Pathfinder Chessboard

Andres Armendariz, Freddy Pacheco, Ryan Ramdihal

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

Abstract --- The Pathfinder Chessboard is a smart embedded system designed to modernize physical chess without replacing it with a screen-based alternative. The Pathfinder Chessboard addresses this by integrating embedded systems into a traditional chessboard, enabling it to communicate and validate game rules, providing interactive feedback. The system uses an 8x8 array of hall effect sensors in each chessboard square to detect piece presence and movement across the chess board. Sensor signals are read through I/O expanders communicating with an ESP32 over the I2C protocol, continuously polling across all 64 board positions. Addressable RGB LEDs beneath each square provide visual feedback to players by highlighting piece selections with its legal moves and capture opportunities. A custom PCB integrates the sensing system, LED control, and power regulation components into a unified embedded platform.

I. INTRODUCTION

The Pathfinder Chessboard's objective is to enhance accessibility for beginners by reducing errors in gameplay, and creates a more engaging experience for all players, by teaching through interaction, not instruction. Even though chess is one of the most widely recognized strategy games, its complex rules often create a learning barrier for new players. Learning the movements of each piece, understanding rules, and developing effective strategies can be overwhelming, especially without having any guidance. But learning by instruction may not be engaging for all players.

To overcome this limitation, the Pathfinder Chessboard integrates embedded systems into a physical chessboard to provide real-time assistance and feedback. The system is designed to preserve the tactile and social experience of traditional chess while enhancing it with intelligent guidance. At the core of the design is an ESP32 microcontroller that processes inputs from a grid of Hall-effect sensors embedded beneath each of the 64 squares of the chessboard. These sensors detect the presence and movement of chess pieces through the use of magnets integrated within each chess piece.

The system processes board state changes and provides real-time visual feedback to assist gameplay. This feedback includes highlighting selected pieces, indicating legal moves based on the picked up piece, and identifying capture opportunities, thereby assisting players in understanding the game in real time.

The Pathfinder Chessboard is implemented on a custom printed circuit board (PCB) that integrates the sensing system, LED control, and power distribution. Our design can allow for future enhancements, including gameplay features such as adjustable AI difficulty and timed matches.

By combining embedded sensing, real-time processing, and visual LED feedback within a physical chessboard, this project demonstrates how traditional tabletop games can be enhanced using embedded systems. The Pathfinder Chessboard demonstrates how embedded systems can improve accessibility and provide real-time interactive feedback in traditional gameplay. In addition to move validation, the system provides immediate visual and informational feedback through integrated LEDs and an LCD interface. This allows players to understand the results of their actions in real time. The combination of embedded sensing and interactive feedback creates a more intuitive learning environment compared to traditional instruction-based methods.

II. SPECIFICATIONS

This section outlines the specifications laid out for our project to ensure reliability,

performance, and overall system functionality. These specifications will be elaborated throughout the subsequent sections where individual subsystems and design decisions are analyzed in greater detail.

Table 1 : Project Specifications		
#	Parameters	Target Value
1	Chess Board Dimensions	Main chess board:<20x20x2 in Side box: <5x5x2 in (Length x Width x Height)
2	Battery Life	≥ 2 hours
3	Temperature	< 100 degrees Fahrenheit
4	Modes	3
5	Power Consumption	<22 Watts
6	LED Response Time	≤ 1 second
7	Magnetic Flux detection	~0 V for values >3.32mT ~5V for values < 1.17
8	Sensor Detection Speed	< 500 µs

III. MAIN HARDWARE COMPONENTS

This section outlines the components used for the operation of the Pathfinder Chessboard. Each component plays a critical role for enabling the sensing, processing, communication, and user interaction within our project.

A. Microcontroller

We decided to use the ESP32 microcontroller for our Pathfinder Chessboard due to its processing capabilities and support for multiple communication protocols.The system requires real-time processing and control of multiple peripherals, which the ESP32 can handle efficiently with its dual core processor that can operate at up to 240 MHz.

