Review the assignment due date

NeuroTouch

**Team Number: 12 **

Team Name: Tri-State Buffers

Team Member Name	Email Address
Ananya Shivarama Bhat	ananya9@seas.upenn.edu
Anushka Jain	anushkaj@seas.upenn.edu
Devanshi Kalpeshbhai Patel	patel5@seas.upenn.edu

GitHub Repository URL: https://github.com/upenn-embedded/final-project-f25-f25_final_project_t12_tri-state_buffers

GitHub Pages Website URL: [for final submission]*

Final Project Proposal

1. Abstract

Individuals with congenital insensitivity to pain or loss of touch sensation lack the sensory feedback needed to safely interact with their environment. This project proposes a Smart Haptic Glove that restores a sense of touch awareness using a compact embedded system built around the ATmega328PB microcontroller. Each glove hand integrates multiple force-sensitive resistors (FSRs) for contact pressure, a precision temperature sensor (MCP9700) for surface heat detection, and ERM haptic motors (P1012 class) that reproduce tactile feedback through vibrations proportional to force or temperature. A low-power OLED display provides real-time sensor data and system status. The system runs on NiMH AA batteries with on-board power regulation and noise filtering, ensuring full compliance with laboratory safety requirements. The glove's firmware performs continuous analog sampling, threshold detection, and PWM-based haptic control while employing sleep and duty-cycling techniques for efficient power management. The end goal is a lightweight, wearable assistive device that enables users without tactile sensation to perceive pressure and temperature cues safely and intuitively.

2. Motivation

People affected by **Congenital Insensitivity to Pain with Anhidrosis (CIPA)** or other neuropathic conditions lack the ability to feel **pain, temperature, or touch**—fundamental sensations that protect the body from harm. Everyday tasks such as holding a hot object,

applying pressure, or gripping tools can result in severe injuries without the person realizing it.

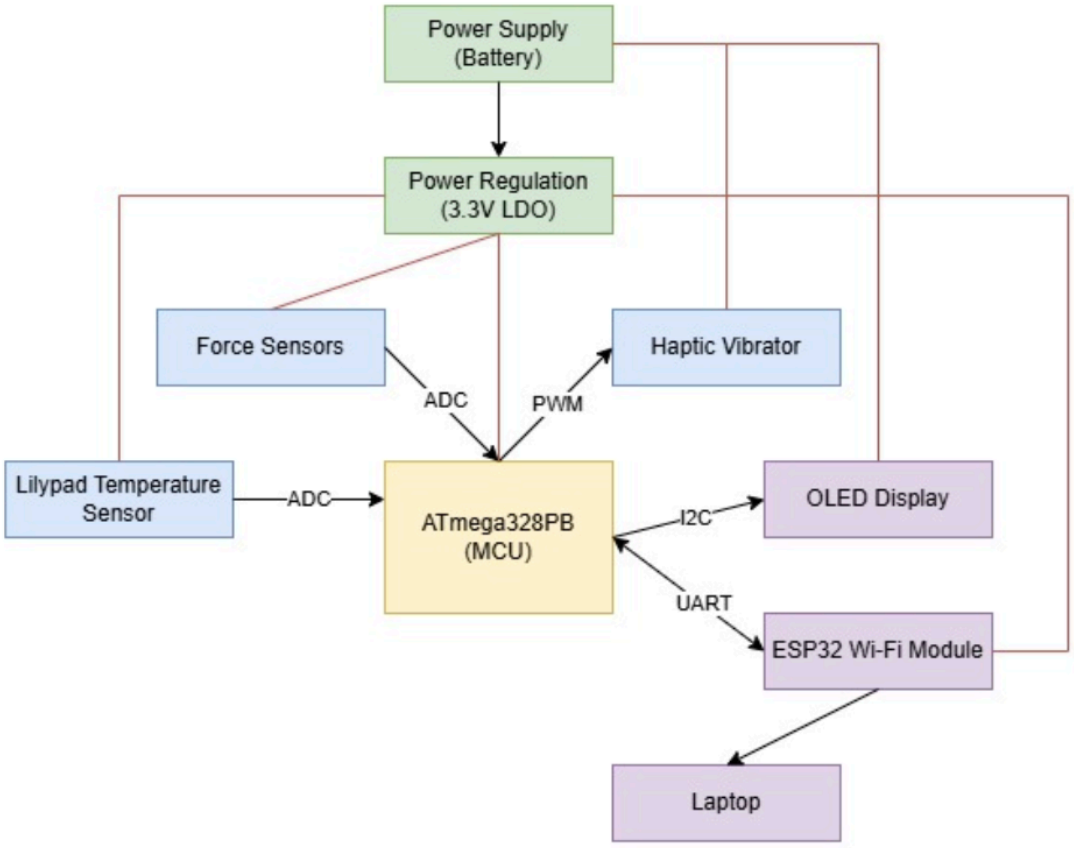
Our motivation is to **restore safe sensory awareness** through an assistive wearable device that translates real-world stimuli into **intelligent haptic feedback**. Rather than simply detecting hazards, the glove allows the user to *feel* through controlled vibrations and cues—making safety instinctive again.

