SENIOR DESIGN I

FALL 2023

Dr. Lei Wei

75 Page Report

INTERACTIVE CHESS BOARD FOR BEGINNERS

Group 6

Alec Barno	Photonics Engineer
Alejandro Felix	Photonics Engineer
Cassidy Phillips	Electrical Engineer
Vinicius Resende	Electrical Engineer
Nikolai Coletta	Computer Engineer

Advisors

Patrick LiKamWa	patrick@creol.ucf.edu
Justin Phelps	justin.phelps@ucf.edu

Table of Contents

1.0 Project Description

1.1 Project Background

Board games are a great way to spend time with friends and family. Like all new games, beginners learn how to play from a manual, or they might be lucky enough to have a player present that has played the game before and can coach them through a game. Some games rely more on luck while others depend on strategy. Chess is one of those board games that relies heavily on strategy. Professional chess players can often see 10-15 moves ahead and plan their actions accordingly when playing a match. For beginners though, they do not have that much experience to predict their opponent's moves, especially when they do not know the basics of the game yet. For chess beginners, having that extra person present to help guide them would be ideal, but that is not always possible. Our group wants to create a physical board that helps teach beginners the basic rules of chess and possibly build on it to further teach them basic chess theory.

1.2 Goals

The overall goal of this project is to create a physical chess board that will help teach beginners how to play chess. To teach players the basics of chess, they need to first be shown how to initially set up the board and how each piece functions on the board. In order to accomplish this goal, we first have to create smaller goals that need to be accomplished first. We have divided these smaller goals into basic goals, advanced goals, and stretch goals.

We have two basic goals for this project. The first is to detect and identify the different chess pieces. The second is to light the LED array underneath the chess board properly so that it shows the player where their selected chess piece can move to and to show a player where their piece cannot move to. These two basic goals are the most important goals we need to accomplish so that our chess board works as we intend it to.

For the advanced goals, we want to convey additional information about a player’s selected piece and to be able to save the game state for players to pick up and resume at a later point in time. By using a small built-in display, when a player selects a piece to move, the screen will display a small paragraph of text that describes how the piece moves. By using both the visuals of the board and the text on the screen, players should be able to gain a solid understanding of the basic mechanics of the game. This screen will also serve additional purposes based on the mode the board is setup in.

Our stretch goals for this project include making further use of the chess engine that will be programmed into the microcontroller to have an AI opponent for singular players to challenge against. This will allow singular players to still have an opponent to challenge even when a second player is not present. A second stretch goal we have in mind is implementing a small speaker into the board that will play sound effects when certain cases arise. We can play sounds that can both encourage and discourage players in their decisions on selecting the best moves.

1.3 Objectives

In order to achieve these goals, we must first figure out what steps must be taken to achieve them.

In order to detect and identify the different chess pieces, we must first create a system that will accomplish that. We plan to use fiber optic cables to carry light to each square so that it can be reflected to a photodiode by the chess piece that will be sitting on top of the square. The source of light for these fiber optic cables must be selected so that the maximum amount of light can be supplied to each square. By comparing different characteristics of fiber optic cables, such as length and core diameter, we will select the best fiber optic cable possible to carry and supply light to each of the 64 squares on the chess board. The fiber optic cable has to be positioned under each square so that the maximum amount of light is provided. The light from the fiber optic cables will then be reflected by the bottom of a chess piece. Each chess piece will have a mirror on the bottom to reflect any light but for each kind of chess piece, a different filter will be placed on top of the mirror. We can then calculate the amount of expected reflected light and assign a range of values for each piece. To detect this reflected light, each square will contain a photodiode. The photodiodes will have a change in current depending on how much light it is detecting. The more current it has, the more light it is receiving. By reading the current of these photodiodes, we can determine the type of chess piece on that square.

In order to implement enough LEDs below the chess board so that each square has an LED, we have to determine the best LEDs to accomplish our goals. Ideally, we want to be able to implement more than one color into the final project so that players can associate a certain color with a specific message. We then have to determine the best way to place these LEDs so that we can easily control which LEDs turn on and which ones we want to keep off. A driver must then be selected to control the LEDs so that consistent power and current are supplied to them without burning out any of the individual LEDs.