The ESP32 operates at 3.3 V logic levels and can support communication protocols such as I²C and UART. The I²C protocol is used to communicate with four MCP23017 I/O expanders, enabling the system to poll across all 64 board positions whilst also overcoming the limited 34 GPIO pins of the ESP32. The ESP32 also supports hardware timers and interrupt capabilities, which can be used to manage precise timing operations such as LED updates and input handling. Its dual-core architecture allows separation of tasks such as sensor polling and game logic processing, improving overall system efficiency. A CP2102 USB-to-UART bridge is used to communicate with an IDE for programming, debugging, and serial monitoring.

The ESP32 can be programmed using development environments: Arduino IDE and ESP-IDF. Additionally, the ESP32 has built-in Wi-Fi and Bluetooth capabilities that provide opportunities for any potential future features such as wireless connectivity and remote gameplay.

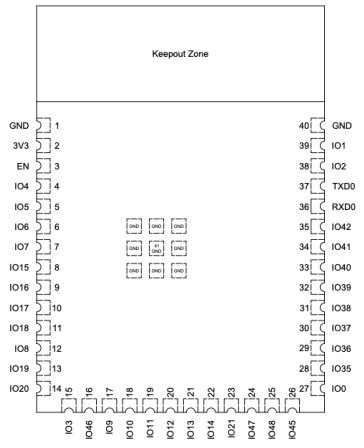

Figure 1. Pin layout of the ESP32-S3 WROOM1 module.

The ESP32 serves as the central component of the system, networking both the sensor and LED subsystems while managing game logic using the programmed chess engine: mcu-max. Its performance and flexibility make it ideal for the requirements of the Pathfinder Chessboard features.

B. Sensors

The sensor subsystem uses 64 DRV5056 Hall-effect sensors on each board square to detect any presence or movement of a chess piece. These sensors detect the chess pieces by using the magnetic fields produced by magnets within each piece.

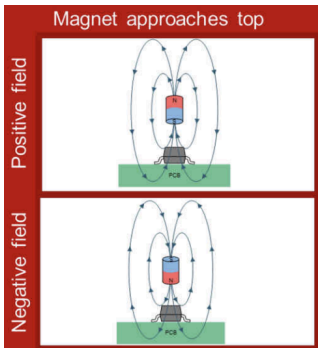

Figure 2. Illustration of the magnetic field produced by a magnet and its detection by the DRV5056 Hall-effect sensor on a PCB.

The DRV5056 is a linear analog Hall-effect sensor that operates at 3.3V and produces an output voltage proportional to the magnetic flux density generated by the magnet embedded in each chess piece described by:

$$V_{OUT} = V_Q + B \times \left(\text{Sensitivity}_{(25°C)} \times (1 + S_{TC} \times (T_A - 25°C))\right)$$

Equation. Analog output voltage, where V_Q is the quiescent output voltage at zero magnetic field, B is the applied magnetic flux density, $S_{25°C}$ is the sensor sensitivity at 25°C, STC is the sensitivity temperature coefficient, and T_A is the ambient temperature.

In the Pathfinder chessboard, this analog output allows the system to detect whether a chess piece is present on a square by comparing the measured voltage to a preset voltage value selected in the ESP32. As the distance between the magnet and sensor changes, the magnetic flux density and corresponding output voltage also change, for reliable distinction between placement and removal.

The sensors are arranged in an 8x8 grid corresponding to the chessboard coordinates and are connected to the ESP32 through 4 MCP23017 I/O expanders. This allows for efficient monitoring of all sensor inputs while reducing direct GPIO connections to the ESP32. By using Hall-effect sensors, it ensures consistent performance regardless of lighting conditions and avoids false triggers that could occur with optical or capacitive sensing methods. The DRV5056 Hall-effect sensors provide a reliable and contactless solution for piece detection compared to any other method.

C. LEDs

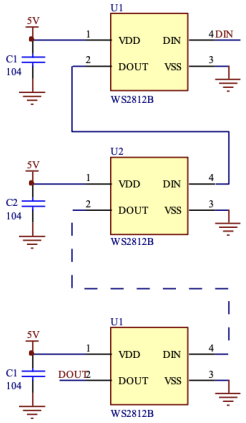

Figure 3. Illustration of a typical application circuit for the WS2812B LED chain.

