Pathfinder Chessboard

"Light the Path"

University of Central Florida

Group 35 Authors:

Andres Armendariz	Freddy Pacheco	Ryan Ramdihal
Electrical Engineering	*Computer Engineering*	*Computer Engineering*

Mentor: Dr.Sundaresh

Reviewers:

Dr. Qifeng Li
Dr. Kalpathy Sundaram
Dr. Azadeh Vosoughi
Dr.Chris Iannello

Table of Contents

Chapter 1: Executive Summary

The Pathfinder Chessboard is a smart embedded system designed to modernize physical chess without replacing it with a screen-based alternative. While online digital platforms have made chess more approachable through automated move validation, they sacrifice the tactile experience and social engagement of playing on a physical board. The Pathfinder Chessboard addresses this gap by integrating embedded electronics into a traditional chessboard, enabling it to communicate rules, validate gameplay, and provide interactive feedback while preserving the physical nature of a tabletop play style.

The primary objective of the Pathfinder Chessboard is to create a fully functional smart chessboard capable of recognizing piece positions, guiding legal moves through LED illumination, and enforcing gameplay rules in real time. The system will utilize an array of sensors embedded beneath each of the 64 squares to detect chess pieces, paired with individually addressable LEDs to indicate valid movement options. An ESP32 microcontroller serves as the central processing unit (CPU), continuously emulating the internal game state and synchronizing it with user actions on the board. The design further incorporates mode-selection allowing players to potentially switch from playing locally, to a speed chess gamemode, or to playing against an AI opponent.

The project establishes both baseline and advanced performance requirements. At minimum, the chessboard must reliably detect all piece movements, illuminate valid moves within an acceptable response time, and sustain continuous operation for at least 2 hours battery powered. A custom printed circuit board (PCB) will be developed to integrate the microcontroller, power regulation, LED drivers, and sensor interfaces into a compact and manufacturable design. Advanced goals include integration of a chess engine such as Stockfish for autonomous computer play, an implementation of a timed mode for speed chess, and audio or visual cues for move confirmation.

Functionally, the Pathfinder Chessboard enhances accessibility for beginners by reducing rule ambiguity and teaching through interaction rather than instruction. Simultaneously, it improves utility for experienced players through automated validation and a fast-paced game mode. This project distinguishes itself through its combination of real-time rule guidance and multi-mode operation.

Upon completion, the Pathfinder Chessboard is expected to provide a modernized yet authentic chess experience that preserves physical gameplay while offering the intelligence and adaptability of digital systems. By blending embedded design with human-centered interaction, the project demonstrates how traditional games can be reimagined for broader accessibility, educational value, and lasting engagement.

Chapter 2: Project Description

2.1 Motivation and Background

Chess is widely considered the most popular board game of all time. Originating around 600CE, it has long been regarded as a symbol of strategy and intelligence. Whether introduced through family and friends, pop culture or online media almost everyone in the world has at least heard about chess. Yet despite its popularity, learning how to play can be intimidating. Beginners often feel overwhelmed when learning rules, openings, and strategies and as a result often quit. Fortunately with the growth of digital platforms and artificial intelligence the opportunities to learn and experience chess have never been as accessible, interactive, or engaging than ever before.

However, many digital platforms sacrifice the immersive experience of playing on a physical board. Many players still value the traditional feel of moving pieces by hand but want the added benefits of digital assistance and connectivity.

The Pathfinder Chessboard bridges this gap by combining the physical experience of a traditional chess game with the functionality of an online platform. Our design consists of LED indicators embedded within the chessboard to guide players through valid moves and capture opportunities, making the game more accessible to beginners while still engaging for experienced players. This interactive feedback reduces the steep learning curve for beginners, encouraging skill development through visualization for potential move sets.

In addition to local two-player games, the integration of a computer AI and remote play offers versatility that extends the chessboard's appeal. The AI opponent provides a practice environment for solo players, while online connectivity via Wi-Fi enables real time matches with friends or competitors from anywhere. These features enhance the board's usability beyond recreational play, making it suitable for chess clubs, classrooms, and home use.

The Pathfinder Chessboard presents an intricate design requiring coordination of sensors for piece detection, LEDs for move guidance, wireless communication for online play, and custom PCB design for efficient power and signal management, showcases innovation by integrating traditional gameplay with modern interactive features.

2.2 Goals and Objectives

The primary goals of the PathFinder Chessboard are to create a fully functional physical chessboard equipped with accurate piece detection and LED guidance. The system should allow

smooth local two-player gameplay, providing real-time feedback to highlight legal moves and capture opportunities. Another key goal at this stage is to design and implement a custom PCB that integrates the microcontroller, power regulation, sensor interfaces, and LED drivers into one cohesive system.

To meet these goals, our basic objectives are to achieve accurate detection rates for piece movement, ensure that LEDs illuminate the correct legal moves and capture squares within a timely manner, and demonstrate the stable gameplay for the duration of a full match without any subsystem failures. The chessboard must operate reliably in local two-player mode with all rules enforced, and the custom PCB should provide continuous power and coordination for all subsystems for at least 2 hours of battery-powered play.

Building upon the basic goals, the advanced goals focus on enhancing gameplay through additional features that improve user interaction and create variety. These goals include integrating a chess engine such as Stockfish to provide a computer opponent and introducing a speed chess mode complete with a timer and sound effects for move notifications.

The advanced objectives are to run the chess engine efficiently on the embedded platform, generating valid AI moves and integrating them to appear through the LED system. For speed chess mode, the objectives are to implement a timer within per move timeframe and to generate audio cues that alert players when a move is made or when time is running low.

The stretch goals for the PathFinder Chessboard emphasize expanded connectivity and user experience enhancements. These include enabling remote multiplayer matches by synchronizing moves through a companion app, allowing a player to play against another remotely. Additional goals include incorporating customizable LED animations and a win-loss record display.

Corresponding stretch objectives include developing a companion app that can transmit moves between two PathFinder Chessboards or between a PathFinder Chessboard and a Wi-Fi enabled device, ensuring that gameplay remains synchronized across locations. The system should maintain minimal lag during move updates to provide a seamless remote play experience. Other objectives include creating dynamic LED animation patterns to highlight moves or outcomes and integrating a win–loss record feature that updates automatically at the conclusion of each game. These stretch objectives would elevate the PathFinder Chessboard beyond local play, blending traditional physical chess with modern digital connectivity.

2.3 Features and Functionalities

The PathFinder Chessboard is designed to combine the tactile experience of traditional chess with interactive benefits of modern technology. A central feature of the system is its LED-guided move indication, where embedded LEDs beneath each square highlight legal moves when a piece is lifted from the board. To improve clarity, valid moves will be displayed in one color, while capture opportunities will be distinguished by a different color. This guidance system enhances accessibility for beginners, reduces errors in gameplay, and creates a more engaging experience for all players.

The board supports three primary modes of play. First, a local two-player mode allows for traditional in-person matches. Enhanced with real-time LED guidance. Second, a computer opponent mode integrates a chess engine, such as Stockfish, enabling solo play. Third, a remote play mode uses Wi-Fi connectivity to synchronize games with players not near the chess board, ensuring accurate move tracking and consistency across devices. A mode selection switch will allow users to toggle seamlessly between these modes.

Additional functionalities include automatic piece detection using magnets attached to the chess pieces and hall sensors embedded in each square to reliably register the presence and movement of chess pieces. This system ensures synchronization between physical actions and digital processing. A timer for each player may also be implemented for a player's move.

Potential extended functionalities may include game logging, which would record moves for post-game review, as well as a companion application for an online multiplayer feature.

2.4 Existing Products/Prior Work

2.4.1 SMART Chess Board

While researching for our design, we came across a previous Senior Design project from 2023, where the group also made a smart LED chess board titled “SMART Chess Board”. Their design made use of hall effect sensors to determine where and when a piece was lifted, and would then indicate on the chess board where valid moves were using LEDs. This group also made their own chess library to avoid having to alter an existing one, but found it difficult as time went on. This project gives us an understanding of one way to implement our systems, though we have chosen to alter ideas, such as adding in the ability to play against a computer player, and also deciding to use an existing open-source chess library that we can change as needed.

2.4.2 DIY Super Smart Chessboard

Another source we came across, this time while trying to develop our idea of being able to connect the chess board with Wi Fi, was an existing hobbyist project, named the “DIY Super Smart Chessboard”. This project makes use of a LED strip to light the board similar to our plan, though it is controlled by manually selecting the space the player wishes to move to through the use of buttons, which then light the LEDs, then after a manual confirmation, the player moves the piece. To make sure the move is legal in chess, libraries are implemented. While this program lacks a sensor to indicate when a piece is moved, we found value in reading how they implemented their ability to play against a computer player and human player using Wi-Fi.This project uses Wi-Fi capabilities granted by the raspberry pi board and various libraries it has to connect with other Super Smart Chessboards to allow online play. Though we do not plan to use the raspberry pi board, this project still provides an outline of what we are striving for, and also shows us one way to do it, which we can alter as needed.

2.5 Engineering Specifications

Table 1: System-Level Specifications

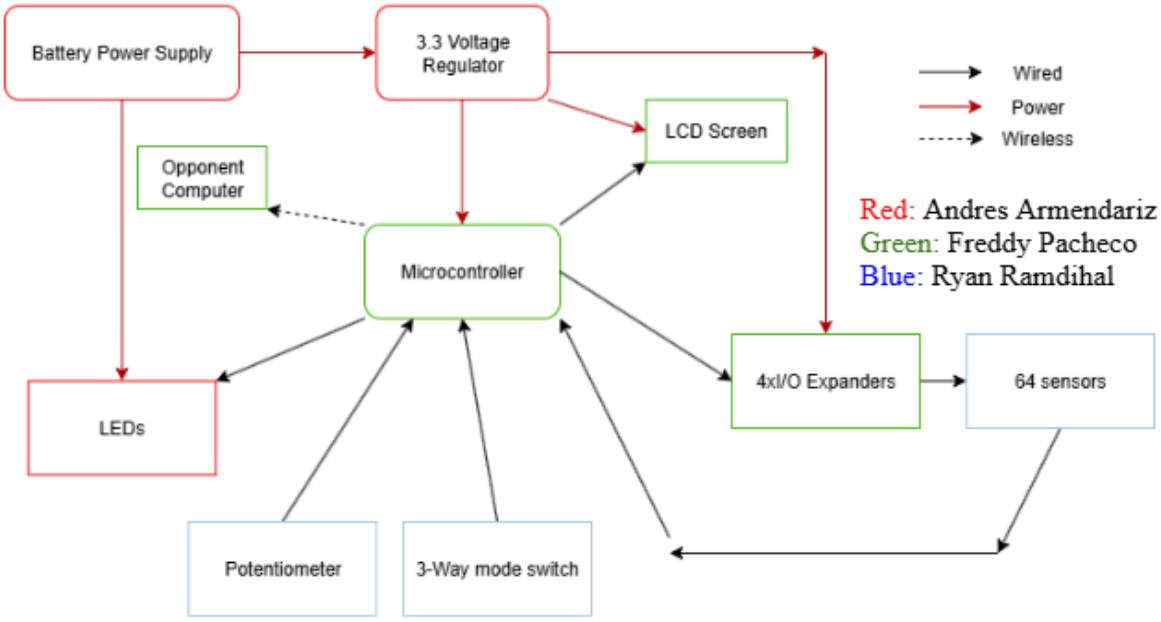
Parameters	**Target Value**	**Notes**
Chess Board Dimensions	Main chess board:<20x20x2 in Side box: <5x5x2 in (Length x Width x Height)	Standard chessboard size with space for electronic components.
Battery Life	≥ 2 hours	Duration of normal gameplay session on one charge
Manufacturability (Cost)	< $350	Budget target for prototype parts
Temperature	< 100 degrees Fahrenheit	Safe for indoor operation
Weight	< 15 lbs	Portability
Modes	3	The path finder chessboard should have a minimum of 3 different modes the user can

		select.

Table 2: Component-Level Specifications

Component	**Parameter**	**Subsystem and Description**	**Target Value**
LED	Power	Electrical - The LED watt usage should ideally be less than the maximum wattage.	<22 Watts
ESP32	Wifi	Wifi signal strength should be strong enough for reliable online play within 5-15 meters.	>-75 dBM
Sensors	Magnetic Flux	Electrical-The magnetic sensors should be able to pick up when a piece is there or not.	~0 V for values >3.32mT ~5V for values < 1.17 mT

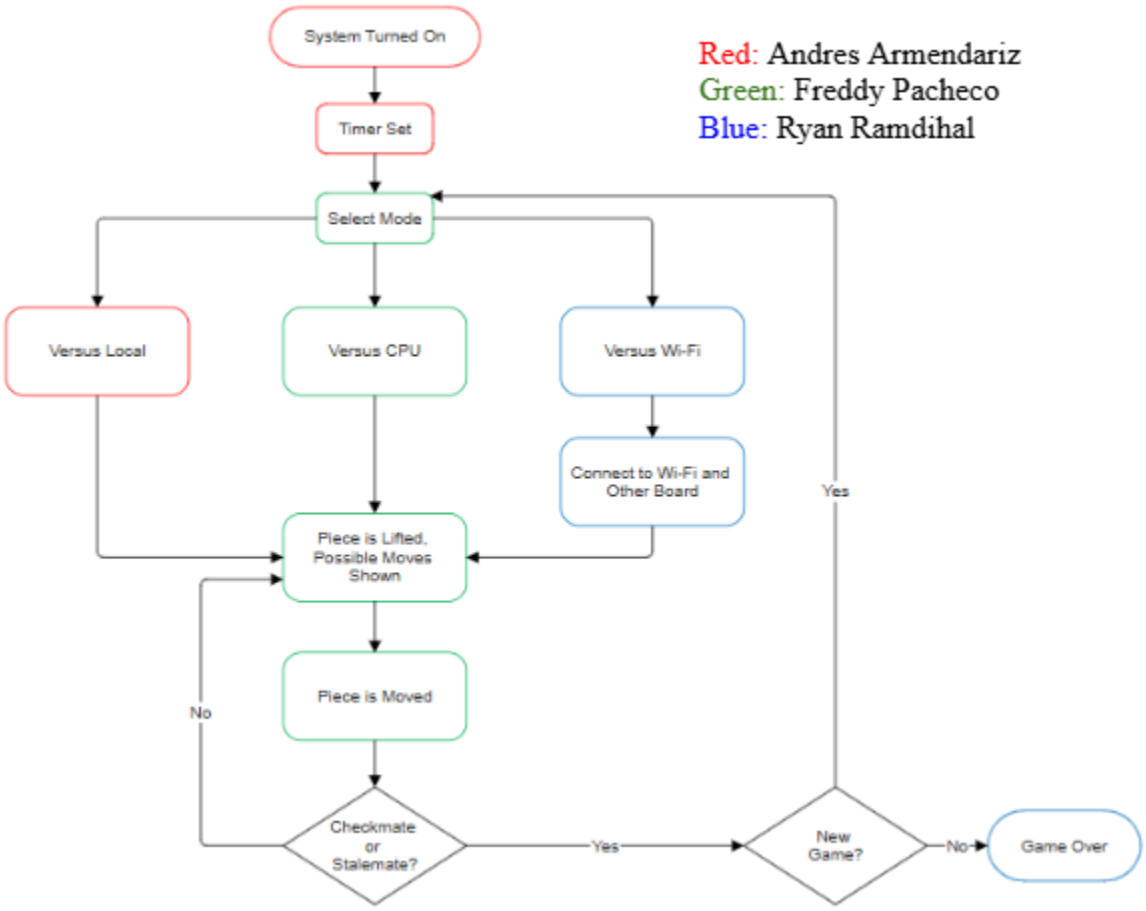
2.6 Hardware Block Diagram



Block	**Status**
Battery Power Supply	Investigating
3.3 Voltage Regulator	Investigating
LCD Screen	To be acquired
Microcontroller	To be acquired
LEDs	To be acquired
I/O Expanders	To be acquired
Opponent Computer	Acquired
3 mode switch	Investigating
64 Sensors	Investigating

