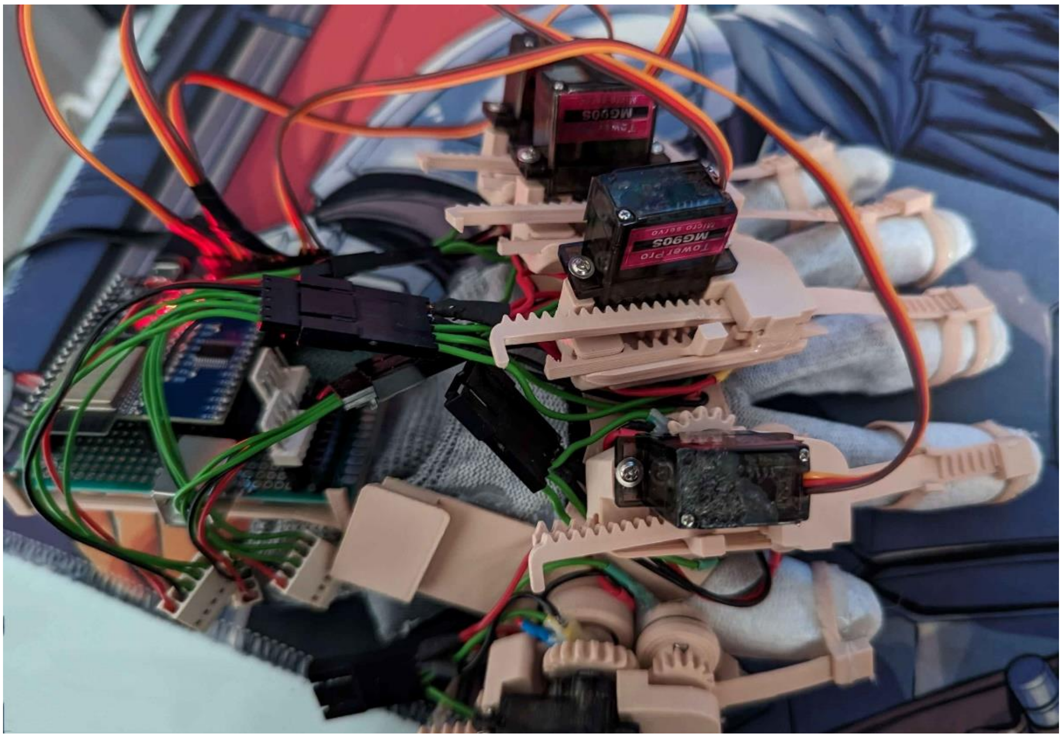
UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

Design and implementation of a haptic glove for use in virtual reality simulators



Student: DIONYSIA KOTSAKI
Registration Number: 18387121

Supervisor

CHARALAMPOS Z. PATRIKAKIS
Professor Dept. of Electrical and Electronics Engineering

ATHENS-EGALEO, SEPTEMBER 2024

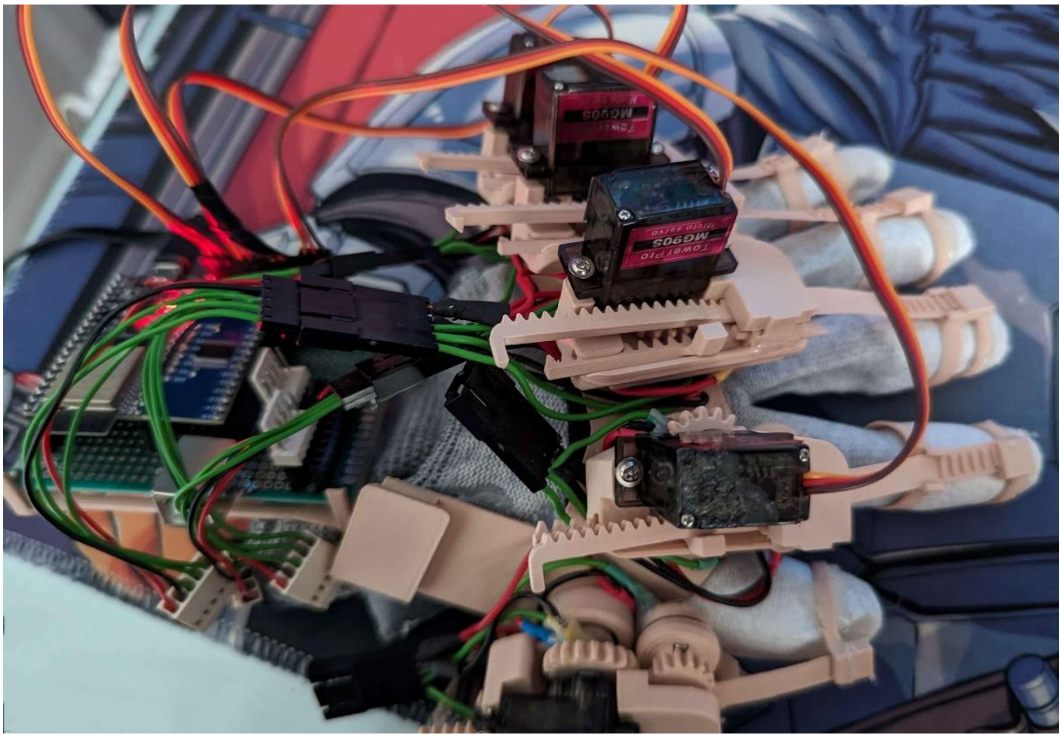
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

Σχεδιασμός και υλοποίηση απτικού γαντιού για χρήση σε προσομοιωτές εικονικής πραγματικότητας



Φοιτητής: ΔΙΟΝΥΣΙΑ ΚΩΤΣΑΚΗ

ΑΜ: 18387121

Επιβλέπων Καθηγητής

ΧΑΡΑΛΑΜΠΟΣ ΠΑΤΡΙΚΑΚΗΣ

Καθηγητής στο Τμήμα Ηλεκτρολόγων Και Ηλεκτρονικών Μηχανικών

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΣΕΠΤΕΜΒΡΙΟΣ 2024

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Χαράλαμπος Πατρικάκης, Καθηγητής	Ξενοφών-Διονύσης Κανδρής, Καθηγητής	Στυλιανός Μυτιληνιαίος, Καθηγητής
 Digitally signed by CHARALAMPOS PATRIKAKIS Date: 2024.10.18 00:56:30 +03'00' (Υπογραφή)	XENOFON DIONYSIOS KANDRIS Digitally signed by XENOFON DIONYSIOS KANDRIS Date: 2024.10.18 19:59:37 +03'00' (Υπογραφή)	Stylianos Mytilinaios Digitally signed by Stylianos Mytilinaios Date: 2024.10.18 16:51:20 +03'00' (Υπογραφή)

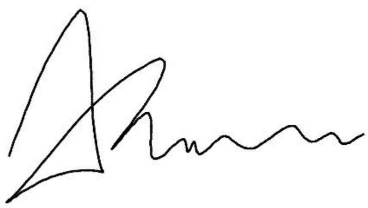
Declaration

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη **Κωτσάκη Διονύσια**, του Σπυρίδωνα, με αριθμό μητρώου eee18387121 φοιτήτρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ, **δηλώνω υπεύθυνα ότι:** «Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου. Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι 28 Σεπτέμβριου του 2024 και έπειτα από αίτηση μου στη Βιβλιοθήκη και έγκριση του επιβλέποντα καθηγητή.»

Ημερομηνία 24 / 09 / 2024

Κωτσάκη Διονύσια



Abstract

Haptic gloves for Virtual Reality (VR) are a new technology that simulates tactile sensations to give users immersive and engaging experiences in virtual environments. These gloves are designed to be worn on the hands and are equipped with sensors and actuators that enable them to detect hand movements and provide haptic feedback. VR haptic gloves can be used in various fields. One of the main uses for haptic gloves is in gaming and entertainment, where they can increase a user's sense of presence and immersion by giving them realistic touch feedback, such as the ability to feel an object's texture or feel the impact of a virtual object. Moreover, Virtual Reality haptic gloves are utilized in training and simulation applications, such as in the training of medical professionals, where they can imitate the feeling of palpating a patient's body or carrying out surgical procedures, or even in the training of rescuers, where they can train in unfavorable and trial some circumstances.

In this thesis, we will explore the creation of the Virtual Reality Haptic Glove and the simulation that the glove will be used on. The majority of VR haptic gloves on the market right now are very expensive and intended for commercial use. The goal is to create a budget-friendly haptic glove that is well-functioning in VR simulations. Furthermore, for the tracking of the fingers and for the haptic feedback, the glove will be using hall effect sensors that will track the differences of magnetic field depending on the motion of the fingers. Additionally, the tracking of the hand will be accomplished by using an Inertial Measurement Unit (IMU), that can capture the motion of the hand by measuring changes in acceleration, angular velocity, and magnetic field strength.

Keywords

Haptic Technology, Virtual Reality, Angular Measurement with hall effect sensors, Finger Tracking Glove, Robotics

Περίληψη

Τα απτικά γάντια για εικονική πραγματικότητα (VR) είναι μια νέα τεχνολογία που προσομοιώνει τις απτικές αισθήσεις για να προσφέρει στους χρήστες καθηλωτικές και συναρπαστικές εμπειρίες σε εικονικά περιβάλλοντα. Αυτά τα γάντια έχουν σχεδιαστεί για να φοριούνται στα χέρια και είναι εξοπλισμένα με αισθητήρες και ενεργοποιητές που τους επιτρέπουν να ανιχνεύουν τις κινήσεις των χεριών και να παρέχουν απτική ανάδραση. Τα απτικά γάντια VR μπορούν να χρησιμοποιηθούν σε διάφορους τομείς. Μία από τις κύριες χρήσεις των απτικών γαντιών είναι στα παιχνίδια και την ψυχαγωγία, όπου μπορούν να αυξήσουν την αίσθηση παρουσίας του χρήστη παρέχοντάς του ρεαλιστική ανατροφοδότηση αφής, όπως η ικανότητα να αισθάνεται την υφή ενός αντικειμένου ή να αισθάνεται την επίδραση ενός εικονικού αντικειμένου. Επιπλέον, τα απτικά γάντια Εικονικής Πραγματικότητας χρησιμοποιούνται σε εφαρμογές εκπαίδευσης και προσομοίωσης, όπως στην εκπαίδευση επαγγελματιών γιατρών, όπου μπορούν να μιμηθούν την αίσθηση της ψηλάφησης του σώματος ενός ασθενούς ή τη διενέργεια χειρουργικών επεμβάσεων, ή ακόμα και στην εκπαίδευση διασώσεων, όπου μπορεί να προπονηθεί σε δυσμενείς και δύσκολες συνθήκες.

Σε αυτή τη διατριβή, θα διερευνήσουμε τη δημιουργία του απτικού γαντιού εικονικής πραγματικότητας και την προσομοίωση στην οποία θα χρησιμοποιηθεί το γάντι. Η πλειονότητα των απτικών γαντιών VR στην αγορά αυτή τη στιγμή είναι πολύ ακριβά και προορίζονται για εμπορική χρήση. Ο στόχος είναι να δημιουργηθεί ένα φιλικό προς τον προϋπολογισμό απτικό γάντι που να λειτουργεί καλά στις προσομοιώσεις VR. Επιπλέον, για την παρακολούθηση των δακτύλων και για την απτική ανάδραση, το γάντι θα χρησιμοποιεί αισθητήρες μαγνητικού πεδίου (hall effect sensor) που θα παρακολουθούν τις διαφορές του μαγνητικού πεδίου ανάλογα με την κίνηση των δακτύλων. Επιπλέον, η παρακολούθηση του χεριού θα επιτευχθεί με τη χρήση μιας Μονάδας Αδρανειακής Μέτρησης (IMU), που μπορεί να καταγράψει την κίνηση του χεριού μετρώντας τις αλλαγές στην επιτάχυνση, τη γωνιακή ταχύτητα και την ένταση του μαγνητικού πεδίου.

Λέξεις-κλειδιά

Απτική Τεχνολογία, Εικονική πραγματικότητα, Γωνιακή Μέτρηση με αισθητήρες Μαγνητικού Πεδίου, Ρομποτική, Γάντι Εντοπισμός Δακτύλων

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Patrikakis Charalampos for his invaluable guidance and support throughout my diploma. His knowledge and encouragement have been extremely helpful to my research and contributed to this thesis's success. I would like to thank also Vasileios Doulgerakis, serving as the Supervisor's assistant for helping me with his knowledge about electronics and 3D printing. He was always available to help me and for your brilliant comments and suggestions.

I am also grateful to my classmates Pavlidis Forestis Petros and Koutoulakis Grigorios, for their endless encouragement and wise counsel as I worked toward my diploma.

Lastly, I would be remiss not to mention my family and friends. Their belief in me has kept my spirits and motivation high during this process. An extraordinary thanks to my cats for all the entertainment and comfort.

I want to express my gratitude to everyone who has helped me during this journey. This thesis would not have been feasible without the assistance and direction you provided.

Table of contents

List of Figures

List of Images

Chapter 1: Introduction

Virtual reality (VR) technology has grown quickly, from simple graphical interfaces to highly immersive environments that mimic real-world experiences. These advancements are being driven by improvements in computing power, graphics rendering, and sensory feedback systems. The inclusion of haptic feedback to Virtual Reality is a significant step forward, allowing users to see, hear, and feel virtual items and settings. Haptic technology is crucial for applications requiring high precision and realism, such as surgical simulators, remote robotic operations, and sophisticated training programs. The haptic Virtual Reality glove produced in this study will be assessed for its ability to provide genuine tactile feedback, its impact on user experience, and possible applications in a variety of sectors. This thesis seeks to expand the capabilities of VR systems by investigating the convergence of virtual reality and haptic technologies, resulting in more meaningful and subtle interactions that bridge the gap between the virtual and physical worlds. This study aims not only to push the frontiers of existing VR technology but also to pave the path for future developments that will incorporate human sensations into virtual settings.

1.1 Object of the Diploma Thesis

This diploma thesis aims to explore the concept, creation, and use of a haptic virtual reality glove that will greatly improve tactile interaction in Virtual Reality settings. In order to improve the realism and immersive experience of Virtual Reality applications, our study will concentrate on the integration of sophisticated haptic feedback systems to deliver accurate and responsive tactile sensations. The project intends to solve several important technological issues, such as improving user comfort and ergonomics, minimizing feedback latency, and optimizing haptic actuator positioning. This thesis aims to add to the scientific knowledge and engineering concepts underpinning haptic feedback in VR systems through a thorough investigation of current haptic technologies and a methodical approach to prototype creation.

1.2 Related Work

In the realm of haptic technology, while numerous innovations have emerged, the LucidVR glove, developed by Lucas, stands out as a particularly noteworthy contribution. Beyond only developing hardware, Lucas also developed a SteamVR driver that connects any custom glove that uses the same encoding method. This driver allows developers and enthusiasts to easily connect their own solutions with popular virtual reality platforms, hence greatly expanding the accessibility and flexibility of haptic gloves within the virtual reality ecosystem. LucidVR's contributions demonstrate how individual creativity may lead to larger applications and improvements in the user experience within virtual worlds, highlighting the potential of open-source and community-driven initiatives in improving virtual reality technology. This thesis seeks to further improve haptic feedback systems and investigate novel uses of haptic technology in virtual reality, building on the groundwork established by LucidVR and related initiatives.

Chapter 2: Virtual Reality and Haptic Technology

2.1 Extended Reality

Technological developments in computers and display have produced new gadgets that can combine elements of the real world into virtual scenarios or overlay digital data on top of the actual environment. Extended realities are these amalgamations of digital and physical settings (Andrews, 2019/03/30). The phrase "extended reality" (XR) technology has garnered a lot of interest lately. It encompasses virtual reality (VR), augmented reality (AR), and mixed reality (MR) technologies more generally.

- Augmented Reality (AR) overlays digital information onto the real world.
- Mixed Reality (MR) enhances the actual world with virtualized material, and enables the virtual content to be aware of and interact with the real world.
- Virtual Reality (VR), which completely submerges individuals in a computer-generated virtual environment. (Stanney, 2021)

Hardware, software, and computational technique developments are driving the fast expansion of XR technology. High-resolution screens, accurate motion tracking, and complex rendering algorithms are features of modern XR systems that enable extremely lifelike and interactive experiences. Furthermore, by combining machine learning and artificial intelligence, XR apps are becoming more responsive and adaptive, offering context-aware and customized experiences. The emphasis on a haptic VR glove in the context of this thesis is especially pertinent to the larger XR scene, but the main focus it will be in Virtual Reality (VR).

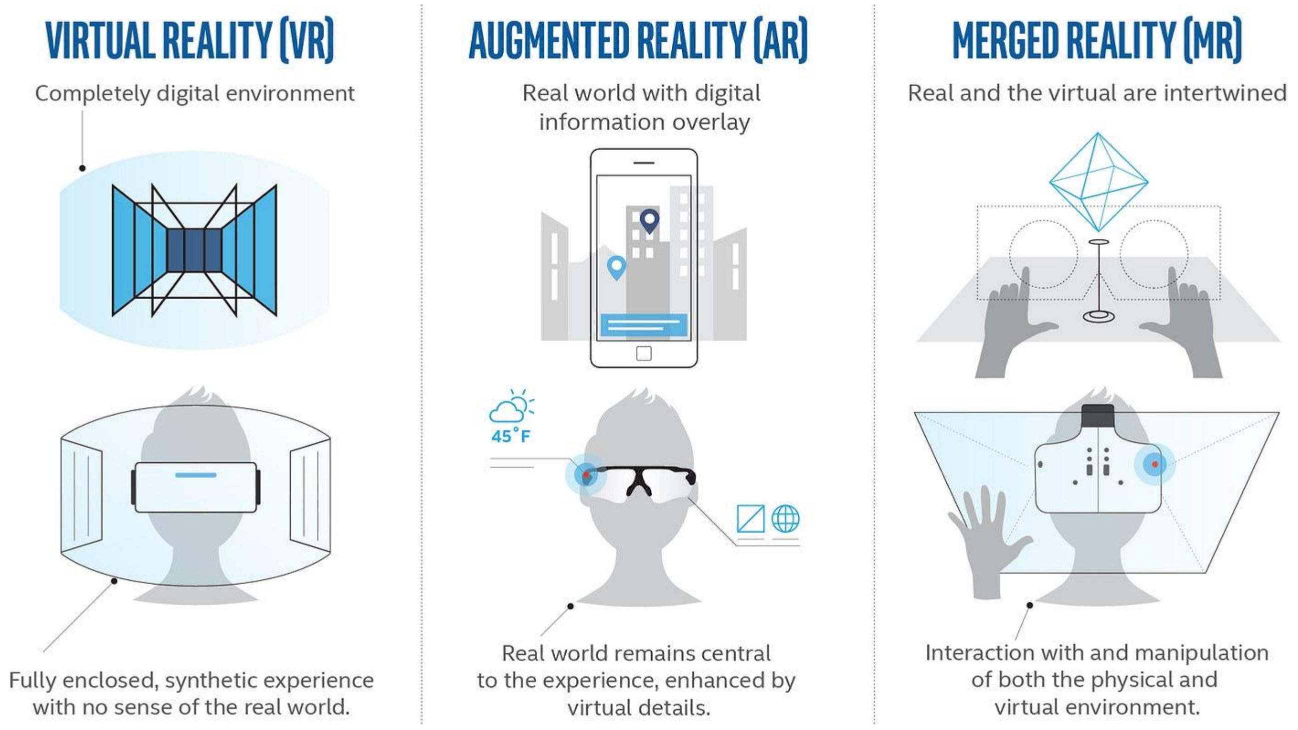


Image 2-1 Extended Reality (Technology, n.d.)

2.2 The World of Virtual Reality

2.2.1 Definition of Virtual Reality

Virtual Reality (VR) is a groundbreaking technological innovation that has captivated the imagination of people worldwide. Through the use of this immersive technology, individuals can interact with and immerse themselves in a computer-simulated, produced environment known as virtual reality. Virtual reality is a computer-generated 3D (three-dimensional) environment with realistic-looking sceneries and items, providing the person a sense of being fully present in a Virtual Reality World. (Robert, 2022)

2.2.2 History of Virtual Reality

The term Virtual Reality first was heard by a philosopher named Immanuel Kant on the eighteenth century, even though it was not used in technology. The comprehension of virtual reality has been greatly aided by Kant's notion. Since then, a vast number of individual technologies supporting this sensory stimulation have emerged, for example, "the Sensorama", a machine that offered multisensory experiences through a combination of visuals, sound, and even smells developed by Morton Heilig in 1957 or the first head-mounted display (HMD) system called the "Sword of Damocles" created by Ivan Sutherland in 1968. (Christoph, 2023)

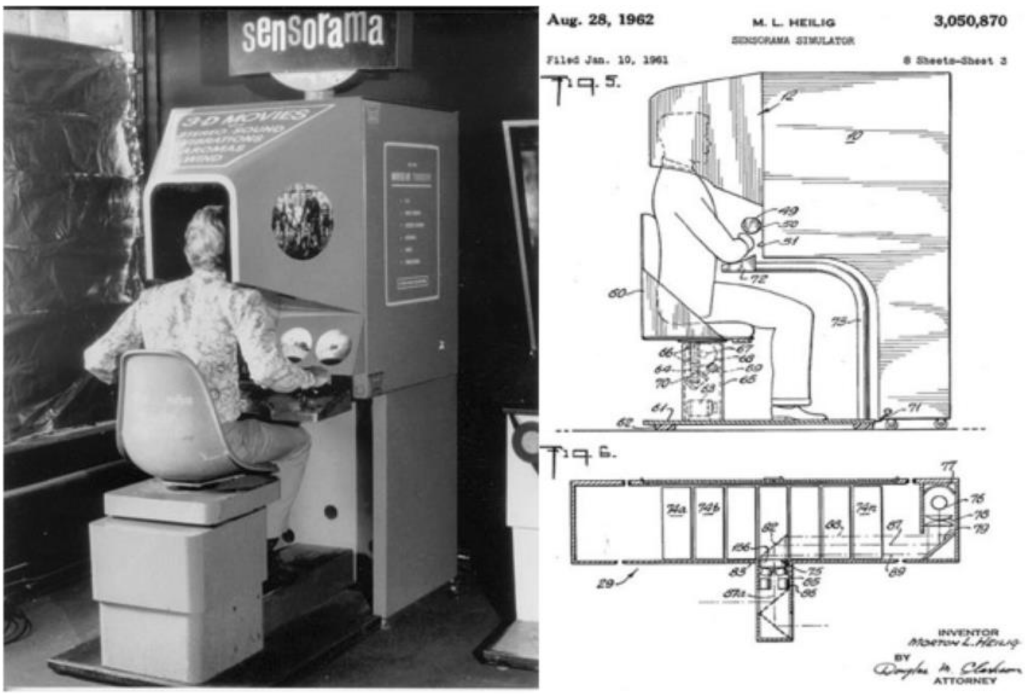


***Image 2-2** Horton Helligs's concept, "the Sensorama",1962 (Parveau & Adda, 2020)*

Image 2-3 Ivan Sutherland's first VR Head Mounted Display, "The Sword of Damocles",1968 (Paro, Bulsara, & Hersh, April 2022)

Although, the modern term Virtual Reality was first coined by Jaron Lanier, a computer scientist and pioneer in the field. (Elaine.T, 2016)He founded VPL Research, one of the first companies to develop and sell VR hardware. Some of the innovative products VPL developed in its time include:

- The Data Glove. Fiber-optic bundles within the fully interactive glove monitored and let users modify and real-time reorient virtual items.

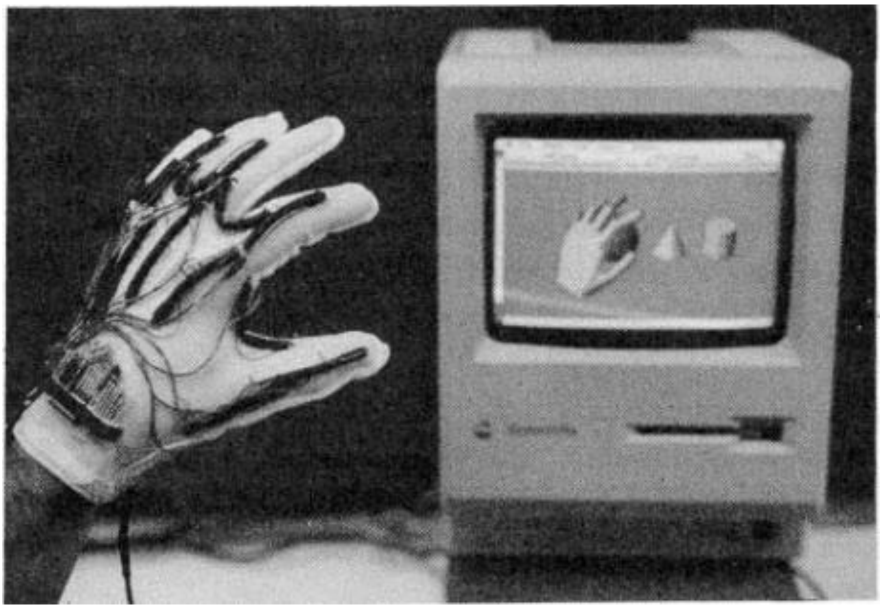


Image 2-4 The Data Glove Gesture Recognition Hardware and Software (Pinto, June 2010)

- The Eyephone. It was a mounted headset that could track user movement and provided immersive simulation.

Image 2-5 The Eyephone Head-Mounted Display (VPL, 2013) and Data Suit (Pape, 1999)

- The Data Suit. a full-body outfit with sensors for measuring the movement of arms, legs, and trunk.

The second wave of Virtual reality technology started in 2012, almost 25 years after the first Virtual Reality Headset. The famous project named Oculus Rift was launched in 2012, with the aim of giving the general public a cheap, high-quality Head-Mounted Display (HMD). Numerous products are being developed and will soon flood the market in the effort to make Virtual reality Device more affordable and efficient. For instance, four years later Valve Corporation released the HTC Vive, featuring room-scale tracking and a hand controller, and Sony also entered the VR market with PlayStation VR, a headset compatible with the PlayStation gaming console (CNET, 2023) (LaValle, 2003).

Image 2-6 Oculus Rift DK1, the first Valve Headset Released on the market (Arth, et al., May 2015)

2.3 Technology Overview

VR technology typically includes headgear and peripherals like as controllers and motion trackers. The technology, which is powered by proprietary downloaded apps or web-based VR, is available through a web browser. Most commonly, hardware for virtual reality includes headgear, hand trackers, treadmills, controllers, and, for content developers, 3D cameras.

2.3.1 Head-Mounted Displays

The most important hardware tool for VR is a headset, which is a device that you wear on your head that covers your eyes. A VR headgear shows visuals and noises that mirror your head motions, giving the sense of being in a different place. Head-mounted displays (HMD) are 3D interactive displays that operate in real-time and allow users to move their heads freely.

An HMD is made up of an optical system that views a modulated light source with driving electronics through each eye (Rolland & Hong, 4 Mar 2024). These displays are strategically placed within the headset, often employing cutting-edge technologies like OLED (Organic Light-Emitting Diode) or liquid-crystal displays (LCD), delivering high-resolution visuals aimed at minimizing the "screen door effect". The housing and optical system are attached to the user's head using a headband or a helmet. Moreover, visual coupling systems, also known as trackers, must be used to provide users with suitable views based on their head position and perhaps gaze point. Connectivity choices, whether tethered to external devices or freestanding, as well as integrated audio solutions, contribute to the adaptability of HMDs. With constant advances, these headsets aim to provide increasingly realistic and pleasant VR experiences (Takashi, Apr. 2002).

