ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Smart Cognitive-Motor Rehabilitation Mat for Remote Exercise Monitoring

Team #47

SCOTT LOPEZ (slope22@illinois.edu)
JASHAN VIRDI (jvird2@illinois.edu)
ADITHYA BALAJI (abalaji5@illinois.edu)

TA: Micheal Gamota

May 7, 2025

Abstract

This final report describes our design for a Smart Cognitive-Motor Rehabilitation Mat for improving at-home physical rehabilitation, particularly for individuals with Multiple Sclerosis. We describe details of the design and implementation of our project, also describe changes that we have made to our design throughout the duration of the project.

Contents

1 Introduction

1.1 Problem

The smart-mat was built by a previous team in order to help patients with Multiple Sclerosis (MS). MS is a lasting disease that affects the central nervous system by corroding the myelin that protects the nerves. The loss of this protection causes scar tissue to form which hinders the transmission of signals along the nerves. MS can progress in two different ways: relapsing-remitting MS where there are repeated attacks and primary progressive MS where it progresses over time.[1] The smart-mat makes rehabilitation for these patients more accessible and safe in the patient's home.

Square stepping exercise is a method for the rehabilitation of all individuals with MS. The current smart mat prototype has unreliable sensors and a dependency on Wi-Fi for the routines. The sensors sometimes show that a square that wasn't stepped on was, which gives false feedback and can discourage someone who is trying to improve. Also, the mat uses an html webpage to store the routines, and if Wi-Fi is not available in the location, these routines are unavailable.

1.2 Solution

The solution to the unreliable sensors was to use mechanical switches(buttons). The buttons were connected in rows and columns to allow for easy scanning of the squares. We verified the new mat design could withstand repeated rolling and unrolling without degrading sensor reliability.

In order to solve the reliance on Wi-Fi, we transferred over to Bluetooth and an IOS app to store the relevant data. Bluetooth does not require Wi-Fi to function and allowed the smart mat to work in locations with low Wi-Fi signals.

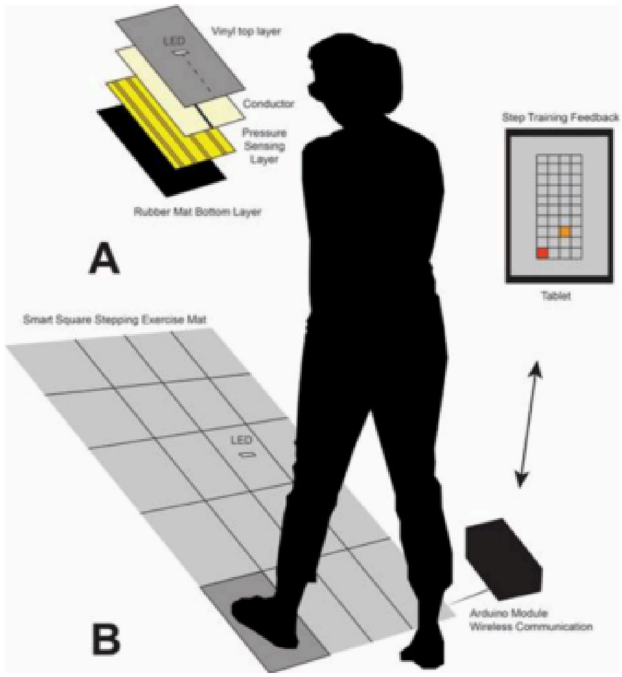

Figure 1: Visual aid for smart mat (image credits: Professor Hernandez)

1.3 High-Level Requirements

We considered our project a success by completing the following requirements:

- Data Transmission Latency: The system had to provide real-time feedback with a data transmission delay of no more than 500 ms between the smart mat and the mobile device.
- Reliability: Improved reliability of sensors to withstand a minimum of thirty rolls .
- Power Source: Ran the mat on a 5 V USB-C to allow ease of replacing power adapter.

2 Design

2.1 Block Diagram

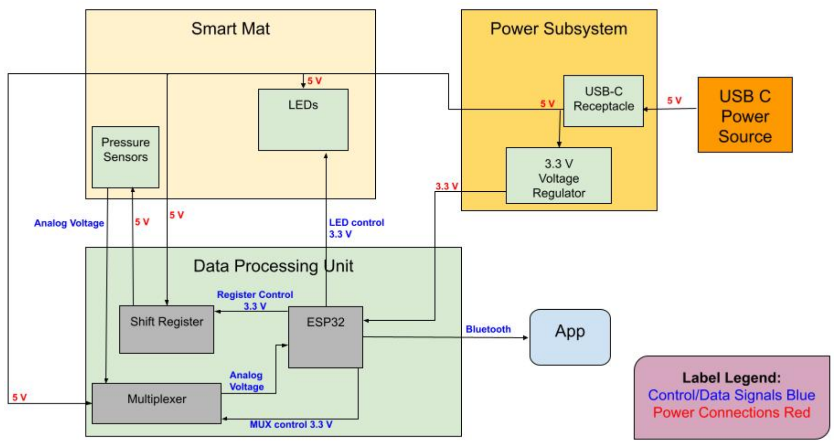

Figure 2: Block Diagram

2.2 Overview of Subsystems

2.2.1 Mat Subsystem

The mat consists of four main layers. The bottom layer is the anti-slip yoga mat which is intended to prevent the mat from slipping out from under the patient. The middle layer is the wires and buttons. The LEDs are next, and the top layer is a protective film.

The first prototype design of the pressure sensor involved two copper strips crossing with a Velostat layer in the middle as shown in Figure 3. Velostat is a pressure-sensitive conductive material that has decreasing resistance as pressure is applied. This was to allow for the detection of a patient's step by measuring the voltage difference across these strips. This design was used due to Velostat's flexibility, allowing for use in a foldable mat. We decided to cover each side of the Velostat in copper foil as shown in Figure 4 to allow for a larger detection area to reduce the patient missing the sensor.

