CAPSTONE PROJECT

ON

VR HAPTIC GLOVES

<u>Project Team Members</u>: EE, 7th SEMESTER

Lakshay Sablok	102004086
Zenith	102004111
Melvin Bansal	102004114
Ojas Nagta	102184009

Under the Guidance of
Dr. Parag Nijhawan
Associate Professor
Electrical & Instrumentation Engineering
Thapar Institute of Engineering & Technology

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

2023

Electrical & Instrumentation Engineering Department

Thapar Institute of Engineering and Technology

(Declared as Deemed-to-be-University u/s 3 of the UGC Act., 1956)

Post Bag No. 32, Patiala – 147004

Punjab (India)

DECLARATION

We hereby declare that the project entitled **VR Haptic Gloves** is an authentic record of our own work carried out in the Electrical & Instrumentation Engineering Department, Thapar Institute of Engineering and Technology, Patiala under the guidance of **Dr. Parag Nijhawan** Associate Professor during July-December 2023.

Date: **10/12/2023**

S. No.	Name of the Student	Roll No.	Signature
1	Lakshay Sablok	102004086	Lakshay Sablok
2	Zenith	102004111	Zenith
3	Melvin Bansal	102004114	Melvin
4	Ojas Nagta	102184009	Ojas N.

Project Report Mentor:

Dr. Parag Nijhawan
Associate Professor
Electrical & Instrumentation Engineering
Thapar Institute of Engineering & Technology

Parag

Signature

I thoroughly reviewed the project report, and students amended it in accordance with my recommendations.

Faculty Supervisor(s):	Name & Designation	Signature

ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude to all those who have contributed towards the successful completion of the project and converge thanks to our supervisor **Dr. Parag Nijhawan** and all the faculty and staff members of Electrical and Instrumentation Engineering Department (EIED), Thapar Institute of Engineering & Technology for generously extending their support and for sparing their valuable time to guide us towards the completion of this project work.

We would like to place a special word of appreciation to **Dr. Vineet Srivastav** and **Dr. Faheeem Khan** for their guidance in allowing us to use their **laboratories**. Also, we would like to thank all other respondents and group members whose responses and coordination were of utmost importance for the completion of this project work.

S. No.	**Name of the Student**	**Roll No.**
1	Lakshay Sablok	102004086
2	Zenith	102004111
3	Melvin Bansal	102004114
4	Ojas Nagta	102184009

ABSTRACT

With the development of virtual reality technology, VR gloves began to popularize various fields widely. Some of the force feedback VR gloves at this stage still have problems, such as high cost and complex structure. The mechanical force feedback VR glove proposed in this paper uses the principle of mechanical rotation, using a potentiometer as the rotating shaft and a nylon rope to drive the potentiometer to rotate so that it records the relevant data instead of sensors, and uses a servo to bring force feedback experience. Experiments indicate that the force feedback glove in this paper has both the force feedback effect of the existing force feedback VR gloves and is lower in cost than the existing data gloves and has good generalizability (Lin and Bin 2022, 60).

Furthermore, the proposed mechanical force feedback VR glove offers advantages such as ease of use and maintenance due to its simple structure. Using a potentiometer as a rotating shaft eliminates the need for additional sensors, which can further reduce the cost of the device. Additionally, the nylon rope used to drive the potentiometer's rotation is a durable and easily replaceable material. These features make the glove an attractive option for a wide range of applications, including virtual reality gaming, rehabilitation, and industrial training. Overall, the proposed mechanical force feedback data glove presents a promising solution to the current challenges facing force feedback data gloves and has the potential to greatly enhance the immersive experience of virtual reality.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

VR haptic gloves are a wearable device that aims to provide users with a more immersive and interactive VR experience. They work by using sensors and actuators that are embedded in the gloves to offer users haptic feedback, allowing them to feel and interact with virtual objects as if they were real.

The haptic feedback is achieved by using different types of actuators that can create a variety of sensations, including vibrations, pressure, and temperature. The sensors in the gloves can detect the position and movement of the user's hands, allowing for accurate and responsive tracking of the user's movements.

The gloves are typically connected to a computer or VR headset using Bluetooth or a wired connection. This connection allows the device to communicate with the VR software, allowing it to provide haptic feedback that is synchronized with the user's actions in the virtual environment.

One of the key benefits of VR haptic gloves is that they can provide users with a more natural and intuitive way to interact with virtual objects (DesignsRock Editorial 2019). This can make it easier to learn new skills, practice complex tasks, and explore virtual environments in a more engaging and immersive way.

There are many potential applications for VR haptic gloves across various industries. For example, in gaming, haptic gloves can enhance the realism of virtual environments by allowing users to feel the weight and texture of virtual objects. In education, haptic gloves can be used to create immersive learning experiences, allowing students to explore scientific simulations or historical events more engagingly. In healthcare, haptic gloves can be used to train medical professionals in surgical procedures or to simulate patient interactions in a safe and controlled environment.

Overall, VR haptic gloves are a powerful tool that can provide users with a more immersive and realistic VR experience. As technology advances, we will likely see more applications of haptic gloves in a wide range of industries, providing users with even more engaging and interactive experiences.

1.2 LITERATURE ANALYSIS

One significant focus is integrating haptic feedback within VR environments. Studies such as that by Schorr et al. (2020) emphasize the pivotal role of haptic feedback in enhancing the sense of presence and realism within virtual worlds. This has led to exploring how VR haptic gloves can mimic the sensation of touch, pushing the boundaries of immersive experiences.

In the realm of sensory perception, researchers like Kim and Kim (2019) have examined the intricate interaction between haptic interfaces and human sensory perception. Their work delves into the nuances of creating realistic touch simulations, which are crucial for making VR environments more lifelike and responsive to users' actions.Ergonomic and mechanical design considerations also occupy a substantial portion of the literature. Ribeiro et al. (2015) highlight the importance of creating VR haptic gloves that are comfortable, well-fitted, and easy to wear for extended periods. Designing gloves that seamlessly integrate with the human hand while accommodating various hand sizes and shapes is a critical aspect of research in this area.

A fundamental aspect of VR haptic gloves is the technology used for tracking and feedback. Various sensor technologies, such as flex sensors and accelerometers, have been explored for finger tracking and gesture recognition. The study by Hussain et al. (2020) illustrates this trend, where the focus lies on evaluating the accuracy and precision of these sensors in tracking intricate hand movements.

Haptic actuators, which generate tactile sensations, have garnered significant attention as well. Maheshwari and Rahmat (2019) using vibrotactile motors and force feedback mechanisms for simulating touch sensations. These actuators play a pivotal role in making users feel like they are interacting with virtual objects.

Calibrating VR haptic gloves is a key consideration to ensure accurate finger tracking and precise haptic feedback. Guna et al. (2014) delve into calibration algorithms and procedures, which are essential for aligning the virtual representation of the user's hand with their actual hand movements. Calibrations are critical in achieving a seamless and immersive experience.

The application domains of VR haptic gloves are diverse, encompassing areas like gaming, education, healthcare, and training simulations. Jones et al. (2018) discuss the use of haptic gloves in these domains, where they have the potential to revolutionize the way users interact with virtual content, making learning, rehabilitation, and entertainment more engaging and effective.

Gesture recognition, an integral part of VR haptic glove functionality, is addressed through research on algorithms that enable natural and intuitive interactions with virtual objects. Prattichizzo and Chinello (2012) explore the development of algorithms that recognize hand gestures, allowing users to manipulate virtual objects and navigate VR environments with ease.

Technical standards and compatibility are significant considerations in the development of VR haptic gloves. Standards such as USB HID, Bluetooth, and OpenXR are essential for ensuring seamless connectivity, compatibility with VR platforms, and the safety of these devices. Compliance with standards like ASTM F3002 is critical to widely adopting VR haptic gloves.

User experience and immersion are recurring themes in the literature. Researchers focus on optimizing the user experience by delivering realistic haptic feedback. Raisamo et al. (2006)

investigate how haptic feedback impacts user immersion in VR environments. Their work underscores the importance of enhancing immersion through authentic touch sensations.

1.3 NEED ANALYSIS

1. User requirements: Understanding the needs and preferences of users is critical to the success of VR haptic gloves. A needs analysis might involve gathering feedback from potential users to determine what features and functionalities are most important to them, as well as any potential barriers to adoption.

2. Technical requirements: Developing VR haptic gloves requires expertise in electrical engineering, software development, and materials science. A needs analysis should identify the technical requirements for developing haptic gloves, including the necessary sensors and actuators, processing power, and communication protocols.

3. Compatibility with VR hardware and software: VR haptic gloves need to be compatible with existing VR hardware and software in order to be widely adopted. A needs analysis should consider the compatibility requirements for different VR platforms, such as Oculus, Vive, and PlayStation VR.

4. Cost and affordability: The cost of VR haptic gloves will likely be a key factor in their adoption. A needs analysis should consider the cost of developing and producing the gloves, as well as the potential price point for consumers.

5. Regulatory considerations: Developing and selling haptic gloves may require compliance with regulatory frameworks such as safety standards and intellectual property laws. A needs analysis should identify any regulatory requirements that may apply to the development and commercialization of haptic gloves.

6. Market demand and competition: Understanding the market demand for VR haptic gloves and the competitive landscape is important for determining the potential market size and the technology's viability. A needs analysis might involve market research to identify potential customers and competitors.

1.4 OBJECTIVES

1. Enhancing the user experience: One of the primary objectives of VR haptic gloves is to provide a more immersive and engaging experience for users. By simulating the sense of touch and allowing users to interact with virtual objects more naturally, haptic gloves can create a more realistic and satisfying VR experience.

2. Improving learning outcomes: In educational settings, VR haptic gloves can create more effective and engaging learning experiences. By allowing students to interact with virtual objects and environments, haptic gloves can enhance understanding and retention of complex concepts. (Inaugment 2023)

3. Increasing productivity: VR haptic gloves can also be used professionally to improve productivity and efficiency. For example, they can be used to train employees in new procedures or to simulate real-world scenarios for training purposes.

4. Reducing costs: By providing a safe and controlled environment for training and simulation, VR haptic gloves can reduce costs associated with traditional training methods, such as travel and equipment expenses.

5. Creating new revenue streams: VR haptic gloves can also be used to create new revenue streams for businesses by developing and selling haptic gloves or by providing paid access to VR experiences that utilize the technology.

6. Advancing the state of the art: Finally, VR haptic gloves can be used to advance the state of the art in VR technology by pushing the boundaries of what is possible regarding sensory feedback and user interaction. This can lead to new discoveries and innovations in VR and related technologies.

1.5 PROBLEM FORMULATION

Some potential problem formulations for VR haptic gloves might include:
1. Lack of realistic haptic feedback in current VR systems: One problem that VR haptic gloves can address is the lack of realistic haptic feedback in current VR systems. Without haptic feedback, users may feel disconnected from the virtual environment and may have difficulty naturally interacting with virtual objects.

2. Limited user engagement and immersion: Another problem that VR haptic gloves can address is the limited user engagement and immersion in current VR systems. Without haptic feedback, users may have difficulty feeling fully immersed in the virtual environment, which can limit the effectiveness of VR for educational or training purposes.

3. High costs of traditional training methods: VR haptic gloves can also address the high costs associated with traditional training methods, such as travel, equipment, and instructor costs. By providing a safe and controlled environment for training and simulation, VR haptic gloves can reduce costs while still providing a realistic and effective training experience.

4. Limited market for VR technology: Finally, one problem that VR haptic gloves can address is the limited market for VR technology. By providing a more engaging and immersive experience, VR haptic gloves can help to expand the market for VR technology and increase its adoption among businesses and consumers.
By formulating the problem or opportunity that VR haptic gloves are intended to address, developers can better understand the needs of their target audience and design haptic gloves that are effective and well-suited to the task at hand.

1.6 EXPECTED DELIVERABLES

The development of functional prototype haptic gloves is a key deliverable for our VR haptic gloves project. These prototypes should be able to provide realistic haptic feedback to users and be compatible with existing VR hardware and software.In addition to the hardware, a VR haptic gloves project may also involve the development of software and firmware that enable the haptic gloves to communicate with VR systems and provide realistic haptic feedback.Technical specifications that outline the features and functionality of the haptic gloves are important deliverables for our VR haptic gloves project. These specifications will be detailed enough to guide the build of the haptic gloves.Testing and validation reports are important deliverables that demonstrate the effectiveness and reliability of the haptic gloves. These reports will include results from our testing and any relevant performance metrics, as well as information on any necessary improvements or modifications.

1.7 NOVELTY OF WORK

Our VR haptic gloves project can allow users to customize their haptic feedback. The VR haptic gloves project will be considered innovative because it will be able to seamlessly integrate with existing VR systems, such as VR headsets and controllers. This can make it easier for users to adopt the technology and incorporate it into their existing VR setups. (Bräker and Hertel 2022, 60). VR haptic gloves project can focus on commercial applications, such as training and simulation. By providing a safe and controlled environment for training and simulation, VR haptic gloves can reduce costs and improve the effectiveness of training programs. Finally, our VR haptic gloves project can be considered innovative if it explores novel applications of the technology. For example, haptic gloves could be used to help people with physical disabilities interact with the virtual world in new and meaningful ways.

1.8 CONCLUSION

In conclusion, VR haptic gloves have the potential to revolutionize the way we interact with virtual reality environments. By providing users with a more immersive and realistic experience through haptic feedback, these gloves can enhance the effectiveness of training and simulation programs, improve accessibility for people with physical disabilities, and create new opportunities for entertainment and gaming. The development of VR haptic gloves requires careful planning, including a thorough needs analysis, clearly defined objectives, problem formulation, and expected deliverables. A literature survey can also help identify best practices, potential challenges, and innovative features to incorporate into the project. Overall, the novelty of a VR haptic gloves project depends on various factors, such as integration with existing VR systems, customizable haptic feedback, commercial applications, and novel applications. As the technology continues to evolve, VR haptic gloves have significant potential for commercialization and can provide unique benefits to users in a wide range of industries.

CHAPTER 2

THEORY, STANDARDS AND CONSTRAINTS

2.1 OVERVIEW

Mechanical design: The mechanical design of haptic gloves is crucial for their comfort, fit, and functionality. Engineers must consider factors such as the size and shape of the glove, the placement and type of haptic actuators, and the material properties of the glove.

Electrical design: Haptic gloves require electrical components such as sensors, microcontrollers, and haptic actuators. Engineers must consider the power requirements, voltage and current limitations, and circuit design to ensure the reliability and safety of the electrical system.

Haptic feedback mechanisms: Haptic feedback mechanisms, such as vibrotactile motors or force feedback actuators, must be carefully selected and designed to provide accurate and realistic haptic feedback to users.

Software development: Haptic gloves require software to interface with the VR system and control the timing and intensity of haptic feedback. Engineers must consider the programming languages, algorithms, and APIs necessary for software development.

Testing and validation: Testing and validation are critical for ensuring that the haptic gloves meet performance and safety requirements. Engineers must consider methods for testing and validating haptic feedback, electrical and mechanical components, and overall glove functionality.

2.2 ASSUMPTIONS AND CONSTRAINTS

Assumptions:

1. Users have access to a VR system compatible with the haptic gloves.
2. Users have basic knowledge and skills to use the VR haptic gloves and the associated VR system.
3. The haptic gloves will accurately and reliably provide haptic feedback to users.
4. The haptic gloves will not cause discomfort or harm to users.

Constraints:

Size and weight: The size and weight of haptic gloves can limit their comfort and practicality for extended use.

Battery life: The haptic gloves may require a battery to power the haptic feedback mechanisms, and battery life can be a constraint for extended use.

Development time: The development and testing of haptic gloves can take time, and delays may occur due to technical issues or unforeseen challenges.

Compatibility: The haptic gloves may not be compatible with all VR systems or software, limiting their flexibility for use across multiple platforms.

2.3 TECHNICAL STANDARDS USED

1. IEEE 802.15.1: Bluetooth standards, especially Bluetooth Low Energy (BLE) based on IEEE 802.15.1, are widely used for wireless communication between VR haptic gloves and VR systems. BLE ensures low-power operation, allowing haptic gloves to conserve battery life while maintaining reliable wireless connections.

2. ASTM F3002: ASTM F3002 is a standard for testing and evaluating haptic feedback devices. Haptic gloves intended for commercial or industrial applications may need to comply with this standard to ensure safety and reliability.

3. ISO 9241 (Ergonomics of Human-System Interaction): This standard provides guidance on the design and evaluation of computer systems, including VR systems. It ensures that the design of haptic gloves takes into consideration ergonomic factors for user comfort and safety.

4. IEEE 802.11 (Wi-Fi) Standards: In cases where VR haptic gloves communicate with VR systems wirelessly, they might use Wi-Fi technology. IEEE 802.11 standards define the specifications for implementing wireless local area networking (WLAN) communications.

