ECE 445

Senior Design

Final Report

Move Displaying Chessboard

Tim Chen

Matthew Trela

Jeanjuella Tipan

TA: Wenjing Song

Spring 2026

Group 49

Abstract

Chess is a complicated game with a steep learning curve. The move displaying chess board is a low-cost, embedded smart chessboard designed to give real-time feedback to accelerate learning. The board uses an ESP32-S3 microcontroller, 64 reed switches, and magnetized pieces to track the board state. When a piece is lifted, addressable LEDs illuminate all legal destination squares with distinct colors for normal moves, captures, and an engine recommended move. OLED displays and buttons provide user control to toggle LED illumination and perform error recovery. The system can run for over four hours on a portable power bank. With a total component cost of $173.04, this product is suitable for school chess clubs and other educational settings.

Contents

1 Introduction

1.1 Problem

Chess supports critical thinking and strategic planning and is common in elementary and middle school enrichment programs. However, beginners face a high barrier to entry due to difficulties understanding piece movements, legal moves, and check/checkmate conditions. These challenges slow learning and reduce engagement, particularly in school clubs with limited instructor attention.

Although existing smart chess boards provide real-time feedback and move guidance, they are typically expensive and designed for individual consumers, making them impractical for budget-constrained classrooms and clubs. This leaves a gap between low-cost traditional boards (no feedback) and high-end smart boards (inaccessible to most schools). This project addresses that gap by developing a low-cost smart chess board that offers immediate, visual guidance on legal moves, helping beginners learn intuitively while remaining affordable and suitable for educational use.

1.2 Solution

The solution is a low-cost, embedded smart chess board that provides real-time visual feedback to help beginners learn legal piece movement and basic gameplay. When a player lifts a piece, the system detects the board state change and lights up all legal destination squares using LEDs beneath each square, reducing cognitive load and allowing players to focus on strategy. Optionally, the board can display a recommended move from an onboard chess engine, toggled via a simple user interface.

The system uses an ESP32-S3 microcontroller for board state tracking, move validation, and engine computation. Each chess piece contains two small neodymium magnets, and each square has a reed switch sensor to detect piece presence. The microcontroller maintains an internal board representation to determine when a piece is lifted and compute all legal moves. Addressable LED strips, segmented into eight rows and daisy-chained, illuminate specific squares. A small OLED display and push-button interface allow users to toggle engine assistance, handle special cases like pawn promotion, and receive feedback for illegal moves or system status. Power is supplied by a mobile battery pack with current management for the LED load.

1.3 Visual Aid

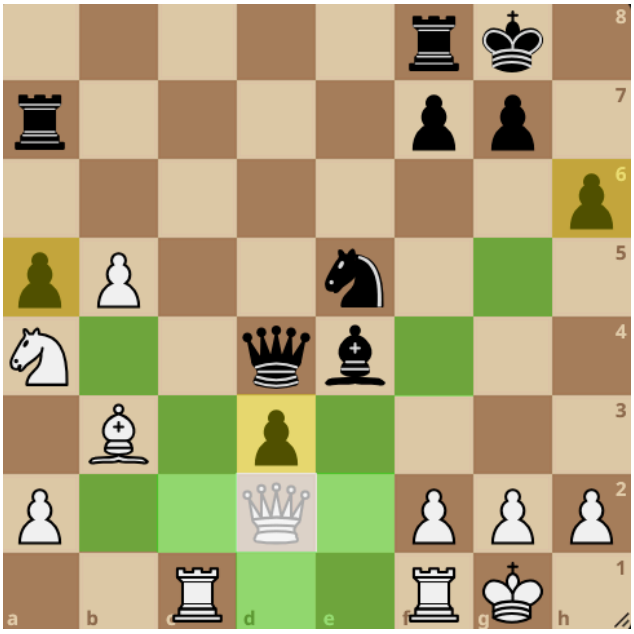

Figure 1: Visual Aid (Colors indicate a lit LED)

1.4 High Level Requirements List

1. Correctly display moves that are legal in chess at the current position and the best move (if turned on) within 5 seconds
2. Detect when board reaches an error state and execute subroutine to return to valid state
3. Whole system can run continuously for 1 hour without intervention

2 Design

2.1 Block Diagram

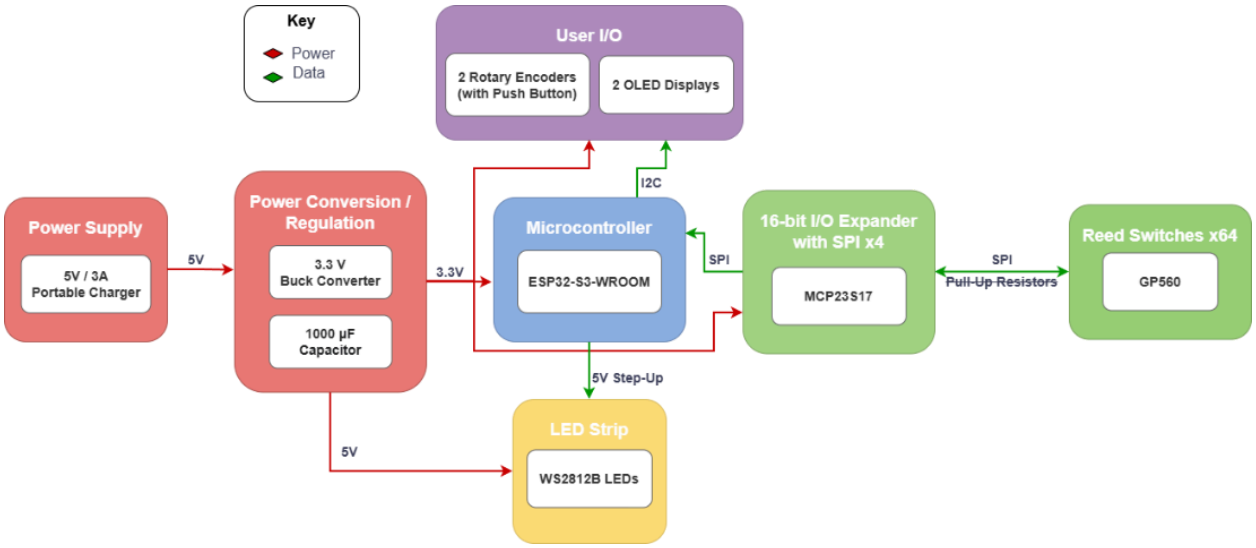

Figure 2: Block Diagram

Our block diagram, Figure 2, served as a good high-level representation of our design throughout the whole process. The power subsystem is described by the red blocks. There is one block for the supply and another for the conversion and regulation circuity. The user I/O subsystem is described by the purple block. This subsystem is integral to communicating with users. The microcontroller, which processes the software, is included in the blue block. The addressable LEDs on the LED strip are included in the yellow block. The sensor subsystem, which monitors the location of the chess pieces, is described by the green blocks.