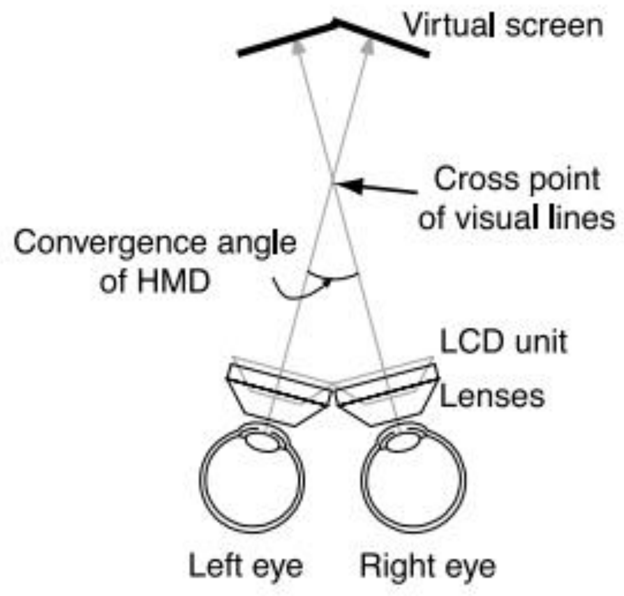


Image 2-7 Relationship between virtual screen and Head-Mounted Display (Takashi, Apr. 2002)

Modern VR headgear may be broadly divided into two categories: tethered and standalone. Standalone VR systems, unlike tethered ones, have all the necessary computational hardware to function in VR without requiring extra gear. Standalone VR headsets use graphics processing units to create a three-dimensional virtual environment on displays. They also have embedded sensors that record the device's position and orientation in space, allowing for accurate user perspective when displaying images. One of the most important drawbacks of this type of VR headsets is that the performance is restricted by the processing power of their hardware. As a result, the inability to handle demanding applications on high-performance platforms and the limited operating time due to the battery's capacity. Tethered VR HMDs are still the most popular option today for these reasons. VR HMDs don't contain any computing hardware except sensors and need a separate computer to function. Since the processing power of the computers utilized with tethered headsets is the sole constraint to their capability, these headsets may operate in more complex and sophisticated virtual reality settings. However, new hybrid devices, such as the Oculus Quest, offer both tethered and standalone modes, allowing users to use the headset in their preferred mode (Angelov, June 2020).

Image 2-8 Relationship between virtual screen and Head-Mounted Display (Takashi, Apr. 2002)

2.3.2 Position Tracking

Tracking handheld controllers in 3D space is essential for interacting with head-mounted displays in VR and AR systems. Handheld controllers include benefits such as tactile buttons, haptic feedback, and accurate tracking. Controller tracking is categorized as either 3-DoF (degrees of freedom) rotational tracking (i.e., roll, pitch, yaw) or 6-DoF rotational and positional tracking (i.e., roll, pitch, yaw, x, y, z). With limited support for positional tracking, mobile systems like the Samsung Gear VR and Google Daydream employ a tiny handheld controller that depends on 3-DoF inertial orientation tracking. Rotational tracking (3DoF) is always done with microscopic electromechanical gyroscopes. On the other hand, desktop VR systems like the Oculus Rift, use a larger controller with external light emitting diodes (LEDs) for 6-DoF optical tracking. Cameras positioned across the surroundings are used to track your body and move around the room.

If the Controllers have 6 degrees of freedom, the user has the ability to communicate directly with virtual objects and move their hands from top to bottom and front to back, allowing them to do all of the moves that they would in the real world (Whitmire, June 2019).

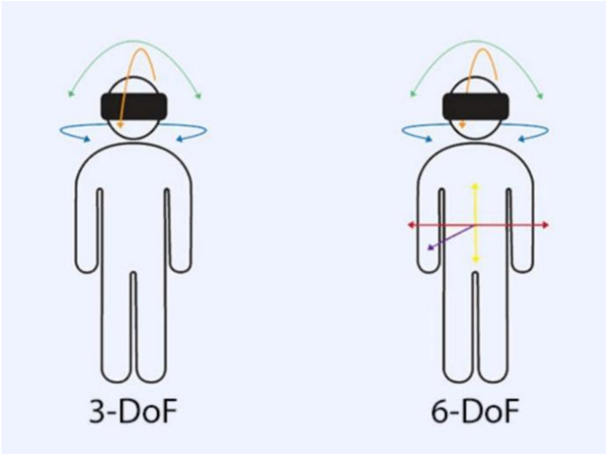


Image 2-9 Relationship between virtual screen and Head-Mounted Display (Takashi, Apr. 2002)

The tracking of the hand can be most common with a controller. A controller is a low-cost hand-gesture-sensing device that is attached to a VR headset allowing them to track the position of the hand and interact with virtual reality objects (Khundam, September 2021).

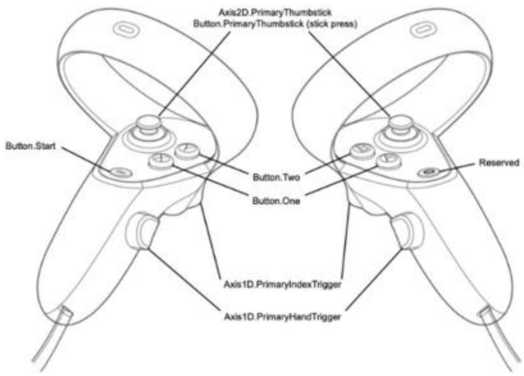


Image 2-10 Oculus Quest controller mapping (Takashi, Apr. 2002, pp. 57-64)

2.4 Usage of Virtual Reality

Nowadays, Virtual Reality Devices have become more mainstream as they have begun to gain purposes other than gaming. It became widely utilized for training, simulations, and immersive experiences in industries including healthcare, education, architecture, and business.

2.4.1 Education

Virtual reality technology has been applied in education to enhance student performance, and VR is being utilized as a teaching aid in many schools. Prior research has indicated a beneficial correlation between virtual reality usage and educational outcomes. For instance, Domingo & Bradley (2017) discovered that virtual reality can offer a more effective learning environment. Alfalah (2018) discovered that virtual reality brings about a more appealing teaching process. These illustrations highlight the potential of VR and AR technologies in encouraging students to take an active role in their education. Furthermore, a variety of methods may be used to enhance learning results. In a mathematics course, game-based learning demonstrated a noticeably greater improvement than the standard method (Chen, Zou, Cheng, & Xie, 2020). Just as important, based on a study of two universities in Thailand that provided data through online questionnaires (using the multi-stage cluster random sampling method), the results show that being comfortable with VR usage improves learning performance both directly and indirectly by making studying in class easier and creating motivation for students to learn (Tirasawasdichai & Pookayaporn, 2021) (Jiao, 2011).

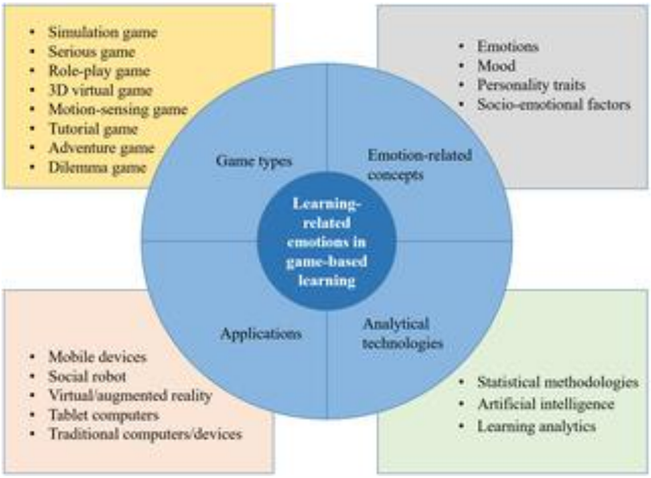


Figure 1 Xieling Chen Digital game-based learning research (Xie, Zou, Chen, Kohnke, & Xie, July 2021)

2.4.2 Healthcare

The use of virtual reality technology in medical education and practice is growing. It has also resulted in a plethora of fresh learning possibilities, for example in surgery simulation and skill training. According to an industry ARC research analysis, the VR and AR healthcare market will be worth 2.54 billion US dollars by 2020, with applications ranging from surgery to medical rehabilitation, consultation, diagnosis, and teaching and training. VR and AR are rapidly being employed in the medical profession as VR surgical simulation systems that provide surgical training to trainee or inexperienced surgeons. They can simulate the realism of the actual surgery and lower the occurrence of errors during future operations.

Another usage of Virtual reality in healthcare is VR exposure therapy. VR technology brings the real environment into people's minds and is able to heal people's anxiety and fear such as Acrophobia, claustrophobia, and social anxiety through VR "exposure therapy".

Last but not least, can help medical students understand more about the anatomy of the human body. An app called Anatomy 4D was created to help educate anatomy by displaying the intricate structures of every bodily component, including the head and neck, chest, belly, pelvis, joints, and other areas (Hsieh & Lee, 2017).

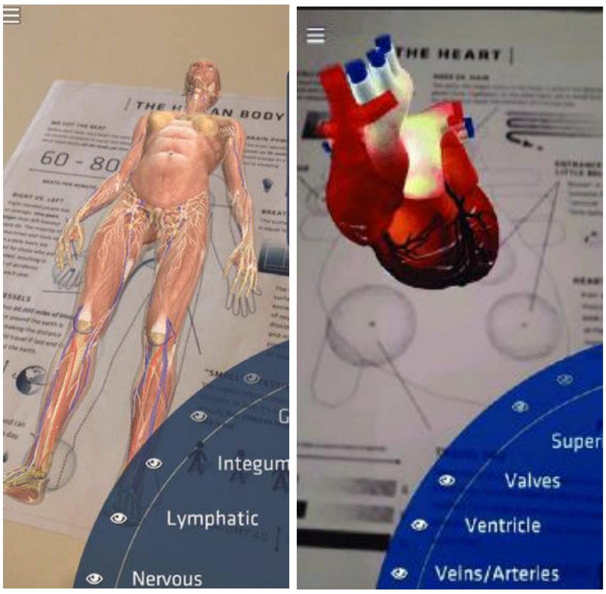


Image 2-11 Anatomy 4D App human anatomy AR display (Khan, Johnston, & Ophoff, 2019)

2.4.3 Entertainment

The development of virtual reality technology has improved people's access to entertainment. Individuals can fully immerse themselves in the virtual world while shopping, traveling, playing games, and watching movies. The virtual reality community has developed over the last 20 years by synthesizing previous work in interactive 3D graphics, visual simulation, and user interfaces. This boosted the number of people working in 3D, and allowed developers to construct a more open technology than the visual simulation community could. Whether creating a virtual reality environment or a game, developers try to create the illusion that they have entered a virtual world (Zyda, September 2005). Virtual reality in gaming is reshaping player experiences, turning static gameplay into interactive journeys. Also, Virtual Reality Art Gallery are changing how people experience art galleries. The Louvre has featured VR tours, with "The Mona Lisa: Beyond the Glass" experience being just one of the options offered to VR enthusiasts. Moreover, the speed of roller coasters combined with the fantastical aspects of virtual worlds is what makes theme parks' adoption of VR so exciting. The hyper-realistic attractions of "The VOID" serve as an example of this (Tech-Mag, n.d.).

Image 2-12 "Mona Lisa Beyond the Glass": the Louvre's first Virtual Reality experience (Louvre, 2019)

Image 2-13 The VOID, Virtual Reality Theme Park (VR, 2016)

2.5 Haptic Technology

Haptics is the study of touch-based sensing and manipulation. The definition of haptics encompasses all elements of gathering data and manipulating objects with touch by humans, by robots, or by a mix of the two in real, virtual, or teleoperated settings. Three categories can be applied to haptics:

- Human Haptics, the investigation of human touch perception and manipulation,
- Machine Haptics, the design, construction, and use of machines to replace or increase human touch.
- Computer Haptics, software, and algorithms used to create and portray virtual objects' tactile qualities (Srinivasan).

Applications for haptic devices are numerous and include teleoperated robotic surgery, rehabilitation, and medical education. The subsequent segment delves into the many haptic devices that are employed for micromanipulation, including wearable technologies, medical procedures and surgery, and teleoperation duties.

2.5.1 Wearable Virtual Reality Haptic Devices

With portability and wearability in mind, several sectors have begun to build and develop haptic devices over time. Better human-interface interaction, collaboration, and communication are made possible by these wearable haptic devices. A wearable haptic device called SenseGlove Nova Blue is intended to provide a very engaging tactile experience in virtual settings. The SenseGlove Nova Blue allows users to engage and feel with virtual items much like they would with actual ones since it was designed with an emphasis on ergonomic design and precise force feedback. The Nova can simulate realistic button clicks, vibrations, and impacts thanks to senseGlove's sophisticated voice coil actuator technology. These features are essential for virtual reality training with tools and dashboards. The voice coil actuator is housed in the glove's hub, whereas the vibrotactile actuators of the thumb and index finger are independent. (Sense Glove, n.d.) The price range of the glove is around 4500$, therefore is not meant for the most daily consumers. Nevertheless, the SenseGlove Nova Blue is only affordable for a select group of people, mostly professionals and businesses where accurate haptic feedback is crucial, due to its complex design and advanced technology.

Image 2-14 Sense Glove Nova (Sense Glove, n.d.)

Another device worth mentioning is the Weart TouchDIVER. In contrast to conventional virtual reality gloves, the TouchDIVER is a small, wearable gadget that can be fastened to the user's fingers to provide accurate tactile input without adding bulk to the glove. The TouchDIVER is a smaller and more covert gadget than the SenseGlove Nova, which is a complete haptic glove that covers the entire hand and delivers force feedback via a flexible exoskeleton. It is intended to be worn on the fingertips, providing accurate tactile input precisely where it is needed. Unlike the SenseGlove Nova, which gives extensive force feedback to imitate the experience of grasping and manipulating items in a virtual environment, the TouchDIVER uses mechanical actuators and thermal feedback to simulate touch, texture, and warmth. This method enables a different type of haptic experience.

Also, the glove contains an SDK (software development kit) for integration with major programming platforms like Unity and Unreal Engine, and it is compatible with different controllers. (weart, n.d.) However, the Weart TouchDIVER, like the SenseGlove NovaIt, is expensive with a price of roughly 3800$ due to its superior technology and specific design.

Image 2-15 Weart TouchDIVER (Knoxlabs, n.d.)

Chapter 3: From Theory to Practice

3.1 Purpose of the Application

The goal of this project is to create an inexpensive and accessible haptic VR glove that is particularly built for the daily consumer, bridging the gap between high-end, professional-grade haptic devices and consumer products. While current options, such as the SenseGlove Nova and Weart TouchDIVER, provide superior tactile feedback and immersive experiences, their expensive prices and specialized designs make them unavailable to the typical user. This project attempts to produce a haptic VR glove that provides an engaging and immersive experience without being prohibitively expensive. The objective is to make haptic feedback technology accessible to ordinary VR aficionados by concentrating on low-cost materials, simple design, and streamlined functionality, hence improving gaming, entertainment, and educational applications in the home.

3.2 Methodology

The process of developing a haptic VR glove for home usage includes design, component and tool selection, and production. This chapter explains the methods and techniques used to develop a practical, cost-effective, and user-friendly haptic glove, with a focus on accurately detecting finger placements and providing immersive haptic output.

3.2.1 Design and Prototyping

The first step in the design process is to conceptualize the glove's form factor and make sure it is comfortable, ergonomic, and long-lasting. In order to create a glove that is both lightweight and durable while being reasonably priced to manufacture, the design must strike a balance between utility and cost. A comprehensive analysis of previous haptic glove projects including the LucidVR glove, is carried out to expedite the design process. The design of LucidVR is an invaluable resource that offers insights into useful design components and user interface considerations. The LucidVR Prototype 3 and 4 used potentiometers due to their simplicity and low cost. Attaching strings to your fingertips using 3D-printed rings and finger caps allows you to quantify how far a string is pulled, which may be translated into fist clenches and finger extensions. In this case, the potentiometers vary their electrical resistance dependent on how far the knob is turned, allowing them to be used to map resistances to string pulls induced by finger motions (Sertoglu, 2021). An Arduino Nano is needed for each hand, and since it already has the requisite number of analog pins, no further circuitry is needed (Rust, n.d.). However, this method resulted in a system that was both big and heavy. The potentiometers had lower accuracy in dynamic settings, resulting in less dependable tracking. For the reasons above, Prototype 5 considerably increased tracking accuracy and operational efficiency over its potentiometers.

Image 3-1 LucidVR Glove Prototype3 and 4 with potentiometers (LucidVR, Lucidglove, n.d.)

Prototype 5 was deployed in the beta phase, where its performance was assessed, and useful insights were gathered. Building on these discoveries is critical for furthering the project's success. The purpose is to improve the present design by fixing any identified constraints and introducing new enhancements. The physical design was kept unchanged, but considerable changes were made to the software. All prototypes of The LucidVR project are entirely open-source, and all the manufacturing instructions as well as the different STL (stereolithography-3D printing technology) files to print are published on GitHub (LucidVR, n.d.).

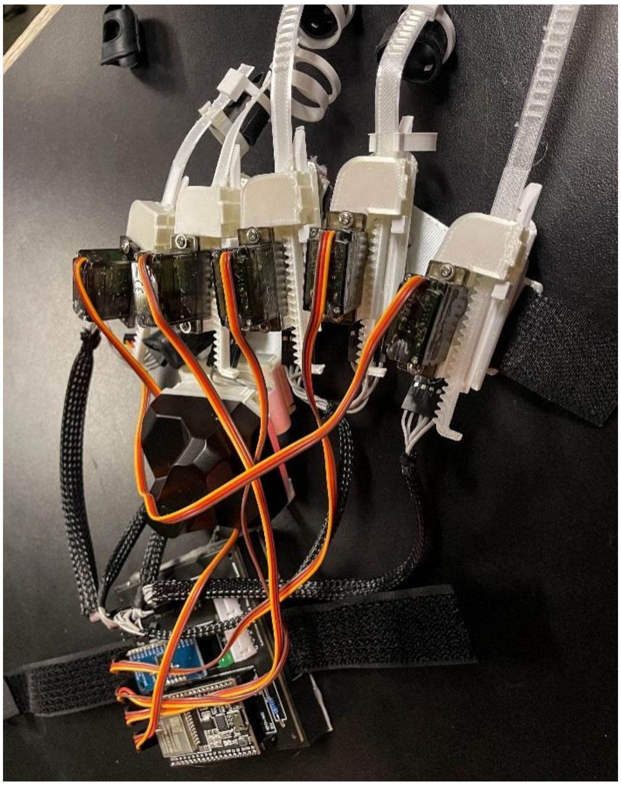


Image 3-2 LucidVR Prototype 5 with hall effect Sensors (LucidVR, Lucidglove, n.d.)

3.2.2 Component Selection

The most important electrical parts of the haptic glove are the actuators provide haptic input, sensors for tracking the finger position, and microcontrollers that interpret sensor data.

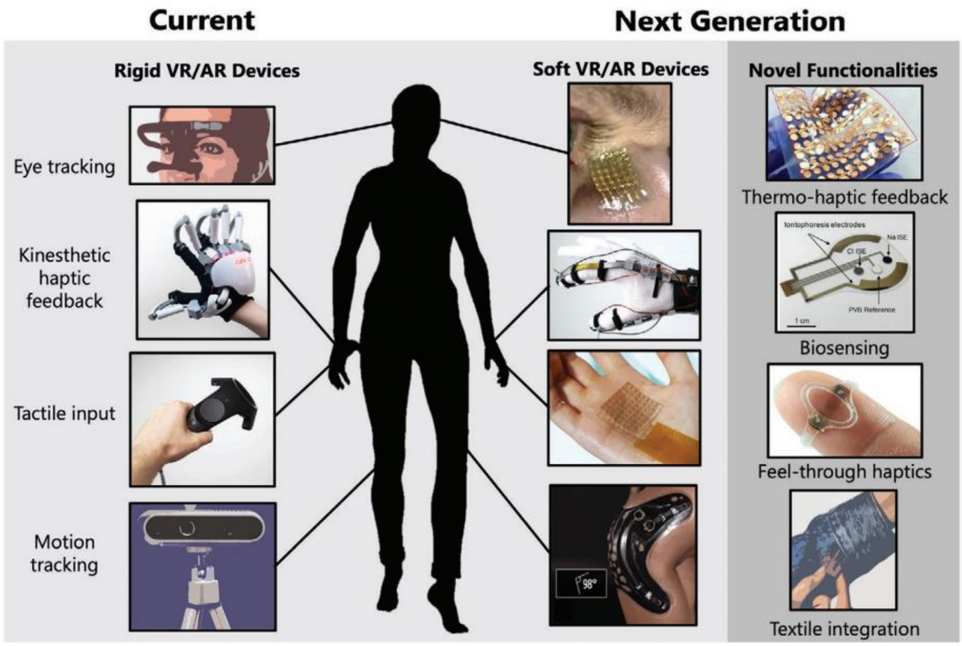


Figure 2 VR/AR devices for tracking and haptics (Jessica, Ronan, Herbert, & Carmel, 2020)

❖ **Finger Tracking**

Choosing the right sensor technology throughout the development of a finger-tracking glove is essential to obtaining precise and economical tracking performance. Some of the methods to measure the angle of the finger are with Hall effect sensors, potentiometers, flex sensors or bend sensors, inertial measurement units (IMUs), and camera-based systems.

- **Hall effect sensors** are a transducer that reacts to a magnetic field by changing its output voltage. They provide precise angle measurements at a reasonable cost while being lightweight and small in size, striking a good compromise between performance and cost and making them perfect for wearable devices. One significant shortcoming of Hall effect sensors is their sensitivity to interference from external magnetic fields.
- **Potentiometers** are adjustable resistors with a wiper that glides over a resistive strip to raise or decrease resistance. Although they are affordable, potentiometers are quite big and undergo mechanical wear over time, which can reduce tracking accuracy.
- **Flex Sensors** (or bend Sensors) are ultra-thin and flexible printed circuits that can measure the amount of deflection or bending and readily incorporate into force-measuring applications (Tekscan, n.d.). They detect angle shifts with excellent precision, but their high cost and requirement for integration into a flexible form factor may render them unsuitable for general usage in consumer-grade applications.

- **Inertial Measurement Units** (IMUs) are electronic devices that can measure a moving object's orientation, velocity, and acceleration. IMUs can give comprehensive motion data, but their complexity, increased cost, and the requirement for specialized data fusion techniques to effectively interpret the sensor data all add to the overall system's complexity.
- **Camera-Based Systems** use visual tracking that precisely recreates the movement of a real-world camera within a 3D digital environment. These systems analyze visual input using image processing techniques, which may be quite accurate but need a lot of computing power and real-time capabilities. Moreover, performance can also be impacted by external variables like occlusions and lighting conditions.

Among all those methods, Hall effect sensors provide the best value in terms of cost, precision, and portability. Another noticeable reason is that they are a well-supported option for incorporation into a finger-tracking glove because of this substantial study, which offers a solid foundation for comprehending their performance characteristics and improving their use.

- **Haptic Input**

  Modern user interfaces (UI) include feedback as a fundamental component that makes it possible to interact with machines more quickly, accurately, and intuitively**.** The are several technologies for the proper implementation of a haptic system that may simulate sensations like force, texture, or pressure. The main techniques for incorporating haptics into VR gloves are listed below, each having advantages and disadvantages.

  - **Vibrotactile Feedback** is the most common use of haptics in VR technology. They are often in the form of small vibrating motors that produce distinct vibration patterns that simulate the sense of touching or gripping an object. Those actuators can be really lightweight and energy-efficient, but they cannot provide any detailed sensation as pressure, resistance or complex textures
  - **Force Feedback** (Exoskeleton or Tendon-Based Systems) is designed to imitate the actual resistance felt while interacting with items. These devices imitate the feeling of grasping or pushing items by restricting the movement of the user's hand or fingers via mechanical actuators. These systems can offer a realistic sense of resistance, simulating weight, stiffness, and hardness, however, they are bulky and power-hungry.
  - **Thermal Feedback** refers to the utilization of heat or cold to provide a tactile sense. The glove incorporates thermoelectric modules, like Peltier components, to provide localized heating or cooling effects. Thermal feedback has limited applications and have lower response time compared to force or vibration feedback (SenseGlove, 2023).

There are a lot more methods to simulate haptic for a glove in a VR environment, for example with Electroactive Polymers, Pneumatic Systems, or even with Ultrasonic Haptics, but they are complex and there is not a lot of research behind them. For applications requiring precise and tactile contact with virtual worlds, force feedback offers the highest level of realism and immersion, making it the best choice for research and development into a VR glove, in contrast to more basic haptic techniques like vibrotactile feedback, which can merely replicate surface textures and vibrations.

- **Microcontroller**

  The choice of microcontroller board is of essential importance since it forms the heart of the system and influences both the functioning and the overall design of the project. To guarantee that the VR glove stays lightweight and comfortable for prolonged usage, the design called for a small board. Moreover, the board had to have integrated Bluetooth capabilities to provide wireless connection without requiring extra, bulky components. A significant quantity of input/output (I/O) pins was required in order to interact with the several actuators and sensors that power the haptic feedback system of the glove. The board's cost was also a major concern because of the limited funds, which required a solution that offered these characteristics without going overboard. The choice where between ESP-WROOM-32 and Arduino Nano BLE 33. Both boards are equipped with Bluetooth and Wi-Fi modules, which wearable gadgets require for wireless connectivity. The Arduino BLE 33, although it contains these inbuilt sensors, has fewer memory, fewer I/O ports, and is a lot more expensive than the ESP32. For all the reasons above, the board ESP-WROOM-32 was the best option, as it meets all the factors. For futureproofing, the integrated sensors, such as the 9-axis inertial sensor, humidity and temperature sensor, gesture, proximity, light color, and light intensity sensor, of the Arduino BLE 33 can offer full hand tracking and add more features to the glove.

3.3 Tool Selection

This application was carefully developed using a variety of technologies, each of which had a specific function in influencing the design of the system. These technologies included Unity, KiCad, and Autodesk Fusion.

- **Unity,** developed by Unity Technologies**,** is a 2D (two-dimensional) and 3D video game engine assisting developers in bringing their gaming ideas to life since 2005. Unity is ideal for beginner developers because of its well-known user-friendly interface and some free tools. In contrast to Unreal Engine developed by Epic Games, a well-known game engine, this one is harder to use and has less documentation available for inexperienced creators. Unreal Engine has enhanced visual quality and faster generation of more exquisite, realistic visuals.

However, Unity is a preferable alternative because of its large community and ease of use when it comes to replicating an environment just for testing the functionality of a haptic glove (Kevuru Games, 2023).

- **KiCad EDA** is a free software package for electronic design automation (EDA). It makes electronic hardware design and simulation easier for printed circuit board (PCB) production. An alternative option for designing a PCB is Altium. With its very potent auto-routing features and integrated interactive components, Altium ensures optimal routing. While KiCad supports 3D viewing, Altium has superior rendering. Nonetheless, KiCad EDA is completely free, however, Altium licenses are around $7,000, making Altium not ideal for hand-learning and exploration without budgetary limitations. Furthermore, KiCad's robust documentation and vibrant user base provide helpful tools and assistance that are beneficial for resolving issues and learning the program (Chimdi & Sherri, 2024).

- **Autodesk Fusion** is a platform for 3D CAD (Computer-Aided Design), modeling, manufacturing, industrial design, electronics, and mechanical engineering, previously Fusion 360**.** With the ability to combine powerful simulation and production capabilities with parametric and freeform modeling, Fusion unifies an extensive tool suite into a single platform. Moreover, provides complex functionality for seasoned designers, but its user-friendly interface and wealth of tutorials and tools make it accessible to beginners (Autodesk, n.d.).

- **Arduino IDE,** initially designed for use with Arduino boards, is an open-source program that allows you to write and upload code. The Arduino IDE offers a flexible platform for a variety of applications by supporting a large selection of Arduino boards and associated devices, (for example ESP32 Boards, SAM Boards, Renesas Portenta Boards etc.). The extensive collection of pre-built and community-contributed libraries makes complicated code unnecessary by streamlining the interface process with sensors, actuators, and other peripherals. Additionally, it is compatible with C and C++ programming languages (Javatpoit, n.d.).

- **Steam** is a digital distribution platform and shop owned by Valve for video games. The Steam client allows customers to install PC games straight to their cloud storage after purchase (PC Games for Steam, n.d.).

- **SteamVR** is a tool for experiencing VR content on any gear you want. SteamVR supports Valve Index, HTC Vive, Oculus Rift, Windows Mixed Reality headsets, among others (SteamVR, n.d.).

Also, with the help of an API and runtime called **OpenVR**, apps may access VR devices from a variety of vendors without needing to be specifically aware of the hardware they are targeting. Hardware manufacturers may simply enable compatibility for VR content on Steam by creating OpenVR drivers for their devices. These drivers allow the system to identify the custom hardware as a VR controller or peripheral by converting the raw sensor data into OpenVR's input format (Steamgames, n.d.).

3.4 Implementation

To construct a working haptic VR glove, the implementation phase focuses on combining hardware and software. Finger motions are tracked by hall effect sensors, and tactile feedback is provided by haptic actuators(motors). In combination with finger tracking, Hand tracking will be accomplished using a VR Controller such as Vive Trackers or Oculus hand controllers and other similar things. A microcontroller that processes sensor data in real time is coupled to these components. To interpret this data and interface with VR systems such as Unity and SteamVR, custom firmware is designed to guarantee precise tracking and prompt response. The objective of creating an inexpensive VR Haptic Glove is aligned with extensive testing and iteration that improves sensor accuracy, user comfort, and feedback realism.

