



An-Najah National University

Faculty of Engineering & Information Technology

Presented in partial fulfillment of the requirements for Bachelor  
degree in Computer Engineering

Graduation Project 2

ChessWhisperer

Students:

Masa Sayegh

Masa Zablah

Supervisor:

Dr. Aladdin Masri

January 12, 2024

## Acknowledgment:

We want to sincerely thank everyone who helped to see this project through to completion. As our supervisor, we would especially want to thank Dr. Alaadin Masri for all of his help and support. We would like to express our gratitude to our families and friends for their steadfast support during this academic journey.

## Disclaimer Statement:

This report was written by Masa Zablah and Masa Sayegh, students at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

# Table of Contents

Acknowledgment:.....	2
Disclaimer Statement: .....	3
Table of Contents .....	4
Table of Figures .....	6
Abstract:.....	8
1 Chapter 1: Introduction .....	9
1.1 Statement of the problem.....	9
1.2 Objectives of the work .....	9
1.3 Significance of the work .....	9
1.4 Organization of the report .....	10
2 Chapter 2: Theoretical Background and Previous Work .....	11
2.1 Intelligent Chessboard Using IOT: .....	11
2.2 Wizard Chess: An Autonomous Chess Playing Robot:.....	11
2.3 Instructive Smart Chess Board: .....	11
3 Chapter 3: Methodology.....	13
3.1 Hardware Modules.....	13
3.1.1 Microcontrollers:.....	13
1.1.1 Motors and Drivers: .....	14
3.1.2 Sensors:( Reed Switches): .....	16
3.1.3 Multiplexers (16-Channel Analog Multiplexer):.....	17
3.1.4 Input/Output Devices: .....	18
3.1.5 Addressable RGB LED Strip APA102: .....	20
1.1.2 Other devices: .....	20
3.1.6 Mechanical Parts:.....	23
3.2 Standards and codes: .....	26
3.2.1 Software Implementation:.....	26
3.2.2 AI: .....	26
3.2.3 Flow chart: .....	30
3.3 Constraints: .....	33

4 Chapter 4: Dissociation and Results.....	34
5 Chapter 5: Conclusion and Recommendations.....	35
5.1 Future Work: .....	35
6 References.....	36

## Table of Figures

Figure 1 Arduino Mega	13
Figure 2 ESP32	14
Figure 3 Stepper Motor Nema17	14
Figure 4 Stepper motor driver YS-DIV268-5A	15
Figure 5 Reed Switch	16
Figure 6 64 Reed Switches wiring	17
Figure 7 16-Channel Analog Multiplexer	18
Figure 8 LCD with I2C with Arduino	19
Figure 9 2 Arcade Buttons With internal resistors with Arduino	19
Figure 10 Limit Switch with Arduino	20
Figure 11 Addressable led with Arduino	20
Figure 12 Electromagnet with power transistor, diode, and with Arduino	21
Figure 13 Power Supply PC	22
Figure 14 BreadBoard	22
Figure 15 DIY Prototype PCB Board	23
Figure 16 Linear PCB Board	23
Figure 17 Decision Tree	27

## Table of Tables.

Table 1 Mechanical Parts.

---

25

## Abstract:

The ChessWhisperer project, an automated chessboard incorporating artificial intelligence (AI) to enhance both the educational and gaming aspects of chess. ChessWhisperer is designed to transform the traditional chess experience by seamlessly blending physical gameplay with intelligent algorithms. The project's core features include an AI-powered gaming mode, enabling users to face a dynamic and challenging opponent, and an interactive learning mode that guides users through the principles of the game. The significance of ChessWhisperer lies in its potential to meet the rising demand for tech-enhanced gaming experiences, appealing to both beginners and seasoned players. The hardware components utilized in the project, such as microcontrollers, motors, sensors, and input/output devices, are detailed, showcasing the intricate integration required for its functionality. The AI algorithms, based on MinMax and Alpha-Beta, contribute to ChessWhisperer's strategic gameplay. The report highlights the successful implementation of the project, overcoming constraints related to hardware and components. ChessWhisperer stands as a comprehensive and innovative addition to the landscape of automated chessboards, offering an engaging and educational experience for chess enthusiasts of all levels.

# Chapter 1: Introduction

## 1.1 Statement of the problem

Chess is a challenging game that people often learn by themselves. Regular chess sets don't let fans play much because they're not interactive. That's where our project, ChessWhisperer, comes in. We're developing a chessboard based on artificial intelligence to make chess more fun and educational. With ChessWhisperer, you can play against a computer opponent and use it as a tool to learn. We want ChessWhisperer to change how people learn and play chess, making it more enjoyable for beginners and experienced players. It's like combining the feel of a real chessboard with the brainpower of a computer.

## 1.2 Objectives of the work

The objective of the ChessWhisperer project is to create, develop, and deploy an automated chessboard that improves learning and gameplay for players of different abilities. The main objectives are as follows:

**AI-Powered gaming:** We want the chessboard to be like a smart opponent by using computer algorithms. This way, players can get better by facing a challenging and adaptable opponent.

**Interactive Learning:** Enable users to interact with the chessboard in order to help beginners understand the game's principles. It's about making learning chess more fun and hands-on for everyone.

## 1.3 Significance of the work

ChessWhisperer holds great significance in the world of chess education and gameplay, addressing a growing demand for effective learning tools and aligning with market trends. As interest in chess continues to rise globally, ChessWhisperer meets this demand by offering an interactive platform suitable for both beginners and experienced players. ChessWhisperer merges classical chess with modern technology, meeting the rising demand for tech-enhanced gaming experiences. It also appeals to the growing popularity of AI-driven skill development. Moreover, ChessWhisperer's commitment to accessibility and inclusivity breaks down barriers, making chess more welcoming to diverse demographics. As a leading force in market projections, ChessWhisperer adds significant value and innovation to the evolving chess-related product and educational tool landscape.