Except for the exclusion of the pull-up resistors on the SPI connections, there were no big changes to this design. To add functionality while maintaining a smaller footprint, we pivoted from regular push buttons to push buttons that were integrated with a rotary encoder for one of the main devices that accepts user input. The lack of major changes reflects the straightforward nature of our design which allowed the development of our device to be executed at a high level.

The Requirements and Verifications Table, which outlines criteria to objectively evaluate each subsystem, is included in Appendix A. More detailed discussion of the design will reference this table.

2.2 Physical Design

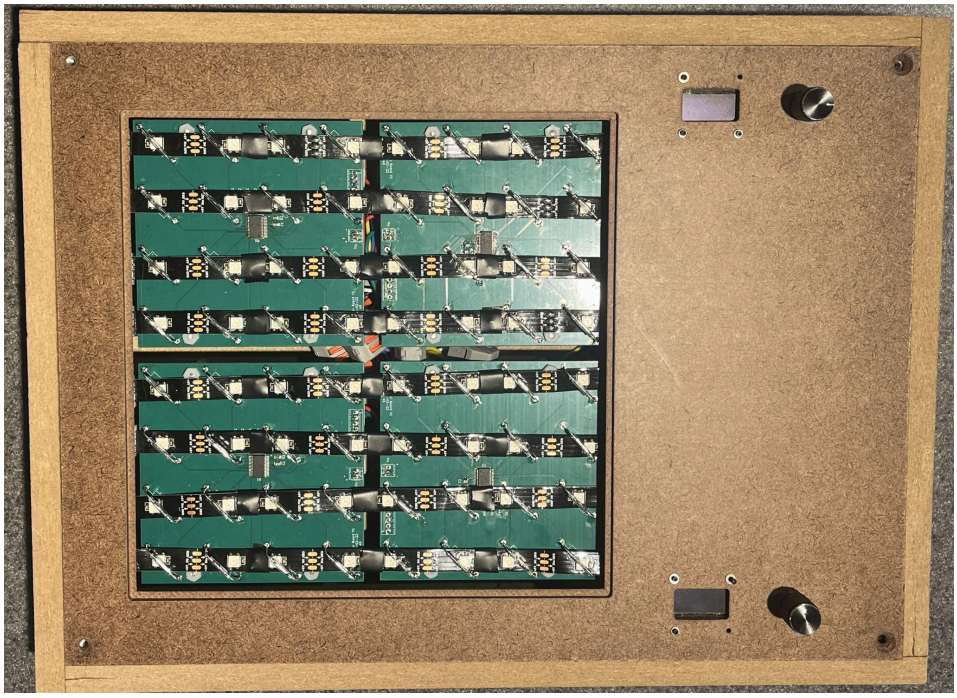

Figure 3: Physical Board Design

As shown in Figure 3, the chess board is composed of two sections. Beneath the playable area are the sensor boards which contain the reed switch sensors and GPIO expanders. Sections of the addressable LED strip are laid on top of the sensor boards with the rows daisy chained together. The reed switches on the board are arranged in a 4x4 grid. The reed switches are spaced and oriented in such a way that isolates one reed switch per square on the chessboard. The sensor boards are then also arranged in a 4x4 grid to create a chessboard with an 8x8 grid of squares. The design of the sensor board dictated the size of each square and the playable area. The playable surface that is placed on top is 3D printed in such a way to be both durable enough for play and thin enough to allow light from the LEDs to pass through. The underside of the playable surface is also sectioned by barriers so that an LED only illuminates the square directly above it.

The second section features a different PCB which contains the MCU and power subsystem. The main MCU board updates the software representation of the board state based on data received from the sensor board. This information is used to determine the output signals which illuminate the appropriate LEDs. The main board also connects to the user I/O subsystem which includes the rotary encoders with push buttons and the OLED displays. The main board PCB is supplied power by the portable power bank.

Both sections are contained within a rectangular housing made from MDF that was made in collaboration with the ECE Machine Shop. The top cover rests on standoffs that are also able to

accept screws to secure the cover. When the screws are removed, the top cover can be removed to access the internal hardware. The 3D printed playing surface rests on the groove in the top cover.

Schematic and CAD representations of the sensor and main MCU boards are included in Appendix B for more detailed viewing.

2.3 Software Design

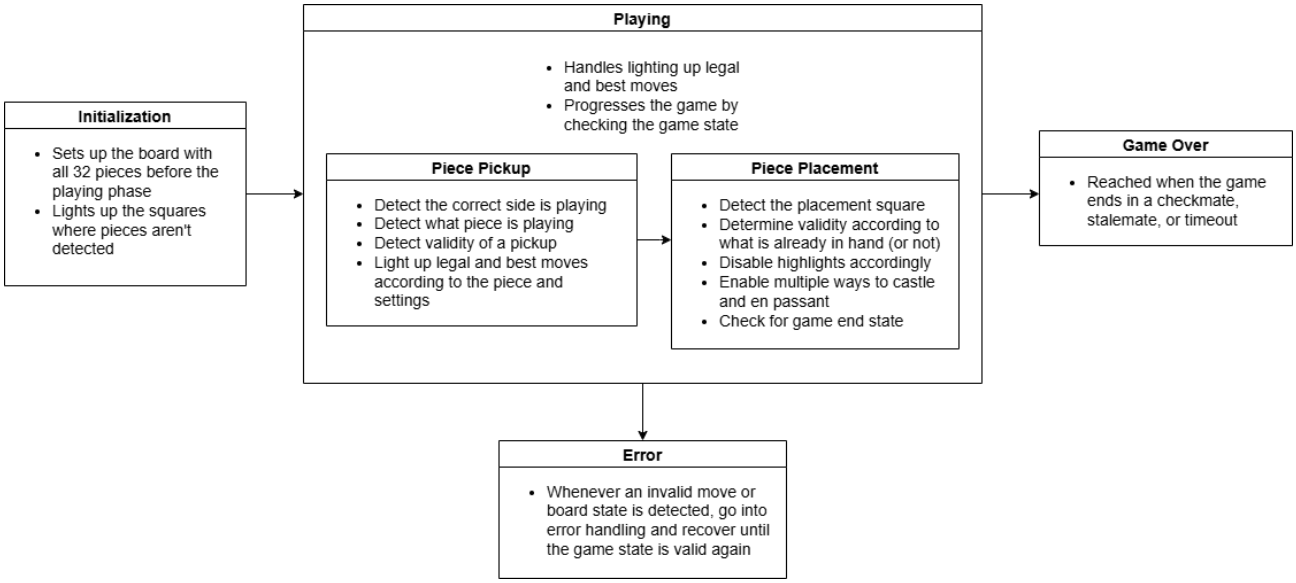

Figure 4: Software System State Diagram

As shown in Figure 4, the software is composed of four main states that define what mode of operation the system is in. The system starts in the main menu where the game time can be set from a minimum of 15 seconds to 1 hour. Once the time is confirmed and the game begins, the system shifts into the initialization phase. The initialization phase ensures piece locations match up with an expected chess board starting position.

