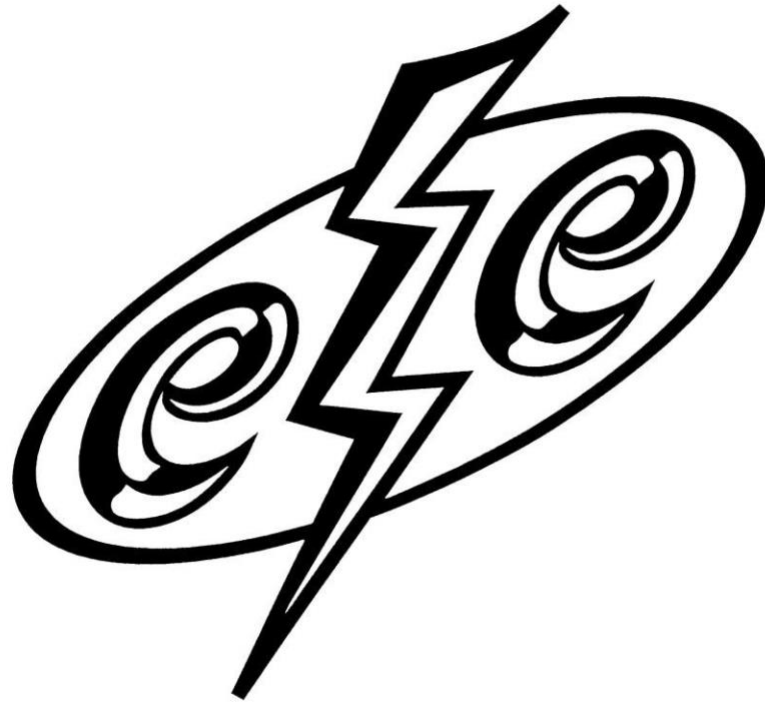


**ELECTRICAL ENGINEERING DEPARTMENT**  
**California Polytechnic State University San**  
**Luis Obispo, California**



**EE 462**  
**VR Force Feedback Gloves**  
**SENIOR PROJECT REPORT**  
**Mark Wu 2023**  
**Claire Chen 2023**

## TABLE OF CONTENTS

<b>Acknowledgements .....</b>	<b>3</b>
<b>Abstract .....</b>	<b>4</b>
I. Introduction .....	5
II. Background.....	7
III. Requirements.....	10
IV. Design.....	12
V. Test Plans.....	16
VI. Development and Construction.....	18
VII. Integration and Test Results.....	27
VIII. Conclusion.....	32
IX. Bibliography.....	33
<b>I. Appendices.....</b>	<b>35</b>
Appendix A. Senior Project Analysis.....	36
Appendix B. Specifications and requirements.....	39
Appendix C. Parts List and Costs .....	40
Appendix D. Schedule - Time Estimates.....	42
Appendix F. PCB Layout .....	45
Appendix H. Software (Unity) .....	51

## **Acknowledgements**

We would like to dedicate this acknowledgement to Dr. William Ahlgren, who has the most genuine desire for all students to learn. Dr. Ahlgren has aligned goals with the student and spends no effort to impede their growth.

## **Abstract**

The goal of this project is to produce a manufacturing plan for a consumer VR glove. The total addressable market of VR is over 170 million global users as of 2022 (Kolmar , 2022) with a serviceable available market of 300,000 users on Meta’s own social platform (Heath, 2022). The targeted Quest 2 platform utilizes handheld controllers, which causes a lack of immersivity in social settings and gaming scenarios. One common use of the platform involves social platforms such as “Horizons” where users meet in a virtual world to interact; handheld controllers don’t allow users to shake hands, make finger gestures, or touch each other’s avatars. Furthermore, users enjoy a breadth of different immersive games from first person shooters to rhythm block slicing games; in this setting, users often interact with physical objects and cannot sense their shape or their mass, nor can they finely interact with simple virtual mechanisms like weapon triggers. At the current time of writing, VR gloves that are readily available for purchase start at \$5,000 and can go up to \$10,000. However, the previously mentioned 300,000 users of Horizons fall into the consumer market, where accessibility/cost is more constraining; the hardware they use, the Meta Quest 2, costs \$400 (Meta, 2022). Thus, these consumer users are unserved by the current market options. This project aims to provide increased immersivity in social and gaming scenarios at a price point that Quest 2 users can afford. Without a thorough customer analysis other than overall sentiment for mass adoption of VR glove technology, this project aims to produce the necessary manufacturing plan and demonstration to appropriately crowd-fund further efforts in offering a product. Deliverables of this project include two prototype demonstrations of the electrical system, mechanical system, and firmware as well as a manufacturing and production plan.

## I. Introduction

Based on market sentiment, there is a projected growth of sales in the VR gloves market (Expert Market Research, 2022); although the exact projections are behind a paywall, positive market sentiment from the YouTube community (Linus Tech Tips, 2022) is enough to justify a low-stakes prototype. The LucidGloves project (LucidVR, 2021) is an open-source, low-cost project that gained traction on YouTube and has gained sentiment from the community for the commercialization of the project; however, the project remained open source and provides guides for users to assemble them at home for the cost of \$60 (Linus Tech Tips, 2022). VR users are demanding commercialization of the LucidGloves project so that the glove design can be available for purchase.

The two primary methods of providing information to the user are force feedback and haptics; as the name implies, the former produces a force on the user's fingers while the latter produces vibrational-like sensations. Force feedback communicates absolute information, giving rise to information such as shape and mass. On the other hand, haptics communicates relative information, lending itself nicely to texture sensations. While haptics is simpler mechanically, it requires more in-depth knowledge and much more technical development than force feedback, which was proven by the open-source LucidGloves project. Thus, this project proposes a VR Glove design that is low-cost and manufacturable, while building off the lessons-learned from the LucidGloves project as well as sourcing as many COTS components as possible. With a plan for manufacturing, the community can vote with their own investment into the project in exchange for a pair of functionally similar gloves rather than building the LucidGloves project themselves. The project will focus on instilling confidence into our crowd investors in the capabilities of the design and the feasibility of the manufacturing run.

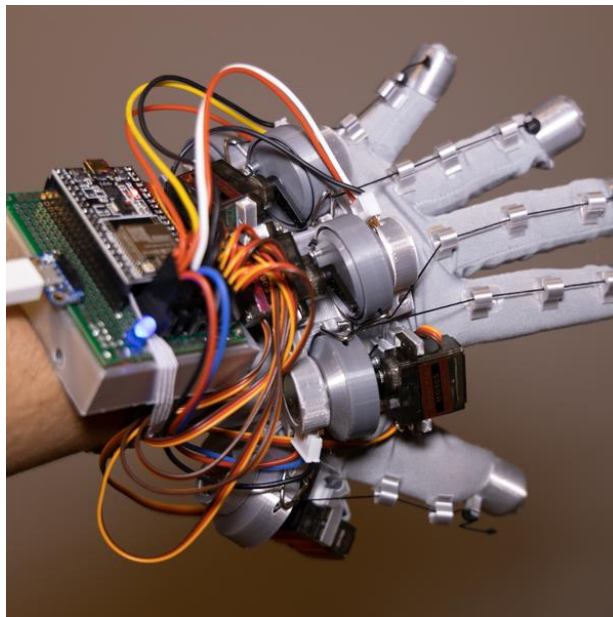
Consumers who are on the Meta Quest 2 platform should be able to afford this accessory device. Because no other attainable products exist, our consumers want to experience the novelty of VR gloves at the cost of lacking features the bleeding edge technology offers. The core value proposition that these VR gloves offer shall improve immersivity in their social and gaming experiences through force feedback at an affordable price; this comes at the cost of other requirements such as battery life, size, and comfort taking a backseat. Ultimately, this skew of benefits towards low-cost while allowing for a limited feature set will attract the mass consumer targeted customer.

The direct competitor to this product is the new and upcoming "TactGlove" by BHaptics. The TactGlove is only compatible with the Meta Quest 2 and will be available for purchase at \$299 in Q4 2022; this is a real-world baseline for the pricing that this project needs to compete with. VR reviewers have tested the TactGlove and gave it general praise; however, one reviewer mentioned that haptics-only feedback falls short when conveying sensations such as mass and density, which is a core part of the gameplay in games such as "Hand Physics Lab" (Nathie 6:48, 2022). This project aims to compete directly by providing consumers with an alternative feedback technology which suits gameplay that involves more shape sensing than texture

sensing. Of course, the LucidGloves that inspired this project are undergoing development and have much more traction than this project. As of writing, there is no information about the commercialization progress of the LucidGloves design.

## II. Background

The history of consumer haptic VR gloves began in 2021 with Lucas from LucidVR developing the LucidGloves, initially a \$22 BoM prototype that leveraged hobbyist COTS components. There was large sentiment for the project to be commercialized such that consumers can purchase the gloves. The response from LucidGloves is as follows: "Right now Lucas is a student and doesn't yet have the resources to mass produce and sell gloves/kits, though this may change in the future. In the meantime, anyone can have a glove by building it, since the project is open source" (Nyan man | Mod, 2021). The fourth design of the LucidVR gloves uses hobbyist focused COTS components such as development boards, servo motors and salvaged springs from badge reels; this lack of integration and disassembly makes it a laborious design to manufacture. The fifth design is currently underway by Lucas as of this writing (November 2022) and appears to reduce many of the barriers that make assembly difficult. Overall, the LucidVR gloves has pathed the way for the future of consumer VR gloves by demonstrating market demand and developing a foundation for the electrical, mechanical, and software subsystems for a force feedback glove. As stated by Lucas, the project is open-source and all the CAD, firmware, and electrical diagrams are available on GitHub.



*Figure 1. LucidGloves Prototype 3 (LucidVR)*

At around the same time in 2021, a company called BHaptics started developing their full body haptic suit called the Tactsuit X40. BHaptics has recently announced the TactGlove, a \$300 pair of VR haptic gloves to be sold with their haptic suit and has not released it on the market at the time of this writing (October 2022). These gloves utilize acoustic haptic motors instead of force feedback which the LucidGloves use. While the TactGloves provide increased immersivity in “feeling” situations where the user interacts with surface textures, they lack the ability to

increase the immersivity in situations where the user interacts with the shapes of their environment. However, they are the only VR gloves targeting the consumer market right now.

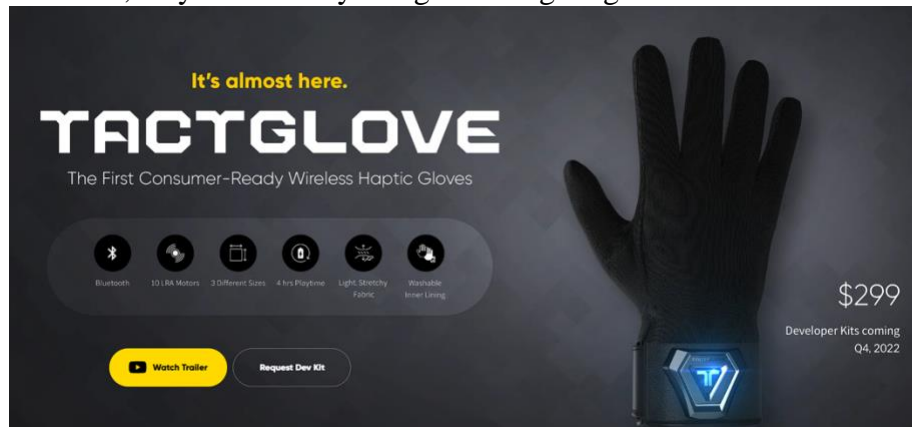


Figure 2. Tactglove by BHaptics coming in Q4 2022

The history of VR gloves in general spans much longer, dating back to the 1980s where the first glove was patented; these were dubbed the “Sayre” gloves in 1982 by Daniel Sandin and Thomas DeFanti (Gupta et al., 2020). Currently, in 2022, there are a lot more companies working on VR gloves for enterprise and high-end consumers. The demand for the technology spawned recently so the technology is in its infancy. Some notable companies and their technologies include HaptX, (from Cal Poly) who utilize micro-fluidics for pressure-induced haptics, Manus who utilizes haptics, SenseGlove who utilize force feedback magnetic brakes and vibrational haptics, DextaRobotics who utilize force feedback mechanical exoskeletons, and Hi5 who use vibrational haptics. The images of these companies’ gloves are shown below. These companies’ technologies demonstrate potential candidates for what standard VR glove design may look like in future. Notably, the SenseGlove utilizes both force feedback and haptics mechanisms which would paints a path for a feature complete version of this project.





Figure 3. HaptX Gloves



Figure 4. Manus Gloves



Figure 5. SenseGlove Gloves



Figure 6. DextaRobotics Gloves



Figure 7. Hi5 Glove

### III. Requirements

There are 3 primary factors that consumers care about at a high level: cost, immersivity, and user friendliness. Given that options exist at the enterprise level which target immersivity and user friendliness, the factor that consumers care most about is cost. Given that the Meta Quest 2 headset costs \$400, the price range of an accessory should be some fractional amount of that price. As for immersivity, there are the two categories of “touching” and “holding” as stated by a reviewer who tried the BHaptics Tactgloves (Nathie, 2022). “Touching” is achieved using AC signals; this is where a lot of advanced haptics development gets categorized. “Holding” is achieved using DC biased signals; this is where force feedback mechanisms are required; these mechanisms need to produce a force, and the smallest step of force the mechanism can produce determines the resolution of information that can be conveyed to the user. Higher resolutions are correlated with more experiences that can be replicated, especially fine interactions with complex virtual objects. Finally, user friendliness separates a DIY VR glove from a professional glove. The LucidGloves have very little user-friendly features, completely sacrificing low-bulk, hygiene, and a low equip time, whereas the consumer option from BHaptics offers all these features as well as a long battery life. To be useful to consumers, these VR gloves will have to fall somewhere in between these two products, offering the features that matter while providing a pleasant experience. The Needs Hierarchy was produced using the following Pairwise Comparison.

	Low-Cost	Haptic Actuators	Force Production on Fingers	Fine Resolution Linear Actuator	Low Bulk	Hygienic	Long Battery Life	Low Equip Time	Mean (Geometric)	Weight
Low-Cost	1	9	1	7	3	9	3	3	3.394	0.198
Haptic Actuators	1/9	1	1/7	1	1/3	3	1/3	1/5	0.536	0.031
Force Production on Fingers	1	7	1	3	9	9	7	7	4.177	0.244
Fine Resolution Linear Actuator	1/7	1	1/3	1	5	7	5	3	1.371	0.080
Low Bulk	1/3	3	1/9	1/5	1	7	1	1/3	0.579	0.034
Hygienic	1/9	1/3	1/9	1/7	1/7	1	1/9	1/9	0.177	0.010
Long Battery Life	1/3	3	1/7	1/5	1	9	1	1	0.857	0.050
Low Equip Time	1/3	5	1/7	1/3	3	9	1	1	1.072	0.063

Figure 8. Pairwise Comparison

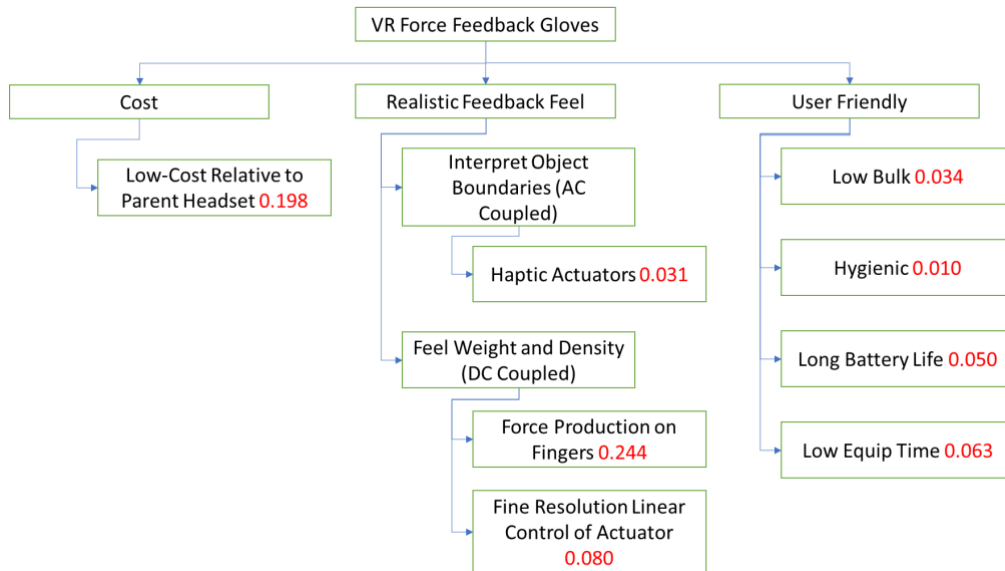


Figure 9. Needs Hierarchy

The following Engineering Specification Table was derived from the student’s experience with the design and build of various projects. As derived from the Needs Hierarchy, critical success criteria are highlighted in orange, preferred features are highlighted in yellow, and non-critical features are highlighted in green. Incompletion of the critical success criteria lead to complete project failure, whereas incompletion of the preferred features only leads to partial project failure. The non-critical features shall only be implemented if they do not impede the completion of the primary project requirements.