## **1.4 Organization of the report**

The report is organized into several key sections, commencing with an introduction that outlines the ChessWhisperer project's objectives and significance. A literature review follows, summarizing existing research in automated chessboards, while the methodology section details the hardware and software components employed. Results and analysis present the project outcomes and address encountered challenges, leading to a discussion that contextualizes ChessWhisperer's impact and suggests improvements. The conclusion summarizes key findings, and recommendations propose avenues for future development. Acknowledgments express gratitude, followed by a list of references and any supplementary appendices. This structured approach ensures a coherent presentation of the ChessWhisperer project and its implications.

## Chapter 2: Theoretical Background and Previous Work

### **2.1 Intelligent Chessboard Using IOT:**

Users will be able to play chess against an AI opponent from anywhere with the Automatic Chessboard, a ground-breaking project that combines robotics, computer vision, and artificial intelligence to create an Intelligent Chessboard. The system includes a robotic arm with four degrees of freedom that is managed by a central unit. The project's modules address serialization, machine learning, image processing, and the robotic arm's motion engine. But difficulties occur, especially in image processing because a constant light source free of shadows from objects around it is required. Move detection encountered problems with human and chess piece shadows. Black shadows under white pieces were removed by changing the lighting, and image filtering was used to reduce specular reflections. As an improvement over image processing, a sensor was introduced to increase efficacy and efficiency. Furthermore, moving the system to various locations presented logistical challenges due to the usage of the robotic arm.

### **2.2 Wizard Chess: An Autonomous Chess Playing Robot:**

The paper presents Wizard Chess, an autonomous robotic chess player intended for home, educational, and recreational use. In contrast to current chess-playing robots, which depend on elaborate and expensive camera and articulated robotic arm setups, Wizard Chess uses an inexpensive and architecturally straightforward method. Sixty-four magnetic reed switches are used by the system to determine the current game state, and an X-Y Cartesian table is used to move the chess pieces. The prototype's deployment shows that the system is workable in practical settings. The software is modular for easy updates and modifications, and the design places an emphasis on locally accessible, simple, and low-cost materials. It is additionally planned to include a voice recognition system so that people with disabilities can use the system. All things considered, Wizard Chess is a novel and user-friendly approach to robotics that plays chess autonomously.

### **2.3 Instructive Smart Chess Board:**

The original purpose of the autonomous robotic chess board was to help students learn the game and comprehend the fair gameplay mechanics. The prototype made use of multiplexer chips, electrical circuits, reed switches for piece position detection, and LED lights for confirmation of legal moves. Testing, however, turned up problems with the circuits and reed switches, which prompted studies into PCB boards and hall sensors to

increase accuracy. A mechanical arm was left out of the proposed gantry robotic tree because of practical limitations. The project's primary objectives of teaching students' legal moves and generating opponent moves were refocused in order to create an Instructive Smart Chess Board. The focus shifted from an intricate robotic system to an educational prototype with the goal of making improvements in the future. It was decided that adding a robotic arm would not be necessary to accomplish the project's main goal of teaching students' chess.

Our suggested chess whisperer is different from these methods in that it is more feasible and simpler. This little, affordable chess-playing chessboard performs consistently and with dependability. For example, only three stepper motors with reed sensors are controlled in the ChessWhisperer, however the two motor's movement is independent of the other. Its mechanical simplicity makes it useful for individual use in recreational and educational contexts, and has the feature of teaching players how to play with the optimal moves.

## Chapter 3: Methodology

### 3.1 Hardware Modules

In order to build this project, we used many hardware components:

#### 3.1.1 Microcontrollers:

##### 3.1.1.1 Arduino Mega 2560

We used Arduino Mega 2560, it's a microcontroller board based on the ATmega2560. It contains 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, 54 digital input/output pins (of which 15 may be utilized as PWM outputs), a USB connector, a power jack, an ICSP header, and a reset button. It is the best microcontroller for our project because we needed a lot of pins unlike the Arduino uno.



Figure 1 Arduino Mega

##### 3.1.1.2 ESP32:

ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC lowpower 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios. We used it in order to establish a serial connection between the ESP32 and the Arduino Mega. In our project, we connected the ESP32 to the Arduino Mega. This pairing allowed us to link our system to a mobile app created with Blynk. The ESP32, with its Wi-Fi and Bluetooth features, acted as a communication bridge. Through the Blynk app on our phones, we could easily control and monitor different aspects of our project remotely. The

ESP32's flexibility made it possible for the Arduino Mega to communicate in real-time with the Blynk app, making our project user-friendly and responsive.



Figure 2 ESP32

### 1.1.1 Motors and Drivers:

#### 3.1.1.3 Stepper Motor Nema17:

Stepper motors enable accurate positioning with ease. They are used in various types of equipment for accurate rotation angle and speed control using pulse signals. They generate high torque with a compact body, and are ideal for quick acceleration and response. Stepper motors also hold their position at stop, due to their mechanical design.

In order to control the XY movement, we used 3 stepper motors nema17, one stepper to control the x movement and 2 steppers to control the y movement.