The game then automatically moves into the playing phase where users can play chess normally, but with highlighting of legal moves and/or best moves enabled. If the board state is ever detected as having a mismatch with a valid position or an illegal move occurs, the system transitions into an error state which helps guide users of the board to return to the last valid position and then returns to the playing state. Inside the playing state, signals from the sensors can be interpreted as a change of either a piece pickup or placement. Piece pickups result in highlighting happening while placement results in an update to a new board position and disables highlighting. Both functions check for validity of an action to ensure there are no illegal moves or inconsistencies on the board. The software has exceptions built in for castling, en passant, and promotions. Promotions guide the user to the menu where the game pauses and the side that has

promoted a pawn can choose which piece (Queen, Rook, Bishop, Knight) to promote into. Once a piece placement happens and a new valid state is established, the system will check if the game has ended according to chess rules (checkmate, stalemate, insufficient material) and if so, move into the game end state which ends the game. There is separate continuous check for timeouts for both sides which can also end the game.

The user IO allows for a couple options to be adjusted. The highlighting of legal moves as well as best move suggestions can be toggled on and off. Additionally, the game can be reset at any time through the reset option which brings the user back to the main menu.

2.4 Power Supply and Conversion/Regulation

2.4.1 Description

To power our system we aimed for something both reliable and user friendly, eventually settling on using a power bank. We used a 5V 3A Camoner power bank with 12000mAh. This power bank was specifically designed to supply a stable 5V to power consumer products and it performs without issue. This power bank never ran out of power during testing even when plugged in for multiple hours. We estimate that the power bank could power our system for 6 hours. Since we wish to target players who are at a beginner level, this should be sufficient. The power bank can easily be swapped for one with full charge when the charge of one has been depleted.

A buck converter receives 5V from the power supply and steps it down to 3.3V. This voltage level is required by the OLED display, microcontroller, and GPIO expanders. The power supply also directly charges a 1000 uF capacitor. This capacitor provides power to the LED strip, protecting our microcontroller from sudden LED current draw.

2.4.2 Requirements and Verification

1. The power subsystem must reliably deliver stable 3.3V DC to the MCU and sensors.
   Result: This requirement was verified with the lab multimeter recording a stable voltage of around 3.334 V.

2. The power subsystem must reliably deliver stable 5V DC to the LED array.
   Result: This requirement was verified with the lab multimeter recording a stable voltage of around 5.1723 V.

3. The power subsystem must reliably supply power to all components (sensors, LEDs, MCU, OLED) for at least 1 hour of continuous operation.
   Result: We ran the test as described in Table 2 and got stable power the whole time. The power bank went from 96% to 78%, leaving plenty of power for continued play.

2.4.3 Design Alternatives

The decision to use a power bank worked well so we stuck with it for the duration of the project. On our first main MCU board, we were unable to get a stable 3.3V out of the buck converter. After ordering new buck converters and being more careful with the design on a second main MCU board design, we were able to get the correct voltage. The only design alternative we considered was to use DC batteries, but we decided on a power bank for its ability to detect and limit sudden current spikes from the LEDs as an extra measure to protect our components.

2.5 User I/O

2.5.1 Description

Along with the physical chess pieces, users interact with the chess board through rotary encoder buttons and two OLED displays. Each user is able to independently control options from a menu on their OLED screen, using the rotary encoder button for their side. The OLED response to user action is instantaneous. Players can turn off and on best moves and legal moves, reset the board, and set time control from the menu. Rotating the encoder allows users to navigate the menu, and pressing the button selects options. The rotary encoders are debounced in software.

2.5.2 Requirements and Verification

1. User input buttons must detect only one action per single physical button press (debounced).
   Result: After running the test as described in Table 2, we found that encoder input is never detected twice per action and button presses are also mostly okay. Our encoders are very bouncy so the button gets triggered on hold and release. If a user holds the button down longer than debounce time the button release is interpreted as a new press. Not a big issue as this is unlikely to happen during regular use.

2. The user interface must reflect user actions on the OLED display within 0.2 seconds.
   Result: After running the test as described in Table 2, we found that user action never took longer than 0.2 seconds.

2.5.3 Design Alternatives

The connection between the user IO components and the main PCB board proved to be quite fragile due to the lack of slack on the connecting wires. Additionally, the wiring configuration between the natively provided order of ports on IO components did not directly match up with configuration on the PCB ports. Fortunately the wiring configuration was easy to configure by simply paying attention to port/wire settings. The connection issues were resolved by replacing one of the encoder buttons.

2.6 Microcontroller

2.6.1 Description

The microcontroller, an ESP32-S3, uses sensor data to determine the location of the chess pieces and outline the state of the board. The microcontroller also stores this information and uses that data to determine all of the possible moves for a specific piece. If requested, it will also determine the best move based on the current state of the board. The microcontroller reacts to interrupts when the state of a reed switch changes or when the user sends input through the rotary encoder button. Depending on the type of interrupt the microcontroller will either update the board state and control the LEDs or update the OLED.

2.6.2 Requirements and Verification

1. The MCU logic must find all legal moves for a given chess position within 5 seconds.
   Result: Upon running the tests as described in Table 2, legal move highlighting was found to display under one second, well within requirements.

2. The MCU logic must find the best move for a given chess position within 5 seconds.
   Result: Upon running the tests as described in Table 2, best move highlighting was found to display under five seconds, as specified in code configuration (4.8 seconds).

3. The MCU logic must determine an invalid board state (e.g., illegal position, missing piece) and attempt a recovery operation within 5 seconds.
   Result: Upon running the tests as described in Table 2, invalid board states were detected under one second, well within requirements.

2.6.3 Design Alternatives

An initial design for the MCU board had problems properly converting power from 5V to 3.3V for the MCU and most components on the board, other than the LEDs. The PCB was revised to use thicker copper layers, 2oz instead of 1oz, and the whole PCB design was overhauled to simplify routing, improve heat dissipation, and provide better power conversion. New components were also used for the power conversion circuit.

2.7 LED System

2.7.1 Description

An addressable LED strip, divided into eight segments, is placed under the board. Each segment is divided further into eight smaller sections, one for each square in the rank. Each LED receives signals from the microcontroller that determine when the LED turns on and the color that the LED displays when it is on. The color of the LED conveys different kinds of information that are relevant to chess such as normal moves, capture moves, and best move.

2.7.2 Requirements and Verification

1. Each LED in the 64-LED array can be lit up individually without affecting adjacent LEDs.
   Result: Following the tests as described in Table 2, all 64 LEDs under each respective square could be lit up individually.

2. The LED array will light up with different colors to distinguish between legal moves, capture moves, and the best move.
   Result: Following the tests as described in Table 2, all LEDs could display correctly designated colors as programmed.

2.7.3 Design Alternatives

An issue that was encountered was that the spacing of LEDs on the addressable strip were not desirable for the board's square spacing. The fix implemented was to cut apart the LED strip, shorten strip spacing in between LEDs and reconnect the smaller strips by stripping away the plastic on top to reveal the copper pads and soldering together the smaller strips to achieve the desired spacing. Even though spacing on these modified strips were not exact to the spacing of the squares, the design of the 3D printed board cover included a grid protruding downwards to isolate the light projected by LEDs to only the square it is supposed to illuminate.