Chapter 4: Electronics and Parts

4.1 Introduction

In the world of modern technological advancement, the strategic inclusion of electronic components is critical to the success of many initiatives. This chapter focuses on the careful selection and integration of sensors, microcontrollers, and actuators, revealing the transformational effect these electronic components have on project efficiency and imaginative problem-solving.

4.2 Electronics

4.2.1 Esp32 Modules

The ESP32-WROOM-32D is a versatile module developed by Espressif Systems. suitable for a range of applications, including low-power sensor networks and demanding operations like voice encoding, music streaming, and MP3 (MPEG Audio Layer 3) decoding. At the core of this module is the ESP32-D0WD chip, with adjustable CPU (central processing unit) clock frequency ranging from 80 MHz to 240 MHz The ESP32-WROOM-32D is notable for its wireless connection. It supports both Wi-Fi (802.11 b/g/n) and Bluetooth (Bluetooth 4.2 and BLE (Bluetooth Low Energy)) protocols, making it very versatile for Internet of Things (IoT) applications. Because of the module's adaptability, it may be seamlessly integrated with sensors and external devices using a variety of peripheral interfaces, including GPIO (general-purpose input/output) pins, I2C (Inter-Integrated Circuit), I2S (Inter-IC Sound), SPI (Serial Peripheral Interface), and UART (universal asynchronous receiver/transmitter). Furthermore, the ESP32-WROOM-32D's low-power design, combined with hardware-based security features such as Secure Boot and Flash Encryption, makes it a top choice for applications ranging from home automation to industrial IoT, thanks to its extensive feature set and support for widely used development frameworks such as the Arduino IDE and ESP-IDF.

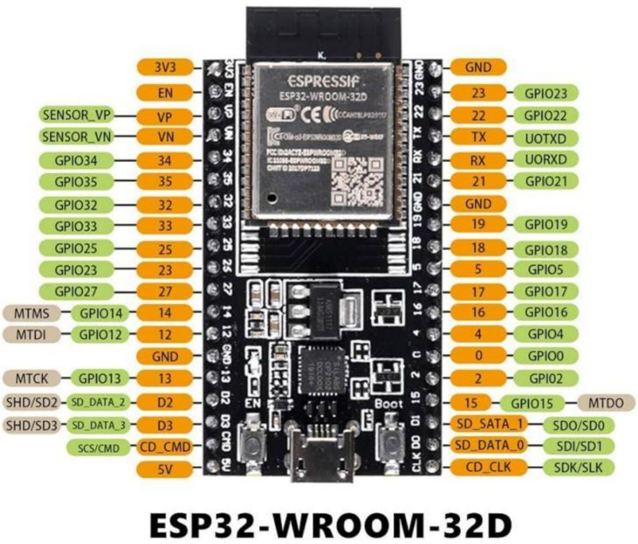


Image 4-1 ***ESP32-WROOM-32D and the pinout (BestArduino.com, n.d.)***

Module	ESP32-WROOM-32D
Core	ESP32-D0WD
SPI flash	32 Mbits, 3.3 V
Crystal	40 MHz
Antenna	on-board PCB antenna
Dimensions	18 × 25.5 × 3.1
Operating voltage/Power supply	3.0 V ~ 3.6 V
Operating current	Average: 80 mA
Module interfaces	SD card, UART, SPI, SDIO, I2C, LED PWM (Pulse width modulation), ADC (analog-to-digital converter), DAC (digital to analog converter)
operating ambient temperature range	40 °C ~ +85 °C

Figure 3 ESP32-WROOM-32D Characteristics

4.2.2 Servo Micro Motors

Servo motors have become a vital component in many industrial applications requiring high dynamics on position control, such as automation, numerically controlled equipment, and other mechanisms requiring precise and fast starting and stopping. The servo motors that were used in this project were the Servo Motor MG90S. Compact and powerful, the Servo Motor MG90S finds widespread application in robotics, remote-controlled vehicles, and diverse electrical applications. The MG90S is well-known for its precise control and high torque output, and it provides remarkable performance in a compact design (Haidar & Benachaiba, 2013). Despite weighing only 13.4 grams, this servo can provide torque up to 2.2 kg/cm2, which makes it appropriate for this project. Its working voltage range of 4.8 to 6.0 volts assures compatibility with typical power sources, and its sturdy design ensures longevity in harsh environments. The MG90S, with its metal gears, is exceptionally strong and resilient, allowing for smooth and precise movement for the haptic application of the glove (MG90S Micro Servo Motor Datasheet, March 2024).

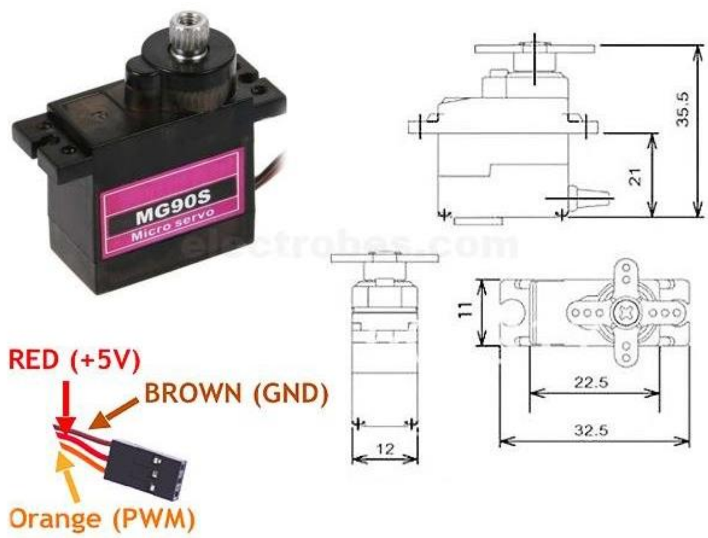


Image 4-2 Servo Motor MG90S with pinout and dimensions (Scaldas)

4.2.3 Multiplexer

Multiplexing is a generic name for the act of delivering one or more analog or digital signals across a similar transmission line at different times or speeds, and the equipment used to do so is known as a multiplexer. The multiplexer, often known as an "MPX" or "MUX," is a combinational logic circuit that uses a control signal to switch one or more input lines to a single common output line. Multiplexers function similarly to very quick-acting rotary switches with several positions, linking or managing several input lines, referred to as "channels," one at a time to the output. The multiplexer used from this prototype was multiplexer CD74HC4067 (Tutorials, n.d.). The CD74HC4067 is a multiplexer/demultiplexer integrated circuit, designed with CMOS technology, that excels in effectively managing multiple analog or digital signals in electronic systems. This IC (integrated circuit), which has 16 channels, enables the routing of one output to any of 16 destinations or the selection of one input from a potential set of 16 sources. It has a wide working voltage range and is compatible with both digital and analog signals, typically between 2V and 6V, making it compatible with this application (Semiconductor, 2013).

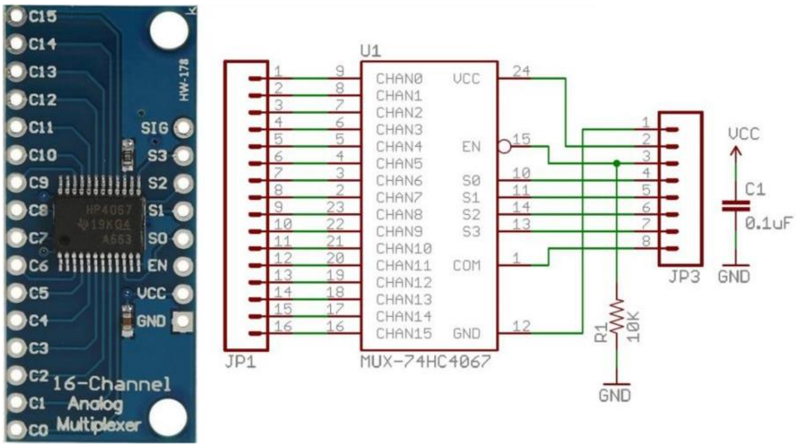


Image 4-3 Multiplexer CD74HC4067 (Lab.IT, n.d.)

4.2.4 Hall Effect Sensors

Hall effect sensors are critical components in modern electronics, extensively used for their ability to detect magnetic fields with accuracy and durability. A thin plate of conductive material, such as copper, is used in Hall effect sensors to convey a current that is provided from a source, such as a battery. When a magnetic field is provided to the plate at right angles to the current flow, as seen in image 12, a small voltage emerges across the plate. If you reverse the direction (polarity) of the magnetic field, the polarity of the induced voltage will likewise change. The Hall Effect is named after Edwin Hall. These sensors produce a voltage in direct proportion to the strength of the magnetic field they are subjected to, using the Hall effect as its operating principle (Ramsden, 2011).

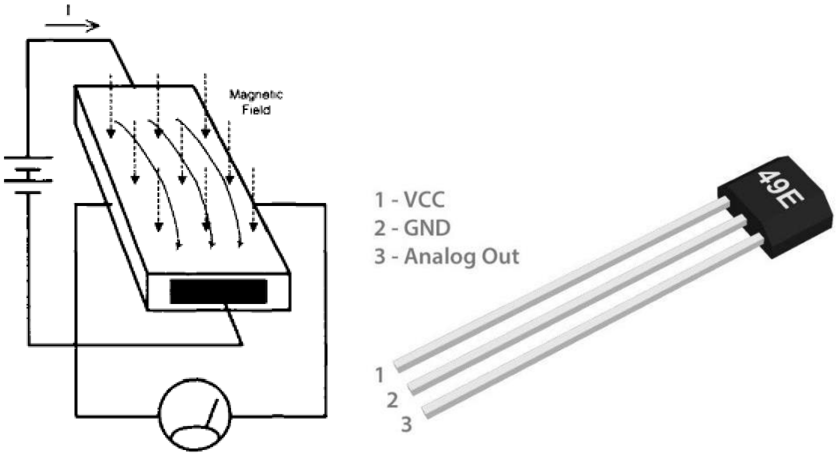


***Figure 4** Hall Effect in conductive sheet and Hall Effect Sensor 49E (Ibrahim, Ramsden, Huddleston, Ball, & Wilson, September 2008)*

The three most popular varieties are switch Hall effect, threshold, and linear sensors. Linear Hall effect sensors are perfect for applications like proximity detection and current sensing because they produce a constant output voltage that is proportional to the intensity of the magnetic field. On the other hand, threshold Hall effect sensors are appropriate for applications needing binary sensing, such as speed detection in automobile systems, as they provide an output signal when the magnetic field surpasses a predetermined threshold. With the additional capacity to toggle between two states, Switch Hall effect sensors perform similarly to threshold sensors and are commonly employed in non-contact switching and position-sensing applications.

The Hall effect sensors 49E were utilized in this research. Because of its tiny form size, low power consumption, and excellent sensitivity, the 49E version is especially well-known for being appropriate for a wide range of applications. The integrated circuitry features low noise output, which makes it unnecessary to use external filtering. The linear hall effect sensor 49E has an operating temperature range of 40^{o}C to 85^{o}C and provides a continuous output voltage (Figure 3) completing it perfectly for tracking the finger position (AllDataSheet).

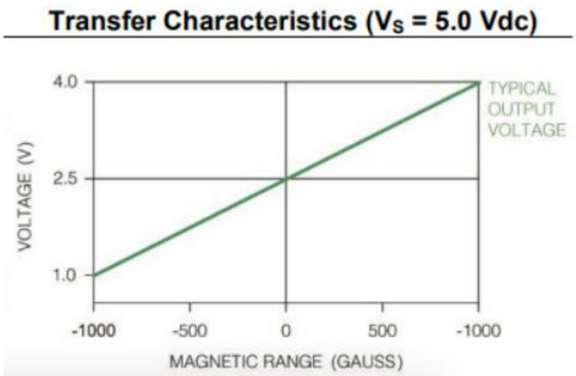


***Figure 5** Hall Effect 49E and the output Voltage (Ltd, n.d.)*

4.2.5 Printed Circuit Board- Design

Prototyping boards, often referred to as breadboards or solderless breadboards, offer a quick and easy way to design and test electrical circuits without having to solder anything. These boards have an interconnected grid of holes into which jumper wires can be used to connect and insert electrical components. Rapid prototyping and experimentation are made possible by prototypes, which make it simple and quick to make iterations to circuit designs. Printed circuit boards, or PCBs, are made to hold electrical components and make connections between them easier. Compared to traditional breadboarding, they have several benefits, such as greater repeatability, compactness, and dependability. Depending on the intricacy of the circuit, they can be produced in single-sided, double-sided, or multilayered configurations utilizing specialized processes like etching, milling, or printing. Both Breadboards and custom PCBs were applied to receive the best results.

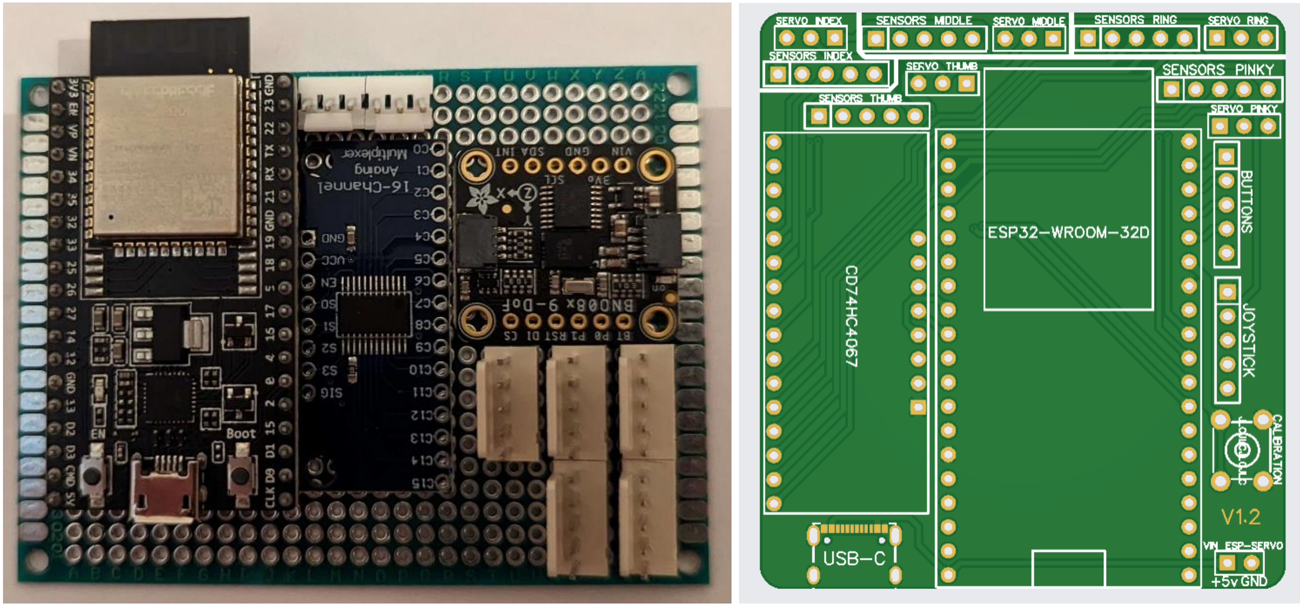


Image 4-4 Prototyping board (left) and PCB (right) designs

4.3 Parts

This section provides a thorough examination of the fundamental elements necessary for the building of electronic systems. Based on the fundamentals of material science and electrical engineering, we want to clarify how these elements interact and highlight how important they are in determining the performance and usefulness of electronic systems.

4.3.1 Connectors

Crimping connections to wires is an essential skill for electronic assembly and maintenance. By compressing a connector firmly around a wire with a specialized tool, crimping a connector onto a wire creates a dependable and secure connection. Because of its speed and simplicity of usage, this approach is frequently used, especially for applications that need numerous wire connections. There are several varieties of wire connectors, each customized to meet certain needs and uses. With a threaded screw mechanism that firmly clamps wires in place, screw terminals provide a straightforward yet reliable solution. In contrast, crimp connections compress a metal sleeve around the wire using a crimping tool to create a secure connection that is perfect for situations requiring strong mechanical strength. Lever connectors require no equipment to insert and clamp wires, just a quick flip of the lever does the job. In this prototype, the wire connectors that were used were the JST (Japan Solderless Terminal) wire connectors. JST connectors include a special crimping mechanism that makes it possible to join wires firmly without soldering, suitable for easy and compact electrical connections. This connector provides modular electronics systems that are easily customizable, upgradeable, and interoperable, thus separating the glove circuit into modules rather than one complete setup.

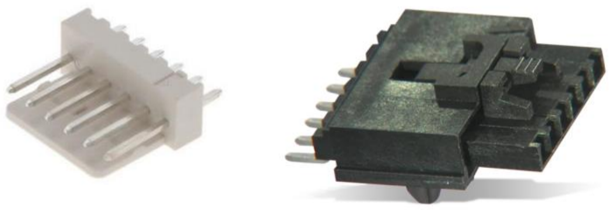


***Image 4-5** Wurth WR-WTB 2.54mm Male 5pin Connector (DigiKey, n.d.) and Molex CGrid SL IDT opt G TIN 5pin Connector (Electronics, n.d.) that were used in PCB and glove.*

Another type of connector that was used in this research was headers. Headers are used to make electrical connections between electronic components, circuit boards, and peripheral devices. These connections usually include rows of female sockets or male pins enclosed in a metal or plastic enclosure. Single-row female/male header strips were utilized to provide a reliable and standardized interface for connecting various components, such as hall effects sensors and servos motors, on the custom PCB.

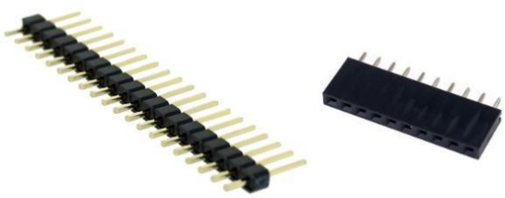


***Image 4-6** Male (Grobotronics, Grobotronics.gr, n.d.) and Female (Grobotronics, Grobotronics, n.d.) 2.54mm Single Row Pin Header*

4.3.2 Permanent Magnets and Bearings

Permanent Magnets are crucial components of contemporary technology, known for their ability to retain their magnetism after being magnetized without requiring any external magnetic field to function. Permanent magnets are made from ferrite (ceramic), alnico, and neodymium. Ceramic Magnets, made from iron oxide, barium, or strontium carbonate, are affordable and frequently utilized in applications such as refrigerator magnets, loudspeakers, and crafting. Alnico Magnets consisting of aluminum, nickel, and cobalt, are used in electric motors, guitar pickups, or microphones because they provide a strong magnetic field and good temperature stability. Neodymium magnets are the strongest magnets out of these three and are located in magnetic resonance imaging (MRI) machines, hard disc drives, and mobile phones. Hall effect sensors are ideal for using neodymium because of their powerful magnetic field for their size, which is thus used in the glove.

Image 4-7 Cylindrical Neodymium Magnets (Magnitech, n.d.)

Bearings play an important role in machinery because they provide support, reduce friction, and allow shafts or other mechanical elements to rotate or move smoothly. Typically made comprised of inner and outer races with rolling components like balls or rollers, they may handle radial and axial stresses. The bearings used for the Magnet Pinion and Bearing Retainer (Image 4-10, Image 4-13) for the side-to-side movement of hand and magnet gear, were the 6700Zz bearing steel metal shielded ball bearing thin-walled roller 10X15X4 mm because they were affordable and the perfect size.

Image 4-8 Cylindrical Neodymium Magnets (Amazon, n.d.)

4.3.3 3D Printed Parts

Sensor Housing (Image 4-9 Sensor Housing) is the most important 3D printed part because it's the home for the hall effect sensors and the motors. Furthermore, is the one you should print with the utmost care since there are so many parts connected with it. Magnet Pinion is the part where the magnets are stored so the hall effect tracks the magnetic field with the rotation of the gear (Image 4-10). Servo Gear is attached to the servo motor and moves the servo rack (Image 4-11) so can stop the movement of the finger and simulate the illusion of haptics. Sensor Top (Image 4-10) just ensure that the magnet pinion and sensors stay in place. The Finger Rack (Image 4-11) is a flexible part designed to follow the finger and move the magnet pinion therefore tracking the movements of a user's finger. Moreover, the guide ring and End ring (Image 4-12) are the parts to guarantee that the finger rack is well connected to the user's finger. Last but not least, the Rigid Mount attaches all pieces to your hand/palm and the Bearing Retainer holds the bearings onto the Rigid Mount (Image 4-13). Not to mention, a specially made breadboard holder (Image 4-14) was made to keep the entire circuit in place and hold it together while the user wears and operates the glove. All designs and modifications of the 3D designs were executed through Autodesk Fusion.

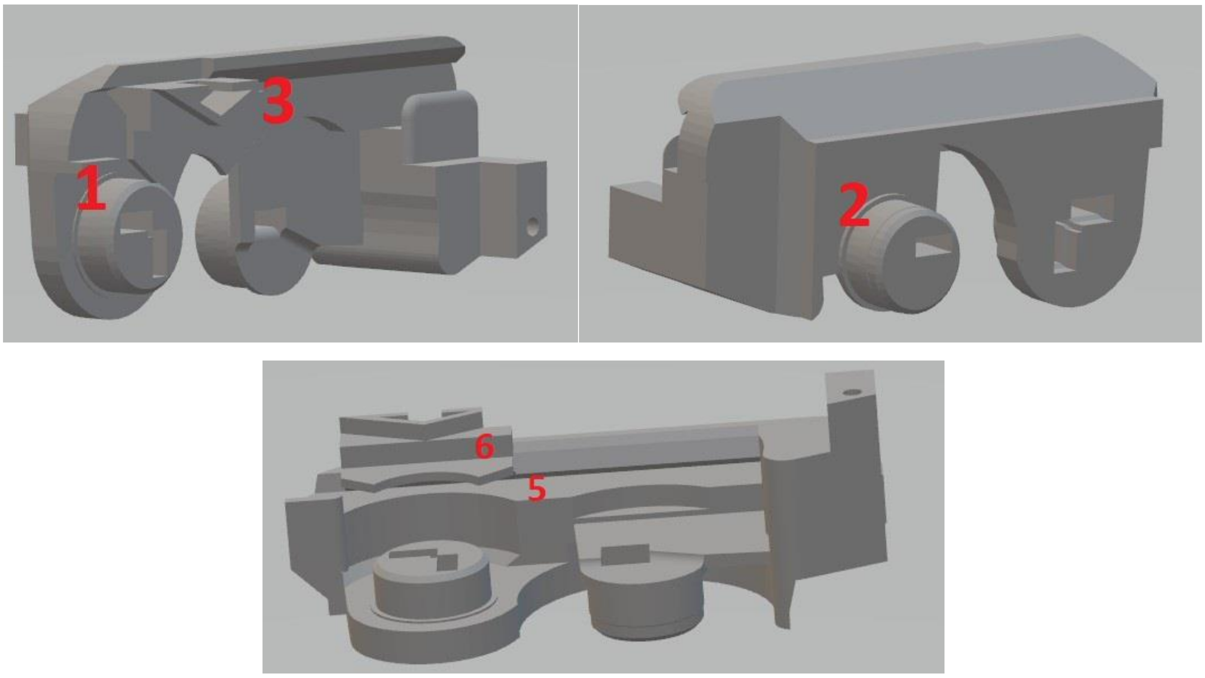


Image 4-9 Sensor Housing

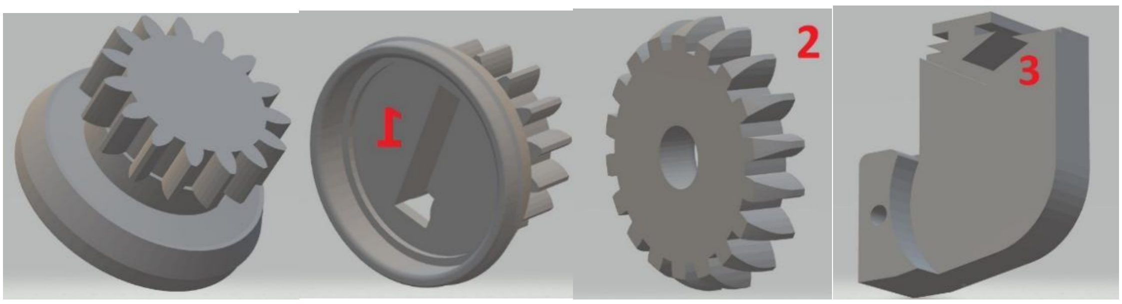


Image 4-10**Magnet Pinion (1), Servo Gear (2) and Sensor Top (3)**

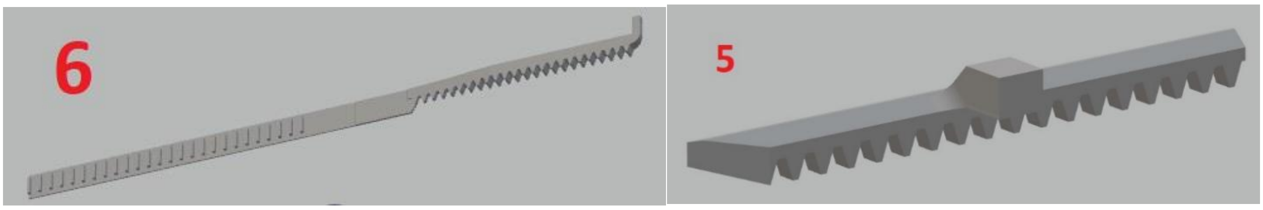


Image 4-11 ***Finger Rack (6) and Servo Rack (5)***

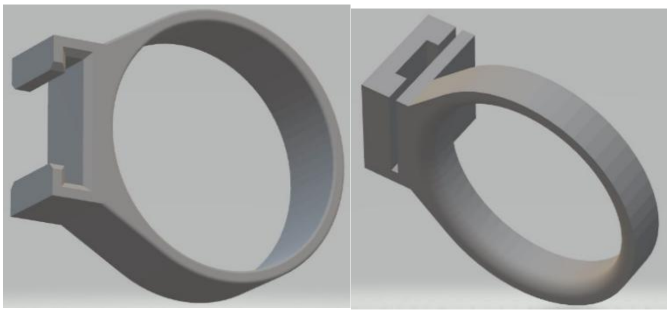


Image 4-12Guide ring and End Ring

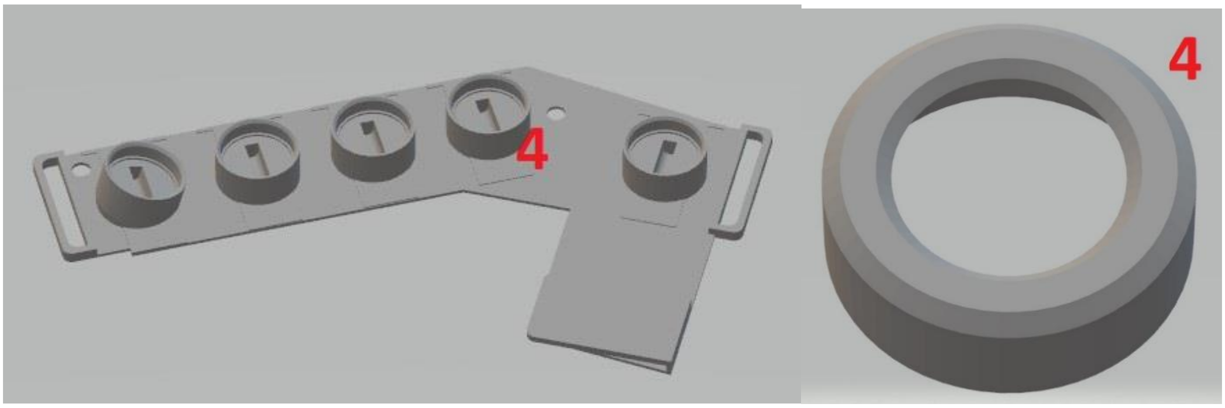


Image 4-13 ***Rigid Mount and Bearing Retainer (4)***

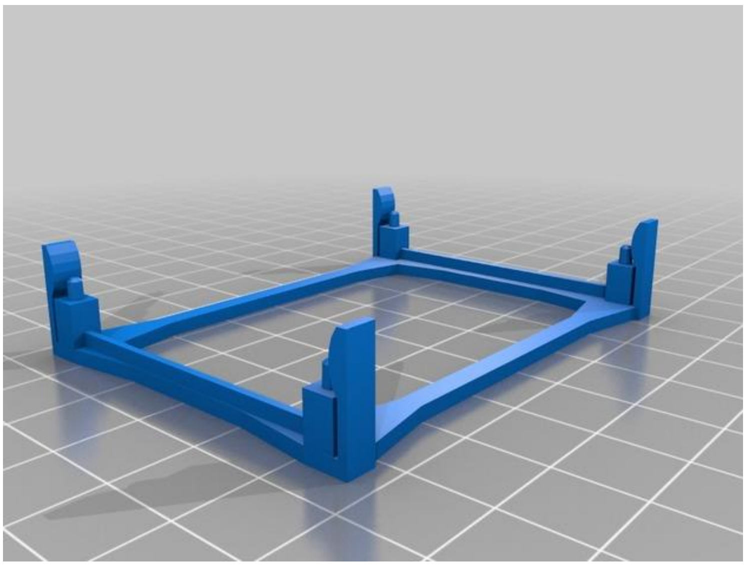


Image 4-14 PCB Holder

4.4 List of all Electronics and Parts

- Solderable Breadboard with solder mask on both side 50x70mm
- CD74HC4067 16 Channel analog bidirectional MUX Breakout Board
- Servo Micro Motor 9G MG90S 180m degree
- ESP32-WROOM-32D
- Dual-Axis XY Joystick Module for Arduino
- USB-A to USB-B Wire
- Cylindrical Neodymium Magnets diameter 3x10mm
- Honeywell SS49E Analog Hall Effects Sensors
- UJ2-Bh-W1-TH CUI Devices USB 2.0 type B Jack 4 pin TH (Image 4-16)
- Wurth WR-WTB 2.54mm Male 5pin Locking
- Wurth WR-WTB 2.54mm Female Crimp Contact
- Wurth WR-WTB 2.54mm Female 5pin Terminal Housing
- Molex Housing SG 5P W/O Ears Single Row
- Molex SL Crimp 22-24 AWG Male
- Molex CGrid SL IDT opt G Tin 5Ckt
- TE Connectivity/Amp Friction Lock Header 3P Straight Post tin
- 6700Zz bearing steel metal shielded ball bearing thin-walled roller 10X15X4 mm
- 40-Pin Female Single Row 2.54mm Pin Header Strip
- JST XH2.54 2P 2Pin Connector MALE-FEMALE (Image 4-15)
- 12x12x6mm Momentary Tactile Push Button Switch
- ST XH2.54-5P 5Pin Connector MALE-FEMALE
- 6 Pin SMT Socket Connector Micro USB Type C 3.1 Female Placement SMD DIP (Image 4-16)
- Standard 0.22mm^2 stranded wire
- Velcro Straps
- Silicone Grease for gears
- PCB

***Image 4-15** JST XH2.54 2P 2Pin and 5Pin Connector MALE-FEMALE for the hall effect sensor and power (Cableworks, n.d.)*

***Image 4-16** Type C (Shouhan, n.d.) and Type B (RS, n.d.) connectors for supplying the circuit with power*

Chapter 5: Prototyping of the Haptic Glove

5.1 Introduction

The process of prototyping haptic glove hardware involves a complex interplay between complex sensor systems and sophisticated feedback mechanisms in an effort to close the gap between the digital and physical domains. Using actuators that provide immersive feedback and sensors that capture tactile details strategically, designers want to provide an intuitive interface that goes beyond conventional input devices. This method offers a canvas for creativity, offering life-changing experiences that redefine human-computer connection, despite the difficulties of ergonomic optimization and technical limitations. This project was motivated by the innovative virtual reality technology project LucidVR. Our haptic glove research was made possible by LucidVR's creative use of cutting-edge haptic feedback methods to improve virtual reality experiences. Our design aims to deliver more tactile and immersive interactions within virtual environments, building on the foundation laid by LucidVR. Through the integration of sophisticated hardware components and state-of-the-art feedback algorithms, our haptic glove aims to improve user engagement and realism in virtual reality applications.