For our LED subsystem, we are using the WS2812B addressable RGB LEDs to provide the visual feedback on each square for the Pathfinder Chessboard. Each square has one LED partnered with a sensor, allowing the system to display any changes in the game state. The response of the LEDs need to be considered seamless with the movement of a chess piece. Thus, we set our target specification of at most 1 second for the LED response time.

The WS2812B LEDs operate at 5V and can be controlled individually through only one data line. This allows all 64 LEDs to be connected in a chain and controlled using one GPIO pin from the ESP32, which simplifies wiring and design for our PCB layout. Since the ESP32 operates at 3.3V logic and the LEDs require a 5V data signal, a logic level shifter is used for reliable communication between the ESP32 and the LED chain.

These LEDs are easy to control in software and provide a wide range of colors, making them befitting for real-time visual feedback. The individually addressable nature of the LEDs allows each square to be controlled independently, enabling dynamic patterns and responsive updates across the board.

D. I2C expanders

Using 4 MCP23017 I2C I/O expanders to increase the number of input pins for our chessboard. The system needs to read 64 sensors, which is more than the ESP32 can support with its available GPIO pins.

Each expander provides 16 GPIO pins and can communicate with the ESP32 using the I2C protocol. By using four expanders, we are able to connect all 64 sensors on the board while only using two lines (SDA and SCL) from the ESP32. Each expander has a unique address allowing multiple devices to share the same communication bus, reducing wiring complexity and enabling easy system expansion.

The MCP23017 helps simplify wiring by reducing the number of direct connections to the ESP32 and provides a simple and efficient way to handle all sensor inputs.

E. Logic Level Shifter

The SN74AHCT125 logic-level shifter allows for effective data transmissions between the ESP32 and our WS2812B LEDs which operate on different voltage levels. The ESP32 outputs 3.3V signals, while the WS2812B LEDs require a 5V data signal.

The SN74AHCT125 is used to translate the signal from the ESP32 into a 5V signal that the LEDs can reliably read. This ensures stable LED operation.

F. Batteries

The Pathfinder Chessboard requires approximately 17.64 W of power to support all onboard components. To meet this demand the system must provide 3.3 V and 5 V supply rails for the various subsystems.

The power supply is powered by a pack of 4 Nickel Metal Hydride (NiMH) batteries which supplies energy to both a 3.3 V and a 5 V voltage regulation stage. We specifically choose the TENERGY D 10,000 mAh NiMH batteries to ensure a lifespan that is over two hours with our power calculations. This as well as rechargeability allows for long term use with the Pathfinder Chessboard.

The 4 NiMH batteries provide an input voltage between 4-5.4 V, which closely aligns with the 5 V and 3.3 V supply requirements of the system components, therefore reducing the burden on the voltage regulation stages.

G. 3.3V Regulator

Many of the pathfinder chessboards require a stable 3.3 V supply for proper operation. In order to accommodate for the depleting voltage of the batteries a 3.3V regulator must be used.

To address this the TPS63000DRCR was selected for its buck-boost capabilities, allowing it to both step up and step down the input voltage as needed. This ensures a steady 3.3 V output even as the battery voltage fluctuates. Additionally the device is well suited for use with both three alkaline or four NiMH batteries making it a reliable choice for this application.

H. 5V Regulator

The LEDs and other system components require a 5 V supply for proper operation. To meet this requirement the LTC3113 is utilized. This regulator provides both step-up and step-down functionality ensuring a consistent 5 V output regardless of variations in the battery voltage.

The use of a buck-boost regulator is particularly important in this design as the input voltage from the NiMH can fall both above and below 5 V during operation. By maintaining a constant output voltage the regulator ensures that the 5 v dependent components are all functionally operating.