2.8 Sensor System

2.8.1 Description

Reed switches will detect when a chess piece has been placed on or removed from a specific square by reacting to magnets embedded in the pieces. There will be 64 reed switches, one for each square on a chess board. Every reed switch will be connected to a GPIO expander. This will allow information about piece location to be sent from the reed switches to the microcontroller.

Four GPIO expanders are required because the chosen microcontroller cannot natively support receiving data from all 64 reed switches. One GPIO expander is connected to 16 reed switches. The GPIO expanders use SPI (Serial Peripheral Interface) to send sensor data from the reed switches to the microcontroller.

2.8.2 Requirements and Verification

1. Sensors must detect when a piece is on or off a particular square, accurately triggering only the sensor associated with that specific square.
   Result: We performed the test as described in Table 2 and found that piece pick up is always detected, piece placement is detected when the piece is centered in the square, and never observed triggering of adjacent squares.

2.8.3 Design Alternatives

The sensor boards we designed worked from the first PCB design with very few difficulties. Error on our end with some improperly soldered pins did occur, but the system itself was sound and did not require modification. For the sensor system, the only design modification we considered was initially using a switch matrix instead of IO expanders, but the limitations of detecting multiple piece movements and wiring complexity dissuaded us from this.

2.9 Tolerance Analysis

Reed Switches

One concern with the reed switches is ensuring reliable activation by the magnets in each chess piece while preventing unintended activation of adjacent switches. The selected reed switches (SW GP560/15-20 AT) require 15–20 Ampere-Turns to actuate, which can be approximated as 15–20 gauss [8].

We figured out with experimentation that the field of the magnets were strongest a little bit to the side of their center. To maximize the magnetic field in the center of the reed switch, we placed two magnets side by side under each chess piece. Magnetic field strength decays approximately with the inverse cube of distance, meaning it drops rapidly away from the magnets. Our magnet geometry concentrates the magnetic field in the center of the reed switch while minimizing fields at neighboring switches.

Based on data collected by K&J Magnetics, for N42 magnets of our approximate size, the activation threshold of approximately 20 Gauss is achieved within a distance of around 7 mm [10]. There is 10mm of space between the PCB surface and the chessboard surface, but the reed switches are raised several mm above the PCB surface so they fall within this range.
This behavior was validated experimentally using the final two-magnet configuration. No unintended activations of adjacent reed switches were observed, confirming that the design satisfies both activation and isolation requirements.

Power Consumption

According to the WS2812B datasheet and component documentation from Cirkit Designer [3], each WS2812B LED can draw up to about 60 mA at full brightness when all three color channels are driven at maximum drive current (approximately 20 mA per channel). having 1 LED per square, means 64 LEDs total. In the worst case: I_max = 64 * 60mA = 3.84A. This would exceed our power bank max output of 3A from just the LEDs alone. However, based on the use case of lighting up legal moves, we would light up at most 28 LEDs at once if we placed a queen in the middle of the board. This results in a max current draw of 28 * 60mA = 1.68A. So in our worst case scenario, we would be well under the specifications of the power bank.

The power bank is specified at 12000mAh. The specifications says this is rated for 5V, but we can assume it might be labeled wrong and is actually 3.7V. Under this assumption, actual power output is 12Ah * 3.7V = 44.4Wh (Energy = Capacity * Voltage). We can assume that the internal converter is not 100% efficient and assume a conversion efficiency of 80% based on estimates used by TexasInstruments [4], and take the usable power at 44.4Wh * 80% = 35.52Wh at 5V output.

The LEDs will draw at max 840mA at 5V for 5V * 1.68mA = 8.4W. According to datasheets [6], the ESP32-S3 with Wi-Fi and Bluetooth disabled, actual active CPU current will be lower than the listed RF-receive current 91mA. This means for the MCU, power draw is 91mA * 3.3V (input voltage) = .3W. The 2 OLEDs draw 28mA each at max power draw. With a 3.3V power line, this means 3.3*2*28 = just .185W during normal operations according to the datasheet [5]. The GPIO expanders datasheets show that they draw 1mA when active [7]. Multiply with the input voltage of 3.3V for 3.3mW of power draw. Added together, 8.4W + .3W + .185W + .0033W = 8.8883W. To get the operating time, we divide the watt hours by the watts used: 35.52Wh/8.8883W. = 4 hours of operation. This should be sufficient to reach our goal of one hour of continuous operation.

3 Cost and Schedule

3.1 Cost Analysis

A table with an itemized list of parts, Table 3, is included in Appendix C. The total cost of those parts is $173.04.

According to the post graduation section of the Grainger college of engineering Electrical Engineering page [11], the average starting salary for a fresh graduate is $90,115. If we assume 50 working weeks at 40 hours per week, then that is 2000 hours for the whole year. This calculates out to an hourly rate of $45.06. We anticipate that every week, we will all work for 6 hours a week. We have about 10 weeks to work on the project. In total for the labor cost: 6hrs/week * 10 weeks * $45.06 per hour * 3 members = $8110.80.

For our board, which is approximately 15 in by 11 in according to Figure 3, we only need a simple housing to hold electronic components and hooks for the led strip. This should take the machine shop about 5 hours.

The labor cost along with the total cost of everything in our itemized list is $8283.84.

3.2 Schedule

Week	Task	Person
5	First talk with machine shop to establish our project and goals	Matthew, Tim, JJ
	Research implementation details to prepare for design document	Matthew, Tim, JJ
6	Finish first design of main PCB	JJ
	Finish design of sensor PCB	Matthew
	Turn in Design Document	Tim, Matthew
	Source components and compile list	Tim, Matthew
7	Present Design Document	Tim, Matthew, JJ
	Finalize main PCB, optimizing size and adding all headers	JJ
	Put together breadboard demo of MCU + 16 reed switches	Tim, JJ

	Write software to run breadboard demo	Tim, Matthew
8	Breadboard Demo	Tim, Matthew, JJ
	Get final design and revisions to machine shop	Matthew Trela
	Test breadboard demo, settle on magnet size, and order new magnets	Tim, Matthew, JJ
	Teamwork Evaluation I (3/11)	Tim, Matthew, JJ
	Must finish and order main PCB 2	JJ
	Solder and test sensor board	Tim
	Resolder LED strips	Matthew
9	Last week for PCB orders, so thoroughly test both boards	Tim, Matthew, JJ
	Order missing components that we don't have yet	Tim
	Solder Components on boards	JJ, Tim
	Design and 3D print playing surface	Matthew
10	Solder Components on boards	JJ, Tim
	Write software (chess engine, error detection, display)	Matthew, Tim
	Design and 3D print chess pieces	Matthew
11	Solder Components on boards	JJ
	Write software (chess engine, error detection, display)	Matthew, Tim
	Progress Demo	Tim, Matthew, JJ
	Team Contract assessment	Tim, Matthew, JJ
12	Resolve software bugs and add features	Tim, Matthew

