Ashton Billings, Hamza Lutfi, and Aditya Nebhrajani

ECE 445 Proposal

Spring 2025

TA: Jason Zhang

March 7, 2025

Contents

1 Introduction

1.1 Problem

Although research into VR has been done for many years now, even here at this school, its only until the late 2010's when VR technology advanced enough to be a marketable technology. These initial devices, such as the HTC Vive and the Oculus Rift, were heavy, wired devices, that often needed a dedicated room and dedicated sensors spaced around the room to properly operate. The graphics left much to be desired, and the youth of the industry meant there weren't many virtual experiences out there that could work with such devices. There was also a big problem with VR sickness, where many couldn't properly have these experiences due to motion sickness. Another area that was lacking, the area that most interests us here, is that despite these being virtual reality experiences, users still needed to use controllers with buttons and joysticks to navigate this virtual world.

As of now, these devices have advanced greatly, solving many of these issues listed. Much of this is thanks to great interest and investment into this market, with big moves such as Facebook buying Oculus and re-branding themselves to Meta; completely shifting there companies R&D to this market. Issues like needing a dedicated room with cameras spaced around it have been worked out, with most flagship devices being able to work stationary, with no external devices required for tracking. Issues like wireless-ness have also been improved upon, with Meta introducing there Meta Quest series of headsets that require no cable, as well as creating there "Air-Link" software that allows for a wireless connection of these devices to an external PC. Despite these improvements on the headset side of things, the controllers given with most flagship devices haven't changed all that much since these devices inception. Most flagship devices still use the single handed controller with buttons and a joystick, with certain exceptions such as the Valve Index, which although is very much still a handheld controller, it also offers finger tracking, as well as the Apple Vision Pro not using controllers at all, instead tracking hands via the external cameras. There are also many smaller companies producing hand tracking devices, however most are quite large and very expensive. The best of these we've found is the UDCAP Kickstarter project, which offers joint tracking gloves that contain buttons and a joystick such that they can be used similarly to regular controllers. This Kickstarter project, as will be seen, will be our main basis of inspiration and improvement. It is also important to mention the open source VR glove project LucidGloves, which we also have taken much inspiration from.

All of these alternatives to regular controllers mentioned more or less are aimed at one goal, making these virtual experiences feel real by attempting to create the ability to "feel". A device that allows you to grab and hold objects, and for it to feel like your holding them, while still being able to navigate this virtual environment as you would with handheld controllers.

1.2 Solution

Our solution to this problem is to develop our own hand tracking gloves compatible for the VR devices already on the market. As already mentioned, we will be taking great inspiration to the UDCAP Kickstarter project, so it would be useful to first go into there current solution. The UDCAP gloves offer joint tracking through proprietary sensors they call "elastic sensors" laid over each finger joint. We believe these to be some form of stress gauge sensor. on the side of the pointer finger, you can attach a module containing a hall effect joystick and buttons, allowing for regular controller operations. the glove is wireless communicating over 2.4GHz with a latency of ¡10ms. It has a rechargeable battery with a battery capacity of 760mAh, claiming to have 10-15 hours of usage per charge. To position track these gloves, you need an external tracker such as the Vive Tracker or a Quest controller strapped to the glove which they sell connection adapters for. The glove is compatible with SteamVR, which allows it to work for any PC game on any mainstream headset.

Although UDCAP is a Kickstarter project, there team contains 40-50 engineers and have been developing this product for the last three years. As such, we are aware we cannot make a better product, so instead we plan to imitate the product as best we can as well as doing certain things differently. We will be doing something similar to the kickstarter project where we track finger curl. Rather than using elastic sensors, we will opt for potentiometers to track finger curl. We will create an index finger module containing buttons and a joystick allowing for normal controller actions to be done with these gloves. We will be making these gloves wireless, using a Li-ion battery with 2000mAh at 7.4V which we will step down with LDOs. For hand tracking, we will do as the UDCAP gloves do by strapping an oculus quest controller to our glove. To wirelessly communicate data back and forth, we will be using BLE which communicates at 3.4GHz. From this point on, this is where our design will begin to deviate from the UDCAP gloves. The first main difference is that we plan to create a primitive haptic system, inspired by a mix of the haptic solutions of the LucidGlovesV3 [8] and LucidGlovesV5 [9]. This haptic system will contain strapping badge reels to each finger, and using a servo motor to lock the reel in place when our fingers collide with an object in game.

