

Design Document

Team 30

Ankur Prasad, Matthew Uthayopas, Tunc Gozubuyuk

TA: Mingrui Liu

Spring 2026

ECE 445

Contents

1. Introduction.....	3
1.1 Problem.....	3
1.2 Solution.....	3
1.3 Visual Aid.....	3
1.4 High-Level Requirements.....	5
2 Design.....	3
2.1 Physical Design.....	6
2.2 Block Diagram.....	6
2.3 Robotic Hand System.....	6
2.3.1 Hand Interface Subsystem.....	7
2.3.2 Hand Control Subsystem.....	7
2.3.3 Power Subsystem.....	8
2.4 Glove System.....	8
2.4.1 Sensing Detection Subsystem.....	9
2.4.2 Sensing Control Subsystem.....	9
2.4.3 Power Subsystem.....	10
2.5 Tolerance Analysis.....	10
3. Cost and Schedule	
4. Ethics and Safety	
5. Citations	

1 Introduction

American Sign Language (ASL) serves as an important mode of communication for the Deaf and hard-of-hearing community. In ASL, proficiency is judged not on vocabulary, but on the execution of hand poses and gestures. As such, effective instruction requires a clear visual demonstration and, more importantly, immediate feedback so that learners build gestures with severe inaccuracies. As it stands, the current educational methods fall short of providing the nuanced, 3-D spatial feedback that is essential for mastering ASL. This feedback gap is acutely felt in learning environments such as high school classrooms, where a single instructor is responsible for making corrections and guiding numerous students. The resulting inefficiency can lead to student frustration and poor retention that translates into a systemic barrier to inclusive ASL education.

1.1 Problem

In today's time, the methods for learning American Sign Language (ASL) in structured educational settings fail to really provide the immediate, spatial feedback that is needed for accurate gesture development. Students primarily rely on classroom observation, supplemental videos, and websites, which lack the 3D perspective necessary to truly replicate the hand positions and unique movements. Direct correction is the most common feedback type being used in 60% of error responses. While the student repeats these corrections more than 90% of the time, this approach places the full burden of feedback and accurate demonstration on the instructor [1]. The lack of individualized feedback leads to developing incorrect signs, poor retention, and an increasing feeling of frustration. Recent advancements in machine learning for ASL recognition are promising, yet significant challenges remain across varied conditions for practical educational applications [2].

This feedback gap is widened in high school classrooms where a single teacher must divide their attention among 20-30 students with varying proficiency levels, truly making individualized correction strenuous and even impossible during practice sessions. As such, these teachers are faced with the challenge of remediating ingrained errors in students rather than helping them proactively guide progress, which reduces the effectiveness of the ASL course. This inefficiency represents a significant resource constraint in the form of a teacher's time in the

classroom. With over 100,000 students receiving services for hearing disabilities in the United States public schools, the systemic inability to scale good-quality ASL instruction that is individualized is a large barrier to inclusion. The problem here is an absence of tools that deliver the essential feedback required to obtain mastery eventually, which ultimately puts strain on students and teachers.

1.2 Solution

We are creating a tendon-driven bionic hand that is capable of performing all of the letters of the alphabet in the American Sign Language, designed as an interactive and cost-efficient tool. At a high level, the system consists of two components: a robotic hand and a glove sensor. The robotic hand will contain five movable fingers that will change positions to achieve the ASL letter positions. The sensor glove will be worn by a proficient user and capture their finger movements, and wirelessly transmit that data to the actual robotic hand. The robot hand will then assume the same pose, which allows the hand to serve as a physical three-dimensional demonstration tool that students can observe from any angle while practicing their own signs or the robot hand will cycle through letters on its own so the user can follow along and mimic the robot arm's movements. This system is designed to remove the burden of a constant one-to-one feedback loop from the teacher to the student and will enable self-guided practice with a physical tool to give immediate physical feedback.

1.3 Visual Aid

This system is designed around two modes of operation. One mode will be where the robotic hand learns gestures from an instructor, and a learning mode where the student observes and mimics the gestures done by the robotic hand.

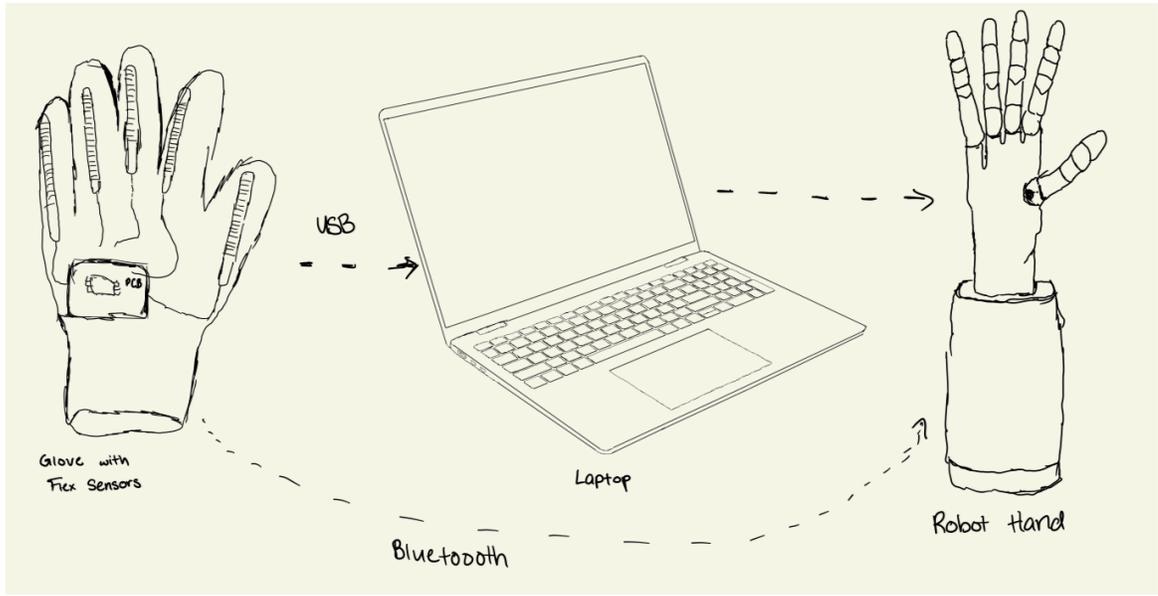


Figure 1: Training Mode Operation

Figure 1 shows the training phase of the system. An instructor or a person who knows sign language will wear the glove and perform the alphabet. The data from the flex sensor glove will be transmitted to the laptop and the robotic hand. Here, the computer will keep recording the flex sensor data. Once the gesture is finished and the sign is held, the robotic hand will move its fingers and assume the identical pose. The laptop will also read the PWM signals sent to the motor and keep a log of the respective PWM signals for each letter, which the robotic hand will then upload to the robotic hand for later.