	Test main board	JJ
13	Add engine implementation and finish UI design.	Tim, Matthew
	Mock Demo and Presentation	Tim, Matthew, JJ
14	Final Demo and Presentation	Tim, Matthew, JJ
15	Final Paper	Tim, Matthew, JJ

Table 1: Schedule for Project Timeline

4 Conclusion

4.1 Accomplishments

Overall, our move displaying chess board is a resounding success. All three high level requirements were met and we were able to accomplish all functionality. The chess board is able to detect piece pickup and placement on every square and maintain a running internal representation of the board. For both sides legal moves are displayed within 1 second. Legal moves are represented with blue and orange lights, blue for non-capture and orange for capture. The chess board also has a fully functional chess engine which displays best moves in under 5 seconds, indicated by a cyan led on that square. Our system has resilient error detection. If any amount of pieces are taken off the board or rearranged in any order, our system automatically enters error recovery and the game will not resume until the board surface matches the last safe state. In our testing, every error state was able to be recovered from. The system has multiple hours of battery life and does not require reflashing or manual intervention to keep it running.

Additionally, all subsystems are fully functional. The LEDs are clearly visible for every square. The user input is properly debounced, and user action is reflected on the OLED screens promptly. The users are able to change settings independently using the buttons, toggling legal move display and best move display for both users. Finally, the one feature implemented beyond our project scope is automatic time management. At the beginning of the game, the users agree on a time control and then the time remaining for each user is automatically calculated and updated when moves are played. If a player runs out of time, they lose on time.

4.2 Uncertainties

Since the chess board was fully functional, according to the criteria that we outlined, there are minimal uncertainties with the device itself. Looking forward to the viability of this product on the wider market, there is a little bit of uncertainty regarding the price that this device could be offered at. To differentiate our work from what is already available, we wanted to make something that was significantly cheaper. Based on the itemized list of components in Table 3, we were able to put together something that was cheaper than the other products that we researched.

As part of this class, however, there were things that we did not need to account for financially. We factored in an estimation of the cost of the labor that both our group and the ECE Machine Shop provided but we did not factor in other hidden costs, like manufacturing and shipping costs, that would also drive up the cost. The price would also likely increase to ensure that the product is profitable enough to sustain a business. In theory, with a large enough number of units being produced and sold, the cost of things like components, manufacturing, and labor would decrease when viewed from a per unit basis. This would hopefully allow the device to still be offered at a competitive price point. Since it falls outside the scope of this class, we did not do a deeper analysis of the financial side of this project so there is some uncertainty there.

4.3 Ethical Considerations

This smart chess board project uses embedded electronics and software to assist users without collecting personal data. Ethical considerations are addressed through adherence to the ACM and IEEE Codes.

Per the ACM Code of Ethics [2], the system avoids harm (ACM Section 1.2) and is presented honestly: move recommendations are clearly disclosed as advisory, not guaranteed optimal, consistent with ACM Section 1.3. Risks such as sensor failure or misleading feedback are evaluated (ACM Section 2.5) and mitigated via redundant sensing and error states.

Under the IEEE Code of Ethics [1], the design minimizes electrical and mechanical hazards (IEEE Principle 1). IEEE Principle 3 guides truthful reporting of system accuracy and limitations in all documentation.

No personal data collection occurs, avoiding privacy concerns. Third-party software complies with licenses and is properly cited, respecting intellectual property (ACM Section 1.5).

4.4 Future Work

Several enhancements would improve the functionality and completeness of the smart chess board. First, critical game-over conditions are currently missing, including threefold repetition and the fifty-move rule for draws; implementing these would be a natural next step. Additionally, an undo feature and a more powerful chess engine that makes better use of the available hardware to compute more optimal moves more quickly would be necessary for a polished product.

On the physical side, a power button was overlooked. Adding a charging port, so the board does not need to be opened to recharge the power bank, along with visible battery indicators on the OLED display would improve usability. The overall board dimensions could also be increased to more closely match a full-sized board, which was not feasible in this project due to PCB size limitations. A larger form factor would allow for wider square spacing and could include a storage compartment beneath the board for pieces when not in use.

5 References

[1] Institute of Electrical and Electronics Engineers, "IEEE Code of Ethics," Jun. 2020. Available: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-code-of-ethics.pdf

[2] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct," *Association for Computing Machinery*, Jun. 22, 2018. https://www.acm.org/code-of-ethics

[3] Cirkit Design, "How to Use the Addressable LED Pixel Board WS2812B," *Cirkitdesigner.com*, 2024. https://docs.cirkitdesigner.com/component/b01f6324-ce84-4ebb-8e99-3f34ad7ca280/addressable-led-pixel-board-ws2812b (accessed Feb. 13, 2026).

[4] B. Hauke, "Basic Calculation of a Boost Converter's Power Stage." Accessed: Feb. 13, 2026. [Online]. Available: https://www.ti.com/lit/an/slva372d/slva372d.pdf?ts=1770979377734&ref_url=https%253A%252F%252Fwww.google.com%252F

[5] 好好搭搭论坛, "扩展库使用说明——SSD1306" https://haohaodada.com/new/bbs/forum.php?mod=viewthread&action=printable&tid=639 (accessed Feb. 13, 2026).

[6] "Espressif Documentation," *Espressif.com*, 2025. https://documentation.espressif.com/esp32-s3_datasheet_en.pdf

[7] Microchip, "MCP23017/MCP23S17," *https://ww1.microchip.com/downloads/aemDocuments/documents/APID/ProductDocuments/DataSheets/MCP23017-Data-Sheet-DS20001952.pdf*.

[8] LittleFuse, "Application Note: Ampere*turn versus mT and Gauss." https://www.littelfuse.com/assetdocs/ampere-turn-versus-mt-and-gauss-application-note?assetguid=5ff44b03-b3a7-4818-8b73-f79fbf75dbf7 (accessed Feb. 13, 2026).