Three designs using this configuration were tested. The first configuration, as shown in Figure 5, had the copper foil sticking to the Velostat using adhesive. This did not work due to the pressure applied by the adhesive constantly shorting the connection. The second prototype, as shown in Figure 6, wrapped the copper foil around the wire and had better

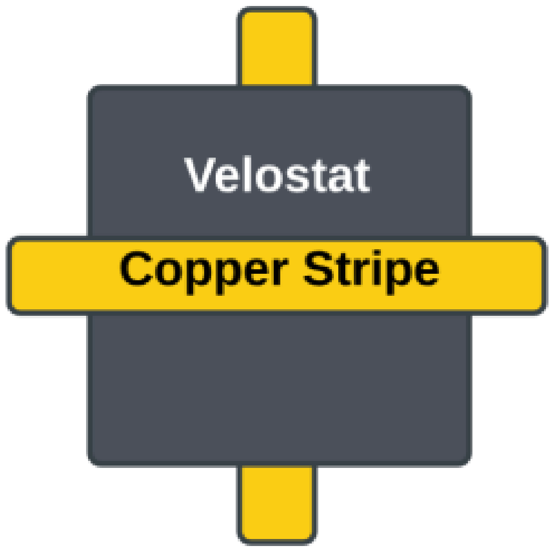

Figure 3: Current wire and Velostat configuration

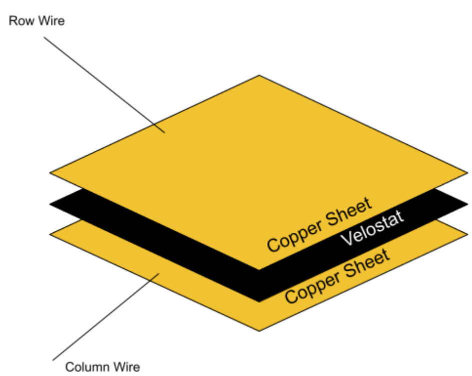

Figure 4: New Velostat configuration

readings; however, it was inconsistent. Lastly, aluminum was used on each side of the Velostat, as shown in Figure 7. This design was tested in one square and worked, so a 2 by 6 prototype was made. After testing again, it was realized that Velostat was too inconsistent and we decided to use buttons.

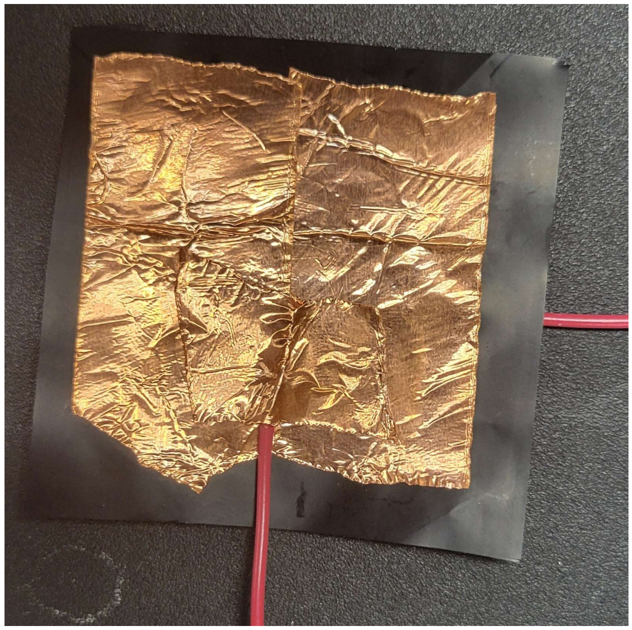

Figure 5: First Velostat prototype

Figure 6: Second Velostat prototype

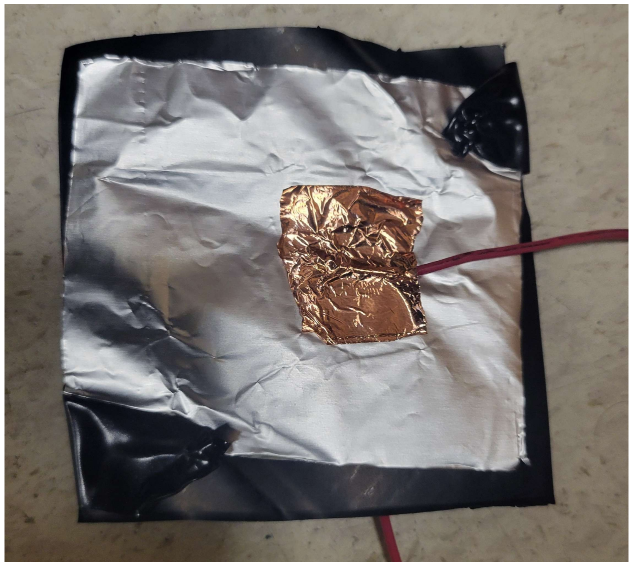

Figure 7: Third Velostat prototype

To reduce incorrect readings, insulated wires were used between each button to better isolate them and provide safety. The buttons are arranged in a 4 x 8 grid with the wires connecting the sensors forming an (x,y) grid. These rows and columns are read by the DPU to determine correct and incorrect steps. Each square on this grid is 20 cm x 15 cm due to the small size of a yoga mat.

Since we are powering this project with USB-C, we used WS2812B LEDs because they run on 5 V. These LEDs are programmable through one data pin which will be connected to the ESP to allow for easy indication of which square needs to be stepped on. There are 60 LEDs per meter to reduce the amperage of the LEDs. Each square has seven LEDs on either side and these change between three different colors. Green indicates that the correct square was stepped on, while red indicates an incorrect square was stepped on. Blue is used to show the user the next square in the routine that needs to be stepped on.