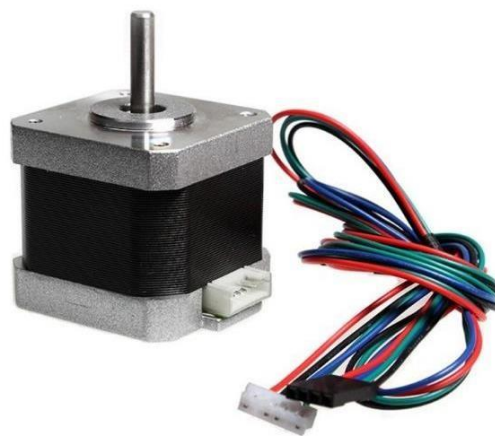


Figure 3 Stepper Motor Nema17

#### 3.1.1.4 Stepper motor driver YS-DIV268-5A:

Stepper motor drivers are electronic devices that control the movement of stepper motors by sending the appropriate electrical signals to the motor windings. These drivers are crucial for ensuring precise control of the stepper motor's position, speed, and direction. Stepper motor drivers convert pulse signals from the controller into motor motion to achieve precise positioning. We used 3 drivers for each stepper motor.

Control signal interface:

- PUL +: step pulse signal is input side or the positive pulse signal input positive terminal
- PUL -: the negative input of the negative input pulse signal or a positive pulse signal
- DIR +: stepping direction signal input to the positive terminal or negative pulse signal input to the positive terminal
- DIR -: stepping direction signal input of the negative side or reverse step pulse signal input negative terminal
- EN +: offline can reset signal input
- EN -: offline can reset signal input negative terminal

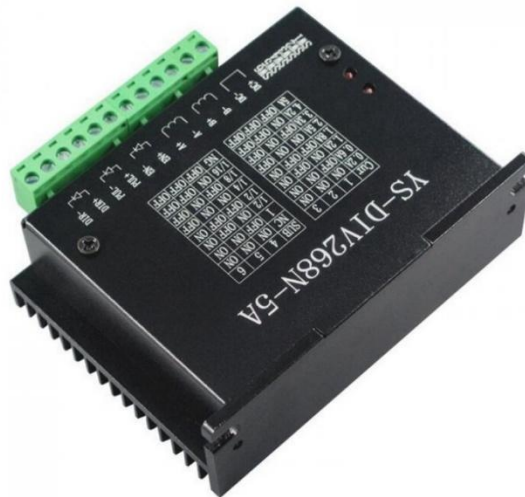



Figure 4 Stepper motor driver YS-DIV268-5A

### 3.1.2 Sensors:( Reed Switches):

Chess piece localization: It is crucial to localize the chess pieces. The goal is to communicate to the program the chess piece displacement made by the human player. Beneath every chess piece square are magnetic sensors. Thus, the sensor will become active when the chess piece's magnet is above it. The displacement of the chess piece may be ascertained by comparing the active sensors before and after the move. The magnetic sensor, also known as the Reed switch, is made up of two flexible metal contacts housed inside a hermetic glass envelope. In the absence of a magnetic field, the contact is typically open; in the presence of a magnetic field, it closes.

0  There is chess piece above this square.

1  There is no presence of a chess piece above this square.



*Figure 5 Reed Switch*

We connected the 64-reed switch as organized as possible to keep them safe and have a neat design. We connected one side of reed switches to the ground, and the other as an input to the multiplexer.



Figure 6 64 Reed Switches wiring

### 3.1.3 Multiplexers (16-Channel Analog Multiplexer):

Since we will be using 64 sensors, and it needs too many pins to the Arduino, we used multiplexers. Four multiplexers were utilized to control the 64 sensors since one multiplexer can handle 16 sensors. Channel pins (C0 to C15) are used to link the multiplexers to the reed switches. There is a unique 4-bit binary address for every channel pin.

The multiplexer pins:

- Select Pins (S0-S3): Utilized to choose the preferred channel. Which analog input is connected to the common output depends on the binary combination given to these pins.
- Output Pin (SIG): The chosen analog signal is shown on the common output pin. So, it is

used to retrieve the reed sensor signal once the channel has been chosen.

- Enable Pin (EN): This pin is used to turn on or off the multiplexer. The multiplexer responds to control inputs when it is active and ignores them when it is inactive.

SO, we decreased it to only 12 pins needed to connect to the Arduino.

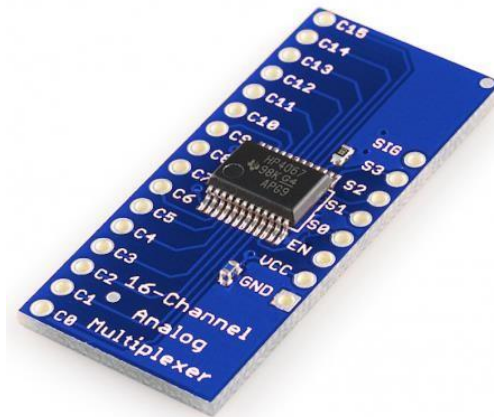


Figure 7 16-Channel Analog Multiplexer

### 3.1.4 Input/Output Devices:

#### 3.1.4.1 5.1 LCD and I2C:

An LCD (Liquid Crystal Display) with an I2C interface is a kind of display module that combines the ease of use and simplicity of the I2C communication protocol with the capabilities of a regular LCD.

I2C is a two-wire serial communication protocol that makes it possible to link several devices to a single bus, which reduces the number of pins needed on microcontrollers and other controlling devices and simplifies wiring.

In our project we used LCD with I2C to display:

- The two modes of our game: Learn and Play.
- A message in case of a no valid move.
- Win Or lose.

- Case Kill.
- The timing countdown of the game.
- Human turn and AI turn.