[9] Bunting, "Magnet Applications Technical Datasheet." Accessed: Feb. 13, 2026. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/7182/N42P062062.pdf

[10] K&J Magnetics, "Magnetic Field Calculator," *Kjmagnetics.com*, 2024. https://www.kjmagnetics.com/magnetic-field-calculator.asp

[11]Grainger Engineering Office of Marketing and Communications, "Electrical Engineering," grainger.illinois.edu, 2026. https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/electrical-engineering

Appendix A: Requirements and Verifications Table

Requirement	Verification
The power subsystem must reliably deliver stable 3.3V DC to the MCU and sensors.	**1. Connect a multimeter to the 3.3V output rail of the power subsystem. Ensure the board is in an idle state. Record the voltage reading. Confirm the voltage is 3.3V (±0.15V).** **2. While the system is idle, use an oscilloscope to measure the ripple on the 3.3V rail. Confirm the peak-to-peak voltage is less than 50mV.**
The power subsystem must reliably deliver stable 5V DC to the LED array.	**1. Connect a multimeter to the 5V output rail of the power subsystem. Ensure the board is in an idle state. Record the voltage reading. Confirm the voltage is 5.0V (±0.25V).** **2. Command half the LEDs to light up at maximum brightness. While the LEDs are active, measure the voltage at the 5V rail. Confirm the voltage remains above 4.75V.**
The power subsystem must reliably supply power to all components (sensors, LEDs, MCU, OLED) for at least 1 hour of continuous operation.	**1. Ensure the system is powered solely by the designed power source. Record the start time.** **2. Operate the board in a typical usage scenario (e.g., playing a game, requiring sensor scans and LED updates) for 60 minutes.** **3. After 60 minutes, verify all subsystems remain operational. Perform a sensor scan and test each LED. Confirm the MCU does not reset and the LEDs light up successfully.**
Each LED in the 64-LED array can be lit up individually without affecting adjacent LEDs.	**1. Ensure the board is powered and in an idle state.** **2. Using a test script on the MCU, command a single, specific LED to turn on. Visually confirm only the intended LED is lit.** **3. Repeat step 2 for at least seven other LEDs in different quadrants of the board and each LED strip per row. Visually**

	confirm only the intended LED is lit each time.
The LED array will light up with different colors to distinguish between legal moves, capture moves, and the best move.	**1. Set up a known chess position on the physical board. 2. Command the MCU to calculate and display legal moves for a selected piece. Record the color displayed. Confirm the color is distinct. 3. Command the MCU to calculate and display capture moves for a selected piece. Record the color displayed. Confirm the color is distinct from the legal move color. 4. Command the MCU to calculate and display the best move. Record the color displayed. Confirm the color is distinct from both legal and capture move colors.**
The MCU logic must find all legal moves for a given chess position within 5 seconds.	**1. Ensure the board is powered and the MCU is running the chess engine. 2. Achieve a standard mid game chess position on the board (via debug loading or playing on board). Start a timer. 3. Command the MCU to calculate legal moves for the side to move. 4. Record the time displayed when the MCU outputs "Calculation Complete." Confirm the elapsed time is less than 5 seconds. 5. Verify the list of legal moves against a known chess engine to ensure accuracy.**
The MCU logic must find the best move for a given chess position within 5 seconds.	**1. Ensure the board is powered and the MCU is running the chess engine. 2. Load/achieve a standard mid-game chess position onto the board. Start a timer. 3. Command the MCU to calculate the best move for the side to move (using its evaluation function). 4. Record the time displayed when the MCU outputs the move. Confirm the elapsed time is less than 5 seconds. 5. Verify the move is a reasonable candidate (no blunders) against a known chess engine.**

The MCU logic must determine an invalid board state (e.g., illegal position, missing piece) and attempt a recovery operation within 5 seconds.	**1. Ensure the board is powered and a valid game is in progress. 2. Simulate an invalid state by removing a piece from the board while it is the opponent's turn, creating an illegal position. Start a timer. 3. Observe the system response. Record the time when the MCU flags an "Invalid State" or initiates a recovery (prompt user to replace the piece via OLED). 4. Confirm the error is detected and the recovery prompt is displayed within 5 seconds of the piece being removed.**
Sensors must detect when a piece is on or off a particular square, accurately triggering only the sensor associated with that specific square.	**1. Ensure the board is powered and in an idle state with no pieces on the board. Record the 64-bit sensor state via serial debugging. Confirm all values read as "0" (no piece). 2. Place a piece with a magnet in the center of square e4. Record the sensor state via serial debugging. Confirm the bit for square e4 reads as "1" and all other 63 bits read as "0". 3. Repeat step 2 for squares on the edge (a1) and a corner (h8). Confirm only the intended sensor triggers each time. 4. Place pieces on two adjacent squares (d4 and d5) simultaneously. Record the sensor state. Confirm both corresponding bits read as "1" and that no other sensors are triggered by magnetic interference.**
User input buttons must detect only one action per single physical button press (debounced).	**1. Ensure the board is powered and the MCU is logging button presses to the serial console. 2. Press the "Select" button once firmly and release it within 0.5 seconds. 3. Observe the serial output. Confirm that only one button press event is registered. 4. Repeat steps 2-3 for the "Left," "Right," and "Back" buttons. Confirm each registers only one event per press.**
The user interface must reflect user actions on the OLED display within 0.2 seconds.	**1. Ensure the board is powered and the OLED is displaying the main menu.**

	2. Press a button to navigate the menu. Start a timer simultaneously. **3. Record the time when the OLED display updates to highlight the next menu item.** **4. Confirm the elapsed time between the button press and the display update is less than 0.2 seconds.**