5.2 Haptic Glove Assembly

The sensor housing (Image 4-9) for our haptic glove detects finger movements using a simple yet effective method. When the user opens or shuts a finger, a gear (Image 4-10) inside the housing rotates. This gear is equipped with strategically placed magnets. As the gear turns, these magnets pass through Hall effect sensors, which can detect magnetic field changes. By properly detecting these fluctuations, the sensors can estimate the degree of rotation and, as a result, the finger's position. This data is then used to deliver real-time feedback and interactivity within virtual worlds, hence improving the overall user experience through precise and responsive control (Image 5-1).

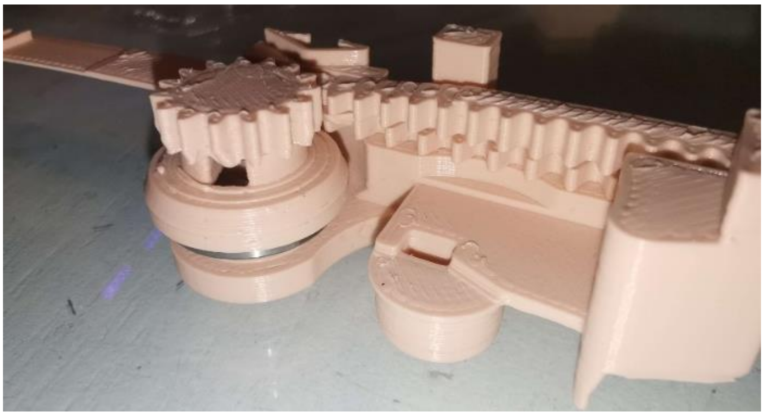


Image 5-1 Sensor Housing without hall effect sensors

The first stage in making the haptic glove is precisely connecting the Hall effect sensors (Ramsden, 2011) and fastening them with heat shrink tubing (Image 5-3 Wired Hall effect sensors with heat shirks. Each sensor has three wires: red for power, black for ground, and green for data transmission.

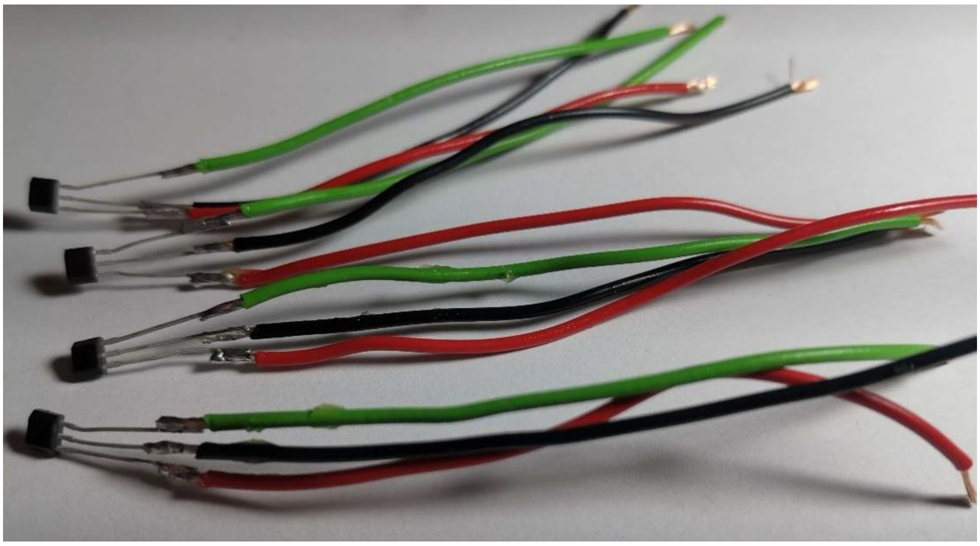


Image 5-2 Wired Hall Effect sensor

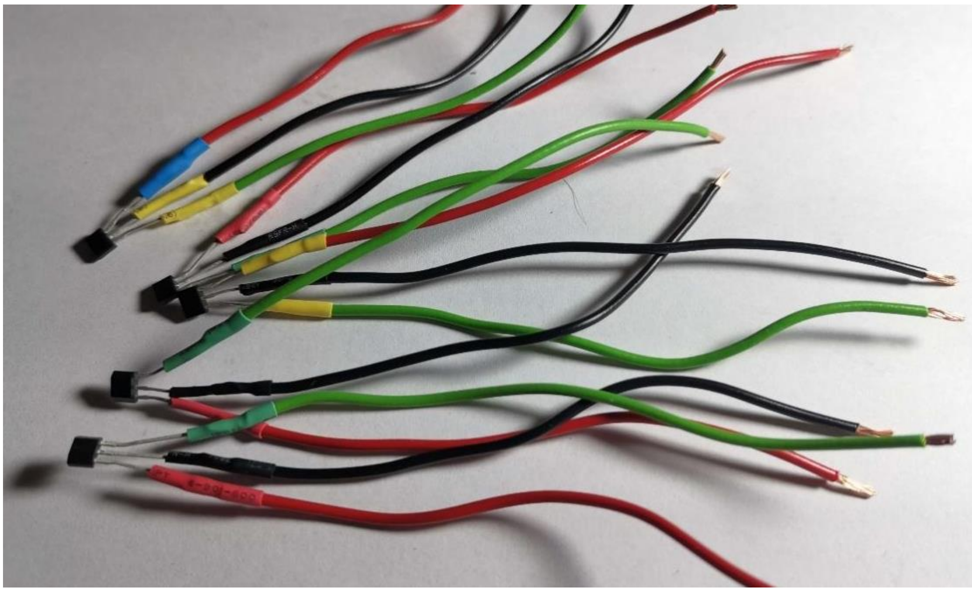


Image 5-3 Wired Hall effect sensors with heat shirks

To simplify the wiring process, Hall effect sensors are crimped with SL Crimp 22-24 AWG Male connectors (Image 4-5). This method simplifies the wiring by lowering the number of wires that go to each finger. Instead of having three wires per sensor (for a total of nine wires per finger), we combine the power and ground lines for all sensors on each finger to create a single five-wire connection (Image 5-4). This strategy reduces wiring complexity while also increasing modularity, making it easy to repair or upgrade individual finger modules.

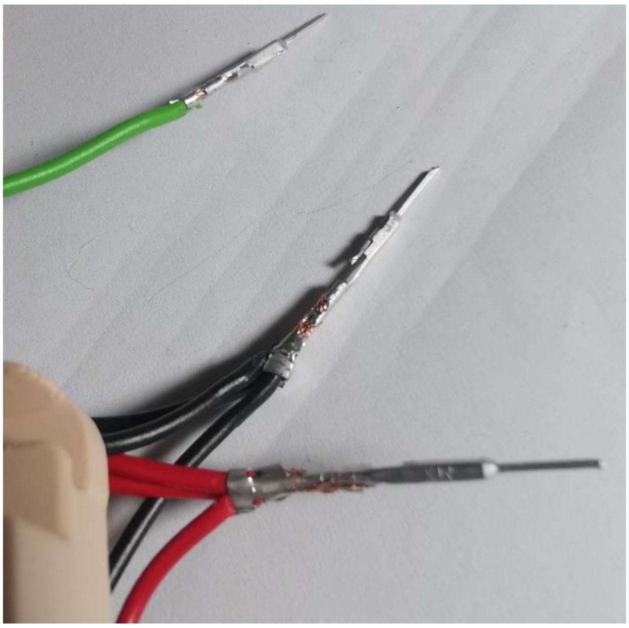


Image 5-4 Crimped Wires of Hall Effects Sensors

Hot glue is used to secure the Hall effect sensors into the corresponding holes on the sensor housing. This extra glue firmly binds the sensors in place, limiting movement and maintaining consistent performance during operation.

The sensor inputs end in a 5-pin JST connector on the back of the sensor enclosure, allowing for a tidy and fast installation while ensuring dependable connections. After the wires are properly connected and insulated, the sensors are carefully inserted into the specified holes in the sensor housing. This exact arrangement ensures that finger movements are accurately detected, setting the groundwork for the glove's capacity to offer detailed haptic feedback in response to user actions (Image 5-5).

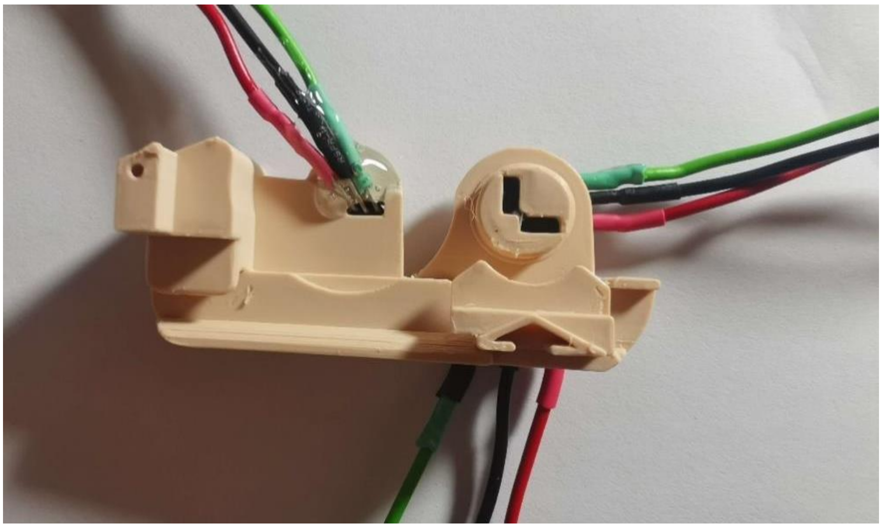


Image 5 5 Sensor Housing with the hall effects sensors

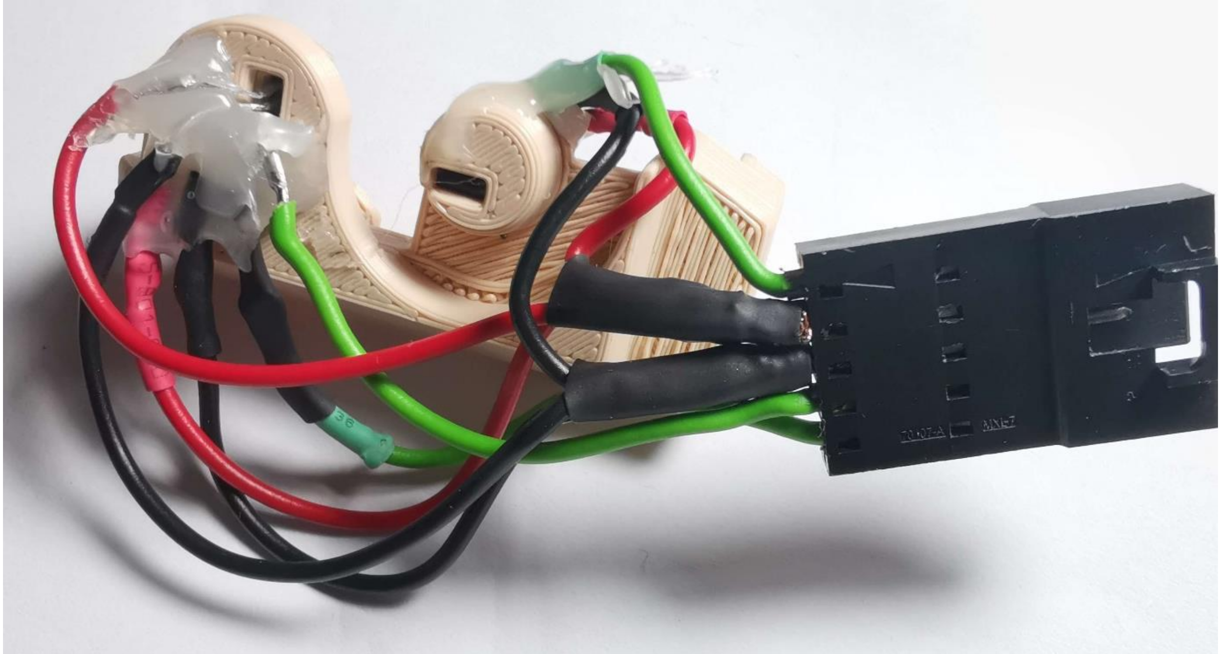


Image 5-5 Hall Effects sensors of one finger with a Molex CGrid SL IDT opt G Tin 5Ckt connector

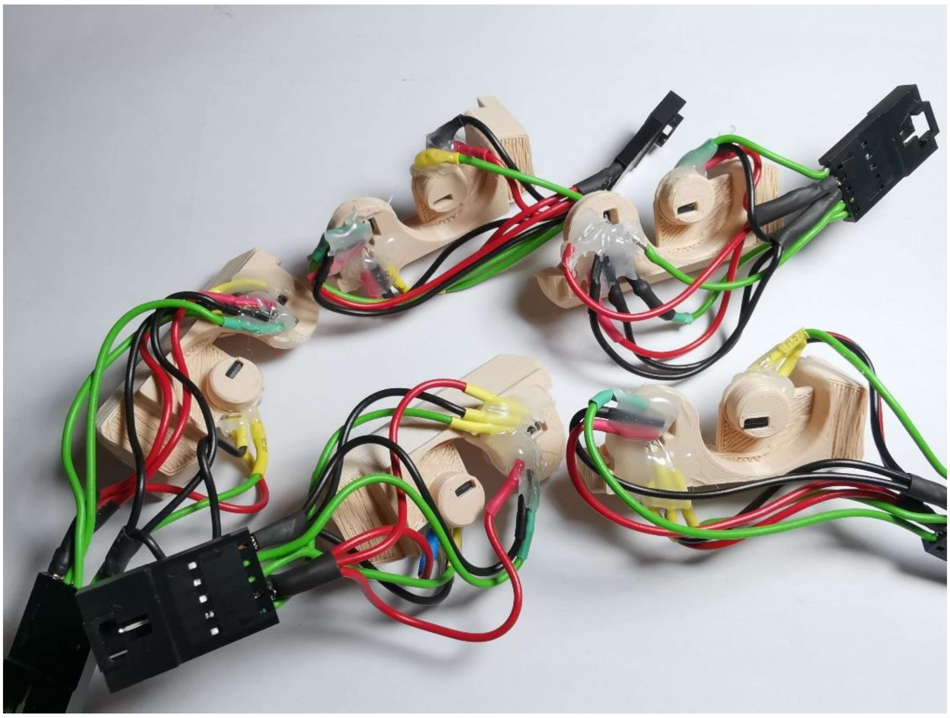


Image 5-6 The wired hall effect sensors for all five fingers

This procedure is conducted methodically on all five fingers to achieve uniform performance throughout the glove. Each Hall effect sensor is securely connected to its proper position within the sensor enclosure with hot glue to ensure stability and dependability (Image 5-6). The gears, embedded with strategically placed magnets, the rolling bearings, and finger racks (Image 4-11) are then carefully mounted above the sensors (Image 5-7).

Image 5-7 Gear with the magnets and rolling bearing (LucidVR, Lucidglove, n.d.)

This design enables the Hall effect sensors to detect fluctuations in the magnetic field caused by finger movements, allowing for accurate and real-time monitoring of finger positions and gestures (Image 5-8).

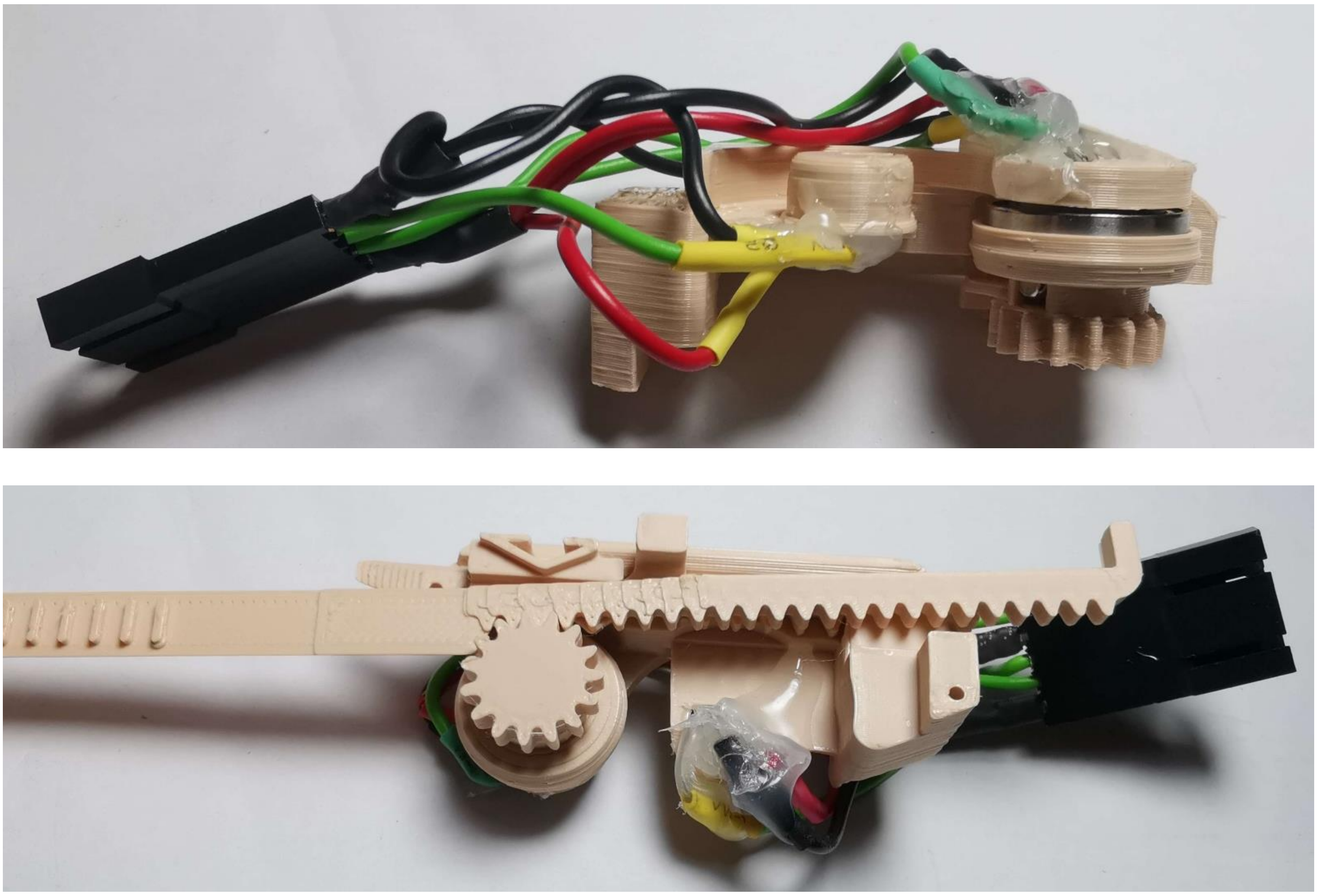


Image 5-8 The final Sensor Housing

To improve the haptic feedback experience, the design includes a motor connected to a gear mechanism and a rack system. This unique setup simulates resistance when a user grasps an object in a virtual world. As the finger flexes, the motor activates, forcing the gear to move the rack into position and thus stopping further finger movement at a predetermined point (Image 5-9). This feedback simulates the sense of holding an object, giving users a more immersive and realistic connection by imitating the physical resistance found in real-world circumstances.

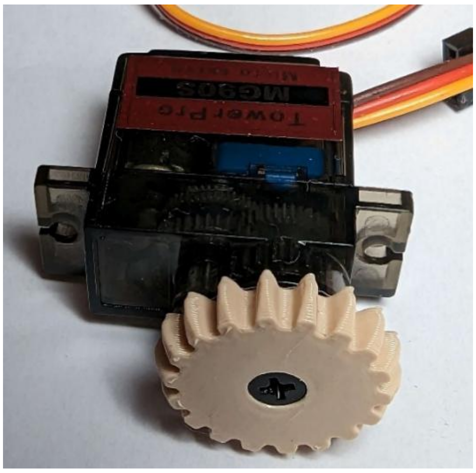


Image 5-9 Motor with the attached gear

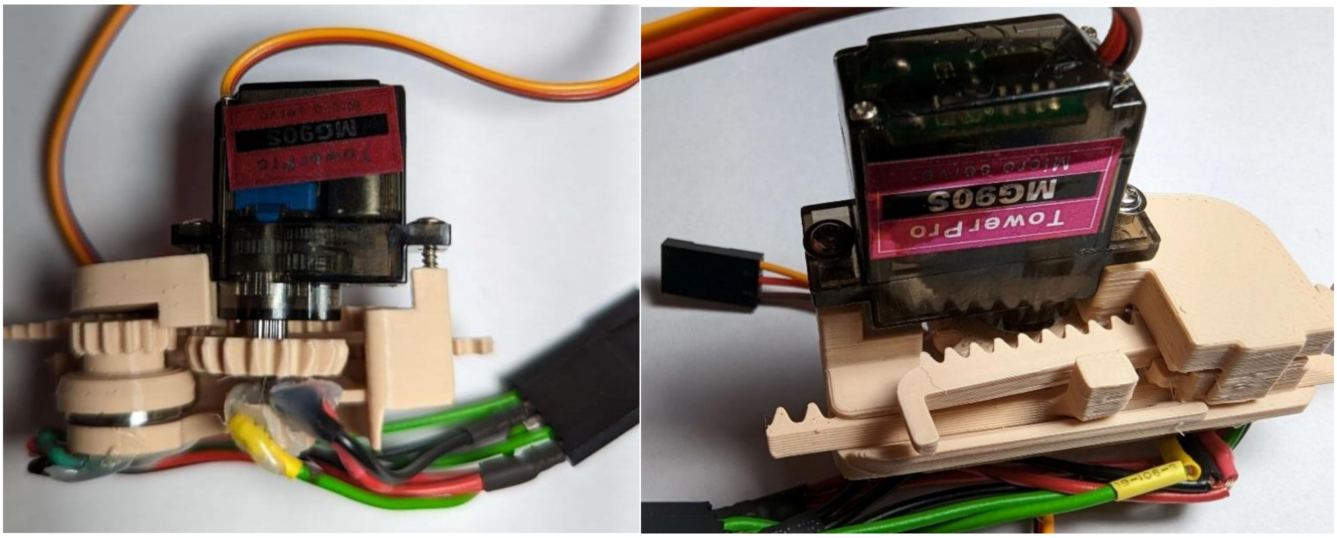


Image 5-10 The Sensor Housing with the haptic feedback

A sturdy mount is designed with rolling bearings and magnets (Image 5-7) to enable accurate tracking of lateral finger movements, ensuring a secure connection of the sensor housing to the hand. During operation, its design maintains stability and encourages smooth articulation (Image 5-10). Once finished, the stiff mount (Image 4-13) is attached to a glove, allowing the complete haptic feedback system to be seamlessly integrated for user wear.

In order to maximize the user experience in interactive virtual environments, Velcro straps are also used to fasten the glove, minimizing any displacement during use and guaranteeing a consistent and comfortable fit.

Image 5-11 Glove for mounted sensor housing

Then, using cyanoacrylate adhesive, the sensor housings for each of the five fingers were firmly attached to the sturdy mount. By ensuring a strong bond, this adhesive improves the assembly's structural stability. We attain ideal alignment and stability by fastening each sensor housing in this way, which is essential for precise sensor performance and trustworthy haptic input when using the sensors. The final product with all sensors and parts attached is shown in Image 5-12. The next step is building and wiring the electronics on the circuit board.

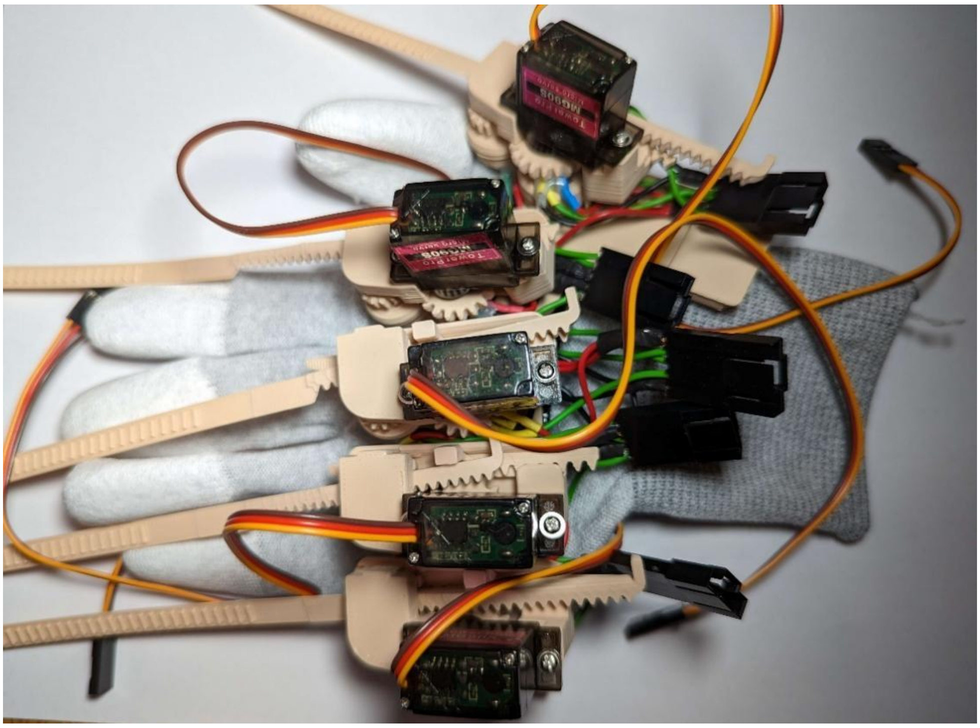


Image 5-12 Glove with the tracking mechanism

5.3 Custom Circuit Board

5.3.1 Wiring

The original idea was to use a protoboard to design and wire a custom prototype board. Finding a suitable protoboard size that could hold all the required electronics and connectors was the first step in this procedure. This necessitated carefully planning the arrangement to guarantee effective use of space and simple access to the components. Our goal in choosing the appropriate protoboard size was to make it easier to integrate and wire the many electronic components in an ordered manner, which would improve the haptic glove system's overall performance and dependability.