1.4 Engineering Specifications/Requirements

TABLE 1: ENGINEERING SPECIFICATIONS/REQUIREMENTS FOR THE COMPONENTS THAT WILL BE USED TO BUILD THE SYSTEM

Component	Parameter	Specification
Shift Register	Clock speed	24Mhz
Photodiodes	Viewing angle	50°
Fiber optic cables	Core diameter	0.75 mm (0.03 in.)
Battery	Watts	72.15 w
Display	Screen length size and Resolution	4'' 480 × 320 pixels
Microcontrollers	CPU speed	600 Mhz

TABLE 2: ENGINEERING SPECIFICATIONS/REQUIREMENTS FOTHE PERFORMANCE AND PHYSICAL CHARACTERISTICS OF THE SYSTEM

Requirements	Units
Battery Life of the entire chess box	≥ 4 hours
Chess Box Dimensions* (LxWxH)	24 in. × 18 in. × 6 in.
Chess Board Dimensions* (LxW)	18 in. × 18 in.
Weight of the entire chess box	≤ 10 Ibs
Delay/Activation from when piece is picked up to when move is show on the chess board	≤ 5 seconds
Currents to differentiate the types of chess pieces	6 different currents
Piece detection accuracy	≥ 95%

Asterisk (*): These dimensions are the max dimensions that our project will go to. We will aim to build the board as small and compact as we can.

Note: These specs in the table above are only estimations and guidelines for our project. We will aim to improve upon all the specs listed.

1.5 Hardware Block Diagram

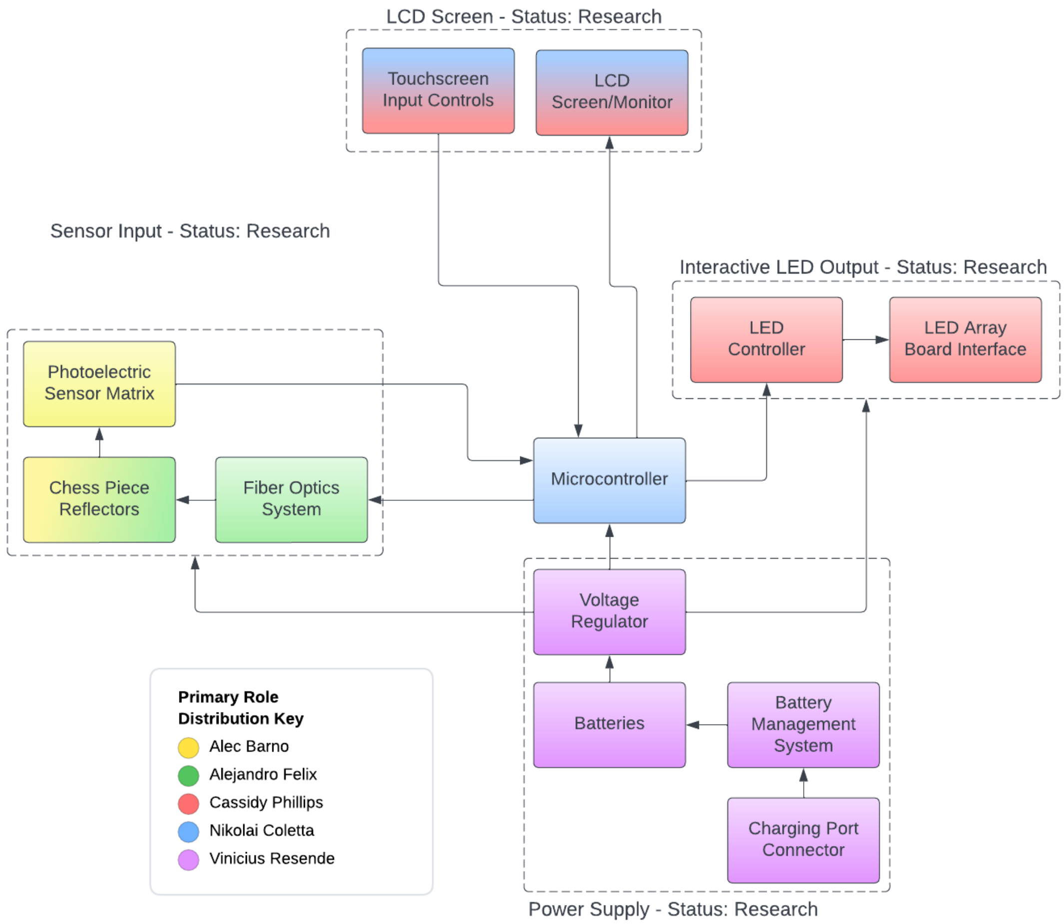


FIGURE 1: HARDWARE BLOCK DIAGRAM

1.5.1 Hardware Description Summary

This section will provide a description and overview of the project's hardware components focusing on the core infrastructure of our design in a manner to simplify understanding of the device's working principles and streamline the design process.