The project also serves as a **demonstration of embedded systems integration**, combining analog sensing, PWM motor control, and real-time signal processing on a compact **ATmega328PB** platform. By designing a power-efficient, sensor-rich glove that operates entirely on **safe NiMH batteries**, we aim to create an accessible, wearable prototype that showcases how embedded technology can bridge the gap between human perception and digital intelligence.

Ultimately, this project is driven by two goals:

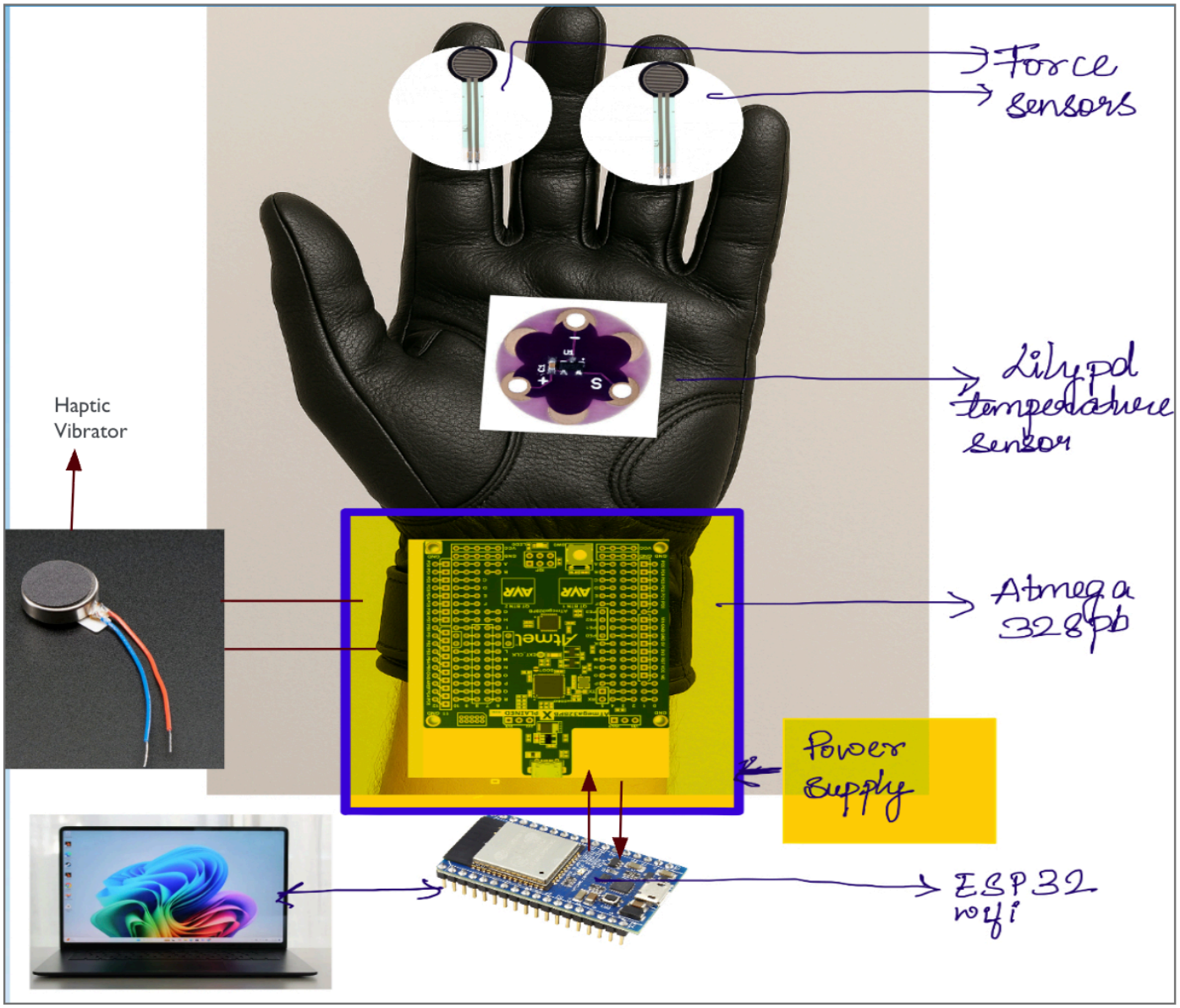
1. **Human impact** — improving quality of life for individuals who have lost tactile perception.
2. **Engineering innovation** - proving that low-cost embedded systems can deliver meaningful assistive technology when designed thoughtfully.