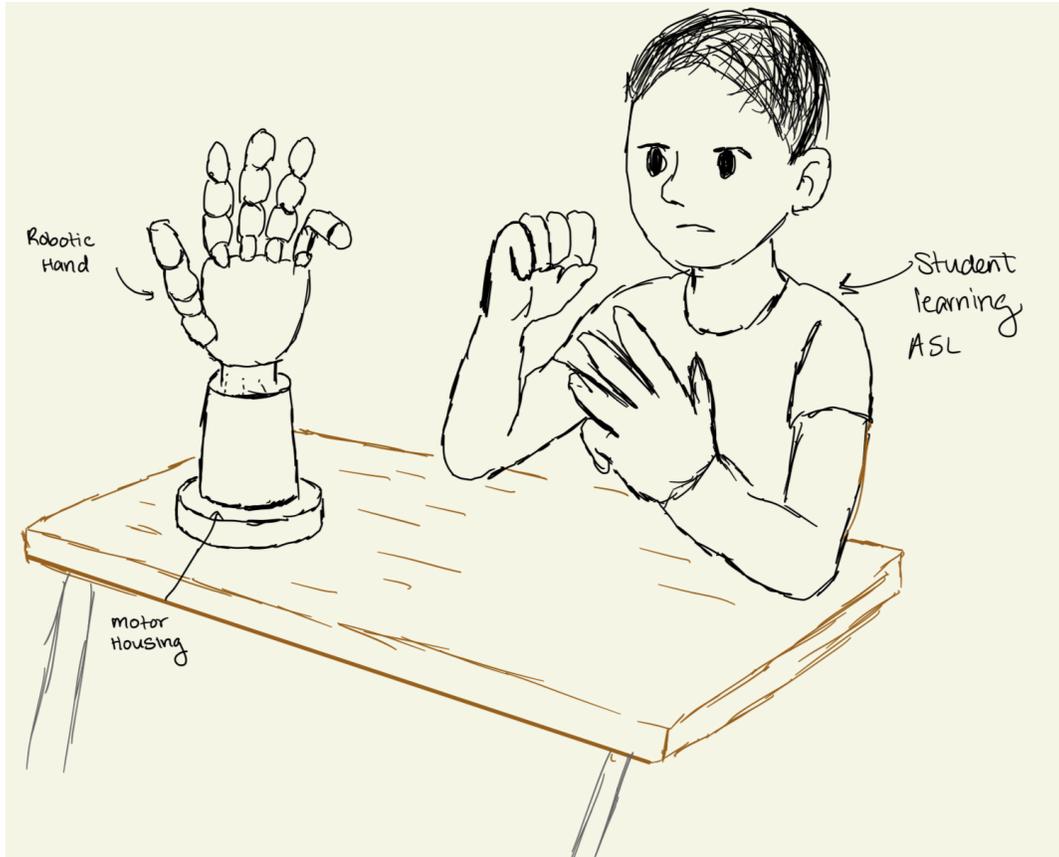


Figure 2: Learning mode in Classroom Context.

Figure 2 places the robotic hand in the intended classroom learning environment. The robotic hand will be turned on and placed in recite mode, where the robotic hand will cycle through letters. Here, a student will observe the robotic hand from any angle while attempting to replicate the ASL letter. The motor housing and the electronics will be placed in the arm with an enclosure to keep the students' focus on the movement of the fingers. The setup allows the student to receive consistent visual feedback without constantly requiring instructor support.

1.4 High-Level Requirements

We understand that being able to perform complete sign language can be very challenging due to physical limitations. Thus, we will be focusing on the technical aspects of the project and aim to perform basic movements with the robotic arm. To consider our project successful, our robotic arm should fulfil the following:

1. The robotic hand should sign 7 simple letters of American Sign Language, which do not require wrist movement and minor thumb movement (e.g. A, B, D, F, U, W).
2. Any words or letters signalled should be able to be recognized by at least 3 testers (The testers will observe the robot hand and write down what was signed
3. The device should be able to sign letters in succession 6 times
4. The robotic hand must be able to replicate the signs that were performed from the glove at 85% accuracy (85% of the time it should give the right sign)
 - a. A correct sign is one that matches an image of the specific letter performed by someone well-versed in sign language (can be online)

2 Design

2.1 Physical Design

The actual robotic hand will be 3D printed with a Bambu 3D printer. Specifically, we will print the 4 fingers and thumb, including each of the joints, the palm, and the back side of the hand. Additionally, we will print the arm that will go below the hand. All of these parts will be connected via gorilla glue, 3d printed hardware, and the traditional hardware that includes screws, nuts, washer and bolts.

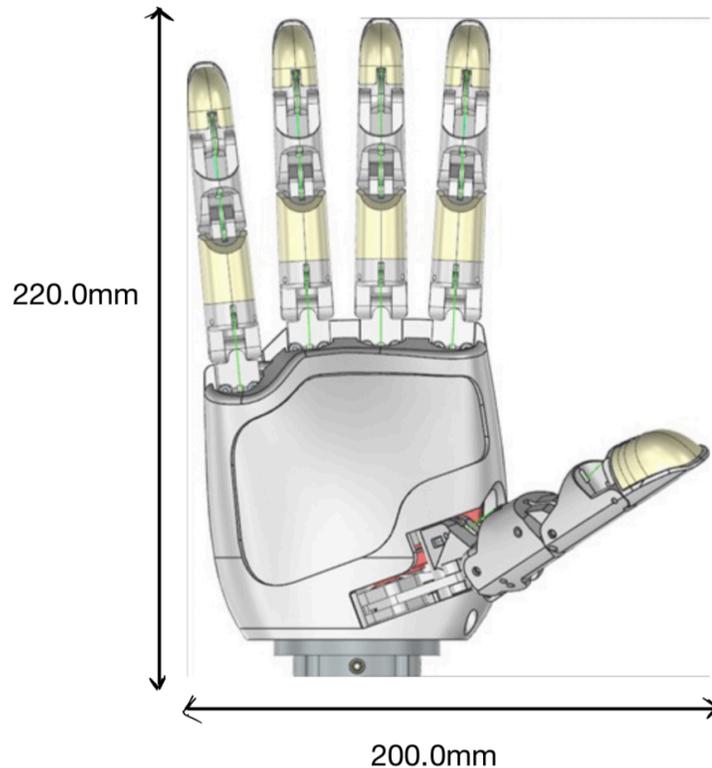


Figure 3: Model of Robotic Hand

The physical hand will be about 22cm long and 20cm wide, as shown in Figure 3. This 3D model has been taken from an open-source model made by Tether IA and serves as the inspiration for our design [3]. Each of the four fingers will contain 3 joints, which will each be able to rotate about 90 degrees. They will have a nylon/Kevlar string that will run through the entire finger, which will go on to a spool that will be mounted on a servo motor. The purpose of this string is to flex the fingers when the servo motor spins. Each of the 4 fingers will have one servo motor because we felt that it is sufficient for the fingers to just bend forward, and the 3 joint design will make the finger bend naturally. On each finger as well, we will place an elastic cord or a spring on the back to act as the restoring force to bring the fingers back to being straight. The middle knuckle joint (formally known as the DIP joint) and the base knuckle (formally known as the MCP joint) will have the elastic cord or spring because having the restoring force on each joint will make the motion of the finger flexing and restoring look natural.