Our solution cannot be tested without some software to test our glove on. In order to test our glove, we will be using the Unity game engine as it has easy VR game development tools in which we will make a virtual scene containing objects to interact with as well as virtual hands. We will be using a Meta Quest 2 to deploy this virtual scene too. In specific, we will be taking in our BLE data via a BLE plugin for Unity. Once its in Unity, we will register the buttons and joystick through Unity's Input Action Manager, as well as register the glove data through a custom Unity VRHands package provider. With this, our gloves inputs will be able to effect our virtual scene as well as control a virtual hand. In Unity, objects know when they interact through colliders. When our fingers colliders trigger, we will send a signal to our gloves through BLE to tell the servo motors to lock, giving the illusion of holding virtual objects.

1.3 Visual Aid

1.4 High-level Requirements

For our design to be successful, our gloves must meet the following requirements:

1. We can pick up virtual objects with our gloves, with them properly locking in place for haptic feedback.

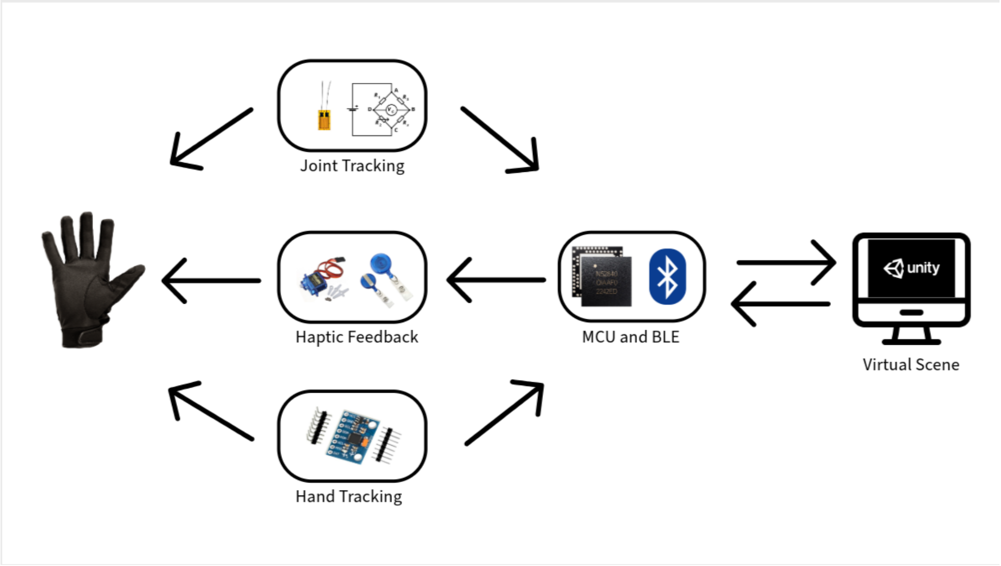

Figure 1: Visual-Aid

2. The communication between the gloves and software have a latency of ¡500ms. This latency is low enough such that our actions to the virtual won't mess up testing.
3. The accuracy of our virtual finger curl is +/- 15% of our actual finger curl.

