability of total failure occurring, making this risk negligible.

The third largest risk was battery/MCU malfunction, which could be due to user mishandling like excessive force or static shock of the board. We mitigated this by wrapping the device in electrical tape and placing the battery in a capsule, secluding it from the user, lowering the consequence of this failure, which reduced the risk to a negligible degree.

The fourth largest risk was task management. Early on in the project, we realized that design modularization and skill-based task assignment was critical for achieving project efficiency and collaboration. We broke complex deliverables into smaller, more manageable modules which enabled parallel development, simplified debugging, and improved our project's scalability. Assigning tasks based on our team members' skills and skill-maturity ensured that work would be completed effectively while fostering growth opportunities for each of us.

IX. Ethical Issues

Our project's main goal is to support the health and safety of rock climbers by preventing debilitating pulley injuries before they happen. If our project became widely adopted as a safety or training device, the ideal user would be rock climbers of any size, weight, and experience level. Since our goal is to improve the safety of all climbers, the ability to use CLIMB should not be affected by gender, age, or any other discriminatory factors. This potential effect is mitigated by customizable pulley thresholds and the glove which can be fitted to a wide range of hand sizes via the velcro wrist strap.

Our project does not rely on any outside data, and instead only uses session data collected from the user via sensors. While privacy is often an ethical concern when personal data is concerned, the data collected by CLIMB does not contain personally identifiable information (PII) andt cannot be used to identify, contact, or target individuals in any way. Furthermore, the data collected by CLIMB is stored locally to the user's device, further mitigating privacy concerns.

Additionally, our project's effectiveness is heavily dependent on the correct placement of piezoresistive sensors on the climber's fingers. As a result, new users of the device, independent of their climbing ability and experience, are most vulnerable to failure and misapplication. Users who are unfamiliar with the device are more likely to misplace the resistor bands on their fingers, which could result in inaccurate readings relative to the injury thresholds which have been set. Additionally, new users may not adjust the alarm thresholds properly, which would negatively impact the device's ability to alarm the user before an injury occurred. To mitigate this, the product would likely be shipped with an included starter's manual, ensuring that the user is properly informed as to the proper and safe use of CLIMB.

There are several public health considerations that must be made during the design and realization of our project. Since our device aims to prevent pulley injuries in climbers, the public health considerations of our project are clear. A healthy climber is one who is physically and mentally uninjured, and one way our device aims to achieve this is by keeping the climber's fingers healthy. By alarming users before they potentially rupture their pulley tendons, our device will keep the climber healthy. We also have included multiple haptic sensors in the design of our wearable glove such that the climber can feel the alarm when extreme risk is iminent, and they can get to a safe spot before their health is compromised by an untimely injury.

Our device is designed to ensure the personal safety of climbers, and as such it must not have any negative impact on the safety of the climber or others while in use. While a pulley injure in and of itself is a danger to the safety of the climber, it is also crucial to consider the fact that when a climber ruptures their pulley, they will fall off the climbing wall, which can also be a major safety concern should they fall the wrong way and injure other parts of their body (head, ankle, arms). One design consideration we made to mitigate this potential effect from usage of our device was minimizing the amount of coverage our sensors took up on the fingers. Less surface area on the fingers can cause a climber to lose grip, so it is essential that we design our device to take up as little of that surface area as possible to maximize the safety of the user while wearing our device.

When considering the overall welfare of users, we must also consider the psychological impact of climbing injuries. By having a device which will alarm users when they are approaching dangerous force distributions on their fingers, users who are anxious about climbing can have assurance that they will be physically safe. This could be especially important to both new climbers who are anxious about the seemingly dangerous activity of bouldering as well as experienced climbers who have previously experienced pulley injuries and are thus aware and afraid of reinjury and the pain and rehabilitation that goes along with it. Finally, being scared or stressed on the wall can increase the danger associated with climbing, since smart decision making is key in this type of activity, so our device can help users be worried about one less thing while climbing and thus increase the psychological and physical safety of the climber.

X. Related Work

Currently, there are no products like CLIMB. However, there are products with a similar goal to CLIMB. For instance the NiceClimbs PulleyPal [10], a pulley-injury rehabilitation device that operates purely mechanically by clamping down torn pulley-tendons to the bone. This is very similar to the pulley-splint [11], a common miniature mechanical contraption that compresses the area around a climber's finger to keep the tendon squeezed to the bone. And besides physical products, there are rehabilitation/training programs climbers can purchase if they need guidance when navigating pulley injury rehabilitation. For instance, the Rock Rehab Protocol claims to "take you step-by-step through the entire rehabilitation process." [12] And there are many more products and guides like these out on the market. However, all of them fail to solve the central issue: pulley-injury prevention.

XI. Summary

CLIMB aims to greatly improve safety in the sport of rock climbing by equipping climbers with a wearable device that can monitor the weight distribution across their fingers and alert them via a haptic motor when the risk of pulley injuries is dangerously high. There currently are no real-time injury prevention devices available on the market for climbers to use. Our product provides a solution to this problem, not only by providing near-instantaneous injury prevention while climbing via a wearable device, but through data analytics and visualization on the CLIMB mobile app.