2.2.2 Data Processing Subsystem

The final ESP used for this project is the ESP32-S3. We used this because it supports Bluetooth Low Energy (BLE) that we used to connect to an app that stores the routines. The -S3 is also a dual-core microcontroller. Core 0 runs the FreeRTOS system tasks, BLE stack, and background services handled by ESP-IDF. Core 1 runs the ESP32-S3 code which includes setup(), loop(), sensor scanning and LED updates. The main data handled by the ESP BLE module are the routines and steps.

There is also an HTML-based website that can be used with this functional prototype. In this website, users will be able to connect to BLE, choose a routine from a drop-down list and observe their performance in real time with a virtual mat displaying the sensor to step on as well as correct or incorrect steps. Finally, there will be a screen that shows statistics such as correct steps, after incorrect steps, and total duration.

The routines are pre-programmed through having an array with the sequence of sensor numbers to be pressed on, from sensor 0 to sensor 31. The ESP will be given the routine based on what is selected on the app. Based on this routine, it will scan the squares and check which step the person is on with the routine. The step to press will be highlighted in blue. If a correct press is made it will switch to green. If an incorrect press is made it will be highlighted with red. The ESP will then output these steps to the website where the virtual grid will have the same color coding.

The scanning method is efficient and allows for keeping the GPIO pins to a minimum. This method uses an 8-bit shift register (SN54HC595) to supply a high signal to the columns. The ESP provides the initial one that will be shifted across four of the bits and the ESP also provides the clock signal which runs at around 100Hz rather than the previous 10 Hz. This change in clock frequency will allow for more data samples. For each of the columns the register shifts through, the multiplexer (CD74HC4067) scans through 8 rows. If the voltage difference is greater than 3V, then the ESP will read what column and row the multiplexer and shift register are currently on to determine which square was stepped on. As stated above, the ESP will then compare this with the routine.

For testing, there were a series of sequential steps taken. The multiplexer and shift register were unit tested individually to make sure the scanning logic would work. Through this we managed to identify an inconsistency in our shift register IC. Swapping it out for a new one fixed this issue. Next, BLE was tested to ensure messages could be sent via BLE logs, and the latency was tested through a "PING-PONG" mechanism where the ESP sent a signal to the website, and the website would send back a signal to the ESP, and this total roundtrip duration divided by two would give the one-way latency. This latency was observed to be no greater than 60ms at any point, falling well within the requirement. The range was tested as well, and BLE continued to function even from separations as big as 20 meters between the ESP and the phone or laptop.