Moving on to the thumb, which is a difficult part of the design, it will have three servo motors attached to it. This is because the thumb flexes forward and backward like any other

finger, but it also can move inwards to touch other fingers (opposition), and it can move towards the index finger while keeping the actual finger straight (adduction). This means that there are three types of movements that the thumb must accomplish and thus needs 1 servo motor per type of movement. We will mount the 4 servos that move the 4 fingers in the arm, and then run the nylon/Kevlar string through the hand. As far as the thumb is concerned, we have the servo motors on the actual hand itself because placing the servos on the joints ensures that all of the movements of the thumb can happen without a servo preventing the thumb from moving in the desired direction. Moving the joints will also require less torque if the motors are located in the palm. However, the design might change, and we might place all of the servos in the palm. The thumb will also have an elastic cord or springs placed on the joint at the base of the thumb and the middle of the thumb to act as the restoring force, similarly to the other fingers.

The custom-designed PCB of the robotic hand will also be put in the arm of the robotic hand, where the servos will then be directly connected to the PCB. The PCB will be attached to the arm, and an opening in the arm will be left to be able to access the PCB. The robot arm will be supplied power through plugging it into the wall, and so a traditional adapter will connect the wall and the pcb of the robot arm, allowing the microcontroller and the servo motors to be powered and be able to perform their respective functions.

The other physical component of the project is the training glove that will contain flex sensors. This glove will be a standard winter glove that you can find at any store or buy online, and will have flex sensors stitched onto the glove on the back of each finger and an additional one placed on the base of the thumb. This is shown below in Figure 4.

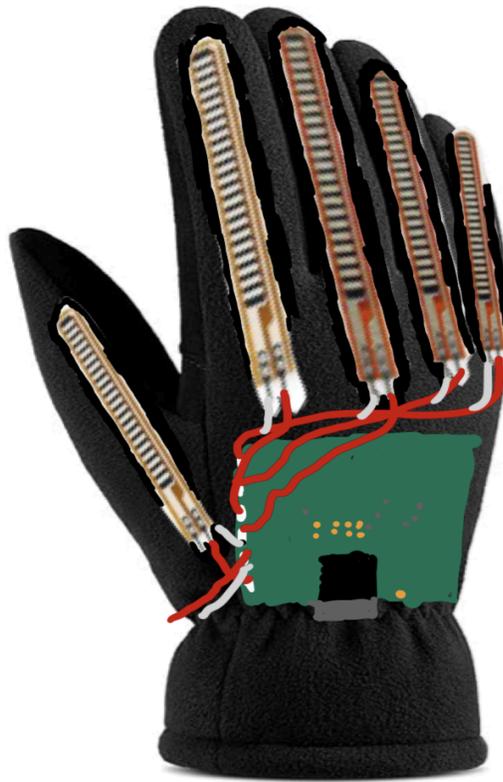


Figure 4: Sensor Glove Design

The glove will also have a custom-designed PCB that will be mounted on the back of the glove. We have decided to place it in the middle of the glove on the back because that provides enough area to keep a rectangular shaped pcb flat, as well as being a central location to plug in all of the flex sensors. The flex sensors will connect to the pcb through a normal 2-pin connector. To do that, we will have to solder a wire on each terminal of the flex sensor and then place the other end of that wire into a female connector clip. Additionally, the glove sensor will be battery-powered with a standard 9V battery connected to a power adaptor on the board. When the fingers bend, the resistances of the flex sensors will change, which will tell the microcontroller how far or how many degrees the finger has moved.

2.2 Block Diagram

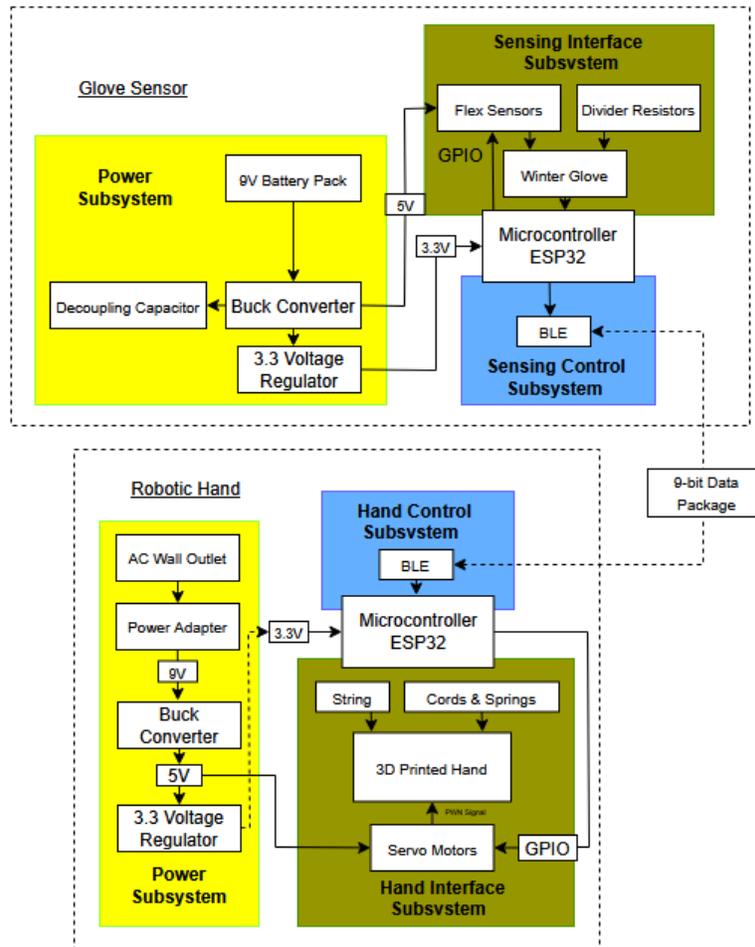


Figure 5: ASL Robotic Hand Block Diagram

Our system design consists of two core systems, each containing three subsystems. We decided to make the subsystems have the same roles in each system for ease of understanding. These subsystems can be seen as power, interface, and control. Power will focus on making sure that every component in our design receives the voltage and current needed. The interface subsystems will focus on performing the system's specified task. The control subsystems will focus on linking the two main systems together.

2.3 Robotic Hand System

... This system is where the main execution happens in our project. This system will translate data received from the glove sensor system in order to perform the sign language tasks required.

2.3.1 Hand Interface Subsystem

.. The Hand Interface Subsystem contains seven servo motors and an ESP32 microcontroller. The ESP32 microcontroller is powered by 3.3V from the power subsystem and the servo motors, which operate within a voltage of between 4.4V to 6V, are supplied with 5V. The motors are connected to the microcontroller through input/output pins, which will determine when and how much each motor rotates. This determination is through PWM control which is handled in the sensing control and hand control subsystems. Essentially, a pulse will be sent from the ESP32 microcontroller through GPIO which will tell the servo motor to turn on.