Pulley injuries are common amongst climbers, and the pain of the injury and long recovery process are dreaded. With CLIMB, pulley injuries will no longer be a source of concern, and climbers will climb confidently, knowing that they will not be hurt by the sport they enjoy.

Through validation and verification testing, we determined that the current iteration of CLIMB successfully fulfills our original design specifications.

A. *Future Work*

There are several features that could improve the usability and performance of CLIMB in future iterations of the device. Firstly, developing an environment specification feature for the app could help the device account for and adapt to changes in rock surface, specifically adjusting for differences we encountered during the testing of force readings for sharper vs smoother surfaces. Another helpful feature could be the addition of positional sensors along the joints of the finger to account for the increased risk caused by certain finger poses.

As mentioned earlier in Section VI.L, finishing the app-side component of dual wielding would also be a main goal for future iterations of CLIMB. Along with these app-side changes, including instructions for successful use of the wearable device within the app would be greatly beneficial for users. Finally, we also believe that a threshold suggesting feature would be greatly appreciated by new users who are unfamiliar with their pulley strength limitations.

B. *Lessons Learned*

We learned that design modularization and task assignment via skill-maturity is vital in achieving parallelization, managing progress effectively, and improving collaboration. Another key takeaway we had regarding our design process was that users are unpredictable. As a result, identifying risk and implementing safe states and redundancy in our system to avoid catastrophe is critical to creating a successful product.

GLOSSARY OF ACRONYMS

ADC – Analog to Digital Converter
ACK – Acknowledge
ESP-WROOM-32 – Espressif Systems Microcontroller Board
GPIO – General Purpose Input-Output
LDO – Linear Drop-Out Regulator
NACK – Not Acknowledged
MCU – Microcontroller Unit
PCB – Printed Circuit Board
PLA – Polylactic acid
RPi – Raspberry Pi

REFERENCES

[1] Josephsen G, Shinneman S, Tamayo-Sarver J, et al. “Injuries in Bouldering: A Prospective Study.” *Wilderness & Environmental Medicine.* 2007;18(4):271-280. doi:10.1580/06-WEME-OR-071R1.1

[2] Zafonte B, Rendulic D, Szabo RM. “Flexor pulley system: anatomy, injury, and management.” *J Hand Surg Am*. 2014 Dec;39(12):2525-32; quiz 2533. doi: 10.1016/j.jhsa.2014.06.005

[3] J. M. Blackstone, “Children and gender-differences in exposure and how anthropometric differences can be incorporated into the design of computer input devices,” *Scandinavian Journal of Work*, Environment & Health, 26 (2007).

[4] MDSave, Hand/Finger Tendon Repair. 2024 https://www.mdsave.com/procedures/hand-finger-tendon-repair/d78bfecc

[5] Mouser Electronics. (n.d.). MF01A__A01 Datasheet [PDF]. Retrieved from

[6] https://www.mouser.com/datasheet/2/13/MF01A__c3_a2_c2_96_c2_a1_A01-1915118.pdf

[7] Interlink Electronics. (2010). FSR 402 Datasheet [PDF]. Retrieved from https://cdn.sparkfun.com/assets/8/a/1/2/0/2010-10-26-DataSheet-FSR402-Layout2.pdf

[8] Tekscan, Inc. (n.d.). A301 sensor datasheet [PDF]. Retrieved from https://www.tekscan.com/sites/default/files/FLX-Datasheet-A301-RevI.pdf

[9] Amphenol ICC. (n.d.). 1020-15-003-001 sensor specification sheet [PDF]. Retrieved from https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2234/1020-15-003-001_Spec.pdf

[10] Espressif Systems. (n.d.). *ESP32-WROOM-32 datasheet* [PDF]. Retrieved from https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

[11] NiceClimbs. (n.d.). NiceClimbs Pulley Pal. NiceClimbs. Retrieved [10/12/2024], from https://niceclimbs.com/product/niceclimbs-pulley-pal/

[12] The Climbing Doctor. (n.d.). Pulley protection. The Climbing Doctor. Retrieved [10/12/2024], from https://theclimbingdoctor.com/pulleyprotection/

[13] The Climbing Doctor. (n.d.). Rock rehab protocol: Pulley sprain. The Climbing Doctor. Retrieved [10/12/2024], from https://theclimbingdoctor.com/product/rock-rehab-protocol-pulley-sprain-3/

[14] Microchip Technology Inc. MCP6001/1R/1U/2/4 1 MHz Low-Power Op Amp Datasheet (DS20001733L). https://ww1.microchip.com/downloads/en/DeviceDoc/MCP6001-1R-1U-2-4-1-MHz-Low-Power-Op-Amp-DS20001733L.pdf [Retrieved 10/12/2024]

[15] Jay, P. (2019) MPAndroidChart (Version 3.1.0) https://github.com/PhilJay/MPAndroidChart/