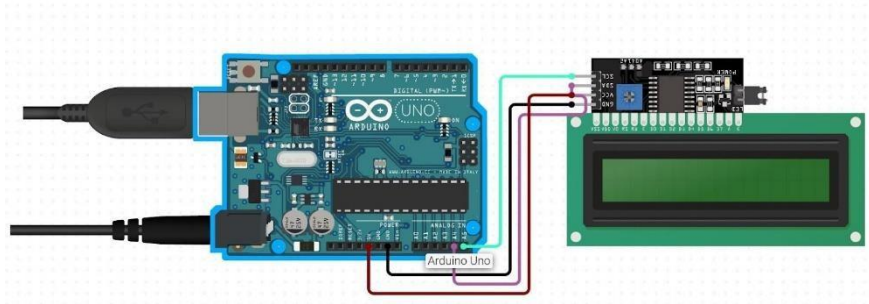


Figure 8 LCD with I2C with Arduino

### 3.1.4.2 Buttons (Push Button):

Push Buttons or switches connect two points in a circuit when you press them. In our project we used 2 buttons, one for case play, and the other for case learn.

Also, we used them for the timing, which is when it is the human turn, he should press the button before and after his turn.

The arcade buttons are connected to the arduino with internal pull-up resistors.

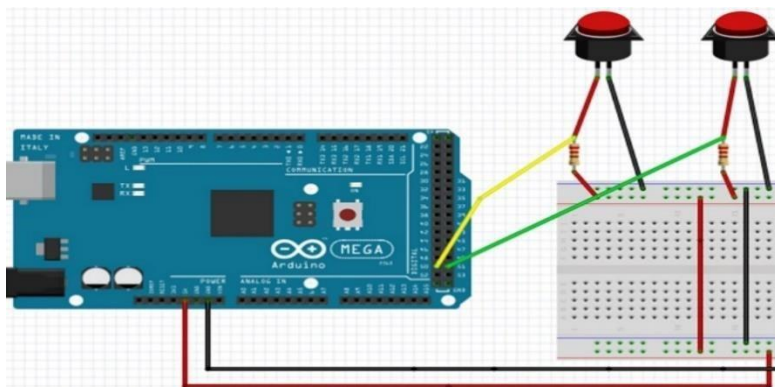


Figure 9 2 Arcade Buttons With internal resistors with Arduino

### 3.1.4.3 Limit Switches:

A limit switch is an electro-mechanical device used to send an electrical signal based on a physical interaction. Limit Switches are used to detect the presence of objects.

We used 2 limit switches, for x axis, and y axis. We needed them in order to make a reference to the movement of the motors.

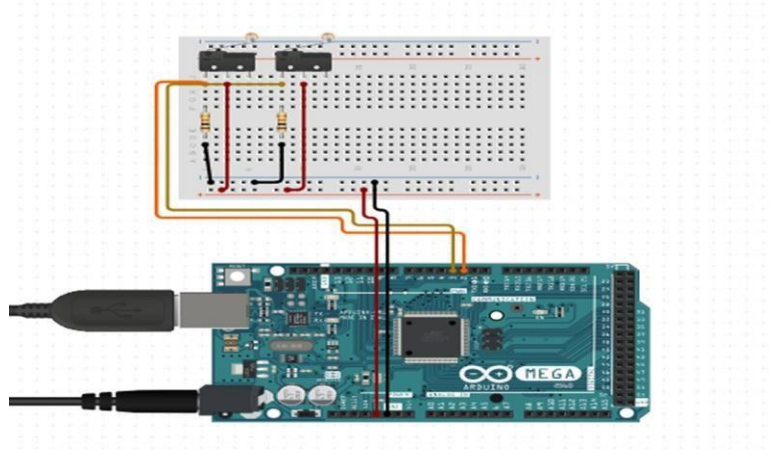


Figure 10 Limit Switch with Arduino

### 3.1.5 Addressable RGB LED Strip APA102:

Addressable RGB LED strips are a kind of LED lighting solution that provide dynamic and adaptable lighting effects by enabling independent control over each LED on the strip. Because every LED on the strip has a unique address, color and brightness can be precisely controlled.

In our project we used the addressable led strip in order to achieve the learning feature, we used 2 leds for each square.

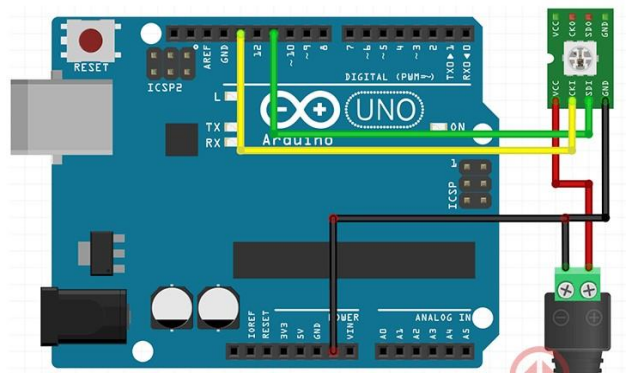


Figure 11 Addressable led with Arduino

### 1.1.2 Other devices:

#### 3.1.5.1 Electromagnet:

Electromagnets are made of coils of wire with electricity passing through them. Moving charges create magnetic fields, so when the coils of wire in an electromagnet have an electric current passing through them, the coils behave like a magnet. When the electricity stops flowing, the coils don't act like a magnet anymore. Electromagnets are used in a lot of electronic devices when magnetic forces are only needed for short periods of time.

In our project we used the electromagnet with a power transistor, when the power transistor is enabled, it functions as a switch that permits current to pass through the electromagnet and turns it off when it is deactivated.

The electromagnet is responsible for holding the chess piece when it needs to move and leave it when it is settled.