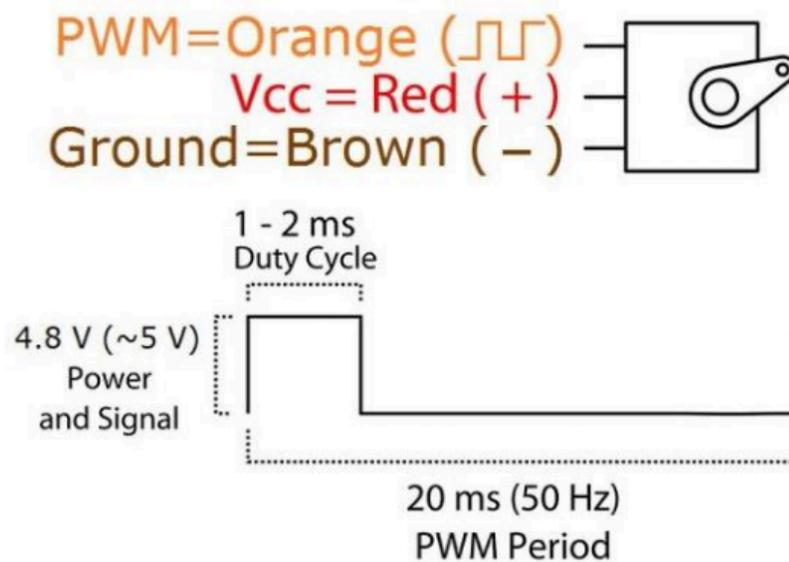


Figure 6: Control Signal of Servo Motor

Our robotic hand is designed to rest in a normal open position. Each servo motor will be responsible for closing a specific joint of the hand by pulling on a nylon string, which acts as a tendon, attached to that joint. Each finger of the hand will require one servo motor, while the thumb contains three and will require three servo motors. When a servo rotates, it pulls the string to flex the corresponding joint and close the finger. In order to bring the joint back to the original position, restoring forces provided by springs will extend the joints when the servo releases tension.

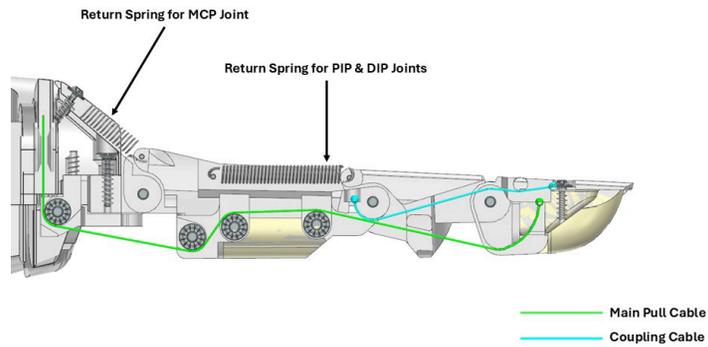


Figure 7: Potential implementation of string & spring in finger

Table 1: Hand Control Mechanical Subsystem - Requirements & Verification

Requirements	Verification
The tendon system must be placed with minimal slack to allow the servo motors to rotate less than 10 degrees before causing any finger movement	We will be programming the servo motor to move in small increments and observing the angle of the finger's movement. This requirement will fail if the finger does not move with less than 10 degrees of servo motor rotation.
PWM signal must be 50 Hz \pm 1 Hz	We will use debugging LED's to probe the PWM signal. Then we will command the servo motor and record the time it takes for the system to respond.
The restoring force must generate sufficient force to fully extend each finger against gravity when the servo releases the tension	We will hold each motor at its most closed position for at least 3 seconds. If the springs are able to bring the hand back to its open

	position, then the requirement is satisfied.
Servo motor must begin motion within a reasonable amount of time	If the servo is able to start running within 200ms of command, then this design requirement is satisfied.
The servos must be able to hold the fingers at their most closed position. The finger must be able to rotate almost $180^\circ \pm 10^\circ$	We will be measuring the angle from where the finger begins and where it ends up after running the servo motor until the finger stops moving due to lack of string (until the motor is stalled).

2.3.2 Hand Control Subsystem

The Hand Control System is where we will be able to send and receive signals from the hand and glove systems. This signal will be coming from the sensing interface subsystem. Specifically, the flex sensors will provide a 9-bit resistance value in the form of PWM which can be used to control the movement of each finger in the hand. This 9-bit data package will be transmitted through Bluetooth Low Energy, which is built into the ESP32 microcontrollers. Once we receive the signal, the ESP32 in the robotic hand system will execute its code accordingly. Note that we have placed testing LED's into our microcontrollers so we will be able to test our requirements for sending signals using them.

Table 2: Hand Control Interface Subsystem - Requirements & Verification

Requirements	Verification
The ESP32 must maintain a stable BLE connection, receiving sensor data at a minimum rate of 10 Hz.	If we are able to receive at least 600 data packets in a 60 second span, then the rate of 10 Hz will be satisfied. This can be done by keeping a timestamp on the arduino IDE and

	adding to a count of each data package received.
The control firmware must translate received sensor integers into servo PWM commands with a total latency of less than 100 ms from packet receipt to PWM update.	We will program an LED to light up on each microcontroller. If the LED turns on within 100 ms apart, then the requirement will be satisfied.
The BLE signal must transmit data packages reliably	We will be transmitting a known amount of data packages (around 500) and counting the amount of data packages actually received. If the number of successful transmissions has an error rate less than 1% then this requirement is satisfied.

2.3.3 Power Subsystem

The power subsystem will handle all the power consumption for the robotic hand. Initially, we will be providing power to our systems through a wall outlet through a 9V power adapter. This power adapter will be then connected to a buck converter to convert the 9V of power to 5V, which will be used for the servo motors. The reason as to why we use a buck converter here instead of a regular voltage regulator is that each servo motor requires a stall current of 2.5A. Voltage regulators provided by the self-service shop only go up to a current flow of 2A while the buck converter passes 3A of current through. Additionally, the buck converter will create less heat than the voltage regulator.

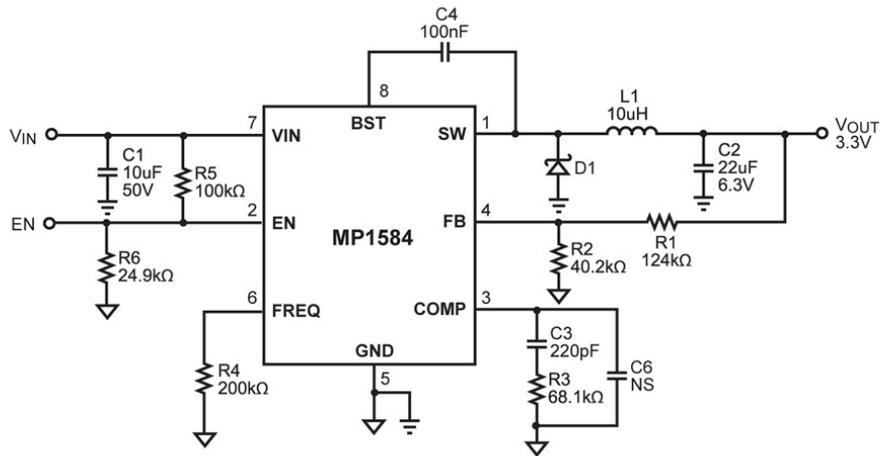


Figure 9: Schematic of buck converter

The ESP32 microcontroller requires significantly less current flow than the servo motors and operates at 3.3V. Thus, we will use a 3.3V regulator to step down the voltage here.