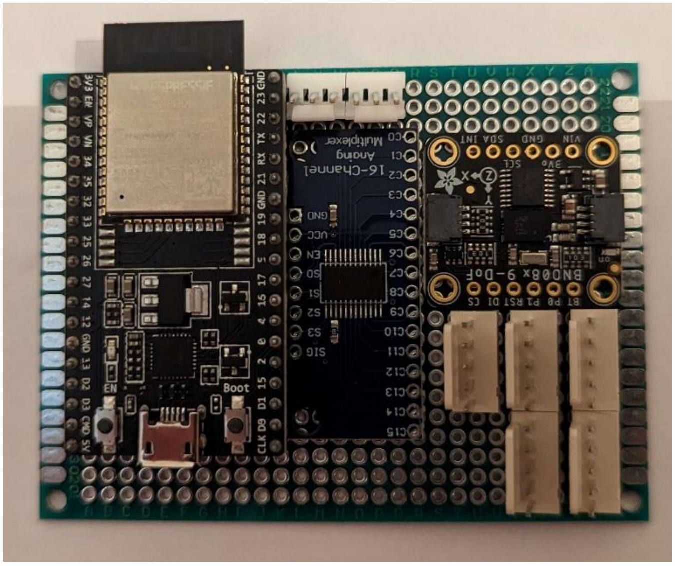


Image 5-13 Arrange of the prototype board

Following the selection of the right protoboard, the next step was to precisely solder female headers onto the board, allowing modularity and easy removal of electronic components for maintenance or updates. This design choice encourages flexibility in the system's architecture.

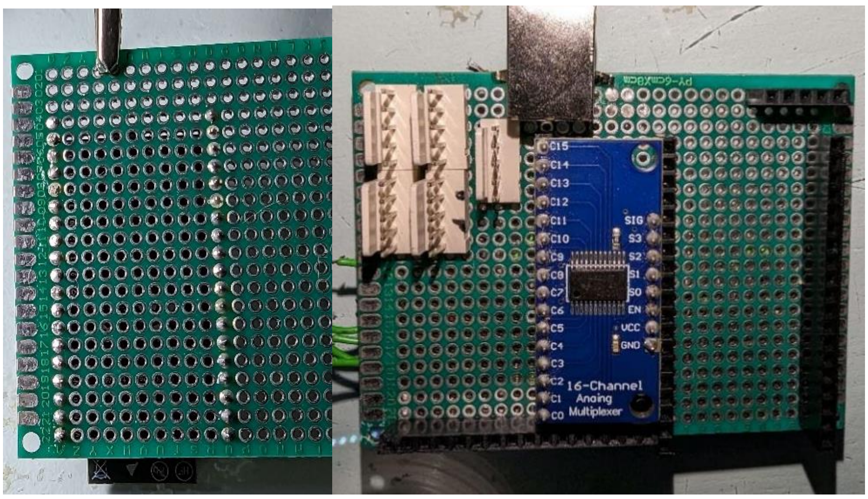


Image 5-14 Soldered parts on the protype board

After the headers had been firmly soldered, the following step was to wire the electronic components and connectors according to the schematic (Image 5-15). This systematic technique assures proper communication, which improves the overall functioning and reliability of the haptic glove system. The Schematic was made by using KiCad EDA.

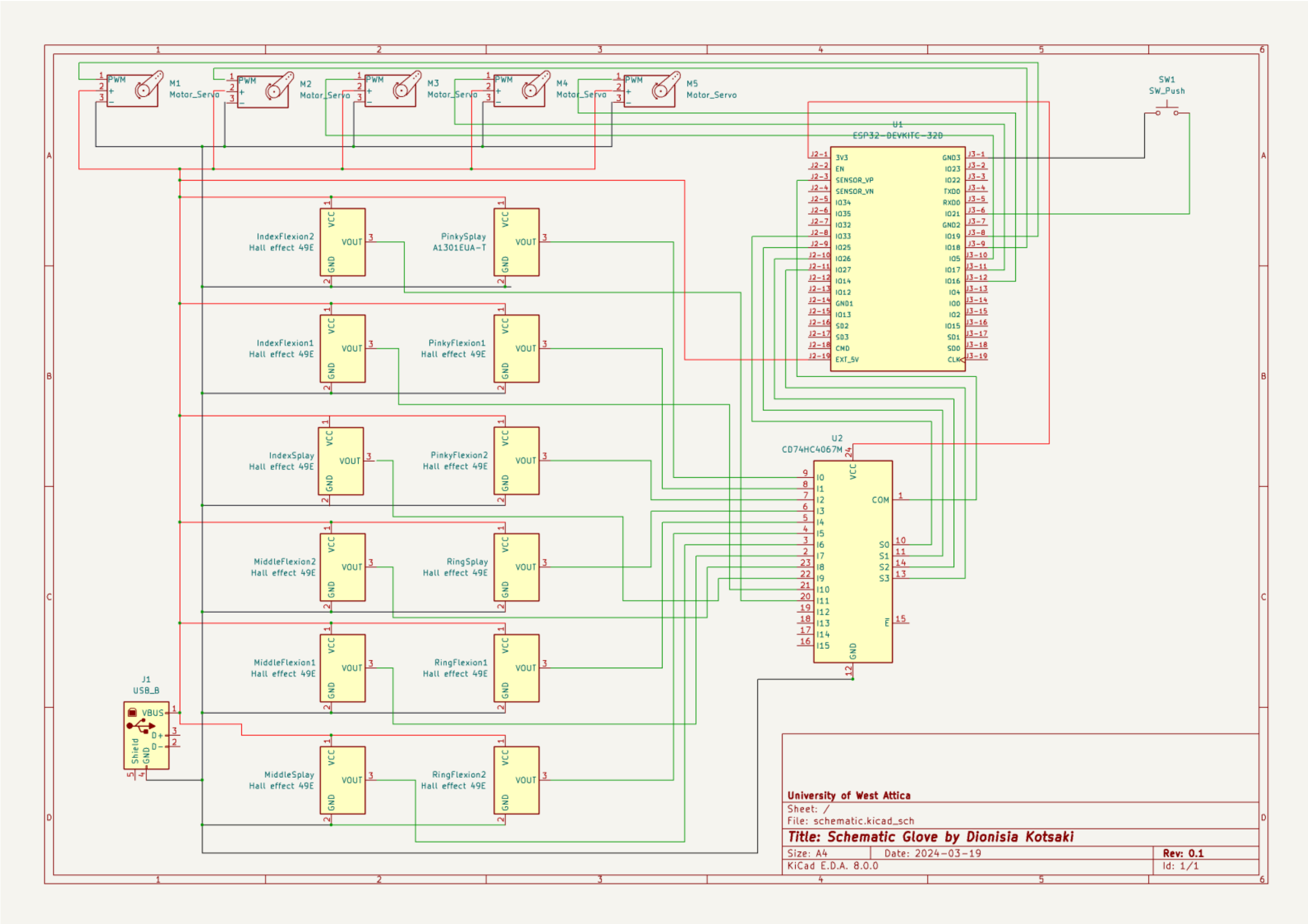


Figure 6 Schematic of the PCB of the Haptic Glove

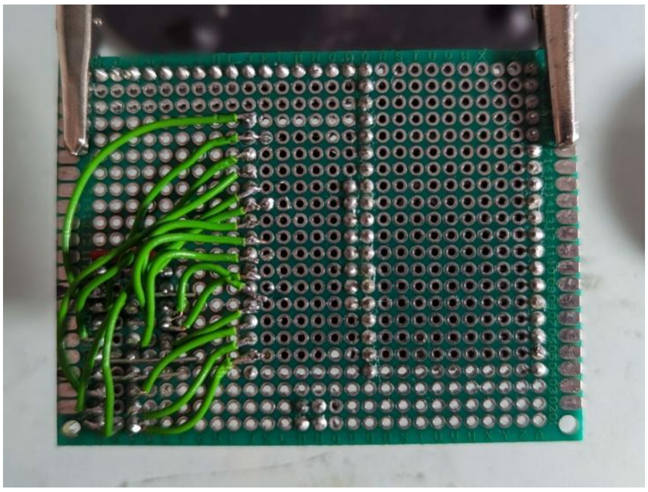


Image 5-15 Wiring the prototyping board

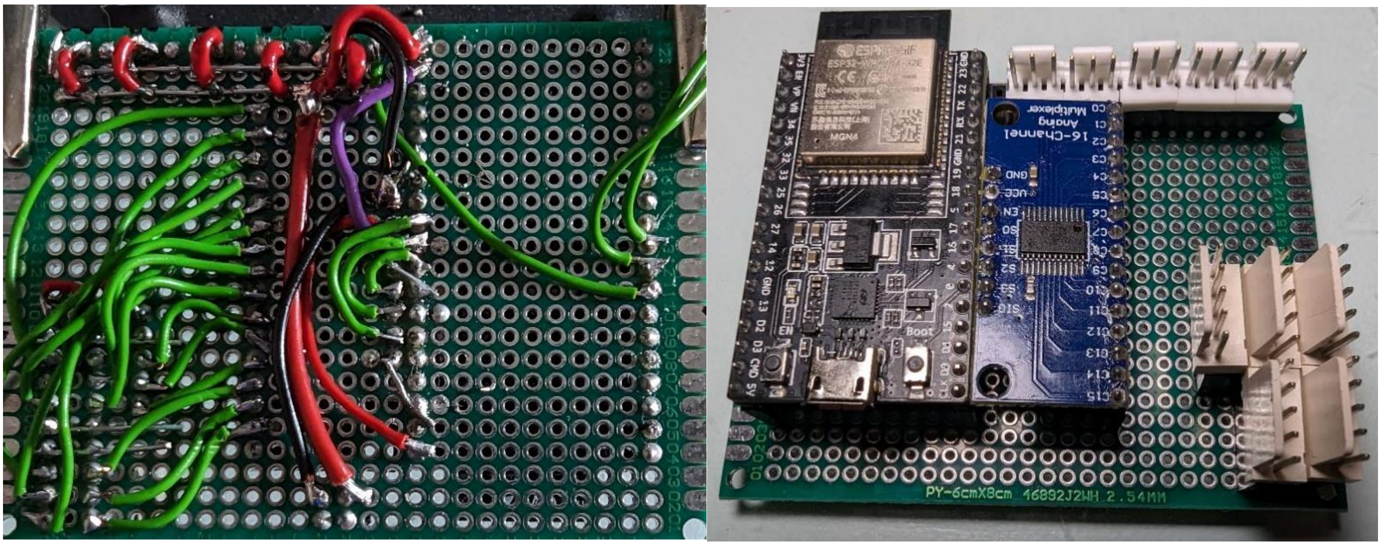


Image 5-16 The final wiring of the custom prototyping board

Then, the prototyping board was placed on a PCB holder made specifically for its dimensions. The usage of the PCB holder not only made it easier to reach the connections but also reduced the chance of component breakage, increasing the efficiency of the development process.

Image 5-17 PCB Holder

5.3.2 Problems

Several technical obstacles arose when working on the PCB prototype. The first problem was with the headers positioned between the connectors, which caused instability for the Hall effect sensor connectors. These connectors tended to detach with minimum movement. The approach entailed desoldering the headers off the board and directly attaching the 5-piece connectors, which improved mechanical stability and electrical dependability.

The second problem stemmed from the hand-wired board's intrinsic complexity. The large number of manual connections increased the risk of short circuits, complicating debugging attempts. Using a multimeter, a comprehensive analysis of each connection was performed, ultimately revealing the problem as incorrectly connected motor wires. Despite correcting the connection problem, the prototype's performance remained inconsistent.

To solve these concerns, a decision was made to switch to a professionally manufactured printed circuit board (PCB). This technique was intended to reduce the dangers associated with hand wiring, maintain consistent connectivity, and improve the overall resilience of the electrical assembly.

5.4 Printed Circuit Board

In search of a more dependable and effective solution, I discovered PCB design files for a related project created by MrAdriaki. Using these files as a starting point, I used KiCad, a powerful open-source electronic design automation software, to carefully tweak and customize the original PCB layout to match the specific needs of my haptic glove system. This required extensive changes to the layout and routing of traces to accommodate my specific combination of connections and electronic components, assuring maximum compatibility and performance.

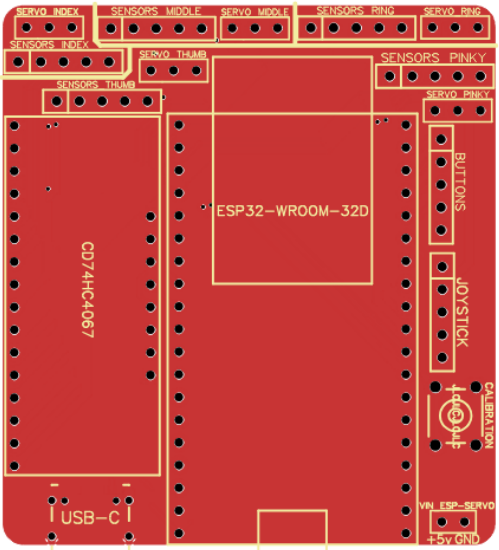


Figure 7 PCB Layout by MrAdrianki

By adopting and adapting MrAdriaki's PCB design, I was able to speed up the development process while dramatically improving the structural integrity and dependability of the final electronic assembly. This method not only saved me time, but it also made it easier to create a robust, custom-engineered solution that was tailored to my project's specific requirements.

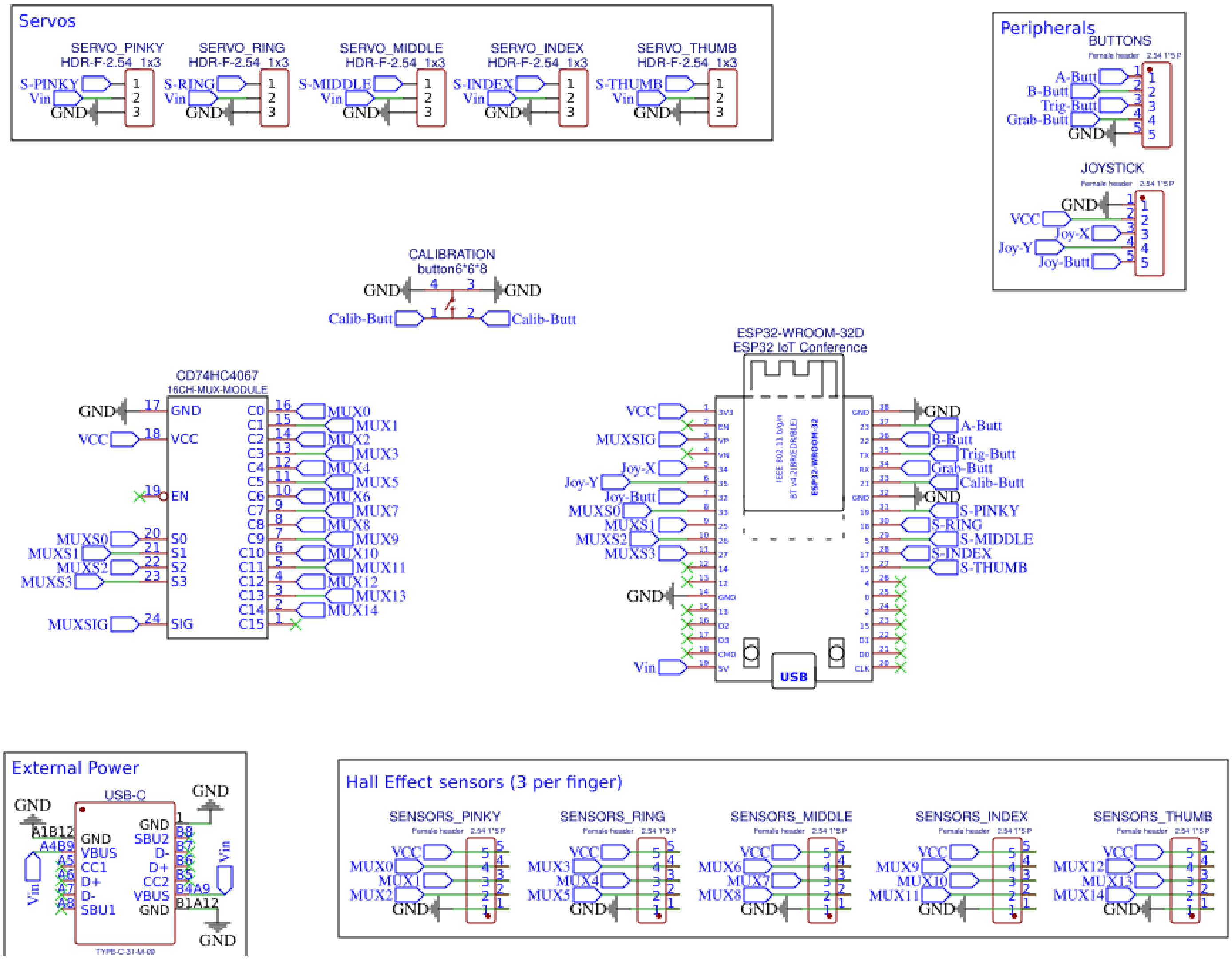


Figure 8 Schematic of the new PCB

The final construction of the PCB required precise soldering of various key components, such as headers, JST connections, and a button. The headers were carefully soldered to create strong connections for modular components, making them easy to remove and replace as needed. The JST connectors were used to enable secure and dependable connections between the various electronic modules, resulting in constant performance. In addition, a button was added to the board to allow for user input, improving the functioning of the haptic gloves system. The finished result displayed below demonstrates a well-organized and fully operating electronic assembly that is ready for incorporation into the haptic glove.

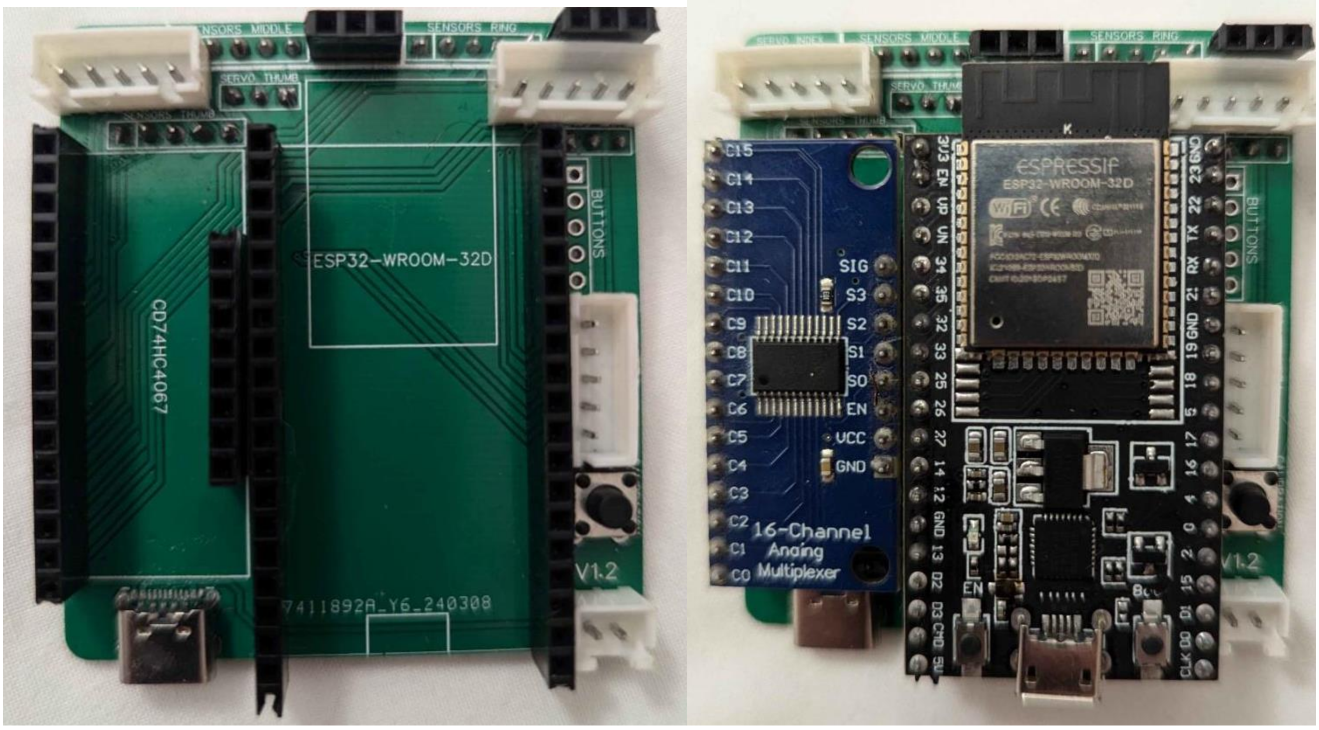


Image 5-18 Final PCB

The PCB was originally planned to be powered via a Type-C port or a proprietary USB connector, giving users more power options. However, due to a lack of 12-pin Type-C connectors in Greece, an alternate approach was required. To overcome this, a USB Type-A connector was connected and soldered directly onto the printed circuit board. This adaptation guaranteed that the power supply needs were met, preserving the design's functionality and integrity while accommodating the limits of local component availability. This pragmatic approach enabled ongoing progress in the development of the haptic glove system despite supply chain constraints. The circuit was powered by a power bank or battery pack, a portable device that stores energy in a battery (Wikipedia, n.d.).

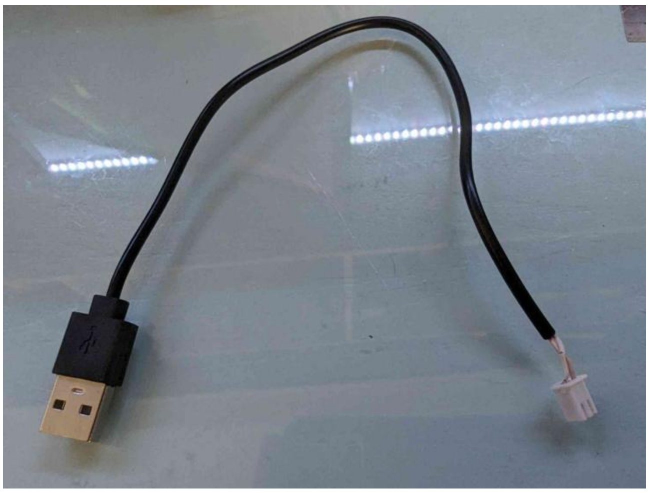


Image 5-19 Power Cable

Chapter 6: Programming of Haptic Glove

6.1 Introduction

The haptic glove is programmed using a two-pronged approach that includes both Arduino and Unity to ensure smooth integration and operation. The Arduino platform is used to provide low-level control over hardware components such as sensors, actuators, and feedback mechanisms. By building custom firmware, we can precisely read sensor data and control haptic feedback in real-time, resulting in accurate and responsive interactions.

On the other side, Unity is used for high-level development of the virtual environment and user interface. Unity allows us to construct immersive virtual experiences that interact dynamically with the haptic glove. This requires C# scripting to control communication between the virtual environment and the physical glove, allowing for synchronized feedback and interaction. The haptic glove provides a full and compelling user experience by combining the capabilities of Arduino and Unity, linking the physical and virtual worlds.

6.2 Arduino

6.2.1 Initialization

An important part of setting up the ESP32 microcontroller for the haptic glove system is initializing the pins. The procedure entails designating distinct pins for diverse inputs and outputs to guarantee appropriate communication and operation. Each finger on the ESP32 board has multiplexed pins assigned to it, allowing the multiplexer to precisely detect finger bending. Comprehensive user interaction is made possible by the addition of pins designated for the calibration button, several control buttons, and the joystick axes. Additionally specified are pins for force feedback motors, which simulate resistance during finger movements to enable haptic feedback. The system provides precise sensor readings, dependable user inputs, and efficient motor control by carefully initializing these pins, setting the foundation for the sophisticated functionalities of the haptic glove. The code utilized for reading these analog values and processing them to track finger movements is based on a foundation from LucidVR. However, modifications were made to better suit the specific requirements of this project and to improve overall functionality and accuracy. These adaptations ensure that the system operates more reliably within the unique context of this haptic glove design. (LucidVR, Lucidglove, n.d.)

```
#if defined (ESP32)
  //(This configuration is for ESP32 DOIT V1 so make sure to change if you're on another
board)
  //To use a pin on the multiplexer, use MUX(pin). So for example pin 15 on a mux would be
MUX(15).
  #define PIN_PINKY     MUX(2) //These 5 are for flexion
  #define PIN_RING      MUX(5)
  #define PIN_MIDDLE    MUX(8)
```

```
  #define PIN_INDEX     MUX(11)
  #define PIN_THUMB     MUX(14)
  #define PIN_JOY_X     34
  #define PIN_JOY_Y     35
  #define PIN_JOY_BTN   32
  #define PIN_A_BTN     23
  #define PIN_B_BTN     22
  #define PIN_TRIG_BTN  41 //unused if gesture set
  #define PIN_GRAB_BTN  43 //unused if gesture set
  #define PIN_PNCH_BTN  23 //unused if gesture set
  #define PIN_CALIB     21 //button for recalibration (You can set this to GPIO0 to use
the BOOT button, but only when using Bluetooth.)
  #define DEBUG_LED 2
  #define PIN_PINKY_MOTOR     19  //used for force feedback
  #define PIN_RING_MOTOR      18 //^
  #define PIN_MIDDLE_MOTOR    5 //^
  #define PIN_INDEX_MOTOR     17 //^
  #define PIN_THUMB_MOTOR     16 //^
  #define PIN_MENU_BTN        4
```

6.2.2 Finger Tracking

Tracking finger movements is done with angular measurement entails applying trigonometric principles to precisely calculate finger positions. The system can determine the angle of flexion or extension by using sensors to measure the angular displacement of each finger segment. The hall effect sensors are positioned at 90 degrees to a setup where the output of one sensor can be related to the output of the other through trigonometric functions. This data is frequently processed with sine and cosine algorithms to turn angular observations into precise positioning information. The use of sine and cosine calculations allows for the precise computation of finger joint angles because these trigonometric functions are directly related to the geometry of the movement.

The first hall effect sensor(A) is aligned with the cosine function, and the second(B) is aligned with the sine function of the angle θ.

$$(1)\ \boldsymbol{VA = Vmax \cdot cos(\theta)}$$

$$(2)\ \boldsymbol{VB = Vmax \cdot cos(\theta)}$$

Were the Vmax being the max voltage of the sensor, in this case, is 5V.

To calculate the angle θ θ, use the inverse tangent function, which is based on the ratio of the two sensors' outputs:

$$(3)\ \boldsymbol{\theta = arctan(\frac{VA}{VB})}$$

By combining the sine and cosine components, you can precisely measure angles across a full 360-degree range. This provides better tracking of the finger than just using one hall effect sensor.