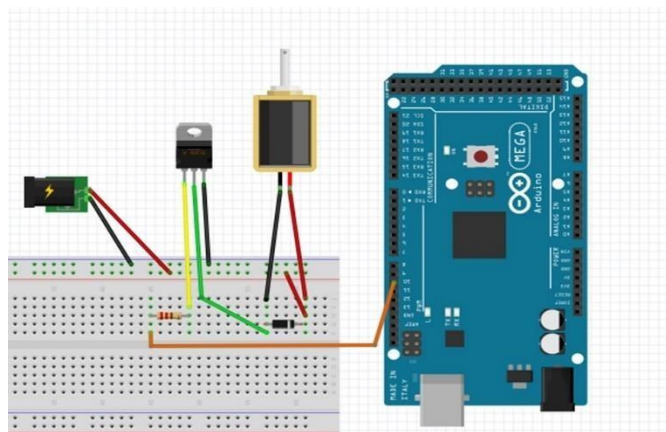


Figure 12 Electromagnet with power transistor, diode, and with Arduino

### 3.1.5.2 Power Supply:

In our project we needed different voltages, so we used a computer power supply that provides 5v, and 12v, which we needed both.



Figure 13 Power Supply PC

### 3.1.5.3 BreadBoard:

In electronics, a breadboard is an essential tool for testing and developing circuits without soldering. It offers an easy method for joining electronic parts to form transient circuits. We used the breadboard for connecting the reed sensors to the muxes.

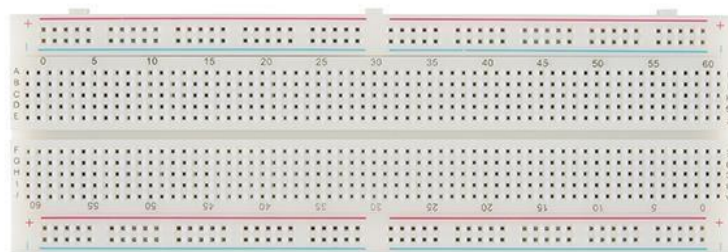


Figure 14 BreadBoard

### 3.1.5.4 DIY Prototype PCB Board

In our project we used it to connect the circuits needed. The limit switches, electromagnet, arcade buttons, lcd and i2c, etc.

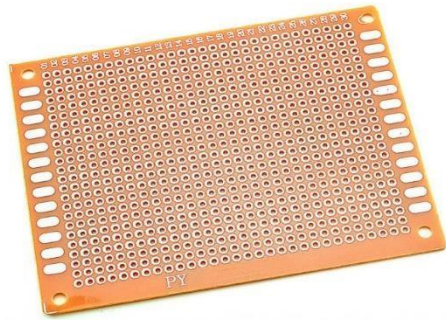


Figure 15 DIY Prototype PCB Board

### 3.1.5.5 Linear PCB board:

We used it to connect the addressable leds, which is easier using this linear board.

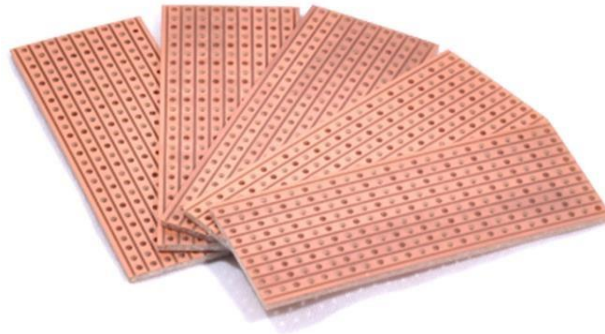


Figure 16 Linear PCB Board

### 3.1.6 Mechanical Parts:

Item name	Item image	Quantity
-----------	------------	----------

---

**Timing belt**

1



---

**Motion ball bearing**

2



---

**Stepper motor**

3



---

**Stepper motor holder**

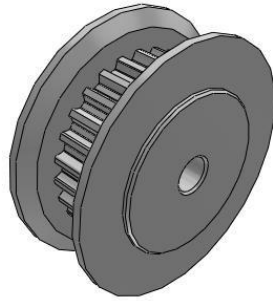
3



---

**3d pulley**

2



---

**Metallic rod**

4



---

**Rod holder**

2



---

**Magnet holder**

1



---

*Table 1 Mechanical Parts.*

## **3.2 Standards and codes:**

Software and IDEs used:

- Arduino IDE: to program the Arduino and the ESP32.
- Circuit.io and Fritzing: to draw the circuits.
- Draw.io: to draw the flow charts

### **3.2.1 Software Implementation:**

All of the ChessWhisper functions are controlled by code. Engage in player interaction, recognize human movement, compute AI movement, and control XY table motors.

We used these libraries:

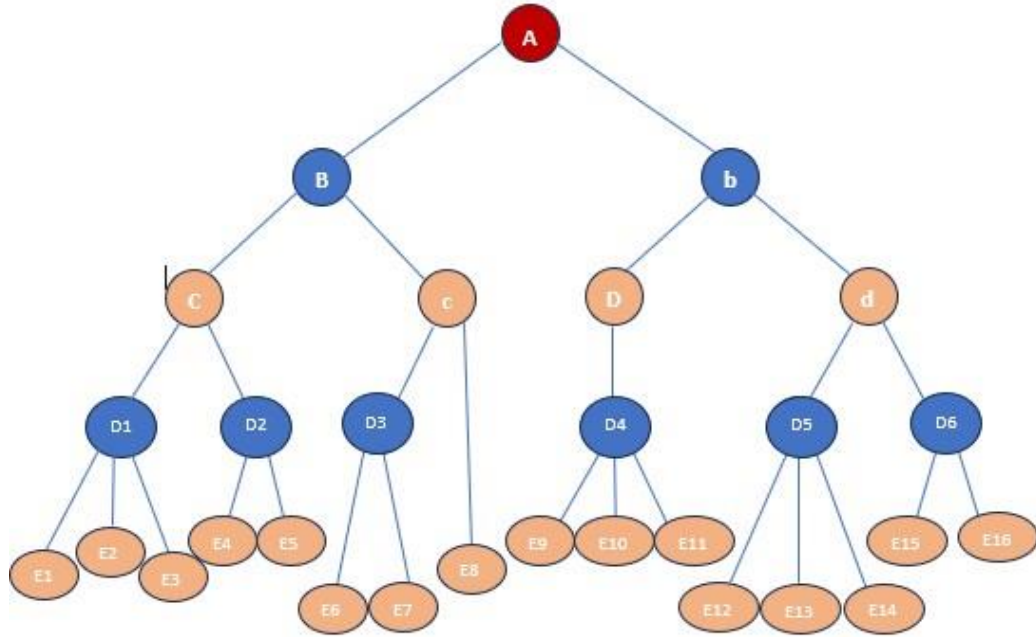
- Wire.h
- FastLED.h
- LiquidCrystal\_I2C.h
  
- WiFiClient.h
- BlynkSimpleEsp32.h
- Wifi.h

### **3.2.2 AI:**

The main feature in our project is the AI agent. Our AI is based on MinMax and Alpha-Beta algorithms.

MinMax → The Minimax algorithm operates much like the thought process of a human brain. During its turn to play, the AI thoroughly examines all potential moves it can make. It assesses the potential responses from the human player, predicting their actions and subsequent reactions in a comprehensive manner.

This analysis is visually represented by a decision tree, where each branch considers different moves for both the AI and the human player.



Max

Min

Max

Min

Figure 17 Decision Tree

Finding the optimal move between B and b is the aim of the analysis, with the top of the tree (A) representing the current chess piece position.

An evaluation function evaluates the bottom (E) and assigns a value depending on many criteria:

- Chess piece values that remain
- Positioning of chess pieces
- Separated pieces
- Positions of checks or checkmates

The chess piece locations are rated by the evaluation function, which assists in determining the position's advantage. Higher scores indicate that the AI has a higher probability of winning.

Following the assessment of each bottom (E), the ratings ascend the tree structure. The AI will keep the maximum value for the computer player's turn, maximizing its chances of winning. Additionally, the AI will minimize the AI's probability of winning on the human player's turn by assuming that the human will make the best move possible and retaining the minimum value.

An Arduino board cannot provide the necessary CPU power to assess every possibility in a chess game with a large number of moves and a higher game depth. The Alpha-Beta algorithm plays a crucial role in branch pruning.

Alpha-Beta: It is not required to assess each decision tree branch. The alpha-beta algorithm's function is to prune the decision tree precisely, and compares all the branches in the aim to prune the decision tree and reduce the calculation time.

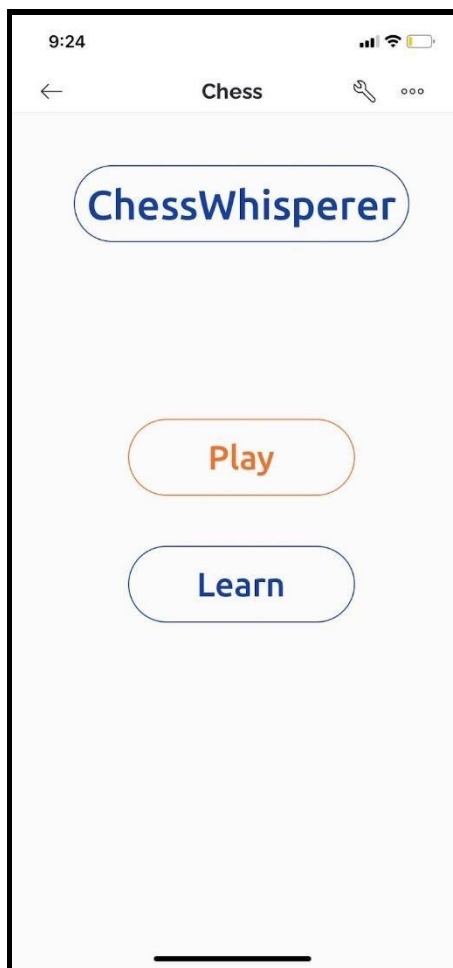
Detection of human player movements:

For the AI to complete its computation, it must know the displacement of the human participants. It has to know the move's arrival and departure coordinates in order to do that. A letter indicates the column and a number indicates the line, which determines the coordinates. The algorithm checks the state of the reed switches before and after the motion to calculate the displacement of the human player.

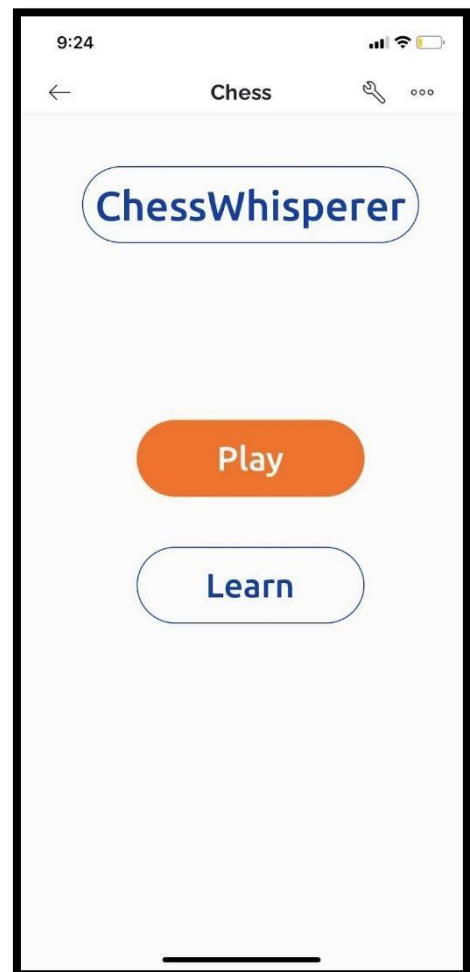
## Mobile Application:

We developed a very simple application on blynk to control the modes by the app instead of the actual buttons on the board.