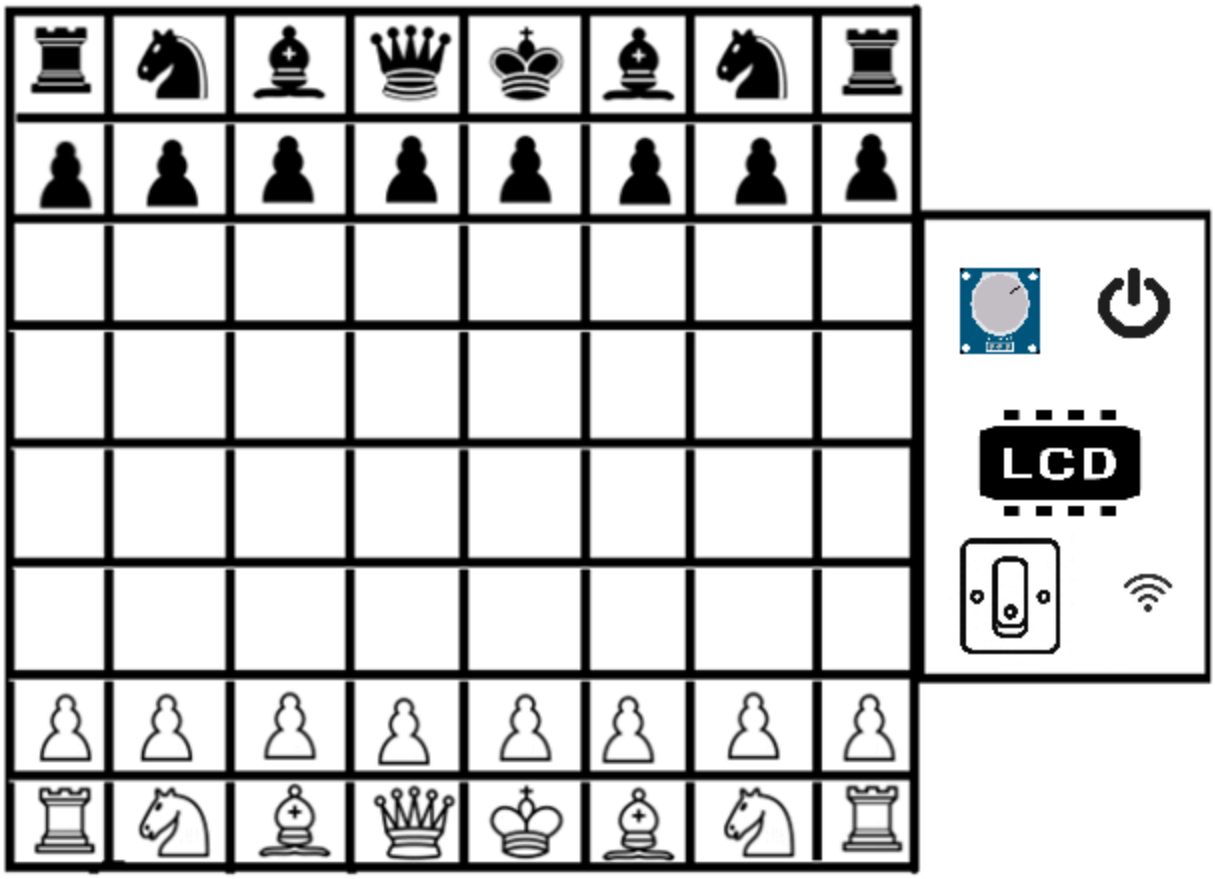
Potentiometer	To be acquired

2.7 Software Diagram/Flowchart



2.8 Prototype Illustration/Blueprint

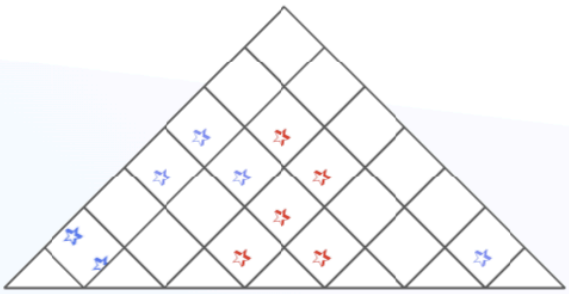
The image below is a prototype illustration of the PathFinder Chessboard.The illustration does not show the height of the box which would be around 1.5 inches. The dimensions of the chess board measure around 18x18x2 for the main chessboard and 4.5x4.5x2 for the box to the right of it. Underneath each of the 64 tiles of the Chessboard will be a magnetic sensor and a LED mounted on the PCB. The LEDs would be daisy chained with the microcontroller, with the regulators and other components embedded in the PCB as well. To the right of the chess board will be the on/off switch, 3-way mode switch, LCD screen, potentiometer used as a timer, and wi-fi antenna.



2.9 House of Quality

Key

Positive Correlation	☆	↑
Strong Positive Correlation	☆☆	↑ ↑
Negative Correlation	☆	↓
Strong Negative Correlation	☆ ☆	↓ ↓



Consumer Requirements	LED Response time	Sensor accuracy	Component Costs	Chess engine integration	Power Consumption	Material Quality	Weight
							Engineer Requirements
Accessibility	↑ ↑	↑ ↑		↑		↑	↓
Power Efficiency					↑ ↑		
Speed	↑ ↑	↑ ↑		↑	↑		
Affordable Cost			↓ ↓				
Portability							↓ ↓
Durability					↑	↑ ↑	↓
Target for Requirements	≤ 250ms	≥ 90%	≤ $100	-	≤ 5 W	scratch resistant	≤ 10 lbs

Chapter 3: Research and Investigation

3.1 Hardware Technology Comparison and Selection

To develop an effective engineering project, it is crucial to research and investigate possible technologies that can be used. Different technologies have to be compared amongst one another to see which technologies are appropriate and specialized to the project's constraints and specifications. The choice in technology is an essential part of ensuring a successful and functional project. This section discusses and compares different technologies and explains the rationale behind the component selections made to meet the specific needs of the Pathfinder Chessboard project.

3.2 Hardware Part Comparison and Selection

3.2.1 Light Sources

A core specification of the Pathfinder Chessboard is the illumination system. The illumination system serves as the user interface for conveying optimal move guidance. The selected component had to satisfy a rigid set of criteria to achieve this goal. Such criteria included that the light source must be bright enough to remain clearly visible through the acrylic layer, but not so bright as to possibly harm the user. Additionally, it must also have the power efficiency necessary to keep the battery preserved for at least 2 hours. Finally, compatibility with the ESP32-S3 microcontroller is also a critical constraint that allows the illumination system to communicate with the rest of the components.

Three potential light sources were evaluated for this project: electroluminescent displays, light emitting diodes, and laser diodes. The following sections describe each component, how they work, and the variating types of each component.

3.2.1.1 Electroluminescent Display

The Electroluminescent (EL) Display is a type of flat panel display created by putting a layer of electroluminescent material between two layers of conductors. When current is applied the electroluminescent material starts to glow. This type of display is valued for its uniform light output, thin profile and low power consumption. EL Displays require a high voltage AC power source in order to power them and usually have lower brightness and shorter lifespans compared to other light sources, which limits their use in applications that require higher visibility or color control. Although EL displays can be made thin and flexible they do require more circuitry to operate, which can increase the system size and complexity compared to other light source alternatives.

The most common types of EL Displays are Powder EL, Thin-Film EL, and Thick Film EL. The most common type among these is the Thin-Film Electroluminescent (TFEL) display.The TFEL display is nichely used in industrial, military, and automotive applications due to its extreme durability and ability to operate in extreme environments. Powder EL displays, on the other hand, are typically found in simpler devices such as backlighting or night lights due to their low cost and ease of production. Thick Film EL displays combine features from both the powder and TFEL display to create a brighter and more robust screen. While Thin-Film and Thick Film EL displays offer greater durability and performance they are generally more expensive to manufacture compared to the Powder El types.

3.2.1.2 Light Emitting Diodes (LEDs)

Much like the EL display Light Emitting Diodes (LEDs) produce light when a current is applied through the forward bias direction. The light is produced by the movement of electrons through the semiconductor junction, which releases energy that can be seen as visible light, invisible infrared, or ultraviolet radiation. The wavelength of the light depends on the semiconductor material and its bandgap energy. Different materials are able to produce different wavelengths with the two extremes being infrared on the lower wavelength and ultraviolet on the longer wavelength. Advantages with LEDs include their high efficiency, long lifespan, and small size which allows them to be used in a wide range of different applications. With power LEDs unlike EL displays do not require a high AC voltage in order to operate but instead operate on a low DC voltage. Leds are also generally small this in turn makes them energy efficient and easier to control for projects where many different lights are needed.

The three most common types of LEDs are RGB, UV, and infrared. RGB LEDs can emit a single color of light and are the most common type found in indicator lights and simple displays. The RGB LEDs can also mix different shades of red, green, and blue in order to achieve different colors. By controlling the intensity of each color, these LEDs can produce millions of color combinations, which makes them useful for color coding lights. Another type of LED is the UV LED. These are commonly used in sterilization and disinfecting applications and can also be used in printing or other black light applications. These types of LEDs emit an ultraviolet light that can kill bacteria or viruses and can cause fluorescent or phosphorescent materials to glow.Similar to the UV LEDs infrared LEDs emit a light that is on the opposite side of the electromagnetic spectrum. Infrared LEDs however produce a light that is invisible to the human eye.These particular LEDs are commonly found in cameras and proximity sensors due to their ability to be invisible to the human eye.

3.2.1.3 Laser diode

Laser diodes, unlike LEDs, have a coherent, high intensity unidirectional beam of light that can travel much longer distances. Laser Diodes work similarly to LEDs as both use a PN junction made of semiconductor materials to produce light when an electrical current flows

through them. However, laser diodes include mirrors inside an optical cavity which allows the light to bounce back and forth in order to amplify itself. These components allow all the light waves to be in phase with each other and travel in the same direction which results in a highly concentrated beam that can travel long distances. Laser diodes require precise current control to operate properly as too much current can damage the device while too little will not produce a laser. Laser diodes also produce more heat compared to LEDs due to their higher power density which can oftentimes complicate systems. Despite all these requirements laser diodes are rated highly for their ability to produce extremely focused and powerful beams of light making them essential for implementations that require precision and long range transmission.

The most common types of laser diodes are Fabry-pérot, Distributed Feedback(DFB), and Vertical-Cavity Surface Emitting Lasers (VCSELs). Fabry-pérot are the simplest and most common type. Mirrors at both ends of the internal cavity are used to reflect the light back and forth. This type is commonly found in optical storage like CD, DVD and blue ray players as well as in laser pointers and short range communication systems. DFB laser diodes use diffraction grating instead of mirrors to produce a stable, single wavelength output which makes them ideal for fiber optic communication. VCSELs emit light perpendicular to the surface of the chip which makes them easier to manufacture and test. They are commonly found in computer mice, and facial recognition systems like face ID, and high speed communication.

Table 3: Illumination Component Comparison Table

Component	**Power Consumption**	**Color/Brightness Control**	**Cost**	**Lifespan/Durability**
Electroluminescent display	Low	Basic	Moderate	Medium
LED	Very Low	Excellent	Moderate	Long
Laser Diodes	Medium	Good	High	Medium

3.2.1.4 Illumination Component Selections

To determine the most suitable lighting technology for the Pathfinder Chessboard three components were evaluated: Electroluminescent displays, LEDs and Laser Diodes. Each component was assessed based on power consumption, color and brightness control, cost and lifespan and durability. Looking at the table we can see an array of the different options from which to pick the light source to be.

From this data we can conclude that the component that outperforms in all categories are

the LEDs. They provide the optimal combination of very low power consumption, excellent color and brightness control, moderate cost that fits within the project budget and long lifespan that ensures reliability. The ability to use addressable RGB LEDs allows for color coding where different colors can represent legal moves, suggested optimal moves, and other game states. The addressable RGB LEDs also have excellent compatibility with the ESP32-S3 microcontroller through GPIO control making them the clear choice for the Pathfinder Chessboard illumination system. For this project we specifically chose the smart WS2812B LEDs because of their ability to be daisy-chained which reduces the number of PCB traces and wires significantly. The WS2812B LEDs can also be addressed individually which allows for full control of which colors they can display and where. The individual addressability also means that they only need a single data pin from the microcontroller which simplifies the circuit design even further.

3.2.2 Sensors

In order for the illumination system to function there must be some sort of trigger. In our case this trigger will be set off by a sensor underneath the casing of the chess board. When picking a sensor we must consider several important factors to ensure reliable detection of chess piece movements. The sensor must be able to accurately detect what piece is placed on or moved from a square without being affected by the color or material of the chess pieces. It should also be compact enough to fit beneath each of the 64 squares on the chessboard without adding too much thickness to the overall design. The sensor should also be able to not pick up any false readings that could occur due to playing. Along with these constraints the sensors must also be cost effective and integrate easily with the ESP 32 microcontroller.

3.2.2.1 Hall Effect Sensors

A hall effect sensor works by detecting a magnetic field and converting it into an electrical signal. When a magnetic field passes through the sensor it creates a voltage difference across the sensor due to the Hall effect, which occurs when charged particles moving through a conductor are deflected by the magnetic field. In order for the Hall effect sensors to work properly in this project a magnet must be attached to each of the chess pieces which would increase the budget of the project.

There are two main types of hall effect sensors: linear and switch Hall effect sensors. Linear hall effect sensors have their output voltage demonstrate a linear relationship with the strength of the magnetic field. The linear variation in output voltage is best for applications where precise and accurate measurements of the magnetic field intensity are needed, such as position sensing or measuring the distance between the magnet and sensor. Switch Hall effect sensors activate when the strength of the magnetic field surpasses a specific threshold value and conversely deactivate when the magnetic field is below that threshold. The switch hall effect sensor is best for applications where the sensor only needs to detect when the magnet is near or not near making it ideal for simple presence detection like determining if a chess piece is on a

particular square.

3.2.2.2 Capacitative Sensors

Capacitive sensors work similarly to Hall effect sensors except instead of a magnetic field the capacitive sensor works by detecting an electrostatic field. These sensors measure changes in capacitance that occur when a conductive or dielectric object comes near the sensor surface. The sensor then creates an electric field between two conductive plates and changes the capacitance by altering the distribution of the electric field lines. Capacitive sensors are used to detect the presence or absence of virtually any object regardless of material. However capacitive sensors could be prone to false triggers from hands, cables, or other conductive materials.

There are two main types of capacitive sensors: self-capacitance and mutual capacitance sensors. Self-capacitance sensors use a single electrode and measure the capacitance between the electrode and ground, making them simpler and less expensive but more susceptible to noise. Mutual capacitance sensors use two electrodes, a transmitter and receiver and measures the capacitance between them. The capacitance decreases as an object gets closer to touching it by disrupting the electric field between them.

3.2.2.3 Inductive Sensors

Inductive sensors are much like capacitive sensors in that they can detect material without contact, but unlike capacitive sensors they can only detect conductive metals. Inductive sensors work by generating an electromagnetic field and detecting changes in that field that are caused by the presence of metallic objects. The sensor contains a coil of wire that carries an alternating current, which creates a magnetic field around the coil. When a conductive object enters this magnetic field eddy currents form on the metal surface producing a magnetic field that opposes the coils magnetic field. The changes in inductance of the coil can then be detected by the sensor's circuitry. Inductive sensors are highly reliable and have fast response times, making them suitable for many applications. They are commonly used in position sensing and metal detection applications.

However, inductive sensors have some important limitations. They can only detect conductive metals such as iron, steel, aluminum, copper, and brass, which means they cannot detect non metallic materials such as wood, glass or plastic. Depending on the sensor size and the type of metal being detected, detection ranges from a few millimeters to several centimeters. Inductive sensors can also be affected by electromagnetic interference from nearby electrical equipment.

3.2.2.4 Piezo Electric Sensors

Piezo electric sensors work by applying mechanical pressure on the sensor which then

releases an electric signal due to the piezoelectric effect. It also does not need an external voltage source or current source in order to operate. The piezoelectric effect occurs only in certain crystalline materials such as quarts, ceramics and some polymers. Mechanical stress causes the internal electric dipoles to realign and produce a voltage across the material. Piezo electric sensors are very durable and have long operational lifespans since they have no moving parts. Additionally, piezoelectric sensors have fast response times and can detect changes in pressure almost instantaneously.

Three common types of piezoelectric sensors include piezoelectric accelerometers, piezoelectric force sensors, and piezoelectric pressure sensors. Piezoelectric accelerometers measure acceleration through the stress of a force on a piezoelectric object. These are commonly used in vibration monitoring and motion detection applications. Piezoelectric force sensors measure the magnitude of an applied force by detecting the voltage generated when pressure is applied to the piezoelectric material. They are used in applications such as impact testing or tactile feedback systems. Piezo electric pressure sensors measure changes in pressure that are caused by the mechanical stress on the sensor. These sensors are used in dynamic pressure measurements such as in fluid or gas systems.

However, Piezoelectric sensors are best suited for detecting dynamic changes in pressure rather than static pressure. This means they generate a voltage signal when a force is applied or removed but do not maintain a continuous output while constant pressure remains on the sensor. This characteristic makes them ideal for detecting impact, vibrations or other pressure changes, but limits their use in applications that require continuous monitoring of static loads or constant pressure states.

3.2.2.5 Ultrasonic Sensors

Ultrasonic sensors are the most unique sensor out of all the other sensors being compared. Like most of the other sensors the ultrasonic sensor can output signals without needing to be touched. What separates the ultrasonic sensor from other sensors is the way it processes information. Ultrasonic sensors measure the distance to an object by using ultrasonic sound waves that reflect back to the sensor allowing it to measure distance. By measuring the time it takes for the sound wave to travel to the object and back, the sensor can calculate the distance using the speed of sound in air. Ultrasonic sensors can detect most materials except any type of object that is highly sound-absorbent like carpet or foam.