Symbol	Parameter	Min	Typ	Max	Unit
VDD33	Power supply voltage	3.0	3.3	3.6	V
I_{VDD}	Current delivered by external power supply	0.5	—	—	A
T_A	Operating ambient temperature	-40	—	65	°C
				85	
				105	

Figure 10: Recommended Operating Conditions for ESP32 MCU

From the table, we can see that we want our voltage range to be in the area of 3.0 to 3.6V. This will be the values which we will be aiming for in our requirements.

Table 3: Hand Control Power Subsystem - Requirements & Verification

Requirements	Verification
The Motor Power Rail must maintain a voltage of above 4.75 when all the servos are	We will be using the multimeters located in the lab room at ECE 2072. If the measured

connected to the power supply to prevent the servos from stalling	voltage is within the threshold then the requirement is met. All motors will be turned on simultaneously and voltage will be checked during this process
The 3.3V Voltage Regulator must provide a stable power supply of 3.0-3.6V at 500 mA to ensure the microcontroller is sufficiently powered.	We will be using the multimeters located in the lab room at ECE 2072. If the measured voltage and current is within the threshold then the requirement is met.
Buck converter must supply 5.0 V \pm 5% and up to 2.5A of continuous current	We will be using the multimeters located in the lab room at ECE 2072. If the measured voltage and current is within the threshold then the requirement is met.

2.4 Glove System

This system is the core sensing and training unit of our project. By capturing the user’s finger and joint movements through 6 flex sensors mounted on a wearable glove and translating them into electrical signals, this unit allows the user to train our robotic hand to mimic their hand movements. Our Glove System is divided into three subsystems. The Sensing Detection Subsystem, the Sensing Control Subsystem, and the Power Subsystem manage user hand movement data collection and conversion of this data into usable signals, transmission of the glove motion data, and regulated power distribution, respectively.

2.4.1 Sensing Detection Subsystem

Our Sensing Detection Subsystem is mainly responsible for converting the user’s finger motion into electrical signals which can represent the user’s finger bend in real-time. The flex sensor behaves as a variable resistor whose resistance changes with bending angle. In our Glove Sensor PCB, each flex sensor forms a resistive divider with a fixed 10 k Ω reference resistor and these

resistors are R10 through R14 and R17. The resulting sensor output voltage is limited to the ESP32's ADC safe range since the divider is driven from the regulated 3.3 V rail. On the Glove Sensor PCB, each flex sensor is interfaced through a dedicated 2-pin connector, and these 2-pin connectors are J2 through J7.

The divider output for each sensor is routed to a dedicated ESP32 GPIO configured for analog input (GPIO1, GPIO2, GPIO4, GPIO5, GPIO6, GPIO7). The sensor resistance changes as the user bends his/her finger. Then the divider voltage shifts accordingly. By this way, a continuous analog representation of finger position is obtained which can then be sampled and converted to digital values by the ESP32 ADC.

Table 4: Sensing Detection Subsystem - Requirements & Verification

Requirements	Verification
Each of the flex sensors should be generating an analog voltage that remains within the ESP32 ADC input range which is 0 to 3.3 V.	While the flex sensors are straight and fully bent, we are going to measure the output voltage at each sensor ADC node and verify that these voltages are within the 0 to 3.3 V range.
As the bending angle increases, each of the flex sensors should be producing a monotonic change.	While checking the ADC values, we are going to start bending the flex sensors from open to closed and verify that the readings we get either decrease continuously or increase continuously.

2.4.2 Sensing Control Subsystem

To convert sensor voltages into usable control information for the robotic hand, The Sensing Control Subsystem performs the acquisition, processing, and system-level management that is required.

To ensure stable startup behavior and to reduce susceptibility to glitches, the EN line uses a pull-up and RC network (R1 and C5). BOOT is connected to GPIO0 and conditioned similarly to the EN line using R3 and C11. For manual reset and boot-mode entry (SW1 for EN/reset and SW2 for BOOT) pushbuttons are provided. To provide immediate visual feedback when the BOOT or EN buttons are pressed (D10,R20 and D11,R21), LED branches are also included. Additional debugging LEDs (D4–D9 with resistors R6–R9, R18, R19) are connected to selected GPIO pins (GPIO10, GPIO11, GPIO12, GPIO13, GPIO14, GPIO15) to support quick verification of firmware state and timing.

To enable programming, the subsystem also interfaces to the USB-C connector. D- and D+ lines are routed to the ESP32 USB_D- and USB_D+, respectively.

Table 5: Sensing Control Subsystem - Requirements & Verification

Requirements	Verification
EN should correctly reset the ESP32-S3-WROOM-1 on every press.	We are going to press the EN button for 30 times and verify that a reboot happens after each press and release (Each press and release combined does the EN action).
The LEDs used for debug should correctly reflect the timing and output states.	We are going to load a GPIO program and with known timing, the program will toggle each LED. While observing the correct states with correct timing, we are going to verify whether the LEDs are working in the desired way.

During reset, ESP32-S3-WROOM-1 should be put into programming mode when BOOT is pressed and released.	We are going to press and release the BOOT button and press and release the EN button and confirm whether the program uploads are successful.
---	---

2.4.3 Power Subsystem

The power subsystem will handle all the power consumption for the robotic hand. Initially, we will be providing power to our systems through a 9V battery or through a USB-C connection (5V). To choose between them, two Schottky diodes are used (selects the voltage). The input voltage is first passed through a switching buck converter that steps the voltage down from 9 V to a regulated 5 V when operating from the 9V battery. Instead of a linear voltage regulator, a switching regulator is used here since the system should support high current loads. While using the buck converter, a higher efficiency can be obtained when compared to using a linear regulator. In order to get 3.3 V out of the 5 V power net, a linear voltage regulator is used to step the 5 V rail down to 3.3 V. Then, the flex sensors, voltage-divider circuits, ESP32-S3-WROOM-1 microcontroller, and debugging circuits are powered from the 3.3 V rail. To reduce power supply noise, the decoupling capacitors are placed near the microcontroller and regulators.

Table 6: Glove System Power Subsystem - Requirements & Verification

Requirements	Verification
Either the 9V battery or the USB-C should be accepted as the input power.	We are going to connect each of the power sources alone and verify whether the board turns on.

