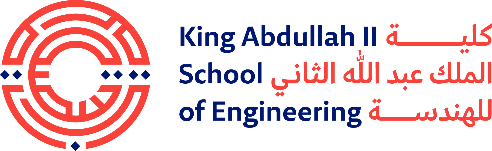
Princess Sumaya University for Technology

King Abdullah II Faculty of Engineering

Electrical Engineering Department



|  |
| --- |
| **Embedded Systems 22442**  **Final Design Project**  **Automated Chess Board System** |

|  |  |  |  |
| --- | --- | --- | --- |
| *Authors:* |  |  | *Supervisor:* |
| Laith Taani  Osama Alfaqeeh  Ahmad Alsayeh | 20200333  20200503  20200997 | Comp. Eng.  NIS.  NIS. | Dr. Belal Sababha |

*January 20, 2024*

***Abstract***

*The report details the process of designing of an Automated Chess Board System (ACBS) using PIC16F877A microcontroller to allow the interaction between physical chess play and digital strategy computation. The system utilizes stepper motors for piece movement and establishes serial communication for move execution. The programming is carried out in C for the microcontroller and Python for serial interactions. The focus of the investigation was to integrate mechanical components with software to autonomously play chess.*

Table of Contents

[1 Introduction 2](#_Toc500411309)

[1.1 Objectives 2](#_Toc500411310)

[1.2 Theory 2](#_Toc500411311)

[1.3 Equations Guidelines 2](#_Toc500411312)

[2 Procedure and Methods 3](#_Toc500411313)

[3 Results and Discussions 3](#_Toc500411314)

[3.1 Figures 4](#_Toc500411315)

[3.2 Tables 4](#_Toc500411316)

[3.3 Engineering Notation 4](#_Toc500411317)

[3.4 Guidelines for a Good Schematic 5](#_Toc500411318)

[3.5 In-Lab Questions and Measurements 5](#_Toc500411319)

[4 Conclusions 6](#_Toc500411320)

[5 References 6](#_Toc500411321)

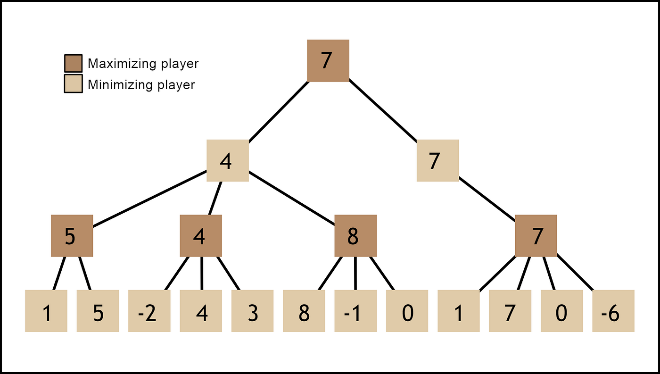
# Introduction

Chess is an ancient strategy game in which you aim to move pieces strategically to defeat your opponents and finally checkmate them. We want to combine the time-tested chess techniques with automation technology with our Automated Chess Board System. Chess players now have a platform to practice and enjoy the game by competing against artificial intelligence (AI). Players may successfully develop their skills and view their strategies play across the board. It transforms commands into movements on an electronic chessboard.

## Theory

The system is based on assigning values to pieces and positions to evaluate the best move possible. Each type of piece has a standard value: pawns are worth 1, knights and bishops 3, rooks 5, and queens 9. The king's value is set to 0 since it cannot be captured. The algorithm uses these values to calculate a score for the board at any given state, aiming to maximize the score for one player and minimize it for the other.

The decision-making process in the code uses a simplified version of the Minimax algorithm, selecting moves that lead to the highest possible score for the player while assuming the opponent will play perfectly to minimize that score.

Expressed as: , where P is the player and O is the opponent.

The algorithm evaluates all potential moves up to a certain depth and chooses the one with the best score after considering both the material and positional advantages. Changing the depth can be regarded as the level of the chess engine.

Figure Minimax Algorithm

# Design

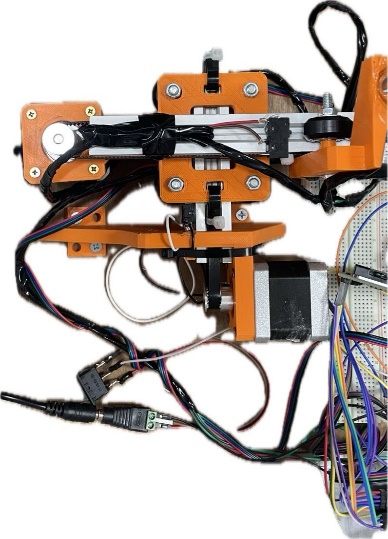
The design process for the Automated Chess Board System began with a detailed schematic design. This schematic served as a guide, detailing all electronic components and their connections. Then, each component went individual testing to ensure functionality. The NEMA17 stepper motors, were linked to DRV8825 drivers. These motors function by converting a negative edged step signal into precise 1.8-degree rotational movement.

Figure X and Y Axes

The physical structure of the chessboard was then constructed using 3D-printed parts, attached to aluminum profiles and GT belts for the X and Y axes, with careful measurements to determine the correct size of the squares on the board.

To ensure precise movements on the chessboard, limit switches were attached to define the 'home' position. Following this, an electromagnet was added for piece manipulation. The electromagnet was attached to a custom 3D-printed part and secured with a screw. Due to the 12V requirement for the electromagnet, it was connected through an NMOS transistor. Further tests were made to calibrate the chessboard, ensuring the movements executed by the electromagnet accurately corresponded to the designated chess squares.

Figure Electromagnet

With the physical components in place, the system's code was developed to interpret and execute chess moves received via serial communication from a Python script. The received data contains information on the type of move and the specific board squares involved. The microcontroller's program was refined to respond accurately to the serial commands, translating them into the precise movements of the chess pieces on the board. This ensured that the Automated Chess Board System could react to the player's strategy as communicated through the Python interface, completing the integration of the hardware and software components.

# Results and Discussions

The design and code for the Automated Chess Board System was functional in the movement of the magnet accurately. Nevertheless, the system's piece movement accuracy was approximately 80%. This was due to the magnet-to-board surface alignment and the variance in chess piece weights. The arrangement of the wiring also presented issues, occasionally disrupting piece movement. The initial code also posed a challenge due to its size (above 2K) which caused us to alter the design, were we had to remove the use of a keypad and depend on serial communication.

Possible fixes to the stated issues could include the use of a servo motor to control the position of the magnet accurately allowing it to be lowered when not needed. Another fix could be the use of a stronger magnet which allows to increase the distance between the magnet and the surface.

# Conclusions

With an astounding 80% accuracy in piece movement, the Automated Chess Board System demonstrates the effective integration of artificial intelligence with mechanical chess. Further improvement is possible, especially in the area of magnet alignment.