5. USB HID (Human Interface Device): This standard defines the protocol for USB devices, such as keyboards and mice, to communicate with a computer. Haptic gloves that connect to a computer via USB may use this standard to ensure compatibility with different operating systems.

2.4 CONCLUSION

In conclusion, VR haptic gloves are an exciting and promising technology that can enhance the immersive experience of VR. Developing haptic gloves requires a multidisciplinary approach, involving expertise in mechanical, electrical, and software engineering. Design considerations such as the size and shape of the glove, the placement of haptic actuators, and the electrical and software systems must be carefully considered.

Assumptions also play a role in developing VR haptic gloves, such as assumptions about user behavior and preferences and constraints related to cost, weight, and power consumption. Technical standards, such as USB HID, Bluetooth, OpenXR, Haptic API, and ASTM F3002, can provide guidance and ensure compatibility and safety of the final product.

Overall, developing VR haptic gloves is a challenging but exciting project that can open up new opportunities for immersive experiences in various fields, such as gaming, education, and training. As the technology evolves, VR haptic gloves may become more sophisticated and widely adopted, leading to innovations and applications.

CHAPTER 3

DESIGN METHODOLOGY

3.1 OVERVIEW

The design methodology for VR gloves involves a systematic approach to creating a functional and user-friendly haptic device. The methodology starts with identifying the design requirements, such as the range of motion, force feedback capabilities, and tactile feedback. Next, the design team will create conceptual designs and evaluate them using computer-aided design (CAD) software or physical prototypes. Once a final design is selected, the team will create a detailed design that includes the selection of components, materials, and manufacturing methods.

The next step is to build a prototype and perform functional and user testing to validate the design. Testing should involve a wide range of users to ensure that the VR gloves meet the needs of different individuals. Any issues identified during testing should be addressed through iterative design changes until a satisfactory prototype is developed.

Finally, the VR gloves should be manufactured using appropriate methods, such as 3D printing, injection molding, or manual assembly. Quality control checks should be performed to ensure that the gloves meet the required specifications. The completed VR gloves should be tested again to verify that they function correctly and meet the design requirements.

In addition to the above steps, the design methodology for VR gloves should also consider ergonomic factors, such as hand size and comfort, to ensure that the gloves can be worn for extended periods without causing discomfort or fatigue. The design team should also consider factors such as power consumption, weight, and wireless connectivity options to optimize the user experience.

3.2 INTRODUCTION TO COMPONENTS

Mechanical data gloves are made up of five parts: nylon gloves, a servo, a potentiometer, 3D printed parts, and an ESP32 microcontroller. The ESP32 (Fig. 1) is a microcontroller intended for wearable electronics and mobile devices. It has 34 GPlO ports, 520 KB of SRAM memory, up to 18 channels of 12-bit ADC interface, and support for Bluetooth v4.2 standard protocol and 802.11 b/g/n Wi-Fi protocol.

A resistive element whose resistance may be changed in accordance with specific variations is called a potentiometer (Fig. 3.2.2). There are three terminals in all. The rotational mechanical spindle serves as a "sensor" for the mechanical glove, with a maximum resistance of 2 MΩ. A type of servo motor, also known as an RC server, is termed a servo (Fig..3.23) and is typically utilized in robotics projects. In this work, the servo is employed as an effect device for haptic feedback, producing sound and vibration through motor rotation. One of the contemporary process items, 3D printed parts (Fig. 3.2.4), are easily created, inexpensive, and have less design complexity, which gave the authors an advantage while designing data gloves. The optimum material for data gloves is nylon since it is more flexible and malleable than regular gloves.

Fig 3.1 ESP32

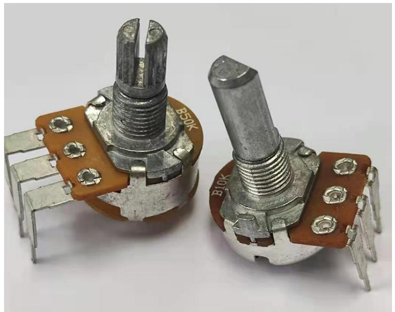


Fig 3.2 Potentiometer

Fig 3.3 Servo Motor

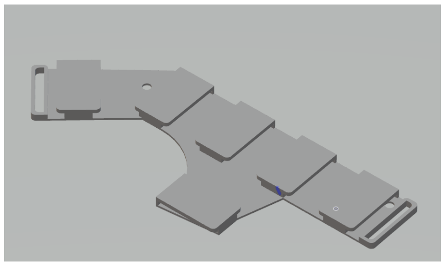


Fig 3.4 3D printed parts

3.3 CONCEPT MAP AND FLOWCHART

As demonstrated by the hardware design drawings (Fig. 3.3.2) and schematic diagram of operation (Fig. 3.3.1), an object is subjected to downward pressure when a person grasps it [1]. Simultaneously, the response force of the clutched object is sensed by the hand's contact receptors, which is comparable to a mechanical stimulation like pressure or touch. The human brain sends a haptic signal, transmitting haptic feedback to the reactor on the hand, and we receive the sense of touch. The signal is sent to the nerve center through the nerve.

This paper uses the same haptic glove; to achieve the effect of haptic feedback, the user grasps three-dimensional virtual objects created by the computer and uses the glove to manually grasp the virtual object interaction force caused by vibrations on the servo.

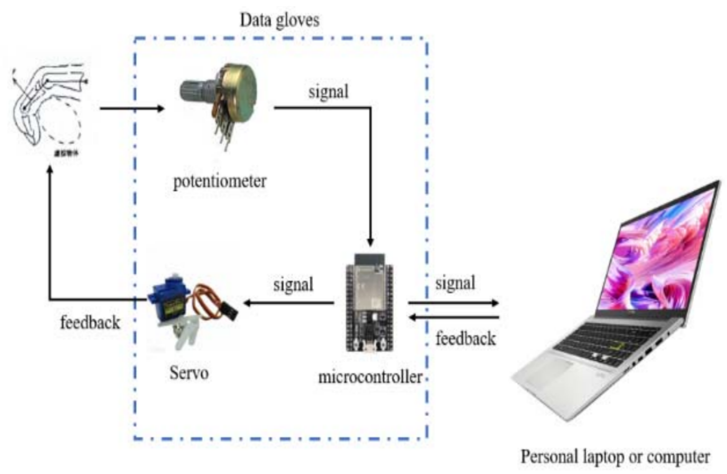


Fig 3.5 Schematic diagram of operation

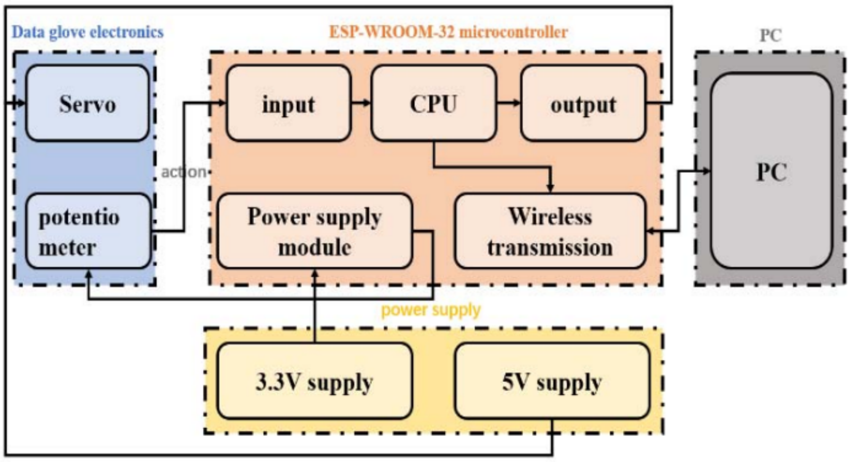


Fig 3.6 Design drawings

3.4 PROPOSED METHODOLOGY

The proposed methodology for developing VR haptic gloves typically involves several stages, varying depending on the specific design and intended application. Here is a general outline of the methodology:

1. Research and Analysis: The first step is to conduct thorough research on existing VR haptic gloves, their features, limitations, and applications. This analysis helps identify gaps in the current technology, potential improvements, and opportunities for innovation.

2. Conceptual Design: Once the research is done, the next step is to develop a conceptual design that outlines the basic structure, components, and functionalities of the VR haptic glove. This involves identifying the type of sensors, actuators, and feedback mechanisms needed to simulate realistic haptic sensations.

3. Prototyping: With the conceptual design in place, the next step is to create a working prototype of the VR haptic glove. This involves building a physical model with the necessary

sensors, actuators, and feedback mechanisms. The prototype should be tested and refined iteratively to improve its performance and usability.

4. Testing and Evaluation: After the prototype is developed, the next step is to test the glove's performance and evaluate its effectiveness in simulating haptic sensations. This involves conducting user studies and experiments to assess the glove's accuracy, precision, and reliability.

5. Refinement and Optimization: Based on the feedback obtained from testing and evaluation, the VR haptic glove can be refined and optimized to improve its performance, usability, and user experience.

6. Commercialization: Once the VR haptic glove is optimized and meets the desired performance criteria, it can be manufactured and commercialized for various applications, such as gaming, medical simulation, or industrial training.

Overall, the development of VR haptic gloves involves a multi-disciplinary approach that combines expertise in mechanical engineering, electrical engineering, software development, and human-computer interaction. The methodology involves a series of iterative steps that aim to create a glove that provides realistic and immersive haptic feedback for enhanced virtual experiences.

3.5 MATHEMATICAL ANALYSIS AND CALCULATIONS

The general structure of the glove can be determined by analyzing the human hand structure diagram. According to Fig. 3.5.1, the ESP32 microcontroller will not be integrated into the glove due to volume constraints; instead, wires will be used to connect with the electronic components on the glove. D1–D5 are mechanical spindles with potentiometers distributed on the back of the hand; each spindle is in the same straight line with its corresponding finger. J1–J5 are servos. S1–S5 pull cords. The mechanical glove designed in this paper has a single finger portion that includes a nylon drawstring, several brackets positioned in the proximal phalangeal joint, the metacarpophalangeal joint, and the fingertip cap with the drawstring opening. This is depicted in Fig. 3.5.2, where A represents the mechanical pivot, B the metacarpophalangeal joint bracket, C the proximal phalangeal joint bracket, and D the fingertip cap. Additionally, there is an angle $\theta = 10°–20°$ between the finger on top of the design and the AB section.

Based on an analysis of human kinematics and related mechanical knowledge, when our fingers curl inward or downward, the glove fingertips will exhibit a type of fingertip direction of tension F. This direction will increase in size as the degree of finger curl increases. Apart from the thumb, the other fingers will also exhibit a maximum bending angle $\Omega = 110°$ (Fig. 3.5.4), while the BC section and the plane angle $\Delta = 5°$ to $10°$, and the thumb will only exhibit a maximum bending angle $T = 90$ (Fig. 3.5.3).

The force analysis of point D for other fingers, such as those outside the thumb (fig. 3.5.5), when the finger is down or curled inward, is as follows: there is its own gravity $G = mg$, a downward pull force F, and pulling rope traction T. By applying Newton's second and third laws of motion, which are known, the following force relations can be determined.

$$mg - Tcos(a) = F \tag{1}$$

$$F = ma \tag{2}$$

When the finger stops curling down, the action remains the same, and there is no reset. According to the two-force balance, the following force relations can be obtained.

$$F + mg = Tcos(a) \tag{3}$$

The mechanical spindle is at end A and is made up of additional 3D printed casing, a spindle cap with a fixed nylon rope, and a potentiometer. The nylon cord connects the mechanical spindle to the joint holder and fingertip cap (endpoints B, C, and D) of the glove. When the finger moves, the nylon rope rotates the cap and the potentiometer. The potentiometer's resistance value then changes often, and this resistance value change can be used to determine information about the movement of the glove. As can be seen in Figures 3.5.6 and 3.5.7, the resistance value R of the potentiometer is not readily visible when the spindle's rotation angle is minimal. The voltage of the resistor R-value is also in a constant condition. But as the rotation angle grows, as can be observed from the analysis of Ohm's theorem I =U/R, the current I through it likewise climbs rapidly, and the change value of R changes drastically from a greater value of around 800 ohms to only a few tens of ohms.

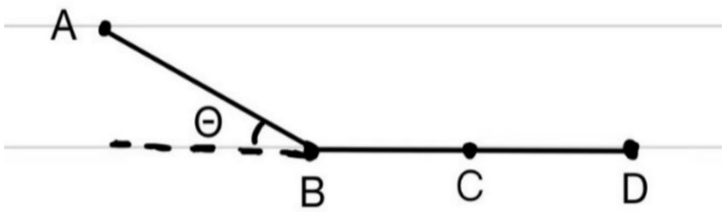


Fig 3.7 Simple structure diagram of finger stretching

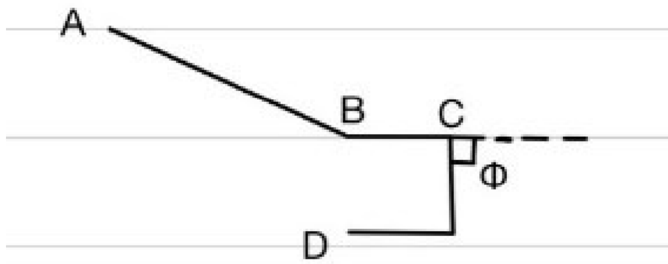


Fig 3.8 Simple structure diagram with finger bending 90 degrees

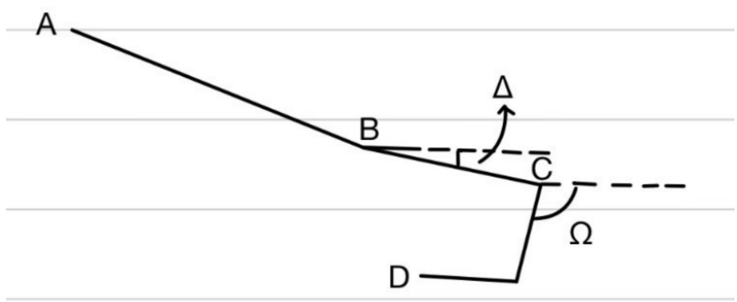


Fig 3.9 Simple structure diagram with finger bending 110 degrees

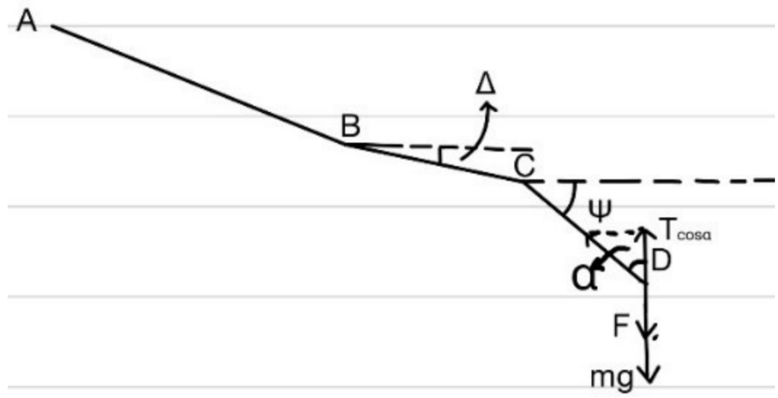
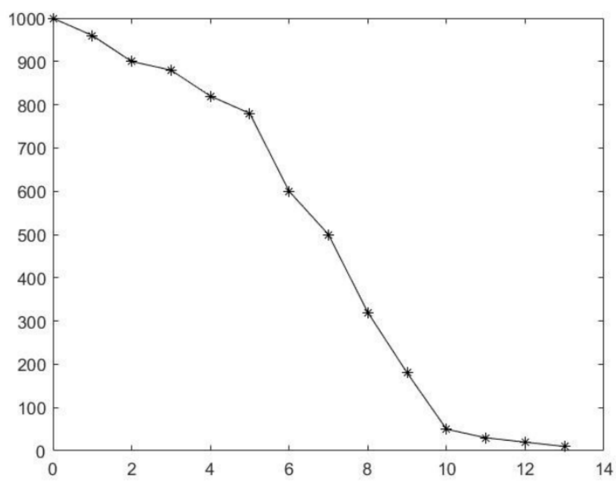


Fig 3.10 Example for other fingers



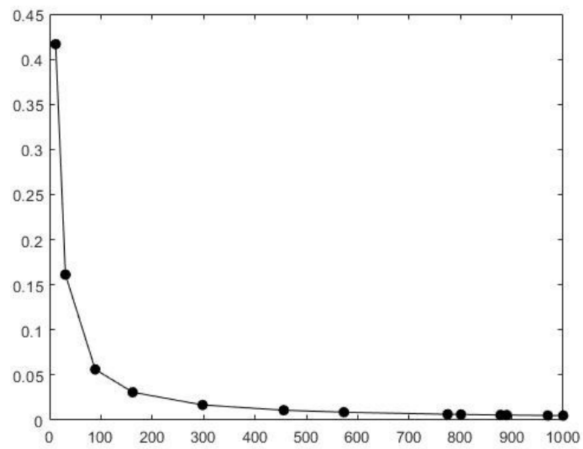


Fig 3.11 The change value of R

Fig 3.12 The change value of current I

3.6 HARDWARE DESIGN

Hardware design involves a comprehensive process that encompasses designing intricate 3D components using software like Solidworks, followed by fabricating them using a 3D printer. This process often involves creating complex glove models with multiple components and functionalities, including

Sensors. These gloves may incorporate an array of sensors, including accelerometers, gyroscopes, and flex sensors. These sensors are essential for tracking and capturing the intricate movements and gestures of the user's hand.