Table 2: Requirements and Verification of Subsystems

Appendix B: Subsystem Schematics and PCB Designs

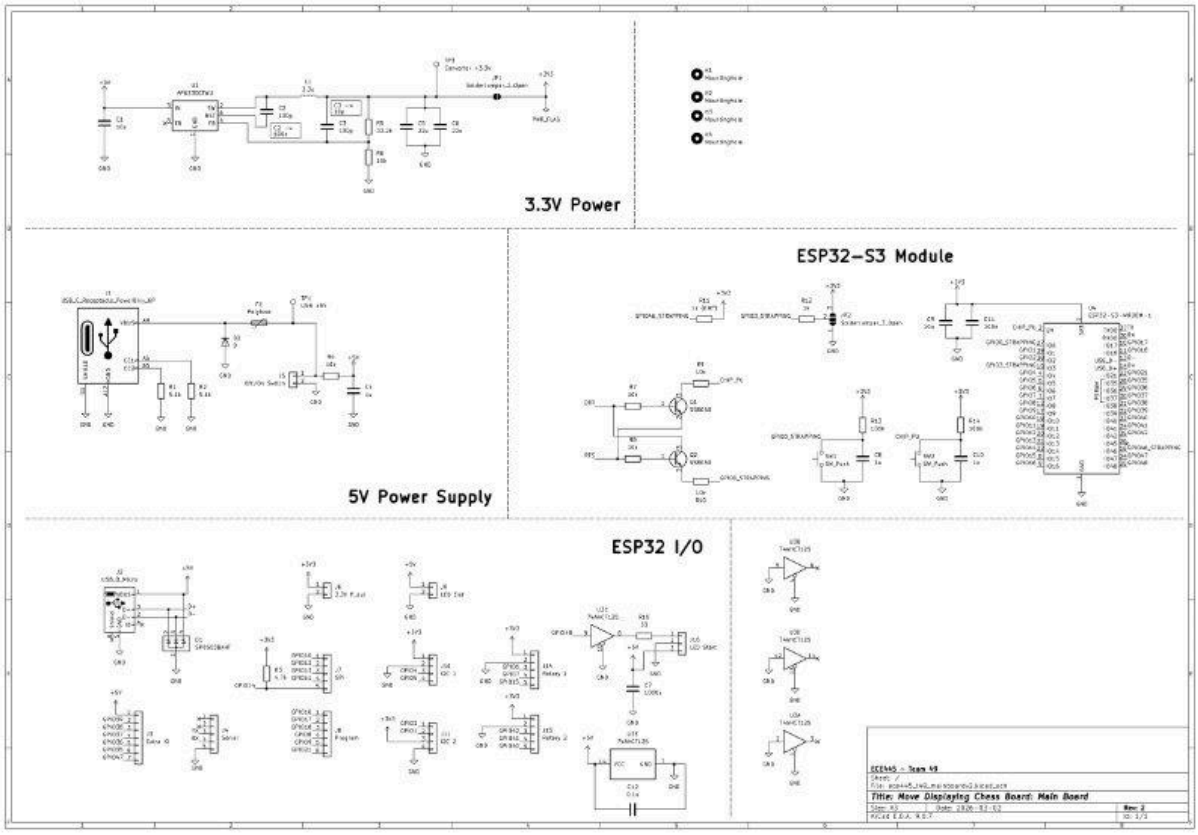

Figure 5: Schematic of main MCU board PCB

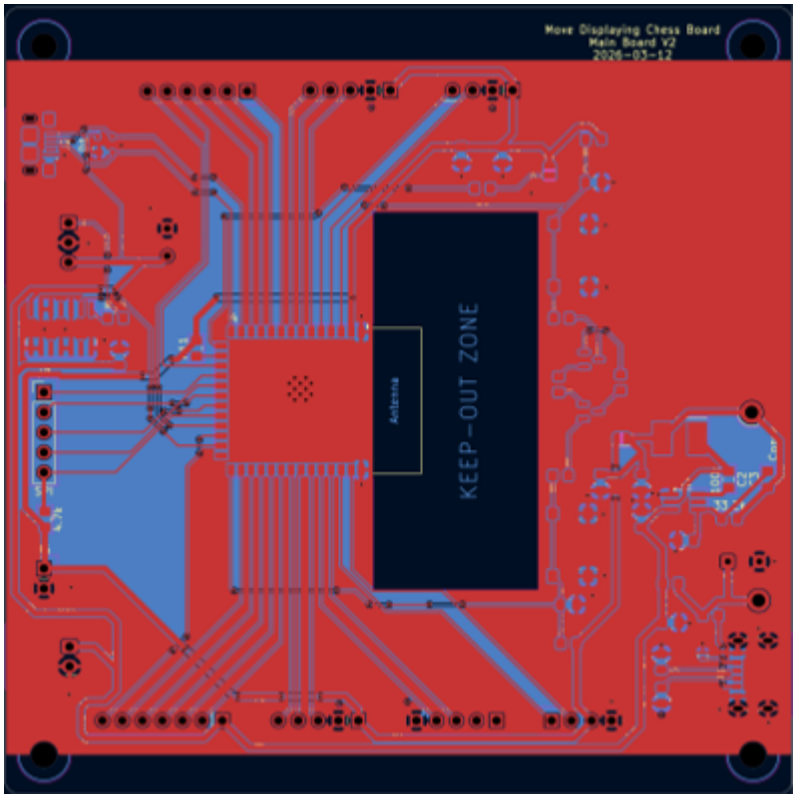

Figure 6: Layout of main MCU board PCB

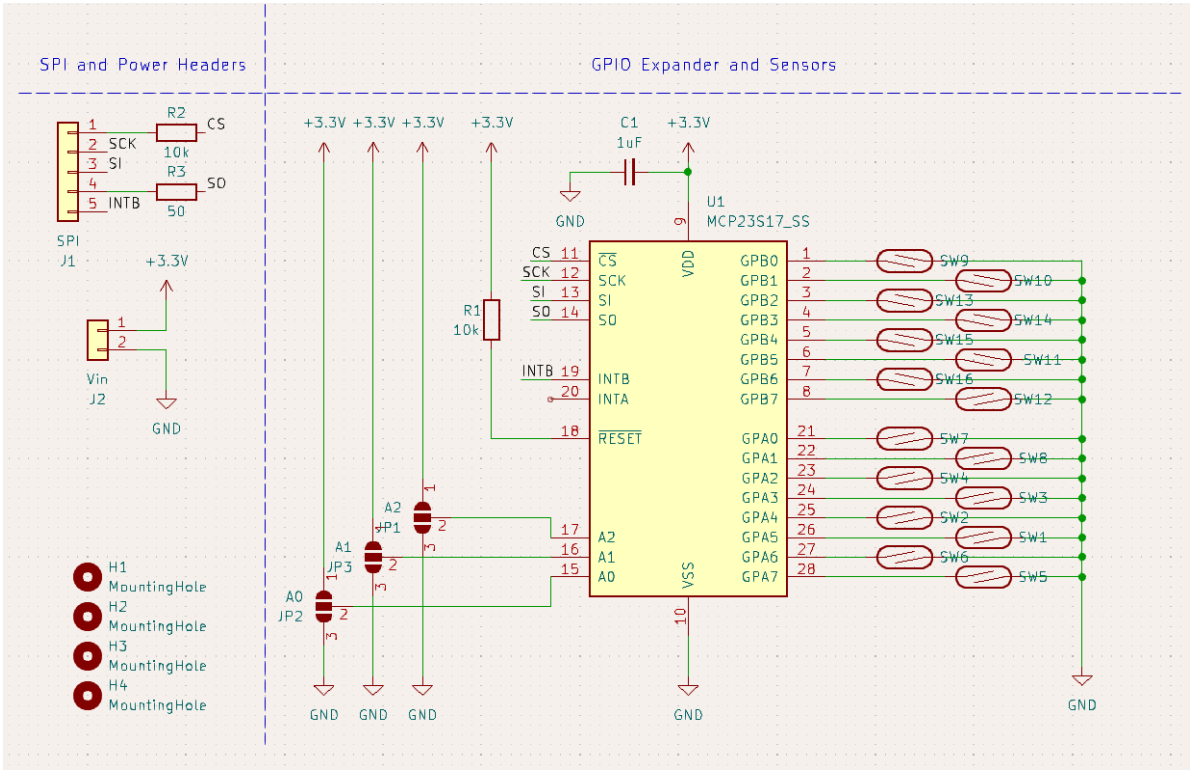

Figure 7: Schematic of sensor board PCB

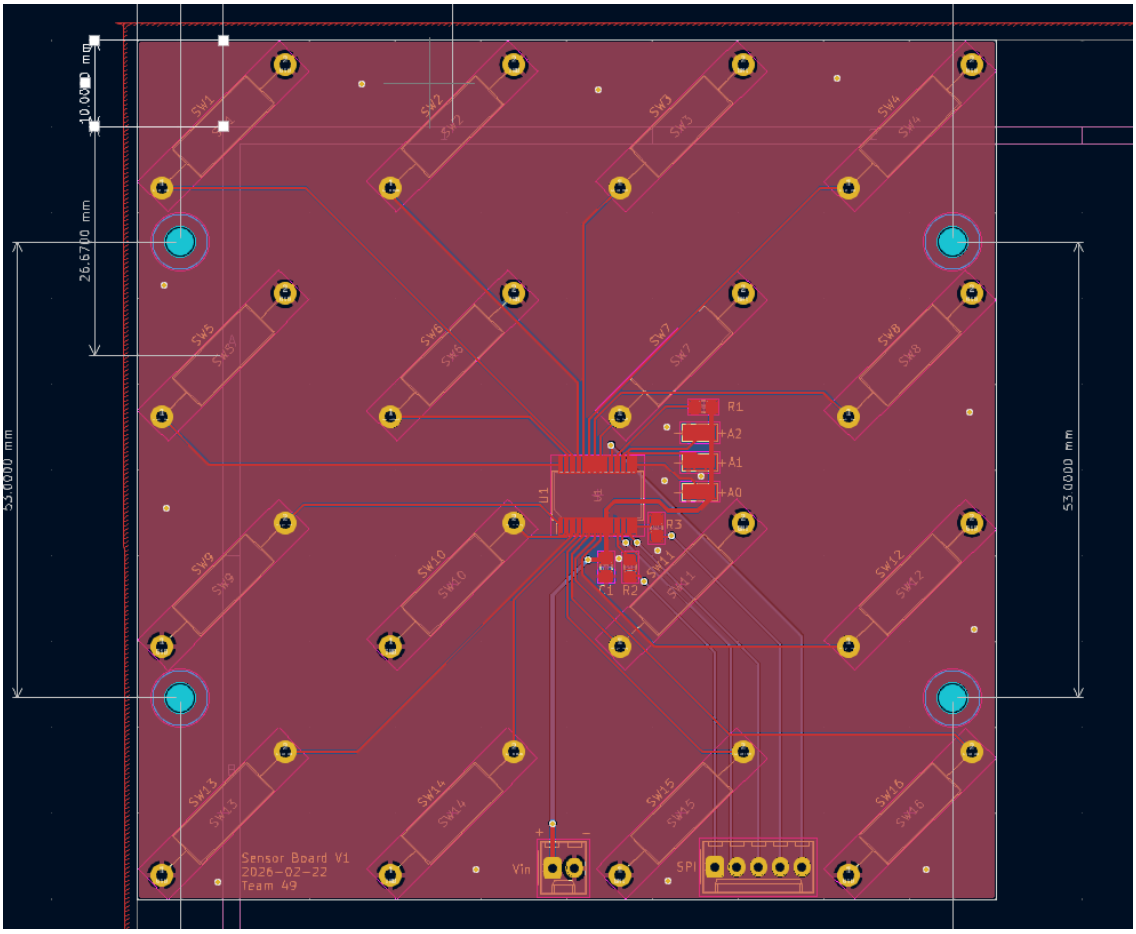

Figure 8: Layout of sensor board PCB

Appendix C: Itemized Cost Table

Description	Manufacturer	Quantity	Price ($)	Link
SW GP560/15-20 AT Reed Switches	Standex-Meder Electronics	64	31.41	Link
MCP23S17-E/SP-ND I/O expander 16 inputs	Microchip Technology	4	7.12	Link
WS2812B ECO LED Strip Light 16.4 FT	SEZO	1	13.99	Link
FIT08613Pin LED Strip Connector	DFRobot	8	7.92	Link
ESP32-WROOM-32-N4 MCU	Espressif Systems	1	6.56	Link

SSD1306 128X64 OLED Screen	UCTronics	2	13.98	Link
KY-040 Rotary Encoders	JTAREA	2	7.99	Link
Small Neodymium Magnets	rhinocats	128	12.99	Link
Portable Charger 12000mAh	Camoner	1	19.99	Link
Right Angle USB C Male to USB C Female Adapter	GHQXZT	1	5.99	Link
Lever Wire Electrical Connectors Set	XHF	1	9.99	Link
AEC1000M35V1320 1000 µF 35 V Aluminum Electrolytic Capacitor	Lumimax Optoelectronic Technology	1	0.33	Link
RNCP0805FTD10K0 10k ohm resistor	Stackpole Electronics Inc	8	0.80	Link
CRCW080550R0FKTA 50 ohm resistor	Vishay Dale	4	0.84	Link
1276-1066-2-ND 1uF capacitor	Samsung Electro-Mechanics	4	0.40	Link
WM4114-ND 5 pin header (2.54mm) 1x5	Molex	8	3.76	Link
A1921-ND 2 pin header (2.54mm)	TE Connectivity AMP Connectors	4	0.56	Link