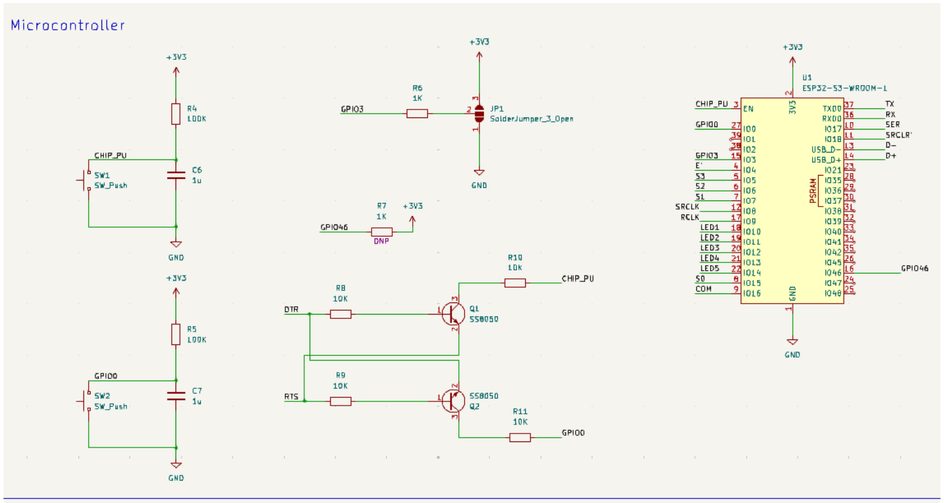

Figure 8: Microcontroller and connections

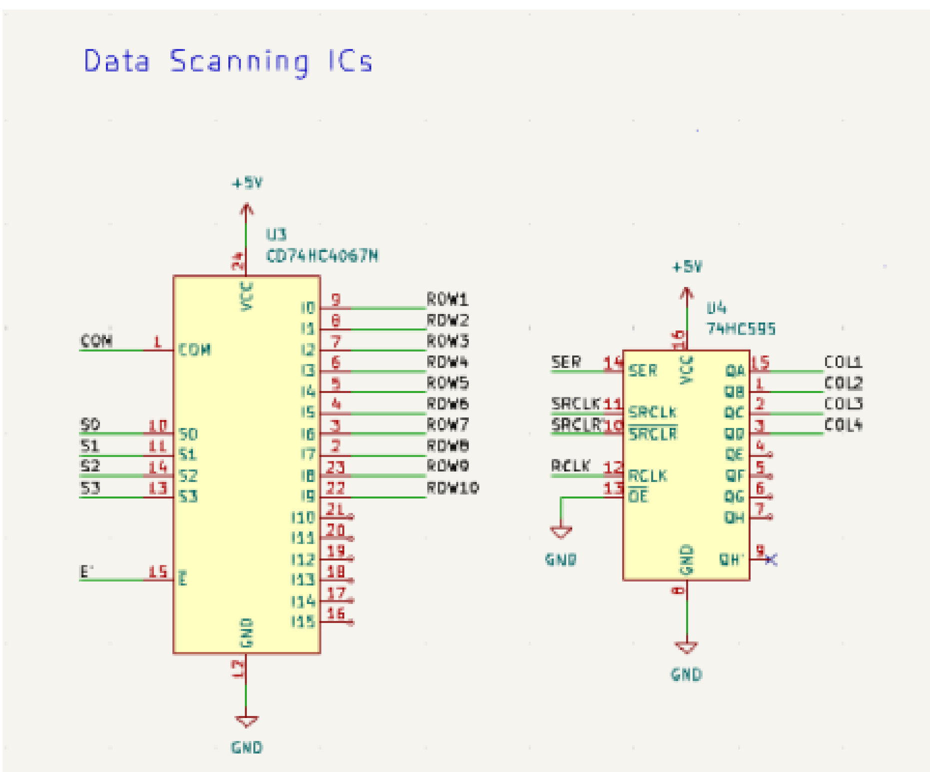

Figure 9: ICs and connections

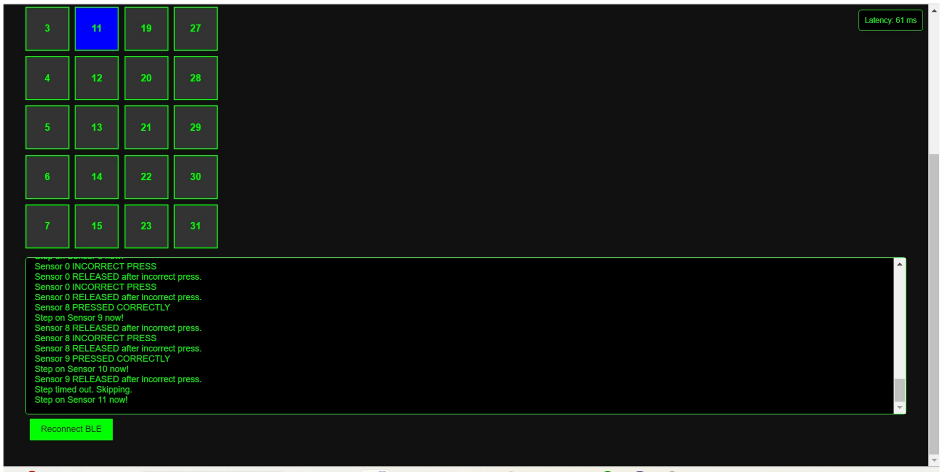

Figure 10: Application interface

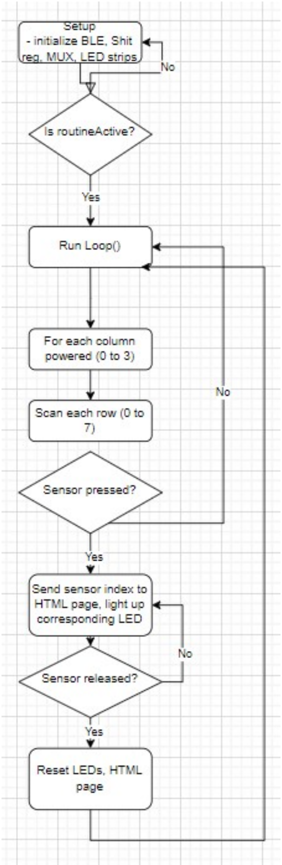

Figure 11: Flowchart of ESP code

2.2.3 Power Subsystem

The power subsystem provides power for the LEDs, the ESP32-S3, and the shift register and multiplexer ICs on the board. A USB-C cable is plugged into a USB-C receptacle port on the PCB, which gets power from power sources such as the USB ports on a laptop or tablet, or a charging brick plugged into a wall. This 5 V power from the USB-C port is distributed to the LEDs. There is a 5 V to 3.3 V voltage regulator (LM1117) on the PCB. This provides the 3.3 V needed for the ESP32-S3, shift register and multiplexer.

Based on information from the data sheets of the ESP32-S3 and the LEDs, the voltage requirements of the power subsystem are supplying 5 V ± 0.3 V from the USB-C port, and 3.3 V ± 0.2 V to the ESP32-S3. After assembling the board, we tested these requirements with a multimeter. We found that when powered by USB-C, the voltage at the 5 V node was 4.992 V, which was well within range. When probing the 3.3 V node, we observed a voltage of 3.287 V, also well within specifications.

According to the datasheet, the ESP32-S3 typically consumes $\tilde{2}$40 mA when WiFi/Bluetooth are active but may spike up to 500 mA during transmission. The LEDs, depending on brightness, can consume significant current. Based on data sheets each LED consumes around 60 mA. To determine the maximum current drawn at a time we need to determine the ESP current and LED current and use the equation

$$I_{total} = I_{ESP} + N_{LED} I_{LED} * 4 \tag{1}$$

Using this equation we know that the ESP maximum current is 500 mA and there are N=7 LEDs on each side of the square. Since the lights surrounding at most two squares can be at any given time, the maximum current draw of the LEDs is 7 * 4 * 60 mA is 1.68 A. The hypothetical total current draw during a squares activation is 2.18 A. However, when we tested the current draw, we observed a much lower number. In order to test the current draw, rather than powering the board from the USB-C port, we powered it by connecting a DC power supply to one of the LED screw terminals. Since these were connected to the 5 V on the board, supplying power here would work fine. From this, we tested the board with 2 zones LEDs on and observed a current draw of 0.651 A. From this, we know the real maximum possible current draw is less than two times 0.651 A, or less than 1.32 A. Since the USB-C port is designed to supply 3 A, and the USB ports on our laptops, tablets can supply more than 1.32 A, this means that the current drawn can safely be provided by USB-C.