The function used for tracking of the fingers called `sinCosMix`:

```
int sinCosMix(int sinPin, int cosPin, int i){

  int sinRaw = analogPinRead(sinPin);
  int cosRaw = analogPinRead(cosPin);

  #if INTERFILTER_MODE != INTERFILTER_NONE
    sinSamples[i].add(sinRaw);
    cosSamples[i].add(cosRaw);
    int sinCalib = sinSamples[i].getMedian();
    int cosCalib = cosSamples[i].getMedian();
    #if INTERFILTER_MODE == INTERFILTER_ALL
      sinRaw = sinCalib;
      cosRaw = cosCalib;
    #endif
  #else
    int sinCalib = sinRaw;
    int cosCalib = cosRaw;
  #endif

  if (!savedInter){
    //scaling
    sinMin[i] = min(sinCalib, sinMin[i]);
    sinMax[i] = max(sinCalib, sinMax[i]);

    cosMin[i] = min(cosCalib, cosMin[i]);
    cosMax[i] = max(cosCalib, cosMax[i]);
  }

  if (i==target){
    targetSinMin = sinMin[i];
    targetSinMax = sinMax[i];
    targetSinCurrent = sinRaw;
    targetCosMin = cosMin[i];
    targetCosMax = cosMax[i];
    targetCosCurrent = cosRaw;
  }

  // Scaling of the raw sin and cos values to a range of -ANALOG_MAX to ANALOG_MAX
  int sinScaled = map(sinRaw, sinMin[i], sinMax[i], -ANALOG_MAX, ANALOG_MAX);
  int cosScaled = map(cosRaw, cosMin[i], cosMax[i], -ANALOG_MAX, ANALOG_MAX);


  //trigonometry stuffs
  double angleRaw = atan2(sinScaled, cosScaled);
  //counting rotations
  if (((angleRaw > 0) != atanPositive[i]) && sinScaled > cosScaled){
    totalOffset1[i] += atanPositive[i]?1:-1;
  }
```

```
  atanPositive[i] = angleRaw > 0;
  double totalAngle = angleRaw + 2*PI * totalOffset1[i];

  return (int)(totalAngle * ANALOG_MAX);
}
#endif
```

In this code, sinPin represents Hall effect sensor A ((1), whereas cosPin represents Hall effect sensor B ((2). The variable (i) denotes the finger number, with 0 corresponding to the thumb and 4 to the pinky. The calibration procedure is carried out in this section of the code, where the range of observed finger movements is evaluated and corrected. Calibration is assessing the range of sensor outputs and then normalizing it to a specified scale from 0 to analogMax (4095). This normalization guarantees that sensor values are precisely mapped within the intended output range, resulting in consistent and reliable finger position tracking across all fingers.

Normalization of sensor values:

```
  for (int i = 0; i < 2*NUM_FINGERS; i++){
  if (i == target){
    targetFlexionMin = minFingers[i];
    targetFlexionMax = maxFingers[i];
    targetFlexionCurrent = rawFingers[i];
    targetMaxTravel = maxTravel[i];
  }
    if (minFingers[i] != maxFingers[i]){
      fingerPos[i] = map( rawFingers[i], minFingers[i], maxFingers[i], 0, ANALOG_MAX );
      if (i == target)
        targetProcessed = fingerPos[i];
      #if CLAMP_ANALOG_MAP
        if (fingerPos[i] < 0)
          fingerPos[i] = 0;
        if (fingerPos[i] > ANALOG_MAX)
          fingerPos[i] = ANALOG_MAX;
      #endif
    }
    else {
      fingerPos[i] = ANALOG_MAX / 2;
    }

  }
```

Then, the angle is calculated with the scaled sin(sinScaled) and cos(cosScaled) ((3).

double angleRaw = atan2(sinScaled, cosScaled);

The total angle calculates the angular offset in radians. The raw angle (angleRaw) and an offset, scaled by 2*PI, are added to get the totalAngle. Multiplying 2*PI by totalOffset1[i] moves the raw angle by a number of full rotations, depending on the value of totalOffset1[i].

double totalAngle = angleRaw + 2*PI * totalOffset1[i];

6.2.3 Side-to-Side Movement Tracking

```
#if USING_SPLAY
  int rawFingersSplay[NUM_FINGERS] = {NO_THUMB?0:analogPinRead(PIN_THUMB_SPLAY),
                                    analogPinRead(PIN_INDEX_SPLAY),
                                    analogPinRead(PIN_MIDDLE_SPLAY),
                                    analogPinRead(PIN_RING_SPLAY),
                                    analogPinRead(PIN_PINKY_SPLAY)};
#else
  int rawFingersSplay[NUM_FINGERS] = {0,0,0,0,0};
#endif
```

An analog Hall effect sensor reading is used to precisely track a finger's side-to-side movement. Lateral finger motions generate fluctuations in the magnetic field, which the Hall effect sensor measures and converts into analog voltage values. The finger's splay angle is determined by the system by continuous reading of the analog data from the sensor using the function `analogPinRead`. I didn't utilize the thumb on this project due of certain design and connection issues I ran into (mentioned in chapter 5).

Real-time processing of the analog signals yields the finger's exact lateral location. In order to achieve high-resolution tracking and make sure that the virtual and real movements of the fingers match closely, this data is necessary. For applications that demand precise motion capture and haptic feedback, the continuous stream of analog data enables responsive and fluid feedback.

6.2.4 Flexion

```
bool grabGesture(int *flexion){
  return (flexion[PINKYIND] + flexion[RINGIND] + flexion[MIDDLEIND] + flexion[INDEXIND]) /
4 <= ANALOG_MAX/2 ? 0:1;
}
bool pinchGesture(int *flexion){
  return (flexion[INDEXIND] + flexion[THUMB_IND]) / 2 <= ANALOG_MAX/2 ? 0:1;
}

bool triggerGesture(int *flexion){
  return flexion[INDEXIND]<=(ANALOG_MAX/2)?0:1;
}
```

The above code implements three Boolean functions, `grabGesture`, `pinchGesture`, and `triggerGesture,` that use finger flexion values obtained from analog sensors to identify particular hand motions. The function `grabGesture` computes the average flexion of the middle, index, ring, and pinky fingers. If this average is greater than half of the maximum analog value (ANALOG_MAX/2), it is considered a grab gesture and returns true. In a similar manner, the `pinchGesture` function uses the thumb and index finger's average flexion to identify a pinching motion. Trigger motions are detected by the `triggerGesture` function, which only considers the flexion of the index finger. If the flexion of the finger is more than half of the maximum analog value, the function returns true. These functions handle the analog flexion data to enable the haptic glove and support real-time gesture detection.

6.2.5 Haptics

```
void setupServoHaptics(){
  pinkyServo.attach(PIN_PINKY_MOTOR);
  ringServo.attach(PIN_RING_MOTOR);
  middleServo.attach(PIN_MIDDLE_MOTOR);
  indexServo.attach(PIN_INDEX_MOTOR);
  thumbServo.attach(PIN_THUMB_MOTOR);
}

//static scaling, maps to entire range of servo, haptic input values (0 to 1000) to servo
angles (0 to 180 degrees)
void scaleLimits(int* hapticLimits, float* scaledLimits){
  for (int i = 0; i < 5; i++){
    #if FLIP_FORCE_FEEDBACK
    scaledLimits[i] = hapticLimits[i] / 1000.0f * 180.0f;
    #else
    scaledLimits[i] = 180.0f - hapticLimits[i] / 1000.0f * 180.0f;
    #endif
  }
```

In this part of the code, the servo motors utilized for haptic feedback are initialized by the SetupServoHaptics function by fastening each motor to the appropriate pin, which is indicated by constants like PIN_THUMB_MOTOR and PIN_INDEX_MOTOR. This configuration makes sure that every servo, which regulates the force feedback for a particular finger, is linked appropriately and prepared for use.

The servo angles, which vary from 0 to 180 degrees, are mapped to haptic input values, which span from 0 to 1000, using the scaleLimits function. The function makes sure that the servos correctly represent the desired haptic feedback, whether it be a direct or inverted reaction to the input data, by scaling and possibly inverting these values. Thus, this feature is essential to producing accurate and lifelike haptic experiences in the glove.

6.2.6 Buttons and Joystick

```
int getJoyX(){
  #if JOYSTICK_BLANK
  return ANALOG_MAX/2;
  #elif JOY_FLIP_X
  return ANALOG_MAX - analogReadDeadzone(PIN_JOY_X);
  #else
  return analogReadDeadzone(PIN_JOY_X);
  #endif
}
```

```
int getJoyY(){
  #if JOYSTICK_BLANK
  return ANALOG_MAX/2;
  #elif JOY_FLIP_Y
  return ANALOG_MAX - analogReadDeadzone(PIN_JOY_Y);
  #else
  return analogReadDeadzone(PIN_JOY_Y);
  #endif
}

bool getButton(byte pin){
  return digitalRead(pin) != HIGH;
}
```

Three methods in the given code are intended to manage button states and joystick input in the haptic glove system. Applying conditional logic based on compile-time flags, the `getJoyX` method retrieves the analog value from the joystick's X-axis. Joystick input is essentially neutralized if JOYSTICK_BLANK is specified, which yields the midway value of ANALOG_MAX/2. If JOY_FLIP_X is set, the axis is essentially flipped and the inverted joystick value is obtained by subtracting the dead zone-adjusted reading from ANALOG_MAX. Otherwise, the dead zone is adjusted and the X-axis measurement is provided. Comparable conditions are used by the `getJoyY` method to handle the Y-axis input, flipping or returning a neutral midway based on the provided flags. When a digital button is pushed, it indicates a low signal. If the button is not pressed, it returns false. This method, `getButton`, reads the button's state. This method makes sure that button and joystick inputs are precisely recorded and modified for different scenarios, enabling adaptable and snappy control over the system.

6.2.7 Encoding

- Flexion Encoding

```
char* encode(int* flexion, int joyX, int joyY, bool joyClick, bool triggerButton, bool
aButton, bool bButton, bool grab, bool pinch, bool calib, bool menu){
  static char stringToEncode[75];
  int trigger = (flexion[1] > ANALOG_MAX/2) ? (flexion[1] - ANALOG_MAX/2) * 2:0;
  #if USING_SPLAY
  sprintf(stringToEncode,
"A%dB%dC%dD%dE%dF%dG%dP%d%s%s%s%s%s%s%s%s(AB)%d(BB)%d(CB)%d(DB)%d(EB)%d\n",
  flexion[0], flexion[1], flexion[2], flexion[3], flexion[4],
  joyX, joyY, trigger, joyClick?"H":"",
  triggerButton?"I":"", aButton?"J":"", bButton?"K":"", grab?"L":"", pinch?"M":"",
menu?"N":"", calib?"O":"",
  flexion[5], flexion[6], flexion[7], flexion[8], flexion[9]
  );
  #else
  sprintf(stringToEncode, "A%dB%dC%dD%dE%dF%dG%dP%d%s%s%s%s%s%s%s%s\n",
  flexion[0], flexion[1], flexion[2], flexion[3], flexion[4],
  joyX, joyY, trigger, joyClick?"H":"",
```

```
  triggerButton?"I":"", aButton?"J":"", bButton?"K":"", grab?"L":"", pinch?"M":"",
menu?"N":"", calib?"O":""
  );
  #endif
```

The encode function successfully converts raw sensor and control data into a structured format, allowing for straightforward communication and data management. The function serves a variety of applications by grouping this information into a string, such as real-time data monitoring, system integration, and performance analysis. This organized method guarantees that complicated sensor readings and control states are appropriately represented, allowing for more effective data handling and increasing the overall functioning and usability of the haptic gloves system. This method encodes several arguments into a formatted character string, including flexion values from various fingers, joystick locations, button statuses, and gesture indications. To support various system setups, the code snippet uses conditional. During the encoding process, several sensor readings and control states are combined into a readable string by formatting the data using sprintf and computing a trigger value based on flexion thresholds.

Parameters of Function:

- The flexion levels of the fingers are represented by an array of integers called int* flexion.
- The joystick's X and Y coordinates are represented by the integer values int joyX, int joyY.
- The variables bool grab, bool pinch, bool calib, bool menu, bool joyClick, bool triggerButton, bool aButton, bool bButton, and other Boolean flags denoting the states of different buttons and motions.

The "trigger" variable is calculated to represent the degree of bending of a particular finger. A scaled value representing the intensity of flexion beyond this threshold is calculated if the flexion value is more than half of the maximum analog value (ANALOG_MAX/2). That in this case is 2047.

The data is formatted into a stringtf(stringToEncode) using the sprintf function (Image 6-1).

- Flexion values for various fingers (A through E and, if applicable, F through J) are included in the prepared string.
- Positions of the joystick (F and G).
- The trigger value that was calculated.
- different boolean flags that, under certain situations, are displayed as single letters.

After the encoding is done, the prepared string can be used for data transfer, logging, or additional processing after the method delivers the pointer to it. In that instance, the encoding data is transferred in Unity and LucidVR Driver (Lucid VR driver).

- Haptic Encoding

```
  hapticLimits[0] = getArgument(stringToDecode, 'A'); //thumb
  hapticLimits[1] = getArgument(stringToDecode, 'B'); //index
  hapticLimits[2] = getArgument(stringToDecode, 'C'); //middle
  hapticLimits[3] = getArgument(stringToDecode, 'D'); //ring
  hapticLimits[4] = getArgument(stringToDecode, 'E'); //pinky
}

int getArgument(char* stringToDecode, char command){
  char* start = strchr(stringToDecode, command);
  if (start == NULL)
    return -1;
  else
    return atoi(start + 1);
}
```

This code is used for haptic in the VR Glove, where an MG90S servo motor is installed in each finger. The encoded information for each finger's haptic reaction is contained in the data string that is delivered to the VR environment as OpenVR or Unity. A unique character is used to represent each finger, for example, the letter 'A' designates the thumb, 'B' designates the index finger, and so on.

For example (Image 6-1), if the received string `getArgument` ("A68B40C70D10E20", 'A') it will locate the character 'A', proceed to the following segment of the string ("20"), and return 20 as the thumb's value (represented by 'A'). The function `decodeData`() later on will interpret it as 68 units of force for the thumb, 40 for the index finger, 70 for the middle finger, and so forth. These values are then saved in the `hapticLimits`[] array, which is later utilized to operate the servo motors linked to each finger, mimicking the required force feedback through virtual interactions.

```
    //Send the finger positions and button states over the communication interface
    //comm->output(encode(fingerPosCopy, getJoyX(), getJoyY(), joyButton, triggerButton,
aButton, bButton, grabButton, pinchButton, calibButton, menuButton));
    #if USING_FORCE_FEEDBACK
      char received[100];
      if (comm->readData(received)){
        int hapticLimits[5];
        if(String(received).length() >= 5) {
           decodeData(received, hapticLimits);
           writeServoHaptics(hapticLimits);
        }
      }


void writeServoHaptics(int* hapticLimits){
  float scaledLimits[5];
  scaleLimits(hapticLimits, scaledLimits);
  if(hapticLimits[0] >= 0) thumbServo.write(scaledLimits[0]);
```

```
  if(hapticLimits[1] >= 0) indexServo.write(scaledLimits[1]);
  if(hapticLimits[2] >= 0) middleServo.write(scaledLimits[2]);
  if(hapticLimits[3] >= 0) ringServo.write(scaledLimits[3]);
  if(hapticLimits[4] >= 0) pinkyServo.write(scaledLimits[4]);
}
```

To start, the VR application sends a string of encoded data to the system. This data tells the system how much force each servo should put on the associated finger. The `writeServoHaptics` function is invoked to operate the servo motors after decoding. However, the decoded data must first be scaled to fit the motors' working range, which is in this case is between 0 and 180 degrees of rotation. This is handled by the `scaledLimits` function mentioned in subchapter 6.2.5. After scaling, the `writeServoHaptics` method uses `Servo.write` () to transmit the modified values to the finger servos, setting each servo to the required angle. The whole process, from receiving the data to decoding it, scaling it for the servos, and writing the output to the motors, ensures that the user receives dynamic and realistic force feedback.

```
A2047B24C59D0E41F733G1061P0N(AB)2047(BB)780(CB)3778(DB)819(EB)2970
A2047B24C59D0E41F727G911P0N(AB)2047(BB)780(CB)3778(DB)819(EB)2970
A2047B24C59D0E41F720G862P0N(AB)2047(BB)877(CB)3778(DB)819(EB)2970
A2047B24C59D0E41F714G834P0N(AB)2047(BB)975(CB)3778(DB)1638(EB)2970
A2047B24C59D0E41F736G1094P0N(AB)2047(BB)877(CB)3778(DB)1638(EB)2970
A2047B24C59D0E41F735G930P0N(AB)2047(BB)877(CB)3778(DB)1638(EB)2970
A2047B24C59D0E41F725G877P0N(AB)2047(BB)780(CB)3778(DB)1638(EB)2970
A2047B24C59D0E41F719G842P0N(AB)2047(BB)877(CB)3778(DB)1638(EB)2970
A2047B24C59D0E41F740G1101P0N(AB)2047(BB)780(CB)3778(DB)1638(EB)2959
A2047B24C59D0E41F739G944P0N(AB)2047(BB)780(CB)3778(DB)1638(
```

Image 6-1 Encoding Example taken from Arduino IDE Serial monitor

6.2.8 Serial and Bluetooth Communication

- Serial Communication

```
  public:
    SerialCommunication() {
      m_isOpen = false;
    }
    bool isOpen(){
      return m_isOpen;
    }
    void start(){
      //Serial.setTimeout(1000000);
      Serial.begin(SERIAL_BAUD_RATE);
      m_isOpen = true;
    }
    void output(char* data){
      Serial.print(data);
      Serial.flush();
    }
```

The SerialCommunication class provides capabilities for handling serial communication in an embedded system. Within its public section, the class has methods for controlling and verifying the status of the serial connection. The SerialCommunication () constructor creates an instance of the class with the m_isOpen flag set to false, indicating that the serial port is initially closed. The function isOpen() returns the state of this flag, allowing other components to determine if the serial connection is active.

The start() function starts serial connection by calling Serial.begin() with a predetermined baud rate (SERIAL_BAUD_RATE) and setting m_isOpen to true, indicating that the serial port is now open and available for data transfer.

The Serial baud rate is set at 115200 because strikes an appropriate compromise between speed and dependability. It is quick enough to handle a significant quantity of data transfer without overloading the system's processing capacity or creating severe data loss or corruption. In addition, using Serial.print() to send a string of characters to the serial port and Serial.flush() to guarantee that all data is transferred, the output(char* data) method transmits data. This process is essential to guaranteeing complete and dependable data transmission. When combined, these techniques offer a simplified interface for controlling serial connection, which is essential for embedded programs' data interchange and system integration.

- Bluetooth Commutation

```
#include "BluetoothSerial.h"
  private:
    bool m_isOpen;
    BluetoothSerial m_SerialBT;

  public:
    BTSerialCommunication() {
      m_isOpen = false;
    }

    bool isOpen(){
      return m_isOpen;
    }
    void start(){
      m_SerialBT.begin(BTSERIAL_DEVICE_NAME);
      m_isOpen = true;
    }
    void output(char* data){
      m_SerialBT.print(data);
      }

```

The class named BTSerialCommunication manages Bluetooth connection between a VR Haptic Glove and another Bluetooth-enabled device.

Two of the private elements are m_SerialBT, which manages low-level communication between the device and a linked Bluetooth, and m_isOpen, a Boolean flag that monitors the connection state. The fundamental function that starts Bluetooth communication is the `start` () method. To link a Bluetooth device with an external device, it calls m_SerialBT.begin(BTSERIAL_DEVICE_NAME), where BTSERIAL_DEVICE_NAME is the name of the device. The device name is set on the Initialization part of the code (mentioned in 6.2.1). By using m_SerialBT.print(data), which sends the supplied string to the linked Bluetooth device, the `output` () function is intended to deliver data via Bluetooth.

This technique makes it possible for the class to transfer data wirelessly, which is useful for providing real-time data to a linked device, such a computer, smartphone, or other microcontroller, such as sensor readings or haptic feedback orders.

6.3 Unity

This section explains how the VR Glove is integrated with Unity and how finger-tracking simulation is created and carried out. The basic functionality centers around receiving serial data, decoding it into useful flexion values, and using those values to control the rotations of finger bones in a 3D model.

6.3.1 Serial Communication

The first step is connecting the hardware with the virtual hand model, allowing it to read flexion values in real time. To transmit data, the script first establishes a serial connection with an external device, in this case the VR glove, configuring it with a certain portName (such as COM3) and baud rate that is 115200. The serial communication is controlled using Unity's SerialPort class, ensuring that the connection is correctly created during setup (`void OpenConnection`) and gracefully terminated when the program stops (`void OnApplicationQuit`). The first objective of the `Update` () method is to read and check data from the serial port on a constant basis. Within the function `Update` is the function `ReadData`, which checks whether the serial port is open and whether any data is available for reading. Then, the decoding of the flexion values is utilized.

```
private SerialPort serialPort;
  public string portName = "COM3"; // Change this to your actual port name
  public int baudRate = 115200;
  public bool autoConnect = true;

  void Start()
  {
      if (autoConnect)
      {
          OpenConnection();
      }
  }

  void Update()
```

```
    {
        ReadData();
        // Update the rotation of each finger bone based on the flexion values
        UpdateFingerRotation(thumb, thumbFlexion);
        UpdateFingerRotation(index, indexFlexion);
        UpdateFingerRotation(middle, middleFlexion);
        UpdateFingerRotation(ring, ringFlexion);
        UpdateFingerRotation(pinky, pinkyFlexion);
    }

    private void OpenConnection()
    {
        try
        {
            serialPort = new SerialPort(portName, baudRate);
            serialPort.Open();
            Debug.Log("Serial port opened: " + portName + " at " + baudRate + " baud.");
        }
        catch (Exception e)
        {
            Debug.LogError("Failed to open serial port: " + e.Message);
        }
    }

    void ReadData()
    {
        if (serialPort != null && serialPort.IsOpen && serialPort.BytesToRead > 0)
        {


                string data = serialPort.ReadLine();
                DecodeString(data);

        }
    }

    private void OnApplicationQuit()
    {
        CloseConnection();
    }

    private void CloseConnection()
    {
        if (serialPort != null && serialPort.IsOpen)
        {
            serialPort.Close();
            Debug.Log("Serial port closed.");
        }
    }
```

6.3.2 Decoding the String

The encoded string format indicates that the incoming data is presumed to follow a predetermined pattern (Image 6-1): "A68B40C70D10E20" (`pattern = @"A(\d+)B(\d+)C(\d+)D(\d+)E(\d+)";`), where each segment (thumb, index, middle, ring, and pinky) reflects a flexion value for a particular finger. The integer values between the letters are extracted using the `DecodeString` () method. These integers range from 0 to 4095, recorded by Hall effect sensors. By using the formula int.Parse(value) / 4095f, the parsing logic converts these values into a normalized range (0 to 1), enabling the raw sensor data to be proportionately transferred to Unity's 3D environment and guaranteeing constant movement scaling.

```
  void DecodeString(string input)
    {
        string pattern = @"A(\d+)B(\d+)C(\d+)D(\d+)E(\d+)";
        Regex regex = new Regex(pattern);
        Match match = regex.Match(input);

        if (match.Success)
        {
            thumbFlexion = ParseFlexion(match.Groups[1].Value);
            indexFlexion = ParseFlexion(match.Groups[2].Value);
            middleFlexion = ParseFlexion(match.Groups[3].Value);
            ringFlexion = ParseFlexion(match.Groups[4].Value);
            pinkyFlexion = ParseFlexion(match.Groups[5].Value);
            Debug.Log("A:" + thumbFlexion + "B:" + indexFlexion + "C:" +middleFlexion
+"D:" +ringFlexion+"E:" + pinkyFlexion);
        }
        else
        {
            Debug.Log("Failed to decode the string.");
        }
    }
    float ParseFlexion(string value)
    {
        // the flexion value is between 0 and 2047, normalize it to a range of 0 to 1
        return int.Parse(value) / 4095f;
    }
}
```

6.3.3 Finger Tracking

Following their decoding, the flexion data are transferred to certain finger bones inside the virtual model (Image 6-2). The `UpdateFingerRotation` function applies the current flexion value (for example, thumbFlexion) to the associated bone (thumb) by interpolating its rotation. Vector3() is used to interpolate between the initial rotation (Vector3 zero, indicating no flexion) and the maximum rotation (maxFingerRotation), set to 90 degrees on the z-axis that represents a completely bent finger. The `Lerp`() guarantees that the flexion value rises from 0 to 1 when the finger gradually bends.

As a consequence, the simulated fingers bend in sync with the real-world flexion, enabling real-time monitoring of finger motions.

```
 public Transform thumb;
    public Transform index;
    public Transform middle;
    public Transform ring;
    public Transform pinky;
    // Flexion values for each finger
    [Range(0, 1)]
    public float thumbFlexion = 0;
    [Range(0, 1)]
    public float indexFlexion = 0;
    [Range(0, 1)]
    public float middleFlexion = 0;
    [Range(0, 1)]
    public float ringFlexion = 0;
    [Range(0, 1)]
    public float pinkyFlexion = 0;
    // The maximum rotation for finger flexion
    public Vector3 maxFingerRotation = new Vector3(0, 0, 90);
 void UpdateFingerRotation(Transform finger, float flexion)
    {
        if (finger != null)
        {
            finger.localRotation = Quaternion.Euler(Vector3.Lerp(Vector3.zero,
maxFingerRotation, flexion));
        }
    }
```

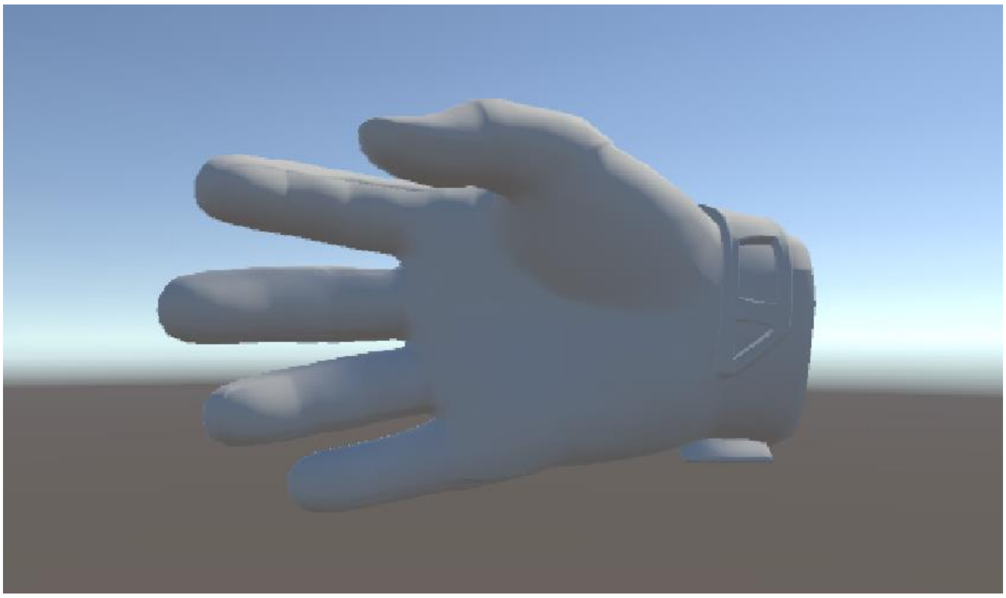


Image 6-2 3D Hand Model through Unity

The first step in integrating the hand model with the control code is to confirm that each finger bone in the 3D hand model is accurately mapped to the corresponding variable in the Unity script. To enable the code to access and dynamically modify each bone's rotation, which corresponds to the thumb, index, middle, ring, and pinky fingers, must be assigned to the relevant Transform fields in the Unity Inspector. After, the script decodes real-time flexion values from a serial input, converts them to normalized float numbers (0 to 1), and then updates the associated finger's rotation with the given values. In this way, the 3D hand model can replicate the user's hand's actual movements, with bone rotations in the virtual space precisely reflecting each finger's bending. The flexion values are standardized between 0 and 1, signifying the degree of bend in each finger. The thumb, index, middle, ring, and pinky fingers are represented by the values A through E. Through dynamic reading from the serial port, processing, and real-time updating of the virtual hand, these encoded data enable precise finger tracking and the encoding of actual hand movements into virtual gestures.