However, ultrasonic sensors have several limitations that affect their performance in certain applications. The accuracy of distance measurements can be affected by objects in the way and other environmental factors. Ultrasonic sensors also may struggle to detect objects with angled surfaces that scatter sounds away from the receiver. Ultrasonic sensors also have relatively slow response times compared to other types of sensors because they must wait for the sound wave to travel to the object and back. They also require external power to generate the

ultrasonic pulses and process the return signals. Additionally multiple ultrasonic sensors operating near each other can interfere with one another if they emit pulses at the same frequency, which can lead to false readings.

Table 4: Sensor Comparison Table

Sensor Component	**Triggered by**	**Detects**	**More likely to cause False triggers**	**External Power required**	**Cost**
Hall effect	Magnetic fields	Magnetic materials	Low - only responds to magnetic fields	Yes	Moderate
Capacitative	Electrostatic fields/changes in capacitance	Any conductive or dielectric material	High - reacts to hands, moisture or cables.	Yes	Low to moderate
Inductive	Electromagnetic field changes	Conductive metals only	Moderat - can pick up interference	Yes	Moderate
Piezo Electric	Mechanical stress or vibration	Changes in pressure force or acceleration	Low - only triggers from impact or movement	No	Moderate to high
Ultrasonic	Sound wave reflection	Most solid and liquid materials	Moderate - echoes and angles can cause errors	Yes	Moderate

3.2.2.6 Sensor Selections

To determine the most suitable sensor technology for the Pathfinder Chessboard, five sensor types were evaluated: Hall effect sensors, capacitive sensors, inductive sensors, piezoelectric sensors, and ultrasonic sensors. Each sensor was assessed based on what triggers it, what materials it can detect, susceptibility to false triggers, external power requirements and cost.

In order to determine what type of sensor will be used in this project we must take into account all the advantages and disadvantages of each sensor type. Hall effect sensors require magnets to be attached to each chess piece, which would increase the cost but would provide reliable detection with low false trigger rates. Capacitive sensors can detect any object without piece modification but have high susceptibility to false triggers from hands, moisture or cables during gameplay. Inductive sensors only detect conductive metals and would require metal pieces or modifications, making them less practical. Piezoelectric sensors do not require external power and have low false trigger rates, but they only detect dynamic pressure changes rather than static presence. This would require additional software to track the board state. Ultrasonic sensors can detect most materials but having the pcb underneath the board would make it tricky for the ultrasonic sensors to have precise detection.

From this data we can conclude that Hall effect sensors provide the best combination of reliable detection, low false trigger rates, moderate costs, despite requiring magnets to be attached to the chess pieces. For the Pathfinder chessboard

3.2.3 Displays

In order to interact with the user the Pathfinder Chessboard must also have a display in order to serve as the main visual output of the system. It will show information such as player turns, detected piece positions, and other useful information. In order to satisfy the specifications of this project we have to consider how these screens can communicate with the microcontroller. Size and power consumption also need to be taken into consideration when choosing a display. choose a display that can be programmed alongside our micro controller.

3.2.3.1 Organic Light Emitting Diodes (OLEDs)

Organic light emitting diodes (OLEDs) use a similar way to function as LEDs but are commonly used to produce high quality displays. OLEDs are able to display text, graphics and full color images.OLEDs can come in many different shapes in sizes but for our project we will be focusing on the smaller variants of OLED displays.These types of displays typically use high speed serial protocols like SPI or I2C. Furthermore it can be noted that OLED screens are also able to display in monochrome up to full color but this is more commonly seen in bigger OLED screens.

3.2.3.2 Liquid Crystal Display (LCD)

Liquid crystal displays (LCDs) are another common type of screen used in embedded systems and PCB projects. The LCD screen can interact with a microcontroller by using I2C, SPI, or parallel communication protocols. They consume relatively little power, making them an energy efficient option. One limitation however is that LCDs are typically limited only to display

text and characters. Due to their simplicity and low cost LCD screens are a popular choice for many basic user interfaces.

3.2.3.3 Seven-Segment Display

The seven-segment display is the simplest and most affordable display option among those evaluated. It uses GPIO pins or I2C for communication and is designed primarily for showing numeric characters. The sizes of the 7 segment display are typically small but ease of integration makes it a practical choice compared to LCD or OLED displays.

Table 5: Display Comparison Table

Display	**Communication Protocol**	**Size (In)**	**Power Consumption**	**Display Capability**
OLED	SPI, I2C, Parallel	0.96-3	Low to medium	High
LCD	UART SPI, I2C	16x2 to 20x4 characters	Low	medium
7 Segment Display	GPIO, I2C	Small	Very low	Low

3.2.3.4 Display Component Selections

For the Pathfinder Chessboard design we decided to go with the LCD screen for a variety of reasons. We specifically choose an I2C based LCD screen to communicate with microcontrollers in order to efficiently allow for less pins to be used on the microcontroller. The LCD screen can come in different sizes and can fit on the side compartment of the Pathfinder Chessboard comfortably. The selected display also only needs to be able to display numbers or characters so there is no need for a display that has graphics or full color images. The LCD screen is more versatile than the 7 segment display but does not have the undesired added cost and power management complexity OLED display. This makes the LCD screen a perfect middle ground to take when picking a display.

3.2.4 Microcontrollers

The microcontroller is the brain of our system and controls and reads the data imputed by the sensor and outputs data to the LEDs.The microcontroller also has to be able to transmit or receive data from the potentiometer, buttons, and LCD screen. In order for the microcontroller to fully integrate with the system it will have to have I2C and analog communication protocols in

order to communicate with the rest of the components. As a stretch goal the microcontroller should also have on board wi-fi if the user wishes to play with someone online.

3.2.4.1 ESP32

The ESP32, developed by Espressif Systems, is a dual-core 32-bit microcontroller that can operate up to 240 MHz. The ESP32-S3 supports multiple peripheral interfaces, including SPI, I2C, UART, and ADC, providing versatile hardware support for sensors and displays.

This device is particularly suitable for the Pathfinder Chessboard due to its ability to handle concurrent tasks, such as continuously scanning piece positions, validating moves in real time, and updating the display - using its dual-core processing capability. The ESP32-S3's GPIO availability enables direct interfacing with the chessboard's sensor grid and LED display modules, while its integrated wireless capability allows for future expansion of our stretch goal with Bluetooth connectivity for mobile applications or Wi-Fi based online play.

Although it consumes slightly more power than the other two MCUs, the ESP32 provides the best balance of scalability, connectivity, and performance.

3.2.4.2 Arduino Uno

The Arduino Uno, powered by the 8-bit ATmega328P microcontroller, operates at a clock speed of 16 MHz. It provides basic communication interfaces such as UART, SPI, and I2C, with a total of 14 digital and 6 analog GPIO pins.

While it is well-known for simplicity, affordability, and large development community, the Arduino Uno is not optimal for the Pathfinder Chessboard system. The board's limited memory and single-core 8-bit architecture restrict its ability to perform multiple tasks concurrently. Handling simultaneous sensor readings, move validation, and LED control would likely result in timing conflicts or slow response.

3.2.4.3 MSP430G2553

The MSP430G2553, developed by Texas Instruments, features a 16-bit architecture running up to 16 MHz and emphasizes ultra-low power consumption. The MCU includes common interfaces such as UART, SPI, and I2C, and operates within the 1.8 V - 3.6 V range.

While the MSP430 excels in low power consumption, its limited processing speed makes it unsuitable for the computational requirements of the Pathfinder Chessboard. The project's need to scan multiple sensors, manage LCD updates, and process game logic in real time, exceeds the MSP430's capabilities.

Table 6: Microcontroller Comparison Table

Micro controller	Processing Power	Clock Speed	Development Environment	Communication Interface	Operating Voltage
ESP32-S3	Dual-Core RTOS support (HIGH)	MAX speed 240 MHz	Arduino IDE ESP-IDF	I2C I2S SPI UART ADC	2.3 V - 3.6 V (3.3V)
Arduino Uno (ATmega328P)	8-bit Single Core (LOW)	16 MHz	Arduino IDE	I2C SPI	5V
MSP430G2553	16-bit (MODERATE)	16 MHz	Code Composer Studio	I2C SPI UART	1.8 V - 3.6 V (3.3V)

3.2.4.4 Microcontroller Selection

After evaluating the three microcontrollers, the ESP32 was selected as the primary microcontroller for the Pathfinder Chessboard. It has superior processing power, dual-core capability, and integrated wireless interfaces provide the performance and flexibility necessary for the system's functional requirements. The ESP32-S3's GPIO pins simply connect multiple hall-effect sensors embedded in the board squares and should be compatible with I/O extenders for additional pins when needed.

In addition, the ESP32's support for multithreading allows separate cores or tasks to manage different system components, for example, one core handling sensor scanning and rule enforcement while the other controls the LCD display and LED matrix. The microcontroller's compatibility with both the Arduino IDE and native ESP-IDF framework ensures simplified prototyping and scalable software development.

While the MSP430 offers exceptional power efficiency and the Arduino Uno provides simplicity, neither platform delivers the required real-time responsiveness or computational throughput. Therefore, the ESP32 represents the optional choice for the Pathfinder Chessboard, ensuring reliable operation and sufficient capacity for future enhancements such as wireless functionality or game data logging.

3.3 Power Supply Unit

3.3.1 Distribution of Component Power

The power design for any pcb is a crucial aspect to consider. One of the first steps in designing a power system is to calculate the maximum total power draw from all active components being used in the design of the chess board. When calculating the power we also need to keep in mind the constraints and specifications of the design. One of the constraints being that the battery life had to last for at least 2 hours. In order to reach a design with this constraint in mind the maximum power consumption wattage needs to be calculated. Due to overestimating power supply specifications as a safety margin for our component, active components that used a negligible amount of power were not taken into consideration. Using the datasheets we were able to find the maximum current drawn. The voltage supplied to each active component can then be multiplied by the maximum current drawn in order to get each component's maximum power consumption. Summing the maximum power consumed by each component can then give the necessary power constraints needed by the system.

Based on our calculations, the power output of the system is estimated to be around 7.45W. This would entail that in order for our design to meet the constraint of lasting at least 2 hours we have to choose a power supply with an energy specification of at least 14.9 Watt-hours (Wh). Below is a table outlining the different active components, their voltage and maximum current draws, their power usage, and the entire system's power and energy consumption.

Table 7: Power/Energy Usage Table

Component	**Voltage (V)**	**Current Draw**	**Power Usage (W)**
LEDs	5.0	1.28 A	6.40 W
ESP32	3.3 V	240 mA	0.792 W
LCD	3.3V	74 mA	0.2442 W
I/O expanders	3.3V	4mA	0.0132 W
Other components	Negligible	Negligible	~0 W
System total Power (W)			7.45
System total Energy (Wh)			14.90

3.3.2 Batteries

Some of the primary specifications needed when choosing a battery are the nominal voltage(V), capacity(Ah), energy capacity (Wh). The nominal voltage of the battery must match near 3.3-5(V) devices in order to be regulated to 3.3 and 5 volts in order to keep efficiency during voltage regulation and therefore improving stability. Aiming for an overestimate of around 35% the batteries should have a capacity of around 4.02-6.10 (Ah) in order to power all the components in the system efficiently. Due to the system only using what it needs we can safely overestimate the capacity of the battery chosen. The energy of a battery can then determine how long the battery can deliver power.

Another constraint to consider for the Pathfinder Chessboard is portability. The unit that indicates how much mass or volume a battery will have is its energy density (Wh/kg). To take into account the portability the weight of each type of battery was estimated. To complete the portability constraint the power system had to be supplied by a lightweight non bulky battery power delivery system.

Safety is also an important specification to consider when researching batteries. Toxicity and combustion are two key aspects that must be considered when evaluating the safety of each battery type. Due to the chemistry of different batteries, some batteries are known to be more likely to explode compared to others. The same consideration applies to toxicity levels.

Although not a constraint, recharability is a specification that we can consider when picking a battery. Rechartibility offers for batteries to be reused without having to dispose of them when they deplete their voltage. Rechargeability does however add much more complexity to a system by adding a battery management system (BMS), which monitors and protects the battery during operation and charging.

3.3.2.1 Lithium Ion

Lithium Ion (Li-Ion) batteries are used as a simple rechargeable battery that can be made in almost any shape or size. These batteries are commonly found in very light and compact forms. Li-Ion has higher energy capacities compared to other battery chemistries. Each Li-Ion cell has a nominal voltage of around 3.7 volts and an energy density of 150-220 Wh/kg. Li-Ion batteries are often used for their small size and due to their small size and ability to recharge. However, due to the chemistry of Li-Ion batteries they do pose a serious risk of exploding if not managed properly with a BMS.

3.3.2.2 Lead Acid

Lead acid batteries are some of the oldest types of rechargeable batteries. Each lead acid battery cell has a nominal voltage of 2 volts and their energy density ranges from 30-50 Wh/kg. These batteries are known for their reliability, low-cost, and ability to deliver high power, which is why they are commonly used in automotive applications. However, they are heavy and bulky making them unsuitable for portable systems.

3.3.2.3 Nickel-Metal Hydride (NiMH)

Nickel-Metal hydride (NiMH) batteries are another type of rechargeable battery. Each NiMH cell has a nominal voltage of about 1.2 volts each and an energy density ranging from 60-120 Wh/kg. NiMH batteries are known for being relatively safe and environmentally friendly, but they do take up more space compared to lithium based batteries.

3.3.2.4 Alkaline

Alkaline batteries are non rechargeable with each cell having a nominal voltage of 1.5v except for their 9 V variation. They have energy densities of roughly 80-120 Wh/kg and are commonly found in household AA, AAA, C,D and 9 V batteries. They are inexpensive and widely available.

Table 8: Battery Comparison

Battery type	**Nominal Voltage (V)**	**Energy Density (Wh/kg)**	**Lifespan (Cycles)**	**Size**	**Combustion likelihood**	**Toxicity**	**Recharability**
Li-ion	3.7	250-670	300-1,000	Smallest	Moderate	Moderate	Recharable
Lead Acid	2	60-90	300-1,200	Largest	Low	High	Recharable
NiMH	1.2	60-120	500-2,000	Medium	Very Low	Low	Recharable
Alkaline	1.5/9	100-150	Single use	Medium	Very Low	Low	Not recharable

3.3.2.5 Battery Selection

Alkaline batteries fit the specifications and constraints for this system the best out of all the different types of batteries.Although alkaline batteries take up more space compared to other types of batteries they do not require a complex BMS in order to operate and are able to last a long time due to their large energy capacity. Alkaline batteries are also much less likely to explode compared to the other types of batteries and are generally very safe. Although they do take more space and are heavier than other batteries the side compartment of the PathFinder chessboard is large enough to fit them without any problems

For this design three C alkaline batteries are going to be used. The three C alkaline batteries are perfect for this design as their 1.5 nominal voltage in series would create a nominal voltage of 4.5V. Their large capacities also make it so they can power the current draw from all the active components. Due to the simplicity of their battery management syst

3.3.3 Voltage Regulators

The path finder chess board has two main voltage rails due to different components specifications requiring different input voltages and currents. These are the 5 volt and 3.3 voltage rails. The battery on the other hand has a nominal voltage of 4.7 V meaning that in order to effectively power the components without damaging them voltage regulators must be used. This section will be covering the different types of voltage regulators and how they can be used.

The two main types of voltage regulators are linear and switching. Linear regulators operate by acting as a variable resistor in order to dissipate excess energy as heat and maintain a constant voltage. Switching regulators in contrast work by rapidly switching the input voltage on and off using inductors and capacitors in order to regulate the output voltage. A key difference between the two types of regulators is that linear regulators can only step down voltage whereas switching regulators can step down or step up voltage. More differences lie in that linear regulators are more simple to build compared to switching regulators. However it must be noted that switching regulators which do require more complexity to build but their power conversion efficiency is much greater than the linear regulators.

3.3.3.1 LP3856 (Linear Voltage Regulator)

The LP3856 is a linear voltage regulator that can help meet our design constraint of regulating the input voltage to 3.3 V. The LP3856 has an input voltage of 2.5-7 V which meets our input voltage of 4.5. The output voltage can be fixed to be 1.8, 2.5, 3.3, or 5 volts. The LP3856's maximum current output is also 3 A which is enough to power all the 3.3 V components. The Efficiency can be calculated by taking Vout/Vin which gave us 73.3%

3.3.3.2 TPS61022 (Switching Voltage Regulator)

The TPS61022R is a boost converter, therefore this switching voltage regulator can boost the input voltage to the desired LED voltage of 5 V. The TPS61022R can boost voltages from a minimum Vin of 0.5-5.5 V to an output of voltage of 2.2-5.5 V. These voltage input/output specifications match what our system needs. The maximum current output of the TPS61022R switching regulator is 3 A which meets our 1.62 A requirement for the LEDs. The efficiency of this regulator is also rated at around 90 %.