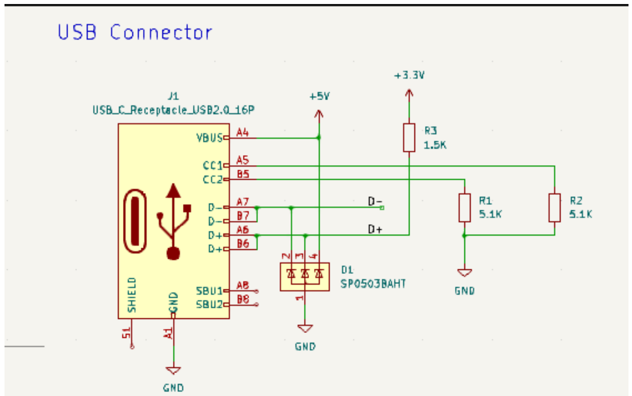

Figure 12: USB-C connector

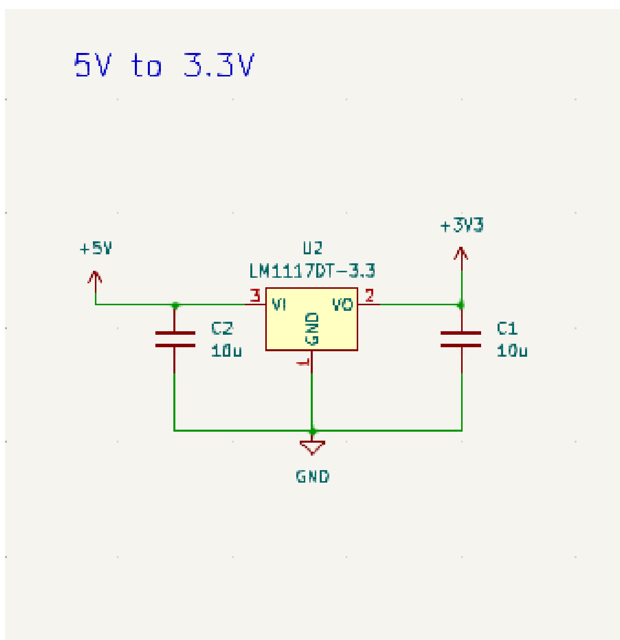

Figure 13: 5V to 3.3V Voltage Regulator

2.3 Requirements, Verifications and Results

Table 1: Mat Subsystem

Requirements	Verifications	Results
1) Can roll at least 30 times and sensors aren't affected 2) The incorrect squares should not be activated if not stepped on 3) Velostat should accurately detect if a person has stepped on them	1) Roll and unroll the mat 30 times. After this step on squares and measure the voltage. If within $\pm .05$ V from original test then good 2) Step and unstep on multiple squares in sequence. If an incorrect square was recorded isolate it and determine what the problem is related too. Fix problem then repeat first part 3) Step on each square individually and vary how it is stepped on. Record if it was sensed or not. Adjust voltage threshold if not detecting	1)Mat was rolled and unrolled 30 times and none of the sensors were affected 2) Only one square could be activated at a time to prevent this. 3) Since buttons are used, the accuracy was improved

Table 2: DPS Subsystem

Requirements	Verifications	Results
1) Should correctly load shift register to keep correct scanning frequency 2) Should send data during routine within 500 ms and accurate 3) Connected device should be able to transmit data across a room	1) Measure shift-register outputs and if incorrect change ESP load timings 2) Check time to send and receive data through ESP timestamps and compare input to output for accuracy 3) Read data sent by ESP from varying locations in a room and compare accuracy	1) Shift register correctly scanned through squares so timing was correct 2) Using Ping Pong method the latency recorded was 60 ms 3) Walked down the hall and measured a distance of at least 20 m without any issues

Table 3: Power Subsystem

Requirements	Verifications	Results
1) System should get 5$\pm$ 0.3 V from the USB-C port, and be able to draw enough current for the microcontroller and the LEDs (nominally 2.66 A) 2) System should be able to provide 3.3 V $\pm$0.2 V to the ESP32-S3	1) Test with multimeter during use since LEDs take a lot of amperage. Voltage should be within bounds and current should be sufficient for the LEDs and microcontroller. 2) Use multimeter to test voltage	1) Used DC supply and got around 0.65 A of current draw for two LEDs which is less than 3 A and the multimeter showed 4.992 for USB-C voltage 2) Voltage of 3.286 was measured

3 Cost and Schedule

3.1 Cost Analysis

Assuming an hourly rate of $44 per hour, at around 2.5 hours a day, and 50 days total, the cost of labor per person would be $\$44 * 2.5 * 50 = \$5500$. For 3 team members, the total cost would be three times $5,500 = $16,500.

3.2 Estimated Part Costs

Table 4: Estimated Part Costs

Part Number/Description	Manufacturer	Quantity	Cost
ESP32-S3-WROOM-1	Espressif Systems	1	$5.06
USB4085-GF-A (USB-C Connector)	GCT	1	$0.88
1935174 (3 position terminal block)	Phoenix Contact	5	$0.55
0022232101 (10 position header)	Molex	1	$0.54
0022232041 (4 position header)	Molex	1	$0.28
PPTC051LFBN-RC (5 position header)	Sullins Connector Solutions	1	$0.42
LM1117DT-3.3/NOPB (Voltage Regulator)	Texas Instruments	1	$1.69
SN74HC595DR (Shift Register)	Texas Instruments	1	$0.26
CD74HC4067M96 (16:1 MUX)	Texas Instruments	1	$0.58
B3S-1000 (Button)	Omron Electronics Inc-EMC Div	34	$9.69
SP0503BAHTG (Diodes)	Littelfuse Inc.	1	$0.70
GRM21BR61H106ME43L (10 uF capacitor)	Murata Electronics	3	$0.26
CL21B105KBFNNNG (1 uF capacitor)	Samsung Electro-Mechanics	3	$0.10
CL21F104ZAANNNC (0.1 uF capacitor)	Samsung Electro-Mechanics	1	$0.10
RMCF0805JT1K00 (1kOhm resistor)	Stackpole Electronics Inc	2	$0.10
RMCF0805FT1K50 (1.5kOhm resistor)	Stackpole Electronics Inc	1	$0.10
RMCF0805JT5K10 (5.1kOhm resistor)	Stackpole Electronics Inc	2	$0.10
RMCF0805JG10K0 (10kOhm resistor)	Stackpole Electronics Inc	4	$0.10
RMCF0805JT100K (100kOhm resistor)	Stackpole Electronics Inc	2	$0.10
(WS2812B) Addressable LEDs	LOAMLIN	5	$99.95