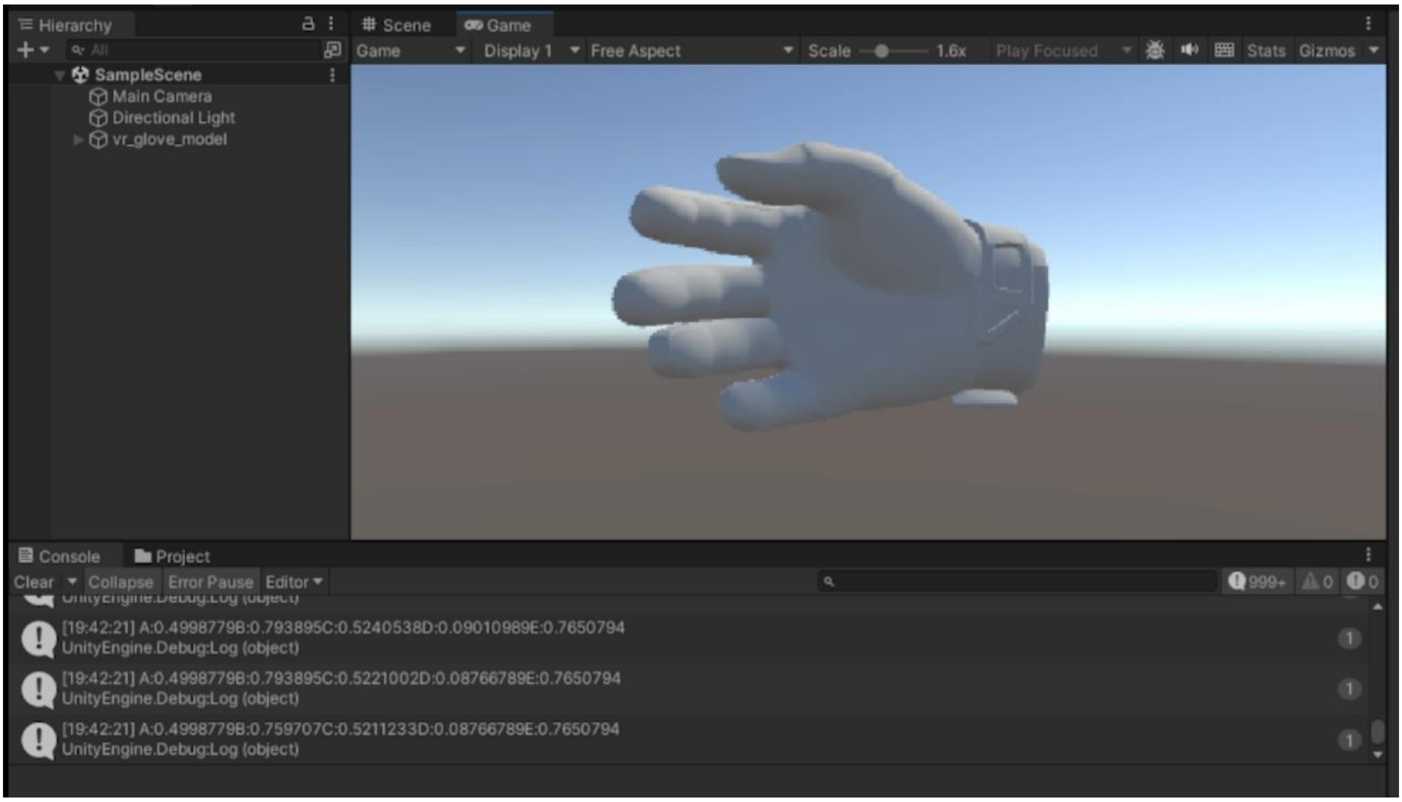


Image 6-3 Unity Environment and Console Logs

6.4 Lucid VR driver

The original idea behind this project was to create a customized VR haptic glove that would work as an add-on for an already-existing VR tracking system or tracker controller, like the Oculus Controller or HTC Vive Tracker. Under this method, the VR controller would control the hand tracking, and the specially designed haptic glove would control the finger tracking. This idea is based on the open-source LucidVR driver, which OpenVR developed and is accessible on Steam, which is called OpenGloves driver (Image 6-4 OpenGlove Software on Steam Store page (LucidVR & danwillm, Steam, n.d.)Image 6-4).

Haptic Glove for Virtual Reality Simulation

Custom hardware may be integrated with a range of VR controllers using OpenGloves, enabling the augmentation of existing hand-tracking technologies with unique finger-tracking systems. By using this method, this integration makes it easier to concentrate on exact finger motions and haptic feedback, which improves the immersive experience.

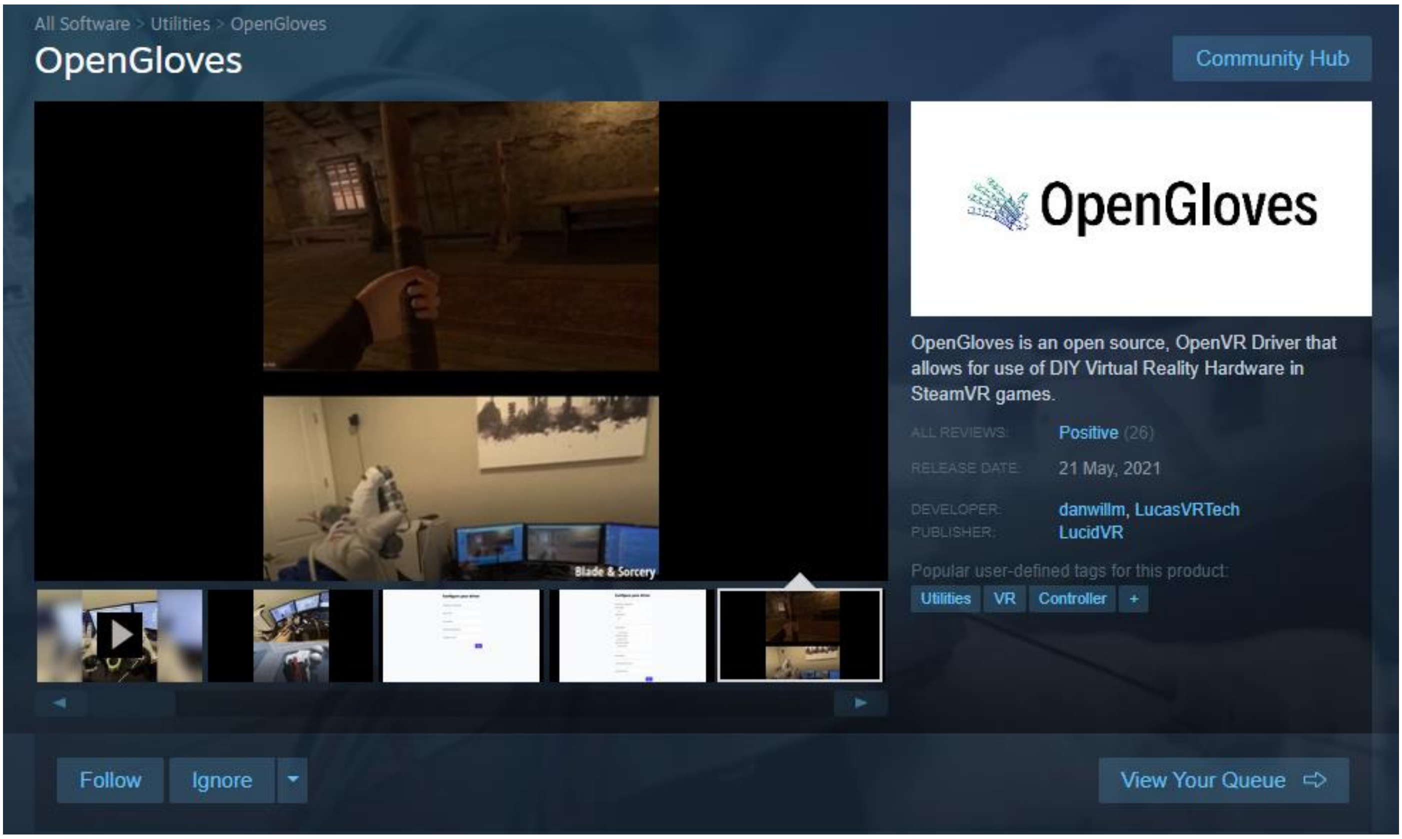


Image 6-4 OpenGlove Software on Steam Store page (LucidVR & danwillm, Steam, n.d.)

VR tracking controllers already can handle complicated positional tracking with various sensors and algorithms, thus adding a bespoke haptic glove as an addition leverages this existing infrastructure. Similar to LucidVR, OpenVR's driver approach allows programmers to create unique drivers that integrate with the VR ecosystem. In this way, the glove functions as an add-on that sends force feedback and finger position data into the same pipeline that conventional VR controllers utilize. This is accomplished by translating the bespoke hardware's inputs (hall effect sensor inputs) into the current SteamVR input system. After this configuration is finished, the driver builds a set of input elements that map to the buttons, triggers, and haptics on the controller and enable bespoke inputs from the glove. The driver additionally manages posture adjustments by utilizing the controller's orientation and location in relation to the VR headset, ensuring that the bespoke glove inputs are registered in sync with the controller's motions. Moreover, the opengloves driver activates the glove's haptic feedback system, combining it with the haptic portion of the VR controller.

When a haptic event happens, such as touching or grabbing a virtual item, the driver detects it via the event queue (e.g., vr::VREvent_Input_HapticVibration) and initiates the appropriate reaction in the glove, resulting in an immersive and tactile VR experience. Through the use of the SteamVR framework, the driver connects these two systems, making sure they operate together harmoniously to deliver an accurate and rich virtual reality experience.

A companion program called OpenGloves UI (User Interface) is included with the driver on Steam and is meant to give users an easy-to-use interface for controlling several driver-related parameters. The UI capabilities include controlling driver configurations, testing force feedback capability, and automatically calibrating the offsets between the controller and glove. This program lets users set up and fine-tune the settings for the haptic glove and controller integration. This guarantees seamless synchronization between the controller and glove, improving the VR experience's overall accuracy and personalization.

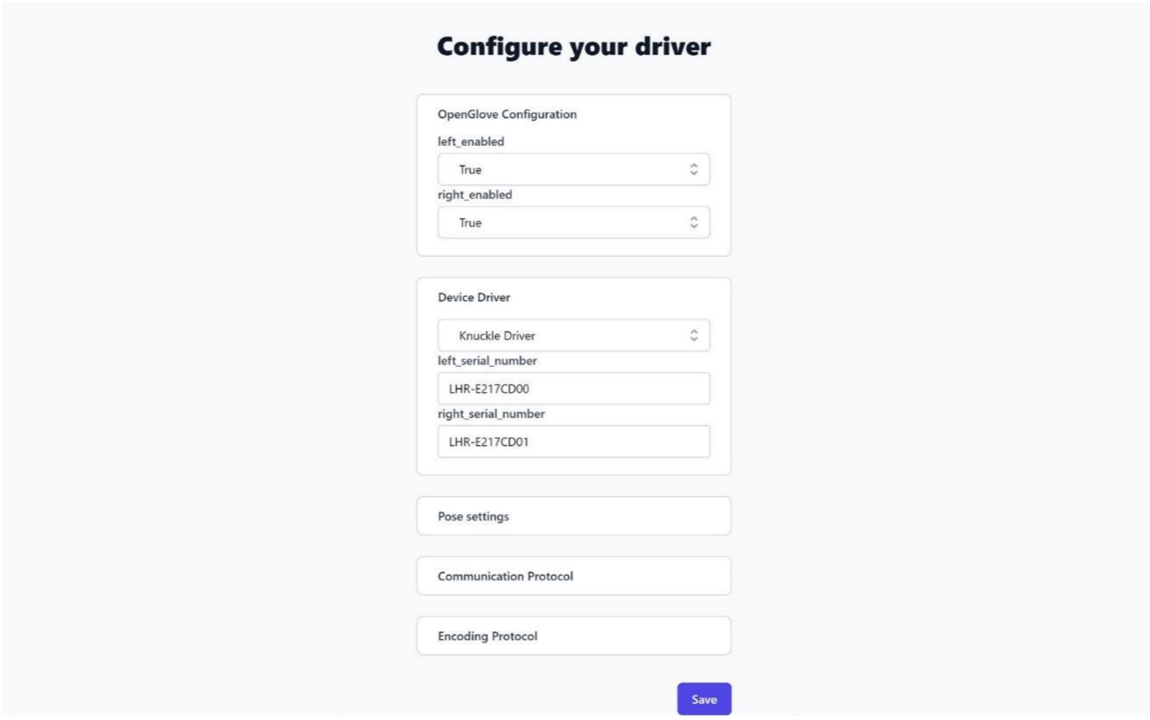


Image 6-5 OpenGloves UI/ Application for configuring the driver (LucidVR & danwillm, Steam, n.d.)

Lastly, an example of Force Feedback combined with the SteamVR Unity Interaction Example is available in the OpenGloves UI. This presentation shows how haptic feedback may be used practically in a virtual reality setting, giving users the ability to feel tactile reactions in real time that match virtual interactions. Through this interface, developers and users may assess and optimize the haptic performance of the glove, guaranteeing that the feedback it provides is in precise alignment with the actions and feelings shown inside Unity's VR framework.

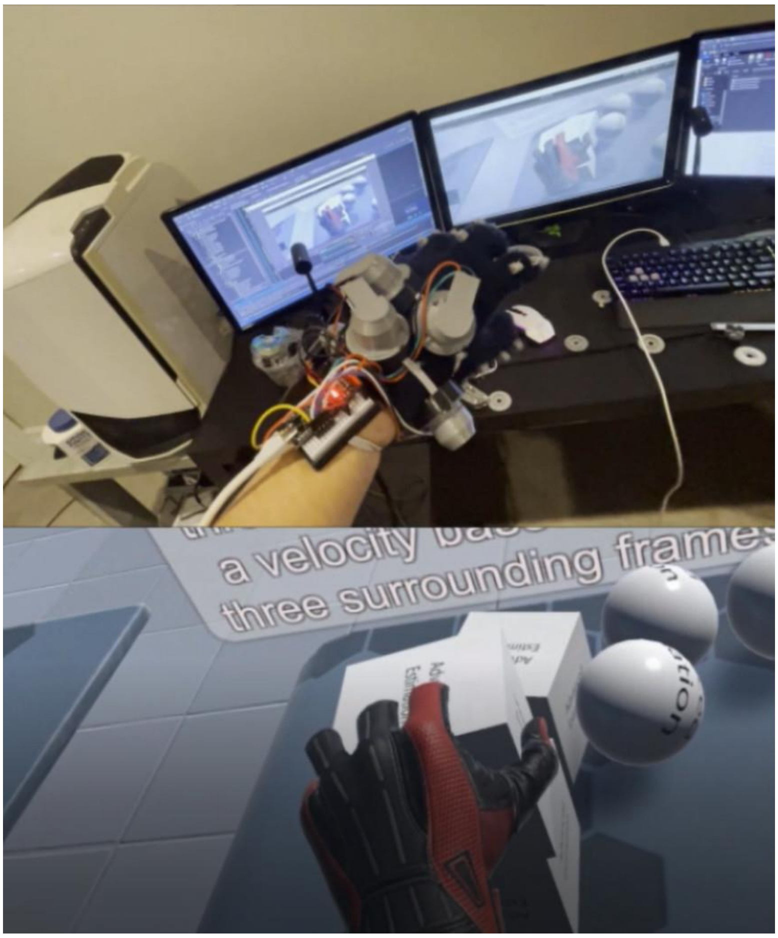


Image 6-6 VR Haptic Glove in OpenGloves Unity Force Feedback Example (LucidVR & danwillm, Steam, n.d.)

Image 6-7 Unity Force Feedback example connected with SteamVR

In the course of experimentation with the LucidVR glove and how SteamVR integrated with it, I encountered some problems. First of all, continuous disconnections between the glove and SteamVR were faced.

This issue may be caused by a number of things, such as erratic Bluetooth connectivity, problems with firmware compatibility, or software incompatibilities within SteamVR itself. Furthermore, during testing the haptics using the Unity force feedback example, the Oculus Quest 3 controller that was used overrode the finger tracking capability even if the haptic feedback was working as intended. This issue may be due to potentially competing data streams or mapping anomalies, this unexpected behavior raises the possibility of conflicts between the input signals from the Oculus controller and the finger tracking data from the LucidVR glove. The finger tracking functioned well when tested separately with the Arduino IDE and in Unity, which is interesting since it suggests that there could be a compatibility problem between the custom glove and the OpenGlove driver in the SteamVR environment. These difficulties highlight the difficulty of integrating proprietary VR gear with current platforms and the requirement for exact calibration and compatibility testing in order to ensure smooth operation.

Image 6-8 SteamVR Application with connected the custom glove, VR headset and controller

Chapter 7: Problems and Solutions

7.1 Introduction

Several technical and design obstacles developed throughout the project, ranging from hardware constraints and sensor errors to software integration difficulties and communication inefficiencies. To guarantee the project's success, each difficulty presented a different set of challenges that needed to be carefully analyzed, experimented with, and adjusted. This chapter presents the main issues that arose at various points in the project and describes the approaches and strategies that were employed to address them.

7.2 Technical issues

In order to emphasize the iterative nature of development and the significance of problem-solving in engineering practice, the discussion will include concrete instances of challenges, the effects they had on the project, and the techniques utilized to overcome them.

7.2.1 Parts

- **Sensor Pins**

  Hall Effect Sensors are delicate components. When these sensor pins are exposed to high temperatures or mechanical stress during assembly or operation, many issues can occur.

  Mechanical stress can damage electrical connections, resulting in intermittent or full sensor functioning loss. If the Hall effect sensors are not correctly secured during the assembly process or during regular operation, or because of physical stress that can cause the opening and closing of fingers, the pins can experience mechanical strain. Bending repeatedly over time might wear out the pins and finally cause them to break.

  Electronic components, like Hall effect sensors, have certain **Temperature** ranges within which they may perform successfully. Excessive heat during the soldering procedure may pose a risk to the Hall effect sensor pins. The pins may soften if the soldering iron is heated too much or if the heat is applied for an extended period of time. Also, to secure the Hall effect sensor in place, hot glue was used. On the other hand, applying the glue at too-high temperatures or in too-large quantities may cause the sensor pins to experience extra stress and even weaken the sensor.

  Inconsistent sensor performance can cause unpredictable behavior, poor user experience, or even system failures. Therefore, many hall effects were changed throughout the testing of the glove. To address these issues, is secure Hall effect sensors are secured with sufficient support and operating temperature within safe limits. For example, adding less hot glue and avoiding bending the pins in assembly process of the glove. The best approach is, to reduce the strain on the sensor pins, and make design modifications. This could include creating circuit boards for the sensors.

- **Connectors**

  One of the significant challenges we encountered in the development of the haptic VR glove involved the stability and reliability of the connectors used in the system. In particular, the problems were that the connectors lost their steady connections, which resulted in **frequent disconnections** and inconsistent performance. The issue presented itself in several ways, for example, the cables frequently came loose, the metal connector crimps would move or separate from the connector, and generally, the glove's performance was impaired when in use. Crimping is the mechanical process of connecting a wire to a connection by deforming the connector's metal tabs around the wire. For there to be a steady and reliable connection, this procedure must be carried out with extreme accuracy. For example, the connections used in the glove's design call for a particular kind of crimping tool called a wire crimper (Image 7-1), which is made to evenly deliver the right amount of pressure to each pin. The connections could not establish a secure connection if the crimping is not done with the right tool or if the tool is not calibrated appropriately. Frequent use or movement of the glove may cause sporadic or unstable connections where the wires are prone to coming free. Furthermore, even when crimped correctly, the low quality of the connectors themselves made the issue worse by making it difficult to establish a firm connection.

Image 7-1 TUBTAP Crimping Tools for Electrical Connectors (TUBTAP, n.d.)

To address this problem, the solution involved investing in high-quality connectors and precise crimping equipment. For instance, JST connectors with clamps would make a perfect replacement in this specific application. By avoiding unintentional disconnections, JST connectors with clamps provide a reliable locking mechanism that aids in the maintenance of solid connections. Specifically, wire-locking connectors were utilized, such as the Molex Micro-Lock series (Image 7-2) with a lock mechanism. The locking on these connectors securely keeps the connection in place, reducing the possibility of disconnections and guaranteeing solid electrical interconnections.

Image 7-2 Molex Micro-Lock Plus (Tyler, 2018)

Furthermore, connectors with wire-locking capabilities and sturdy construction, such as the TE Connectivity Mini-PV series and the Hirose DF13 series, were taken into consideration due to their exceptional durability and dependability. This makes them an even better choice because even under physical strain, these connections are made to stop wire movement and guarantee that the crimps stay firmly in place.

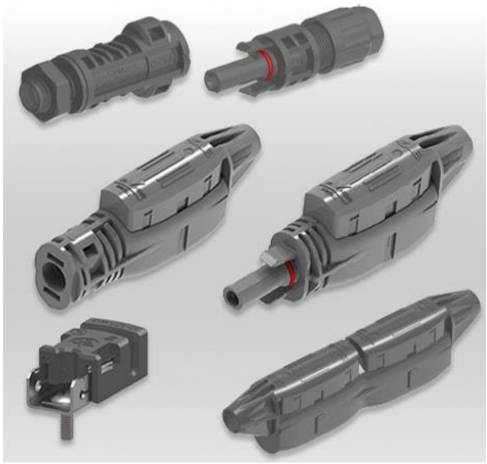


Image 7-3 SOLARLOK Connectors or TE Connectivity Mini-PV (Connectivity, n.d.)

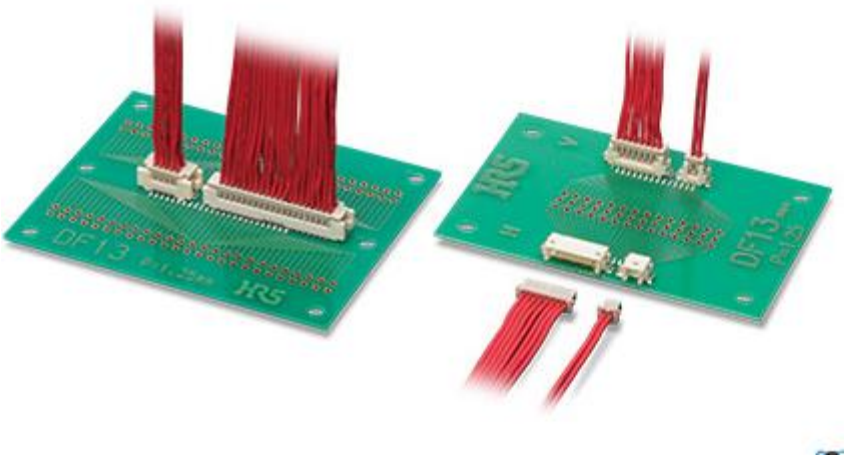


Image 7-4 DF13 Series SignalBee™ 1.25mm Pitch Miniature Crimping Connectors (Hirose, n.d.)

- **Additive Fabrication**

  The inflexibility of the finger racks (Image 4-11) and hand mount (Image 4-13), which frequently broke and made it impossible for the device to properly fit the user's hand and fingers, was one of the major problems faced during the creation of the haptic VR glove. The main cause of the problems was the **usage of stiff materials**. In order to resolve these challenges, the usage of flexible PLA (polylactic acid) in the design and manufacture of the finger racks and hand mounts is a remedy. PLA may be made more flexible without sacrificing enough structural integrity to provide the required adaptability. The natural flexibility of this material allowed it to better fit and accept a variety of hand shapes and sizes without sacrificing durability, which was crucial for enhancing the glove's comfort and fit.

Image 7-5 FLASHFORGE PLA Flexible 1.75 mm, 4X Longer Stretch 3D Printer Filament 1Kg (Amazon, n.d.)

7.2.2 Design and Practicality

- **Splay**

  The restricted finger movement in the "splay" motion, that is a side-to-side movement of the fingers away from or towards each other, was another problem during the creation of the haptic VR glove. The glove's capacity to precisely track finger movements during calibration was greatly impacted by this restriction in finger splay. Poor calibration was the outcome of the design's restricted range of motion, which prevented the system from accurately registering the entire range of finger separation or convergence. The primary reason for this problem was the glove's hand mount's stiffness and the overall design, which limited finger mobility laterally and decreased the tracking system's overall accuracy. To address this, the solution was redesigning the glove and especially the hand mount. Allowing additional room between the fingers and minimizing structural limitations allowed the fingers to move naturally in a side-to-side splay motion.

The glove redesign would increase flexibility and improve not only tracking accuracy but also the user's comfort and the overall realism of the haptic feedback since it better matched normal hand movement.

- **Practicality**

One other key issue encountered was its physical design, which lacked practicality and ease of usage. The glove has multiple stiff components and exterior cables, making it difficult to put on and remove. The exposed wiring, paired with frequent movement, caused durability issues, with the connectors getting loose and finally detaching or falling apart. This jeopardized the glove's long-term dependability and raised the need for regular maintenance.

The arrangement of the power supply was yet another significant flaw. The glove required a large external power bank to operate, which had to be physically connected via a cable. This setup severely limited the user's range of motion and freedom within the virtual environment, as the cable created a constant tethering point. Also, having to control both the glove and the connected power supply increased the risk of tripping or tangling. This configuration demonstrated how crucial wireless operation, portability, and adaptability are to the design of VR technology that puts the user first. Resolving these problems, the redesigning of the glove was crucial.

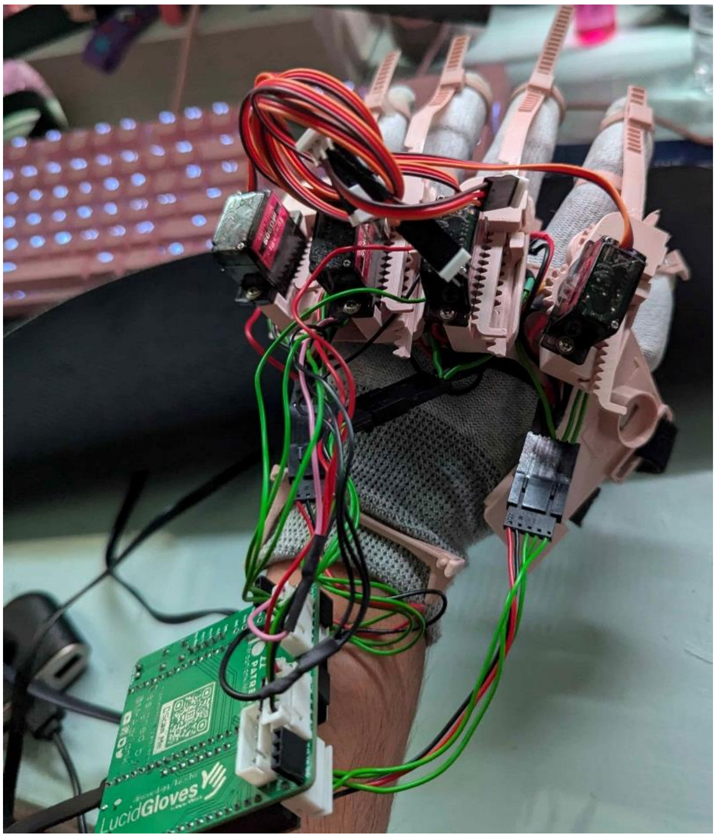


Image 7-6 Loose Wire of the VR haptic glove

7.2.3 Difficulties

- **Limited availability of electronic components**

  Our limited access to electronic components and manufacturing services in Greece was a big issue. The project needed a range of sensors, actuators, and microcontrollers, but local suppliers only had a small supply. This limitation significantly hindered our capacity to test various parts, and modify our design in real-time and additional obtaining certain parts was time-consuming. The development timeline was harmed by waiting for components to be shipped from overseas, making the glove unable to move as swiftly as was intended. The lack of printed circuit board (PCB) manufacturing facilities in the area made the problem worse. In order to create prototypes and finish the project, PCB design, fabrication, and modification are essential steps in the process. As a result, the design, production, and testing were expensive and prolonged.

- **Lack of Equipment**

  Critical testing, which entails real-time interaction with the VR environment, could not be carried out due to a lack of access to VR hardware. Particularly, there were significant limitations because the VR headset and controller could only be acquired two months before the thesis deadline.

  This restriction made it more difficult to fine-tune the glove's performance in real time and guarantee that it delivered responsive and accurate haptic feedback. The tracking system of the VR controller and the glove's sensors must precisely line up for the haptic feedback system to be calibrated. Due to the last-minute availability of the VR hardware, there was little time for repeatedly fine-tuning and perfecting these calibrations, which could have affected the glove's functionality.

- **Glue Fragility**

  The hot glue's capacity to firmly attach the glove's parts was compromised by the high temperatures, which made the glue softer and less sticky. Furthermore, the hot glue's fragility under continual movement and hand flexing exacerbated the situation. The repetitive motion of the user's hand, along with the softened adhesive, resulted in mechanical failure, causing components to shift or detach during use, compromising the glove's performance and user experience. To overcome this issue, one option was to investigate different adhesive methods that are more thermally stable and mechanically durable was one option. To give an example, employing epoxy resins or silicone-based adhesives offers superior heat resistance and greater durability under stress.

- **Short circuit**

  During the development phases of the glove project, a severe raised problem with the ESP32 microprocessor. Serious damage was caused by the high current that passed through the voltage regulator and the ESP32 because the pins of a Hall effect sensor became short. The short circuit problem resulted in an excess of current, which ultimately produced overheating. The voltage regulator started to show signs of thermal degradation, being hot to the touch and eventually burning out. This causes many Hall effects sensors to burn out and connectors to melt under the extensive heat. Current-limiting resistors, adequate insulation and isolation of sensitive components, and the addition of protective circuitry to defend against unintentional short circuits or reverse current flows are some preventive measures for such problems. Additionally, the damaged components had to be replaced, and the circuit design had to be revised to incorporate precautions against future occurrences of the same problems.