Marketing Requirements	Engineering Specification	Justification
Low-Cost Relative to Parent Headset	The total BoM Cost of one unit shall cost less than \$50	By the 1/3 Rule of Thumb, the COGS cost should be 1/3 of the target sale price. \$150 for the target sale price is less than half of the \$400 headset that supports it
Force Production on Fingers	The device's actuator/driver system shall be able to apply a DC bias	This is the core mechanism that communicates absolute location to the user, which is a prerequisite to sensations such as density, mass, and weight
Haptic Actuators	The device's actuator/driver system shall have enough bandwidth for a low frequency 10-50Hz AC signal	This reduces the need to integrate additional vibrational haptic motors which typically oscillate at 10-50Hz
Fine Resolution Linear Control of Actuator	The device actuator/driver system shall take a linear input control signal and output a > 200 step DC bias with +- 2 step tolerance	steps per finger should be enough to produce a wide variety of hand configurations to convey object shapes and is easily realizable by an 8 bit driver. The 2 step tolerance is hard spec to meet due to tolerances of the
Low Bulk	The device shall not weigh more than 200g	The oculus controllers that come with the headset weigh roughly 150g. Given that the hardware will be secured tightly to the user, a slightly higher mass can be permitted.
Hygienic	The device shall be IP68 waterproof	Other gloves support machine washability. IP68 is the rating required to survive for that duration in the washing machine.
Long Battery Life	The device shall have a battery capacity of 3000mAh	Assuming 1A average power consumption (from personal experience on a prosthetic glove project), the gloves will last 3 hours; a reasonable duration of playtime.
Low Equip Time	The device shall limit actions to equip to only 2 motions	Having more than 2 straps to tighten per hand becomes tedious for the user.

Figure 10. Engineering Requirements

#### IV. Design

The following diagram illustrates the L0 system inputs and outputs for a typical VR Feedback Glove system targeting the Meta Quest 2. Stored energy is replenished with each use through a charging interface, information is wireless transferred through the Bluetooth protocol, and optical imaging is used to provide hand/finger tracking information to the virtual environment. The output of the system involves moving the user's fingers, which then in turn change the physical location of the user's hands (which is then fed back into the optical hand tracking input stream).

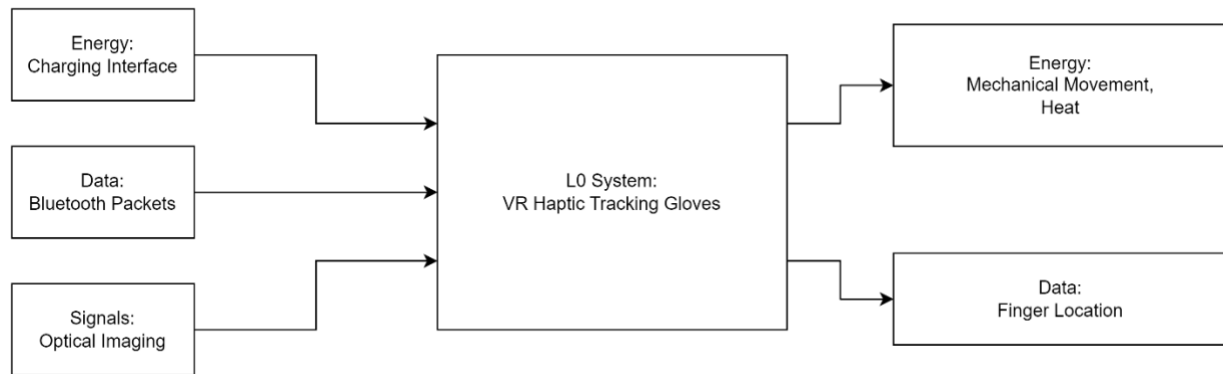


Figure 11. Energy flow diagram of VR Gloves system

The realization of the L0 requirements draws inspiration from both the LucidGloves and the TactGloves. This project takes advantage of the existing optical hand tracking capabilities of the Meta Quest 2 to decrease the scope and complexity of the design which is the method deployed on the Tactgloves, a decision they made to hit the \$300 consumer purchase price. This approach is clever because it reduces the work required from the project hardware down to the other two inputs and outputs: energy and data. In the following L1 decomposition, the complicated hand tracking features have been released on the Meta Quest 2 (and are in active development). The electrical scope of the project is reduced to data to signal processing, energy storage, and energy amplification.

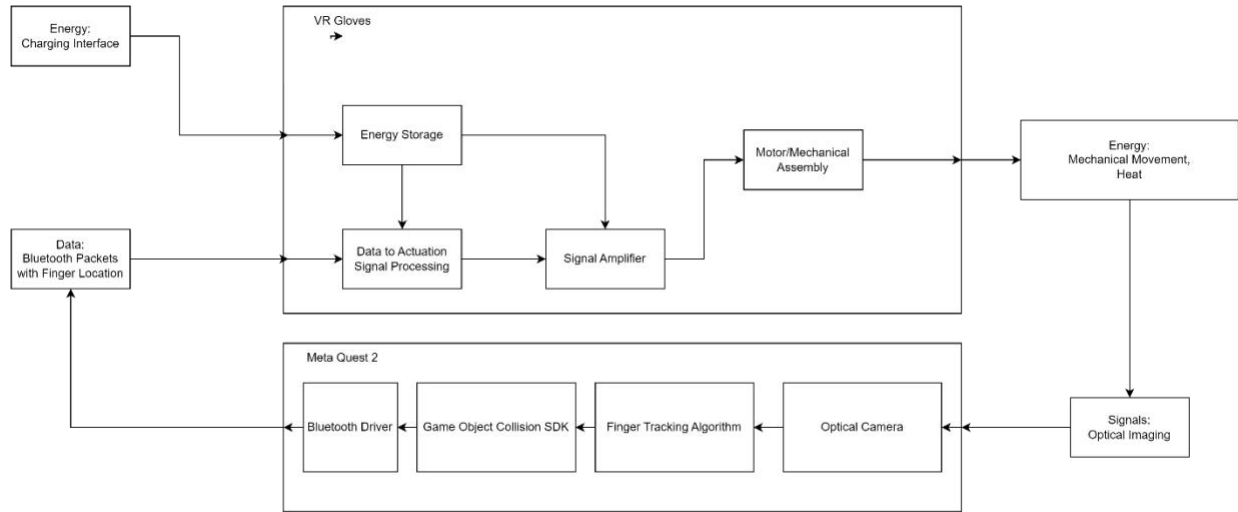


Figure 12. L1 and L2 breakdown of the VR Glove system

The following L2 block diagrams demonstrate a possible implementation route for each of the L1 blocks. The inputs and outputs are well defined and follow common implementation techniques in similar fields of electronics.

A common energy storage device that is used for its safety/energy-density specifications is the Li-Ion battery. These come in standard packages such as the 18650 cylindrical package. The input of the energy storage system shall be a regulated charging signal that is Constant Current limited to  $0.1 \times$  its total capacity and a maximum Constant Voltage limit of 4.2V. The output of the system shall be the unregulated battery voltage which can range from 3.7V to 4.2V. The battery itself shall be able to withstand 2A of continuous current. This subsystem shall also utilize an LDO Constant Voltage regulator to output 3.3V at 1A maximum continuous current for the low power digital circuitry.

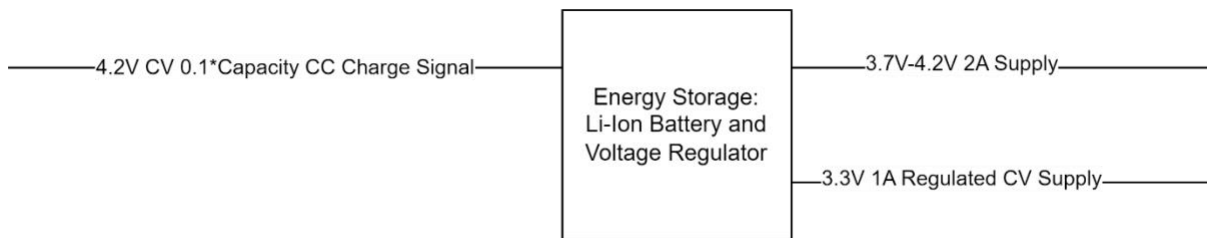


Figure 13. I/O diagram of the energy storage system

The Data to Signal Processing circuitry can easily be implemented using an MCU with an integrated low power Bluetooth modem. Many other digital electronics require digital to signal processing so these integrated devices are common and a suitable candidate should be easy to find. The power inputs required will be a 3.3V CV supply with 1A continuous current sourcing capability. The Bluetooth modem shall be compatible with the Bluetooth 5 standard that the

Meta Quest 2 complies with. The output of the MCU will be a low power, multi-channel digital stepper signal from 0 to 3.3V.

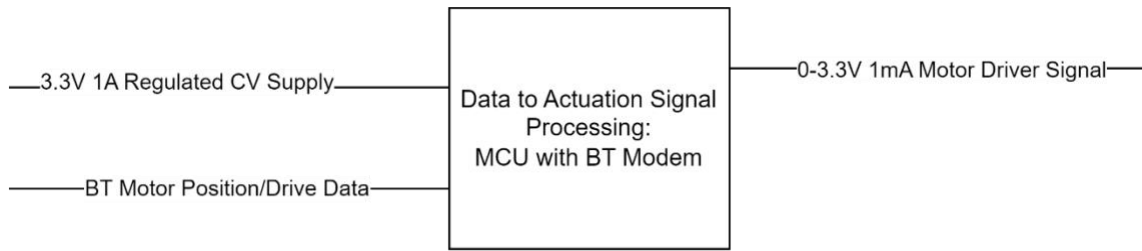


Figure 14. I/O diagram of the data to actuation system

This stepper motor driver signal is then the input to the signal amplifier, which needs to have a high input impedance and a very low output impedance. Specifically, there should be a low impedance from this subsystem’s input battery voltage supply to its output high power motor drive signal.

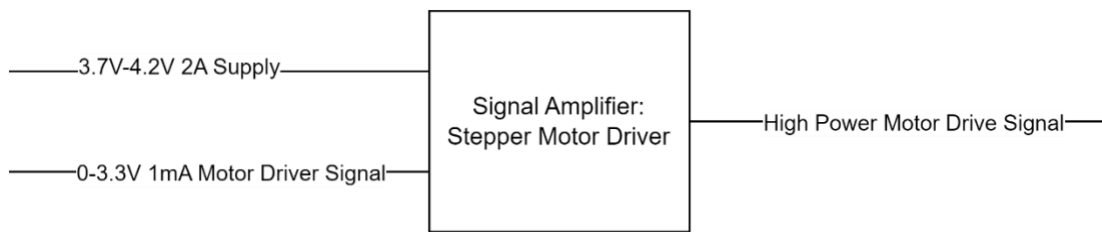


Figure 15. I/O diagram of the signal amplifier system

Finally, the high power/low impedance motor driver signal is converted from electrical energy into mechanical energy through an electromechanical actuator such as a stepper motor. The mechanical energy is coupled through an injection molded chassis which then restrains the user’s fingers to some absolute position.

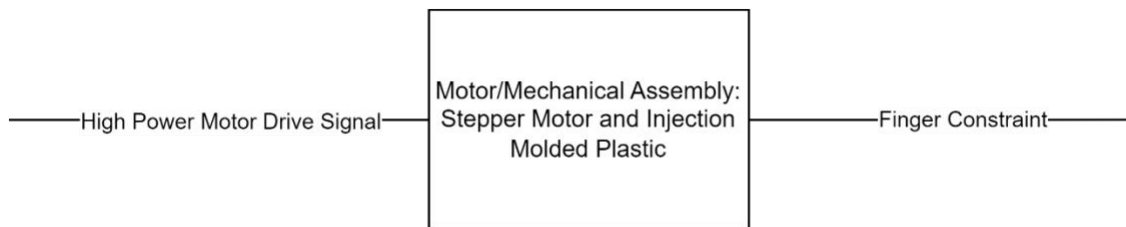


Figure 16. I/O diagram of the mechanical assembly system

An FMEA was created for potential failure modes related to the L2 implementation described above. Two high RPN items include poor glove fitment and motor failure. Poor glove fitment equates to a lower serviceable market. Although this is not too difficult to overcome, it is hard to verify for an initial release unless a large sample group is obtained to try the various glove sizes. It was noted that this is a problem for all glove products and that the project should investigate

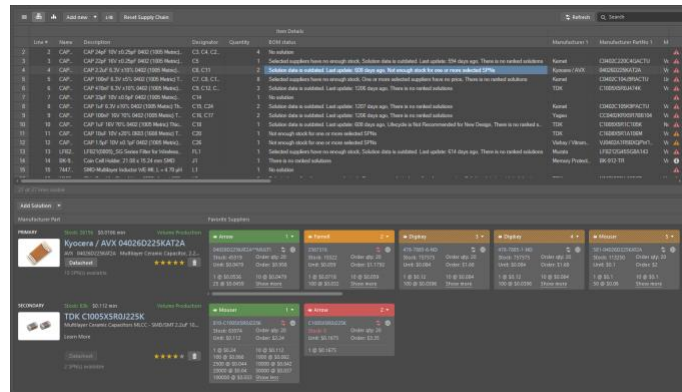
some sizing techniques used by sports such as golf or mountain biking where fit is very important to the end user. As for motor failure, the risk of having a low longevity rating can result in a lower customer base as computer input devices are not typically categorized as disposable. The probability of the motor failing due to debris ingress is moderate because they are an unshielded, high-stress assembly and there are plenty of edge cases that are not accounted for. This high RPN risk will incentivize the design of a fully enclosed chassis if it does not jeopardize the cost constraint.

A sensitivity analysis may be necessary because the electronics will be mass produced. There are no requirements which will trickle down into a requirement for electronics that are particularly sensitive to component changes. Component tolerances are well characterized and low-cost parts at least meet the 5% range. In the digital and power subsystems of this project, this well outperforms the required tolerance for the circuit to behave predictably. However, a yield analysis will be performed on a small fabrication batch to determine how manufacturable the design is. Typically, there are overlooked design flaws in PCB footprints and component package choices that result in tombstoning or misorientation of components which leads to a lower-than-expected yield. An order of 500 units should be incorporated into the manufacturing plan before full scale production proceeds.

## V. Test Plans

The following test plan is derived from each requirement.

1. **Low Cost Relative to Headset:** Verify BoM cost for 1000 units is less than \$50 using the ECAD tool's BoM feature as shown below. The BoM may automatically calculate the cost breaks for ordering in high quantities.



Line #	Name	Description	Quantity	BOM Value	Manufacturer
2	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
3	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
4	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
5	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
6	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
7	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
8	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
9	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
10	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
11	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
12	CAP	CAP 100F 16V 0402 (1000 Pieces)	1000	0.00	Kingston AVX
13	RES	RES 100K 0402 (1000 Pieces)	1000	0.00	Kingston AVX
14	RES	RES 100K 0402 (1000 Pieces)	1000	0.00	Kingston AVX
15	RES	RES 100K 0402 (1000 Pieces)	1000	0.00	Kingston AVX
16	RES	RES 100K 0402 (1000 Pieces)	1000	0.00	Kingston AVX
17	RES	RES 100K 0402 (1000 Pieces)	1000	0.00	Kingston AVX
18	RES	RES 100K 0402 (1000 Pieces)	1000	0.00	Kingston AVX
19	RES	RES 100K 0402 (1000 Pieces)	1000	0.00	Kingston AVX
20	RES	RES 100K 0402 (1000 Pieces)	1000	0.00	Kingston AVX

**Add Solution**

**Favorite Solutions**

Manufacturer Part	Part Name	Description	Quantity	BOM Value	Manufacturer
Ypocera / AVX 0402D25HAT2A	0402D25HAT2A	0402 25V 0.1uF MLCC	1000	0.00	Ypocera
TKD C1005X5R0J225K	C1005X5R0J225K	0402 225K 5V MLCC	1000	0.00	TKD

Figure 17. Example of Altium's LiveBoM feature

2. **Force Production on Fingers:** It is important that the mechanical mechanism can restrain a user's finger from moving without failure. Equip the glove on each one and set the each finger on the glove to maximum distance constraint. Ask each user to squeeze as hard as possible and make ensure that the glove doesn't break. Finger flex is normal.
3. **Haptic Actuation:** Drive the actuators with a 10-50Hz AC haptic signal in 10Hz increments. One primary concern is too much energy stored in the mechanical system. Perform a test with the 10 different test users, asking them to rate if they had felt a noticeable sensation at each step. If more than 5/10 users report no sensation at any frequency, then there is significant attenuation due to the energy stored in the mechanical system and the requirement fails.
4. **Fine Resolution Linear Control of Actuator:** Wear the VR glove and record a video of the program looping through each step of the full span of motion for just the index finger.



5. Low Bulk: Weigh the VR glove on a postage scale. Check that the mass is less than 200g.



*Figure 18. Example of a postage scale*

6. Hygienic: Only if the glove was designed to the IP68 rating, submerge one pair of the gloves in 1.5M of water for 30 minutes. The water container can be made out of schedule 40 PVC pipe. Use the gloves in an intended virtual environment for 30 minutes and observe for anomalies. Any anomalies indicates a failure of the requirement.



*Figure 19. Example of a PVC pipe*

7. Long Battery Life: Measure a single glove's battery capacity by reading the capacity on the battery.
8. Low Equip Time: Count the number of breakable appendages on the final glove design. Ensure that there is a maximum of 2.

## VI. Development and Construction

### **Software VR World Development**

A VR demo world was developed to test and showcase functionality of the VR gloves. The demo world consisted of a table and a grabbable object, a cube, set in a plain environment. It utilizes the Oculus XR Plugin to manage the bulk of the interfacing with the Meta Quest 2 headset.