With the convenience of the user(s) in mind, a rechargeable battery would be the ideal power source for this project as it allows the device to operate without being tethered to a fixed power source, such as a power outlet. Batteries provide flexibility and portability, ensuring uninterrupted operation when it would otherwise be incapable of running. Along with it, a voltage regulator will be required to keep voltage consistent and stable during the device's operation regardless of system load or battery voltage output. This will prevent damage to sensitive electrical components due to overvoltage, prevent undervoltage conditions that can lead to improper operation or data loss, and increase overall power efficiency and battery life. This component pair will work to provide power to the microcontroller, sensors, LEDs, and any other components within the circuit.

To provide the system with the information it requires to operate, a matrix of photoelectric sensors will be in place to read the wavelength of the light from the fiber optic system underneath the board that is reflected by the bottom of the chess pieces and, from there, be able to verify the position of each individual piece in play. This group, labeled "Sensor & User Input" will act as the main source of data collection for our processing functions along with any other manual input from the user interface, such as signals to start or pause the game. Meanwhile, the components listed in the "Interactive LED Output" block will serve as the main form of interaction between the user and the board. Each square in the board will be able to individually light up to communicate different things to the player(s) during the game to convey helpful information to beginners.

A secondary method of communication and interactivity between user and device will be present in the form of an attached LCD Screen that will be used to present more detailed information that would otherwise be hard to convey using only the LEDs on the board. This information may include things such as game time, strategy, current game evaluation, etc.

To connect all these together, the microcontroller serves as the central processing unit for the design. It will receive and process data from all the sensors and inputs, as well as handle any logic and decision-making in a centralized manner to optimize system performance. It will also interface with any peripherals or additional components that should be integrated in the project at a future date, that are not essential for the device's operation but provide additional functionality.