3.3.3.3 TPS63000 (Switching Voltage Regulator)

The TPS63000 is a buck-boost converter meaning it can both increase or decrease the input voltage depending on the configuration of the regulator. The maximum voltage input is 1.8-5.5 V. This regulator has an output voltage of 1.2-5.5 V. The maximum current output of this regulator however is 1800 mA meaning that this could supply the 3.3V rail but not the 5V rail due to a limiting switching output. With this constraint in mind this switching regulator can still be used as a buck converter to regulate the input voltage from 4.5 V to 3.3 V. The efficiency of this regulator is also 96%.

Table 9: Voltage regulator comparison

Regulator	Type	Input Voltage (V)	Output Voltage(v)	Output Current(A)	Efficiency
LP3856	**Linear**	**2.5-7**	**1.8,3.5,3.3,5**	**3**	**~73.3 %**
TPS61022R	**Switching**	**0.5-5.5**	**2.2-5.5**	**3A**	**90%**
TPS63000	**Switching**	**1.8-5.5**	**1.2-5.5**	**1.8 A**	**96%**

3.3.3.4 Voltage Regulator Selection

We decided to go with the TPS61022R as our booster converter to 5V and the TPS6300 as our buck converter to 3.3 V. The two switching regulators may have a higher cost and component complexity compared to the LP3856 regulator, but their power efficiency is significantly superior compared to the linear regulator. Furthermore in order to maximize battery life and minimize heat these 2 regulators were most optimal for the project.

3.4 Software Technology Comparison and Selection

To allow the hardware to communicate with our selected microcontroller, software must

be used. Researching this software has a very high importance in engineering projects, as it significantly alters the efficiency and accuracy of it. We search for technologies which can seamlessly integrate our resources which are then described in this section.

3.4.1 Development Language

When first developing our project we had to decide our programming language we would use in conjunction with our ESP32. For this section we are not necessarily looking for one singular language, but rather the main language we use for our implementation. If existing libraries are in other languages and can be implemented we will take that option.

3.4.1.1 C

The C programming language is a language which is powerful and general purpose. C provides direct access to hardware and memory which can lead to high efficiency. This is important to our project as we are working with limited hardware and want to use it as efficiently as possible. C is also used in the official ESP framework, meaning there is plenty of documentation available on using the C language with it. On the other hand, the use of C means a much more careful and precise development is needed, as we have to manage memory manually. While this provides the possibility for more efficient code it also means small errors can greatly affect the whole system, possibly without us knowing.

3.4.1.2 C++

C++ is an extension of the C programming language, it incorporates object-oriented programming features such as classes, inheritance, and polymorphism. C++ is also part of the official ESP framework, meaning it will also have documentation. Due to the complexities added by the C++ language onto C, it will come at the cost of some efficiencies especially due to resource management.

3.4.1.3 Python

Python, unlike the last two languages, is more of a high level language, abstracting many details of the low level hardware, allowing us to focus more on problem solving. There is wide support for development of AI using python, meaning implementing the AI chess system will be easier using python as more resources are available to us. The main problem with the Python language is the requirement of an interpreter, while the ESP32 works directly with C or C++ it will not with Python. This means there will be a longer run time and delay using it compared to C or C++.

Table X:

Language	Compatibility	Efficiency	Ease of Integration
C	High	High	Medium
C++	High	Medium	High
Python	Low	Medium	Low

When deciding on which programming language to base our project off of we looked at three different factors. The first was compatibility, meaning how easily the language can be used with the ESP32, the microcontroller we decided to use. Then efficiency, meaning how efficiently the language interacts with the hardware when it is implemented. Ease of integration was the final point, which found how easy it would be to use the language when programming the system.

Looking at our comparisons we found the compatibility of C and C++ to be high with our ESP32 as they are both officially supported. For efficiency, we took into account the fact that we have to interpret it, which decreases the efficiency of Python and increases delays. Between C and C++, C is faster as there is less overhead from supporting add ons. For ease of integration we put Python at the lowest, as adding the interpreter will cause delays and complexity to arise. C++ was labeled as higher than C due to the complexities surrounding C and memory management. Overall we chose to go with C++ as our main language due to the overall positives involved with it.

3.5.2 Chess Game Logic and AI

A core foundation aspect of the Pathfinder Chessboard is the ability to light the board according to chess rules and the ability to play chess against a computer opponent. To achieve this, we need to implement the chess game logic into our system through the use of the ESP32 microcontroller we decided on using. We need this logic to be accurate to the current chess rules, as well as being efficient to make sure there is very little delay when lifting a piece or when the computer is thinking. Our first decision regarding implementing the game logic was whether to create it ourselves or using a pre-existing package.

3.5.2.1 Custom Made Logic

Building a custom chess logic package for our use would involve researching the rules of chess and implementing them using a programming language into a custom library. This brings several advantages compared to using a pre-existing library, for example we will have full control and customization. This allows us to only put in necessary parts that we need, which would also lead to a lightweight implementation, possibly decreasing the delay rate. Given we

want to implement hardware with the program, we can directly implement any system we desire as we build the library, leading to a more sophisticated result.

While this may seem advantageous, it does come with its own drawbacks. The main factor in our project is time, as we are limited to a handful of months, and implementing a custom library, including the research and debugging which will go into it, would take a large amount of time. We also need to take into consideration our team experience, and while we are all experienced we may find some aspects too daunting or too out of scope, especially when trying to implement a computer player, given the amount of fine tuning these systems need.

3.5.2.2 Pre-existing Package

Our other option is using a pre-existing system which we will research online. This has the great advantage of freeing us the time which would have been spent on making a custom engine. Another reason to use a pre-existing library is it will be proven to work, as many of these libraries have been refined for years, which means we will not have to debug major sections or have to add obscure rules or edge cases. We will also not have to make an AI system for the computer player, which will also save us time and effort which can be spent elsewhere.

A pre-existing library does come with its own disadvantages, however, one being the limited customization and flexibility. Given that the system is already built, we will not be able to easily change the logic if we need to, and getting the system to communicate with the hardware will prove to be additional work. And while this is not a disadvantage, we will have to make sure we conform to any copyright rules or protection that the library we choose has.

Table 10:

Design Method	**Development Time**	**Customization**	**Reliability**
Custom Made	High	High	Medium
Pre-existing	Low	Medium	High

To determine the most suitable design method for the chess engine of the Pathfinder Chessboard we tested them against 3 aspects. We looked at the time it would take to develop the engine, both the game logic and implementing our hardware, how customizable the engine would be, and how reliable the engine could be.

From our comparisons we can see the pre-existing engine would be our best option. Since it is already made we only need to worry about connecting it to our hardware, and while this will certainly take time, making an engine from scratch would take more. For reliability we also saw

a pre-existing engine having an advantage as these engines have been thoroughly tested, for some cases over the course of years, meaning they will be reliable compared to ours which could have overlooked bugs. While a custom made engine provides a high degree of customization that cannot be matched by a pre-existing engine, we have decided the other benefits outweigh this downside, and will be proceeding with a pre-existing engine.

3.5.3 Chess Engine Libraries

Now that we have decided to use a pre-existing game engine we have to choose one. We are looking for an engine that comes with a computer player included, can run efficiently on the ESP32, and is open source so we can modify it as needed.

3.5.3.1 Stockfish

Stockfish is an open source chess engine written in mostly C++. Where Stockfish shines is the strong AI opponent, with it being powerful enough that the engine has won multiple chess tournaments. We can easily change the strength of the computer, making it so both beginners and experienced players will make use of the program. Stockfish is also very reliable, having first been made 16 years ago there have been multiple years of fine tuning and documentation, so any problems that arise could be easily researched. Stockfish is distributed under the GNU General Public License, so we are free to use it for our project.

The main problem with the Stockfish engine is the resource consumption. Given the intensive computer opponent, the hardware requirements for the engine exceed the capabilities of the ESP32, so continuing with this engine will lead to either slow play, which is undesirable, or crashing, which cannot happen. A solution for this would be running the Stockfish engine on an external device, but this is also undesirable as it adds unnecessary complexity and communication.

3.5.3.2 Micro-Max

Micro-Max is an open-source chess engine written in C, and it has been famous for its compact form. Altogether, the engine only holds about 2 KB of code, but is able to play chess at a high level. This means it can easily fit on the ESP32 and run efficiently so there is little delay compared to heavier engines. It is open under the MIT license, meaning we are able to freely use and modify the code as we need to.

One drawback is that because the engine is so compact and heavily optimized, the code can be dense and hard to read, meaning making any modification or additions, such as communication with the hardware, may be difficult or time consuming. The engine is also

relatively new, only having existed for about a year, meaning there will be less documentation and information on the engine, and also increases the likelihood for small bugs or errors.

3.5.3.3 Sunfish

Sunfish is a python based chess engine which, like Micro-Max, is made to be lightweight and efficient, taking up less than 200 lines of code. This compactness is very useful considering we are using the ESP32. We are also able to easily modify the code since it is open source, meaning adding the communication to the hardware will not be too intensive.

However, the fact that this is a python based engine means we would need to port it to a language for the ESP32, this would either be done using a custom port or a separate host computer. This dissolves the advantage of Sunfish being compact, as needed to constantly communicate with another system or translate through a port will put strain on our project, leading to slower communication and delays. The playing strength of this system can also be called into question, as it is weaker than previous examples.

3.5.3.4 Python-Chess

Python-Chess is a chess library for python which is already built into python, meaning we would just need to import it instead of implementing it. Given its age and ease of access there is a strong ecosystem of documentation online, leading to it being easier to modify for our purposes.

This engine faces the same problems as Sunfish, however, since it is python based we would need a way to translate it, either with a custom port or separate host, which would increase the time and complexity. The other problem is the lack of a computer opponent, meaning we would either need to build our own or use a separate AI to play. This is not ideal, if we were to build our own it would be time consuming and complex, especially given our time range, and finding a separate AI to integrate into our project would also bring up more constraints, as we would need to make sure it is compatible with Python-Chess and not resource intensive.

Table 11:

Engine	**Resource Draw**	**Ease of Integration**	**Language**	**Player Strength**
Stockfish	High	High	C++	High
Micro-Max	Low	Medium	C	High

Sunfish	Low	Low	Python	High
Python-Chess	Medium	Low	Python	N/A

To determine the most suitable chess engine, we compared them with 4 major points in mind. The first being resource draw, since we are using an ESP32 microcontroller we need to be careful and strict in how powerful the engine is, one that is too intensive will cause delays in processing which we do not want. Ease of integration refers to how simple it would be to both add the engine into our project and get it working, as well as how simple it would be to alter it to react with our hardware, we are looking for high ease of integration. For the programming language, we want to make sure it is one suitable for this project, and ideally one we have experience with. Finally, the strength of the AI is considered as we want this project to be used by beginners and pros alike.

For resource draw each engine was mostly similar in consumption, with Micro-Max and Sunfish both being made to be optimized, giving them the lowest draw. Stockfish, on the other hand, was shown to be very resource intensive, with the strong AI opponent being the reason, in fact it is so intensive that running it on an ESP32 does not seem feasible, as if we could get it to work it would be too slow for our purposes. The ease of integration for Stockfish, on the other hand, scored the highest. This is due to the fact that it is so well documented, there being years worth of trial and error and modification, and the cleanliness of the code, we would be able to see how the engine works and from that modify it to our needs. The two python based engines, on the other hand, have low ease of integration as described in their initial write ups, we would need either a separate system or custom port to use python. For player strength we looked at how powerful the AI systems could be for each engine, and found them all to be very similar, each engine can provide a variety of play strength, except for Python-Chess, which does not have an AI built into it.

Overall we decided on using the Micro-Max chess engine. This chess engine is the most well-rounded, able to provide a strong AI player, be modifiable, and not be too resource intensive.

3.6 Communication Protocol

Communication between the ESP32 and MCP23017 expanders follows standard I²C addressing conventions. Each expander is assigned a unique hardware address so the ESP32 can identify and read half of the board grid per device. This protocol supports scalable operation with low wiring complexity and reliable sensor polling.

WS2812B LEDs use a precise one-wire serial protocol where the ESP32 generates encoded pulses to specify individual LED colors. The software maps chessboard locations to

LED indices using a row-by-row configuration, simplifying visual control operations. Timing constraints are managed through dedicated drivers that ensure accurate LED data transmission without CPU-blocking delays.

Chapter 4: Standards and Design Constraints

4.1 Engineering Standards and Constraints

Engineering plays a vital role in societal development, driving technological innovation that improves daily life and enables new capabilities. From delivering electricity to entire communities, to creating new vehicles, to designing advanced electronic devices, engineering impacts almost every aspect of modern living. However, with these innovations comes a responsibility to ensure that engineering products operate safely, reliably, and as intended.

The more significant and useful an engineering development is, the greater the potential for harm if it malfunctions or is misused. For instance, an electrical distribution network powers homes and supports modern conveniences, but a failure could range from temporary inconvenience, such as the loss of lighting or climate control, to serious hazards like electrical fires or city-wide blackouts that disrupt essential public services. Similarly, software or hardware defects in consumer products may not result in physical harm but can cause dissatisfaction, breach of advertised specifications, and potential legal or regulatory consequences.

It is to mitigate these risks that formalized standards exist. Standards are documented technical guidelines developed by recognized organizations composed of engineering professionals. They define best practices for design, manufacturing, testing, and implementation, helping engineers ensure compatibility, assess risks, and comply with regulatory requirements. For the Pathfinder Chessboard, adherence to such standards is essential for both electrical and software components. Standards provide guidance in areas such as printed circuit board (PCB) design, soldering, LED integration, and software development, ensuring proper operation, usability, and safety.

In addition to standards, design constraints establish the limits within which a project must operate, including economic, technical, ethical, environmental, and social considerations. By defining both standards and constraints, engineers create a structured framework that guides the development process, reduces risks, and ensures that the final product meets both functional requirements and societal expectations. For the Pathfinder Chessboard, this structured approach guarantees that the product is safe, reliable, user-friendly, and feasible within the defined scope of the project.

4.1.1 Engineering Standards Overview Listing

Ensuring the correct design and implementation of the electrical circuitry is critical to the proper functionality of the product. By consulting various libraries and the resources of

accredited organizations, numerous standards related to Printed Circuit Board design, electrical lighting systems, and other implementations were reviewed. The following is a selection of standards deemed most relevant to the project's objectives and constraints, along with their corresponding descriptions.

A brief outline of the Engineering Standards used for this project.

Standard	Type	Category	Application in Project
IEEE 29119-4	Software	Testing	Ensures robust testing and validation for firmware operation.
BS ISO/IEC 9899	Software	Programming	Defines structure, safety, and consistency for C-based embedded code.
IEEE 802.11	Electrical	Communication	Defines Wi-Fi communication standards for the ESP32.
IEEE 1789-2015	Electrical	LED Safety	Ensures safe PWM control for LED lighting without visual discomfort.
ANSI/UL 8750	Electrical	LED implementation	Sets safety and design standards for the chessboard's LED indicators.
IPC J-STD-001	Electrical	Soldering	Ensures high-quality solder joints and reliable electrical connections
IPC-2221A	Electrical	PCB Design	Governs layout, spacing, and trace routing for the

			chessboard PCB.

4.1.1.1 IEEE 29119-4 (Testing)

We recognize the importance of ensuring that the software powering our smart chessboard operates reliably and consistently under all expected user conditions. IEEE 29119-4 supports this goal by providing a structured set of software testing techniques that we can apply throughout the development to strengthen the quality and dependability of our product.

These techniques directly apply to functions within our system, for example, detecting piece movement, validating legal chess interactions, updating LED indicators, and responding correctly to unexpected user input. By using the techniques mentioned in this standard, we can design test cases that confirm the software behaves correctly with both standard gameplay scenarios and edge cases.

An important benefit of following 29119-4 is that it improves traceability. Each test we write will connect back to a specific requirement established for the project, such as response timing for a moved piece or LED outputs reflecting check or checkmate conditions. This ensures we maintain full coverage over the most critical features of the smart chessboard.

Additionally, the standard emphasizes risk-based testing, which is especially valuable to us. Components such as the Hall-effect sensors, LED lighting feedback, and move-interpretation logic are more likely to affect overall gameplay if they fail. Therefore, we will prioritize more rigorous and frequent testing on those high-impact areas.

4.1.1.2 BS ISO/IEC 9899 (C Programming Language)

BS ISO/IEC 9899 establishes the official specifications and requirements for the C programming language. This standard ensures consistency, safety, and interoperability across C-based systems, which is crucial when software directly interfaces with hardware components. Given that the smart chessboard incorporates an embedded processor responsible for continuous data exchange with sensors, communication modules, and LED indicators, strict adherence to a software standard is essential for maintaining reliable system behavior.