The development of the demo world started by creating a new 3D project in Unity (2021.3.26f1). The project's general settings, configuration settings, XR management, and rendering settings were configured according to the guidelines outlined by Meta ([here](#)). The Oculus Integration package and Open XR Interaction manager were added to the project.

From there, the camera, controller, and hand inputs were added to the project through the OVRCameraRig and InputOVR prefabs. The OVRCameraRig is responsible for rendering the view that the player will see in-game and for managing the anchors for the controllers and hands displayed in-game. The InputOVR prefab is responsible for managing the different types of interactions the player can do in-game. Since the InputOVR prefab only included one default interaction, poking objects, an additional interaction, grabbing objects needed to be added to each controller and hand object. The project now has both controller and hand tracking support for poking and grabbing objects in-game.

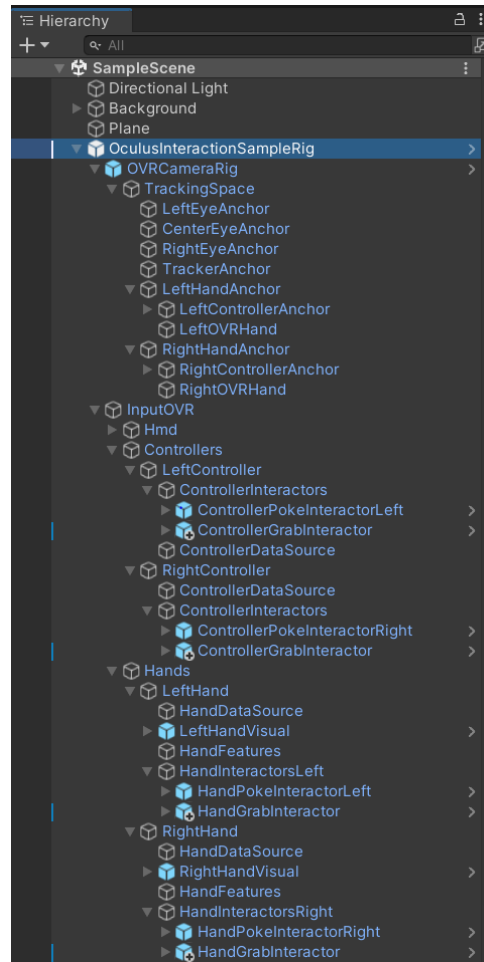


Figure 20. The expanded Unity project hierarchy of the OVRCameraRig and the InputOVR prefabs.

The next step to building the demo world is to create an object for the player to interact with. To do this, a plane had to first be added to the project. The plane serves as the ground for the player to stand on and keeps the player stationary in the world. Then, a table and a small cube were added to the project to give the player something to grab onto.



Figure 21. A screenshot of a playtest of the demo world showing hand tracking visualization, the white table, and the interactable cube. Side by side with the real world hand position.



Figure 22. The player making a gesture with their hands to show the hand tracking visualization. Side by side with the real world hand position.

### **Mechanical VR Glove Mechanism Development**

The design of the mechanical system took the students the longest time because it was outside of their field of expertise.

The first design of the mechanical system targeted to only achieve the finger restraint requirement; as a result, the design sacrifices the ability to provide haptic feedback. The principle of operation relies on a freewheel with notches that can be stopped by a mating rod that slides into the notch. The freewheel is mechanically coupled to the finger via a string; this provides one-way coupling so a spring-loaded reel is added to provide a retractive force for when the user relaxes their fingers.

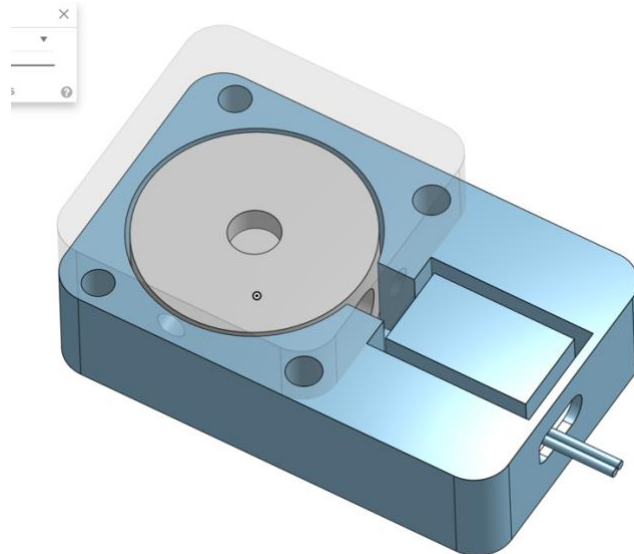


Figure 23. Isometric view of the notched freewheel mechanism

See the Mechanical Integration section below to see the problems that motivated the next design revision.

Although the first design could be improved upon, it was decided upon to move to a completely new mechanism that relied on a different principle of operation. This design couples the solenoid action directly to the string which is then attached to the user's finger. A string is routed through the hole circled in black in the side view of the mechanism below. The turquoise part is the

moving part because it is fixed to the solenoid head with epoxy; the rest of the assembly is one solid piece and does not move relative to the user's hand. When the blue components shown below are compressed by the solenoid action, the friction between the parts will pinch the string and restrain its movement if the user attempts to pull on it. The rails on the bottom of the mechanism make it compatible with LucasVR's rigid mount hand interface which was available to print on [GitHub](#) (LucidVR, 2022). In theory, this design addresses the compression problem by leveraging the rigid mount hand interface which minimizes glove compression between the mount point and actuation point. It also addresses the poor tolerances from the previous design by manufacturing the friction mechanism with an SLA (stereolithography) 3D printer which has much finer tolerances; in the following figures, all blue components are 3D printed with a high-precision SLA printer and all gray components are printed with an FDM printer. Finally, the new design should provide a near infinite engagement resolution and would primarily be limited by the speed at which the solenoids can actuate.

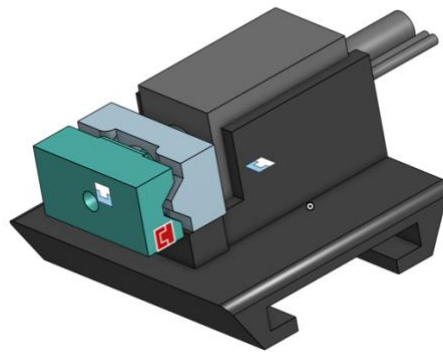


Figure 24. Isometric view of the string friction mechanism

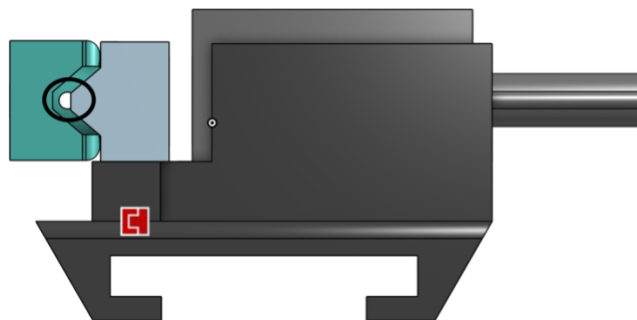


Figure 25. Side view of the string friction mechanism

## Electrical VR Glove Development

The following schematics and layouts can be found in the appendix as well as [viewed interactively in a web browser](#). Screenshots are referenced below for the reader's convenience. The first revision prototype is built in a modular fashion to decrease the risk of failure and increase prototyping speed; however, modularity usually comes at the cost of volume which is the case with the first prototype. The top schematic is split up between a Bluetooth enabled microcontroller, a battery, a boost converter, and solenoid drivers. Some of the modules are

implemented in the browser design software (Flux), some are implemented in other browser design software (EasyEDA), and some are implemented by a third party (Adafruit). This design is quite straightforward as it connects the appropriate power nets and signal nets together. The layout was not constrained by space so large traces widths capable of handling 3A were set in the layout to minimize the calculations/optimizations needed to minimize their size.

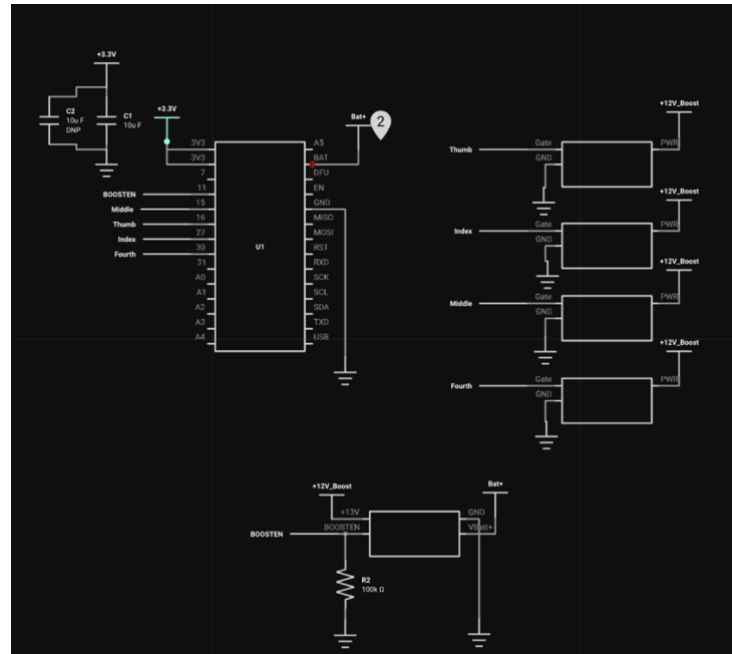


Figure 26. Top level schematic of the first revision of the VR Glove control board

The solenoid driver submodule schematic can be [viewed interactively in a web browser](#). The solenoids will be connected to the output of the boost converter which can be configured to 12V or 13V. The low-side switch uses an AO3400A N-Channel MOSFET. It is pulled low such that the MOSFET remains in a high-impedance state by default; this ensures the solenoids are inactive. When 3.3V is applied to the transistor's gate by the microcontroller, the transistor enters a low impedance state and the solenoid becomes active. The operating point of the MOSFET can be intuited by the On-Region characteristics as cited from the datasheet, assuming a current of around 0.45A which is the measured current draw of the solenoid at 12V; in this case, the voltage drop across the MOSFET in the low impedance state is very minimal, about 20mV, as shown in the simulation figure below. The flyback diode is added so that reverse voltage spikes over 0.9V are dissipated and the downstream circuitry is protected. The sublayout of this circuit was quite straightforward, the only design consideration was that the trace widths should be at least 5 mil to withstand 0.5A at room temperature. The trace widths were unnecessarily set to 40 mil to ensure a very minimal trace impedance.

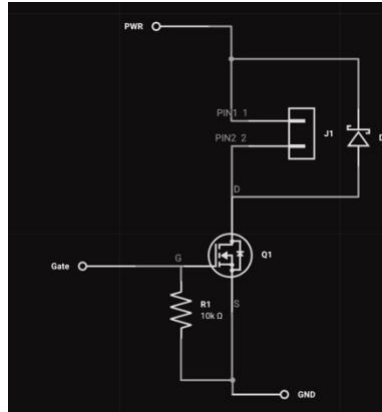


Figure 27. Schematic of Low-Side MOSFET Solenoid Driver Submodule

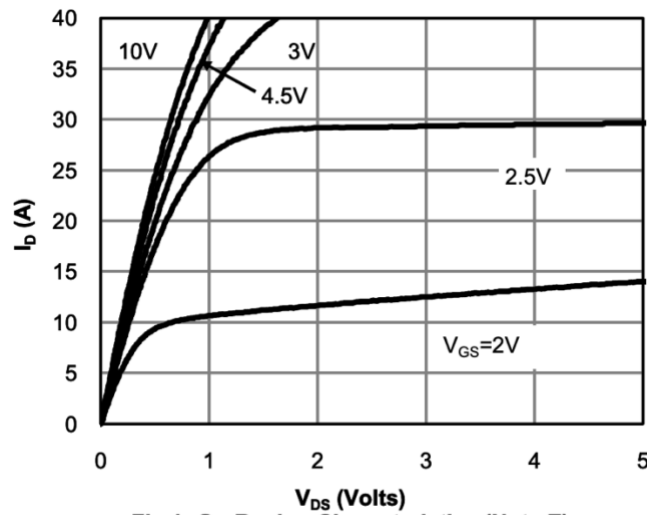


Fig 1: On-Region Characteristics (Note E)

Figure 28. On-Region Characteristics (Note E) from AO3400A datasheet

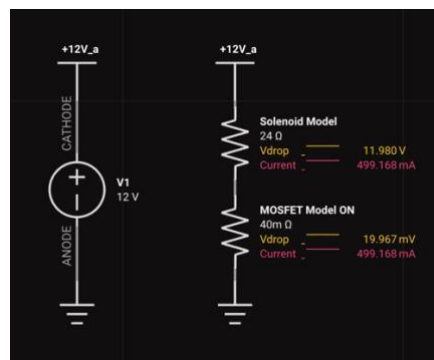


Figure 29. Model of MOSFET when set to the low impedance state

The boost converter was designed in EasyEDA Pro because of its advanced polygon layout tool. The schematic was taken from the [datasheet's](#) application note and the voltage feedback resistor values were calculated using the following formula.

$$1.2 * \left( 1 + \frac{113k\Omega}{11.5k\Omega} \right) = 12.99V \text{ Output}$$

The rationale for overdriving the solenoids at 13V was that it was a simple way to hopefully increase the strength of the magnetic field they produced.

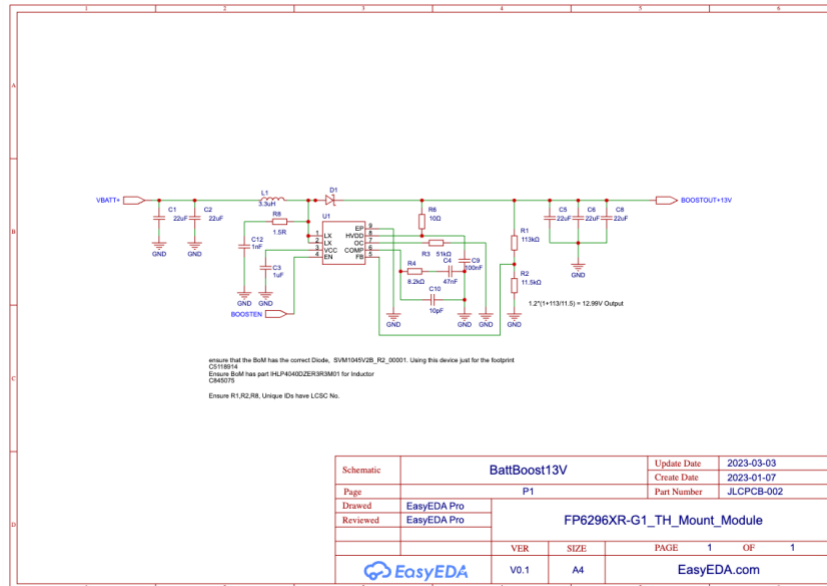


Figure 30. Schematic of DC-DC Boost Converter Submodule

The layout of the boost converter can be seen in the figures below. First, the inductor was placed because it is the most important to minimize the area of the switching polygon; this helps reduce unwanted EMI, because the polygon is essentially an antenna switching at the boost IC's switching frequency of 400kHz. Next, the diode was placed due to its large size. The input and output capacitors were placed next, and finally the feedback, compensation, and bootstrap passives were placed. After the appropriate polygons were drawn, the passive components were routed, making sure to keep their lengths as short as possible. Finally, stitching vias were added to ensure the components had a good electrical return path and thermal path.

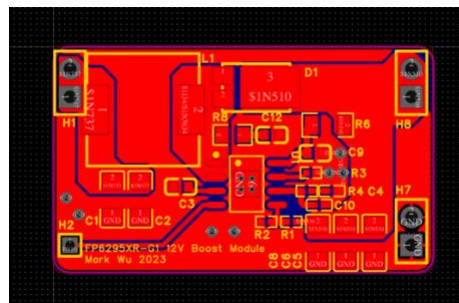


Figure 31. Top copper from Layout of boost converter submodule



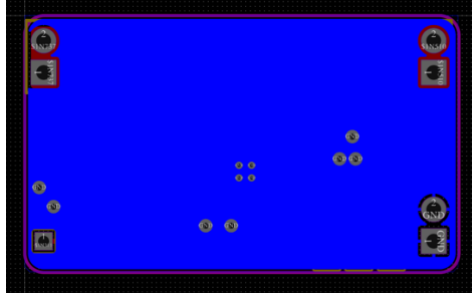


Figure 32. Bottom copper from Layout of boost converter submodule

The Adafruit Feather nRF52840 Express was picked for this prototype because the nRF52840 was a suitable solution for the final microcontroller/wireless module. The primary driving factor was that the nRF52840 is an established platform with strong firmware support by Nordic Semiconductor. On top of that, the Arduino foundation wrote their own open source API wrapper which can be found on [GitHub](#) (Adafruit, 2023). The hardware requirements for a microcontroller on this project are very minimal: four low frequency PWM channels and integrated Bluetooth LE support. Additionally, there is minimal required auxiliary circuitry for the highly integrated nRF52840 so adapting the PCB module to a high-density design would require minimal design effort. Although ESP32s have been popular in the maker community for Bluetooth PCB modules, their firmware development tools are less developed, primarily due to the choice to not use an ARM processor architecture which is an extremely mature embedded platform foundation.