Part Number/Description	Manufacturer	Quantity	Cost
(Anker 313) 45W USB C Charger Block	Anker	1	$19.99
(B8757011) 100W USB C Cable	Anker	1	$5.99
Duct Tape	Duck	1	$7.16
Tablecloth 60 x 84 Inch	sancua	1	$8.99

Adding the estimated labor cost and the cost of all parts results in a total estimated cost of $16,500 + $258.60 = $16,758.60.

3.3 Schedule

The following schedule will allow us to efficiently split up manpower and resources, as well as keep track on the different parts of the project to make sure we will meet the eventual goal of having a functional product ready for the final demo.

Table 5: Schedule by week and Members

Week	**Tasks**	**Members**
3/3	Work on Breadboard prototype	All members
3/10	ESP Bluetooth	Adithya
3/17	Spring break	N/A
3/24	Solder and test PCB	Jashan, Scott
3/31	Build mat	All Members
4/7	Build mat	All Members
4/14	Test/debug project	All Members
4/21	Mock demo, begin final report and final presentation	All Members
4/28	Prepare for final demo, practice final presentation	All members
5/5	Finish final report, presentation	All Members

4 Conclusion

4.1 Ethics and Safety

Data Privacy and Security (IEEE Code of Ethics, Principle #1 & #5) [2]

- The smart mat collects user activity data, which could include sensitive health-related information. Unauthorized access or misuse of this data has the potential to lead to privacy violations.
  - To mitigate risks, we will be aiming to encrypt data transmission for the Bluetooth implementation and have the mat ready to comply with relevant privacy laws such as GDPR and/or HIPAA if used in a medical setting.

Reliability and Accuracy (IEEE Code of Ethics, Principle #3 & #6) [2]

- When making modifications to the existing prototype, we would need to ensure the mat continues to accurately detect user steps and respond with minimal latency to avoid incorrect feedback, which could lead to ineffective training or injury.
- Proper testing and verification against ground truth data would need to be conducted before the mat is ready to be used in a consumer setting.

Accessibility and Inclusivity (ACM Code of Ethics, Principle 1.4) [3]

- The mat would ideally be designed for a diverse range of users, including those with disabilities, to ensure inclusivity and as many recovering patients as possible to use the smartmat.

Responsible Development and Testing (IEEE Code of Ethics, Principle #9) [2]

- We would seek to ensure safety during actual mat testing if any, making sure there is no possibility of loose electrical components or improper sensor readings that could mislead users.

4.2 Summary

Overall, we were able to create a prototype of a mat which was extremely reliable, powered by USB-C, and included wireless communication with BLE. Thus, we accomplished all of our high level goals for the semester. Additionally, we achieved every requirement for each of our subsystems. We implemented wireless communication with BLE which is reliable, has a low latency, and a sufficiently long working distance. We simplified the power subsystem of the mat to run off a single power source plugging into a USB-C receptacle. We increased the robustness of the physical mat by switching to a design utilizing buttons rather than Velostat.

The biggest challenge we faced while working on this project was trying to improve the reliability of the design. We spent the majority of the timeline trying to find a way to make the Velostat more reliable. Ultimately, we ended up reaching the conclusion that Velostat was simply not consistent enough for this use case and switched to using buttons, which happened to be extremely reliable.

In the future, our design will be scaled up onto a full sized mat. This will be easy to do as it just involved physically assembling a new mat and plugging in the wires into the PCB. The PCB also allows for experimentation with different types of sensors, such as air pressure sensors. As long as the row and column design is preserved, our PCB can remain the same and be used on a new mat. After the full sized mat is built and connected to our PCB, it will be tested by real end users.

References

[1] John Hopkins Medicine. ""Multiple Sclerosis (MS)"." (2025), [Online]. Available: https://www.hopkinsmedicine.org/health/conditions-and-diseases/multiple-sclerosis-ms (visited on 03/01/2025).

[2] IEEE. "IEEE Code of Ethics." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/01/2025).

[3] Association for Computing Machinery. "ACM Code of Ethics and Professional Conduct." (2018), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 03/01/2025).

Appendix A

[1] M. E. Hernandez et al., “INTELLIGENT SQUARE STEPPING EXERCISE SYSTEM FOR COGNITIVE-MOTOR REHABILITATION IN OLDER ADULTS WITH MULTIPLE SCLEROSIS,” in Proc. 2025 Design of Medical Devices Conf., Apr. 2025.