2 Design

2.1 Block Diagram

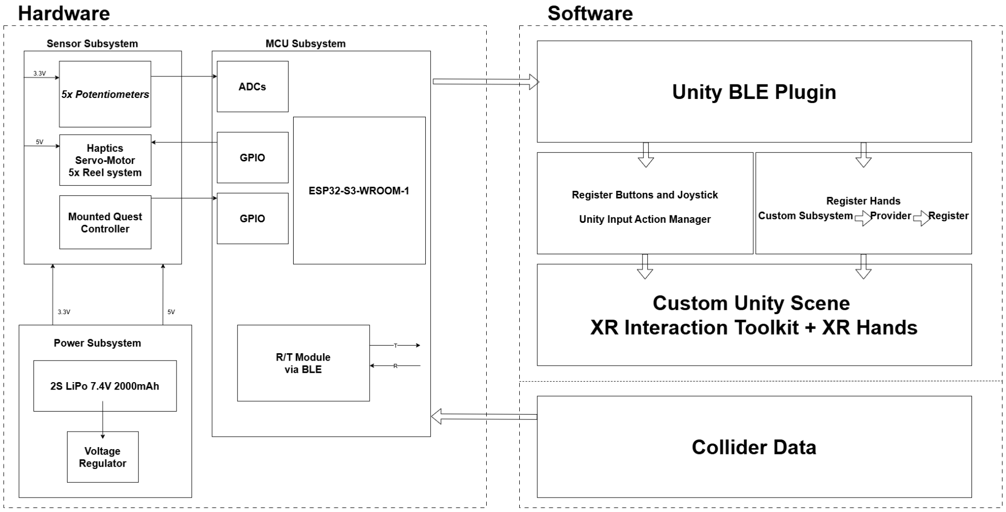

Figure 2: Visual-Aid

2.1.1 Power Subsystem

The power subsystem is responsible for supplying stable and efficient power to all components of the VR glove, enabling wireless operation via a Li-ion battery. This battery was selected for its high energy density and ability to sustain several hours of continuous use, even under peak load conditions.

To generate the required voltage levels, the system includes a low-dropout regulator that provides 5V and 3.3V for the ESP32-S3 microcontroller and potentiometer sensors, ensuring stable ADC readings. These voltage rails allow the glove to function efficiently while preventing unnecessary power dissipation.

Requirement	**Verification**
The power subsystem must provide regulated 3.3V ± 5% for low-power components and 5V ± 5% for servos.	Measure voltage under full load using a DMM. Verify it stays within 3.14V - 3.47V and 4.75V - 5.25V.
The battery system must last at least 3 hours under a 1A load.	Fully charge the battery, apply a 1A load, and measure operation time. Verify runtime is 3 hours.
The BMS must cut off charging when the battery voltage reaches 8.4V ± 0.1V.	Use a variable power supply to simulate battery voltage. Increase to 8.4V and confirm charging stops.

Table 1: Power Subsystem Requirements Verification

2.1.2 Sensor Subsystem

The sensor subsystem is responsible for tracking finger movements, hand orientation, and haptic feedback. It consists of three primary components: potentiometers for joint tracking, a Quest Controller for hand positioning, and a servo motor + reel system for haptic feedback.

For finger tracking, each finger joint is equipped with a potentiometer, which measures resistance changes as fingers bend. These analog signals are converted into digital values via the ADCs, providing tracking of each finger's curl.

For hand positioning, the Meta Quest controller will be mounted to the glove. and directly connected to the software via the XR Interaction Toolkit.

For haptic feedback, the glove employs a servo motor + reel system. Each finger is connected to a retractable badge reel, which extends and retracts freely. When a virtual object is grasped, the MCU receives collision data from Unity via BLE and locks the reel using a servo motor, restricting finger movement to simulate physical contact.

2.1.3 MCU Subsystem

The MCU subsystem is the central processing unit of the VR glove, responsible for sensor data acquisition, BLE communication, and haptic control.

For sensor data processing, the MCU continuously reads analog signals from the five potentiometers via its ADC, converting finger movements into digital values.

For BLE communication, the ESP32 transmits sensor data to a Unity-based VR application using a custom Bluetooth Low Energy protocol. BLE ensures low-latency wireless communication, crucial for real-time hand tracking. Additionally, the MCU receives haptic feedback signals from Unity when a virtual object is touched, triggering the servo motor system to restrict finger movement.