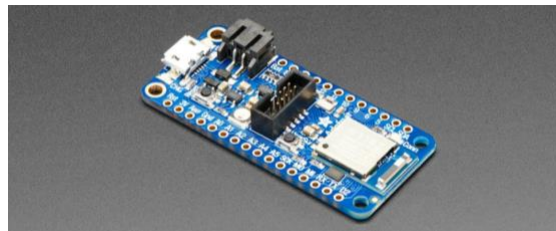


Figure 33. Product picture for Adafruit Feather nRF52840 Express

### **Firmware VR Glove Development**

Because the electrical solution leveraged a microcontroller with Arduino support, a lot of the firmware work was completed in the Arduino blueprint example. The only change that had to be added was defining a data structure to capture the state of each solenoid and writing a function to change those states. The easiest solution to implement is to encode the state of all 4 solenoids as a single ASCII character. Although, this makes the states less human readable, it simplifies the function calls that need to be written because the Arduino example receives one byte at a time. An ASCII character is encoded by 1 byte which can store 8 states which makes this architecture extendable if more actuators are added in future designs. The function that accomplishes this check is appended below.

```

void decodeCharToSolenoid(uint8_t codedStates)
{
    //the codedStates variable represents finger states from the LSB in the
    following order:
    //1      - INDEX
    //10     - FOURTH
    //100    - MIDDLE
    //1000   - THUMB
    if ((codedStates | 0b1) == 0b1) {
        analogWrite(INDEX ,PWMDUTY);    //analogWrite is Arduino's PWM command
    }
    if ((codedStates | 0b10) == 0b10) {
        analogWrite(FOURTH ,PWMDUTY);
    }
    if ((codedStates | 0b100) == 0b100) {
        analogWrite(MIDDLE ,PWMDUTY);
    }
    if ((codedStates | 0b1000) == 0b1000) {
        analogWrite(THUMB ,PWMDUTY);
    }
}

```

*Figure 34. Sanity check test firmware for electrical subsystem integration testing*

## VII. Integration and Test Results

### Electrical Integration

The boost converter submodule was constructed and independently evaluated under no load conditions to regulate at 13V. The boost converter soft start can be seen below as a 500us rise to 13V. The ringing is due to a large probing loop; this was verified because the peak to peak of the ringing was attenuated after the probe was switched from 10X to 1X; the frequency of the ringing was also measured to be 420kHz which is close enough to the boost converter's 400kHz nominal switching frequency. This implies that the probe is acting as a receiver to the switching plane antenna.

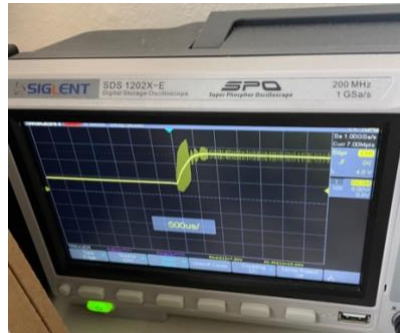


Figure 35. Transient response of boost converter module when the EN pin is toggled high

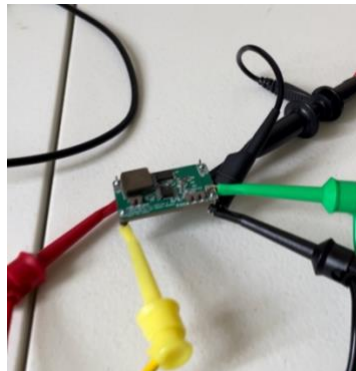


Figure 36. Probing setup of the boost converter module. Inputs: Red was 3.3V, Yellow is 3.3V, Green and Black are GND.

The main PCB incorporated the integration of various electronic subsystems, including the solenoid driver circuitry. Next, test firmware was developed to control the low side solenoid driving circuitry. To initiate the electrical system, the battery and two solenoids were connected.

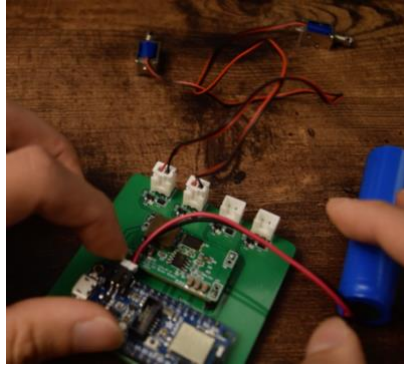


Figure 37. Connecting the solenoids and battery to the assembled main PCB

*During testing, it was observed that the microcontroller effectively enabled the boost converter, and the solenoids were able to alternate between on and off states, with each state lasting for 1 second during a minute of operation.*

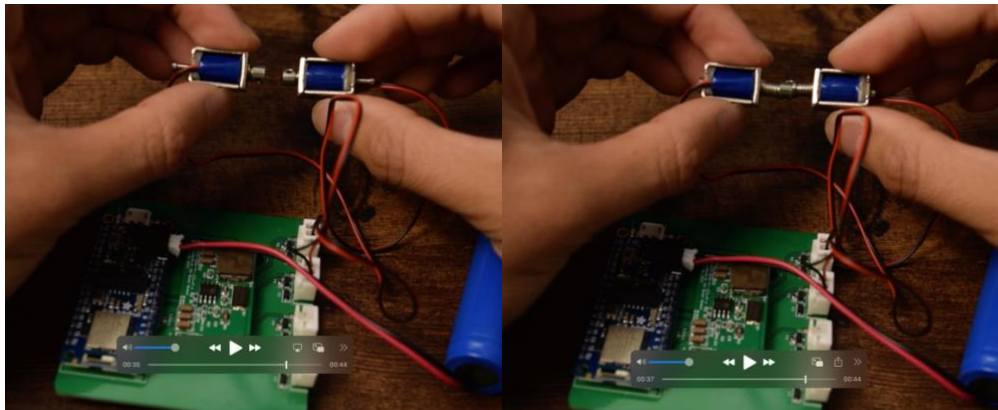


Figure 38. Two states from video footage of the solenoids activating on and off according to the test firmware

However, an issue arose when the battery was removed and then reconnected, as the boost converter no longer regulated at the desired 13V and only outputted 3.5V. To diagnose the failure, the most suspicious cause of failure involves the back EMF that is generated by the solenoids when they are suddenly released. Thus, the schematic for the solenoid driver was checked and it was discovered that the flyback diodes had a maximum 0.9V forward voltage which would cause a 13.9V drop across the boost converter output. 13.9V be too close to the 14V maximum rating of the boost converter IC and new flyback diodes should be selected or the boost converter should be reconfigured to regulate at 12V.

Although this hypothesis was not confirmed with absolute certainty, one piece of evidence for it was that replacing the boost converter IC on the boost converter PCB module restored the 13V regulation as seen in the figure below. The next step to resolve this issue was to replace the feedback resistors on the boost converter to regulate at 12V. In the event of a reverse emf collapse from the solenoids, the total voltage across the internal FET would be 12.9V which provides a 1.1V buffer from the 14V maximum rating. However, the boost converter used 0402 feedback resistors and the only multi-value resistor book that was available to the students at the time had an 0603 package size. Thus, the board was not fully repaired for repeat use.

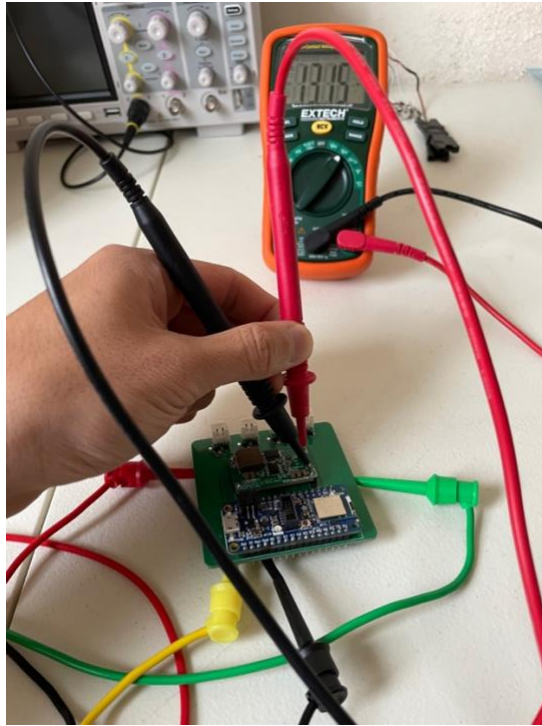


Figure 39. Restored regulation of boost converter after replacing the main boost IC

The suggested next steps to remedy this design is to configure the boost converter to regulate to 12V by changing the feedback resistors and to potentially choose reverse voltage protection diodes that have a lower forward voltage.

### Mechanical Integration



Figure 40. First version of VR Glove with the notched flywheel mechanism

[This video](#) captures the results from integrating the first mechanical design. During the integration test, several issues were encountered. First, it was discovered that mounting the mechanism directly to the glove caused compression, which was an unexpected outcome. Secondly, the tolerances that resulted from FDM (Fused Deposition Modeling) 3D printing were worse than anticipated, leading to excessive friction. Finally, the placement of the notches on the freewheel was not sufficiently close, resulting in the mechanism only engaging twice throughout the entire finger movement; thus, a lower engagement resolution. These findings highlight the need for improvements in the mounting process, tolerances, and notch placement to enhance the overall performance of the design.



Figure 41. The rigid mount hand interface by LucasVR with the string friction-based design mounted on one of the slots.

The mechanical iteration process took much longer than predicted due to 3D printer reliability problems and the friction mechanism has not been rigorously tested yet.

From the initial impressions of the mechanism, the suggested next steps are to add a silicone coating on the string interface surface to greatly increase the friction between the mechanism and the string, decreasing the amount of force required to restrain a user's fingers.

### **Firmware Integration**

The most that could be tested before integrating the firmware with the software was Bluetooth connectivity to the Meta Quest 2. The Meta Quest 2 used was able to discover the Adafruit Feather nRF52840 Express and establish a connection. Unfortunately, it was difficult to screenshot the settings menu of the Meta Quest 2 because it required exiting the settings menu.



Figure 42. Arduino Serial terminal during connection to Meta Quest 2

### Software Integration

To help integrate the hardware Bluetooth capabilities with the Unity demo world, a Unity Bluetooth plugin was added to the Unity project. This plugin included a library of various functions to help scan for Bluetooth devices, connect to other devices, and read from or send information to other devices. The provided Starting Example script from the Bluetooth plugin was attached to the camera rig in the Unity project, allowing the Unity demo world to scan for and connect to the Bluefruit module. However, a build issue with Unity prevented the testing of the Bluetooth plugin with the Bluefruit module.



## VIII. Conclusion

Unfortunately, the work completed in the project did not follow the proposed schedule due to each task taking much longer than anticipated. The mechanical iteration proved to be a large learning experience for both students involved. There wasn't enough time to test all 3 proposed iterations and the proposed verification methods for each requirement were unrealistic to spend time on. However, attempting to achieve such an ambitious schedule and abide by such strictly defined requirements for a project that evolved as much as this one provided a high height to fall from which resulted in both students learning a lot about integrating practical solutions. There were specific situations like how the students learned to double check the forward voltage in protection diodes, and there were higher level learnings like how the students learned not to spend too much time on over-defining a problem. The project still proved and disproved electrical, firmware, and mechanical concepts from one solution to the VR glove problem. This report hopes to inspire or warn other engineers who find themselves working towards this fun problem.



## IX. Bibliography

[10] Expert Market Research, “Global Virtual Reality Glove Market Report and forecast 2022-2027,” Virtual Reality Glove Market Analysis, Size, Share, Report 2022-2027, <https://www.expertmarketresearch.com/reports/virtual-reality-glove-market> (accessed Oct. 18, 2022).

[11] “Meta quest 2: Immersive all-in-one VR headset: Meta store,” Meta, <https://www.meta.com/quest/products/quest-2/> (accessed Oct. 18, 2022).

[12] “Best VR and XR headsets for professional use - purchase at varjo.com,” Varjo.com, <https://varjo.com/purchase/> (accessed Oct. 18, 2022).

[13] A. Heath, “Meta’s social VR Platform Horizon Hits 300,000 users,” The Verge, <https://www.theverge.com/2022/2/17/22939297/meta-social-vr-platform-horizon-300000-users> (accessed Oct. 18, 2022).

[14] C. Kolmar , “25 amazing virtual reality statistics [2022]: The future of VR + AR,” 25 Amazing Virtual Reality Statistics 2022 The Future Of VR + AR Comments, <https://www.zippia.com/advice/virtual-reality-statistics/#:~:text=There%20are%20over%20171%20million%20people%20using%20VR%20today.,with%2083.7%20using%20this%20technology.> (accessed Oct. 18, 2022).

[1] Adafruit, “Adafruit code for the nordic NRF52 ble soc on Arduino,” GitHub, [https://github.com/adafruit/Adafruit\\_nRF52\\_Arduino](https://github.com/adafruit/Adafruit_nRF52_Arduino) (accessed Jun. 6, 2023).

[2] “10a current mode non-synchronous PWM boost converter general description,” LCSC Datasheets, [https://datasheet.lcsc.com/lcsc/1810182103\\_Advanced-Analog-Technology-FP6296XR-G1\\_C133888.pdf](https://datasheet.lcsc.com/lcsc/1810182103_Advanced-Analog-Technology-FP6296XR-G1_C133888.pdf) (accessed Jun. 7, 2023).

[3] LucidVR, “LucidVR/Lucidgloves Github Repository,” GitHub, [https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4/RigidMount\\_rightHand.STL](https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4/RigidMount_rightHand.STL) (accessed Jun. 6, 2023).

[4] R. Gupta, K. Kumar, and S. Shah, “A brief history of virtual reality, in patents,” A Brief History of Virtual Reality, in Patents, <https://www.finnegan.com/en/insights/articles/a-brief-history-of-virtual-reality-in-patents.html#:~:text=While%20plenty%20of%20development%20was,a%20measure%20of%20finger%20movement.> (accessed Oct. 18, 2022).

[5] L. LucidVR, “Lucidgloves: VR haptic gloves on a budget,” Hackaday.io, <https://hackaday.io/project/178243-lucidgloves-vr-haptic-gloves-on-a-budget> (accessed Oct. 18, 2022).

[6] Nyan man | Mod, “LucidVR - faq channel,” Discord, <https://discord.gg/lucidvr> (accessed Oct. 18, 2022).

[7] Nathie, “I felt the metaverse with these VR haptic gloves,” YouTube, <https://www.youtube.com/watch?v=Zm3YQTt6UNU&amp;t=568s> (accessed Oct. 18, 2022).

[8] L. LucidVR, “LucidVR/Lucidgloves: Arduino/ESP32 based DIY VR haptic gloves. compatible with Steamvr via OpenGloves.,” GitHub, <https://github.com/LucidVR/lucidgloves> (accessed Oct. 18, 2022).

[9] Linus Tech Tips, “I’ve never felt this way before - \$60 VR gloves!,” YouTube, <https://www.youtube.com/watch?v=jGyHT0t-mtw> (accessed Oct. 18, 2022).

**I. Appendices**

- A. Senior Project Analysis
- B. Specifications and requirements
- C. Parts List and Costs
- D. Schedule - Time Estimates
- E. Schematics
- F. PCB Layout
- G. Firmware (Arduino)
- H. Software (Unity)
- I. FMEA

## Appendix A. Senior Project Analysis

Project Title: VR Force Feedback Gloves

Students' Names: Mark Wu, Claire Chen

Students' Signature: m.w. c.c.

Advisor's Name: William L. Ahlgren

Advisor's Initials:

Date: 05/31/23

To summarize the functional requirements, the VR Glove shall be able to restrain 4 out of 5 of the user's fingers. Each restraint mechanism shall be electrically controlled by a device that can communicate with the Meta Quest 2. The electrical signals controlling each finger shall be independently able to toggle based on information given to the controller by the Meta Quest 2. Externally, the Meta Quest 2 shall be able to detect collisions between the tracked hand model and virtual objects and instruct the VR Glove to restrain each finger accordingly. Finally, the device must optimize for a small form factor when possible.

The primary constraints with the project to target a manufacturable design was cost. This helped the design process because it helped narrow down the solution space and made the problem more approachable. The limiting factor to the growth of this design was its novelty application and the lack of COTS electromechanical devices that suited the application. Coupled with the students' limited mechanical experience, this made the design of the electromechanical mechanism quite difficult to accomplish. The decision to limit the design to COTS components was driven from time limitations, scope limitations, and least importantly, cost limitations. In the future, it is recommended that a team attempting to solve this problem approach it from an actuator design focused approach if they want to achieve an even smaller form factor.

There are many economic impacts that should be considered for the VR Glove project. With regard to human capital, the VR glove project has proved to involve a lot of manual labor to assemble the small scale mechanisms. While the design leverages COTS components to decrease the COGS, these COTS components are distributed across different disciplines (e.g. mechanical parts, electromechanical actuators) which increases the amount of assembly work that is required. The COGS which includes the assembly cost requires large financial capital if the project were to be scaled to a consumer level. The project aimed to target an open source launch from the onset; this model best aligns the project's design decisions with the customers' desires. As key shareholders of the platform, the end user can have a higher level of autonomy to preserve any interests they have which may become profit opportunities for a company manufacturing the VR gloves; one relevant example of this is user privacy.

In the best case that there is an adoption of 0.1% of the 6.5 million Meta Quest 2 yearly units sold, there will be 65,000 VR gloves sold per year. As seen in the cost breakdown in the appendix, each device costs 32.49 USD to make. The estimated purchase price of each device is 150 USD. This amounts to 117.51 USD of profit per device. Assuming there are 65,000 gloves sold per year, the project can make a profit of 7.64M USD. These figures are summarized in the table below.