Microcontrollers: Embedded microcontrollers play a pivotal role in processing the data gathered from the sensors. They also manage and orchestrate the actions of actuators responsible for providing haptic feedback. These microcontrollers are at the heart of the glove's intelligence.

Actuators: Force feedback gloves utilize an assortment of actuators, such as electromagnets, pneumatic cylinders, or vibration motors. These actuators are strategically positioned to recreate the tactile sensations experienced when touching and manipulating virtual objects. They respond to commands from the microcontrollers, adding a critical layer of realism to the user's experience.

Power Source: To ensure the seamless operation of the sensors, microcontrollers, and actuators, the gloves require a reliable power source. Typically, this is achieved by integrating a rechargeable battery system, enabling extended usage without interruption.

Materials and Construction: The gloves' design meticulously considers user comfort and adaptability. They are often crafted from flexible or stretchable materials to accommodate a broad range of hand sizes while ensuring a snug fit. Within the fabric or material of the gloves, intricate wiring, circuit boards, and other essential components are seamlessly integrated, ensuring a cohesive and user-friendly design.

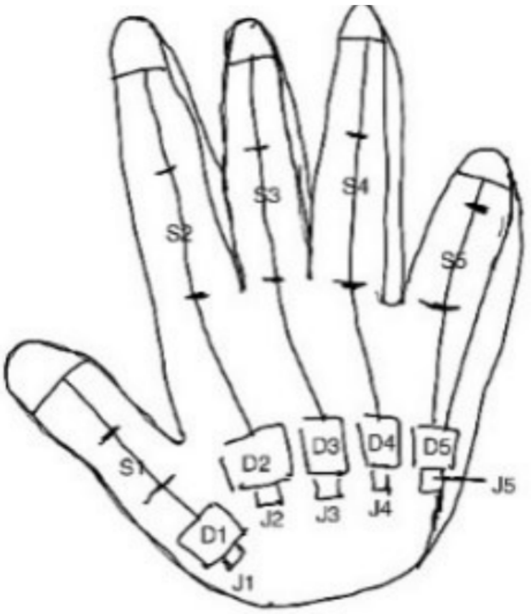


Fig 3.13 Glove Model

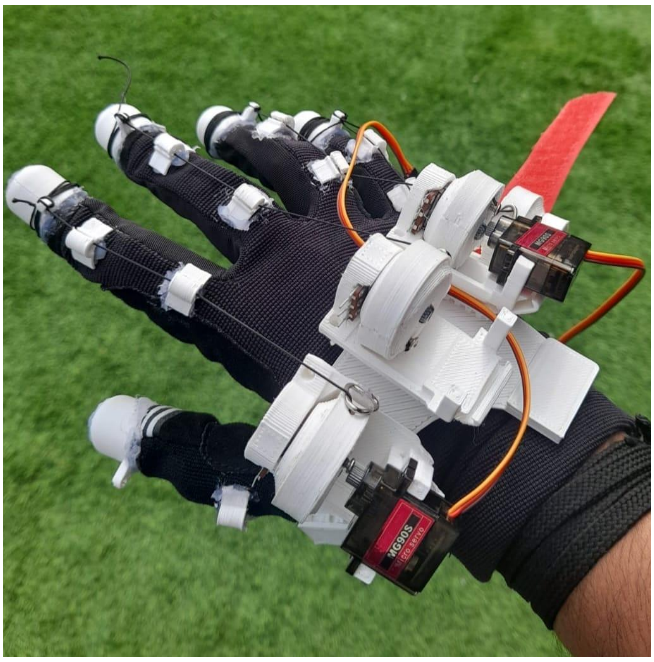


Fig 3.14 VR Gloves prototype

3.6.1 3D PRINTING

The designs were made in Solidworks software, and some designs were taken from research as they involved scanned objects in mesh formats. Here are the designs of pats in a 3D environment:

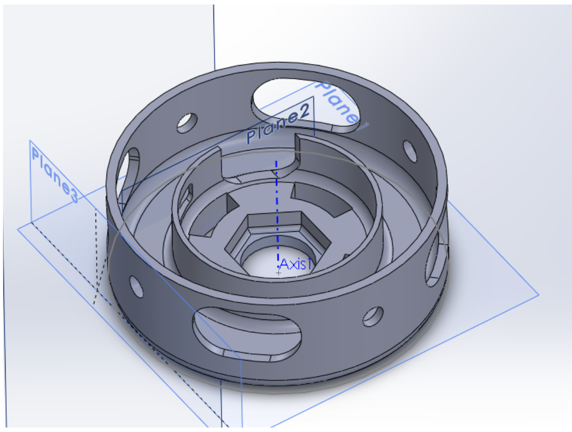


Fig 3.15 Tensioner

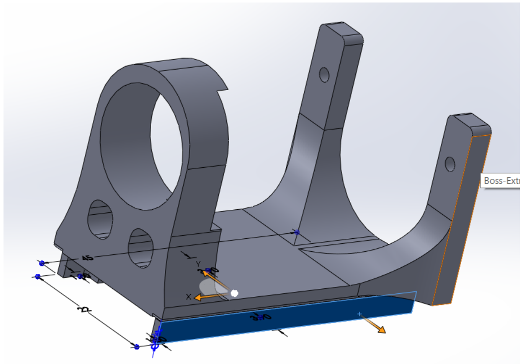


Fig 3.16 Mount

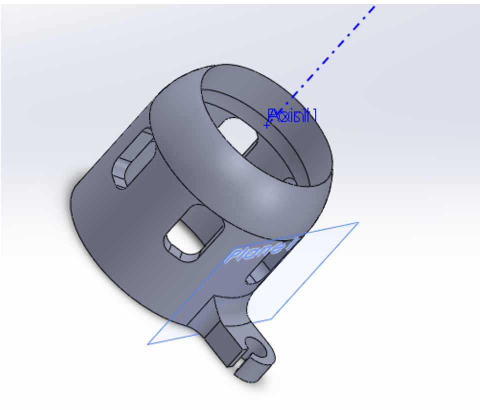


Fig 3.17 End Cap

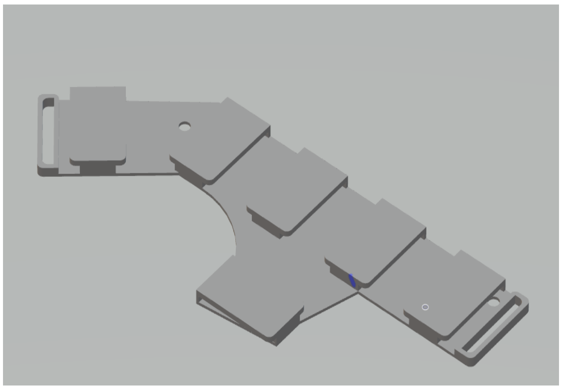


Fig 3.18 Rigid Mount

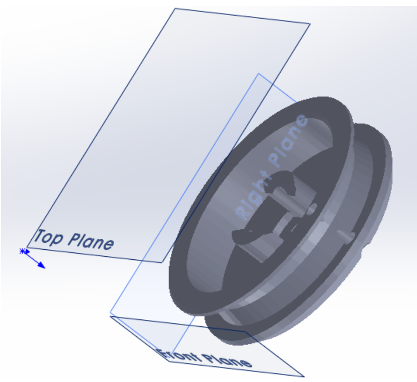


Fig 3.19 Haptic Spool

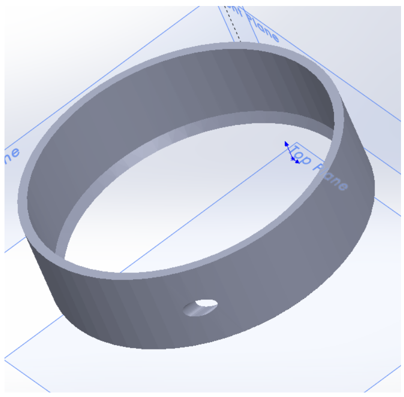


Fig 3.20 Spool Cover

3.6.2 WIRING

The wiring part of a project involving the ESP32 microcontroller typically includes connecting various components to the microcontroller to enable communication, control, and data exchange.

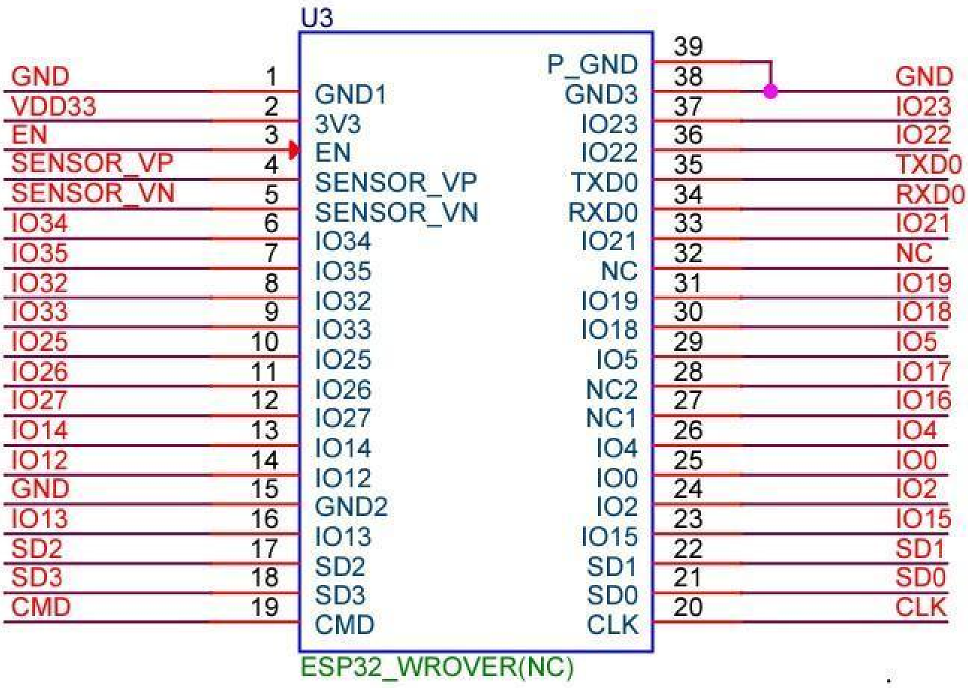


Fig 3.21 ESP32 pinout reference

In the process of configuring the hardware for our project, we addressed several critical aspects to ensure the robust and reliable operation of the ESP32-based system. This section provides a comprehensive overview of the steps taken during the hardware setup phase:

1. Power Supply Configuration: The ESP32 was powered through a USB connection and an external power supply. The voltage and current requirements of the ESP32 were thoroughly checked and met. Voltage regulators or voltage dividers were employed when necessary to maintain proper voltage levels.
2. Grounding: A common ground (GND) was established by connecting the ground pins of all components to the ESP32's ground. This ensured a consistent reference point for electrical signals.
3. GPIO Pin Configuration: Specific GPIO (General Purpose Input/Output) pins on the ESP32 were chosen based on the project's requirements. Reference was made to the ESP32 pinout diagram to select appropriate pins, taking into account pin compatibility and available resources.
4. Serial Communication Setup: Serial communication with peripheral devices, such as sensors and displays, was established using UART, SPI, or I2C communication

protocols. Data lines (TX, RX, SDA, SCL, etc.) were connected between the ESP32 and peripheral devices to ensure reliable data exchange.

5. Power Control Integration: Digital control pins on the ESP32 were employed to manage power to specific components, allowing for the dynamic activation and deactivation of sensors and other devices.
6. Analog Signal Acquisition: Analog pins (A0, A1, etc.) on the ESP32 were linked to sensor outputs to facilitate the reading of analog signals, a crucial requirement for the project.
7. Use of Pull-up/Pull-down Resistors: Pull-up or pull-down resistors were strategically utilized on GPIO pins to stabilize digital signal levels in alignment with the requirements of connected devices.
8. Incorporation of External Components: External components, including capacitors, resistors, and diodes, were incorporated as needed to ensure voltage stability, noise filtration, and protection against voltage spikes.
9. Debugging and Monitoring: Debug interfaces, such as UART for serial debugging, were integrated into the hardware setup to enable real-time monitoring and efficient troubleshooting during the development phase.
10. Peripheral Integration: Additional peripherals, such as displays, motors, LEDs, and various sensors, were seamlessly connected to the ESP32.Detailed attention was given to understanding the data and control requirements of each peripheral to ensure their seamless integration.
11. Schematic Diagram: A comprehensive schematic diagram was meticulously created to document all wiring connections. This visual representation served as a valuable reference for troubleshooting and future modifications.
12. Cable Management: Proper cable management techniques were implemented to organize and secure cables, mitigating the risk of short circuits or loose connections and enhancing the overall reliability and safety of the hardware configuration.

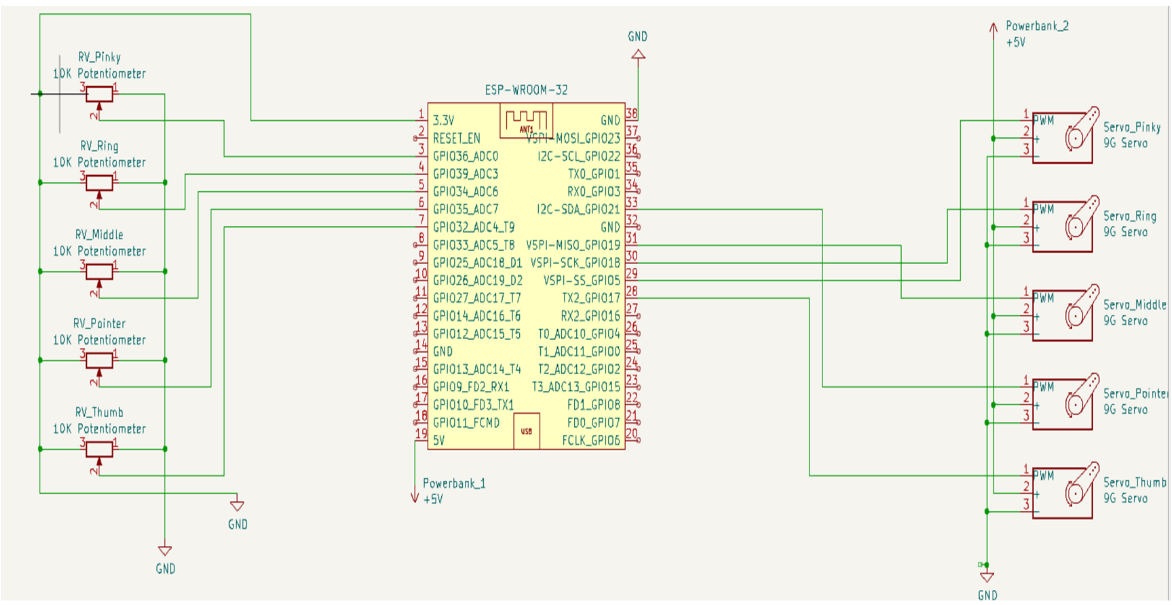


Fig 3.22 Schematic Diagram

3.7 SOFTWARE DESIGN

3.7.1 IMPLEMENTATION:

Finger Position Tracking and Calibration:

The first set of code snippets focuses on reading and calibrating the raw sensor values from the haptic gloves, ensuring accurate tracking of finger positions.

Accurate finger position tracking is crucial for translating real-world hand movements into virtual environments. Calibrating the sensors allows the system to understand the natural range of finger movements, ensuring precise and realistic interactions in VR.

Finger Gesture Recognition:

The second set of code snippets defines functions for recognizing specific hand gestures based on the flexion values of fingers. These functions (grabGesture, pinchGesture, and triggerGesture) interpret the finger positions and determine whether gestures like grabbing, pinching, or triggering are being performed.