3. System Block Diagram



4. Design Sketches

We anticipate our final product to look like this



5. Software Requirements Specification (SRS)

The software for the Smart Haptic Glove is responsible for reading sensor data, interpreting physical contact and temperature levels, and producing real-time haptic, visual, and wireless feedback. The program shall continuously read values from all force sensors and the MCP9700 temperature sensor at least twenty times per second to ensure timely response. The readings will be processed so that small fluctuations are smoothed, keeping the displayed or acted-upon value steady when the finger is held still. When the pressure on any finger exceeds a preset threshold or when the surface temperature rises above 45 °C, the firmware shall immediately trigger the vibration motor for that finger, providing distinct feedback patterns for force and temperature. The delay between detecting a hazard and starting vibration shall be less than 50 milliseconds. The vibration intensity shall scale with applied force—light, medium, and firm pressure will correspond to low, medium, and high vibration strengths.

An OLED display will show live data such as temperature, average pressure level, and battery status, refreshing at least five times per second and automatically dimming after ten seconds of inactivity to conserve power. The system shall also transmit diagnostic and sensor data through an integrated ESP32 Wi-Fi module, which will communicate with a laptop or cloud dashboard in real time. The data packets—containing pressure, temperature, vibration intensity, and battery information—will be updated at least once per second for remote monitoring and analysis. A simple UART interface will handle communication between the ATmega328PB and ESP32 to ensure reliable data transfer. When idle, with no haptic activity and the OLED dimmed, the firmware shall enter a reduced-power state to keep average current consumption below 35 mA. Together, these requirements ensure that the glove reacts quickly and reliably to user interaction, provides meaningful tactile and visual feedback, enables wireless connectivity for external monitoring, and maintains efficient power use suitable for battery operation.

5.1 Definitions, Abbreviations

ESP 32 abreviation : The ESP32 is a small and powerful microcontroller made by Espressif that has built-in Wi-Fi and Bluetooth, making it perfect for Internet of Things (IoT) and embedded system projects. It works like a tiny computer that can read data from sensors, process it, and send or receive information from other devices or the internet. In our case, we are trying to establish communication with the ESP32 so that it can exchange data with another system, like a computer or another microcontroller. The most common way to do this is through serial (UART) communication, where data is sent and received using the TX (transmit) and RX (receive) pins. This allows us to send commands to the ESP32 and get responses back, often through a USB or serial connection. Once this basic connection is working, we can also explore other communication methods like I2C, SPI, or even wireless options such as Wi-Fi and Bluetooth for more advanced applications. OLED : An OLED display is useful because it gives the user and developers real-time visual feedback about the glove's operation. For example, the OLED can show sensor readings like temperature, pressure, or vibration intensity, so the user (or clinician) can see what the glove is detecting even if they can't physically feel it. It also helps during testing and debugging, allowing you to confirm that the sensors and haptic motors are responding correctly. So, the need for the OLED in your project is to display live sensor data and system status, making it easier to monitor performance, ensure safety, and provide a clear interface for the user. It turns invisible sensor activity into something visible and understandable — which is especially important for a project centered on restoring sensory awareness.

6. Hardware Requirements Specification (HRS)

Our device, the CIPA glove, must be able to sense temperature and pressure from the environment and alert the user before any potential harm occurs. It should detect when something is too hot, too cold, or when excessive pressure is applied, and then trigger a haptic vibration to warn the user. The glove must also display real-time sensor readings on the OLED screen so that users or caregivers can visually monitor the conditions. Overall, it should work quickly, accurately, and reliably to provide safe tactile feedback for people who cannot feel pain or touch, helping them avoid injuries in everyday situations.