Annual Users	COGS (USD)	Unit Sale Price (USD)	Unit Profit (USD)	Annual Profit (USD)
65,000	32.49	117.51	117.51	7.64M

The cost for the user to operate the device for 1 hour can be derived from the energy consumption of the device. From similarities in hardware from a previous prosthetic arm project done by the student, the gloves can draw up to an average of 2W of power; this equates to 2Wh of energy in one hour of gameplay. Assuming 0.25 USD per kWh for electricity in California, the gloves would cost the user 0.0005 USD to operate per hour.

There are large environmental concerns with consumer electronics, especially due to the globalized nature of the modern economy. One large concern arises from the disposal method of PCBs and the toxic element lead. This project aims to be as low cost as possible and low-cost consumer electronics are known to undergo Leaded HASL surface treatment as well as leaded reflow soldering which introduces the toxic chemical into the product. Due to improper disposal of these gloves, they may end up in heaps where scavengers burn the electronics to obtain various base materials. The scavengers more closely interact with the toxic lead element and can sustain permanent bodily damage from such activity. Such places where this occurs is in Ghana as reported in one news article. Unfortunately, the problem extends to the toxic fumes inhaled from other plastics and epoxies found in all electronics but the long term effects are not well researched. Additionally, some of the gloves can end up in ocean waste instead of a safe landfill, which can negatively affect local wildlife by releasing trace amounts of lead into their environment. This project uses textiles and electronics; key sources from these sectors are petroleum and metals. Although not significantly contributing to the overall use of these materials, the project contributes to a high consumer demand of these two unsustainable natural resources. By relying on existing low-cost manufacturing techniques that use these highly commoditized resources, this project does not make room to consider sustainable alternatives.

The manufacturing of textiles has ethical concerns due to some countries not respecting internationally recognized human rights. Because this project places such a high importance on low cost and relies on outsourced labour, it is possible that the product may be manufactured in dangerous work environments and with underaged labourers.

PCB manufacturing is not sustainable because none of the product is recycled and none of the copper, epoxy, fiberglass, or water is naturally replenished. At a certain point, modern electronics production as we know it will have to change and the project can adapt to new methods of integrating electronics. However, the project does not aim to sustain itself past a few years of product support.

One large social issue with the use of VR Gloves is the shift closer to a completely digital and simulated society. The stakeholders of this issue are the individuals who have access to the simulated world, the support network around those individuals, groups with interests in power, and individuals who do not have access to the simulated world. With the philosophical implications aside of eventually moving a large proportion of our lives to the simulated world, there are real-world concerns of access and power to the new virtual world that uses VR gloves to provide maximum immersivity. This problem already exists because we spend a majority of our time in a digitally connected world with the most immersive feedback devices being a screen, mouse, and keyboard; with gradual increases in immersivity, we will see more and more activities we normally do migrate to the simulated reality out of convenience. Early adopters already spend most of their time socializing in the virtual world and other adopters are using the platform

to replace their exercise regime. Although we may never simulate all of reality, we only have to simulate a large amount of it for ethical issues to arise.

The first stakeholder to benefit initially are the individuals with access to the simulated world. Because simulating reality on human beings requires hardware to interact with our senses, there is a physical and thus financial barrier that needs to be crossed to participate in the simulated world. If this causes a divide such that only affluent individuals can access the simulated world, this can exacerbate the wealth divide by further adding a barrier which bars less affluent individuals from access to the affluent. Often times, there are channels of mingling between the classes that gives rise to social mobility, such as institutions like colleges or open-to-all events such as networking conferences. These channels may be harder to implement if accessibility to simulation hardware is another hurdle to overcome.

Cybersecurity attacks happen all the time but are motivated by the technology and benefits they produce; a highly immersive and widespread experience can motivate some conflicts between stakeholders with power interests and stakeholders who use simulation hardware. Due to the simulated world's digital nature, it becomes feasible to anonymously perform a large scale attack which maliciously alters the desired simulation to a large number of users. One example of this is governments who may want to influence the mindset of a large group of individuals to relegate more power to them or private organizations who may want to influence the behaviours and beliefs of individuals to provide them with wealth and stability. With access to the headsets and gloves of an entire society, terrorist groups can leverage cybersecurity attacks to show typically censored information to sway the confidence of entire countries. The core problem between these stakeholders relies on the new backdoor to easily send coordinated, high-fidelity information either discretely or openly to a very large group of individuals. If the increased immersion this project aims to commoditize creates perverse incentives, stronger cybersecurity systems need to be developed to disincentivize this sort of malicious activity.

The first conflict with the IEEE code of conduct can be observed with the welfare of the public (No.1 "1. to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment." If the VR Gloves project expands to general virtual immersivity devices, it can give power to the stakeholders who have centralized control of simulation technology programming. This centralized power can increase the likelihood of violation of the privacy of others through the large amounts of wealth that can be gained from gathering more information from the simulation devices. The project shall consider methods to separate information from large VR vendors or watermark it such that it can be identified if misused.

## Appendix B. Specifications and requirements

Marketing Requirements	Engineering Specification	Justification
Low-Cost Relative to Parent Headset	The total BoM Cost of one unit shall cost less than \$50	By the 1/3 Rule of Thumb, the COGS cost should be 1/3 of the target sale price. \$150 for the target sale price is less than half of the \$400 headset that supports it
Force Production on Fingers	The device's actuator/driver system shall be able to apply a DC bias	This is the core mechanism that communicates absolute location to the user, which is a prerequisite to sensations such as density, mass, and weight
Haptic Actuators	The device's actuator/driver system shall have enough bandwidth for a low frequency 10-50Hz AC signal	This reduces the need to integrate additional vibrational haptic motors which typically oscillate at 10-50Hz
Fine Resolution Linear Control of Actuator	The device actuator/driver system shall take a linear input control signal and output a > 200 step DC bias with +- 2 step tolerance	steps per finger should be enough to produce a wide variety of hand configurations to convey object shapes and is easily realizable by an 8 bit driver. The 2 step tolerance is hard spec to meet due to tolerances of the
Low Bulk	The device shall not weigh more than 200g	The oculus controllers that come with the headset weigh roughly 150g. Given that the hardware will be secured tightly to the user, a slightly higher mass can be permitted.
Hygienic	The device shall be IP68 waterproof	Other gloves support machine washability. IP68 is the rating required to survive for that duration in the washing machine.
Long Battery Life	The device shall have a battery capacity of 3000mAh	Assuming 1A average power consumption (from personal experience on a prosthetic glove project), the gloves will last 3 hours; a reasonable duration of playtime.
Low Equip Time	The device shall limit actions to equip to only 2 motions	Having more than 2 straps to tighten per hand becomes tedious for the user.

Figure 43. Engineering Requirements

## Appendix C. Parts List and Costs

Table 1. BoM for single order of VR Glove electronics components

QTY	MPN	Unit Price	Ext. Price	Description	Total Cost
4	B5819W SL	0.03	0.12	Protection Diode	\$36.91
4	HX25003-2WAG	0.014	0.056	2-pin JST Connector	
4	AO3400A	0.04	0.16	N-Channel MOSFET	
2	CL10A106MQ8NNNC	0.0052	0.0104	10uF 0603 Capacitor	
5	RC0402FR-0710KL	0.001	0.005	10kOhm resistor	
1	4062	24.95	24.95	Adafruit Feather nRF52840 Express	
1	1781	9.95	9.95	Lithium Ion Cylindrical Battery - 3.7v 2200mAh	
5	CL31A226KAHNNNE	0.0496	0.248	22uF ±10% 25V	
1	0603B105K250NT	0.003	0.003	1uF ±10% 25V	
1	CC0402KRX7R9BB473	0.0023	0.0023	47nF ±10% 50V	
1	0603B104K500NT	0.0024	0.0024	100nF ±10% 50V	
1	CC0402JRNPO9BN100	0.0011	0.0011	10pF ±5% 50V	
1	CC0603KRX7R9BB102	0.0021	0.0021	1nF ±10% 50V	
1	SVM1045V2B_R2_000	0.2294	0.2294	TO-277B Schottky Barrier Diodes (SBD) ROHS	
1	IHLP4040DZET220M11	0.7048	0.7048	22uH ±20% 4.5A 60.4mΩ	
1	RC0402FR-0751KL	0.0007	0.0007	51kΩ ±1% 62.5mW	
1	RC0402FR-078K2L	0.0007	0.0007	8.2kΩ ±1% 62.5mW	
1	1206W4F100JT5E	0.0029	0.0029	10Ω ±1% 250mW	
1	FP6296XR-G1	0.463	0.463	Boost Vin=2.7V~12V Vout=13V 10A	

Charge Details		Shipping to:  United States Of America			
Engineering fee	\$8.00				
Via Covering	\$0.00				
Surface Finish	\$7.70				
Film	\$1.60				
Board	\$26.40				
Standard PCBA Price					
Setup Fee	\$24.63				
Stencil	\$8.09				
Components	-				
Feeders Loading fee	-				
SMT Assembly	-				
Hand-soldering labor fee	-				
Manual Assembly	-				
Build Time	-				
PCB: 5-6 days	\$0.00				
Calculated Price	<b>\$76.42</b>				
		Shipping Method	Costs	Delivery Time	Restriction
		<input checked="" type="radio"/> DHL Express Worldwide	\$58.96	2-4 business days	≤10000kg
		<input type="radio"/> DHL Express Economy	\$51.20	3-5 business days	0.5-21kg
		<input type="radio"/> UPS Express Saver	\$49.68	3-5 business days	≤15kg ≤120x70cm
		<input type="radio"/> FedEx International Priority	\$38.36	4-6 business days	≤21kg ≤121x121cm

Figure 44. JLCPCB Quote for manufacturing, assembly, and shipping of 100 units

The manufacturing, assembly, and shipping from JLCPCB's initial quote for an order of 100 assembled boards comes out to \$1.15 per board. Note that the cost only goes down as more units are produced. With respect to the component BoM, the PCB cost is almost insignificant.





















The next steps recommended for reducing the BoM cost is to integrate the nRF52840 module without its development board which brings the item cost down from about \$25 to around \$8-\$10 by sourcing the module from Mouser. The 18650 lithium-ion battery sold through Adafruit is also very














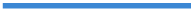







expensive; the cost can be reduced by at least \$5 by sourcing from Mouser. This would reduce the single glove component BoM cost down from \$37 to \$17.

## Appendix D. Schedule - Time Estimates

The appended Gantt chart PDFs represent the targeted schedule for the VR Force Feedback Glove project.

ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Successors	Resource Names	% Complete
1			<b>Define</b>	<b>109 days</b>	<b>Wed 6/1/22</b>	<b>Mon 10/31/22</b>				<b>94%</b>
2			Identify Product	78 days	Wed 6/1/22	Mon 9/19/22		3	Mark	92%
3			Advisor Approval	1 wk	Mon 9/19/22	Fri 9/23/22	2	4	Mark	100%
4			Abstract	1 day	Mon 9/26/22	Mon 9/26/22	3	5	Mark	100%
5			NABC	1 wk	Tue 9/27/22	Mon 10/3/22	4	6	Mark	100%
6			Design Requirements and Marketing Hierarchy	1 wk	Tue 10/4/22	Mon 10/10/22	5	7	Mark	100%
7			Test Plan	1 wk	Tue 10/11/22	Mon 10/17/22	6	8	Mark	100%
8			FEMA	1 wk	Tue 10/18/22	Mon 10/24/22	7	9	Mark	100%
9			LO and L1 Requirements	1 wk	Tue 10/25/22	Mon 10/31/22	8		Mark	80%
10			<b>Design</b>	<b>71 days</b>	<b>Tue 11/29/22</b>	<b>Tue 3/7/23</b>				<b>0%</b>
11			Compile Lessons Learned from LucidVR Glove, create video documentation	1 wk	Tue 11/29/22	Mon 12/5/22	26	12		0%
12			Design 4 single-finger actuator rigs and order parts	2 wks	Tue 12/6/22	Mon 12/19/22	11	33		0%

Project: TutorialEE460SeniorPro  
Date: Wed 11/30/22

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			













ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Successors	Resource Names	% Complete
13		<b>Desgin Rev 1 of Integrated PCB using selected Actuator</b>	<b>13 days</b>	<b>Tue 1/3/23</b>	<b>Thu 1/19/23</b>	<b>33</b>	<b>17</b>		<b>0%</b>
17		Desgin Review and Ordering of Rev 1 PCB (China Lead Times)	2 wks	Fri 1/20/23	Thu 2/2/23	13	27		0%
18		Design of Rev 1 Mechanical Mount System	2 wks	Tue 1/3/23	Mon 1/16/23	33	28		0%
19		<b>Design Rev 2 of Integrated PCB using selected Actuator</b>	<b>13 days</b>	<b>Wed 2/15/23</b>	<b>Fri 3/3/23</b>	<b>34</b>	<b>29</b>		<b>0%</b>
23		Design of Rev 2 Mechanical Mount System	2 wks	Fri 1/27/23	Thu 2/9/23	35	30		0%
24		Design of Rev 3 Mechanical Mount System	2 wks	Wed 2/22/23	Tue 3/7/23	37	31		0%
25		<b>Build</b>	<b>84 days</b>	<b>Tue 11/15/22</b>	<b>Fri 3/10/23</b>				<b>0%</b>
26		Project Kickoff - Build and program LucidVR Glove for learning	2 wks	Tue 11/15/22	Mon 11/28/22		11		0%




















Project: TutorialEE460SeniorPro Date: Wed 11/30/22	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			




ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Successors	Resource Names	% Complete
27		Build of Rev 1 PCB in Toaster Oven	3 days	Fri 2/3/23	Tue 2/7/23	17	34		0%
28		3D Print of Rev 1 Mechanical System	3 days	Tue 1/17/23	Thu 1/19/23	18	35		0%
29		Build of Rev 2 PCB in Toaster Oven	3 days	Mon 3/6/23	Wed 3/8/23	19	36		0%
30		3D Print of Rev 2 Mechanical System	3 days	Fri 2/10/23	Tue 2/14/23	23	37		0%
31		3D Print of Rev 3 Mechanical System	3 days	Wed 3/8/23	Fri 3/10/23	24	38		0%
32		<b>Validate</b>	<b>64 days</b>	<b>Tue 12/20/22</b>	<b>Fri 3/17/23</b>				<b>0%</b>
33		Waiting for parts, video document lessons learned from 4 actuator rigs	2 wks	Tue 12/20/22	Mon 1/2/23	12	13,18		0%
34		Checkout PCB Rev 1 functionality, check off FMEA	1 wk	Wed 2/8/23	Tue 2/14/23	27	19		0%
35		Checkout Mechanical Rev 1 functionality, check off FMEA	1 wk	Fri 1/20/23	Thu 1/26/23	28	23		0%

Project: TutorialEE460SeniorPro  
Date: Wed 11/30/22




















Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

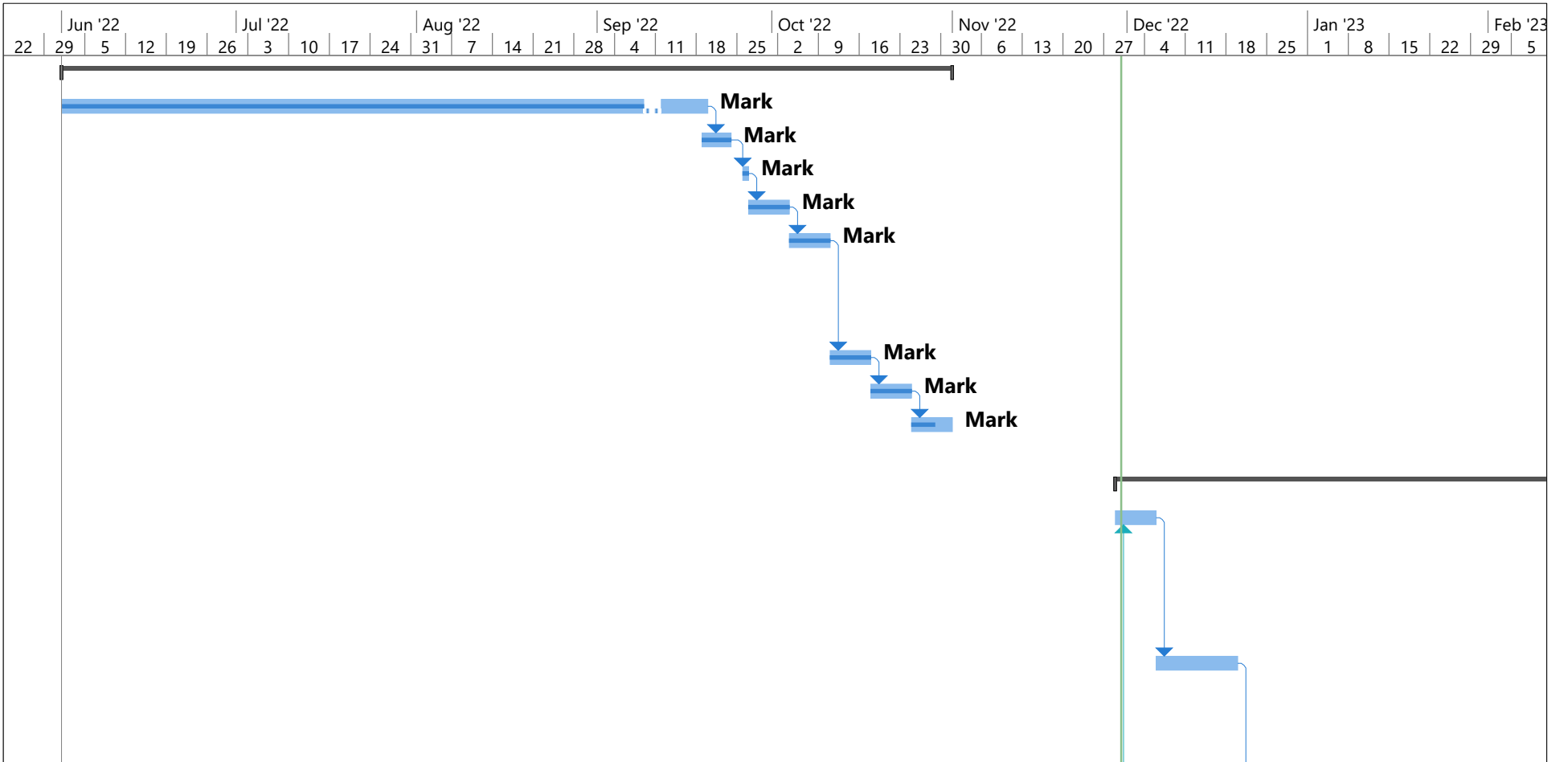
ID	 Task Mode	Task Name	Duration	Start	Finish	Predecessors	Successors	Resource Names	% Complete
36		Checkout PCB Rev 2 functionality, check off FMEA	1 wk	Thu 3/9/23	Wed 3/15/23	29	40		0%
37		Checkout Mechanical Rev 2 functionality, check off FMEA	1 wk	Wed 2/15/23	Tue 2/21/23	30	24		0%
38		Checkout Mechanical Rev 3 functionality, check off FMEA	1 wk	Mon 3/13/23	Fri 3/17/23	31	40		0%
39		<b>Release</b>	<b>25 days</b>	<b>Mon 3/20/23</b>	<b>Fri 4/21/23</b>				<b>0%</b>
40		<b>Create Manufacturing Plan</b>	<b>14 days</b>	<b>Mon 3/20/23</b>	<b>Thu 4/6/23</b>	<b>36,38</b>	44		<b>0%</b>
41	 	Consolidate Electrical BoM, assembly cost, mechanical injection molding, labour, other COGS factors	4 days	Mon 3/20/23	Thu 3/23/23		42		0%
42	 	Prepare Pitch Deck	1 wk	Fri 3/24/23	Thu 3/30/23	41	43		0%
43	 	Edit Pitch Video	1 wk	Fri 3/31/23	Thu 4/6/23	42			0%