The buck converter should be able to regulate 9V to 5.0 V $\pm 5\%$.	Under load, we are going to apply 9V and verify whether the buck converter can give 5.0 V $\pm 5\%$.
---	---

2.5 Tolerance Analysis

2.5.1 Buck Converter

We are using the MP1584 Buck Converter, which is a crucial subsystem component as it is responsible for supplying 5Volts of power to all of the servo motors. The MP1584 is a high-frequency step-down switching regulator with an integrated internal high-side high-voltage power MOSFET. It provides 3A output with current mode control for fast loop response and easy compensation. If the output voltage falls outside the servo motor's operating range of 4.8-5.5V, the motors may stall or behave erratically, causing the hand to fail to hold a position and thus be unable to perform accurate ASL gestures. Due to this, we must verify that our design maintains the output voltage within this range taking real-world tolerances into consideration.

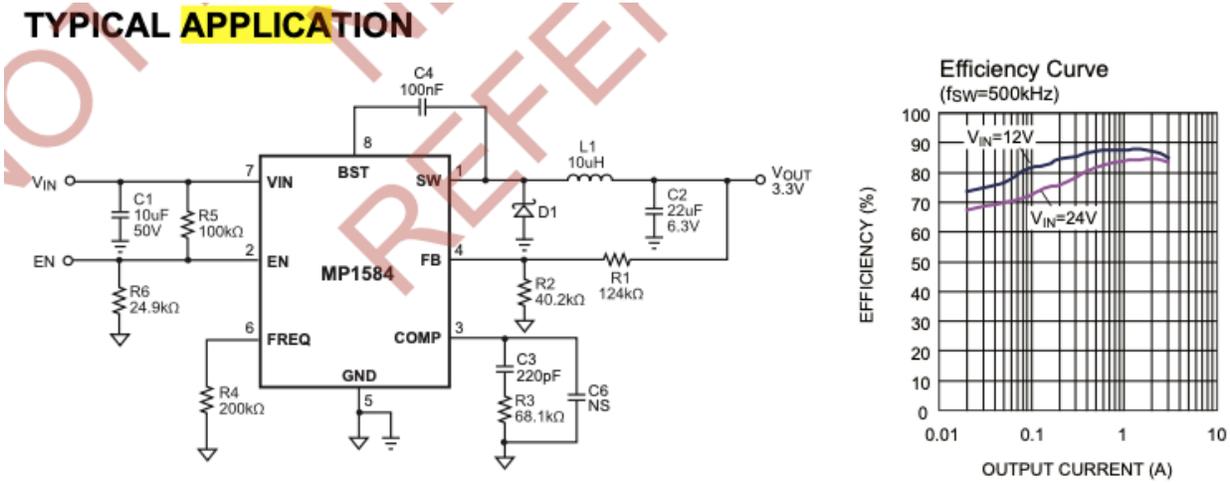


Figure 11: MP1584 Buck Converter Typical Application

Table 7: MP1584 Buck Converter Electrical Characteristics

ELECTRICAL CHARACTERISTICS

$V_{IN} = 12V$, $V_{EN} = 2.5V$, $V_{COMP} = 1.4V$, $T_A = +25^\circ C$, unless otherwise noted.

Parameter	Symbol	Condition	Min	Typ	Max	Units
Feedback Voltage	V_{FB}	$4.5V < V_{IN} < 28V$	0.776	0.8	0.824	V
Upper Switch On Resistance	$R_{DS(ON)}$	$V_{BST} - V_{SW} = 5V$		150		m Ω
Upper Switch Leakage		$V_{EN} = 0V$, $V_{SW} = 0V$, $V_{IN} = 28V$		1		μA
Current Limit			4.0	4.7		A
COMP to Current Sense Transconductance	G_{CS}			9		A/V
Error Amp Voltage Gain ⁽⁵⁾				200		V/V
Error Amp Transconductance		$I_{COMP} = \pm 3\mu A$	40	60	80	$\mu A/V$
Error Amp Min Source current		$V_{FB} = 0.7V$		5		μA
Error Amp Min Sink current		$V_{FB} = 0.9V$		-5		μA
VIN UVLO Threshold			2.7	3.0	3.3	V
VIN UVLO Hysteresis				0.35		V
Soft-Start Time ⁽⁵⁾		$0V < V_{FB} < 0.8V$		1.5		ms
Oscillator Frequency		$R_{FREQ} = 100k\Omega$		900		kHz
Shutdown Supply Current		$V_{EN} = 0V$		12	20	μA
Quiescent Supply Current		No load, $V_{FB} = 0.9V$		100	125	μA
Thermal Shutdown				150		$^\circ C$
Thermal Shutdown Hysteresis				15		$^\circ C$
Minimum Off Time ⁽⁵⁾				100		ns
Minimum On Time ^(b)				100		ns
EN Up Threshold			1.35	1.5	1.65	V
EN Hysteresis				300		mV

Note:

5) Guaranteed by design.

The MP1584 output voltage depends primarily on the feedback divider shown in Figure 11.

FB Divider:

The output setpoint is $V_{OUT} = V_{REF}(1 + R_{TOP}/R_{BOT})$, $V_{REF} \approx 0.8 V$

where $R_{TOP} = R_{28} = 220k\Omega$ (the resistor from V_{OUT} to FB) and $R_{BOT} = R_{27} = 40k\Omega$ (the resistor from FB to GND)

$$V_{OUT} \approx 0.8 \cdot (1 + 220/40) = 5.2 V$$

This value of 5.2V falls within the servo operating range of 4.8V-5.5V, providing a 0.3V margin on the higher end. However, resistors have tolerances due to the way that they are manufactured. Using 1% tolerance resistors for both R_{TOP} and R_{BOT} , we must analyze the worst case scenario with the output voltage. The maximum output voltage occurs when R_{TOP} is at its maximum value, and R_{BOT} is at its minimum value. Thus:

$$R_{TOP, maximum} = 220k\Omega \cdot 1.01 = 222.2k\Omega$$

$$R_{BOT, minimum} = 40k\Omega \cdot 0.99 = 39.6k\Omega$$

$$V_{OUT} \approx 0.8 \cdot (1 + 222.2/39.6) = 5.289V$$

The minimum output voltage occurs when R_{TOP} is at its minimum value, and R_{BOT} is at its maximum value. Thus:

$$R_{TOP, minimum} = 220k\Omega \cdot 0.99 = 217.8k\Omega$$

$$R_{BOT, maximum} = 40k\Omega \cdot 1.01 = 40.4k\Omega$$

$$V_{OUT} \approx 0.8 \cdot (1 + 217.8/40.4) = 5.113V$$

In the worst case scenarios $V_{OUT} = 5.2 \pm 0.089V$. So even in the worst case tolerance scenario, the output voltage remains between 5.113V and 5.289V, which is well within the servo operating range of 4.8V–5.5V. This gives us confidence that all seven servos will receive adequate power regardless of resistor variations.

Another key part of the MP1584 chip's operation is the divider on the enable pin, which sets the input voltage to the system. The EN pin utilizes a resistor divider to set the undervoltage lockout threshold.