During validation testing, we will measure how accurately and reliably the glove works. We will test whether it can correctly detect temperature and pressure, and if the haptic motor and OLED display respond properly. For temperature, we will compare the glove's readings with a thermometer to see if they match. For pressure, we will press the sensors using known weights or a force gauge and check how close the glove's readings are. We will also test how quickly the glove responds after detecting a change, by measuring the time between the input and the motor or display reaction. The OLED screen will be checked to make sure it shows the correct data clearly and in real time. Finally, we will repeat these tests many times to see if the system remains consistent and reliable. All these results—accuracy, speed, and reliability—will help us confirm that the glove performs safely and effectively.

6.1 Definitions, Abbreviations

ADC : In our project, the ADC (Analog-to-Digital Converter) plays an important role in allowing the glove to understand signals from the sensors. The temperature and pressure sensors used in the CIPA glove produce analog signals, which are continuous voltage levels that change depending on what they sense — for example, higher voltage for higher temperature or pressure. However, the microcontroller (like the ATmega328PB or ESP32) can only process digital signals (0s and 1s). The ADC converts these analog voltages into digital values that the microcontroller can read and interpret.

6.2 Functionality

ID	Description
1027-1001-ND (Digikey)	It offers an analog resistive output that changes with applied force (i.e., the more pressure you apply, the lower the resistance). The

ID	Description
	manufacturer lists a response time of approximately 3 µs, which means it reacts very quickly to changes in force.
MCP9700	The LilyPad Temperature Sensor is a small, wearable-friendly sensor that measures temperature and gives an analog voltage output that changes with temperature. It uses the MCP9700 chip, which outputs about 0.5 V at 0 °C and increases by around 10 mV for every degree Celsius rise in temperature.
100614	The Adafruit Vibrating Mini Motor Disc (Product ID 1201) is a small, sealed vibration motor that's ideal for adding haptic feedback in projects. It has a diameter of about 10 mm, a thickness of about 2.7 mm, and operates from roughly 2 V up to about 5 V.
NHD-C12832A1Z-FSW-FBW-3V3	It uses an SPI interface for easy communication with a microcontroller and has a transflective FSTN screen, which means it's clear to read in both bright and low-light conditions. It runs on 3.3 V and is compact, making it ideal for portable or wearable projects. In our glove project, it can be used to show sensor readings, temperature, or warning messages, giving the user a quick visual update of the system's status.

7. Bill of Materials (BOM)

This is the link to our Bill of Materials : Link to Bill of Materials

8. Final Demo Goals

For the final demonstration, our Smart Haptic Glove will be worn by a team member to showcase how touch and temperature can be sensed, interpreted, and transmitted wirelessly in real time. The glove will detect pressure through multiple FSR sensors and surface heat via an MCP9700 temperature sensor. These signals will be processed by the ATmega328PB, which controls vibration motors and sends live data to an ESP32 module for wireless transmission. The ESP32 will relay this data over Wi-Fi to a laptop dashboard or web interface, displaying real-time readings of pressure levels, temperature, and motor activity.

During the demo, we will present two modes of feedback: (1) local tactile feedback, where the glove vibrates immediately in response to sensed force or temperature; and (2) remote visualization, where sensor data appears live on the Wi-Fi dashboard. This demonstrates both the glove's assistive sensory function and its IoT data capability for potential medical or research use. The OLED display on the glove will simultaneously show sensor values and system status, while the vibration motors will produce distinct patterns for pressure and heat, illustrating intuitive feedback mapping.

The demonstration will take place indoors, with a wearable setup powered by AA NiMH batteries and a USB interface for debugging. We will show the glove interacting with various objects—pressing soft and hard surfaces, and touching warm materials—to highlight real-time wireless data transfer, haptic response, and low-power operation. This will effectively demonstrate a complete embedded system that bridges human sensory feedback and wireless communication, offering a proof-of-concept for assistive smart wearables.

9. Sprint Planning

Milestone	Functionality Achieved	Distribution of Work
Sprint #1	We plan to obtain all senors and establish connections. We also plan to setup the power system. We will also consider if we need to 3D print any particular object for the final device. In accordance to this, we will also design on a relevant software.	We have divided smaller tasks between us like assigning who would create the BOM and find the sensors. Overall, we plan to work on the project together
Sprint #2	We plan to write our firmware and test it for different cases of pressure and temperatures and setup ESP32 for communication. For all the problems that we would encounter, we would consult our account manager for advice. Incase we are able to achieve all the functionality that is desires, we plan to add additional features to the product as needed.	We plan to work together while writing the firmware
MVP Demo	We expect to see all electronics and firmware working at a basic level. We would explain	On our demo, all of us would show our work to

Milestone	Functionality Achieved	Distribution of Work
	our firmware implementation, including application logic and critical drivers that we have written and would demo our device.	our account manager and cater to any problems that we are facing
Final Demo	We expect to cater all the possible requirements of the project and display our results. We hope to satisfy our purpose of the project	We would work on the final demo and present together to all the TAs and students

This is the end of the Project Proposal section. The remaining sections will be filled out based on the milestone schedule.