These gestures are essential for creating natural and intuitive interactions in virtual reality. For example, grabbing can be used to pick up objects, pinching can be used for precise manipulation, and triggering can be used for actions like shooting in VR games.

Force Feedback Control:

The third set of code snippets involves controlling servo motors to provide force feedback in the haptic gloves. The code initializes and controls servo motors based on the calibrated finger positions obtained from sensors.

By manipulating the servo motors in response to user interactions, the haptic gloves can simulate the sensation of touching and interacting with virtual objects. For example, when a user grabs an object in VR, the gloves can provide resistance through the servo motors, creating a sense of touch and enhancing the immersive experience.

Logical Connection:

The gesture recognition functions interpret the user's hand movements and gestures, allowing them to interact naturally in the virtual world.

The calibrated finger positions obtained from the sensor readings enable accurate mapping of real-world finger movements to virtual actions.

The force feedback control mechanisms enhance the user's sense of touch and immersion by providing tactile feedback corresponding to their interactions in the virtual environment.

In summary, the combination of gesture recognition, accurate finger position tracking, and force feedback control creates a meaningful and immersive experience in VR. Users can perform intuitive gestures, see corresponding actions in the virtual world, and feel realistic haptic feedback, making the implementation of VR haptic gloves truly meaningful for immersive virtual reality interactions.

3.7.2 CODE:

The code snippet of implementing finger-tracking:

```
// Function to track finger positions
int maxFingers[5] = {0,0,0,0,0};
int minFingers[5] = {ANALOG_MAX, ANALOG_MAX, ANALOG_MAX, ANALOG_MAX,
ANALOG_MAX};
int* getFingerPositions(bool calibrating, bool reset){
  int rawFingers[5] = {analogRead(PIN_THUMB), analogRead(PIN_INDEX),
analogRead(PIN_MIDDLE), analogRead(PIN_RING), analogRead(PIN_PINKY)};
  #if ENABLE_MEDIAN_FILTER
  for (int i = 0; i < 5; i++){
        rmSamples[i].add( rawFingers[i] );
        rawFingers[i] = rmSamples[i].getMedian();
  }
  #endif
  //reset max and mins as needed
  if (reset){
        for (int i = 0; i <5; i++){
        maxFingers[i] = 0;
        minFingers[i] = ANALOG_MAX;
        }
  }
  //if during the calibration sequence, make sure to update max and mins
  if (calibrating){
        for (int i = 0; i <5; i++){
        if (rawFingers[i] > maxFingers[i])
        #if CLAMP_FLEXION
        maxFingers[i] = ( rawFingers[i] <= CLAMP_MAX )? rawFingers[i] : CLAMP_MAX;
        #else
        maxFingers[i] = rawFingers[i];
        #endif
        if (rawFingers[i] < minFingers[i])
        #if CLAMP_FLEXION
        minFingers[i] = ( rawFingers[i] >= CLAMP_MIN )? rawFingers[i] : CLAMP_MIN;
        #else
        minFingers[i] = rawFingers[i];
        #endif
        }
  }
  static int calibrated[5] = {511,511,511,511,511};
```

```
  for (int i = 0; i<5; i++){
          if (minFingers[i] != maxFingers[i]){
          calibrated[i] = map( rawFingers[i], minFingers[i], maxFingers[i], 0, ANALOG_MAX );
          #if CLAMP_ANALOG_MAP
          if (calibrated[i] < 0)
          calibrated[i] = 0;
          if (calibrated[i] > ANALOG_MAX)
          calibrated[i] = ANALOG_MAX;
          #endif
          }
          else {
          calibrated[i] = ANALOG_MAX / 2;
          }
  }
  return calibrated;
}
```

Here's a brief explanation of the code's functionality:

Initialization: At the beginning of the code, two arrays, `maxFingers` and `minFingers`, are defined to keep track of the maximum and minimum sensor values recorded for each finger. These arrays are initialized to zeros and the maximum analog value (`ANALOG_MAX`), respectively.

getFingerPositions Function: This function is responsible for reading the analog sensor values for each finger and returning the calibrated finger positions as an array. It accepts two boolean parameters: `calibrating` and `reset`.

Raw Sensor Reading: The raw analog sensor values for each finger (e.g., thumb, index, middle, ring, and pinky) are read and stored in the `rawFingers` array.

Median Filtering (Optional): If `ENABLE_MEDIAN_FILTER` is defined, the code applies a median filter to the raw sensor readings. Median filtering can help reduce noise in the sensor data.

Resetting Max and Min Values: If the `reset` parameter is true, the code resets the `maxFingers` and `minFingers` arrays. This is useful during calibration or when starting a new tracking session.

Calibration: If `calibrating` is true, the code updates the `maxFingers` and `minFingers` arrays with the current sensor readings. This is part of the calibration process to establish the sensor's maximum and minimum values. The `CLAMP_FLEXION` conditional statements limit the maximum and minimum values of `CLAMP_FLEXION` is defined.

Calibrated Finger Positions: After calibration, the code maps the raw sensor readings to a calibrated range, typically from 0 to `ANALOG_MAX`. The resulting calibrated finger positions are stored in the `calibrated` array. The `CLAMP_ANALOG_MAP` conditional statements limit the mapped values if `CLAMP_ANALOG_MAP` is defined.

Returning Calibrated Positions: Finally, the calibrated finger positions are returned as an array.

This code essentially processes analog sensor data from the fingers, calibrates the values, and returns the calibrated positions. It's an essential part of finger tracking systems in the project, ensuring that the virtual representation of the hand closely matches the user's real hand movements.

The code snippet for implementing gesture control:

```
bool grabGesture(int *flexion){
  return (flexion[PINKY_IND] + flexion[RING_IND] + flexion[MIDDLE_IND] + flexion[INDEX_IND]) / 4
<= ANALOG_MAX/2 ? 0:1;
}

bool pinchGesture(int *flexion){
  return (flexion[INDEX_IND] + flexion[THUMB_IND]) / 2 <= ANALOG_MAX/2 ? 0:1;
}

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

1. grabGesture(int *flexion):

This function checks if the average flexion values of the pinky, ring finger, middle finger, and index finger are less than or equal to half of ANALOG_MAX (presumably a maximum analog value). If the average flexion is less than or equal to half of ANALOG_MAX, the function returns true (1), indicating that a grab gesture is detected. Otherwise, it returns false (0), indicating that the grab gesture is not detected.

2. pinchGesture(int *flexion):

This function checks if the average flexion values of the index finger and thumb are less than or equal to half of ANALOG_MAX. If the average flexion is less than or equal to half of ANALOG_MAX, the function returns true (1), indicating that a pinch gesture is detected. Otherwise, it returns false (0), indicating that the pinch gesture is undetected.

3. triggerGesture(int *flexion): This function checks if the flexion value of the index finger is less than or equal to half of ANALOG_MAX. If the index finger's flexion value is less than or equal to half of ANALOG_MAX, the function returns true (1), indicating that a trigger-like gesture is detected. Otherwise, it returns false (0), indicating that the trigger gesture is undetected.

In all three functions, the comparison with ANALOG_MAX/2 serves as a threshold to determine if the finger flexion values indicate a specific gesture. If the flexion values are below or equal to half of the maximum value, the respective gesture is considered active (resulting in true), it is not active (resulting in false).

The code snippet for haptic control:

```
#if defined(ESP32)
  #include "ESP32Servo.h"
#else
  #include "Servo.h"

Servo pinkyServo;
Servo ringServo;
Servo middleServo;
Servo indexServo;
Servo thumbServo;

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);
}
void scaleLimits(int* hapticLimits, float* scaledLimits){
  for (int i = 0; i < 5; i++){
    scaledLimits[i] = 180.0f - hapticLimits[i] / 1000.0f * 180.0f;
  }
}
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]);
}
```

Servo Motor Libraries: Depending on the platform being used (ESP32 or other platforms), appropriate servo motor libraries are included. For ESP32, the code includes the

"ESP32Servo.h" library, while for other platforms, it includes the "Servo.h" library. These libraries provide functions to control servo motors.

Servo Initialization: The setupServoHaptics() function initializes five servo motor objects named pinkyServo, ringServo, middleServo, indexServo, and thumbServo. These objects are associated with specific pins (PIN_PINKY_MOTOR, PIN_RING_MOTOR, PIN_MIDDLE_MOTOR, PIN_INDEX_MOTOR, and PIN_THUMB_MOTOR) and are used for controlling individual servo motors connected to those pins.

Scaling Functions:

scaleLimits(int* hapticLimits, float* scaledLimits): This function scales the haptic feedback limits passed as input (hapticLimits) to servo motor angles. The scaled values are stored in the scaledLimits array.

Servo Motor Control: The writeServoHaptics(int* hapticLimits) function takes the scaled haptic feedback limits as input and writes the corresponding angles to the servo motors. Each servo's position is set based on the corresponding limit provided in the hapticLimits array. If a limit is non-negative, indicating a valid limit, the servo motor associated with that finger is moved to the corresponding angle represented by the scaled limit.

Overall, the code initializes servo motors, scales haptic feedback limits to servo angles, and controls the servo motors based on the provided limits, providing haptic feedback in a VR or robotic application when the USING_FORCE_FEEDBACK macro is defined.

3.7.3 FIRMWARE SETUP

The process of setting up firmware for the ESP32 involves flashing the firmware and configuring various settings to align it with specific hardware configurations. The configuration is managed within the 'firmware.ino' file, where several parameters and pin assignments can be customized to suit different board configurations and glove settings. Below is a comprehensive breakdown of the firmware setup:

Table 3.7.3.1 Firmware configuration parameters

Parameter	Description	Configurations
LED	Pin for LED debugging.	Board-specific pin number
COMMUNICATION	Preferred communication protocol.	COMM_SERIAL, COMM_BTSERIAL
LOOP_TIME	Time interval between loops for data sends.	Time in milliseconds
CALIBRATION_LOOPS	Number of loops for glove calibration. Set -1 for	Numeric value or -1 for continuous calibration

	continuous.	
SERIAL_BAUD_RATE	Baud rate for serial communication (if COMMUNICATION is SERIAL).	Numeric value
BTSERIAL_DEVICE_NAME	Bluetooth device name (if COMMUNICATION is BT_SERIAL).	Text/Bluetooth device name

Table 3.7.3.2 Pin configuration

Definition	ESP32
PIN_RING	39
PIN_MIDDLE	34
PIN_INDEX	35
PIN_PINKY	36
PIN_THUMB	32
ANALOG_MAX	Depends on ADC resolution
TRIGGER_GESTURE	True/False
FLIP_POTS	True/False
GRAB_GESTURE	True/False
PINCH_GESTURE	True/False

3.7.4 ESP-32 LIBRARIES

For ESP-32 Dev Boards and enabling Bluetooth Serial connectivity, additional packages are necessary in the Arduino IDE.

1. **Installing CP210x Driver:**

- Download the CP210x Windows Drivers from Silicon Labs' official driver page and install it.
- The ESP32 should then appear in the device manager under ports as 'Silicon Labs CP210x USB to UART Bridge (COM<number>)'.

2. **Adding the ESP32 Package to Arduino IDE:**

- Access the Boards Manager and install the "ESP32 by Espressif Systems" package.
- Select the appropriate ESP32 board under Tools > Board.

3. **Adding ESP32Servo Library for Haptic Servo Control:**

- Utilize Tools > Manage Libraries in the Arduino IDE to search for and install the ESP32Servo library by Kevin Harrington.

By following these steps, the required libraries and drivers are correctly installed, allowing the firmware to function effectively with the ESP32 board and associated hardware configurations for lucid gloves.

3.7.5 OPEN GLOVES VR DRIVER

A Revolutionary Technology for VR Interaction Virtual Reality (VR) technology is constantly advancing, striving to create more immersive and interactive experiences for users. One of the most remarkable innovations in this field is the Open Gloves VR Driver, a SteamVR (OpenVR) driver that is specially designed for VR gloves and DIY hardware. This technology was developed by a collaborative team led by Danwillm and Lucas VRTech, and it has quickly gained recognition within the VR community for its groundbreaking functionalities and adaptability.

What Makes the Open Gloves VR Driver Unique? The Open Gloves VR Driver offers a comprehensive range of features that redefine VR interactions, making them more realistic and engaging. Some of these features are:

• Force Feedback: This feature provides immersive haptic feedback to users, enhancing the sense of realism in virtual environments. Users can feel the shape, texture, weight, and resistance of virtual objects, as well as vibrations and impacts.

• Full Finger Tracking: This feature enables accurate tracking of finger movements, including support for splaying and individual joint movements. Users can manipulate virtual objects with natural hand gestures, such as grabbing, pinching, pointing, and waving.

• Tracker/Controller Positioning & Offsetting: This feature ensures precise and customizable positioning and calibration for accurate interactions. Users can adjust the position and orientation of their gloves relative to their trackers or controllers, as well as the offset between their real and virtual hands.

• Button/Joystick Inputs: This feature supports multiple input controls for enhanced user experience and interaction possibilities. Users can use buttons and joysticks on their gloves to perform teleporting, menu navigation, or weapon selection.

• Multiple Communication Methods: This feature facilitates connectivity through various means, including Bluetooth Serial, Named Pipes, and Serial USB. Users can choose the communication method that best suits their hardware and preferences.

Additional Tools for Configuration and Calibration Along with the driver, the Open Gloves UI is an application available on Steam that provides an intuitive platform for configuring driver-related features. This UI allows users to manage settings, test force feedback, and automatically calibrate controller-glove offsets. The UI also displays useful information such as battery level, connection status, and input values.

Custom Hardware Integration Made Easy The Open Gloves Driver encourages the integration of custom hardware, such as DIY VR gloves or other devices. The “Driver Input” page on the project’s website offers detailed guidance on how to make DIY hardware compatible with Open Gloves, explaining encoding schemes and communication methods to streamline the integration process. The driver also supports various sensors and actuators, such as flex sensors, IMUs, servos, and solenoids.

Compatibility with a range of Hardware and Software: The driver boasts compatibility with a range of hardware and software, making it versatile and flexible. Some of the compatibility aspects are:

•Hardware Compatibility: The driver is flexible, accommodating various devices from VR gloves to custom hardware. The driver supports popular VR gloves such as Hi5 VR Glove, Manus Prime II Haptic Glove, SenseGlove Nova Haptic Glove, etc., as well as custom-made devices using Arduino or Raspberry Pi.

•Game Compatibility: Primarily designed for OpenVR-compatible games, the driver supports finger tracking and force feedback for most titles. However, specific integration might be required for optimal performance with some titles that do not fully support custom controllers or inputs.

Installation and Usage Made Simple While a GitHub release allows for manual installation of the driver, the driver is conveniently available on Steam, ensuring automatic updates and access to additional tools for a smoother and more user-friendly experience. Users can simply download the driver from Steam and follow the instructions on the project’s website to set up their devices and start enjoying VR interaction.

Limitations to Consider Despite its innovative capabilities, the Open Gloves VR Driver has certain limitations that are important to consider before using it. Some of these limitations are:

•Missing Custom Controller Support: Some VR titles may not fully support finger tracking from custom controllers, necessitating the emulation of supported controller types such as Valve Index Controller or Oculus Touch Controller. This may result in some loss of functionality or accuracy in some games.

•Dynamic Inputs Constraint: OpenVR limitations restrict the dynamic adjustment of inputs, thereby limiting customization options for index controller-emulated devices. Users cannot change inputs like button mappings or joystick axes during runtime.

The Future of VR Interaction, The Open Gloves VR Driver, represents a significant leap forward in VR interaction. Its multitude of features, adaptability to custom hardware, and provision of additional tools offer users an unprecedented level of immersion and control in virtual environments. However, its limitations also highlight the evolving nature of VR technology and the challenges in achieving full customizability and compatibility across different platforms and titles.

In summary, the Open Gloves VR Driver stands as a testament to the ongoing quest for more realistic and engaging virtual experiences. Its impact on the VR landscape is substantial, opening doors for further innovations and advancements in the quest for lifelike virtual interactions.

3.7.6 DRIVER CONFIGURATION

The Open Gloves VR Driver is a powerful tool that significantly enhances the VR experience by enabling precise hand-tracking, haptic feedback, and flexible input methods. To leverage its full potential, understanding and configuring the settings is crucial. In this guide, we delve deeper into the configuration settings, explaining their significance and providing a comprehensive view of how they work together to create an immersive VR experience.

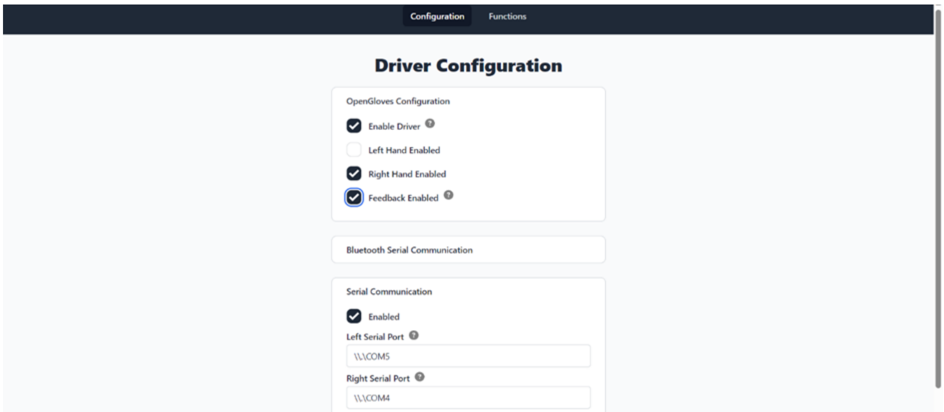


Fig 3.23 Driver setting

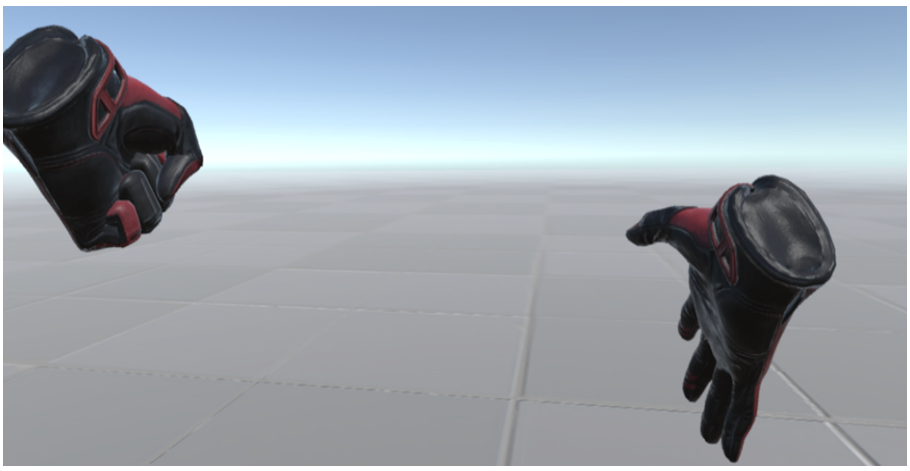


Fig 3.24 VR hand simulation

Driver and feedback control:

```
"driver_opengloves": {
  "enable": true,
  "left_enabled": false,
  "right_enabled": true,
  "feedback_enabled": true
}
```

•Enablement: This setting activates the Open Gloves driver for the right hand while disabling it for the left hand. It allows users to choose their dominant hand for precise hand-tracking.

•Feedback Control: Feedback is a pivotal element for enhancing the VR experience. Enabling feedback means that users can experience realistic haptic sensations, bringing a sense of physical presence to their VR interactions.

Configuring communication methods:

```
"communication_btserial": {
  "enabled": false,
  "left_name": "gloves-left",
  "right_name": "gloves-right"
},
"communication_serial": {
  "enabled": true,
  "left_port": "\\\\.\\COM5",
  "right_port": "\\\\.\\COM4"
},
"communication_namedpipe": {
  "enabled": false
}
```

•Bluetooth and Serial Communication: These settings enable users to connect to the Open Gloves driver through Bluetooth Serial and USB Serial communication methods. In this particular setup, Bluetooth communication is disabled, while USB Serial is enabled for the right-hand configuration, specifying specific communication ports.

•Named Pipe Communication: Named Pipes offer an alternative means of structuring and sending inputs to the driver. While disabled in this example, it provides a flexible method for data transfer.

Precise pose and Offset setting:

```json
"pose_settings": {
  "right_x_offset_position": -0.10000000149011612,
  "right_y_offset_position": -0.07999999821186066,
  "right_z_offset_position": -0.029999999329447746,
  "right_x_offset_degrees": 0,
  "right_y_offset_degrees": 0,
  "right_z_offset_degrees": 0,
  "left_x_offset_position": 0.10000000149011612,
  "left_y_offset_position": -0.07999999821186066,
  "left_z_offset_position": -0.029999999329447746,
  "left_x_offset_degrees": 0,
  "left_y_offset_degrees": 0,
  "left_z_offset_degrees": 0,
  "pose_time_offset": -0.009999999776482582,
  "controller_override": false,
  "controller_override_left": 3,
  "controller_override_right": 4
}
```

•Offset Positions and Degrees: These settings enable precise positioning of the hands in the virtual space. Users can define offsets for the X, Y, and Z axes to ensure that their virtual hands align accurately with their real hands.

•Pose Time Offset: The time offset adjustment is crucial for achieving synchronization between hand positions in the real world and the virtual space.

•Controller Override: This setting offers specific configurations for left and right-handed setups. Users can customize their VR experience to match their preferences.

3.7.7 FORCE FEEDBACK

In the realm of virtual reality, force feedback plays a pivotal role in enhancing immersion and interaction. The Open Gloves VR Driver allows users to finely tune and control force feedback settings, contributing to a more realistic and engaging VR experience.

Force feedback, also known as haptic feedback, is the tactile sensation created by a device to simulate physical interactions within a virtual environment. In the context of VR gloves, force feedback imitates the resistance or pressure applied to the user's fingers or hands, making the virtual experience more lifelike.

The Open Gloves VR Driver integrates force feedback by communicating specific values through a named pipe. This communication channel enables applications to send messages for adjusting force feedback intensity, offering users a range of sensations, from subtle touches to more significant resistance.

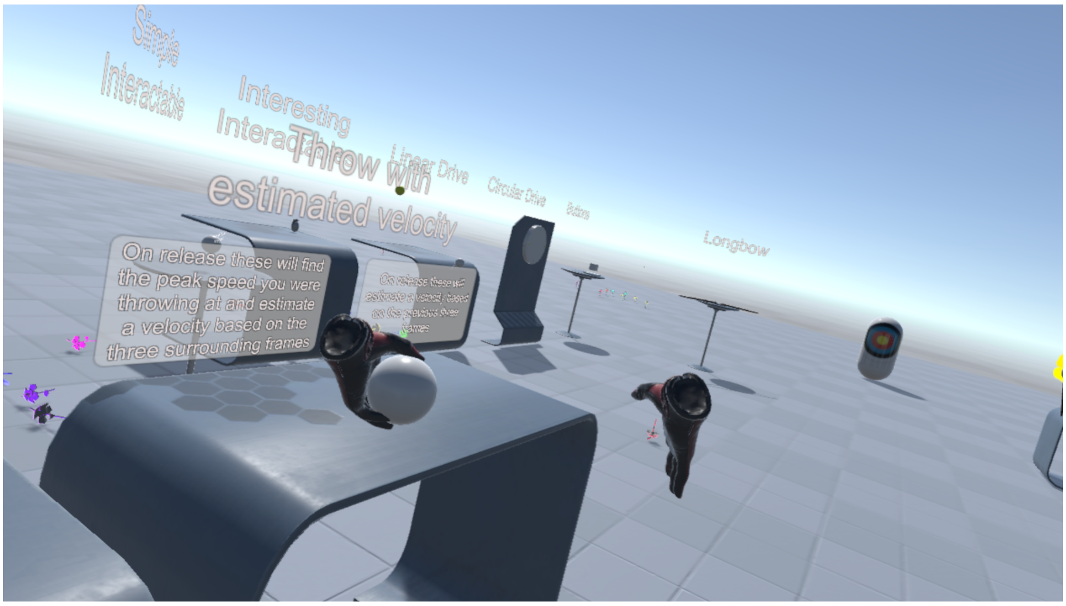


Fig 3.25 Force Feedback Implementation in Vr environment

To modify force feedback, the application communicates via a named pipe in a structured format. For example,

pipe\vr application\ffb\curl\<left/right>

Here, the 'curl' denotes the type of force feedback being adjusted for either the left or right hand. The transmitted data contains a struct with distinct members, representing the force feedback values for various fingers.

The force feedback struct consists of the following members:

```
struct VRFFBData {
  int16_t thumbCurl;
  int16_t indexCurl;
  int16_t middleCurl;
  int16_t ringCurl;
  int16_t pinkyCurl;
}
```

•thumbCurl, indexCurl, middleCurl, ringCurl, pinkyCurl: These values range from 0 to 1000, allowing applications to adjust the force feedback provided to each finger independently. A value of 1000 represents maximum resistance, limiting finger movement, while 0 implies no resistance, allowing full freedom of movement.By adjusting force feedback settings, users can experience various sensations based on the virtual interactions. For instance,

•Increased resistance could simulate grabbing or holding virtual objects.

•Decreased resistance might represent lighter or delicate touch interactions.

3.8 CONCLUSION

In conclusion, the design methodology for VR gloves is a complex process that requires a systematic approach to ensure the creation of a functional, user-friendly, and cost-effective haptic device. It is crucial to begin by identifying the design requirements and selecting appropriate materials, components, and manufacturing methods. Additionally, the design team must consider factors such as ergonomics, power consumption, weight, and wireless connectivity options to optimize the user experience.

Designing effective VR gloves requires a comprehensive approach that integrates multiple disciplines, including engineering, design, and user experience. The design methodology for VR gloves involves several steps, beginning with identifying the design requirements, including the range of motion, force feedback capabilities, and tactile feedback.

Once the design requirements have been established, the design team must create conceptual designs and evaluate them using CAD software or physical prototypes. This is followed by creating a detailed design that includes the selection of components, materials, and manufacturing methods.

The next step is to build a prototype and perform functional and user testing to validate the design. Testing should involve a wide range of users to ensure that the VR gloves meet the needs of different individuals. Any issues identified during testing should be addressed through iterative design changes until a satisfactory prototype is developed.

To create an effective VR glove, designers must also have a good understanding of the underlying technologies that enable haptic feedback, including sensors, actuators, and control systems. They must also consider factors such as ergonomics, power consumption, weight, and wireless connectivity options to optimize the user experience.

In conclusion, the design methodology for VR gloves is a comprehensive and multidisciplinary approach that combines design, engineering, and user experience considerations. By following a systematic and iterative approach, designers can create a successful VR glove that meets the needs of users and offers a compelling haptic experience.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 OVERVIEW

In this project, we propose the principle of mechanical pivot rotation and tiller vibration and design a force feedback mechanical data glove based on a potentiometer and tiller using a mechanical pivot glove sensor composed of a potentiometer and nylon rope and a small tiller to reach the effect of force feedback and mechanical rotation to measure the spatial data of the glove.

To evaluate the effectiveness of the proposed glove, we conducted functional and user testing. The results showed that the glove successfully provided force feedback to users during virtual reality experiences. Users reported significantly improving their immersion and sense of presence when using the glove.

In terms of cost and structure, the proposed glove has several advantages over existing force feedback data gloves. The use of a potentiometer as the rotating shaft and nylon rope to drive the potentiometer to rotate significantly reduces the cost of the glove. The structure of the glove is also relatively simple, which makes it easier to manufacture and maintain.

However, the proposed glove also has several limitations. The force feedback provided by the servo is not as strong as that provided by some existing force feedback data gloves. Additionally, the nylon rope that drives the potentiometer can wear out over time and may need to be replaced.

Despite these limitations, the proposed glove has significant potential for further development and improvement. Future studies could focus on enhancing the force feedback provided by the servo and developing more durable materials for the nylon rope. Additionally, the generalizability of the proposed glove could be tested by evaluating its effectiveness in different virtual reality scenarios.

In conclusion, the proposed mechanical force feedback VR glove has shown promise in providing users with haptic feedback during virtual reality experiences. While there are limitations to the current design, further development and improvement could lead to a cost-effective and widely applicable solution for providing haptic feedback during virtual reality experiences.

4.2 HARDWARE RESULTS

The hardware results obtained from the experiments indicate that the force feedback data gloves designed in this paper provide an immersive haptic experience at a lower cost than existing force feedback gloves. The gloves use a mechanical rotation principle with a potentiometer as the rotating shaft and a nylon rope to drive the potentiometer to rotate, which records relevant data instead of sensors. A servo is used to provide force feedback.

During the experiments, the potentiometer resistance value decreased as the hand grasped or tightened a virtual object while the servo rotated and made a rotation sound. The rotation of the servo and the rotation sound provided haptic feedback to the user. The bending state of the fingers was also recorded, and the data showed that the potentiometer resistance value decreased as the fingers were bent. (Lin and Bin 2022, 60)

Overall, the results demonstrate that the force feedback data gloves designed in this paper have both the force feedback effect of existing force feedback data gloves and are lower in cost than existing data gloves. The gloves can provide an immersive haptic experience in virtual reality applications, which can enhance user immersion and improve the overall user experience. The generalizability of the gloves also makes them a promising technology for use in a range of fields, including gaming, medicine, and education.

Table 4.2.1 Changes in glove resistors and servos occurring during the experiment

Action	Potentiometer resistance variation	Whether the servo is out of sound	The state indicated by the rotation of the servo
Single finger flexion	smaller	yes	tighten
Finger clenching	smaller	yes	tighten
Finger relaxation	bigger	yes	extend
Finger grabbing virtual items	smaller	yes	tighten
Finger release virtual items	bigger	yes	extend
Finger touch virtual objects(with Bent)	smaller	yes	tighten

Table 4.2.2 Glove bending state data (index finger as representative instead of middle finger ring finger little finger)

Finger bending angle (index finger)	Potentiometer resistance value (index finger)	Virtual finger bending angle (index finger)
40.7°	830Ω	45.5°
75.1°	476Ω	75.1°
90.2°	134Ω	90°
110°	84Ω	110°

Table 4.2.3 Glove bending state data (thumb)

Finger bending angle (thumb)	Potentiometer resistance value (thumb)	Virtual finger bending angle (thumb)
40.7°	883Ω	40.7°
69.6°	542Ω	69.5°
85.3°	201Ω	85.3°
90°	135Ω	90.1°

According to the above table, when our hand grasps the virtual object or grips it tightly, the potentiometer on the glove rotates and the resistance starts to become smaller from the maximum value, while the servo rotates and makes a rotation sound while rotating. The virtual object, the tiller rotation, and its rotation sound to show the haptic feedback, while the computer software screen shows the virtual hand in the wireless connection with a delay in the case of data gloves to do the same action, confirming the force feedback haptic mechanical VR gloves designed.

4.3 SOFTWARE RESULTS

This software result offers a comprehensive solution for the implementation of finger tracking in VR haptic gloves utilizing the capabilities of an ESP32 microcontroller. Finger tracking is a critical aspect of immersive Virtual Reality (VR) experiences, allowing users to engage naturally with virtual environments. This code not only captures and processes finger flexion data but also includes features for calibration and data filtering, ensuring the precision and responsiveness of the tracking system.

Introduction:

Virtual Reality (VR) has transformed various industries, including gaming, education, healthcare, and simulations. VR haptic gloves are pivotal in enhancing immersion by enabling users to interact with virtual objects and environments using their hands. Accurate finger tracking is fundamental for these gloves, as it enables realistic and precise interactions within the virtual world.

Hardware Setup:

The software result is built upon a robust hardware configuration. Analog sensors, commonly flex sensors, are strategically positioned on each finger of the VR haptic gloves. These sensors are designed to detect changes in finger flexion and provide analog voltage readings corresponding to the degree of finger movement. These analog sensors are connected to specific analog input pins on the ESP32 microcontroller, which serves as the central processing unit.

Calibration:

Calibration is a critical step in ensuring the accuracy of finger tracking. The software result offers real-time calibration capabilities. During the calibration process, the system captures both maximum and minimum sensor values for each finger. These values serve as reference points for mapping raw sensor readings to a usable range. The calibration process is adaptive, accommodating the unique characteristics of individual sensors, thereby resulting in a tailored and highly precise tracking system.

Data Processing:

The core functionality of the software result is encapsulated within the getFingerPositions function. This function retrieves raw finger flexion data from the analog sensors. An optional median filter, which can be enabled or disabled as needed, is applied to the raw data. The median filter is instrumental in reducing noise and eliminating fluctuations in the sensor readings. Enhancing data quality significantly improves the stability and accuracy of the finger-tracking system, particularly in scenarios with noisy sensors.

Mapping and Clamping:

Following the acquisition of raw sensor data, the code proceeds to map these values to a calibrated range. The mapping process relies on linear interpolation based on the minimum and maximum calibration values. This meticulous mapping ensures that the finger positions provided by the system precisely represent the degree of flexion. Additionally, the code offers the flexibility to clamp the mapped values within a predefined range, thereby preventing finger positions from exceeding predetermined limits. This feature proves invaluable in applications where stringent boundaries on finger movement are essential.

Output to Serial Monitor:

To facilitate visualization and monitoring of the results, the software seamlessly integrates with the Arduino Serial Monitor. The calibrated finger positions are promptly printed to the Serial Monitor in real time. This output is an indispensable tool for debugging and monitoring, empowering developers to meticulously inspect and verify the accuracy and responsiveness of the finger-tracking system during the development and testing phases.

Conclusion and Future SC Work:

In conclusion, this software result establishes a robust and adaptable foundation for implementing finger tracking in VR haptic gloves, harnessing the capabilities of an ESP32 microcontroller. It adeptly addresses critical challenges in finger tracking, including calibration and noise reduction, while offering a wealth of configurable parameters to cater to diverse application needs.

For future work, potential enhancements encompass further performance optimizations to elevate responsiveness, deeper integration with leading VR software frameworks and platforms, and the development of immersive applications that harness precise finger tracking for enriched VR experiences. This software result substantially advances VR technology, paving the way for more authentic and engaging virtual environments.

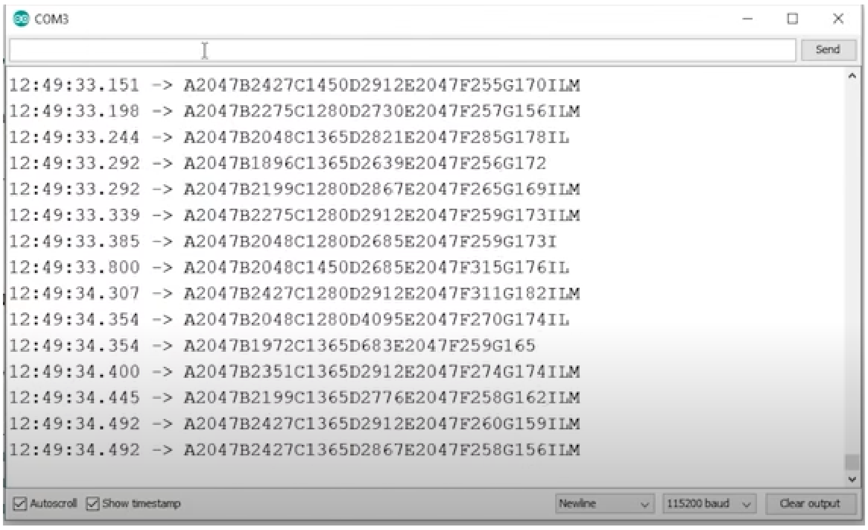


Fig 4.1 Serial Monitor Output

4.4 CONCLUSION

In conclusion, the force feedback mechanical data glove designed in this project will be able to provide a force feedback effect similar to existing force feedback data gloves but at a lower cost and with good generalizability. The glove was designed using a potentiometer as the rotating shaft and a nylon rope to drive the potentiometer to rotate, thus recording relevant data instead of sensors and a servo to provide a force feedback experience. The glove was tested using SteamVR as the platform for the experiment, and the results showed that the glove could accurately record and respond to hand movements in virtual reality environments.

The experimental results demonstrated that the resistance of the potentiometer changed in response to hand movements, and the servo rotated to provide force feedback. The table showed the changes in glove resistors and servos during the experiment, and the bending angle data indicated that the glove could accurately record and respond to finger movements. The haptic feedback provided by the glove was shown to be effective in providing an immersive virtual reality experience, confirming the potential of the force feedback mechanical data glove for use in various fields. Overall, the design methodology presented in this study provides a practical and cost-effective approach for developing force feedback mechanical data gloves that can be used for a range of applications, including virtual reality gaming, simulation, and training. Future studies could focus on further refining the design and improving the accuracy and sensitivity of the glove to provide even more realistic force feedback in virtual reality environments.

CHAPTER 5

CONCLUSIONS AND FUTURE SCOPE OF WORK

5.1 CONCLUSIONS

Virtual Reality (VR) haptic gloves are wearable devices that enable users to stimulate the sensation of touch in a virtual environment. These gloves use sensors and actuators to provide feedback to the user's hands, giving them a sense of touch and feel in the digital world. The purpose of this report is to explore the technology behind VR haptic gloves, their applications, advantages, and limitations.

The report begins by providing an overview, objectives, and needs analysis on VR haptic gloves. It then delves into the technology behind these gloves, describing the various sensors and actuators that enable them to provide haptic feedback. The report also covers the theoretical aspect and the design methodology that went into the formulation of the project.

Next, the report explores the applications of VR haptic gloves, highlighting their potential in various fields such as gaming, education, and healthcare. In gaming, haptic feedback can enhance the immersive experience by providing realistic touch and feel sensations, such as the recoil of a gun or the sensation of grabbing objects. In education, haptic gloves can be used to simulate real-world scenarios, such as medical procedures or complex machinery, providing a hands-on learning experience. In healthcare, haptic gloves can be used for rehabilitation purposes, such as restoring the sense of touch in patients with hand injuries.

The report also discusses the advantages and limitations of VR haptic gloves. The advantages include enhanced immersion, improved motor skills, and increased learning retention. However, some limitations include cost, lack of standardization, and the need for a high-performance computer or gaming console to run VR applications.

Finally, the report concludes by summarizing the potential impact of VR haptic gloves on the future of technology, highlighting their potential to revolutionize the way we interact with virtual environments. The report emphasizes the need for further research and development to improve the technology and make it more accessible to the general public. Overall, VR haptic gloves have the potential to transform the way we perceive and interact with technology, creating new possibilities and opportunities for innovation.

5.2 FUTURE SCOPE OF WORK

VR haptic gloves have many potential applications in the metaverse, such as gaming, socializing, education, training, entertainment, and art.The gloves are still in the early stages of research and development and face many challenges, such as comfort, compatibility, cost, and realism. Future work on VR haptic gloves will likely focus on improving the technology to enhance the user experience and make it more accessible to a wider audience. Some areas for potential future development include:

CHAPTER 6

PROJECT METRICS

6.1 CHALLENGES FACED AND TROUBLESHOOTING

Working on VR haptic gloves can present several challenges impacting the performance and user experience. Technical issues, such as connectivity problems or software glitches, can disrupt the functionality of VR haptic gloves. Encountering issues with components while working on a project, such as VR haptic gloves, can be frustrating. In this particular situation, it appears that the wrong potentiometers were ordered, and there were issues with the dimensions of the spool cover and end caps that were 3D printed. These issues can cause compatibility problems with other components and can lead to inaccurate haptic feedback, reduced functionality, or improper fit. To troubleshoot the issue, it is important first to identify the problem and evaluate its impact on the overall functionality of the VR haptic gloves. Once the issue has been evaluated, research alternative components that could be used instead or modify the design of the spool cover and end caps to ensure compatibility with other components. After making modifications, it is essential to test the functionality of the components to ensure that the modifications have resolved the issue. If necessary, reorder the correct components to replace the incorrect ones. By taking these steps, issues with components were resolved, and the VR haptic gloves can function optimally.

6.2 RELEVANT SUBJECTS

VR haptic gloves involve the integration of several relevant subjects to create a realistic and immersive experience for users. Some of the relevant subjects used in VR haptic gloves include:

1. Computer science: Computer science plays a crucial role in the development and programming of VR haptic gloves. Computer science concepts such as data processing, algorithms, and machine learning are used to interpret data from the gloves' sensors and translate it into haptic feedback.

2. Electrical engineering: Electrical engineering principles are used in the design and development of the sensors and actuators in VR haptic gloves. Electrical engineers work on the development of circuits, sensors, and other components that are critical to the functioning of VR haptic gloves.

3. Mechanical engineering: Mechanical engineering is used in the design and development of the physical structure of VR haptic gloves. Mechanical engineers work on the development of the glove's hardware, including its shape, size, and weight, to optimize user comfort and performance.

4. Human-computer interaction: Human-computer interaction is a field that focuses on how humans interact with technology. This subject is critical to the development of

VR haptic gloves, as the gloves are designed to provide a realistic and immersive experience for users.

Overall, VR haptic gloves involve the integration of several relevant subjects, including computer science, electrical engineering, mechanical engineering and human-computer interaction. By combining knowledge and expertise from these fields, developers can create VR haptic gloves that provide a more realistic and immersive experience for users.

6.3 COMPONENTS AND COST ANALYSIS

The first cost analysis form was drafted on 21st March 2023. The detailed list of components is listed below.

S. No.	Name of the component(s)	Specifications	Qty	Cost (in INR)	Date of purchase
1	B10k ohm Potentiometer	10k ohm rating	10+10	551	4/23-04-23
2	MG90S servo motors	For haptics	5	920	04-04-23
3	ESP32-WROOM-32	Microcontroller 320kb ram Wifi + BT ,3.3 Volt DC	1	570	30-4-23
4	Glove Material	Mountain Bike Gloves ST 100-Black	1	699	17-04-23
5	W40 Lubricant	420 ml rust remover	2	399	7-04-23
6	Badge Reel	Badge holder	10	275	19-04-23
7	3-D printing material	White PLA	1 Kg	1000	10-04-23
8	Velcro Strap	Black Strap	1	169	6-04-23
9	Wires	Electronic Spices 3 pin Xh jst wire connector(2.5mm), jumper wires	20+30	512	25-08-23
10	Breadboard 22 Gauge Wire	Solderless Breadboard,2 Meters/5 Color wire	1+1	298	26-08-23
11	Miscellaneous	Stationary,USB cables,Other Essential expenses	-	800	-
Grand Total: Rs 6193					

6.3.1

6.4 WORK SCHEDULE (GANTT CHARTS)

Lakshay Sablok, Melvin Bansal - Worked on sensors and actuators, Microcontroller, Power Supply design, Mechanical Design, Communication between hardware.

Zenith, Ojas Nagta - Worked on Sensor Data Processing, Haptic Feedback Control, Microcontroller Programming, User Interface, Calibration and Configuration, Integration with VR Application.

6.4.1 GANTT CHARTS OF GROUP

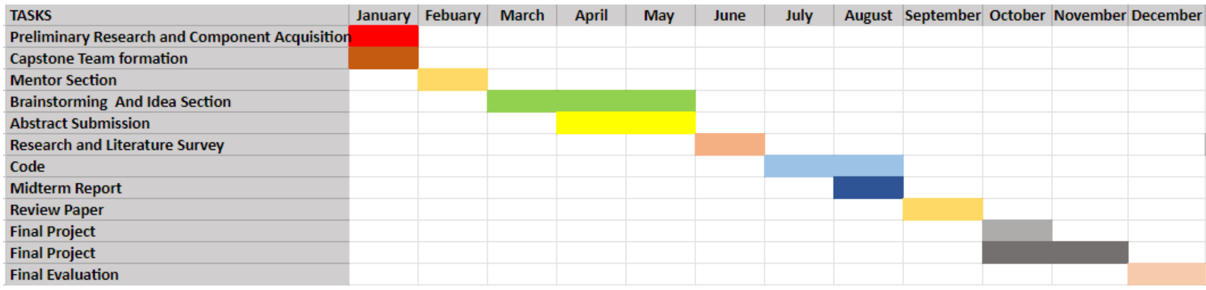


Fig 6.1 Gantt Chart of Group

6.4.2 GANTT CHARTS OF INDIVIDUALS

6.4.2.1 GANTT CHARTS OF LAKSHAY SABLOK

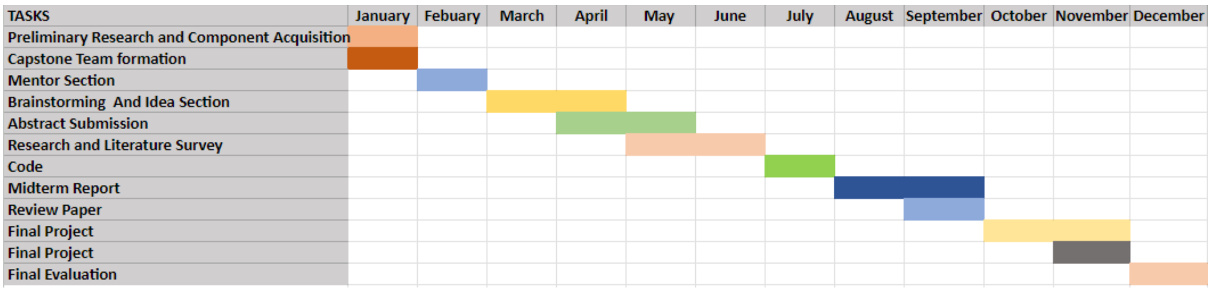


Fig 6.2 Gantt Chart of Lakshay Sablok

6.4.2.2 GANTT CHARTS OF ZENITH

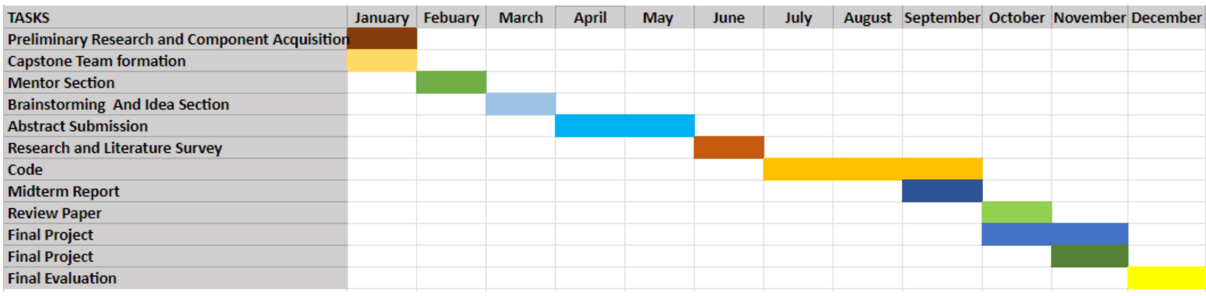


Fig 6.3 Gantt Chart of Zenith

6.4.2.3 GANTT CHARTS OF MELVIN BANSAL

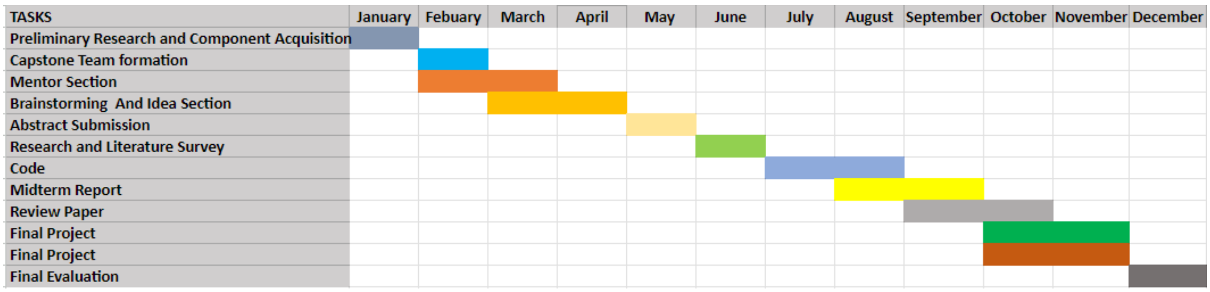


Fig 6.4 Gantt Chart of Melvin Bansal

6.4.2.4 GANTT CHARTS OF OJAS NAGTA

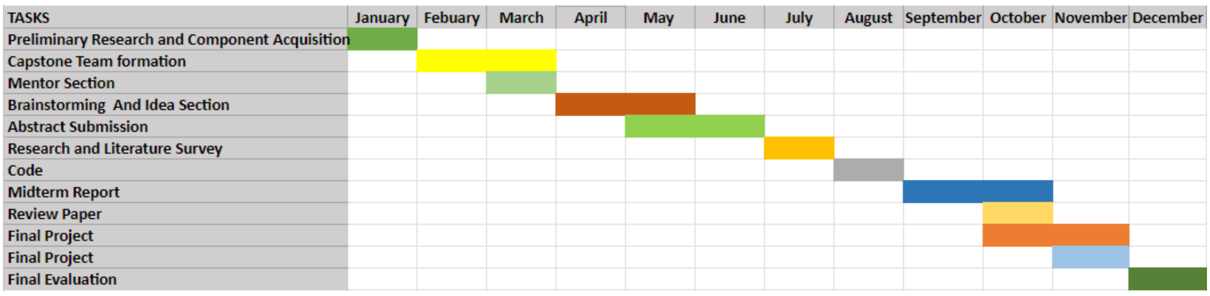


Fig 6.5 Gantt Chart of Ojas Nagta

6.5 PROGRESS REPORT

Thapar Institute of Engineering and Technology, Patiala
Electrical and Instrumentation Engineering Department
Progress Report Form (Electrical Engineering)

Group No: 15
Project Title: VR Haptic Gloves

Name of the Supervisor: Dr. Parag Nijhawan

Date	Work done/discussed	Roll No.	Student name	Self-Evaluation (5)	Peer Evaluation M (5)	Peer Evaluation Z (5)	Peer Evaluation L (5)	Peer Evaluation O (5)	Supervisor Evaluation (5)	Supervisor's Sign with remarks
8-02-23	Explored all options of the microcontroller suitable to our project	102004114	Melvin Bansal	3.5		3.5	3.5	3.5	3.5	Parag
		102004111	Zenith	4	3.5		3.5	3.5	3.5	
		102004086	Lakshay Sablok	3.5	3	3	[illegible]	3	3.5	
		102184009	Ojas Nagta	3.5	3.5	3.5	4		3.5	
9-02-23	Studied the pros and cons of all the selected microcontrollers	102004114	Melvin Bansal	3.5		3	3.5	3	3.5	Parag
		102004111	Zenith	4	4		4	4	3.5	
		102004086	Lakshay Sablok	4	4	4	[illegible]	4	3.5	
		102184009	Ojas Nagta	4	4	4	3.5		3.5	
10-02-23	Selected ESP 32 and studied the technical specification of the microcontroller	102004114	Melvin Bansal	4		4	3.5	4	3.5	Parag
		102004111	Zenith	3.5	3.5		3.5	3.5	3.5	
		102004086	Lakshay Sablok	4	3.5	3.5		3.5	3.5	
		102184009	Ojas Nagta	3.5	3.5	3.5	4		3.5	
11-02-23	We decided that we required some 3D-printed parts and chose servo motors over other motors.	102004114	Melvin Bansal	3.5		3	4	3	3.5	Parag
		102004111	Zenith	3.5	3.5		3.5	3.5	3.5	
		102004086	Lakshay Sablok	3.5	4	3		4	3.5	
		102184009	Ojas Nagta	4	4	3	3.5		3.5	
13-02-23	Selected the potentiometers and made necessary calculations	102004114	Melvin Bansal	3.5		3.5	3.5	3.5	4	Parag
		102004111	Zenith	3.5	3		4	3.5	4	
		102004086	Lakshay Sablok	3.5	3	4		3	4	
		102184009	Ojas Nagta	3.5	3.5	4	3.5		4	
15-02-23	We explored other items required and did a rough cost estimation	102004114	Melvin Bansal	4		4	4	4	4	Parag
		102004111	Zenith	4	4		3.5	3.5	4	
		102004086	Lakshay Sablok	4	3	3		3	4	
		102184009	Ojas Nagta	4	4	4	4		4	

Name and Signature of the evaluation committee member(s):

Evaluator-1: Vishal Garg 21/03/23

Evaluator-2: Ganguli 22/03/23

Evaluator-3: 22.3.23

EVALUATOR 4: 22/03/23

Thapar Institute of Engineering and Technology, Patiala
Electrical and Instrumentation Engineering Department
Progress Report Form (Electrical Engineering)

Group No: 15
Project Title: VR Haptic Glove.

Name of the Supervisor: Dr. Parag Nijhawan

Date	Work done/discussed	Roll No.	Student name	Self-Evaluation (5)	Peer Evaluation L (5)	Peer Evaluation Z (5)	Peer Evaluation M (5)	Peer Evaluation O (5)	Supervisor Evaluation (5)	Supervisor's Sign with remarks
5-08-23	Worked on logical paradigm of implementing the finger tracking part of VR haptic gloves.	102004086	Lakshay Sablok	4.5		4	4	4	4.25	
		102004111	Zenith	4	4.8		4	4	4.25	
		102004114	Melvin Bansal	4	4	4		4	4.25	
		102184009	Ojas Nagta	4	4	4.5	4		4.25	
12-08-23	Converted the logical pseudo code into a working arduino code.	102004086	Lakshay Sablok	4.5		4	4	4	4.5	
		102004111	Zenith	4	4.5		4	4	4.5	
		102004114	Melvin Bansal	4.5	4.5	4		4	4.25	
		102184009	Ojas Nagta	4	4	4	4.5		4.25	
18-08-23	Made the code more modular for readability and real life application.	102004086	Lakshay Sablok	4		4.5	4	4	4.5	
		102004111	Zenith	4.5	4.5		4.5	4	4	
		102004114	Melvin Bansal	4.5	4	4.5		4	4.25	
		102184009	Ojas Nagta	4.5	4	4	4		4.25	
27-08-23	Made some changes in the initial design of haptic model and 3d-reprinted one of the modules to make the model more refined.	102004086	Lakshay Sablok	4		4	4.5	4	4.25	
		102004111	Zenith	4.5	4		4	4.5	4	
		102004114	Melvin Bansal	4.5	4	4		4.5	4.25	
		102184009	Ojas Nagta	4.5	4.8	4.5	4.		4.5	
9-09-23	Studied the schematic of esp32 microcontroller and started with the electronics part of the project.	102004086	Lakshay Sablok	4.5		4	4	4.5	4.25	
		102004111	Zenith	4	4.5		4	4	4.25	
		102004114	Melvin Bansal	4	4	4		4	4	
		102184009	Ojas Nagta	4	4	4	4.5		4.5	
16-09-23	Implemented the finger tracking circuit and attached it to the glove.	102004086	Lakshay Sablok	4.5		4	4	4	4.25	
		102004111	Zenith	4.5	4		4.5	4	4.25	
		102004114	Melvin Bansal	4	4	4.5		4	4	
		102184009	Ojas Nagta	4.5	4.5	4	4		4.5	

Name and Signature of the evaluation committee member(s):

Evaluator-1:

Evaluator-2: Ganguli

Evaluator-3:

Thapar Institute of Engineering and Technology, Patiala
Electrical and Instrumentation Engineering Department
Progress Report Form (Electrical Engineering)

Group No: 15
Project Title: VR Haptic Gloves

Name of the Supervisor: Dr. Parag Nijhawan

Date	Work done/discussed	Roll No.	Student name	Self-Evaluation (5)	Peer Evaluation M (5)	Peer Evaluation Z (5)	Peer Evaluation L (5)	Peer Evaluation D (5)	Supervisor Evaluation (5)	Supervisor's Sign with remarks
4-04-23	Evaluated weight of servos and bulkiness of the modules that will be merged onto haptic gloves	102004114	Melvin Bansal	4		4	4.25	4	4	Parag.
		102004111	Zenith	4.25	4	'	4.25	4	4	
		102004086	Lakshay Sablok	4	4	4		4	4	
		102184009	Ojas Nagta	4	4	4	4.25		4	
8-04-23	Electrical calculation: discussed on whether the rated power supply can power all the servos in idle,no load and full load condition	102004114	Melvin Bansal	4.25		4	4	3.75	4	Parag.
		102004111	Zenith	4	3.25		4	4	4	
		102004086	Lakshay Sablok	4.25	4	4		4	4	
		102184009	Ojas Nagta	4	4	4	4		4	
18-04-23	Collected all the 3d printed parts that will hold and support the haptic module	102004114	Melvin Bansal	4.25	[illegible]	4.25	4.25	4.5	4.25	Parag.
		102004111	Zenith	4	4		4.25	4	4.25	
		102004086	Lakshay Sablok	4.25	4	4.25		4	4.25	
		102184009	Ojas Nagta	4.5	4	4.25	4.25		4.25	
27-04-23	Construction of hardware module: Constructed and assembled 3d parts to make mechanical spindle that will hold potentiometers and servomotors	102004114	Melvin Bansal	4		4.25	4	4	4	Parag
		102004111	Zenith	4	4.1		4	4	4	
		102004086	Lakshay Sablok	4.25	4	4.25		4	4	
		102184009	Ojas Nagta	3.75	4	4.25	4		4	
28-0-23	Mounted 2 of 5 mechanical spindle onto the holder and equipped gloves with end caps & guide nodes	102004114	Melvin Bansal	4.25		4.25	4.25	3.5	4.25	Parag.
		102004111	Zenith	4	4		4.25	3.75	4.25	
		102004086	Lakshay Sablok	4.25	4	4.25		4	4.25	
		102184009	Ojas Nagta	3.75	4	4.25	4.25		4.25	
30-02-23	Constructed rest of the haptic modules remaining and mounted it on the holder	102004114	Melvin Bansal	4	•	4.25	4.25	4	4.25	Parag.
		102004111	Zenith	4	4		4.25	4	4.25	
		102004086	Lakshay Sablok	4.25	4	4.25		4	4.25	
		102184009	Ojas Nagta	4	4	4.25	4.25		4.25	

Name and Signature of the evaluation committee member(s):

Evaluator-1: Evaluator-2: Evaluator-3:

Thapar Institute of Engineering and Technology, Patiala
Electrical and Instrumentation Engineering Department
Progress Report Form (Electrical Engineering)

Group No: 15
Project Title: VR Haptic Gloves

Name of the Supervisor: Dr. Parag Nijhawan

Date	Work done/discussed	Roll No.	Student name	Self-Evaluation (5)	Peer Evaluation L	Peer Evaluation Z	Peer Evaluation M	Peer Evaluation O	Supervisor Evaluation (5)	Supervisor's Sign with remarks
					5	5	5	5		
3-11-23	soldered the haptic circuit onto the PCB.	102004086	Lakshay Sablok	3.5		3.75	3.75	3.75	3.5	
		102004111	Zenith	3.5	3.75		3.75	3.75	3.5	
		102004114	Melvin Bansal	3.75	3.75	3.75		3.75	3.5	
		102184009	Ojas Nagta	3.5	3.75	3.75	3.75		3.5	
12-11-23	continued the soldering of the haptic circuit onto the PCB.	102004086	Lakshay Sablok	3.75		4.5	4.5	4.5	3.75	
		102004111	Zenith	3.75	4.5		4.5	3.75	3.75	
		102004114	Melvin Bansal	3.75	3.75	4.5		3.75	3.75	
		102184009	Ojas Nagta	3.75	3.75	4.5	4.5		3.75	
19-11-23	set up VR finger tracking and haptic drivers with an environment for simulation testing.	102004086	Lakshay Sablok	4.5		3.5	4.5	3.75	3.5	
		102004111	Zenith	3.5	3.75		4.5	4.5	3.5	
		102004114	Melvin Bansal	4.5	3.75	3.5		4.5	3.5	
		102184009	Ojas Nagta	3.75	4.5	3.5	4.5		3.5	
28-11-23	analyzed and changed the finger tracking module to make movement more efficient.	102004086	Lakshay Sablok	3.75		4.5	3.5	4.5	4	
		102004111	Zenith	4.5	4.5			4.5	4	
		102004114	Melvin Bansal	4.5	4.5	4.5	3.5	4.5	4	
		102184009	Ojas Nagta	4.5	4.5	4.5	3.5		4	
3-12-23	3d reprinted finger tracking model after careful evaluation.	102004086	Lakshay Sablok	4.25		3.75	3.75	4.25	4.25	
		102004111	Zenith	4.25	3.5		3.75	4.25	4.25	
		102004114	Melvin Bansal	3.75	2.5	3.75		4.5	4.25	
		102184009	Ojas Nagta	4.25	3.5	3.75	3.75		4.25	
5-12-23	Worked on simulation of haptic modules through VR.	102004086	Lakshay Sablok	4.5		4.5	4.5	4.5	3.75	
		102004111	Zenith	3.75	4.5		4.5	3.75	3.75	
		102004114	Melvin Bansal	4.5	4.5	4.5		4.5	3.75	
		102184009	Ojas Nagta	3.75	4.5	4.5	4.5		3.75	

Name and Signature of the evaluation committee member(s):

Evaluator-1: **Evaluator-2:** **Evaluator-3:**

List of COs/POs/PSOs

CO-PO and PSO mapping Chart

Sr. No.	POs/PSOs	CO's	Description
1	PO1, PO2, PO3, PO5, PO9, PO11, PO12	CO1	The project involves comprehensive design considerations for VR haptic gloves, addressing specific goals such as providing realistic haptic feedback and ensuring compatibility with existing VR systems.
2	PO1, PO2, PO3, PO5, PO9, PO11, PO12	CO2	The project necessitates an integrated design approach, requiring the application of electrical engineering principles to achieve a holistic development of VR haptic gloves.
3	PO1, PO2, PO3, PO5, PO9, PO11, PO12	CO3	Extensive literature analysis and exploration of various technologies involve simulations and iterative synthesis to refine design elements in the project.
4	PO1, PO2, PO3, PO5, PO9, PO11, PO12	CO4	The project entails the utilization of modern engineering tools, encompassing sensors, actuators, and communication protocols, for the development of functional VR haptic gloves.
5	PO1, PO2, PO3, PO5, PO9, PO11, PO12	CO5	Collaborative efforts among team members are integral, considering the multidisciplinary nature of developing VR haptic gloves to ensure a comprehensive and effective outcome.
6	PO1, PO2, PO3, PO5, PO9, PO11, PO12	CO6	The meticulous documentation of technical specifications and expected deliverables in the project contributes to the refinement of technical documentation and presentation skills.

List of COs/POs/PSOs:

Course Outcomes (COs):

CO1: To meticulously identify design goals and analyze potential approaches to meet given specifications with realistic engineering constraints.

CO2: To intricately design an electrical engineering project by implementing an integrated design approach and applying knowledge accrued in various professional courses.

CO3: To perform thorough simulations and incorporate appropriate adaptations using iterative synthesis, refining design elements for optimal outcomes.

CO4: To adeptly use modern engineering hardware and software tools, including sensors, actuators, and communication protocols, for the comprehensive development of VR haptic gloves.

CO5: To collaboratively work as an integral member of an engineering design team, acknowledging and addressing the multidisciplinary aspects involved in the development process.

CO6: To continually enhance technical documentation and presentation skills through meticulous documentation of technical specifications and expected deliverables in the project.

Program Outcomes (POs):

PO1: Engineering knowledge: Apply a profound understanding of mathematics, science, engineering fundamentals, and an engineering specialization to solve intricate engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems, reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet specified needs, considering public health and safety, and cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods, including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling, to complex engineering activities, demonstrating an understanding of the limitations.

PO9: Individual and team work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of engineering and management principles, applying these to one's work as a member and leader in a team, managing projects in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs):

PSO1: To apply knowledge of mathematics, sciences, and professional subjects to formulate, interpret, and analyze problems appropriate to Electrical Engineering, fostering a deep understanding of the subject matter.

PSO2: To employ appropriate engineering techniques, skills, tools, and research-based knowledge to accomplish electrical engineering tasks and engage in life-long learning for continuous professional development, adapting to evolving technological landscapes.

REFERENCES

1. J. Lin, B. Hu, J. Pang and J. Liao, "The design of a mechanical force feedback VR data glove," 2022 IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS), Chongqing, China,2022,pp.327-334,doi:10.1109/IUCC-CIT-DSCI-SmartCNS57392.2022.00058.

2. Zhang R. Research on force feedback data glove based on electromagnetic force [D]. Supervisor: Longqin Gao. Yangzhou University, 2012.

3. Wang H-T, Du H-W, Xiong W, Wang Z-W. Design and verification of aerodynamic feedback data glove. Journal of Computer Aided Design and Graphics, 2012, (11):1493-1499.

4. Wang Shu-Fu, Xie Zong-Wu, Fang Hong-Gen, Zhang Ji-Zhen. An FPGA-based force feedback data glove. Mechanics and Electronics, 2010, (07):56-59.

5. What is the principle of grasping things with human hands? And how was the manipulator created? From Ren Zhenyu's answer [EB/OL]. 1/2/2012 [9/9/2022]. https://www.zhihu.com/question/37352068.

6. Wang Z, Sun Y, Wu L.F. Application of ESP32 in teaching IoT communication course. Internet of Things Technology, 2022, (04):128- 129+132.

7. Chinxue2008. ESP32 core schematic [EB/OL]. 2021 [9/9/2022]. https://blog.csdn.net/wei-xin_37281967/article/details/115255070.

8. Bräker, Julia, and Julia Hertel. 2022. "Conceptualizing Interactions of Augmented Reality Solutions." *Proceedings of the 55th Hawaii International Conference on System Sciences* 1, no. 1 (55th Hawaii International Conference): 20. https://doi.org/10.24251/HICSS.2022.088.

9. Burns Martin et al. "Design and Implementation of an Instrumented Data Glove that measures Kinematics and Dynamics of Human Hand.". Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference 2021(2021): 7229-7232.

10. Lucas LucidVR. LucidGloves – VR Haptic Gloves on a budget [EB/OL]. 4/1/2022 [9/9/2022]. https://hackaday.io/project/178243-lucid.

11. Burdea, G. C., & Coiffet, P. (2003). Virtual Reality Technology (2nd ed.). Wiley-Interscience.

12. Inaugment. 2023."Augmented reality infrastructure: How It Works and Its Benefits."Inaugment.https://www.inaugment.com/augmented-reality-infrastructure/.

13. Prattichizzo, D., & Chinello, F. (2012). Towards Wearable Haptics: A Compact Pneumatic Shape Display for the Fingertip. IEEE Transactions on Haptics, 5(4), 289-300.

14. DesignsRock Editorial. 2019. "How AI Is Revolutionizing Web Design." ArtificialIntelligencesoftware.https://designsrock.org/how-ai-is-revolutionizing-web-design/?amp=1.

15. Sato, M., & Yamamoto, S. I. (2018). Hand-arm motion generator for virtual reality haptic interface. International Journal of Advanced Robotic Systems, 15(5), 1729881418784934.

16. Pasquero, J., Furlan, A., & Prattichizzo, D. (2015). A haptic arm exoskeleton for virtual reality and teleoperation. IEEE Transactions on Haptics, 8(2), 140-151.

17. Steed, A., & Slater, M. (2009). A virtual presence counter. Presence: Teleoperators and Virtual Environments, 18(5), 452-464.

18. Jones, L. A., & Sarter, N. B. (2008). Tactile Displays: Guidance for Their Design and Application. Human Factors: The Journal of the Human Factors and Ergonomics Society, 50(1), 90-111.

19. Kim, D., & Gleeson, B. (2018). A novel approach to integrating touch into a virtual reality headset. IEEE Transactions on Visualization and Computer Graphics, 24(4), 1584-1591.

20. McMahon, M., Jones, L. A., & Sarter, N. B. (2018). Tactile Feedback in Flight Simulation: The Impact of Different Tactile Feedback Characteristics on Pilot Performance. Aerospace Medicine and Human Performance, 89(10), 898-907.

21. Nisky, I., & Buxbaum, L. J. (2014). Human Haptic Perception is Interrupted by Exploratory Stops of Hand and Finger Motion. Journal of Neuroscience, 34(8), 2722-2730.

22. Guna, J., Geršak, G., & Humar, I. (2014). A survey of wearable and wireless glove-based systems for rehabilitation, training, and virtual reality. Journal of NeuroEngineering and Rehabilitation, 11(1), 1-27.

23. Gao, J., Li, Y., & Zhang, X. (2021). Design and implementation of VR haptic glove for rehabilitation of hand function. Frontiers in Neurology, 12, 729840.

24. Raisamo, R., Salminen, K., Rantala, J., & Surakka, V. (2006). Gestures in Human-Computer Interaction and Simulation. Springer

ANNEXURE

CODE:

<u>firmware.ino</u>