1.6 Software Flowchart

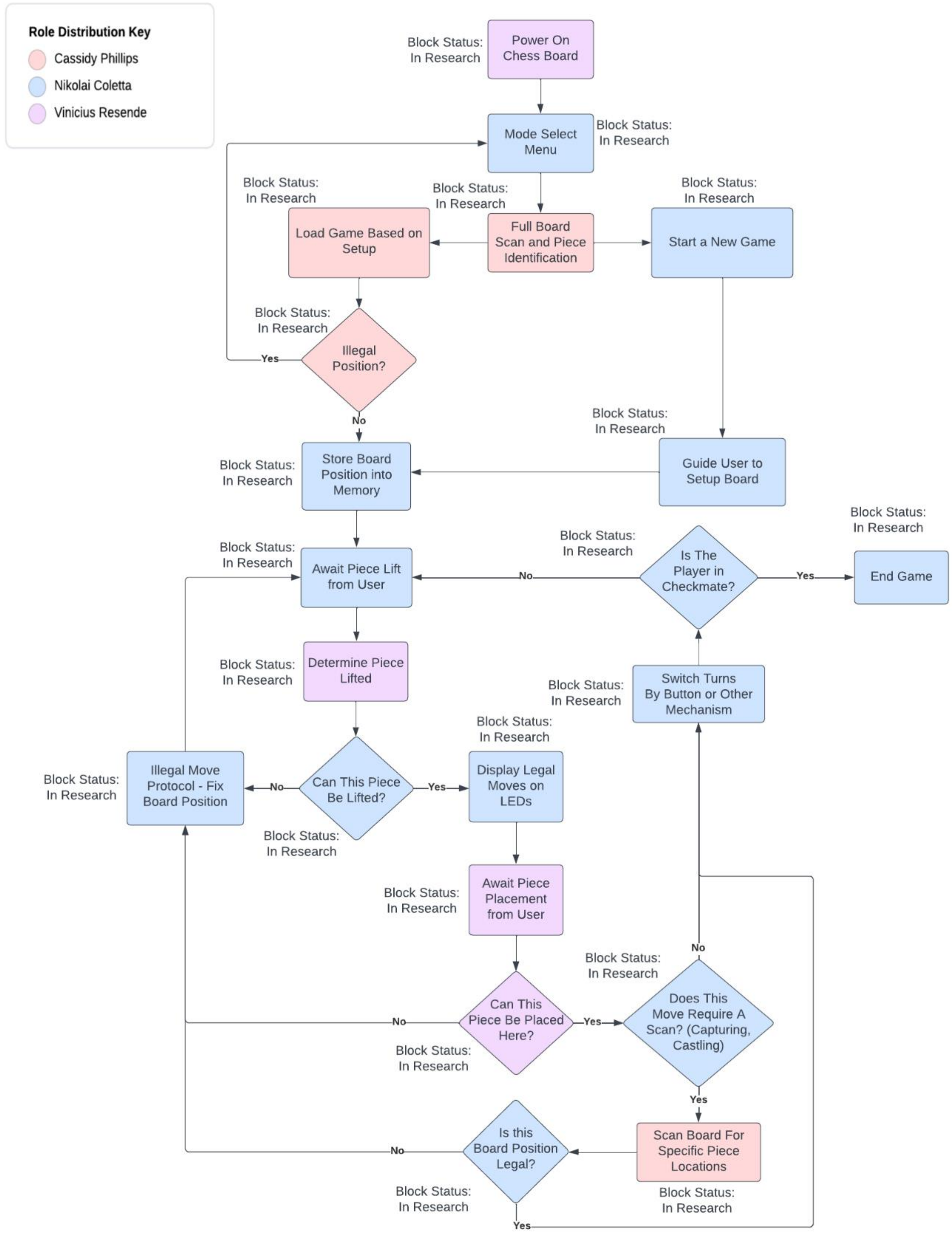


FIGURE 2: SOFTWARE BLOCK DIAGRAM

1.7 House of Quality

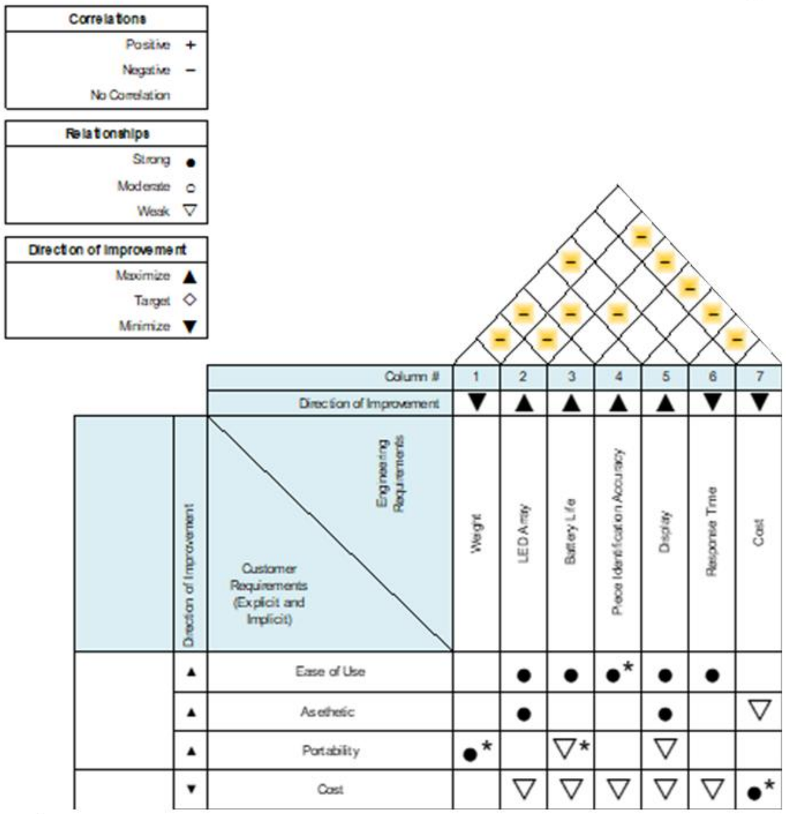


FIGURE 3: HOUSE OF QUALITY

1.8 Chess Gameplay Restrictions for Project

The rules of chess are the foundation for many of the interactive capabilities for this project. A variety of these rules pose restrictions for the design, and this project must be able to cater to each scenario on the board that can occur during a match. All rules of chess for this project, which includes concepts such as piece movement, piece capturing, castling, promotion, en passant, check, and checkmate, follow the standard rules of chess [1].

Currently, there are only three options available for the LED colors per square- red, blue, and purple. These options may expand, or change to different colors, but the system can easily be designed around these three colors. As explained more specifically in the following sections, these colors will be used to communicate certain information on the board, with the general rule of blue being used to show the user where to put a piece, red for where to remove a piece, and purple as replacing a piece. To keep this board accessible to audiences that are colorblind, there will be additional information displayed as needed on the screen display on the side.