EN Divider:

$$V_{EN} = V_{IN} \cdot (R_{EN, BOT} / (R_{EN, TOP} + R_{EN, BOT}))$$

$$V_{EN} = V_{IN} \cdot (25 / (100 + 25)) = 0.2 V_{IN}$$

For many buck converters, the EN threshold is around 1.2 V, turn on voltage: $V_{IN} = 1.2/0.2 = 6 \text{ V}$. The MP1584 typical EN threshold is 1.35V minimum to 1.65V maximum. Solving for the input voltage at which the converter enables:

$$V_{IN, \text{enable, min}} = 1.35\text{V}/0.2 \cdot = 6.75\text{V}$$

$$V_{IN, \text{enable, max}} = 1.65\text{V}/0.2 \cdot = 8.25\text{V}$$

With a 9V battery, the converter will start reliably. As the battery discharges, the converter will shut down between 6.75V and 8.25V, preventing any unwanted operation or the burnout of the ESP32 microcontroller when the battery does get depleted. This aligns with the ESP32-S3 operating voltage requirement of between 3.0V to 3.6V after voltage regulation.

ABSOLUTE MAXIMUM RATINGS ⁽¹⁾

Supply Voltage (V_{IN}).....	-0.3V to +30V
Switch Voltage (V_{SW}).....	-0.3V to $V_{IN} + 0.3\text{V}$
BST to SW	-0.3V to +6V
All Other Pins	-0.3V to +6V
Continuous Power Dissipation ($T_A =$ +25°C) ⁽²⁾	2.5W
Junction Temperature	150°C
Lead Temperature	260°C
Storage Temperature.....	-65°C to +150°C

Figure 12: MP1584 Buck Converter Absolute Maximum Ratings

Recommended Operating Conditions ⁽³⁾

Supply Voltage V_{IN}	4.5V to 28V
Output Voltage V_{OUT}	0.8V to 25V

Figure 13: MP1584 Buck Converter Recommended Operating Conditions

According to Figures 12 and 13, a 5 V output is within the Output Voltage range, and a 9 V input is within the Supply Voltage range.

3 Cost and Schedule

3.1 Cost Analysis

This project aims to be completed within the \$150 budget provided by the university. We will be 3D printing our robotic hand instead, so there are no machine shop labor hours. Below is a table covering the total expenses needed to complete the robotic hand:

Description	Manufacturer	Quantity	Extended Price	Link
ESP 32 Microcontroller	ESP32-S3-WROOM	2	\$10.98	Link
3.3V Voltage Regulator	National Semiconductor (TI)	2	\$1.46	Link
Tower Pro MG996 (Servo Motor)	Tower Pro	7	\$33.80	Link
Flex Sensor 2.2"	Sparkfun	6	\$34.89	Link
DC Step-Down Converter	Monolithic Power Systems Inc.	2	\$5.50	Link
USB Type C Receptacle for USB2.0	GCT	2	\$1.56	Link
CONN PWR JACK 1.35X3.5MM SOLDER	Same Sky	1	\$0.67	Link
Schottky Diode	Vishay General Semiconductor	8	\$5.60	Link
CONN HEADER VERT 2POS 2.54MM	Molex	6	\$0.96	Link
Surface Mount LED's	Lite-On Inc	19	\$2.09	Link
2 Position Wire to Board Terminal Block Horizontal with Board 0.200" (5.08mm) Through Hole	Phoenix Contact	1	\$0.92	Link
CONN HEADER VERT 3POS 2.54MM	Würth Elektronik	7	\$0.91	Link
10uH Inductor	Coilcraft	2	\$0.90	Link
SWITCH TACTILE SPST-	TE Connectivity ALCOSWITCH Switches	4	\$0.52	Link
Slide Switches	C&K Switches	1	\$0.72	Link
1uF 0805 Capacitor	YAGEO	4	\$0.92	Link
10uF 0805 Ceramic	YAGEO	6	\$1.86	Link

Capacitor				
10uF Tantalum Capacitor	Multicorp	2	\$1.12	Link
22uF 1206 Ceramic Capacitor	YAGEO	4	\$2.76	Link
0.1uF 0805 Ceramic Capacitor	YAGEO	6	\$0.60	Link
220pF Monolithic Capacitor	Murata Electronics	2	\$1.06	Link
10K Ohm 0805 Resistor	Vishay	10	\$1.50	Link
1K Ohm 0805 Resistor	Vishay	18	\$3.06	Link
5.1K Ohm 0805 Resistor	Vishay	4	\$0.40	Link
100K Ohm 0805 Resistor	Vishay	2	\$0.20	Link
100 Ohm Axial Resistor	YAGEO	2	\$0.20	Link
4.7k Ohm Axial Resistor	Stackpole Electronics Inc	2	\$0.20	Link
24k Ohm Axial Resistor	Stackpole Electronics Inc	2	\$0.24	Link
40k Ohm Axial Resistor	YAGEO	2	\$0.36	Link
68k Ohm Axial Resistor	Stackpole Electronics Inc	2	\$0.20	Link
200k Ohm Axial Resistor	Stackpole Electronics Inc	2	\$0.20	Link
220k Ohm Axial Resistor	Stackpole Electronics Inc	2	\$0.20	Link
9V Power Adaptor	AVLIS-CO	1	\$4.99	Link
M-35CS - 0.50 Inch Music Wire Extension Springs	Century Springs	4	\$13.08	Link
Elastic Coord (4 Yards pack)	Dritz	1	\$1.62	Link
Kevlar Cord (100lb 0.8mm(Dia.)x100ft)	Everbilt	1	\$6.35	Link
Glue for 3D printed components	Gorilla Glue	1	\$4.99	Link

**highlighted components are ones which are taken from the E-Shop student self-service and therefore do not count toward the university-provided budget.*

Summing up all of the components provided gives a total cost of \$147.59, excluding the 3D printed parts that will be done for free using the 3D printers provided. With an estimated cost of shipping of about 5%, and sales tax in Illinois being about 10% adds another \$22.14. Typically, a graduate from ECE makes around \$43 an hour. We estimate that this project will take around 60

hours to complete. Thus, each partner in the project can expect a salary of $\$43/\text{hr} \times 2.5 \times 60 = \6450 . Since there are three of us, $\$6450 \times 3 = \19350 in labor cost. This comes out to be a total cost of $\$19,519.73$.