```
#include "AdvancedConfig.h"
#define COMMUNICATION COMM_SERIAL
 #define SERIAL_BAUD_RATE 115200
 #define BTSERIAL_DEVICE_NAME "gloves-left"
#define FLIP_POTS  true
#define TRIGGER_GESTURE true
#define GRAB_GESTURE    true
#define PINCH_GESTURE   true
#define INVERT_A false
#define INVERT_B false
#define INVERT_JOY false
#define INVERT_MENU false
#define INVERT_CALIB false
#define INVERT_TRIGGER false
#define INVERT_GRAB false
#define INVERT_PINCH false
#define NO_THUMB true
#define USING_CALIB_PIN false
#define USING_FORCE_FEEDBACK false
#define SERVO_SCALING false
#if defined(ESP32)
  #define PIN_PINKY     32
  #define PIN_RING      33
  #define PIN_MIDDLE    34
  #define PIN_INDEX     35
  #define PIN_THUMB     36
  #define PIN_JOY_X     30
  #define PIN_JOY_Y     25
  #define PIN_JOY_BTN   26
  #define PIN_A_BTN     27
  #define PIN_B_BTN     14
  #define PIN_TRIG_BTN  12
  #define PIN_GRAB_BTN  13
  #define PIN_PNCH_BTN  23
  #define PIN_CALIB     12
  #define DEBUG_LED 2
  #define PIN_PINKY_MOTOR     5
  #define PIN_RING_MOTOR      18
```