IV. SYSTEM CONCEPT

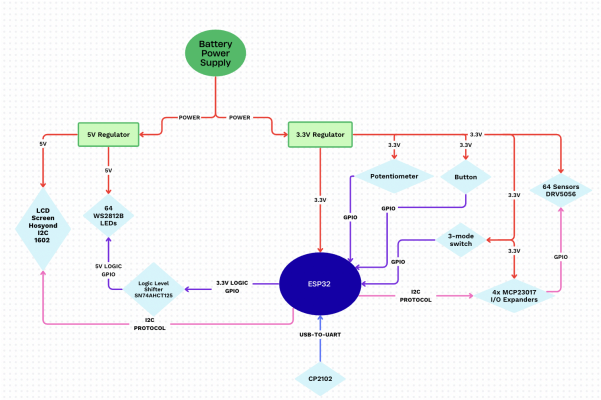

Figure 3. Hardware block diagram of the Pathfinder Chessboard showing connections between ESP32 and all its components.

The Pathfinder Chessboard blends sensing, processing, and visual feedback to create a more interactive version of traditional chess. The system continuously monitors the state of the board and reacts instantly to player actions using the array of LEDs. This is achieved through a polling-based architecture, where the ESP32 continuously reads all sensor inputs to detect state changes with minimal delay.

When a chess piece is placed on or removed from a square, the Hall-effect sensor beneath that square detects the change. These signals travel through the MCP23017 expanders to the ESP32, which updates the board state and decides how the LEDs should respond based on the chess engine's logic.

After processing the move, the ESP32 sends data through the GPIO pin to the WS2812B LED chain array. The signal passes through a logic level shifter to convert the 3.3V data signal to 5V, ensuring reliable communication with the LEDs. The LEDs then provide visual cues—highlighting legal moves, showing capture options, or signaling invalid actions with different colors and animations.

The board also includes a simple user interface: an LCD display, a 3-way switch, a potentiometer, button and an ON/OFF switch. The switch lets players choose game modes, while the potentiometer and button adjust and confirm timer settings. The LCD shows useful information such as moves, selected mode, and active timers, keeping the player informed at all times. The user interface operates alongside the sensing and processing subsystems, allowing configuration changes without interrupting gameplay.

The entire system runs in a continuous polling loop, checking for changes on the board and updating the LEDs immediately so gameplay feels smooth and responsive.

Overall, the system demonstrates a tightly integrated embedded architecture where sensing, processing, and output subsystems operate in real time to enhance a physical chess gameplay.

V. PHYSICAL DESIGN

This section provides a broad description of the printed circuit board (PCB) as well as the housing used to incorporate the Pathfinder Chessboard. The PCB measures approximately 16.65 x 15.27 inches closely mimicking the dimensions of a standard chessboard.

The PCB features an 8x8 array corresponding to the 64 squares of a traditional chessboard. Each square measures 1x1 inch and is

equipped with an LED and an associated sensor, allowing for piece detection and visual feedback. Positioned between each LED column is the associated GPIO expander for that column. To the right of the array additional ICs and components including the voltage regulators, MCU, CP2102, and header pins are arranged to minimize distance between onboard components and external peripherals.

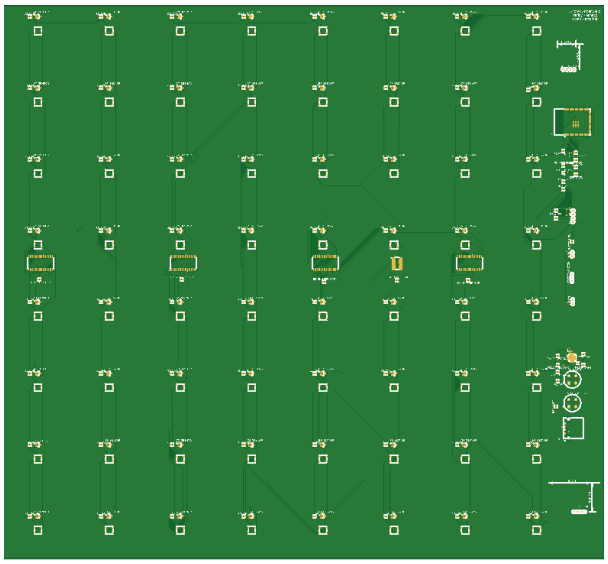

Figure 4: PCB Layout Top View

The housing will feature two distinct sections: the main board section and the side compartment section. The main board section houses the PCB along with its components.