Sprint Review #1

Last week's progress

In the past week, we started off by meeting Kevin and ordering all our parts which would be needed for the final project. Accordingly the BOM was made as well. We also found some force sensors in Detkin that would be used in our project. The force sensors were checked and tested using arduino uno. It was interfaced with atmega .

We planned to include IR proximity sensors in addition to the sensors we already included in the intial plan.

The IR proximity sensors will communicate with the Atmega through I2C. The functionality of these sensors was also verified with the Atmega 328Pb.

Current state of project

We are currently waiting on the parts which we have ordered.

Meanwhile we have set up the force sensors and the IR proximity sensors and have written the code for that.

We are coding for the temperature sensors and trying to explore the coding for the LilyPad Dev board

Next week's plan

Test the sensors.

Use and test the e-sewing technique.

Code and explore the set up with the Lilypad Dev board.

Integrate the circuit with the gloves.

Test the Haptic Vibrator.

Sprint Review #2

Last week's progress

Current state of project

Next week's plan

MVP Demo

1. Show a system block diagram & explain the hardware implementation.
2. Explain your firmware implementation, including application logic and critical drivers you've written.
3. Demo your device.
4. Have you achieved some or all of your Software Requirements Specification (SRS)?
    1. Show how you collected data and the outcomes.
5. Have you achieved some or all of your Hardware Requirements Specification (HRS)?
    1. Show how you collected data and the outcomes.

6. Show off the remaining elements that will make your project whole: mechanical casework, supporting graphical user interface (GUI), web portal, etc.
7. What is the riskiest part remaining of your project?
    1. How do you plan to de-risk this?
8. What questions or help do you need from the teaching team?

Final Project Report

Don't forget to make the GitHub pages public website! If you've never made a GitHub pages website before, you can follow this webpage (though, substitute your final project repository for the GitHub username one in the quickstart guide): https://docs.github.com/en/pages/quickstart

1. Video

[Insert final project video here]

- The video must demonstrate your key functionality.
- The video must be 5 minutes or less.
- Ensure your video link is accessible to the teaching team. Unlisted YouTube videos or Google Drive uploads with SEAS account access work well.
- Points will be removed if the audio quality is poor - say, if you filmed your video in a noisy electrical engineering lab.

2. Images

[Insert final project images here]

Include photos of your device from a few angles. If you have a casework, show both the exterior and interior (where the good EE bits are!).

3. Results

What were your results? Namely, what was the final solution/design to your problem?

3.1 Software Requirements Specification (SRS) Results

Based on your quantified system performance, comment on how you achieved or fell short of your expected requirements.

Did your requirements change? If so, why? Failing to meet a requirement is acceptable; understanding the reason why is critical!

Validate at least two requirements, showing how you tested and your proof of work (videos, images, logic analyzer/oscilloscope captures, etc.).

ID	Description	Validation Outcome
SRS-01	The IMU 3-axis acceleration will be measured with 16-bit depth every 100 milliseconds +/-10 milliseconds.	Confirmed, logged output from the MCU is saved to "validation" folder in GitHub repository.

3.2 Hardware Requirements Specification (HRS) Results

Based on your quantified system performance, comment on how you achieved or fell short of your expected requirements.

Did your requirements change? If so, why? Failing to meet a requirement is acceptable; understanding the reason why is critical!

Validate at least two requirements, showing how you tested and your proof of work (videos, images, logic analyzer/oscilloscope captures, etc.).

ID	Description	Validation Outcome
HRS-01	A distance sensor shall be used for obstacle detection. The sensor shall detect obstacles at a maximum distance of at least 10 cm.	Confirmed, sensed obstacles up to 15cm. Video in "validation" folder, shows tape measure and logged output to terminal.

4. Conclusion

Reflect on your project. Some questions to address:

- What did you learn from it?

- What went well?
- What accomplishments are you proud of?
- What did you learn/gain from this experience?
- Did you have to change your approach?
- What could have been done differently?
- Did you encounter obstacles that you didn’t anticipate?
- What could be a next step for this project?

References

Fill in your references here as you work on your final project. Describe any libraries used here.