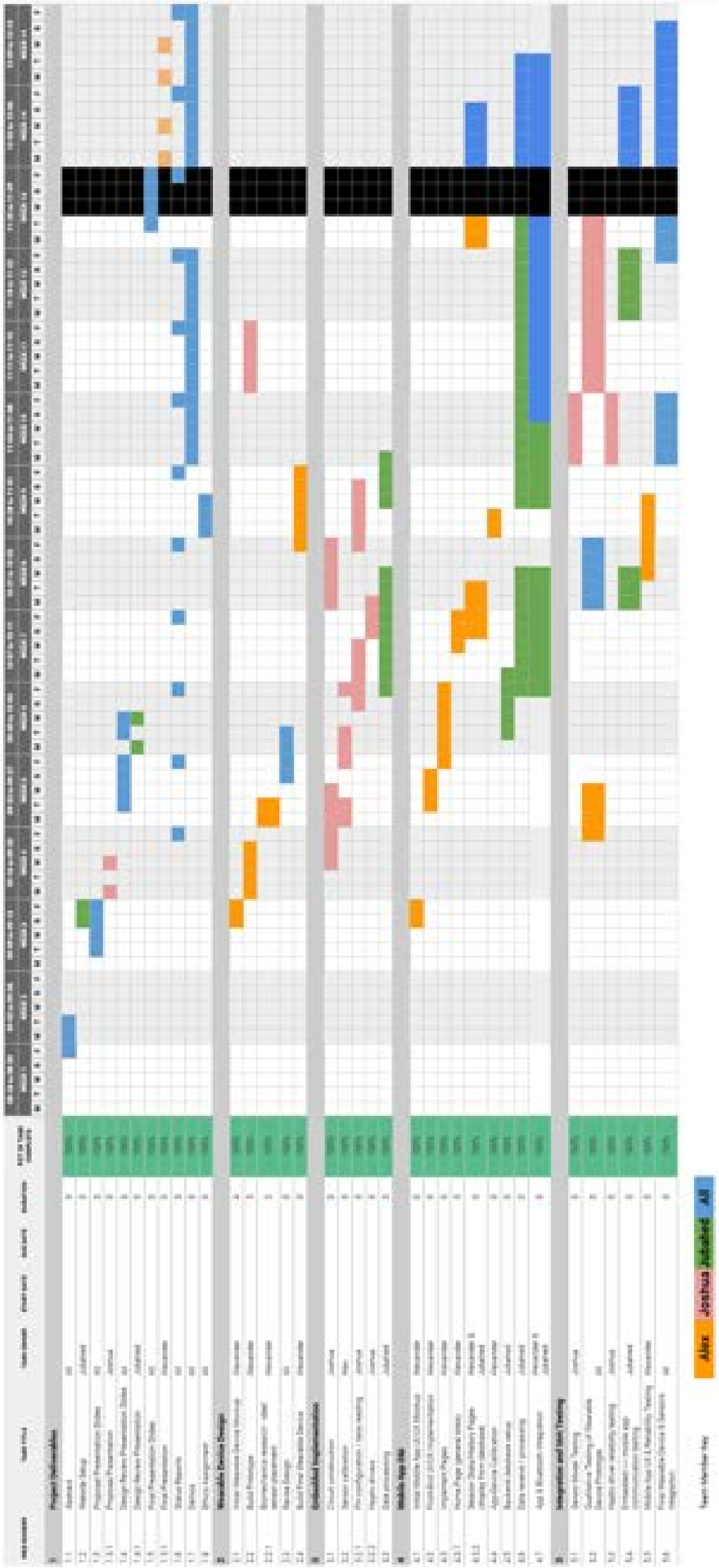


Table III. GANTT Chart (Project Schedule)

Model	Used?	Introduced after the design report?	Bill of Parts and Budget			
			Quantity	**Cost (total)**	**Manufacturer**	**Description**
FlexiForce A301 Sensor	Yes	No	10	$108.93	Tekscan	Piezoresistor force sensor
DC Motor Vibration, ERM 3VDC [316040001]	Yes	No	10	$4.80	Seeed Technology Co.	Haptic motor (used for alarm)
OLIMEX ESP32 DEVKIT-LIPO	Yes	Yes (we changed the MCU DevBoard)	3	$45.13	Olimex	Microcontroller
MCP6004-I/ST	Yes	No, but we increased quantity	6	$3.24	Microchip Technology	Operational Amplifier
PCB Fabrication	Yes	Yes	1	$51.80	OSHPARK	Printed Circuit Board
3.7 V LiPO Battery	Yes	Yes	4	$31.31	Qimoo	Power source
Crack Climbing Gloves	Yes	No	1	$49.95	BLACK DIAMOND	Crack climbing gloves hardware is mounted on
RN73H1JTTD8353F25 Resistor	Yes	Yes	16	$2.50	KOA Speer Electronics Inc.	835 kOhms SMD Chip Resistor
Pull Up Tower	Yes: for testing and demo	Yes	1	$79.19	SogesPower	Pull Up Tower for end-to-end testing/demo
Wooden Hand Model	No	Yes	1	$8.99	Juvale	Originally intended for display and testing
Total			22	$385.84		

Table IV. Bill of Parts and Budget