The side compartment section contains the external components that the user interacts with, including the three mode switch, ON/OFF button, LCD screen, and potentiometer. Internally this section also houses the batteries and the wiring that leads to the through hole components of the PCB.

The dimensions of the housing model allow us to achieve our specifications of a board which is 20 inches by 20 inches, along with the side compartment which is 7.5 inches by 5 inches.

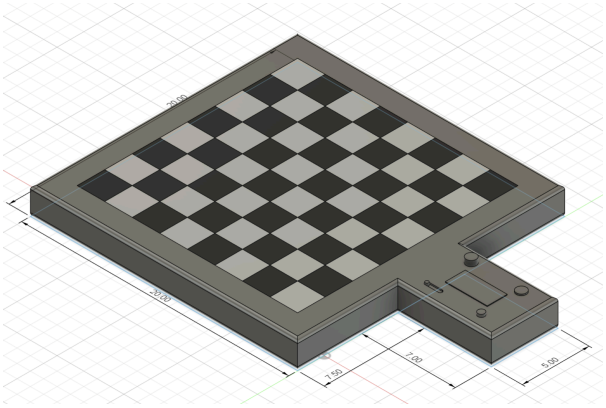

Figure 5: Housing Model Mockup made in

VI. SOFTWARE

This section highlights the software used in the operation of the Pathfinder Chessboard, along with its interaction with the hardware. Our project requires fast and precise communication within the system's software and hardware.

A. System Architecture

The software of the Pathfinder Chessboard interacts with the hardware to allow the user to easily play a game of chess, both against a human and computer opponent. At the setup stage of the board, the user selects which mode they wish to play, and then, if necessary, if they wish to play against a human or computer. The time limit is then chosen, with some modes having the option to choose an infinite time, meaning the game lasts indefinitely.

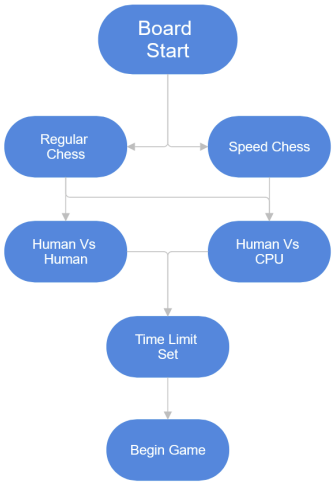

Figure 6: Flowchart of the setup stage of the Pathfinder Chessboard, up until the game begins.

When the game begins, the player first lifts a piece from the board, activating the Hall-effect sensor under the piece, which then sends a signal to the MCU, indicating which piece was lifted, and then the corresponding LEDs light up indicating possible moves for that piece. The player will then drop their piece on a square, which activates the sensor at that square, and the system responds by checking if that is a valid square to move to. If not, then a warning is highlighted on the board, but if the move is valid no warning is sound.

The system then checks if any end game conditions have been met, such as a stalemate or check, and if not, the turn is switched, allowing the other player to lift a piece. If an end game condition is met, however, then the system stops the game, highlights the board, and congratulates the player who has won, ending the game.

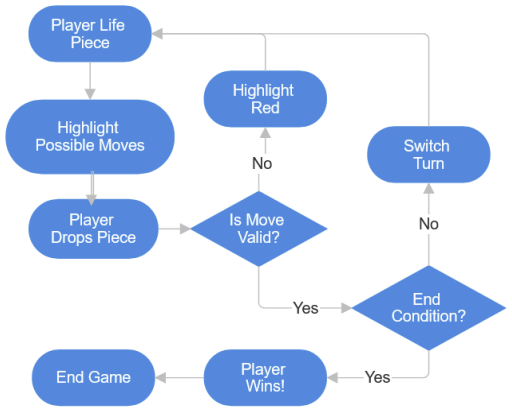

Figure 7: Flowchart indicating the gameplay loop of a standard human versus human chess game.