For haptic control, the MCU outputs PWM signals via GPIOs to drive the servo motors, which lock the badge reel system, simulating force feedback.

2.1.4 Controller Subsystem

The Controller Subsystem provides traditional input controls for the VR glove by integrating buttons and a joystick mounted on the index finger. This allows users to perform standard VR controller actions such as movement, interaction, and menu navigation, enabling seamless compatibility with existing VR applications.

The joystick is a two-axis (X/Y) Hall-effect thumbstick, which provides smooth analog input for directional movement. It outputs two independent voltage signals, corresponding to X and Y-axis displacement, which are read by the ESP32-S3's ADCs. This allows Unity to map joystick movement directly to in-game navigation.

The system also includes two to four tactile push buttons for additional inputs, such as jumping, grabbing, or interacting with objects. These buttons are digital inputs that interface with the MCU's GPIOs, registering as high or low when pressed or released.

2.1.5 Software Subsystem

The software subsystem is responsible for processing sensor data, integrating the VR glove into the Unity XR Hands system, and managing haptic feedback. This ensures seamless interaction between the glove and virtual environments.

At its core, the Unity BLE Plugin receives sensor data via Bluetooth Low Energy from the ESP32-S3 MCU. Unity's Input Action Manager registers joystick and button inputs from the glove's mounted Quest controller, allowing traditional controller functions.

To integrate the glove into Unity's XR Interaction Toolkit + XR Hands, a custom XRHandSubsystem provider is implemented. This module processes raw BLE data and translates it into a format compatible with the XR Hands API, enabling real-time hand tracking. The glove's virtual hand movements are rendered in a Unity scene, synchronized with the user's actual hand position.

For haptic feedback, Unity tracks virtual hand-object collisions using collider data. When a collision occurs, Unity sends a haptic trigger signal via BLE to the MCU, activating the servo motor + reel system to restrict finger movement, simulating physical interaction.

2.2 Tolerance Analysis: Power Supply Stability for ESP32 Module

2.2.1 Critical Block Selection

For the success of the VR glove, a stable power supply is critical. The ESP32 module, sensors, and wireless communication rely on a reliable voltage source. The most challenging requirement is ensuring that the power supply can maintain 3.3V and 5V outputs within acceptable tolerances despite variations in input voltage, component tolerances, and load changes.

This analysis focuses on the **battery and voltage regulation circuit** to ensure it meets the VR glove's power requirements.

2.2.2 System Overview

From the PCB schematic:

- **Battery (BT1)**: 7.4V LiPo (2000mAh)
- **5V Regulator (LM2956S-5)**: Steps down the LiPo voltage to 5V for the servo motors
- **3.3V Regulator (AMS1117-3.3)**: Converts 5V to 3.3V for the ESP32

2.2.3 Key Parameters and Component Tolerances

Component	Nominal Value	Tolerance	Possible Variation
LiPo Battery (BT1)	7.4V	±13.5%	6.4V – 8.4V
LM2956S-5 Output	5V	±5%	4.75V – 5.25V
AMS1117-3.3 Output	3.3V	±5%	3.135V – 3.465V

Table 2: Voltage Tolerances of Key Components

The ESP32 requires a stable 3.3V (±9%) supply. If the **AMS1117-3.3 regulator** fails to stay within tolerance, the ESP32 could experience instability.

2.3 Mathematical Analysis

2.3.1 Voltage Regulation from Battery to 5V

The LM2956S-5 provides an output voltage within the 4.75V to 5.25V range, which falls within the acceptable operating voltage range of the servo motor (4.8V to 6V), ensuring reliable operation. The AMS1117-3.3 provides a maximum output voltage of 3.465V, which is safely within the maximum recommended voltage of 3.6V for the ESP32's power pin, ensuring stable and reliable operation.