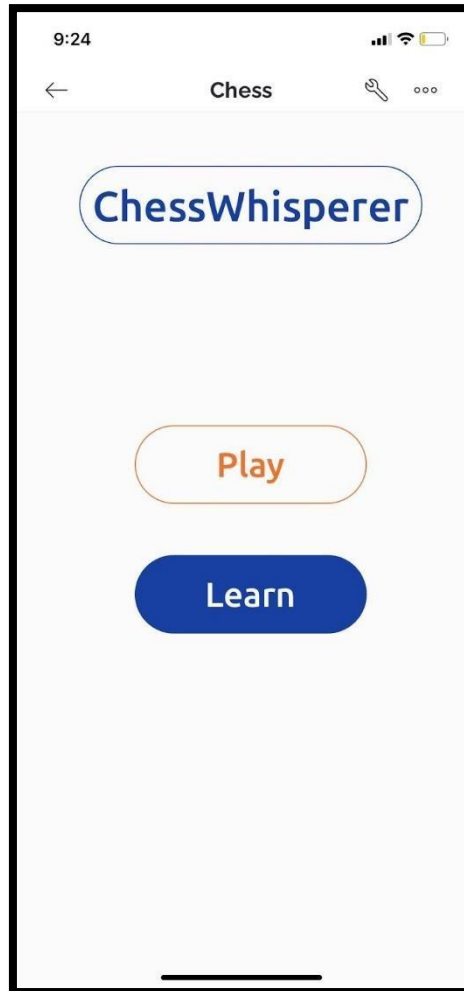
We connected blynk with the ESP32, and then with the Arduino.



There are two buttons for each mode



Mode Play



Mode Learn

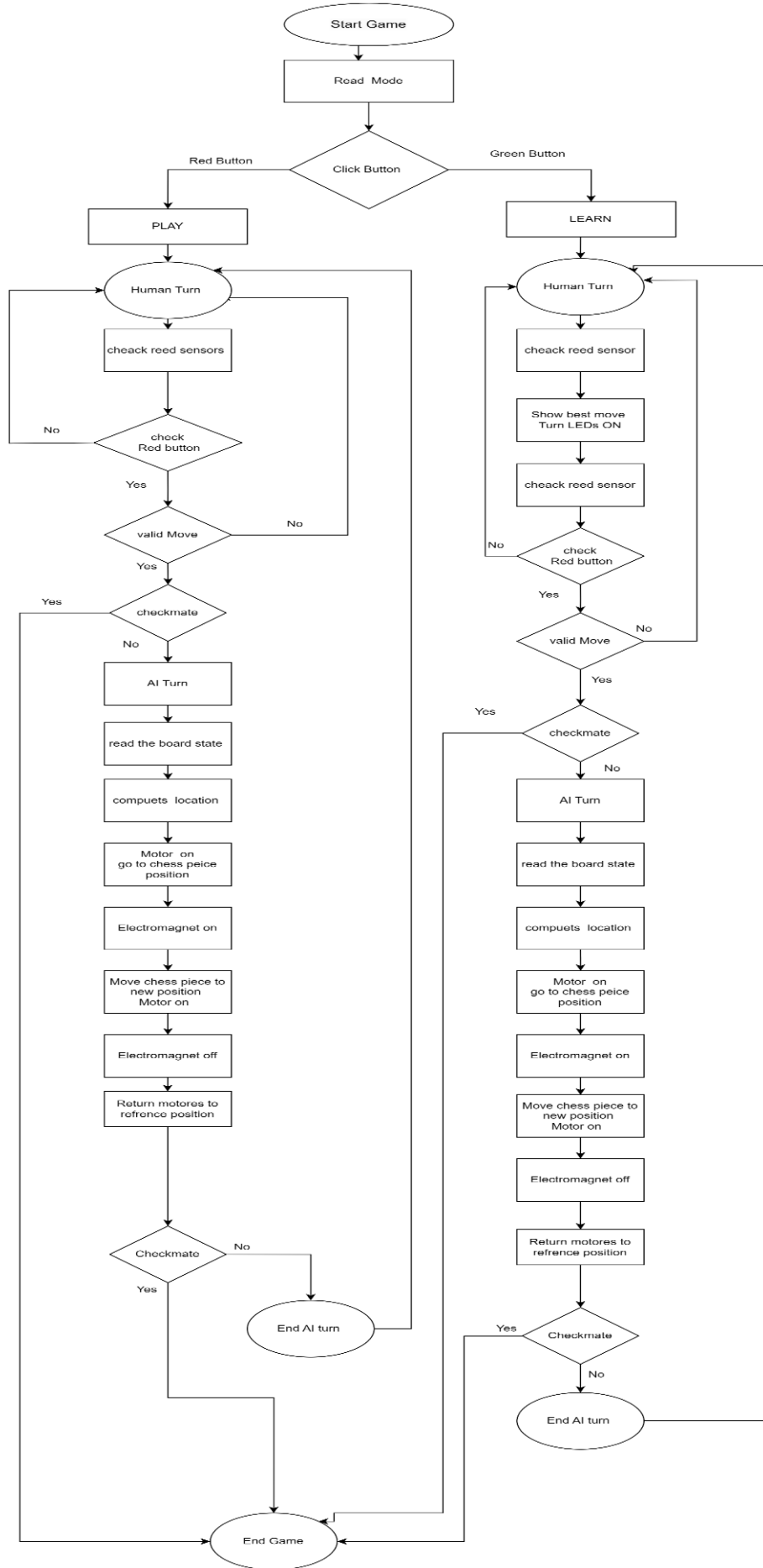
### 3.2.3 Flow chart:

At first, the user determines their preferred mode. If the player chose play mode, then after entering, the player has to move as legally as possible while the timer is displayed on the LCD. If the move is legal, the artificial intelligence (AI) will activate and the motors will move the chess piece using an electromagnet to the position that the AI computes. The player's turn then comes,

and so on. The LCD will show whether a kill was made for a piece by the player or the AI. It will become apparent at the end of the game who won and who lost.

In the second learning mode, the player starts the game. The LEDs turn on and decide the best piece to move and where to go. If the player doesn't take this move and decides to give another, it will be reviewed to see if it is a legal move; if not, it will appear on the LCD as an invalid move, and the player must choose another move or continue with the given move.

Finally, the AI's turn, which will be completed by motors and electromagnets in the bottom, will continue. The led color will always be green unless the given move has a kill state, in which case it will be red.



### **3.3 Constraints:**

1. We needed 64 reed switches, and it was very hard to find that amount so it took us a lot of time to find them, also they are very sensitive and could break easily.
2. The 16 channel analog multiplexers are not practical and after using them for some period of time they stop giving the right output.
3. Our project needed a lot of mechanical knowledge, which we had a hard time figuring them out.
4. For the learning part we needed an addressable led strip, but the only one we found, the leds were very small and very close to each other, therefore, we needed to cut and make some space then connect them, so we can fill all the squares. Also, it was expensive.
5. Because of the situation in our country at the moment, we lost a lot of days of work.
6. Our project in general cost us more money than we thought, and some of the parts were not available in our city.

## Chapter 4: Dissociation and Results.

At the end of this project, by creating a smart, automatic "chess Whisperer" with exceptional features like playing chess with AI, learning the best move, indicating whether a move is valid, and displaying the kill state, we were able to successfully complete this project and produce a chess board that is easy to use and enjoyable to play. Some challenges and how being solved:

1. providing power to the machine's different components. While some only needed 5 volts, others required 12 volts. We found a solution by using a power supply from an old computer that had 12 and 5 volts.
2. Some of the components were working correctly but after adding another component they stopped working or worked in the wrong way, after careful consideration, we united the ground.
3. We found that the chess piece could not be moved by the electromagnetic mechanism. The reason for this issue was that the plastic chess board was thicker than expected and was placed farther away than planned. We replaced the original board with a thinner board placed closer to the electromagnetic system in order to solve the problem.

## Chapter 5: Conclusion and Recommendations

In order to provide a fun and educational gaming experience, this chess project skillfully combines artificial intelligence, sensor technology, and electromechanical components. While the learning mode uses addressable LEDs to lead users toward the best chess strategies, the AI mode presents players with strategic moves. A complex hardware design is demonstrated by the combination of reed sensors, multiplexers, Arduino, LCD, electromagnets, and stepper motors.

The process provides a complete chess playing experience through the combination of automated movement mechanisms driven by electromagnets with a system that validates player moves. The addition of limit switches ensures accuracy in movement, enhancing the system's total dependability. In the field of hardware-based board games, this project is a comprehensive and interesting effort because it not only entertains but also educates chess fans.

### **5.1 Future Work:**

- 1- Make the learning mode more advanced, by lifting any chess piece and turn on the leds for the possible moves, also make a human vs human mode in the learning mode.
- 2- Switching from reed sensors to hall effect.
- 2- Develop our application.

## References

1. Shah, T., Chaudhari, D., Kshirsagar, A., Doiphode, A., & Mahajan, M. R. (2020). Intelligent Chess-board Using IoT. *International Journal of Research and Analytical Reviews*, 7(1), 545. Retrieved from <http://www.ijrar.org>
2. Sarker, S. (2015). Wizard Chess: An Autonomous Chess Playing Robot. In *IEEE Conference WIE*. Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.
3. Gillian, C., Koszykowska, O., Malik, S., & Wu, M. (2020). Title of the Instructive Smart Chess Board [Webpage document]. Trinity College. Retrieved from <https://www.trincoll.edu/engineering/wp-content/uploads/sites/31/2020/06/2020-Malik-Gillian-Wu-Koszykowska-CapstonePoster-Read-Only.pdf>
4. Arduino. (n.d.). Arduino Mega 2560 Documentation. Retrieved January 29, 2024, from <https://docs.arduino.cc/hardware/mega-2560/>
5. FreeCodeCamp. (2019, April 29). Simple Chess AI: Step by Step. Retrieved January 29, 2024, from <https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/>
6. ResearchGate. (n.d.). Applying Alpha-Beta Algorithm in a Chess Engine. Retrieved from [https://www.researchgate.net/publication/319390201\\_APPLYING\\_ALPHA-BETA\\_ALGORITHM\\_IN\\_A\\_CHESS\\_ENGINE](https://www.researchgate.net/publication/319390201_APPLYING_ALPHA-BETA_ALGORITHM_IN_A_CHESS_ENGINE)
7. Explain That Stuff. (n.d.). How Reed Switches Work. Retrieved from <https://www.explainthatstuff.com/howreedswitcheswork.html>