Project: TutorialEE460SeniorPro Date: Wed 11/30/22	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

ID	 Task Mode	Task Name	Duration	Start	Finish	Predecessors	Successors	Resource Names	% Complete
44		Publish on Crowdsourcing - Project	1 day	Fri 4/7/23	Fri 4/7/23	40	45		0%
45		Write Report	2 wks	Mon 4/10/23	Fri 4/21/23	44			0%

Project: TutorialEE460SeniorPro  
Date: Wed 11/30/22

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

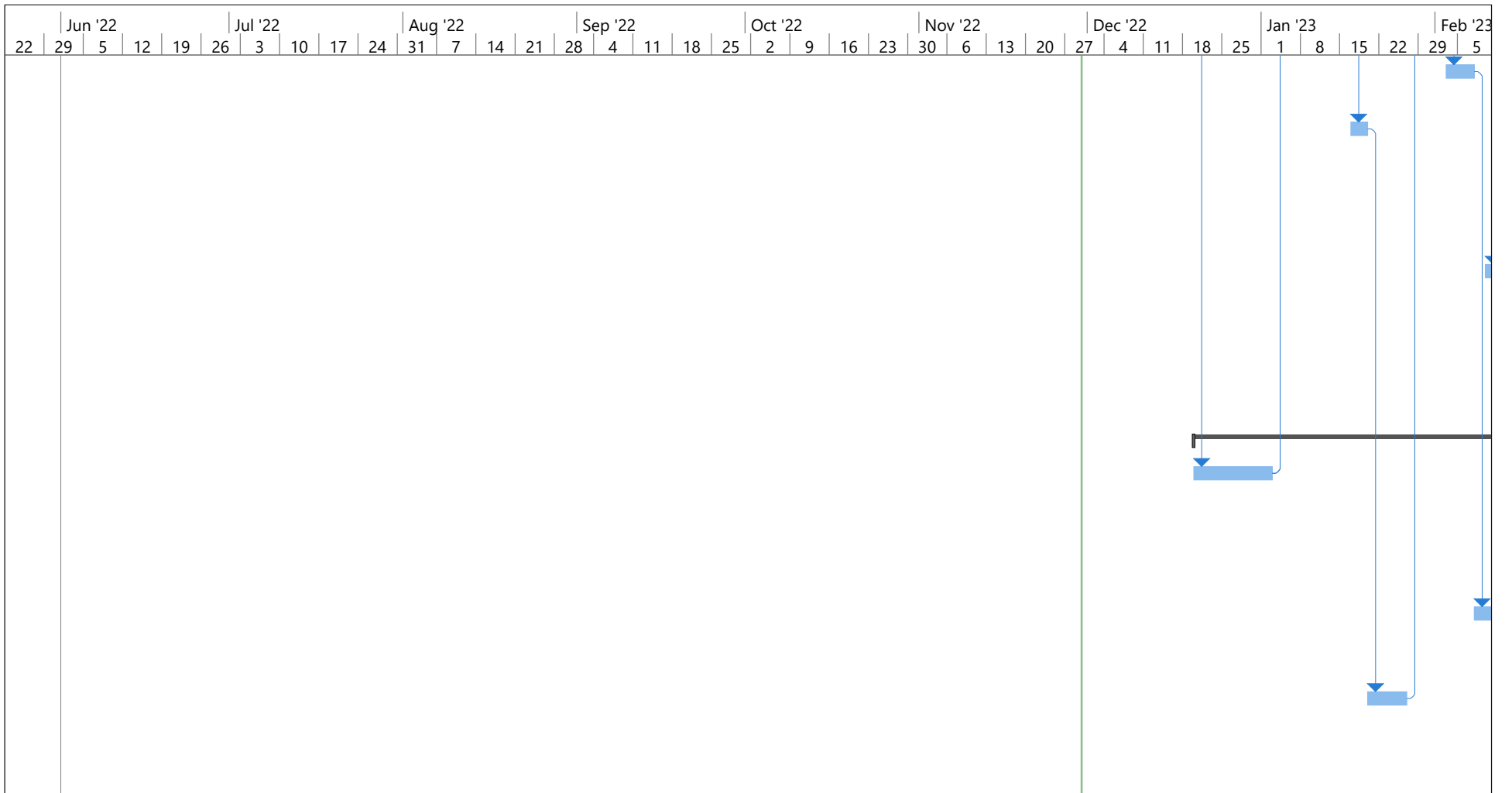


Project: TutorialEE460SeniorPro Date: Wed 11/30/22	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			



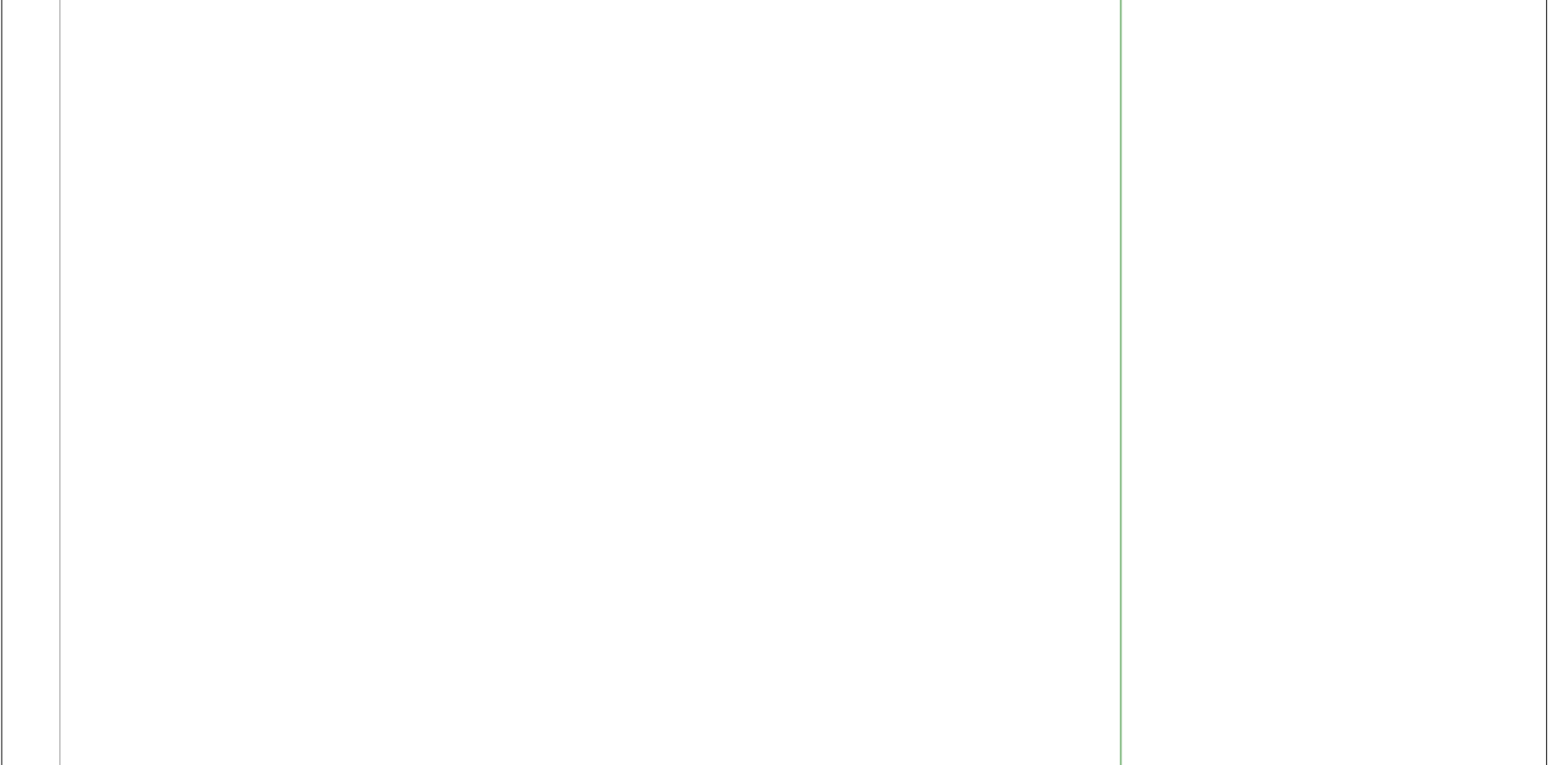


Project: TutorialEE460SeniorPro Date: Wed 11/30/22	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			



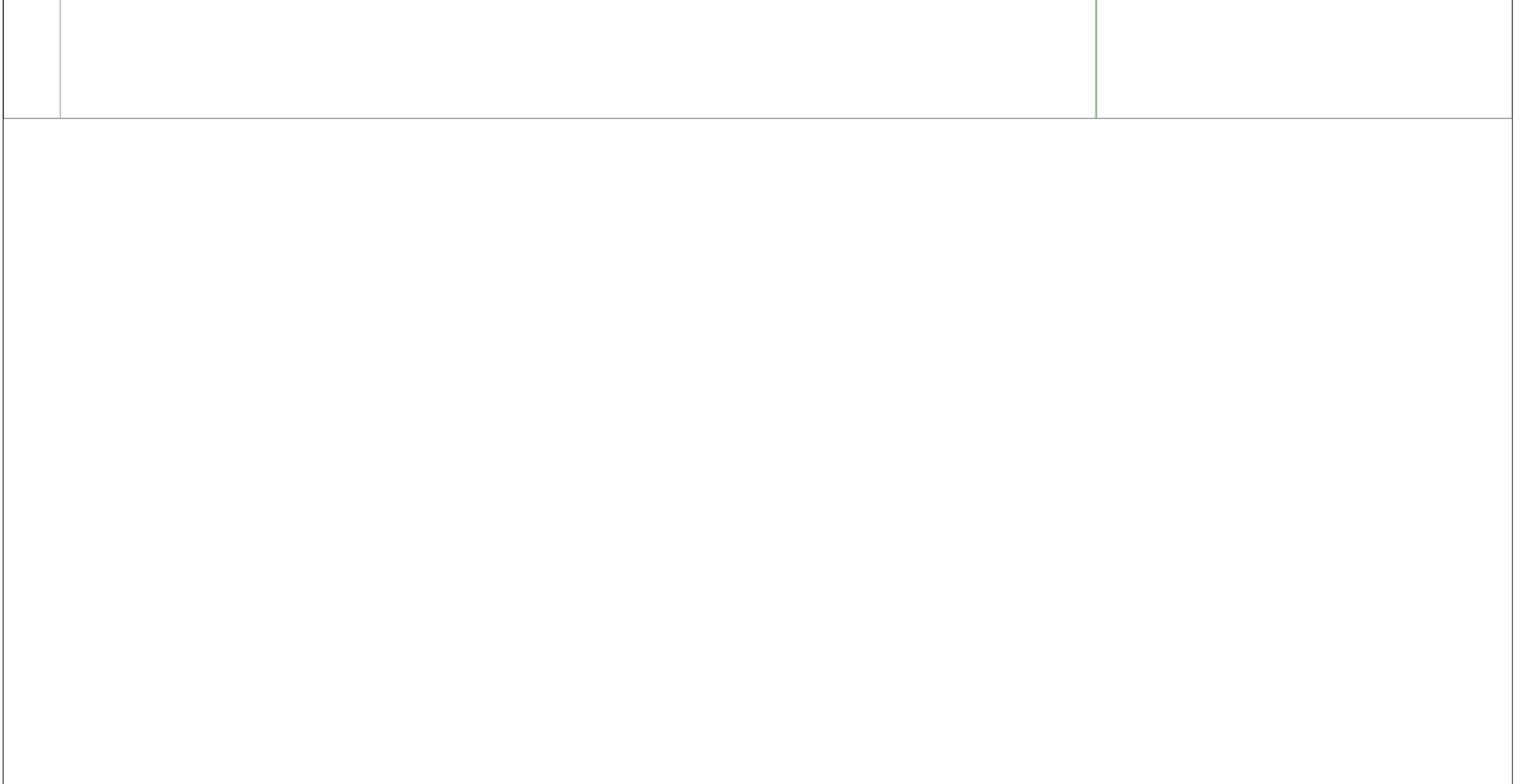
Project: TutorialEE460SeniorPro Date: Wed 11/30/22	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

Jun '22					Jul '22					Aug '22					Sep '22					Oct '22					Nov '22					Dec '22					Jan '23					Feb '23				
22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	5							