This standard covers critical topics such as proper syntax and structure, approved libraries, data representation, variable scope management, memory allocation, and secure handling of pointers. These guidelines work to minimize common programming issues such as undefined behavior, memory leaks, and timing inconsistencies that could otherwise result in system malfunctions. For example, mismanaging memory while tracking live chess piece

locations or processing player movement could lead to incorrect state representation, affecting gameplay accuracy.

Additionally, BS ISO/IEC 9899 supports maintainability and scalability, qualities that are particularly valuable for a project expected to evolve with added features such as Bluetooth communication or autonomous move recognition. Consistency with this standard also facilitates debugging and future code integration by ensuring the software follows widely recognized norms within the engineering community.

Overall, BS ISO/IEC 9899 will serve as the primary guideline for the programming portion of the smart chessboard. Its role in the software design parallels that of the IPC-2221A standard for the electrical subsystem—each ensures that the design remains robust, safe, and compliant with industry best practices. Following this standard strengthens the reliability of the chessboard's user-facing functionality while supporting long-term development and performance objectives.

4.1.1.3 ANSI/UL 8750 (LED Lighting)

UL 8750 outlines safety requirements for LED lighting systems, including electrical, thermal, and fire protection guidelines. Since the smart chessboard uses LEDs for board indication and user feedback, this standard ensures the lighting components operate safely under normal and fault conditions. Compliance helps prevent hazards such as overheating, electrical shock, and short circuits within the LED driving circuitry.

For this project, UL 8750 influences design choices such as LED current limits, power regulation, proper insulation, and wire routing on the PCB. By following this standard, the LED matrix can function reliably without creating risks for users interacting with the chessboard.

4.1.1.4 IEEE 1789-2015 (LED Lighting)

EEE 1789-2015 provides safety recommendations for LED lighting with a focus on preventing health risks such as flicker-induced headaches, discomfort, or photosensitive responses. Since our smart chessboard relies on LEDs to indicate player turns, piece movement, and special game states, ensuring safe lighting behavior is essential.

This standard supplies guidance on proper PWM frequency and modulation depth to avoid harmful flicker. By applying these recommendations during LED driver design, we ensure that users experience smooth, comfortable lighting during gameplay. Implementing IEEE

1789-2015 also supports long-term reliability of the LEDs while maintaining safety in typical home and competitive environments.

4.1.1.5 IPC J-STD-001 (Soldering)

The IPC J-STD-001, developed by IPC, is one of the most widely recognized and utilized standards in the field of electronics manufacturing.Originally released in the 1980s, it has undergone several revisions to reflect modern advancements in soldering technology, materials, and environmental considerations. This standard establishes the requirements for the quality and reliability of soldered electrical and electronic assemblies, making it directly relevant to the development of the smart chessboard project.

In the context of the smart chessboard, IPC J-STD-001 serves as a critical reference for ensuring that all soldered joints connecting the board's LEDs, sensors, and microcontroller meet industry-accepted standards of workmanship and reliability. Proper soldering directly affects the board's electrical performance, durability, and long-term functionality—key aspects for an interactive product expected to withstand frequent use.

The first section of IPC J-STD-001 defines the acceptance criteria for solder joints, specifying the required size, shape, and quality of each connection. Visual diagrams and detailed examples are included in the standard to help engineers identify acceptable and unacceptable joints. For the smart chessboard, this ensures that connections between the LED matrix, sensor grid, and control circuitry maintain consistent conductivity and mechanical stability. Poorly formed joints could lead to intermittent lighting or sensor failures, reducing the board's responsiveness and reliability.

IPC J-STD-001 categorizes assemblies into three classes based on performance and reliability requirements. The smart chessboard aligns most closely with **Class 2**, as it is a specialized consumer-grade electronic device that demands consistent performance and durability without requiring the extreme reliability of aerospace or military applications.

The standard's soldering methods section describes four primary soldering techniques—hand soldering, wave soldering, reflow soldering, and manual surface-mount soldering. Of these, hand soldering and manual surface-mount soldering are most relevant to the chessboard's assembly. These methods will likely be used to attach through-hole connectors, LED indicators, and small SMD components. IPC J-STD-001 provides specific temperature ranges, tip geometries, and dwell times to prevent thermal damage and ensure high-quality joints—guidelines that directly apply to the manual assembly phase of the project's PCB.

The environmental considerations section of the standard emphasizes lead-free soldering and the reduction of hazardous substances in compliance with global environmental regulations such as RoHS. This aligns with the team's goal of producing an environmentally conscious product that adheres to sustainable manufacturing practices.

Lastly, the rework and repair section outlines safe and effective methods for fixing soldering defects or replacing faulty components. These guidelines are critical for the smart chessboard, as they ensure that re-soldering sensors or LEDs during testing or maintenance does not damage the PCB or surrounding traces.

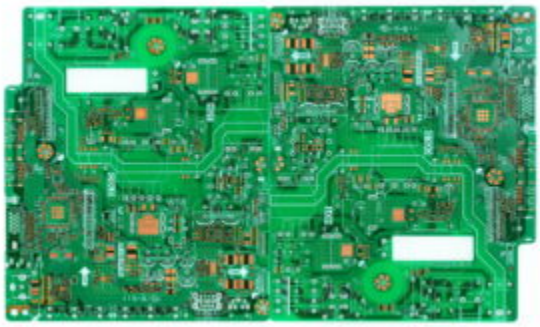
By adhering to IPC J-STD-001, the smart chessboard project ensures that all soldering work achieves high levels of reliability, mechanical integrity, and environmental compliance. Following this standard minimizes the likelihood of cold joints, solder bridges, and thermal stress damage—factors that could otherwise compromise the device's accuracy and performance. Overall, IPC J-STD-001 supports the project's objective of delivering a durable, professional-grade electronic system that meets established engineering quality benchmarks.

4.1.1.6 IPC-2221A (PCB Design)

The IPC-2221A standard, titled *Generic Standard on Printed Board Design*, was developed by IPC, an association dedicated to creating and overseeing standards governing printed circuit board (PCB) design. As the name suggests, this standard provides general guidelines for designing printed circuit boards (PCBs) and is highly relevant to the smart chessboard project. Since the smart chessboard relies on a custom PCB to integrate sensors, LEDs, and a microcontroller, adherence to IPC-2221A ensures reliability, manufacturability, and electrical performance.

The general requirements outlined in IPC-2221A help guide design decisions for the chessboard PCB, including tradeoffs between board size, layer count, and routing complexity. For example, proper line spacing and trace width recommendations ensure that signals from touch or pressure sensors do not interfere with each other, while maintaining the compact size needed to fit beneath the chessboard squares. Environmental considerations, such as heat generated by LEDs or the microcontroller, are also addressed in the standard, guiding choices for materials and layout to prevent overheating during extended gameplay.

Figure X:



Material selection guidelines are particularly applicable to the chessboard design. The standard emphasizes temperature tolerance, electrical properties, and structural strength, which are crucial for ensuring that the PCB withstands repeated stress from piece placement and potential accidental impacts. Selecting appropriate dielectric materials and adhesives, as suggested by IPC-2221A, helps maintain board integrity and signal consistency over time.

Mechanical and assembly considerations, such as board shape, rigidity, and component placement, directly impact the chessboard's manufacturability and durability. IPC-2221A recommendations for uniform board sizes, bow and twist limits, and vibration resistance are essential when designing a board that may be moved or handled frequently. The standard also guides component placement and datum reference use, which helps accurately position LEDs, sensors, and connectors to align perfectly with the chessboard squares.

Electrical properties recommendations, such as proper conductor spacing, power distribution, and crosstalk reduction ensure that the chessboard's sensors and LEDs operate reliably without interference. Transmission line techniques and electrical clearance guidelines help maintain signal integrity, especially in areas where multiple sensor traces run close together.

Thermal management principles from IPC-2221A are applied to prevent hotspots from LEDs or the microcontroller. Proper heatsinking and component spacing improve heat dissipation and contribute to long-term reliability. Similarly, quality assurance recommendations, including testing points and inspection coupons, allow validation of both electrical and mechanical functionality before final assembly.

Finally, hole and interconnection guidelines help define via types, drill sizes, and pad placements for mounting the microcontroller, connectors, and other components, ensuring structural strength and precise alignment. Documentation practices suggested by the standard facilitate future revisions or repairs of the chessboard PCB.

By following IPC-2221A, the smart chessboard's PCB can achieve a balance of reliability, manufacturability, and performance, ensuring that sensors, LEDs, and the control circuitry function correctly while withstanding mechanical stress and thermal effects. This standard serves as a foundational reference for designing a robust, high-quality PCB tailored to the chessboard's unique interactive requirements.

4.1.1.7 IEEE 802.11 (Wi-Fi)

As a stretch goal for our smart chessboard, we plan to explore wireless connectivity using the IEEE 802.11 standard. IEEE 802.11 defines specifications for wireless local area networks (WLANs), including Medium Access Control (MAC) and Physical Layer (PHY) protocols. Implementing Wi-Fi would allow the chessboard to communicate with mobile devices, online apps, or other boards for move logging and gameplay tracking.

While Wi-Fi functionality is not part of the core requirements, referencing IEEE 802.11 provides a framework for future development. Key considerations include reliable data transmission, low-latency updates, interference management, and proper antenna placement. By planning around IEEE 802.11 standards, we ensure that any future wireless implementation can meet common connectivity and interoperability expectations.

4.1.2 Design Constraints

In addition to adhering to established engineering standards, every project is influenced by inherent design constraints—factors that limit the team's options and define the boundaries of the final product's functionality and performance. These constraints may be quantitative, such as numerical limits on cost or dimensions, or qualitative, such as requirements for accessibility or user experience.

For the smart chessboard project, the most significant design constraints fall into two main categories: Economic and Social. The economic constraint defines the financial limitations of the project, including component costs, fabrication, and development resources, ensuring the project remains feasible within the available budget. The social constraint emphasizes the need for the product to be interactive, accessible, and assistive, guiding design decisions to maximize usability and positive social impact.

4.1.2.1 Social

A central objective of our project is to deliver an accessible and assistive chess experience by incorporating multiple game modes and an LED-based lighting system that guides piece movement. These features are intended to make gameplay more interactive and convenient, particularly for users who may face challenges with traditional chessboards, thereby creating a positive social impact.

The product should emphasize user-friendliness and interactivity. The motivation behind selecting this project was to design a chessboard that is simpler, more versatile, and more accessible than existing options. By offering several game modes, players can quickly select their preferred game, enhancing the board's adaptability and overall user experience.

Accessibility is a key foundation of this interactive experience, which introduces constraints on how the product must operate. One major social goal is to implement an assistive system that goes beyond conventional guidance, actively supporting players during gameplay. This ensures the board meets the intended objectives of interactivity, accessibility, and user assistance.

This assistive functionality is realized through a lighting system embedded in the board. The LEDs highlight specific squares and possible movement paths for selected pieces, helping players visualize strategies and tactics. By providing this visual guidance, the smart chessboard enhances both interactivity and accessibility, fulfilling the project's core goals.

4.1.2.2 Economic

The total estimated cost of the smart chessboard project is about **$325**, which is fully self-funded by the team members. This budget includes the costs of all components, tools, fabrication, and other necessary materials.

The main expenses are associated with custom PCB fabrication ($150), 3D printing filament ($50), and development boards ($20). Key electronic components, including the ESP32 microcontroller, I/O expanders, LEDs, and Hall effect sensors, account for a smaller portion of the total cost. The remaining budget covers additional materials such as the LCD screen, magnets, chess pieces, and frosted acrylic top, as well as minor costs for tools and miscellaneous items.

To minimize further expenditures, the team will utilize University of Central Florida laboratory facilities, including available tools and equipment, reducing the need for additional purchases. Free software resources and online design tools will also be leveraged for development, simulation, and calculations wherever possible.

By adhering to this budget, the project remains financially sustainable while covering all necessary components for the construction of the smart chessboard.

Chapter 5: Application of ChatGPT and Other Similar Platforms

The integration of artificial intelligence (AI) platforms such as ChatGPT, Copilot, and Claude, into engineering projects has increasingly become a valuable resource for research, development, and documentation. These platforms utilize advanced natural language processing to interpret technical queries, generate code, explain complex concepts, and assist with technical writing. In the context of the Smart Chessboard project, which involves embedded systems, custom PCB design, sensor integration, and software-hardware interfacing, AI platforms provide both guidance and support across multiple stages of development. The chapter explores the practical applications, advantages, and limitations of these tools within the project, while highlighting specific examples where they contributed to the design, implementation , and research documentation process. By evaluating both the benefits and potential risks of using AI in engineering workflows, this chapter provides insight into how such platforms can enhance learning and productivity in an engineering workspace.

5.1 Limitations, Pros, and Cons

5.1.1 Pros

The use of AI platforms in the Smart Chessboard project offers substantial advantages across multiple areas, including technical research, coding, design conceptualization, and documentation. One of the most notable benefits was the ability to rapidly access and synthesize complex technical information. For example, ChatGPT provided comprehensive comparisons of microcontroller options, such as the ESP32 or Arduino Nano, outlining differences in processing power, I/O availability, communication protocols, and energy efficiency. This rapid synthesis enabled informed component selection early in the design phase, significantly reducing the time otherwise spent on manual research. AI guidance also facilitated the sensor selection and integration process, summarizing the operational characteristics of hall-effect sensors, reed switches, and infrared detectors, while explaining trade-offs in accuracy, response time, and compatibility with the selected microcontroller. Beyond research and component selection, AI platforms assisted in software development by suggesting efficient algorithms for move detection, board state tracking, and communication between the microcontroller and PC-based GUI. Code templates, debugging tips, and guidance on error handling accelerated development and supported modular, maintainable software design. Finally, AI platforms enhanced technical writing and documentation by helping organize experimental results, draft clear explanations of system behavior, and format tables summarizing sensor accuracy, latency, and interface responsiveness.

5.1.2 Cons

Despite the advantages, AI platforms presented several drawbacks. Outputs were sometimes too generic or not perfectly tailored to the project's specific hardware or software configuration. For instance, code snippets occasionally reference incompatible libraries, generic pin assignments, or outdated methods, requiring manual adaptation. PCB layout suggestions were theoretical and needed verification through hands-on prototyping. Another drawback is the potential for overreliance on AI; although platforms offer ready-made solutions, depending too heavily on these suggestions can reduce independent problem-solving skills, critical thinking, and experiential learning, which are essential in engineering design projects. Additionally, AI cannot perform physical testing or validate experimental results, meaning the engineer must manually test sensor placement, move-detection accuracy, and hardware reliability to ensure correct operation.

5.1.3 Limitations

AI platforms also have intrinsic limitations that must be recognized. Accuracy is a primary concern, as AI-generated outputs may contain mistakes, outdated information, or incomplete recommendations. For example, suggested move-detection algorithms or sensor polling strategies might work in theory but be impractical for the specific combination of microcontroller and sensors used in the Smart Chessboard. Context-awareness is another limitation; AI cannot fully account for physical constraints such as PCB size, trace routing restrictions, power distribution, or interference between components. As a result, advice regarding component placement or signal routing often requires significant adaptation during prototyping to achieve reliable performance. Furthermore, AI cannot consider user-specific interactions, such as the speed of piece movement, weight, or potential vibrations on the board, all of which can affect sensor accuracy and responsiveness. AI also cannot conduct real-world testing or observe system behavior, so all theoretical guidance must be verified through hands-on experimentation, iterative debugging, and calibration. Finally, AI cannot anticipate complex interactions between software and hardware, such as timing delays, microcontroller processing bottlenecks, or electrical noise, which are only detectable during actual implementation. These limitations emphasize that while AI platforms are valuable tools for guidance, they cannot replace critical thinking, practical validation, or iterative design in complex engineering projects like the Smart Chessboard.

Aspect	Pros	Cons	Limitations
Research and Design	Rapid access to component	Occasional mismatches in real	Must verify all part specifications

	specifications, standards, and integration examples.	datasheet specs.	manually before selection.
Coding Assistance	Provides debugging help and code structure examples.	May produce non-functional or incomplete snippets.	Needs validation on local compiler or IDE.
Documentation	Speeds up report writing and formatting.	May overgeneralize or use filler content.	Requires careful review to ensure project-specific relevance.
Learning Support	Helps explain unfamiliar technical terms and concepts	Risk of over-reliance can limit independent problem-solving.	Should supplement, not replace, fundamental learning.

Figure 5.1: AI generated table (Pros/Cons/Limitations). Prompt in Appendix E

5.2 Examples of Platform Use

AI platforms played a significant role throughout the beginning stages of the Smart Chessboard project, providing guidance, support, and efficiency improvements. One of the earliest and most impactful applications was during component selection and research. Assisting in evaluating microcontrollers, such as the ESP32 and Arduino Nano, by summarizing differences in their technical specifications. This rapid synthesis allowed the team to make informed decisions early in the design phase and reduced the need for extensive manual research.