2.3.2 Feasibility and Mitigation

2.3.3 AMS1117-3.3 Requires a Proper Dropout Voltage

- The AMS1117 has a dropout voltage of $\sim$1.1V, meaning the input must remain above 4.4V for stable operation.
- Since $V_{5V,\text{min}} = 4.85V$, the regulator will still work, but at the edge of its specification.

2.3.4 Critical Block Selection

For the success of the VR glove, a stable power supply is critical. The ESP32 module, sensors, and wireless communication rely on a reliable voltage source. The most challenging requirement is ensuring that the power supply can maintain 3.3V and 5V outputs within acceptable tolerances despite variations in input voltage, component tolerances, and load changes.

This analysis focuses on the **battery and voltage regulation circuit** to ensure it meets the VR glove's power requirements.

2.3.5 System Overview

From the PCB schematic:

- **Battery (BT1)**: 7.4V LiPo (2000mAh)
- **5V Regulator (LM2956S-5)**: Steps down the LiPo voltage to 5V for the servo motors
- **3.3V Regulator (AMS1117-3.3)**: Converts 5V to 3.3V for the ESP32

2.3.6 Key Parameters and Component Tolerances

Component	**Nominal Value**	**Tolerance**	**Possible Variation**
LiPo Battery (BT1)	7.4V	±13.5%	6.4V – 8.4V
LM2956S-5 Output	5V	±5%	4.75V – 5.25V
AMS1117-3.3 Output	3.3V	±5%	3.135V – 3.465V

Table 3: Voltage Tolerances of Key Components

The ESP32 requires a stable 3.3V (±9%) supply. If the **AMS1117-3.3 regulator** fails to stay within tolerance, the ESP32 could experience instability.

2.4 Mathematical Analysis

2.5 Voltage Regulation from Battery to 5V

The LM2956S-5 provides an output voltage within the 4.75V to 5.25V range, which falls within the acceptable operating voltage range of the servo motor (4.8V to 6V), ensuring reliable operation. The AMS1117-3.3 provides a maximum output voltage of 3.465V, which is safely within the maximum recommended voltage of 3.6V for the ESP32's power pin, ensuring stable and reliable operation.

2.6 Feasibility and Mitigation

2.7 AMS1117-3.3 Requires a Proper Dropout Voltage

- The AMS1117 has a dropout voltage of ∼1.1V, meaning the input must remain above 4.4V for stable operation.
- Since $V_{5V,\text{min}} = 4.85V$, the regulator will still work, but at the edge of its specification.

2.8 Potentiometers for Finger Flex Sensing

The potentiometers (POTs) in our VR glove are critical components as they translate mechanical movement (finger flexing) into an analog voltage signal. The ESP32-S3's **ADC (Analog-to-Digital Converter)** reads this voltage and converts it into a digital value for processing. This analysis focuses on:

- Ensuring that the voltage range from the POTs is within **ESP32's ADC input range**
- Evaluating the impact of POT tolerance on measurement accuracy.
- Deriving an equation to map ADC values correctly.

2.9 System Overview

- **Input Voltage**: 3.3V supplied to the POTs.
- **Output Voltage Range**: 0V to 3.3V
- **ESP32-S3 ADC Resolution**: 12-bit
- **Max ADC Value**: $2^{12} = 4095$

2.10 Voltage Mapping Equation

2.11 General Formula for Normalized Mapping

Given:

- ADC_{min} = ADC reading when **fully contracted** (should map to 0)
- ADC_{max} = ADC reading when **fully flexed** (should map to 1)
- ADC_{value} = Current ADC reading

We define the **normalized flex mapping function** as:

$$Flex\_Value = \frac{ADC_{value} - ADC_{min}}{ADC_{max} - ADC_{min}} \tag{1}$$

2.12 Explanation

- When $ADC_{value} = ADC_{max}$, the equation simplifies to:

  $$Flex\_Value = \frac{ADC_{max} - ADC_{min}}{ADC_{max} - ADC_{min}} = 1$$

  This correctly maps to $Flex\_Value = 1$ (fully flexed).