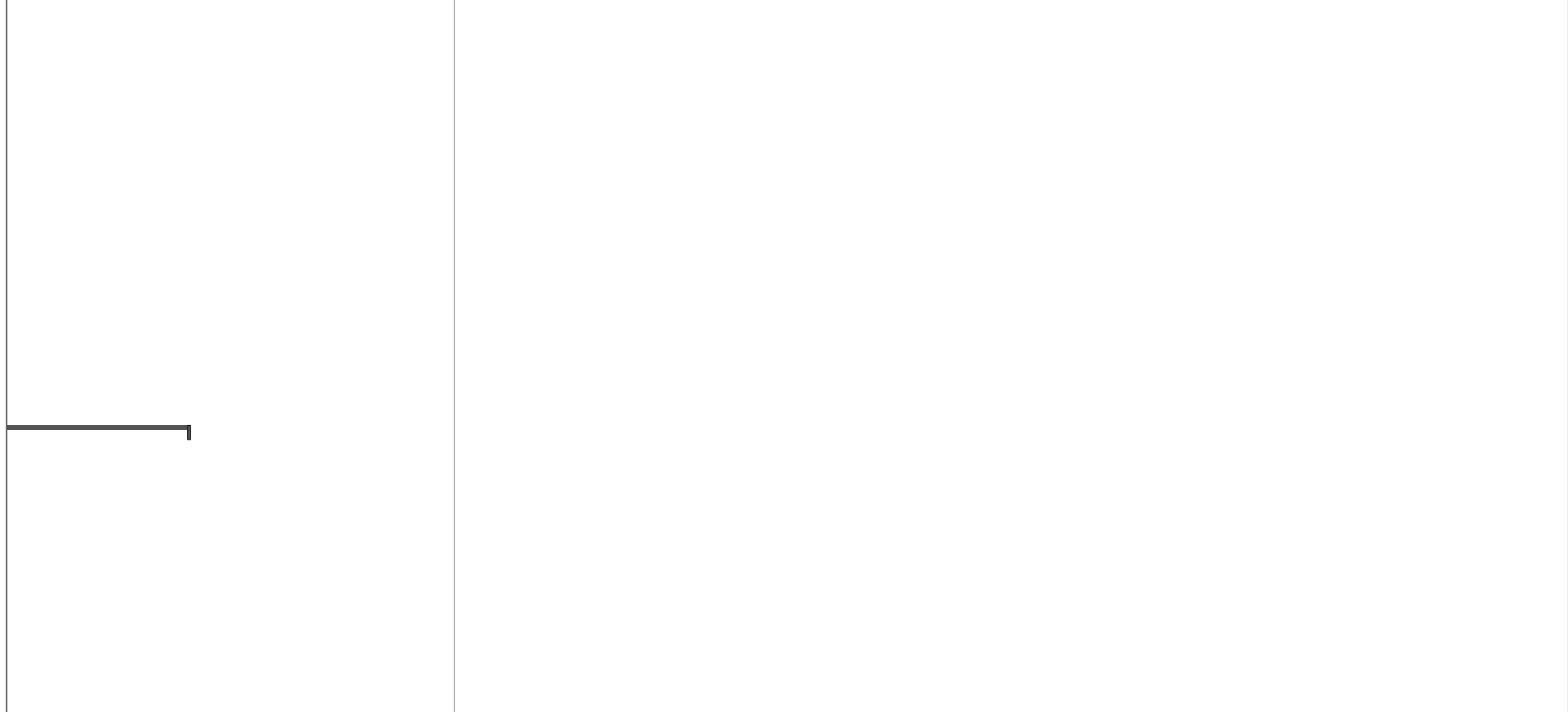


Project: TutorialEE460SeniorPro  
Date: Wed 11/30/22

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

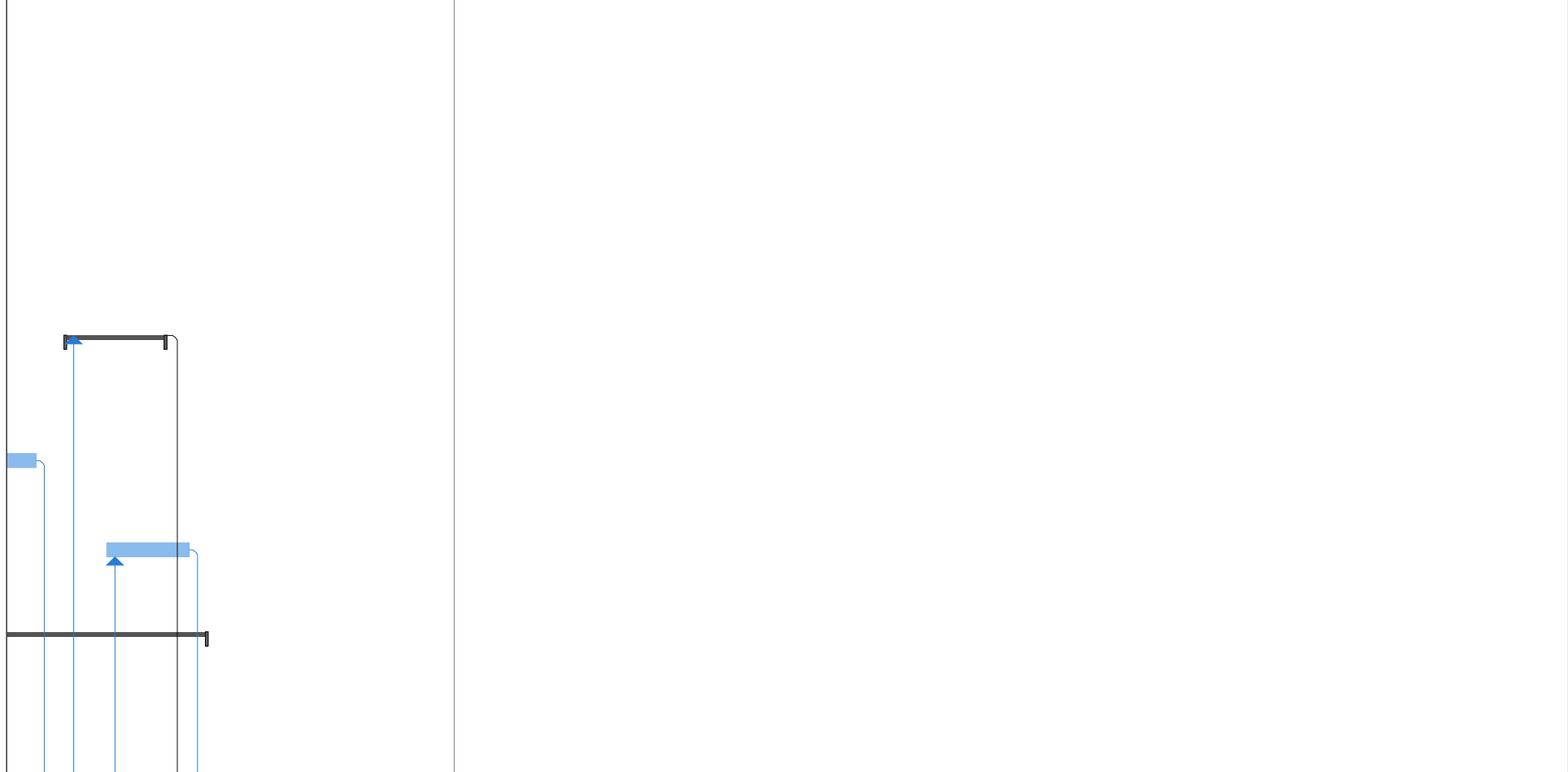


Project: TutorialEE460SeniorPro Date: Wed 11/30/22	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			



Project: TutorialEE460SeniorPro Date: Wed 11/30/22	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

b '23      Mar '23      Apr '23      May '23      Jun '23      Jul '23      Aug '23      Sep '23      Oct '23  
 5 12 19 26 5 12 19 26 2 9 16 23 30 7 14 21 28 4 11 18 25 2 9 16 23 30 6 13 20 27 3 10 17 24 1 8 15 22



Project: TutorialEE460SeniorPro  
 Date: Wed 11/30/22

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			



Project: TutorialEE460SeniorPro Date: Wed 11/30/22	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

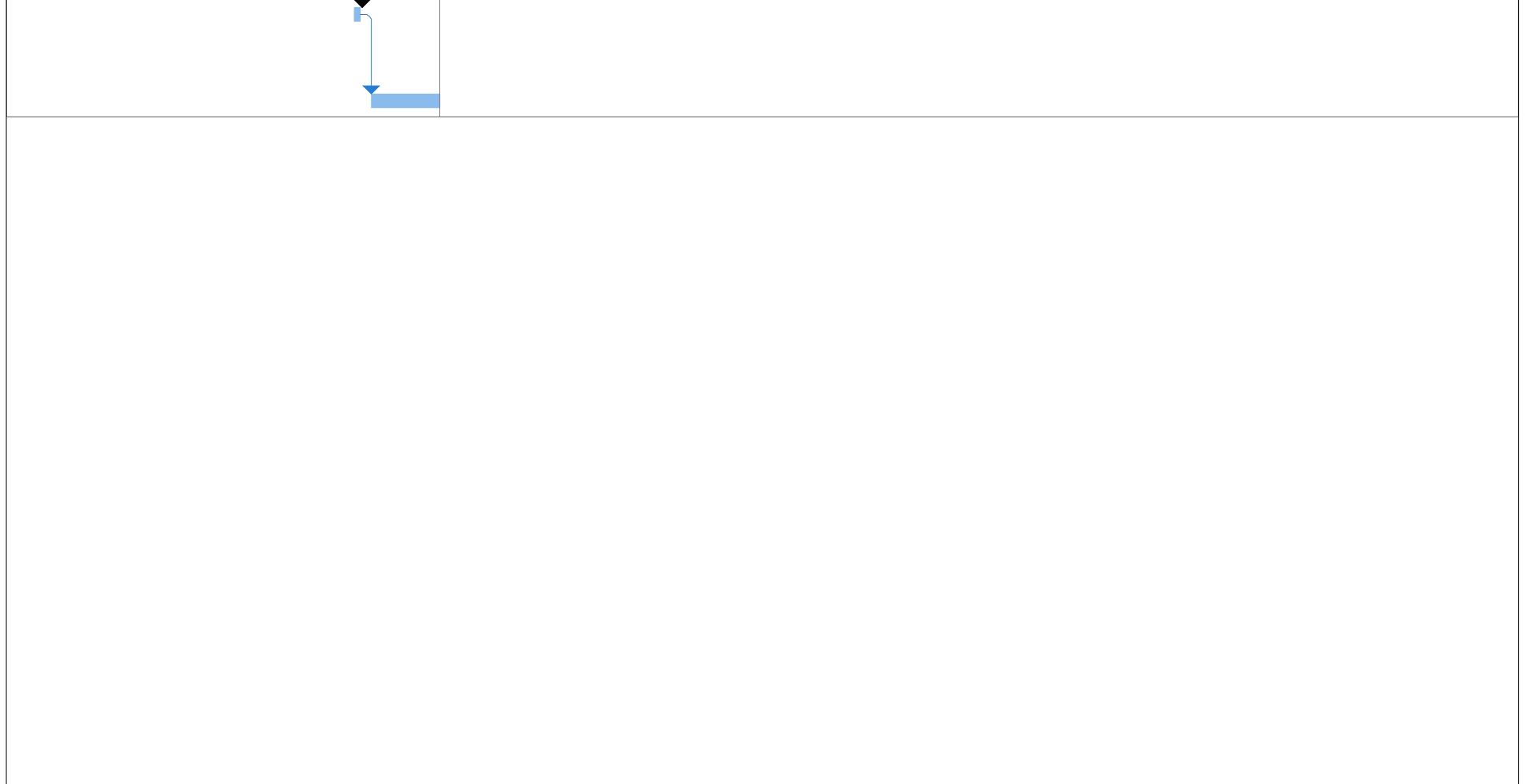
b '23 | Mar '23 | Apr '23 | May '23 | Jun '23 | Jul '23 | Aug '23 | Sep '23 | Oct '23



Project: TutorialEE460SeniorPro  
Date: Wed 11/30/22

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			





Project: TutorialEE460SeniorPro Date: Wed 11/30/22	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

## Appendix E. Schematics

As a reminder, the schematics and layout for the top level design can be viewed interactively in a web browser by following this link: <https://www.flux.ai/markwu2001/vrglovedriverv1?editor=schematic>

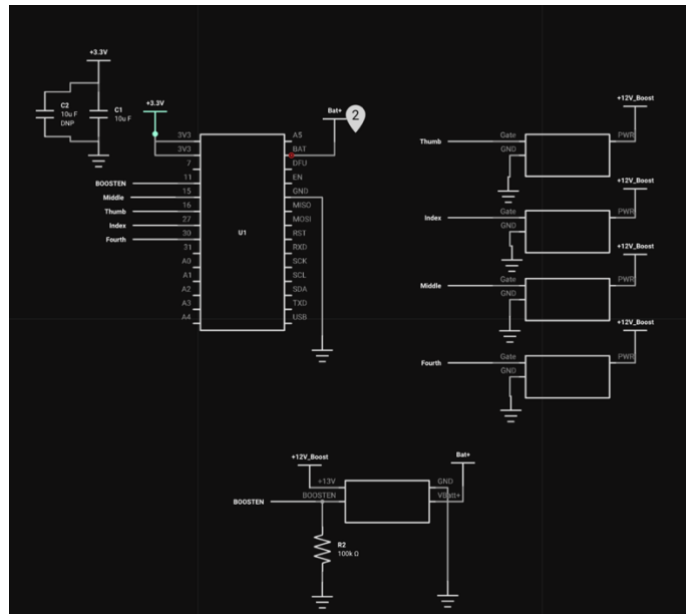


Figure 45. Top level schematic of the first revision of the VR Glove control board

As a reminder, the schematics and layout for the solenoid driver sublayout can be viewed interactively in a web browser by following this link: <https://www.flux.ai/markwu2001/solenoid-fet-driver-3p3v>

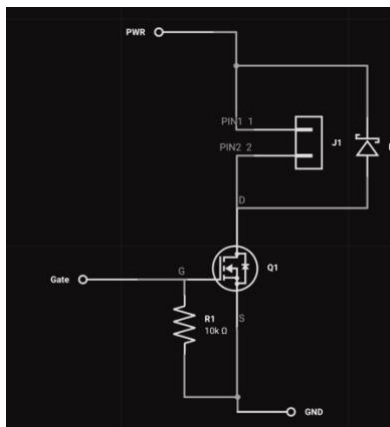


Figure 46. Schematic of Low-Side MOSFET Solenoid Driver Submodule

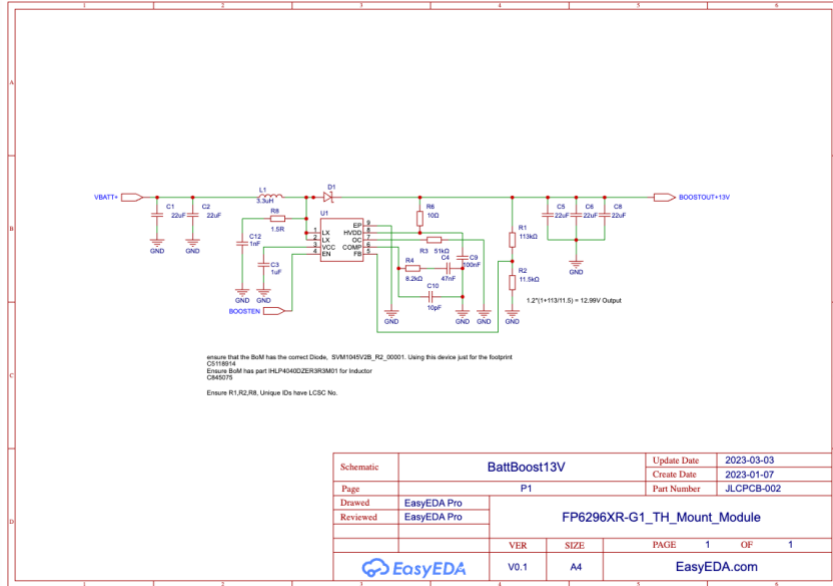


Figure 47. Schematic of DC-DC Boost Converter Submodule

## Appendix F. PCB Layout

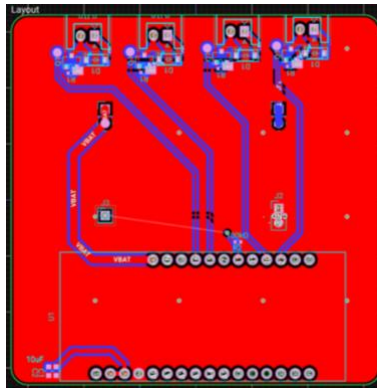


Figure 48. Top copper from layout of main VR Glove controller board

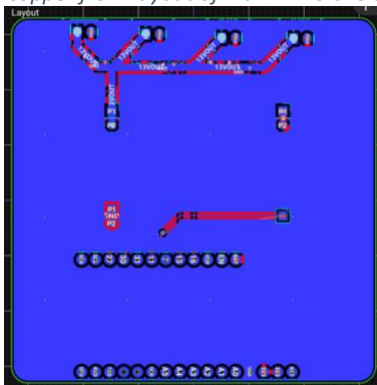


Figure 49. Bottom copper from layout of main VR Glove controller board

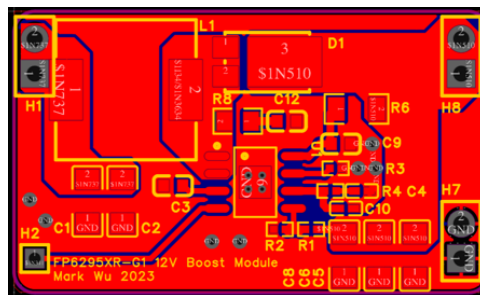


Figure 50. Top copper from layout of boost converter submodule

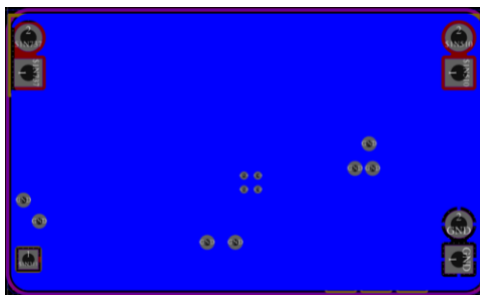


Figure 51. Bottom copper from layout of boost converter submodule

## Appendix G. Firmware (Arduino)

```
#define BOOTEN 10
#define INDEX 5
#define FOURTH 6
#define MIDDLE 12
#define THUMB 13

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(BOOTEN, OUTPUT);
  pinMode(INDEX, OUTPUT);
  pinMode(FOURTH, OUTPUT);
  pinMode(MIDDLE, OUTPUT);
  pinMode(THUMB, OUTPUT);

  digitalWrite(BOOTEN, HIGH); // turn the LED on (HIGH is the voltage
level)
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(INDEX, HIGH);
  digitalWrite(FOURTH, HIGH);
  digitalWrite(MIDDLE, HIGH);
  digitalWrite(THUMB, HIGH);
  digitalWrite(LED_BUILTIN, HIGH);
  delay(800);
  digitalWrite(INDEX, LOW);
  digitalWrite(FOURTH, LOW);
  digitalWrite(MIDDLE, LOW);
  digitalWrite(THUMB, LOW);
  digitalWrite(LED_BUILTIN, LOW);
  delay(3000);
}
```

Figure 52. Sanity check test firmware for electrical subsystem integration testing

```
/*
 * Mark Wu 2023
 * Claire Chen 2023
 * The code below uses, modifies, and adds functions from the API referenced
in the MIT License below
 */
/*****
This is an example for our nRF52 based Bluefruit LE modules
```

Pick one up today in the adafruit shop!

Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

MIT license, check LICENSE for more information  
All text above, and the splash screen below must be included in any redistribution

```
*****/
#include <bluefruit.h>
#include <Adafruit_LittleFS.h>
#include <InternalFileSystem.h>

#define BOOSTEN 10
#define INDEX 5
#define FOURTH 6
#define MIDDLE 12
#define THUMB 13
#define PWMDUTY 200 //PWM duty cycle of each solenoid from 0-255 allows
the designer to fine tune the required power to pinch each finger

// BLE Service
BLEDFU bledfu; // OTA DFU service
BLEDIS bledis; // device information
BLEUART bleuart; // uart over ble
BLEBAS blebas; // battery

void setup()
{
  //initialize all the fingies as output pins
  pinMode(BOOSTEN, OUTPUT);
  pinMode(INDEX, OUTPUT);
  pinMode(FOURTH, OUTPUT);
  pinMode(MIDDLE, OUTPUT);
  pinMode(THUMB, OUTPUT);

  digitalWrite(BOOSTEN, HIGH); //enable the boost converter

  Serial.begin(115200);

#ifdef CFG_DEBUG
  // Blocking wait for connection when debug mode is enabled via IDE
  while ( !Serial ) yield();
#endif

  Serial.println("Bluefruit52 BLEUART Example");
  Serial.println("-----\n");

  // Setup the BLE LED to be enabled on CONNECT
  // Note: This is actually the default behavior, but provided
  // here in case you want to control this LED manually via PIN 19
```

```

Bluefruit.autoConnLed(true);

// Config the peripheral connection with maximum bandwidth
// more SRAM required by SoftDevice
// Note: All config***() function must be called before begin()
Bluefruit.configPrphBandwidth(BANDWIDTH_MAX);

Bluefruit.begin();
Bluefruit.setTxPower(4); // Check bluefruit.h for supported values
//Bluefruit.setName(getMcuUniqueID()); // useful testing with multiple
central connections
Bluefruit.Periph.setConnectCallback(connect_callback);
Bluefruit.Periph.setDisconnectCallback(disconnect_callback);

// To be consistent OTA DFU should be added first if it exists
bledfu.begin();

// Configure and Start Device Information Service
bledis.setManufacturer("Adafruit Industries");
bledis.setModel("Bluefruit Feather52");
bledis.begin();

// Configure and Start BLE Uart Service
bleuart.begin();

// Start BLE Battery Service
blebas.begin();
blebas.write(100);

// Set up and start advertising
startAdv();

Serial.println("Please use Adafruit's Bluefruit LE app to connect in UART
mode");
Serial.println("Once connected, enter character(s) that you wish to
send");
}

void startAdv(void)
{
// Advertising packet

Bluefruit.Advertising.addFlags(BLE_GAP_ADV_FLAGS_LE_ONLY_GENERAL_DISC_MODE);
Bluefruit.Advertising.addTxPower();

// Include bleuart 128-bit uuid
Bluefruit.Advertising.addService(bleuart);

// Secondary Scan Response packet (optional)
// Since there is no room for 'Name' in Advertising packet
Bluefruit.ScanResponse.addName();

/* Start Advertising

```

```

* - Enable auto advertising if disconnected
* - Interval: fast mode = 20 ms, slow mode = 152.5 ms
* - Timeout for fast mode is 30 seconds
* - Start(timeout) with timeout = 0 will advertise forever (until
connected)
*
* For recommended advertising interval
* https://developer.apple.com/library/content/qa/qa1931/\_index.html
*/
Bluefruit.Advertising.restartOnDisconnect(true);
Bluefruit.Advertising.setInterval(32, 244); // in unit of 0.625 ms
Bluefruit.Advertising.setFastTimeout(30); // number of seconds in
fast mode
Bluefruit.Advertising.start(0); // 0 = Don't stop
advertising after n seconds
}

void loop()
{
// Forward data from HW Serial to BLEUART - not needed for VR Glove
// while (Serial.available())
// {
// // Delay to wait for enough input, since we have a limited
transmission buffer
// delay(2);
//
// uint8_t buf[64];
// int count = Serial.readBytes(buf, sizeof(buf));
// bleuart.write( buf, count );
// }

// Forward from BLEUART to HW Serial
while ( bleuart.available() )
{
uint8_t ch;
ch = (uint8_t) bleuart.read();
Serial.write(ch);
//the ch variable represents finger states from the LSB in the following
order:
//1 - INDEX
//10 - FOURTH
//100 - MIDDLE
//1000 - THUMB
decodeCharToSolenoid(ch);

}
}

void decodeCharToSolenoid(uint8_t codedStates)
{
//the codedStates variable represents finger states from the LSB in the
following order:
//1 - INDEX

```



```

//10    - FOURTH
//100   - MIDDLE
//1000  - THUMB
if ((codedStates | 0b1) == 0b1) {
    analogWrite(INDEX ,PWMDUTY);    //analogWrite is Arduino's PWM command
}
if ((codedStates | 0b10) == 0b10) {
    analogWrite(FOURTH ,PWMDUTY);
}
if ((codedStates | 0b100) == 0b100) {
    analogWrite(MIDDLE ,PWMDUTY);
}
if ((codedStates | 0b1000) == 0b1000) {
    analogWrite(THUMB ,PWMDUTY);
}
}

// callback invoked when central connects
void connect_callback(uint16_t conn_handle)
{
    // Get the reference to current connection
    BLEConnection* connection = Bluefruit.Connection(conn_handle);

    char central_name[32] = { 0 };
    connection->getPeerName(central_name, sizeof(central_name));

    Serial.print("Connected to ");
    Serial.println(central_name);
}

/**
 * Callback invoked when a connection is dropped
 * @param conn_handle connection where this event happens
 * @param reason is a BLE_HCI_STATUS_CODE which can be found in ble_hci.h
 */
void disconnect_callback(uint16_t conn_handle, uint8_t reason)
{
    (void) conn_handle;
    (void) reason;

    Serial.println();
    Serial.print("Disconnected, reason = 0x"); Serial.println(reason, HEX);
}

```

Figure 53. VR Glove Main Program Adapted from Arduino bleuart example

## Appendix H. Software (Unity)

```
/*
 * Mark Wu 2023
 * Claire Chen 2023
 * The code below uses and modifies functions from the Bluetooth LE for iOS,
 tvOS, and Android plugin by Shatalmic LLC
 *
 * This is a simple example to show the steps and one possible way of
 * automatically scanning for and connecting to a device to receive
 * notification data from the device.
 */

using UnityEngine;
using UnityEngine.UI;

public class StartingExample : MonoBehaviour
{
    public string DeviceName = "bluefruit";
    //public string ServiceUUID = "A9E90000-194C-4523-A473-5FDF36AA4D20";
    //public string LedUUID = "A9E90001-194C-4523-A473-5FDF36AA4D20";
    //public string ButtonUUID = "A9E90002-194C-4523-A473-5FDF36AA4D20";

    public string ServiceUUID = "6E400001-B5A3-F393-E0A9-E50E24DCCA9E";
    public string TxUUID = "6E400003-B5A3-F393-E0A9-E50E24DCCA9E";
    public string RxUUID = "6E400002-B5A3-F393-E0A9-E50E24DCCA9E";

    enum States
    {
        None,
        Scan,
        ScanRSSI,
        ReadRSSI,
        Connect,
        RequestMTU,
        Subscribe,
        Unsubscribe,
        Disconnect,
    }

    private bool _connected = false;
    private float _timeout = 0f;
    private States _state = States.None;
    private string _deviceAddress;
    private bool _foundRxUUID = false;
    private bool _foundTxUUID = false;
    private bool _rssiOnly = false;
    private int _rssi = 0;
}
```

```

public Text StatusText;
public Text ButtonPositionText;

private string StatusMessage
{
    set
    {
        BluetoothLEHardwareInterface.Log(value);
        StatusText.text = value;
    }
}

void Reset()
{
    _connected = false;
    _timeout = 0f;
    _state = States.None;
    _deviceAddress = null;
    _foundRxUUID = false;
    _foundTxUUID = false;
    _rssi = 0;
}

void SetState(States newState, float timeout)
{
    _state = newState;
    _timeout = timeout;
}

void StartProcess()
{
    Reset();
    BluetoothLEHardwareInterface.Initialize(true, false, () =>
    {
        SetState(States.Scan, 0.1f);

    }, (error) =>
    {
        StatusMessage = "Error during initialize: " + error;
    });

    Debug.Log("Successfully finished initialization.");
}

// Use this for initialization
void Start()
{
    StartProcess();
}

private void ProcessButton(byte[] bytes)

```

```

{
    if (bytes[0] == 0x00)
        ButtonPositionText.text = "Not Pushed";
    else
        ButtonPositionText.text = "Pushed";
}

// Update is called once per frame
void Update()
{
    if (_timeout > 0f)
    {
        _timeout -= Time.deltaTime;
        if (_timeout <= 0f)
        {
            _timeout = 0f;

            switch (_state)
            {
                case States.None:
                    break;

                case States.Scan:
                    StatusMessage = "Scanning for " + DeviceName;

BluetoothLEHardwareInterface.ScanForPeripheralsWithServices(null, (address,
name) =>
    {
        // if your device does not advertise the rssi
and manufacturer specific data
        // then you must use this callback because the
next callback only gets called
        // if you have manufacturer specific data

        if (!_rssiOnly)
        {
            if (name.Contains(DeviceName))
            {
                StatusMessage = "Found " + name;

                // found a device with the name we want
                // this example does not deal with
finding more than one

                _deviceAddress = address;
                SetState(States.Connect, 0.5f);
            }
        }
    }, (address, name, rssi, bytes) =>
    {

```

```

// use this one if the device responses with
manufacturer specific data and the rssi

    if (name.Contains(DeviceName))
    {
        StatusMessage = "Found " + name;

        if (_rssiOnly)
        {
            _rssi = rssi;
        }
        else
        {
            // found a device with the name we want
            // this example does not deal with
            finding more than one
            _deviceAddress = address;
            SetState(States.Connect, 0.5f);
        }
    }

    }, _rssiOnly); // this last setting allows RFduino
to send RSSI without having manufacturer data

    if (_rssiOnly)
        SetState(States.ScanRSSI, 0.5f);
    break;

case States.ScanRSSI:
    break;

case States.ReadRSSI:
    StatusMessage = $"Call Read RSSI";

BluetoothLEHardwareInterface.ReadRSSI(_deviceAddress, (address, rssi) =>
    {
        StatusMessage = $"Read RSSI: {rssi}";
    });

    SetState(States.ReadRSSI, 2f);
    break;

case States.Connect:
    StatusMessage = "Connecting...";

    // set these flags
    _foundRxUUID = false;
    _foundTxUUID = false;

    // note that the first parameter is the address, not
the name. I have not fixed this because
    // of backwards compatibility.

```

```

        // also note that I am not using the first 2
callbacks. If you are not looking for specific characteristics you can use
one of
        // the first 2, but keep in mind that the device
will enumerate everything and so you will want to have a timeout
        // large enough that it will be finished enumerating
before you try to subscribe or do any other operations.

BluetoothLEHardwareInterface.ConnectToPeripheral(_deviceAddress, null, null,
(address, serviceUUID, characteristicUUID) =>
    {
        StatusMessage = "Connected...";

        BluetoothLEHardwareInterface.StopScan();

        if (IsEqual(serviceUUID, ServiceUUID))
        {
            StatusMessage = "Found Service UUID";

            _foundRxUUID = _foundRxUUID ||
IsEqual(characteristicUUID, RxUUID);
            _foundTxUUID = _foundTxUUID ||
IsEqual(characteristicUUID, TxUUID);

            // if we have found both characteristics
that we are waiting for
            // set the state. make sure there is enough
timeout that if the
            // device is still enumerating other
characteristics it finishes
            // before we try to subscribe
            if (_foundRxUUID && _foundTxUUID)
            {
                Debug.Log("Connected to Bluefruit! :D");
                _connected = true;
                SetState(States.RequestMTU, 2f);
            }
        }
    });
    break;

    case States.RequestMTU:
        StatusMessage = "Requesting MTU";

BluetoothLEHardwareInterface.RequestMtu(_deviceAddress, 185, (address,
newMTU) =>
    {
        StatusMessage = "MTU set to " +
newMTU.ToString();

        SetState(States.Subscribe, 0.1f);
    });

```

```

        break;

    case States.Subscribe:
        StatusMessage = "Subscribing to characteristics...";

BluetoothLEHardwareInterface.SubscribeCharacteristicWithDeviceAddress(_deviceAddress, ServiceUUID, RxUUID, (notifyAddress, notifyCharacteristic) =>
    {
        StatusMessage = "Waiting for user action
(1)...";

        _state = States.None;

        // read the initial state of the button

BluetoothLEHardwareInterface.ReadCharacteristic(_deviceAddress, ServiceUUID, RxUUID, (characteristic, bytes) =>
    {
        ProcessButton(bytes);
    });

    SetState(States.ReadRSSI, 1f);

    }, (address, characteristicUUID, bytes) =>
    {
        if (_state != States.None)
        {
            // some devices do not properly send the
notification state change which calls
            // the lambda just above this one so in
those cases we don't have a great way to
            // set the state other than waiting until we
actually got some data back.
            // The esp32 sends the notification above,
but if your device doesn't you would have
            // to send data like pressing the button on
the esp32 as the sketch for this demo
            // would then send data to trigger this.
            StatusMessage = "Waiting for user action
(2)...";

            SetState(States.ReadRSSI, 1f);
        }

        // we received some data from the device
        ProcessButton(bytes);
    });
    break;

    case States.Unsubscribe:

BluetoothLEHardwareInterface.UnSubscribeCharacteristic(_deviceAddress, ServiceUUID, RxUUID, null);

```

```

        SetState(States.Disconnect, 4f);
        break;

    case States.Disconnect:
        StatusMessage = "Commanded disconnect.";

        if (_connected)
        {
BluetoothLEHardwareInterface.DisconnectPeripheral(_deviceAddress, (address)
=>
            {
                StatusMessage = "Device disconnected";
                BluetoothLEHardwareInterface.DeInitialize(()
=>
                    {
                        _connected = false;
                        _state = States.None;
                    });
            });
        }
        else
        {
            BluetoothLEHardwareInterface.DeInitialize(() =>
            {
                _state = States.None;
            });
        }
        break;
    }
}

private bool ledON = false;
public void OnLED()
{
    ledON = !ledON;
    if (ledON)
    {
        SendByte((byte) 0x01);
    }
    else
    {
        SendByte((byte) 0x00);
    }
}

string FullUUID(string uuid)
{
    string fullUUID = uuid;
    if (fullUUID.Length == 4)
        fullUUID = "0000" + uuid + "-0000-1000-8000-00805f9b34fb";
}

```



```

        return fullUUID;
    }

    bool IsEqual(string uuid1, string uuid2)
    {
        if (uuid1.Length == 4)
            uuid1 = FullUUID(uuid1);
        if (uuid2.Length == 4)
            uuid2 = FullUUID(uuid2);

        return (uuid1.ToUpper().Equals(uuid2.ToUpper()));
    }

    void SendByte(byte value)
    {
        byte[] data = { value };
        BluetoothLEHardwareInterface.WriteCharacteristic(_deviceAddress,
ServiceUUID, TxUUID, data, data.Length, true, (characteristicUUID) =>
        {

            BluetoothLEHardwareInterface.Log("Write Succeeded");
        });
    }
}

```

Figure 54. Unity C# script for scanning, connecting, and data handling packets with Bluetooth device

# Appendix I. FMEA

System DFM VR Gloves  
 Subsystem \_\_\_\_\_  
 Component \_\_\_\_\_  
 Design Lead \_\_\_\_\_  
 Core Team \_\_\_\_\_

**Potential Failure Mode and Effects Analysis (Design FMEA)**

Key Date \_\_\_\_\_

FMEA Number \_\_\_\_\_  
 Prepared By Mark Wu  
 FMEA Date ###  
 Revision Date \_\_\_\_\_  
 Page \_\_\_\_\_ of \_\_\_\_\_

Item / Function	Potential Failure Mode(s)	Potential Effect(s) of Failure	Severity	Potential Cause(s)/ Mechanism(s) of Failure	Probability	Current Design Controls	Detect	RPN	Recommended Action(s)	Responsibility & Target Completion Date	Action Results				
											Actions Taken	New Sev	New Occ	New Det	New RPN
Actuator-Finger Linkage	Snapping, Shearing, or destruction of linkage	Complete loss of function in that finger's response	7	Not enough component lifespan testing	7	Stress testing by excessive loading	1	49	Design with a high safety factor						
Battery Pack	Combustion due to high discharge at low charge levels	Loss of Life and bodily harm	10	No battery UVLO protection implementation, user modification	2	Ensuring UVLO functionality by input voltage sweep	1	20	Be wary to implement UVLO on all Ics that support it. Pick Ics with UVLO support						
PCB Assembly	Assembly human errors due to difficulty of assembly	Greater than normal amount of defective products	4	Poor DFM design, not complying with manufacture's capabilities	3	Follow Rules of Thumbs, Check DRCs, Do a couple manufacturing spins	4	48	QA steps or offer a warranty						
Glove Fitment	Not a wide enough range of support for customers sizing	Loss of customers and sustainability of the product	6	Not accounting for various glove sizes in design	5	Sample small batch of users	7	210	When designing, look for glove size range and actuation range						
Firmware	Bugs from human error	Lowered performance, defective product	6	Logical errors when programming	5	Unit and Intergration testing with clear edge case definitions	3	90	Define edge cases before writing functions						
Electrical Design	Longevity Failure due to poor design	Defective product	7	Ics not configured suboptimally, components not rated for operational environment	2	Design with a high safety factor for operational temperatures, vibrations, and other environmental factors	5	70	Check COTS ratings, do not just assume all components have high safety factor						
Moving Motor	Longevity Failure	Defective product	5	Debris and improper maintenance of moving parts	6	Longevity under User Testing	7	210	Factor in user beta testing into manufacturing plan						

Figure 55. Anticipatory FMEA Analysis produced before any hardware was built