Another practical example of AI support was in sensor selection. The team required sensors that could accurately detect chess pieces while remaining compatible with the ESP32 microcontroller. ChatGPT provided a comparative analysis of hall-effect sensors and capacitive sensors, highlighting trade-offs in detection accuracy, response time, integration complexity, and power consumption. Based on this guidance, the team selected a hall-effect sensor array optimized for the EP32's capabilities, ensuring reliable move detection with minimal software overhead. Although the AI recommendations were theoretical, they effectively narrowed the options and guided prototyping, saving time and reducing trial-and-error experimentation.

Feature	ESP32	Arduino Nano	Raspberry Pi

Processor	Dual-core 32-bit Xtensa LX6 @ 240 MHz	8-bit ATmega328P @ 16 MHz	Dual-core ARM Cortex-M0+ @ 133 MHz
RAM/ Flash Memory	520 KB SRAM / up to 16 MB Flash	2 KB SRAM / 32 KB Flash	264 KB SRAM / 2 MB Flash
Connectivity	Built-in Wi-Fi and Bluetooth	Requires external modules for Wi-Fi/Bluetooth	No native Wi-Fi; requires Pico W variant
Power Efficiency	Moderate; supports deep-sleep modes	Low power; suitable for battery use	Efficient but limited wireless options
GPIO Pins	30+ configurable I/O pins	14 digital I/O, 6 analog inputs	26 multi-function GPIO pins
Programming Language Support	C, C++, MicroPython	C, C++	C, C++, MicroPython
Advantages	Excellent performance; built-in wireless; strong library support	Simple, reliable, easy to prototype	Low cost; fast and power efficient
Disadvantages	Slightly higher power draw and complexity	Limited memory and connectivity	Requires external Wi-Fi module
Project Evaluation	Selected for superior performance, built-in wireless communication, and multitasking support	Insufficient memory and connectivity	Suitable backup if wireless integration fails

Figure 5.2: AI generated table (Microcontroller Comparison). Prompt in Appendix E

Sensor Type	How it works	Advantages	Disadvantages	AI Evaluation for Project

Hall Effect Sensor	Detects magnetic fields from small magnets placed in each chess piece	Reliable detection, fast response, easy to connect to ESP32.	Requires a magnet in each chess piece.	**Best Option** – Accurate and consistent for chessboard grid detection.
Capacitive Sensor	Measures small changes in capacitance when a piece is near.	Non-contact sensing, sensitive	Easily affected by board material and humidity.	**Possible Option** – Works, but less reliable on wooden surfaces
Piezoelectric Sensor	Generates voltage when pressed or tapped.	Simple and low power.	Needs physical contact; cannot detect stationary pieces.	**Not Suitable** – Only detects movement or pressure.
Ultrasonic Sensor	Uses sound waves to detect nearby objects.	Detects presence over a distance.	Low precision, bulky for small grid layout.	**Not Suitable** – Too coarse for detecting individual squares.

Figure 5.2.1: AI generated table (Sensor Comparison). Prompt in Appendix E

AI platforms were particularly valuable in comparing programming languages suitable for the ESP32 microcontroller—primarily C++, Python, and C. ChatGPT provided detailed comparisons of each language's performance characteristics, ease of use, and compatibility with embedded systems. C++, as used in the Arduino IDE, was recommended for its balance of high-level abstraction and efficient execution, supported by a vast ecosystem of libraries and community documentation. It offered simplicity in syntax and strong integration with sensor libraries, making it ideal for implementing the Smart Chessboard's move detection, LED control, and communication routines. Python was noted for its fast development cycle and ease of debugging, making it an excellent choice for rapid prototyping and testing algorithms before full implementation. However, it suffers from slower runtime performance and higher memory consumption, which may limit scalability for complex multitasking or timing-sensitive operations. In contrast, C (using the ESP-IDF framework) provided the most direct access to hardware and the highest execution efficiency. It enabled precise timing control and resource optimization but required more extensive setup, a steeper learning curve, and longer

development times. Based on this comparison, the team adopted C++ for the main implementation, reserving Python for preliminary tests and recognizing C as a potential option for future optimization if higher performance were required. This AI-assisted comparison streamlined the decision-making process, helping the team select a language that balanced development speed, maintainability, and hardware efficiency.

Option	Advantages	Disadvantages	Project Application
C	Fast execution; direct hardware access; memory-efficient.	Requires manual memory management; less abstraction.	Used for low-level ESP32 control and real-time data handling.
C++	Object-oriented; integrates libraries easily; supports modular development.	Slightly more resource-heavy than C; steep learning curve for embedded systems.	Used for developing structured firmware and modular logic.
Python	Easy to read and debug; Wide AI/ML library support	Requires interpreter; slower and not ideal for real-time microcontroller control.	Used for initial testing and algorithm prototyping before firmware deployment.

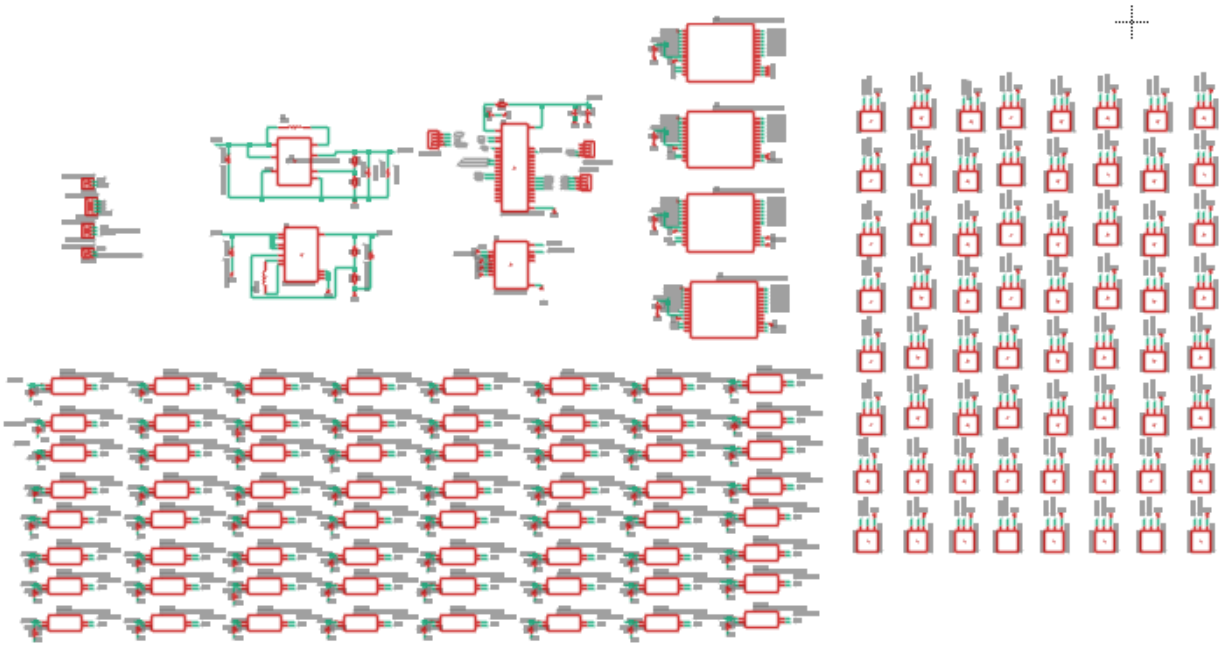
AI platforms also proved beneficial during software development, particularly in implementing move-detection algorithms, board state management, and communication between components. ChatGPT provided algorithmic guidance for sensor polling, validation of chess moves, and real-time updates to the graphical interface. It suggested code templates, debugging strategies, and modular programming structures that allowed faster integration of hardware and software while reducing errors.

Overall, these examples demonstrate that AI platforms significantly enhanced the efficiency and effectiveness of the Smart Chessboard project. They accelerated research, supported coding, informed programming language selection, improved technical writing, and provided conceptual design guidance. However, every AI-generated recommendation required careful validation and adaptation to ensure compatibility with the project's specific hardware, software, and user-interaction constraints.

Chapter 6: Hardware Design

This section details the hardware design of the entire system. More specifically how each component interacts with one another. The system can be divided into 6 subsystems. One of the core subsystems being the power delivery/electrical system that explains how each individual component receives their required input voltage. Another critical design schematic to examine is the Microcontroller and how it will send and receive signals. Directly connected to the microcontroller are the I/O extenders and the logic level shifter. These two systems correlate to the hall effect sensors and the LED's respectively.

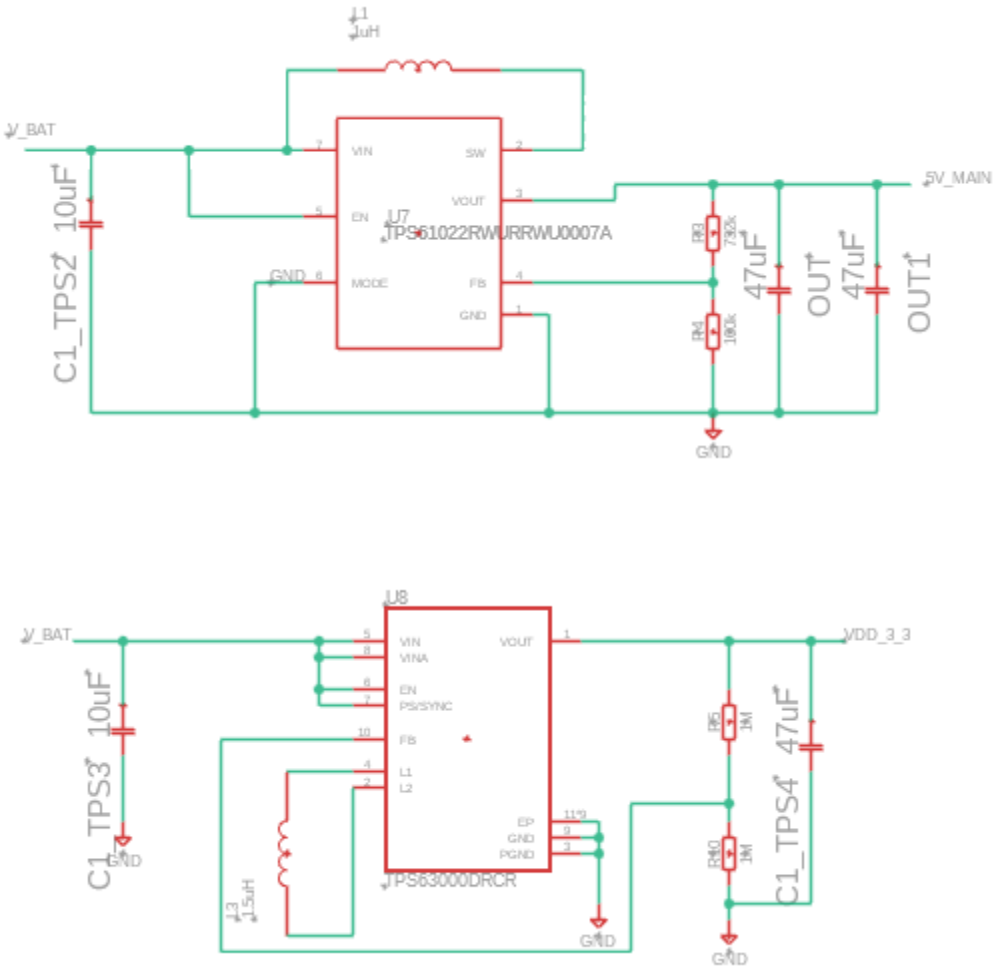
Figure X: Overall Schematic Diagram



6.1 Power Delivery/ Electrical Power system

The power Delivery system is powered via 3 C Alkaline batteries in the Pathfinder Chessboard side compartment. The switch can complete the circuit when on, allowing for power to flow through.Power then flows through the 4.5 v batteries to both a buck converter and buck booster. These converters are responsible for the 3.3V and 5 V regulation needed throughout various components throughout the circuit. Specific components such as the potentiometer, ESP32, hall effect sensors, and I/O expanders all required the regulated 3.3 volts from the buck converter for this design. The buck booster on the other hand was used to step up the 4.5 V to 5 V in order to power the LEDs and logic shifter.

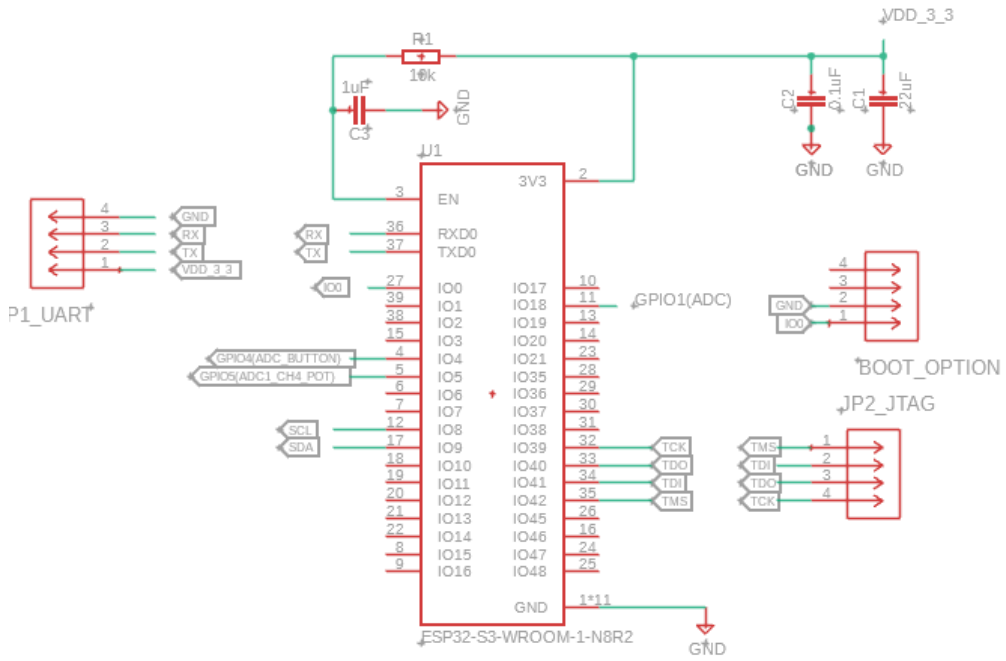
Figure x: Bottom: 3.3 V Buck converter schematic, Top: 5 V Boost converter schematic



6.2 Microcontroller

The microcontroller controls the software that coordinates the entire system and in turn has to send and receive signals from the components being used such as the LCD, potentiometer, button, LEDs, and sensors. In order to achieve this certain GPIO are labeled in order to transmit or receive data. The ESP32-S3-WROOM-1 has a lot of freedom to assign what the GPIO pins can do. Most of the GPIO pins on the microcontroller can be connected to the I2C pins, in this case they are connected to GPIO pin 8 for SCL and 9 for SDA. Other pins being used include GPIO 4 and 5 which are analog pins for the potentiometer and button respectively.

Figure X: Microcontroller Schematic

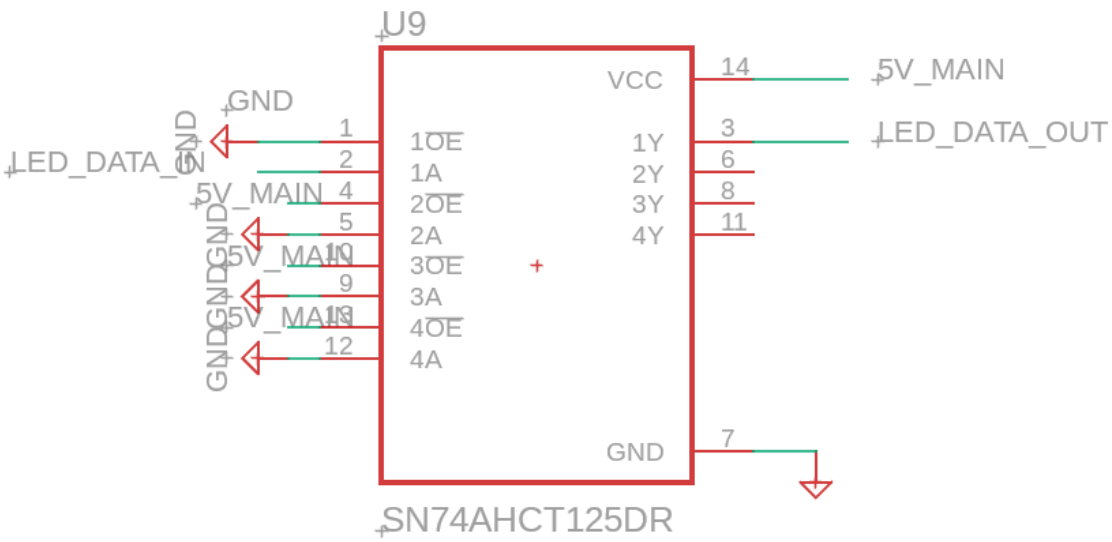


6.3 Logic level Shifter

The SN74AHCT125 is a crucial component for our project because the WS2812B requires the dataline to be within a narrow voltage range of Vdd-0.5 to Vdd+0.5V. Since our LEDs are powered by 5V, the data signal must be close to 5V. The ESP has a logic output of only 3.3V, which is not enough to meet the LEDs specifications.