[2] “ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U Datasheet 2.4 GHz Wi-Fi (802.11 b/g/n) and Bluetooth ® 5 (LE) module Built around ESP32-S3 series of SoCs, Xtensa ® dual-core 32-bit LX7 microprocessor Flash up to 16 MB, PSRAM up to 8 MB 36 GPIOs, rich set of peripherals On-board PCB antenna.” Available: https://www.espressif.com/sites/default/files/docume
s3-wroom-1_wroom-1u_datasheet_en.pdf

[3] Available: ”WS2812B Intelligent control LED integrated light source Datasheet” https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf

Appendix B Mat LEDs

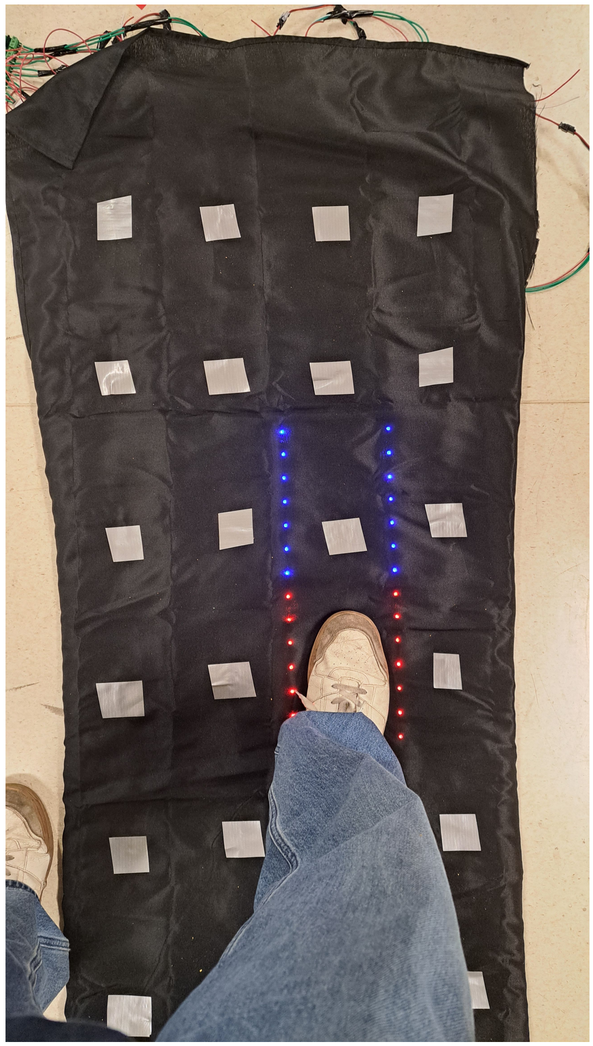

Figure 14: Mat with wrong sensor pressed

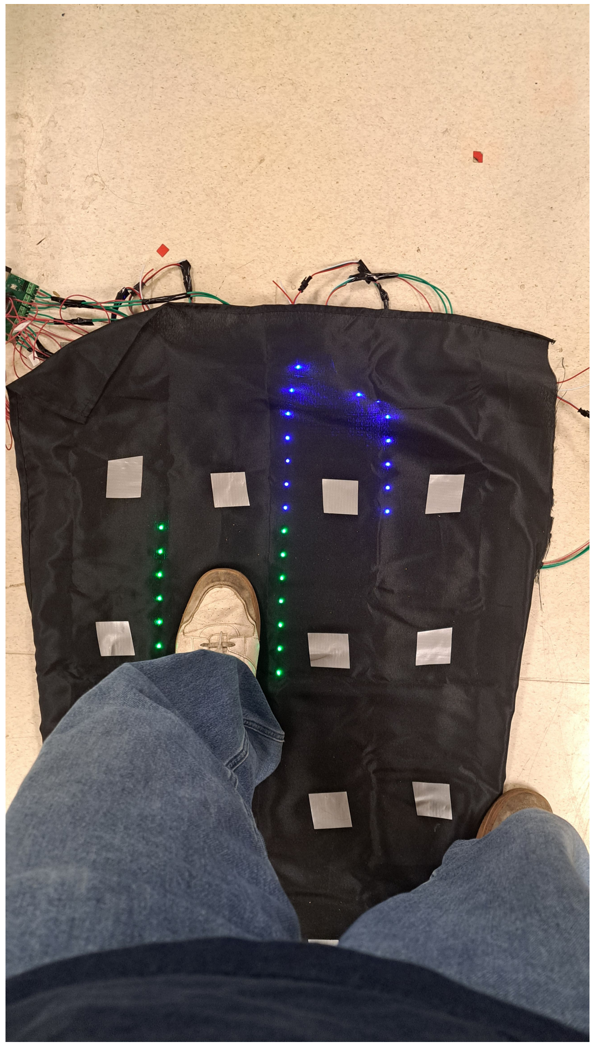

Figure 15: Mat with correct sensor pressed

Appendix C PCB

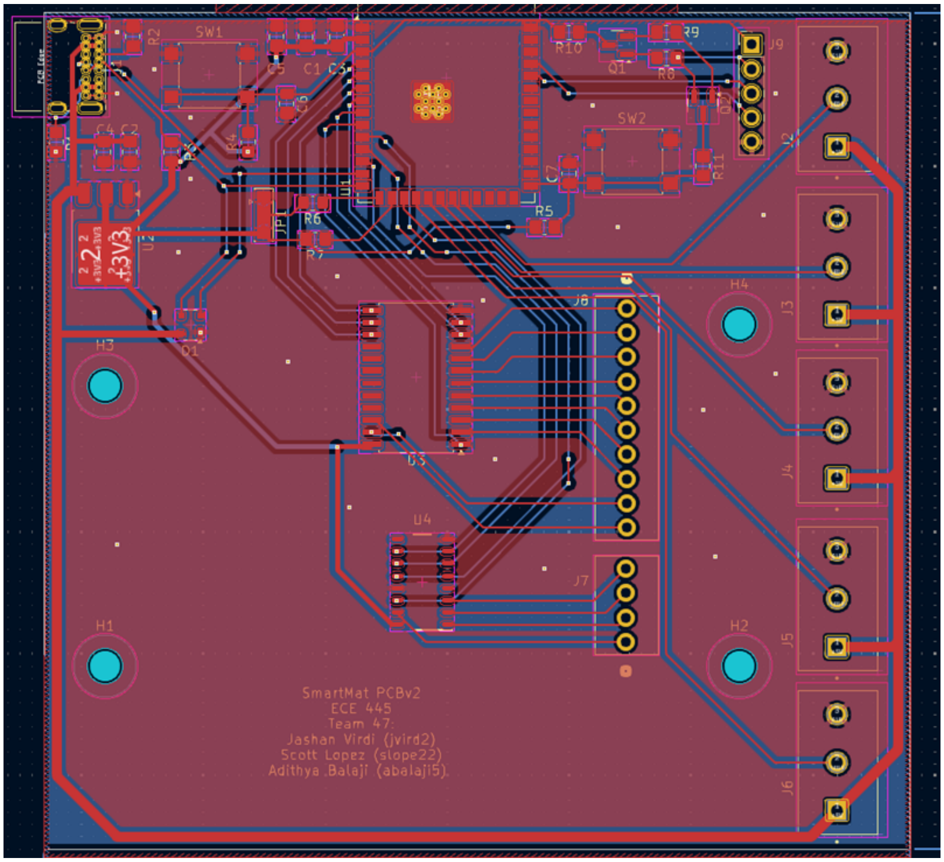

Figure 16: PCB layout

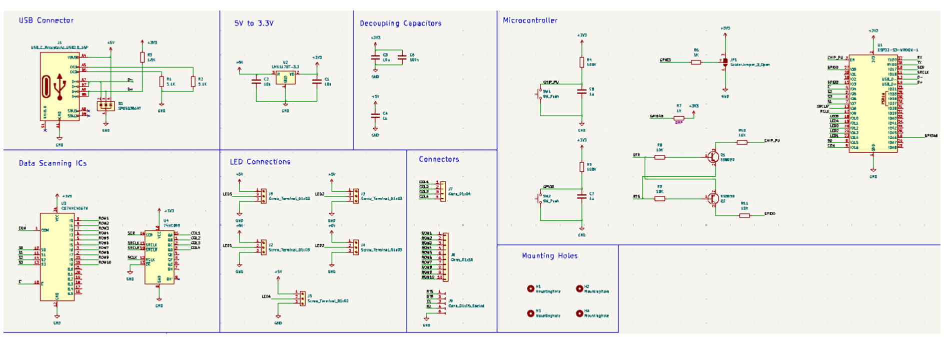

Figure 17: PCB Schematic

Appendix D Power Subsystem Tests

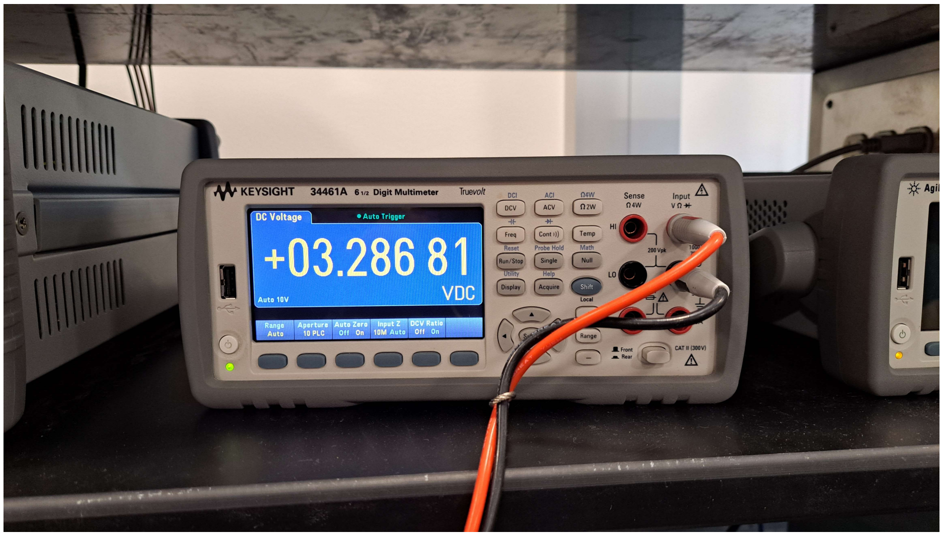

Figure 18: 3.3 V test with multimeter

Figure 19: 5 V test with multimeter

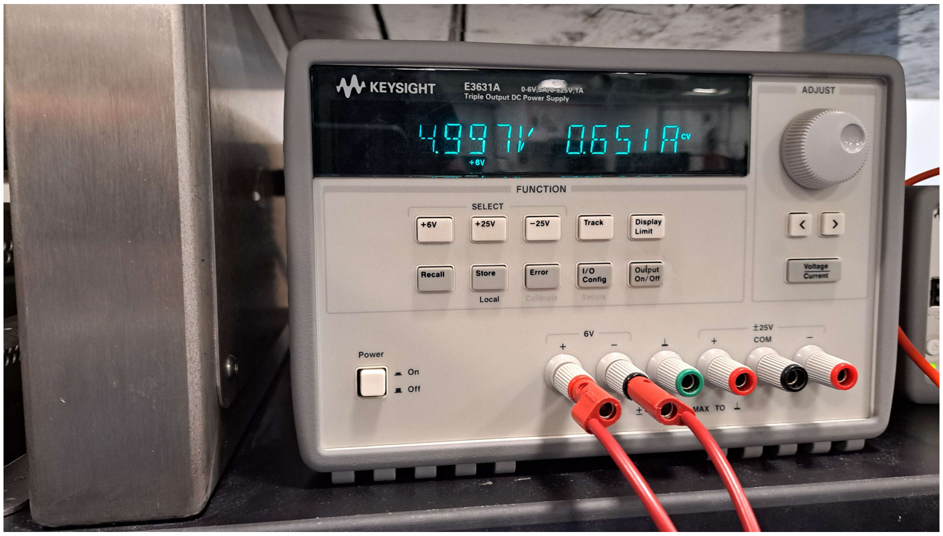

Figure 20: Current draw test with DC power supply