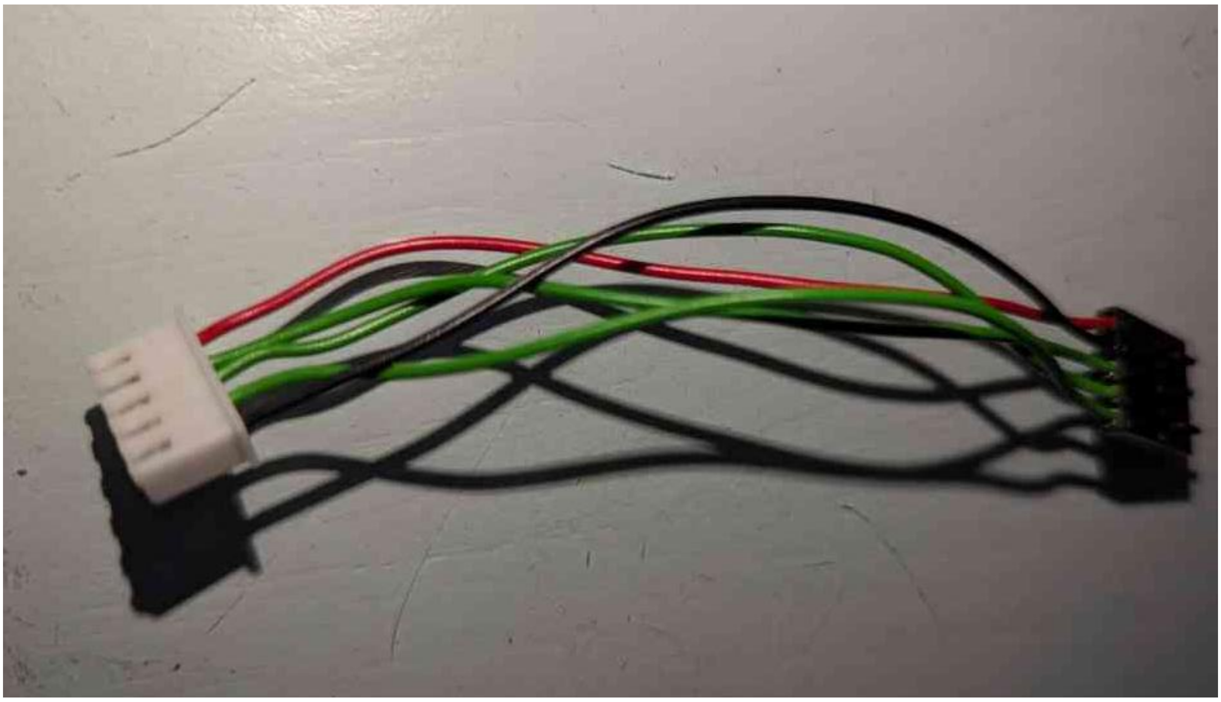


Image 7-7 New wired connectors

7.3 Conclusion

To sum up, the successful resolution of these issues highlights the significance of thorough testing, careful design, and proactive problem-solving in the engineering field. The experiences detailed in this chapter attest to the development process's tenacity and the ongoing pursuit of innovation in the creation of cutting-edge technology.

Chapter 8: Conclusion

8.1 Introduction

This chapter, which concludes the thesis, offers a thorough summary of all the research done on the creation of the haptic VR glove. It seeks to summarize the main conclusions, assess how well the solutions were executed, and consider how the project would advance the field of haptic technology. It will also discuss the research findings' wider importance and how they can affect wearable and virtual reality technology developments. Along with highlighting the lessons discovered and acknowledging the difficulties faced, the debate will also suggest future directions for research and improvement. The goal of this last chapter is to give a comprehensive, yet concise, summary of the research findings and their significance, as well as a last look at how the project will further haptic virtual reality technology.

8.2 Contributions to existing knowledge

This subsection outlines the noteworthy additions made by this thesis to the corpus of knowledge about haptic technology and virtual reality. Several significant breakthroughs that go beyond present methods and provide fresh perspectives on wearable haptic systems have been made possible by the creation of the haptic VR glove.

First off, by combining servo motor control systems and cutting-edge sensor technology, the project has improved haptic feedback mechanisms. The utilization of MG90S servos in conjunction with hall effect sensors to provide accurate force feedback has enhanced the virtual interactions' realism and responsiveness, thereby establishing a new standard for haptic devices intended for general consumers.

The use of angular measurement techniques to improve finger-tracking accuracy is a remarkable addition to this thesis. The tracking of finger motions has been greatly enhanced by using the sine and cosine functions (sin and cos mixing) for angular computations. This technique provides increased accuracy in capturing the complex dynamics of finger movements, resulting in haptic feedback that is more responsive and accurate.

Additionally, the research has made significant progress toward addressing and resolving typical issues related to the construction of haptic VR gloves. Problems with power management and hardware failures were discovered and dealt with methodically. For example, the circuitry was redesigned with preventive measures to address the issue of short circuits damaging the voltage regulator and ESP32 microprocessor. This experience emphasizes how crucial it is to have strong design and testing procedures in place to avoid problems of this nature in future projects.

What is more, a critical component of this project has been creating a Unity application for testing and calibration. Real-time finger-tracking performance evaluation and visualization are made possible by this specially designed application.

The glove's tracking accuracy and responsiveness have been thoroughly evaluated and fine-tuned thanks to the Unity application, which offers a controlled testing environment for haptic feedback scenarios and different finger movements. This tool has been shown to be crucial for continuously improving the glove's functionality and guaranteeing dependable operation in virtual environments.

Overall, this thesis' discoveries enhance the technological capabilities of haptic feedback systems and lay a strong foundation for future advancements in wearable and virtual reality technologies, making them an important addition to the area. The improvements covered in the discussion, such as the improved finger tracking method, hardware problem fixes, and the creation of a testing Unity application, provide insightful information and useful tools for haptic VR research, development, and practical implementation in the future.

8.3 Future work

This chapter describes possible enhancements and modifications that could be applied in upcoming project iterations. Future research will concentrate on incorporating more recent technology and looking for ways to maximize usability and performance.

An interesting route for future research is to update the haptic VR glove project by including the Arduino Nano 33 BLE microcontroller. Compared to the ESP32 that is already in use, this microcontroller has a smaller, more compact form factor, which makes it perfect for designing gloves that are more streamlined and ergonomic. Apart from its compact design, the Arduino Nano 33 BLE also has Bluetooth Low Energy (BLE) and Wi-Fi capabilities, enabling more dependable and effective wireless connectivity. Furthermore, the device's integrated 9-axis inertial measurement unit (IMU) facilitates excellent motion tracking, giving the opportunity for the detection of hand orientation without using a VR controller. Specifically, it features integrated environmental sensors, such as a barometer, accelerometer, gyroscope, and magnetometer, which can track hand position, pressure, and movement with greater accuracy. However, complex software will be needed to fine-tune with a virtual reality environment.

Another intriguing improvement to the haptic VR glove is the replacement of the Oculus controller with a Vive Tracker for hand tracking. A more specialized and small option is provided by the Vive Tracker, which was created especially for accurate positional tracking in virtual environments without the need for other input controls. Also, the Oculus controller, as mentioned in 7.2, overrides the finger tracking and glove due to these extra inputs, especially when navigating with the joystick in virtual reality. The issue of input conflicts would be fixed by integrating the Vive Tracker, which focuses solely on tracking hand location, enabling more precise and reliable finger tracking free from undesired interference. Another option would be to create a custom inertial measurement unit (IMU) board just for hand tracking.

With complete control over the tracking system, this method would provide more adaptability and modification in the tracking algorithms. On the other hand, this would need creating the software as well as the hardware to convert unprocessed IMU data into accurate hand tracking inside the virtual environment. Although it would offer a long-term, highly customized tracking solution, the project would become more complex due to the requirement for bespoke firmware, intensive testing, and interface with current VR systems.

One notable and very important development is the introduction of separate finger modules, each with its own sensor PCBs. The sensors on each finger may be directly attached to its own PCB, thanks to this modular design, which reduces the need for lengthy external cabling, which sometimes results in tangled, loose connections that are prone to breakage and short circuits. The glove's overall reliability is increased because of the simpler assembly procedure and less wiring complexity, which also minimizes error risk. In this way, each module can be individually checked and calibrated to make sure all the parts are working correctly and to lower the possibility of problems during final assembly. Additionally, because the glove system is modular, it is simpler to replace or maintain specific parts without compromising the whole system.

The introduction of a completely new, compact control PCB represents a significant advancement as compared to just switching to a different microcontroller, such as the Arduino Nano 33 BLE. The ESP32-S3 chip would be a good integration into the suggested new PCB design, providing cutting-edge technologies like enhanced memory, increased processing power, and compatibility with Bluetooth Low Energy (BLE) and Wi-Fi. The approach simplifies wiring, removes the need for extra chips like multiplexers, and lowers the overall complexity of the glove by combining the control functions onto a single, small PCB. However, the ESP32-S3 is a very small microcontroller that presents a lot of difficulties for soldering. Because of the chip's tiny pin spacing and fine pitch, manual soldering is challenging and frequently calls either specialized tools or a reflow soldering machine. Also, using software like KiCad to create a PCB using the ESP32-S3 chip has its own set of challenges. In order to guarantee that the board operates as intended and satisfies all performance requirements, proper placement, routing, and signal integrity control are essential.

Instead of depending on an external, bulky power bank connected by a cable, future versions of the haptic VR glove will incorporate rechargeable batteries and a battery management system (BMS). The incorporation of a small, rechargeable battery pack into the glove would enable users to move around with more mobility without being restricted by wires. In addition, the BMS would be in charge of guarding against short circuits, overcharging, and overheating of the batteries. In this way, extending the battery pack's lifespan, and dependability and guarantees the glove's steady operation over prolonged use.

Redesigning some of the haptic VR glove's components and using different materials for printing, including flexible PLA, are crucial areas for further research.

As mentioned in subchapter 7.2.1, important parts that have a direct impact on how well the glove fits and how the user feels are the hand mount and finger rack. An improvement on the design of the hand mount would be breaking the mount into five pieces for each finger and gluing them directly on the glove to mimic the hand's natural movement. This would make also the splay tracking, side to-side finger tracking, more accurate and natural because there would be more space between fingers (more mentioned in subchapter 7.2.2).

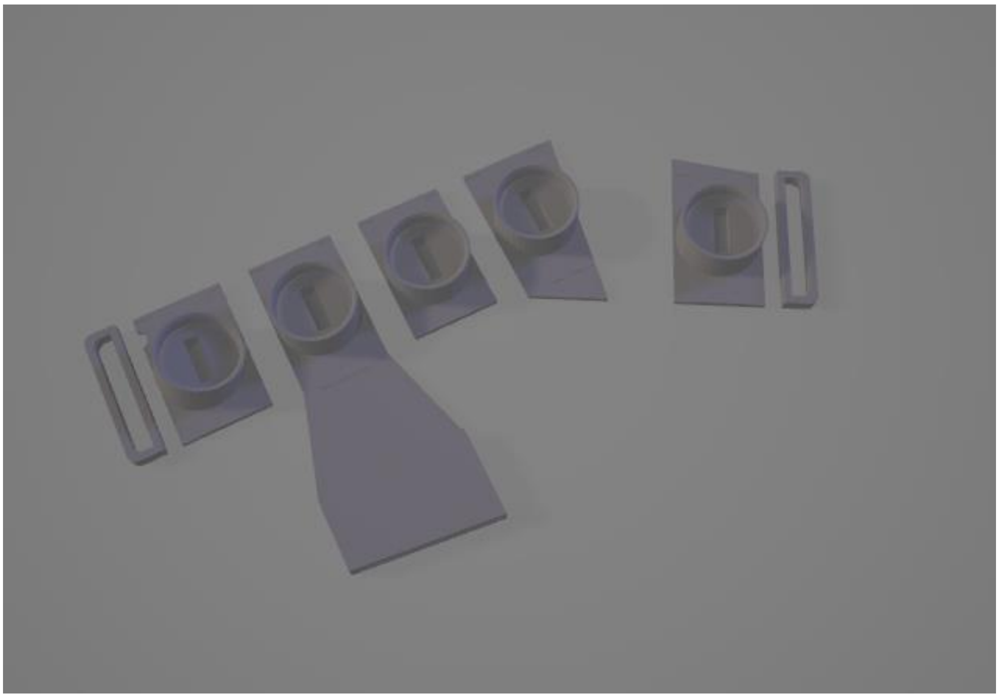


Image 8-1 New rigid mount for more comfortable use and operation (GitHub, n.d.)

Developing specific mounts for the PCB and joystick will be a crucial step in improving the haptic VR glove design going forward since it now has problems with these parts hanging freely from the glove. The present configuration frequently leaves the PCB and joystick exposed or with their attachments loose, which can cause a number of issues. Because of their insecure positioning, the parts are more vulnerable to damage from snagging, unintentional pulls, and physical impacts. Furthermore, it's possible that the hanging parts will obstruct the user's normal motions, which would be uncomfortable and less immersive. For PCB holders (Image 8-2), SOL75 (SOL75, n.d.), provides an expedient solution by letting customers input their board measurements into an online customizer, which then outputs a downloadable STL file for 3D printing. This technology streamlines the process of designing a custom enclosure for the control board, assuring a precise fit that firmly keeps the PCB in place while protecting it from external harm. The joystick's design would be inspired by the Ring Joystick Holder (Image 8-3) seen on Thangs (Thangs, n.d.). An ergonomic way to attach the joystick directly to the glove without causing undue bulk or limiting movement is via a ring-style holder. Additionally, the ring holder would lessen the possibility that the joystick might shift or hang loose while being used, which could cause pain or problems.

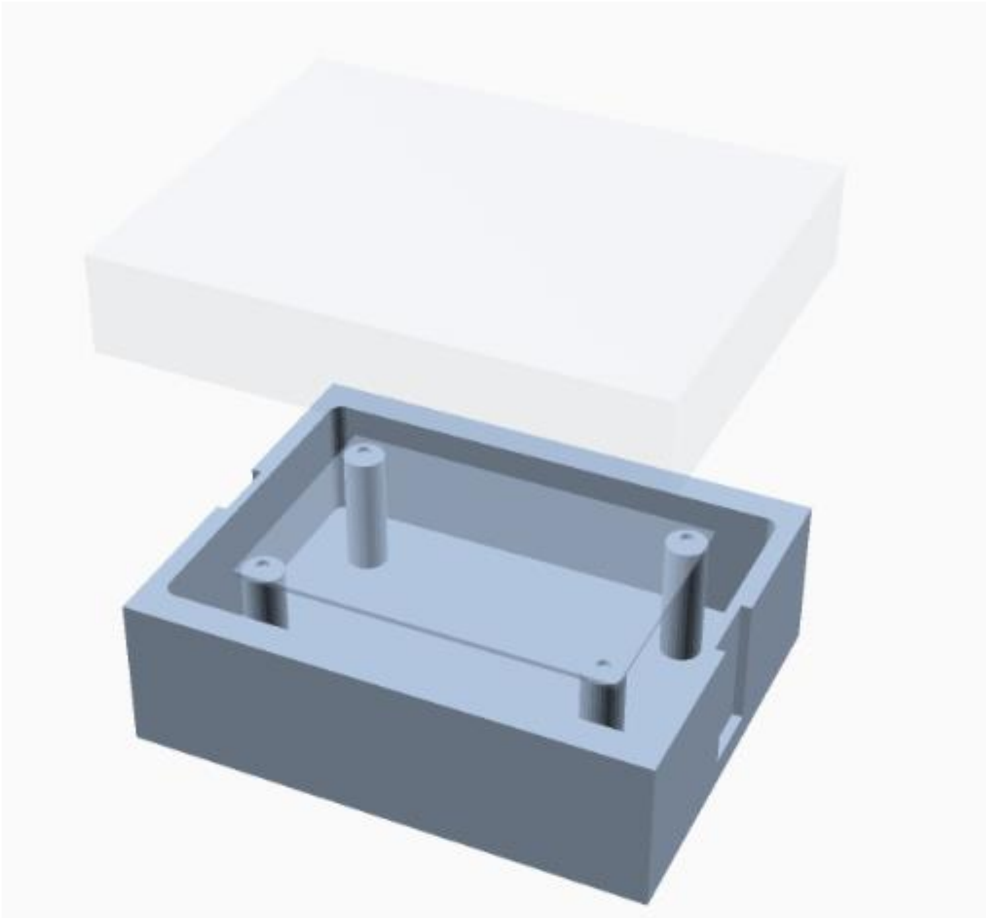


Image 8-2 PCB Case Holder (Instrucables, n.d.)

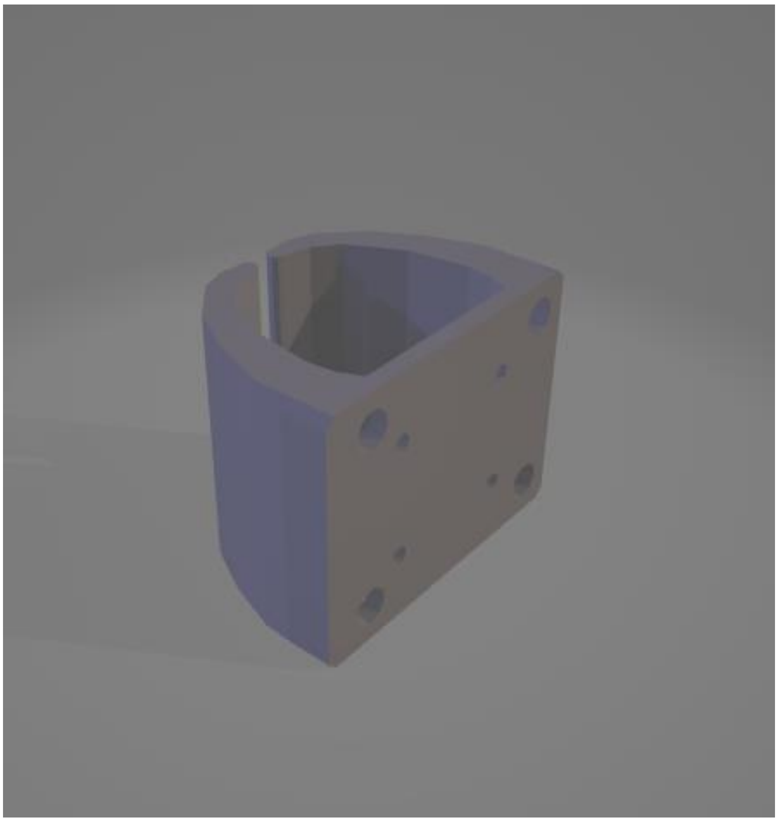


Image 8-3 Ring Joystick Holder (Thangs, n.d.)

Haptic Glove for Virtual Reality Simulation

An important future goal for the haptic VR glove project is to construct a custom glove driver from scratch, moving away from dependency on the existing LucidVR driver. Although the driver for LucidVR has been very helpful in the early stages of development, it has been difficult and time-consuming to configure and troubleshoot it to function with bespoke hardware. Creating a new driver from scratch, It permits complete customization, allowing the software to be adjusted for the glove's unique hardware design and performance features. While it would certainly take a lot of time and effort to create a new driver from scratch, the long-term advantages of flexibility and scalability would be worthwhile.

Last but not least, extending the haptic VR glove's use beyond virtual reality (VR) and into other extended reality (XR) contexts, such as mixed reality (MR) and augmented reality (AR), is another exciting avenue for research. For instance, in AR applications for education or training, users might grab and handle virtual tools or objects while receiving feedback. In this way, the gap between physical and virtual interactions would be bridged, making augmented reality experiences more natural and engaging. For MR, in an industrial or design scenario, users might interact with 3D models and manipulate them in real time while receiving haptic input that replicates roughness or resistance. In domains like advanced design prototyping, collaborative MR environments, or remote surgery, where the real and virtual worlds must interact, this would be especially helpful. Moreover, the glove may be used in areas like remote collaboration and telepresence. More specifically, participating in real-time design collaboration where all participants, no matter where they are physically located, experience the same virtual object's texture and force.

There are many chances to enhance the haptic VR glove's usability, add more features, and make it more adaptable for a range of extended reality applications in the future. The glove may become more dependable, comfortable, and user-friendly by concentrating on essential improvements such as incorporating rechargeable batteries, replacing critical components with modern materials, and fastening electrical parts with specialized holders. These advances will not only solve present limits but will also establish the glove as a potent weapon in the expanding realm of extended reality, assuring its relevance and flexibility to both existing and upcoming technology.

Chapter 9: References

AllDataSheet. (n.d.). *ALLDATASHEET.COM*. Retrieved from Electronic Parts Datasheet Search: https://html.alldatasheet.com/html-pdf/473135/SECELECTRONICS/SS49E/1174/1/SS49E.html

Amazon. (n.d.). *Amazon*. Retrieved from https://www.amazon.com/FLASHFORGE-Flexible-Printer-Filament-Spool/dp/B09V7NMG55?ref_=ast_sto_dp&th=1

Amazon. (n.d.). *Amazon.co.uk*. Retrieved from https://www.amazon.co.uk/Bimetal-10x15x4mm-Miniature-Bearings-Shielded/dp/B0CMT2QXFW

Andrews, C. S. (2019, March 30). Extended reality in medical practice. *Current Treatment Options in Cardiovascular Medicine*.

Angelov, V. (2020, June). Modern virtual reality headsets. *2020 International Congress on Human-Computer Interaction*.

Arth, C., Grasset, R., Gruber, L., Langlotz, T., Wagner, D., & Mulloni, A. (2015, May). The history of mobile augmented reality. *arXiv*.

Autodesk. (n.d.). *Autodesk*. Retrieved from https://www.autodesk.com/products/fusion-360/overview?term=1-YEAR&tab=subscription

BestArduino.com. (n.d.). *BestArduino.com*. Retrieved from https://www.bestarduino.com/p2736/ESP32-DevKitC-core-board-ESP32-development-board-ESP32-WROOM-32D.html

Cableworks. (n.d.). *Cableworks.gr*. Retrieved from https://www.cableworks.gr/ilektronika/arduino-and-microcontrollers/prototyping/connectors/jst-xh2.54-2p-2pin-connector-male-female/

Chimdi, A., & Sherri, Q. (2024, March 3). C2S2. Retrieved from https://c2s2.engineering.cornell.edu/blogposts/SP24/chooseFighter.html

Christoph, A. (2023, September 12). State of the art of virtual reality technology. *Leibniz Supercomputing Centre*.

CNET. (2023). Retrieved from https://www.facebook.com/cnet, https://www.cnet.com/pictures/the-ghost-of-vr-past-the-long-history-of-techs-latest-darling/8/

Connectivity, T. (n.d.). *TE Connectivity*. Retrieved from https://www.te.com/en/products/brands/solarlok.html?tab=pgp-story

DigiKey. (n.d.). *DigiKey*. Retrieved from https://www.digikey.gr/en/products/detail/w%C3%BCrth-elektronik/61900611121/17737789

Tsetserukou, D., & Hasegawa, S. (n.d.). LinkTouch: A wearable haptic device with five-bar linkage mechanism. *Electronics-Inspired Interdisciplinary Research Institute (EIIRIS)*.

Elaine, T. (2016, October). GARP virtual reality. Retrieved from http://etsanggarp.blogspot.com/2016/03/

Electronics, M. (n.d.). *Mouser Electronics*. Retrieved from Molex: https://www.mouser.co.uk/new/molex/molex-sl-modular-connectors/

GitHub. (n.d.). Retrieved from https://github.com/codingcatgirl/lucidgloves-fae-mod

Giri, G. S., & Manohar, Y. (2021). An application-based review of haptics technology. *Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada*, 307.

Grobotronics. (n.d.). *Grobotronics*. Retrieved from https://grobotronics.com/pin-header-1x40-female-2.54-mm.html#group_47266310352ce4d6-1

Grobotronics. (n.d.). *Grobotronics.gr*. Retrieved from https://grobotronics.com/pin-header-1x40-male-2.54-mm-black.html

Haidar, A. M., & Benachaiba, C. (2013, March). Software interfacing of servo motor with microcontroller. *Research Online, University of Wollongong*. Retrieved from https://ro.uow.edu.au/eispapers/468/

Hirose. (n.d.). Retrieved from https://www.hirose.com/en/product/series/DF13

Hsieh, M.-C., & Lee, J.-J. (2017). *Journal of Nursing and Health Studies*, 1–3.

Ibrahim, D., Ramsden, E., Huddleston, C., Ball, S., & Wilson, J. (2008, September). Test and measurement. In *Know It All*. Newnes.

Instrucables. (n.d.). Retrieved from https://www.instructables.com/3D-Printed-Electronic-Enclosure/

Javatpoit. (n.d.). Retrieved from https://www.javatpoint.com/arduino-ide

Jessica, Y., Ronan, H., Herbert, S., & Carmel, M. (2020). Wearable soft technologies for haptic sensing and feedback. *Advanced Functional Materials*, 2. https://doi.org/10.1002/adfm.202007428

Jiao, S. (2011). Familiarity with virtual reality usage positively impacts learning performance in undergraduate students. *Tong University*.

Kevuru Games. (2023, March 15). Retrieved from https://kevurugames.com/blog/unity-vs-unreal-engine-pros-and-cons/

Khan, T., Johnston, K., & Ophoff, J. (2019). The impact of an augmented reality application on learning motivation of students. *Advances in Human-Computer Interaction*.

Khundam, C. (2021, September). A comparative study of interaction time and usability of using controllers and hand tracking in virtual reality training. *Informatics no. 3, MDPI AG*, 60.

Knoxlabs. (n.d.). Retrieved from https://www.knoxlabs.com/products/weart-touchdiver

Lab.IT, M. (n.d.). *Makers Lab.IT*. Retrieved from https://www.makerslab.it/cd74hc4067-16-channel-analog-multiplexer-demultiplexer/

LaValle, S. M. (2003). History of virtual reality.

Louvre. (2019, October 23). *Louvre.fr*. Retrieved from https://www.louvre.fr/en/explore/life-at-the-museum/mona-lisa-beyond-the-glass-the-louvre-s-first-virtual-reality-experience

Alfa Electronics Co. Ltd. (n.d.). *Alfa Electronics Co. Ltd.*. Retrieved from http://www.alfaelecs.com/en/list.asp?classid=28

LucidVR. (n.d.). Lucidglove. Retrieved from Github: https://github.com/LucidVR/lucidgloves

LucidVR, & danwillm. (n.d.). Retrieved from Steam: https://store.steampowered.com/app/1574050/OpenGloves/

Magnitech. (n.d.). *Magnitech.gr*. Retrieved from https://magnitech.gr/?gad_source=1&gclid=CjwKCAjwrcKxBhBMEiwAIVF8rKj5cQ5gf9LIoxTI9wRFpOHNzuPuSAxL7rRT9Dr_2GrIP8jSWVlNkBoCuB4QAvD_BwE

W. D. (2024, March). *MG90S Micro Servo Motor Datasheet*. *Components101*. Retrieved from https://components101.com/motors/mg90s-metal-gear-servo-motor

Pape, D. (1999, October 4). A VPL Research DataSuit, a full-body outfit with sensors for measuring the movement of arms, legs, and trunk. Developed circa 1989. Displayed at the Nissho Iwai showroom in Tokyo.

Paro, M. R., Bulsara, K., & Hersh, D. (2022, April). History of virtual reality and augmented reality in neurosurgical training. *UConn School of Medicine, Farmington, Connecticut, USA*.

Parveau, M., & Adda, M. (2020, June). Toward a user-centric classification scheme for extended reality paradigms. *Journal of Ambient Intelligence and Humanized Computing, 11*(1). https://doi.org/10.1007/s12652-019-01352-9

PC Games for Steam. (n.d.). Retrieved from https://pcgamesforsteam.com/what-is-steam

Pinto, F. M. (2010, June). Multimodal access to social media services. Thesis for: MSc Informatics Engineering.

Ramsden, E. (2011). *Hall-effect sensors*. Elsevier, 2.

Robert, S. (2022, August 3). Tech Target. Retrieved from WhatIs.com: https://www.techtarget.com/whatis/definition/virtual-reality

Rolland, J., & Hong, H. (2024, March 4). Head-mounted display systems.

RS. (n.d.). *nz.rs-online.com*. Retrieved from https://nz.rs-online.com/web