The SN74AHCT125 acts as a logic level shifter for the ESP32's data line and shifts it from 3.3V to 5V. The SN74AHCT125 is an active component that takes a 5 V Vcc supply. It works by taking the logic signal given to its input and mirroring it at the output at the Vcc voltage level without losing any logic state. To ensure the buffer is always driving the signal its $\overline{OE}$ (output enable) must be tied to ground due to this pin operating in an active-low mode. The remaining three unused channels have their $\overline{OE}$ pins tied to Vcc(5v) to keep them disabled and preventing float inputs.

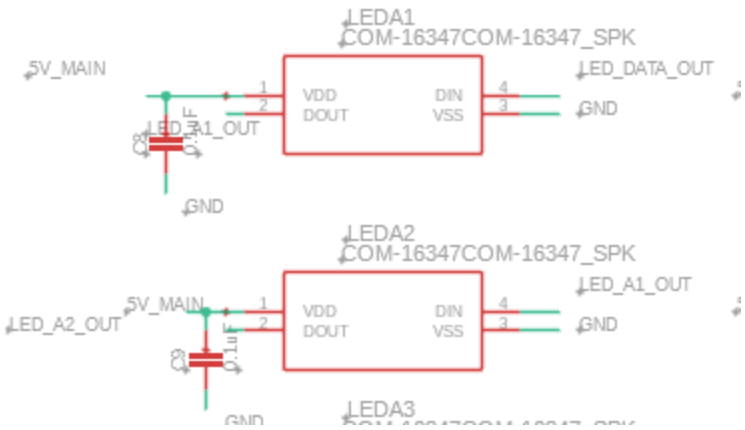
Figure X: Logic level shifter schematic



6.4 LED Design

Each of the 64 WS2812B LED's are powered in parallel by the 5v rail. Each LED has a dedicated 0.1µF for decoupling, which is essential to prevent voltage instability caused by the rapid current draw of the LED circuit. The 5V logic data signal required by the LEDs comes from the SN74AHCT125 level shifter. This single data line is connected to the Din pin of the first LED. The WS2812B are then connected in a daisy-chain architecture, where the Dout pin of one LED connects directly to the Din pin of the next pin which allows the ESP32 to individually address and control the color and brightness of all the squares on the chess board whilst using only one GPIO pin. Below is the image of the first two LEDs out of the 64 LEDs which the first LED receiving the LED_DATA_OUT line and sending the LED_A1_OUT line to the next LED.

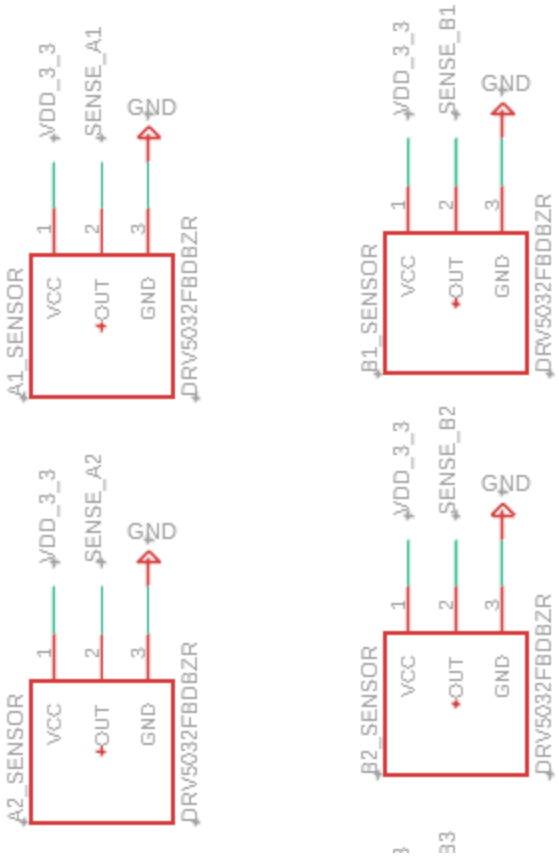
Figure X: LED Schematic



6.5 Hall Effect Sensors

The 64 hall effect sensors are a crucial part in indicating where a piece is on the chessboard. To achieve this each hall effect sensor is named and linked to an individual GPIO pin on the I/O extenders. For example if a piece is but on the C5 square the sensor would output a data signal to GPA5 of the I/O extenders labeled SENSE_C5. Each capacitor is also powered by the 3.3 volt rail. Below is the image of sensors labeled A1, A2, B1 and B2 out of the 64 sensors.

Figure X: Sensor Schematic



6.6 I/O Expanders

The I/O expanders are used to expand the amount of GPIO pins the MCU can control. This was done due to our project needing to control 64 hall effect sensors and therefore requiring 64 GPIOs. In order for the MCU to tell which I/O expander is which, the I/O expanders must have a unique address. The addresses are configured with the A0-A2 pins. These pins are either set to ground or the voltage source. Below is the image of the first two I/O expanders.

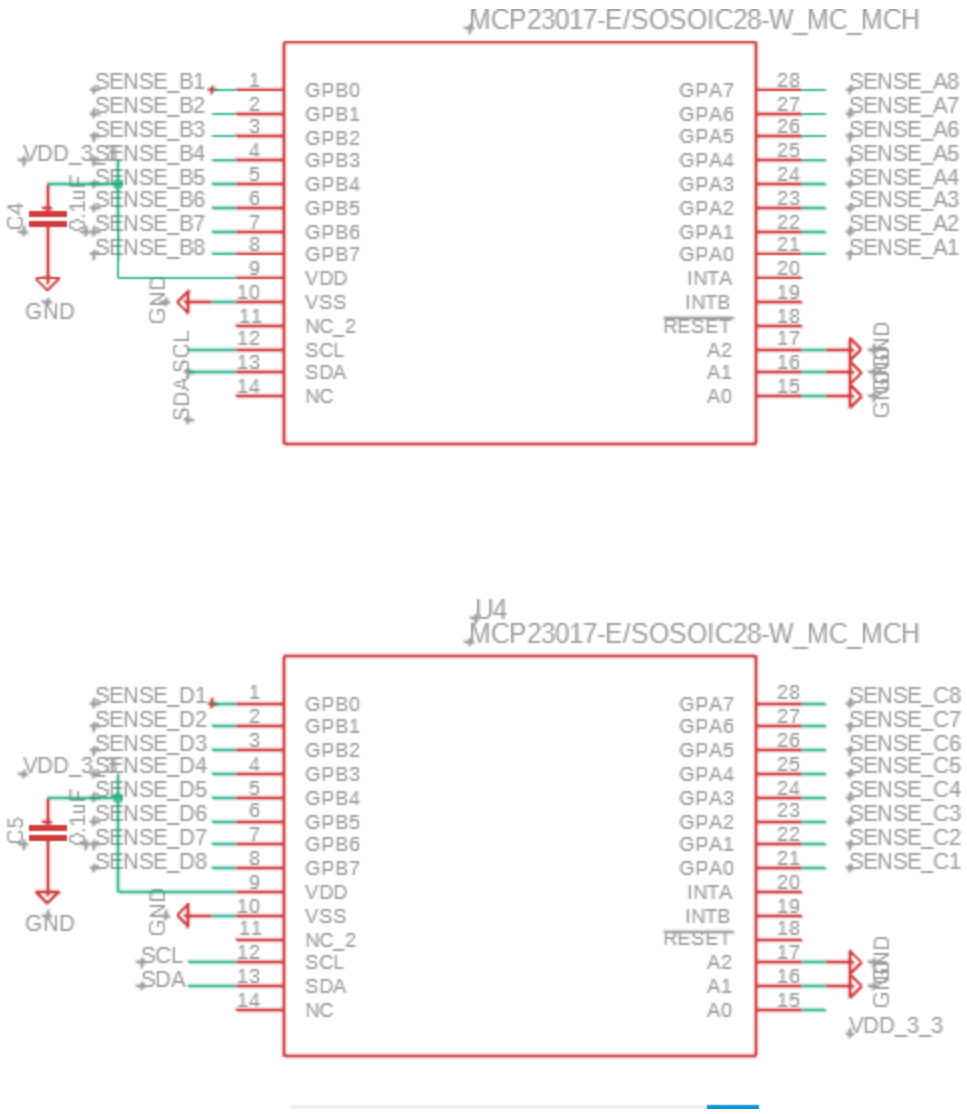


Figure X: I/O Expander Schematic

Chapter 7: Software Design

(Approx. 10-20 pages)
Include detailed software diagrams at the subsystem/component level (flowcharts, state diagrams, class diagrams, etc.).

This section details the software design of the Pathfinder Chessboard, most notably how the chess engine interacts with the rest of the hardware present in the system. We also show how the system takes in player input to set up the board, and how this then affects the system. The system features an ESP32 microcontroller that integrates real-time sensing, rule enforcement, and LED-based user feedback into a single embedded platform. Since the chessboard contains 64 Hall-effect sensors for piece detection and 64 WS2812B addressable LEDs for visual feedback,

the software architecture must ensure reliable measurement of sensor states and precise control of LED patterns. The implementation uses ESP-IDFs, interrupt-driven sensor reading via MCP23017 I/O expanders, and a data structure model for board logic and legal move validation. The focus of the software design is accurate rule enforcement and low-latency response to physical player interaction.

7.1 System Architecture

The Pathfinder Smart Chessboard software architecture is built around an ESP32 microcontroller that executes the full game logic, monitors board state using an array of Hall effect sensors, and drives WS2812B LEDs for visual feedback. All processing occurs locally, without reliance on external connectivity, ensuring that the system can enforce chess rules. The ESP32 interfaces with two MCP23017 I/O expanders over the I²C protocol to acquire magnetic field readings from all 64 DRV5032 Hall effect sensors. These readings enable the system to determine when and where a piece is picked up or placed. Once a potential move is detected, the chess rule engine will validate legality, update the internal board representation, and trigger appropriate LED behavior to guide the players.

7.1.1 LED Configuration

The LED subsystem provides visual feedback by controlling a daisy-chained strip of 64 WS2812B addressable LEDs, one under each board square. These LEDs will be indexed linearly in a row to row mapping pattern so the software can translate a board coordinate (row,column) into a LED address. The subsystem highlights valid destinations when a piece is lifted, uses distinct patterns for illegal moves, and transitioning players' turns. Additionally, states such as check or checkmate should trigger a unique lighting sequence to alert players. All LED operations will be synchronized with gameplay to ensure intuitive and responsive interaction.

7.1.2 Sensor Configuration

The system is responsible for sampling sensor data and detecting piece movement events. Each Hall effect sensor corresponds to a specific board square and outputs a signal indicating whether a magnetic chess piece is present.The ESP32 reads these signals by polling two MCP23017 expanders via I²C, allowing access to all 64 input states. A motion detection algorithm compares the current sensor matrix with the previously recorded matrix to identify piece removal (source square) and placement (destination square). This subsystem abstracts raw sensor data into high-level actions, enabling the game logic to process movement without directly handling hardware-level details.

7.2 User Interaction Logic

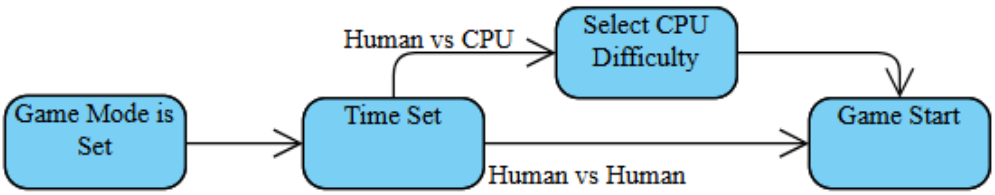
The LED interface serves as the primary user interaction mechanism. Highlighting legal move options educates inexperienced players and speeds up competitive play. Flashing signals and corrective animations guide users during errors while preserving natural game flow. Turn-based color indicators will help visually differentiate sides, and event-based patterns, such as animations for captures or checkmate enhance engagement and clarity. This feedback ensures both usability and accessibility.

7.2.1 Chessboard Initial Setup

When the user first turns the system on, they are met with multiple options. The first of which is selecting which mode to play as, that being human vs human or human vs computer. This is handled through the use of a potentiometer, where a value above 50% indicates human vs human play and a value below it indicates human vs computer. The player then presses a separate button to lock this in and move onto the next setting.

For both modes the player then needs to select a time limit. This indicates how long the game should go on for, with a range of up to an hour, and is once again handled by the potentiometer. When set to the lowest value, however, there is no time limit on the game.

In human vs computer mode, the potentiometer is once again used to determine the level of the CPU, as the values increase with the potentiometer so does the difficulty of the CPU.
Figure X:

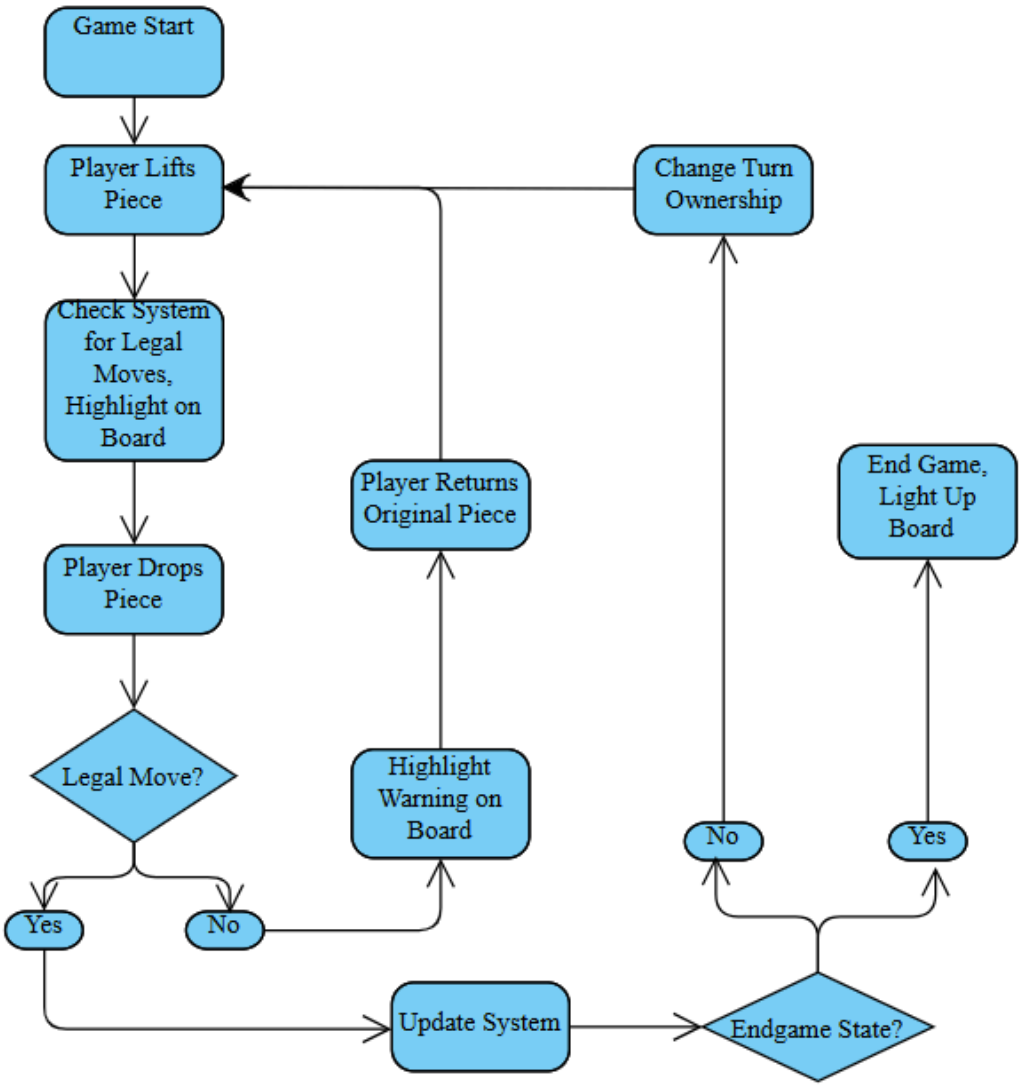


7.2.2 Game State Handling

The game begins in a ready state where no piece is currently selected. When a player lifts a piece, detection of a legal selectable square transitions the system into a state highlighting all valid destination squares. The next state involves waiting for placement; if that placement

results in a legal move, the system transitions into a state that updates game records, toggles turn ownership, evaluates whether the resulting board is in check or checkmate, and then returns to the ready state. Should the placement be invalid, the state machine reverts to the ready state while triggering a visual warning pattern. By following this finite-state model, the firmware prevents ambiguous player interactions and ensures that only rule-compliant moves are allowed to alter the game board.