```
 #define PIN_MIDDLE_MOTOR    19
 #define PIN_INDEX_MOTOR     21
 #define PIN_THUMB_MOTOR     17
 #define PIN_MENU_BTN       27
#elif defined(__AVR__)
 #define PIN_PINKY     A0
 #define PIN_RING      A1
 #define PIN_MIDDLE    A2
 #define PIN_INDEX     A3
 #define PIN_THUMB     A4
 #define PIN_JOY_X     A6
 #define PIN_JOY_Y     A7
 #define PIN_JOY_BTN   7
 #define PIN_A_BTN     8
 #define PIN_B_BTN     9
 #define PIN_TRIG_BTN  10 //unused if gesture set
 #define PIN_GRAB_BTN  11 //unused if gesture set
 #define PIN_PNCH_BTN  12 //unused if gesture set
 #define PIN_CALIB     13 //button for recalibration
 #define DEBUG_LED     LED_BUILTIN
 #define PIN_PINKY_MOTOR     2 //used for force feedback
 #define PIN_RING_MOTOR      3 //^
 #define PIN_MIDDLE_MOTOR    4 //^
 #define PIN_INDEX_MOTOR     5 //^
 #define PIN_THUMB_MOTOR     6 //^
 #define PIN_MENU_BTN       8
#endif
```

Advances config.h

```
#define LOOP_TIME 4
#define CALIBRATION_LOOPS -1
#define COMM_SERIAL 0
#define COMM_BTSERIAL 1
#define ENCODING 1
#define ENCODE_LEGACY 0
#define ENCODE_ALPHA  1
#define PINKY_IND 4
#define RING_IND 3
#define MIDDLE_IND 2
#define INDEX_IND 1
#define THUMB_IND 0
#if defined(__AVR__)
```

```
#define ANALOG_MAX 1023
#elif defined(ESP32)
#define ANALOG_MAX 4095
#endif
//#define ANALOG_MAX 4095
#ifndef ANALOG_MAX
#endif
#define CLAMP_ANALOG_MAP true
#define CLAMP_FLEXION false
#define CLAMP_MIN 0
#define CLAMP_MAX ANALOG_MAX
#define ENABLE_MEDIAN_FILTER false
#define MEDIAN_SAMPLES 20
```

encoding.io

```
#if ENCODING == ENCODING_LEGACY
//legacy 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];
  sprintf(stringToEncode, "%d&%d&%d&%d&%d&%d&%d&%d&%d&%d&%d&%d&%d\n",
  flexion[0], flexion[1], flexion[2], flexion[3], flexion[4],
  joyX, joyY, joyClick,
  triggerButton, aButton, bButton, grab, pinch
  );
  return stringToEncode;
}
//legacy decoding
void decodeData(char* stringToDecode, int* hapticLimits){
    byte index = 0;
    char* ptr = strtok(stringToDecode, "&");  // takes a list of delimiters
    while(ptr != NULL)
    {
        hapticLimits[index] = atoi(ptr);
        index++;
        ptr = strtok(NULL, "&");  // takes a list of delimiters
    }
}
#endif
#if ENCODING == ENCODE_ALPHA
//alphabetic 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;
  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":""
  );
  return stringToEncode;
}
//legacy decoding
void decodeData(char* stringToDecode, int* hapticLimits){
  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
  //Serial.println("Haptic: "+ (String)hapticLimits[0] + " " + (String)hapticLimits[1] + " " +
(String)hapticLimits[2] + " " + (String)hapticLimits[3] + " " + (String)hapticLimits[4] + " ");
}
int getArgument(char* stringToDecode, char command){
  char* start = strchr(stringToDecode, command);
  if (start == NULL)
    return -1;
  else
    return atoi(start + 1);
#endif
```

icommunication.io

```
class ICommunication {
  public:
    virtual bool isOpen() = 0;
    virtual void start() = 0;
    virtual void output(char* data) = 0;
    virtual bool readData(char* input) = 0;
};
```

serialcommunication.io

```
class SerialCommunication : public ICommunication {
```

```
  private:
   bool m_isOpen;
  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();
   }
   bool readData(char* input){
    byte size = Serial.readBytesUntil('\n', input, 100);
    input[size] = NULL;
    return input != NULL && strlen(input) > 0;
   }
};
```

main.io

```
#define ALWAYS_CALIBRATING CALIBRATION_LOOPS == -1

#define CALIB_OVERRIDE false
ICommunication* comm;
int loops = 0;
void setup() {
  #if COMMUNICATION == COMM_SERIAL
   comm = new SerialCommunication();
  #elif COMMUNICATION == COMM_BTSERIAL
   comm = new BTSerialCommunication();
  #endif
  comm->start();

  setupInputs();

  #if USING_FORCE_FEEDBACK
```

```
    setupServoHaptics();
  #endif

}

void loop() {
  if (comm->isOpen()){
    #if USING_CALIB_PIN
    bool calibButton = getButton(PIN_CALIB) != INVERT_CALIB;
    if (calibButton)
      loops = 0;
    #else
    bool calibButton = false;
    #endif

    bool calibrate = false;
    if (loops < CALIBRATION_LOOPS || ALWAYS_CALIBRATING){
      calibrate = true;
      loops++;
    }

    int* fingerPos = getFingerPositions(calibrate, calibButton);
    bool joyButton = getButton(PIN_JOY_BTN) != INVERT_JOY;

    #if TRIGGER_GESTURE
    bool triggerButton = triggerGesture(fingerPos);
    #else
    bool triggerButton = getButton(PIN_TRIG_BTN) != INVERT_TRIGGER;
    #endif

    bool aButton = getButton(PIN_A_BTN) != INVERT_A;
    bool bButton = getButton(PIN_B_BTN) != INVERT_B;

    #if GRAB_GESTURE
    bool grabButton = grabGesture(fingerPos);
    #else
    bool grabButton = getButton(PIN_GRAB_BTN) != INVERT_GRAB;
    #endif

    #if PINCH_GESTURE
    bool pinchButton = pinchGesture(fingerPos);
    #else
    bool pinchButton = getButton(PIN_PNCH_BTN) != INVERT_PINCH;
    #endif
```

```
    bool menuButton = getButton(PIN_MENU_BTN) != INVERT_MENU;

    comm->output(encode(fingerPos, 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() >= 10) {
           decodeData(received, hapticLimits);
           writeServoHaptics(hapticLimits);
        }
      }
    #endif
    delay(LOOP_TIME);
  }
}
```

gesture.io

```
bool grabGesture(int *flexion){
  return (flexion[PINKY_IND] + flexion[RING_IND] + flexion[MIDDLE_IND] + flexion[INDEX_IND]) / 4
<= ANALOG_MAX/2 ? 0:1;
}
bool pinchGesture(int *flexion){
  return (flexion[INDEX_IND] + flexion[THUMB_IND]) / 2 <= ANALOG_MAX/2 ? 0:1;
}
bool triggerGesture(int *flexion){
  return flexion[INDEX_IND]<=(ANALOG_MAX/2)?0:1;
}
```

haptics.io

```
#if USING_FORCE_FEEDBACK
#if defined(ESP32)
  #include "ESP32Servo.h"
#else
  #include "Servo.h"
#endif
Servo pinkyServo;
```

```
Servo ringServo;
Servo middleServo;
Servo indexServo;
Servo thumbServo;
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);
}
void scaleLimits(int* hapticLimits, float* scaledLimits){
  for (int i = 0; i < 5; i++){
    scaledLimits[i] = 180.0f - hapticLimits[i] / 1000.0f * 180.0f;
  }
}
void dynScaleLimits(int* hapticLimits, float* scaledLimits){
  for (int i = 0; i < sizeof(hapticLimits); i++){
    scaledLimits[i] = hapticLimits[i] / 1000.0f * 180.0f;
  }
}
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]);
}
#endif
```

input.io

```
#if ENABLE_MEDIAN_FILTER
  #include <RunningMedian.h>
  RunningMedian rmSamples[5] = {
     RunningMedian(MEDIAN_SAMPLES),
     RunningMedian(MEDIAN_SAMPLES),
     RunningMedian(MEDIAN_SAMPLES),
     RunningMedian(MEDIAN_SAMPLES),
     RunningMedian(MEDIAN_SAMPLES)
  };
```

```
#endif
int maxFingers[5] = {0,0,0,0,0};
int minFingers[5] = {ANALOG_MAX, ANALOG_MAX, ANALOG_MAX, ANALOG_MAX,
ANALOG_MAX};
void setupInputs(){
  pinMode(PIN_JOY_BTN, INPUT_PULLUP);
  pinMode(PIN_A_BTN, INPUT_PULLUP);
  pinMode(PIN_B_BTN, INPUT_PULLUP);
  pinMode(PIN_MENU_BTN, INPUT_PULLUP);
   #if !TRIGGER_GESTURE
  pinMode(PIN_TRIG_BTN, INPUT_PULLUP);
  #endif
  #if !GRAB_GESTURE
  pinMode(PIN_GRAB_BTN, INPUT_PULLUP);
  #endif
  #if !PINCH_GESTURE
  pinMode(PIN_PNCH_BTN, INPUT_PULLUP);
  #endif
  #if USING_CALIB_PIN
  pinMode(PIN_CALIB, INPUT_PULLUP);
  #endif
}

int* getFingerPositions(bool calibrating, bool reset){
  int rawFingers[5] = {NO_THUMB?0:analogRead(PIN_THUMB), analogRead(PIN_INDEX),
analogRead(PIN_MIDDLE), analogRead(PIN_RING), analogRead(PIN_PINKY)};
  //flip pot values if needed
  #if FLIP_POTS
  for (int i = 0; i <5; i++){
    rawFingers[i] = ANALOG_MAX - rawFingers[i];
  }
  #endif
  #if ENABLE_MEDIAN_FILTER
  for (int i = 0; i < 5; i++){
    rmSamples[i].add( rawFingers[i] );
    rawFingers[i] = rmSamples[i].getMedian();
  }
  #endif
  //reset max and mins as needed
  if (reset){
    for (int i = 0; i <5; i++){
      maxFingers[i] = 0;
      minFingers[i] = ANALOG_MAX;
    }
```

```
 }
 //if during the calibration sequence, make sure to update max and mins
 if (calibrating){
  for (int i = 0; i <5; i++){
   if (rawFingers[i] > maxFingers[i])
    #if CLAMP_FLEXION
     maxFingers[i] = ( rawFingers[i] <= CLAMP_MAX )? rawFingers[i] : CLAMP_MAX;
    #else
     maxFingers[i] = rawFingers[i];
    #endif
   if (rawFingers[i] < minFingers[i])
    #if CLAMP_FLEXION
     minFingers[i] = ( rawFingers[i] >= CLAMP_MIN )? rawFingers[i] : CLAMP_MIN;
    #else
     minFingers[i] = rawFingers[i];
    #endif
  }
 }
 static int calibrated[5] = {511,511,511,511,511};
 for (int i = 0; i<5; i++){
  if (minFingers[i] != maxFingers[i]){
   calibrated[i] = map( rawFingers[i], minFingers[i], maxFingers[i], 0, ANALOG_MAX );
   #if CLAMP_ANALOG_MAP
    if (calibrated[i] < 0)
     calibrated[i] = 0;
    if (calibrated[i] > ANALOG_MAX)
     calibrated[i] = ANALOG_MAX;
   #endif
  }
  else {
   calibrated[i] = ANALOG_MAX / 2;
  }
 }
 return calibrated;
}
int analogReadDeadzone(byte pin){
 int raw = analogRead(pin);
 if (abs(ANALOG_MAX/2 - raw) < JOYSTICK_DEADZONE * ANALOG_MAX / 100)
  return ANALOG_MAX/2;
 else
  return raw;
}
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;
}
```

Parag

2023 Capstone Report

ORIGINALITY REPORT

17%	2%	13%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Jiliang Lin, Bin Hu, Jiahao Pang, Jingjie Liao. "The design of a mechanical force feedback VR data glove", 2022 IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS), 2022 Publication	10%
2	Submitted to Kocaeli Üniversitesi Student Paper	3%
3	"Internet of Things. Advances in Information and Communication Technology", Springer Science and Business Media LLC, 2024 Publication	1%
4	www.scilit.net Internet Source	1%
5	Submitted to University of Salford Student Paper	<1%
6	infuture1.blogspot.com Internet Source	<1%
7	ims.ut.ee	

	Internet Source	<1 %
8	Submitted to AlHussein Technical University Student Paper	<1 %
9	Submitted to Deakin University Student Paper	<1 %
10	"Virtual and Augmented Reality (VR/AR)", Springer Science and Business Media LLC, 2022 Publication	<1 %
11	"Virtual, Augmented and Mixed Reality. Design and Interaction", Springer Science and Business Media LLC, 2020 Publication	<1 %
12	cs.stir.ac.uk Internet Source	<1 %
13	tudr.thapar.edu:8080 Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		