- When $ADC_{value} = ADC_{min}$, the equation simplifies to:

  $$Flex\_Value = \frac{ADC_{min} - ADC_{min}}{ADC_{max} - ADC_{min}} = 0$$

  This correctly maps to $Flex\_Value = 0$ (fully contracted).

2.13 Tolerance Analysis for the POTs

Potentiometers have mechanical and electrical tolerances that affect precision. The key parameters include:

Parameter	**Nominal Value**	**Tolerance**	**Possible Variation**
Resistance Value	RPOT	±10%	9kΩ – 11kΩ
Supply Voltage (ESP32)	3.3V	±9%	3.1V – 3.6V
POT Wiper Linearity	Ideal Linear	±5%	Slight non-linearity

Table 4: Tolerance Analysis of POTs

3 Cost and Schedule

Week	Category	Task	Person
March 10 – March 14	Firmware	Boot, Get serial working	Aditya
March 10 – March 14	Glove hardware	Bringup PCB v1	Hamza
March 10 – March 14	Software	VRHands pot data registration	Ashton
March 10 – March 14	Firmware	Send joystick+button data	Aditya
March 10 – March 14	Glove hardware	Finish breadboard glove mount	Ashton
March 10 – March 14	Software	IAM joystick/button with registration	Ashton
March 17 – March 21	Spring break	–	–
March 24 – March 28	Firmware	Send pot data	Aditya
March 24 – March 28	Glove hardware	Finalize haptic system modules	Ashton
March 24 – March 28	Software	Work out collider code	Ashton
March 31 – April 4	Firmware	Receiving Unity collider data	Aditya
March 31 – April 4	Glove hardware	Finalize PCB v2 design	Hamza
March 31 – April 4	Software	Finish making test scene	Ashton
April 7 – April 11	Firmware	Get Bluetooth working	Aditya
April 7 – April 11	Glove hardware	Bringup PCB v2	Hamza
April 7 – April 11	Software	ESP32 integration	Ashton
April 14 – April 18	Firmware	Optimize Bluetooth speed	Aditya
April 14 – April 18	Glove hardware	Finalize PCB final design	Hamza
April 14 – April 18	Software	ESP32 integration	Ashton

4 Ethics and Safety

In our project, user privacy is one of the main ethical issues. Strict data protection protocols must be put in place because VR gloves depend on motion tracking, haptic feedback, and possibly biometric data. This is consistent with Section I.1 of IEEE's Code of Ethics [IEE24], which highlights the importance of putting the public's welfare and privacy first. We are dedicated to creating a system that only collects the bare minimum of data, encrypts sensitive data, and gives consumers complete transparency regarding the usage of their data.

Accessibility is another important ethical factor in addition to privacy. Virtual reality experiences ought to be accessible, but a lot of current technology isn't made for people with different hand sizes or limitations. The goal of our glove's design is to fit and operate differently for a variety of users. We intend to document our design approach and usability testing in accordance with IEEE's Code of Ethics Section I.2 [IEE24], which promotes enhancing public knowledge of technology, in order to add to the larger discussion on accessibility in virtual reality.

Another key issue with wearable electronics is safety. Poor design decisions may result in electrical risks, discomfort, or repetitive strain injuries. Our glove uses ergonomic design concepts to reduce these dangers and make sure that extended use doesn't cause muscular strain or fatigue. Furthermore, haptic feedback components will undergo rigorous testing to avoid using too much force, which could result in pain or harm. Section I.5 of IEEE's Code of Ethics [IEE24] exhorts engineers to be truthful when describing the strengths and weaknesses of their designs. Accordingly, we will give precise instructions on how to wear our glove safely and, rather than overpromising performance, recognize any possible hazards.

5 Citations

References

[IEE24] IEEE. *IEEE Code of Ethics*. Accessed: 2025-03-06. 2024. URL: `https://www.ieee.org/about/corporate/governance/p7-8.html`.