Figure X:



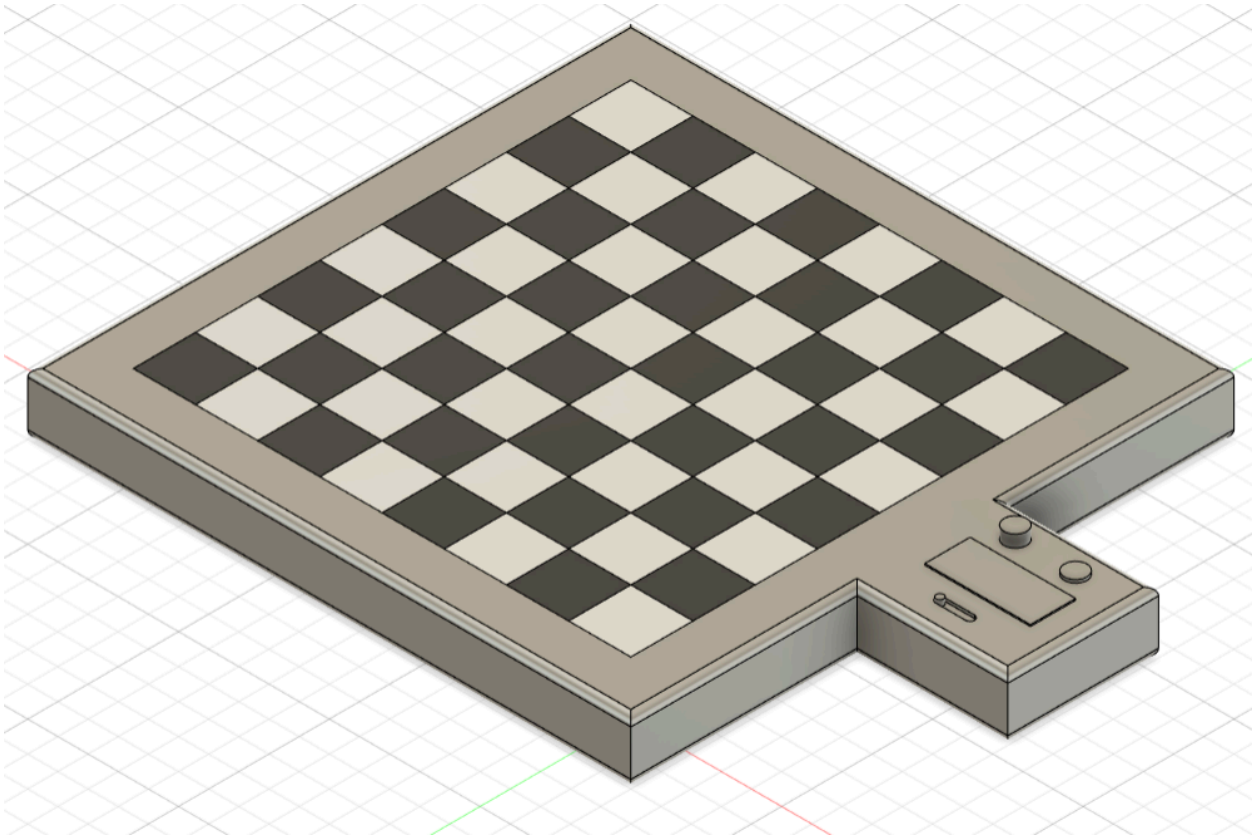
When playing against a computer, the main difference comes from the fact that the player must manually move the computer's piece. The check for a valid move happens on the system, and instead of lighting up the possible paths for any move, the only square which lights up is the one in which the computer has chosen. When the player picks up the piece and sets it down again, there is another check to make sure the piece is on the right spot, and if it isn't, a warning is shown on the board similar to if a player took an illegal move.

Chapter 8: System Fabrication/Prototype Construction

8.1 Mockup 1

Pictured here is our first mockup of the chessboard. It is done using Fusion360 and the specifications are from the original proposal. Using this mockup we already have modifications planned, such as making the overall model smaller to help mimic the size of a standard chessboard, along with increasing the size of the side compartment to accommodate our electronics and to make it seem less cramped.

Figure X:



Chapter 9: System Testing and Evaluation

(Approx. 5-10 pages)
Detail your testing plans and performance evaluations for both hardware and software, and discuss your plan for SD2.

Chapter 10: Administrative Content

10.1 Bill of Materials

Component:	Part name:	Quantity:	Unit Price:	Total Price:
Microcontroller	ESP32-S3-WROOM-1-N8R2	1	$5.71	$5.71
Button	TS02-66-70-BK-160-LCR-D	1	$0.1	$0.1
Switch	EG5619-ND	1	$0.71	$0.71
Potentiometer	09806 Trim Pot	1	$1.25	$1.25
Batteries	C alkaline batteries	3	$2.31	$6.93
Battery Holder	3-C battery holder	1	$4.49	$4.49
Development Board	Arduino uno	1	$20	$20
I/O Expander	MCP23017	4	$1.62	$6.48
3.3 voltage regulator	TPS63000DRCR	1	$2.61	$2.61
5 Volt regulator	TPS61022RWUR	1	$1.68	$1.68
Logic shifter	SN74AHCT125DR	1	$0.39	$0.39
LCD screen	I2C 1602 LCD Display	1	$8.95	$8.95
LEDs	WS2812B addressable LEDs	64	$0.267	$17.09
64 Hall effect	DRV5032	64	$0.276	$17.66

Sensors				
Frosted acrylic top	1	1	$17.99	$35.98
32 chess pieces	Chess Pieces	32	$8.73	$8.73
Magnets	Neodymium magnets	32	$0.07	$6
3d Print fillament	3d print filament	1	~$50	~$50
Custom PCB	PCB	1	$150	$150
TOTAL COST:				$325.43

Estimated Costs:
- Components: $54.72
- Tools: $20
- Fabrication: $235.98
- Other:$14.73

Funding Sources:
- Self-funded: $325.43

10.2 Project Milestones (SD1 & SD2)

Senior Design 1 Task	**Expected Completion**	**Progress**
Divide & Conquer Document	9/5/2025	Completed
First Meeting with Mentor	9/10/2025	Scheduled
Research Chess Libraries (Solo and CPU)	9/20/2025	Researching
Start Gathering Materials	10/01/2025	In Progress
Midterm Report	10/31/2025	In Progress
Finish Breadboard Prototype	11/03/2025	Not Started

Finish PCB Prototype	11/15/2025	Not started
Final Report	11/25/2025	In Progress
Mini Demo Video	11/25/2025	Not Started
Committee Meeting	December 2025	Not started

Senior Design 2 Task	**Expected Completion**	**Progress**
Order PCB	December 2025	Not Started
Construct PCB	December 2025	Not Started
PCB and Software Testing	January 2026	Not Started
General Testing and Bug Fixing	January 2026	Constant
Meeting with Mentor	February 2026	Not Started
Project Completion	April 2026	Not Started
Final Demo Video	May 2026	Not Started
FInal Check Before Presentation	May 2026	Not Started
Final Presentation	May 2026	Not Started

10.3 Work Distribution

Responsibility	**Member(s)**
Hardware Selection	Andres Armendariz
PCB schematic	Andres Armendariz
Website	Ryan Ramdihal
Review Committee Engagement	Ryan Ramdihal
Software Research	Freddy Pacheco

Chapter 11: Conclusion

(Approx. 2-3 pages)
Summarize the project, its outcomes, and the progress made during Senior Design I.

Appendix A: References

Matthew (MisterCutie). "History of Chess: The Basics." Chess.Com, 28 Jan. 2009,] https://www.chess.com/article/view/the-history-of-chess.

DIY Machines. (2021, March 25). DIY Super Smart Chessboard | Play Online or against Rasp Pi. Hackster.io. Retrieved September 3, 2025, from Hackster.io website. https://www.hackster.io/diy-machines/diy-super-smart-chessboard-play-online-or-against-rasp-pi-b57daa

*Andres Perez. "UCF Fall 2023 Senior Design 2 Group 13 Final Demo Presentation." *YouTube* 29th November 2023,* https://www.youtube.com/watch?v=n4JKfAw31GQ.

Stockfish. Stockfish – Strong open-source chess engine, 2025–present, stockfishchess.org
"MaxBotix Inc." MaxBotix, 2025, maxbotix.com/blogs/blog/how-ultrasonic-sensors-work?srsltid=AfmBOooGXP07a7AdD1RTegGyavo3t_QaIfDNGfV8tS8VfYCJigrSlzx

ada, lady. "Li-Ion & LiPoly Batteries." Adafruit Learning System, 29 July 2012, learn.adafruit.com/li-ion-and-lipoly-batteries?view=all&gad_source=1&gad_campaignid=21079267614&gbraid=0AAAADx9JvQD5IPiIQa2pFQlS-Vs3ztXA&gclid=CjwKCAjw04HIBhB8EiwA8jGNbaz1kNg1SjsYcyBJ5XvQOCFceWa4CR4p_HTy6p7sq50ctTDigjlvQhoCSyMQAvD_BwE

SSZT164 Technical Article
Www.ti.com, 2023, www.ti.com/document-viewer/lit/html/SSZT164.

"Client Challenge." Monolithicpower.com, 2025, www.monolithicpower.com/en/learning/resources/hall-effect-sensors-a-comprehensive-guide?srsltid=AfmBOopazyS9MtK2FmSj4oXThAQgzdocOpD0SQXyt29u3KYptMvs71LO

WhatIsaCapacitiveSensor?
*Balluff.*https://www.balluff.com/en-us/blog/what-is-a-capacitive-sensor

What is a piezoelectric sensor

https://www.electronicsforu.com/technology-trends/learn-electronics/piezoelectric-sensics
Stykemain, Adam . “Inductive Sensor Explained | Different Types and Applications - RealPars.” *Www.realpars.com*, 30 June 2021, www.realpars.com/blog/inductive-sensor

IEEE Standards Association. (n.d.). *What are standards? Why are they important?* IEEE. Retrieved October 20, 2025, from https://standards.ieee.org/beyond-standards/what-are-standards-why-are-they-important/

International Organization for Standardization. (2018). *Information technology — Programming languages — C* (ISO/IEC 9899:2018). https://www.iso.org/standard/74528.html

IPC, Generic Standard on Printed Board Design, IPC-2221A, Jan. 2012. [Online]. Available: https://www.ipc.org/

Underwriters Laboratories. (2024). ANSI/UL 8750: Standard for safety — Light emitting diode (LED) equipment for use in lighting products (Ed. 2). ANSI/UL. https://www.intertek.com/siteassets/resources/sun/2024/ul-8750-sun-rev-12-7-2022-ed-5-1-2025.pdf

International Organization for Standardization, International Electrotechnical Commission & Institute of Electrical and Electronics Engineers. (2021). ISO/IEC/IEEE 29119-4:2021 – Software and systems engineering — Software testing — Part 4: Test design techniques. https://standards.iteh.ai/catalog/standards/sist/20536fd0-e74a-4b3c-8285-af54a8441204/iso-iec-ieee-29119-4-2021

IEEE Standards Association. (2015). IEEE recommended practices for modulating current in high-brightness LEDs for mitigating health risks to viewers (IEEE Standard 1789-2015). https://standards.ieee.org/ieee/1789/4479/

Institute of Electrical and Electronics Engineers. (2024). IEEE Standard for Information Technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (IEEE Std 802.11-2024). https://standards.ieee.org/ieee/802.11/10548/

Appendix B: Copyright Permissions

Include written authorization (emails, letters, etc.) for any copyrighted content used.

Appendix C: Datasheets

WS2812B LED

https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf

ESP32-S3-WROOM-1

https://documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

LCD Screen 1602

https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf

Buck-Booster Converter

https://www.ti.com/lit/ds/symlink/tps63002.pdf?ts=1761772102180&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS63002

Arduino Uno

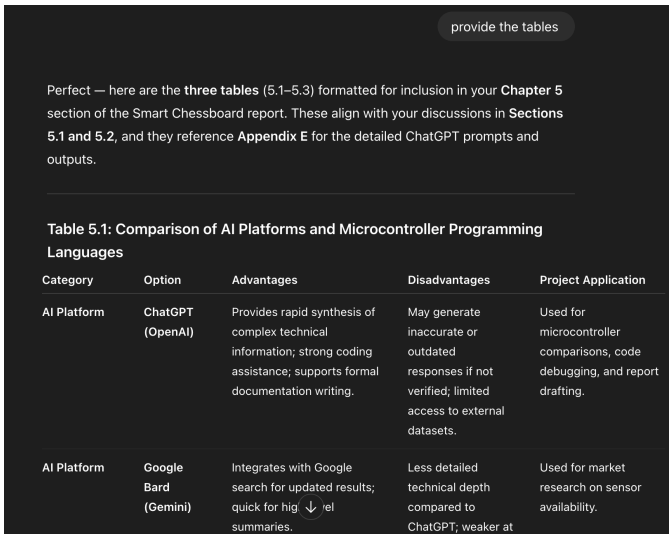
Arduino. *Arduino Uno Rev3 Technical Specifications.* 2024. https://store.arduino.cc/products/arduino-uno-rev3

Texas Instruments. *MSP430G2553 Datasheet.* 2024. https://www.ti.com/product/MSP430G2553

Appendix D: Software Code

(If necessary) Include key sections of your code.

Appendix E: ChatGPT Prompts and Outcomes

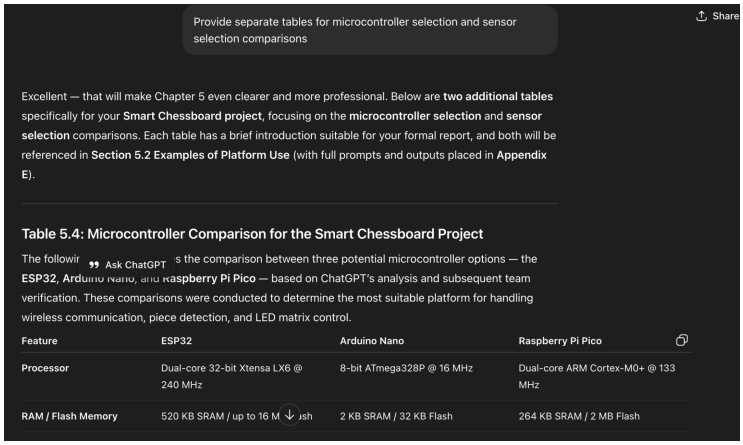


provide the tables

Perfect — here are the **three tables** (5.1–5.3) formatted for inclusion in your **Chapter 5** section of the Smart Chessboard report. These align with your discussions in **Sections 5.1 and 5.2**, and they reference **Appendix E** for the detailed ChatGPT prompts and outputs.

Table 5.1: Comparison of AI Platforms and Microcontroller Programming Languages

Category	Option	Advantages	Disadvantages	Project Application
AI Platform	ChatGPT (OpenAI)	Provides rapid synthesis of complex technical information; strong coding assistance; supports formal documentation writing.	May generate inaccurate or outdated responses if not verified; limited access to external datasets.	Used for microcontroller comparisons, code debugging, and report drafting.
AI Platform	Google Bard (Gemini)	Integrates with Google search for updated results; quick for hig ↓ el summaries.	Less detailed technical depth compared to ChatGPT; weaker at	Used for market research on sensor availability.



Share

Provide separate tables for microcontroller selection and sensor selection comparisons

Excellent — that will make Chapter 5 even clearer and more professional. Below are **two additional tables** specifically for your **Smart Chessboard project**, focusing on the **microcontroller selection** and **sensor selection** comparisons. Each table has a brief introduction suitable for your formal report, and both will be referenced in **Section 5.2 Examples of Platform Use** (with full prompts and outputs placed in **Appendix E**).

Table 5.4: Microcontroller Comparison for the Smart Chessboard Project

Ask ChatGPT

The followir s the comparison between three potential microcontroller options — the **ESP32**, **Arduino Nano**, and **Raspberry Pi Pico** — based on ChatGPT's analysis and subsequent team verification. These comparisons were conducted to determine the most suitable platform for handling wireless communication, piece detection, and LED matrix control.

Feature	ESP32	Arduino Nano	Raspberry Pi Pico
Processor	Dual-core 32-bit Xtensa LX6 @ 240 MHz	8-bit ATmega328P @ 16 MHz	Dual-core ARM Cortex-M0+ @ 133 MHz
RAM / Flash Memory	520 KB SRAM / up to 16 M ↓ ash	2 KB SRAM / 32 KB Flash	264 KB SRAM / 2 MB Flash

Chapter 5 Prompt for Tables.