B. MCU-Max Engine

The Pathfinder Chessboard uses an external chess engine to implement and handle all game logic instead of having a custom made one. This decision was made mostly due to the time constraint, as we saw the time it would take to make and debug the system, specifically a computer player, would far exceed the time limit and scope of our project.

We went with the MCU-Max chess engine, which is a compact chess engine specifically modified to be used on smaller microcontrollers such as our ESP32. It also provides an efficient and powerful computer player.

The MCU-Max engine holds all the needed game logic, this includes the ability to start games, hold a board in memory, and determine possible moves based on which piece is chosen.

C. MCU-Max AI Logic

The MCU-Max engine comes with a prepacked AI logic system which allows the user to play against a computer opponent.

The logic is built around an alpha-beta negative max searcher, where the computer holds two point values, the alpha, the best score found so far, and the beta, the lowest tolerable score. The

computer then calculates every possible move, pruning the branches which would require a score below the beta, which eliminates the vast majority of move combinations, allowing for a more efficient search. The engine does not just look one move ahead, but will look multiple moves into the future to be sure to find the best move.

The negative part of the engine comes from a clever use of the fact that chess is a zero-sum game, meaning one player's advantage is equal to the opponent's disadvantage. When evaluating multiple moves ahead, the computer must simulate its opponent's move, and instead of creating a separate algorithm it simply runs the algorithm with a negated alpha and beta, then negates the final score, giving the best move to the opponent, allowing itself to make the best move on the next simulated turn.

Another component to this engine is the implementation of a quiescence search, which, in short, makes sure that the engine does not end a turn in a vulnerable position, allowing the engine to get out of volatile situations. This functions by making sure the move the system makes finishes on a board where there is little to no possibility of the next human move capturing a computer piece.

To determine the points allocated to a move, the computer uses a variety of factors. The biggest factors are capturing a piece or putting yourself in a position to be attacked, which increase and decrease the value of the move respectively. Points are also awarded to a move for board control, such as having pieces in the center, and a variety of other factors.

Another appealing factor of this engine is the ability to customize the strength of the logic. The user is able to use two different variables to increase or decrease the strength of the opponent. The first is the search depth. This determines how many half moves, meaning only one player's move, the AI is allowed to search through. For example, if the number is 3, the AI will search for its best possible move, the player's response, and then their own response to that, and select the best move there. The higher the number gets leads to better planning and development, but at the cost of time efficiency and resource use. The second variable is the node amount, which represents the amount of branching paths the engine is allowed to make. In conjunction with the depth, we have a powerful system which can make sure to find many different moves.

REFERENCES

[1] Texas Instruments, "DRV5056 Linear Hall-Effect Sensor Datasheet," 2023. [Online].https://www.ti.com/lit/ds/symlink/drv5056.pdf

[2] Espressif Systems, "ESP32-S3-WROOM-1 & ESP32-S3-WROOM-1U Datasheet," 2023. [Online]. https://documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[3] Microchip Technology Inc., "MCP23017 16-Bit I/O Expander Datasheet," 2016. [Online]. https://ww1.microchip.com/downloads/en/DeviceDoc/20001952C.pdf

[4] Worldsemi, "WS2812B Intelligent Control LED Datasheet," 2018. [Online]. https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf

[5] Texas Instruments, "SN74AHCT125 Quad Bus Buffer Gates Datasheet," 2015. [Online]. https://www.ti.com/lit/ds/symlink/sn74ahct125.pdf

[6] Gissio, "mcu-max: A Low-Resource Chess Engine for Microcontrollers," GitHub repository. [Online]. https://github.com/Gissio/mcu-max

[7] Texas Instruments, "Using Hall-Effect Sensors for Contactless Linear Movement Detection," Application Report SLYT826, 2020. [Online]. http://ti.com/lit/an/slyt826/slyt826.pdf