3.2 Schedule

Week	Task	Person
February 16st - February 23th	Work on glove sensor PCB	Ankur
	Work on the robotic hand PCB	Matthew
	Research hand designs	Tunc
	Compile and submit list for ordering parts	Everyone
February 23st - March 2nd	Finish PCB Design and submit for audit	Ankur and Matthew
	Work on Robot Hand Design	Everyone
	PCB Ordering 1st Round 2/26	Everyone
	Finish Design Document 2/27	Everyone
March 2nd - March 9th	Gather all components needed from self service inventory	Tunc
	Finalize the robot hand design	Everyone
	Construct 1st motor prototype on breadboard using dev board	Matthew
	Start printing 3D hand	Ankur
	Begin construction of robotic hand + glove	Tunc
	Solder PCB together	Matthew and Tunc
	Review PCB design if needed	Everyone
	Design Review 3/4 9:30 AM	Everyone
	PCB Ordering 2nd Round 3/5	Everyone
March 9th - March 16th	Begin programming robotic hand algorithm	Ankur
	Begin 2nd prototype of ESP32 microcontrollers	Matthew

	using BLE	
	Solder PCB together	Matthew and Tunc
	Continue programming robotic hand	Ankur
	Finish printing 3D hand	Ankur
	Begin constructing final design	Tunc
	Review PCB design if needed	Everyone
	PCB Ordering 3rd Round 3/12	Everyone
	Breadboard Demo	Everyone
March 16th - March 23th	Spring Break	Everyone
March 23th - March 30th	Continue programming robotic hand	Ankur
	Finish constructing prototypes	Matthew
	Solder PCB together	Matthew and Tunc
	Finish constructing robotic hand	Tunc
	Review PCB design if needed	Everyone
	PCB Ordering 4th Round 3/26	Everyone
March 30th - April 6th	Finish programming robotic hand	Ankur
	Test power consumption of each component in robotic hand	Tunc
	Begin testing worst-case scenarios with motors on hand	Matthew and Ankur
	Solder PCB together	Matthew and Tunc
	Add optional features if time permits	Everyone
April 6th - April 13th	Finalize Design	Matthew and Tunc
	Finalize any code made	Ankur
	Add optional features if time permits	Everyone
	Progress Demo	Everyone
April 13th - April 20th	Finalize Design	Matthew and Tunc
	Finalize any code made	Ankur

	Integration Tests	Everyone
	Add optional features if time permits	Everyone
April 20th - April 27th	Fix any persisting bugs or add minor changes	Everyone
	Mock Demo + Presentation	Everyone
April 27th - May 4th	Final Demo	Everyone

4 Ethics and Safety

With a project of such scope attempting to interact and benefit the deaf and hard-of-hearing community, there are many ethical and safety considerations that must be made. While working on the project, there are things that should be thought of as fundamental obligations, such as hazard mitigation and inclusive design, in order to set up for success.

4.1 Ethics

A primary ethical consideration is being able to honestly represent the capabilities and the scope of the system, as mentioned in the IEEE code of ethics. The goal of this project is to be a tool/supplement for people. Thus, we need to be honest and transparent and clearly outline the capabilities so as not to mislead users, in accordance to IEEE Code of Ethics Section I.5. This section states that engineers are required to “be honest and realistic in stating claims or estimates based on available data.”

Additionally, a large ethical concern surrounding this entire project is the areas of accessibility and inclusivity. The entire goal of the tool is to help bridge the gap and reduce communication barriers in society, along with making a positive contribution on inclusive education. We will have multiple testers from the Deaf community looking at the project in order to provide accurate feedback and avoid single-source bias. Additionally, will have people who have certifications in sign language and other learning professionals guide us on ways that students learn best. This aligns with both IEEE Section I.1, which prioritizes the safety, health and welfare of the public, and ACM Code 1.2, which requires professionals to avoid harm by considering the diverse impacts of their work.

Another aspect of the project is that it involves wireless data transmission through Bluetooth from the sensor glove. Even though the data being transferred does not involve any personally identifiable information, according to the ACM principles regarding privacy and responsible data handling, we should only utilize and obtain the necessary data, following the data minimization rule in ACM Code 1.6. This section states that professionals should only use personal data for agreed-upon purposes with an emphasis on informed consent. The IEEE Code of Ethics Section I.1 also reinforces this by requiring engineers to protect the privacy of others.

A final ethics consideration that we need to be professionally competent and rely on the engineering practices that we have learned throughout our course loads instead of blindly copying implementations from online, as required by the IEEE Code of Ethics Section I.6. Additionally, if we do utilize open source resources, we should reference and cite them upholding the standards of crediting the contributions of others properly as mentioned in IEEE Section I.5. We also commit to treating all of our testers and collaborators with the utmost respect, refraining from any form of discrimination as said in IEEE Section 2.

4.2 Safety

There are several safety risks that can arise during the development and testing of the product. As such, we have also designed mitigation strategies in accordance with the ECE 445 safety guidelines and engineering standards. First and foremost, there are electrical hazards present due to the fact that the system will be powered via the wall outlet. To comply with UL 61010-2-030, the safety standard for electrical equipment, measurement, control, and laboratory use, with circuits that are designed to connect with external devices. We will insulate exposed wiring, regulate the DC power rails with overcurrent protection, and limit the max current to draw the acceptable current values. We will also adhere to the course safety guidelines that require at least two people in the lab at all times while working with powered systems. We have also completed all of the required online safety training.

Similarly, there are mechanical hazards in the form of the servo-driven fingers pinching or snapping, and having sharp objects fly under excessive tension. UL 3100, the standard for safety for Automated Mobile Platforms, applies to this. In accordance with UL 3100 and general safety practices when working with robotics, we will operate the servos at reduced torque values

and place a clear barrier in front of the robot hand. We will also inspect all of the 3D printed parts for any cracks, fatigue, or tangled nylon string before each testing session.

There are also thermal hazards present where several servos may heat up during long periods of operation or while running when the finger stays in place. In order to mitigate this risk, in the software, we will limit the maximum amount of time that a servo will stay on and also monitor the temperature of the servos. This aligns with the course policy of reading and understanding guidelines for safe battery usage, especially for the glove sensor.

Now, looking at the safety considerations during use, potential misuse includes applying excessive force to the joints, trying to wear the robotic hand, and modifying the wireless connections while powered. These concerns came up during the proposal review and have been discussed and addressed. To mitigate these risks, we will place hard stops in the software that limit the angles of the servos so that the finger joints do not overextend beyond their intended range of motion (typical max of 60 degrees).

Following the course safety guidelines, any testing involving electrical current near human testers would require additional review and approval, but our current design only contains the non-invasive sensor glove that operates at low voltage and isolates the user from the main power supply. This isolation shows that our current design will sufficiently protect the user from electrical hazards while still achieving the goals.

5 Citations

[1] Gil, L., & Collins, L. (2022). Corrective Feedback to Second Language Learners of American Sign Language. *Sign Language Studies* 22(4), 668-702.

<https://dx.doi.org/10.1353/sls.2022.0009>.

[2] Pinnington, Jamie et al. "Machine Learning in ASL Fingerspelling Recognition: A Literature Review." 2024 IEEE 24th International Symposium on Computational Intelligence and Informatics (CINTI) (2024): 000055-000062.

[3] Aero-Hand-Open: <https://shop.tetheria.ai/>. Accessed 28 Feb. 2026.

[4] Monolithic Power Systems. "MP1584 – 3A, 1.5MHz, 28V Step-Down Converter." MPS Proprietary Information, Rev. 1.0, 8 Aug. 2011, www.MonolithicPower.com.

[5] Handson Technology. *Handson Technology User Guide MG996R Metal Gear Servo Motor*.

[6] "Espressif Documentation." *Espressif.com*, 2025, documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf.

[7] *MP1584 3A, 1.5MHz, 28V Step-down Converter*.