5 pin wire (2.54mm) and 2 pin wire (2.54mm)	elechawk	1	15.99	Link
CL21A106KOQNNNE 10uF 16V capacitor 0805	Samsung Electro-Mechanics	2	0.20	Link
CL21B105KAFNNNE 1uF 25V capacitor 0805	Samsung Electro-Mechanics	2	0.20	Link
CL21A226MQQNNNE 22uF 6.3 V capacitor 0805	Samsung Electro-Mechanics	2	0.26	Link
KGM21NR71H104KT 0.1uF 50V capacitor 0805	KYOCERA AVX	1	0.08	Link
SP0503BAHTG 8.5V Tvs Diode	LittleFuse Inc.	1	0.53	Link

PR20B05VBDN 1.27mm pitch 2x5 pin header vertical through hole	METZ CONNECT USA Inc.	1	0.39	Link
USB4125-GF-A USB-C Receptacle Connector 24 (6+18 Dummy) - Power Only 6 Pin Position Surface Mount, Right Angle; Through Hole	GCT	1	0.59	Link
575-4 Banana Jack Connector	Keystone Electronics	4	3.76	Link
MLZ2012M3R3HT000 3.3 µH Inductor 500 mA 200mOhm 0805	TDK Corporation	2	0.20	Link
10118194-0011LF USB - micro B USB 2.0 Receptacle Connector	Amphenol ICC (FCI)	1	0.49	Link
826658-5 Connector Header Through Hole 1x10 2.54mm	TE Connectivity AMP Connectors	1	1.81	Link
SS8050 BJT 25V 1.5A SOT-23	Shenzhen Slkormicro Semicon Co., Ltd.	2	0.20	Link
RC0805FR-071KL 1k Ohm resistor 0805	YAGEO	2	0.20	Link
RMCF0805JT5K10 5.1k Ohm resistor 0805	Stackpole Electronics Inc	2	0.20	Link
CRCW080510K0FKEA 10k Ohm resistor 0805	Vishay Dale	5	0.50	Link
CR0805-FX-3322ELF 33.2k Ohm resistor 0805	Bourns Inc.	1	0.10	Link
RC0805FR-07100KL 100k Ohm resistor 0805	YAGEO	2	0.20	Link
B3S-1000 Tactile Switch SPST-NO	Omron Electronics Inc-EMC Div	2	0.86	Link
36-5010-ND PC TEST POINT RED	Keystone Electronics	4	1.36	Link
AP62300TWU-7 Buck Switching Regulator IC 3A	Diodes Incorporated	1	0.30	Link

Table 3: Itemized List of Costs