1.8.1 Illegal Moves

Because this chess board is designed to help teach beginners on how to teach beginners, this board needs to be able to detect and fix all illegal moves. An illegal move in chess is a broad term that refers to any move, or lack thereof, that breaks the rules of the game. A simple example of an illegal move is moving the bishop directly left from where it as it. The bishop can exclusively move diagonally, so this type of move would be illegal, and the game cannot continue until the user undo's this move and makes a valid move instead. This covers the first category of illegal moves, which is simply done when a user moves a piece to an invalid square it can move to.

Whenever the user picks up a piece, the square it was picked up from will blink blue on and off, to show the user where to return the piece if either they change their mind or if they do an illegal move. All squares that the player can place this piece will be lit up blue, or any squares they can capture a piece will be lit up purple instead. For the scenario where the user places their piece in an invalid square, that square they placed it on will light up red, and the display will show an 'X' symbol with text below saying that that was an illegal move. The game cannot continue until a valid move is played, so the red light will stay there until the mistake is corrected. That is how the chess board will display these types of illegal moves.

It is important to note for making invalid moves, more must be considered than just the move-set of as given piece. In chess, there is a concept known as "pinning a piece", where a piece cannot move without sacrificing more valuable pieces behind it. In some cases, a piece may be pinned to the King, which can be legally sacrificed, so that pinned piece moving out of the way and exposing the king will be an illegal move. The protocol will be identical to the previous move correction- it will light up red on the incorrect square and continue to display the valid locations. Certain locations will also be removed on the board for pieces that are pinned to show a reduced list of available moves for these pinned pieces, to reduce further confusion.

Other illegal moves to consider for a beginner to do includes picking up too many or too few pieces. If the player makes a valid move, but then instead of passing there turn to the next player decides to make another 'valid move', there needs to be a system in place to both identify where illegally moved piece went and where it needs to return to. This system will need be extensively designed to make it as easy as possible for the user to return to the initial position of the board, while only using three colors and a display.

1.8.2 Recovering Disrupted Board States

Illegal moves can display helpful information for what went wrong when doing only single moves, but in the situation where multiple pieces are bumped over or switched by accident, the board could reach a disrupted state where multiple corrections are needed. The reason this board needs a guidance system to return the board state is really because it is designed for beginners, and a complete beginner might accidentally move a bunch of pieces at once, or even knock over a couple pieces to the ground.

There are two different routes to take when correcting disrupted board states from the players. First, every piece that is either illegally placed or missing can have LED indicators underneath, blinking red where they need to go and having solid red for underneath the squares that they are currently placed that needs to be removed (in the case that they aren't on the board, there will be no indicator besides the return point). This can get confusing though, as if multiple pieces are missing then there will be lots of flashing squares that the user can't distinguish between until trial and error.

The second solution could be just to have the first additional move displayed and corrected, and waiting until the user fixes that one before displaying the next error, until it is finally returned to a legal state. This method may get tedious if multiple pieces are taken off but will likely work the best in streamlining the corrections and making it as easy for the user as possible, so this method will be the one that is implemented. All LEDs will turn off instead of just the flashing blue of where a piece needs to go, and a solid red LED turned on underneath the piece that needs to be moved there. This is only done for cases where many moves are made instead of just the one.

1.8.3 Special Moves

In chess, there are three special moves that can be made that will have to alter the way the LEDs and display interact with the user. These special moves must be detected for and break the regular pattern done by just the simple capturing and moving of pieces.

Castling is the first of the three special moves in chess, which differs from all other moves in those two pieces of the same color move in that one move. Ignoring this rule would put the board in a disrupted board state as described in the section above. To detect and allow this, when picking up the king the additional move of castling will be shown in its highlighted available moves. After moving the king to its desired square for castling, the light under the rook will flash red and the one square that the rook needs to move for castling will be lit up blue. Note that these moves for castling should only be allowed if castling is legal at that point of the game. If the player has moved there king or rook already, castling on that side is no longer allowed (Or on either side if the king is moved in particular). In addition, if the king is in check, or passed through a square that would put him in check, castling isn’t allowed either. All of these would have to be considered when lighting up the square or not as a valid move for the king. For the sake of the user experience being smooth as well, moving the rook to a castling square like it would castle won’t display castling as a forced next part of the move, since that is a legal move of moving your rook by itself, but it won’t throw an illegal move if you move your king after that rook movement, if it is in the same turn. This way the user doesn’t have to undo the rook move, do the king move, then the rook moves if they want to castle but just moved the rook first.

En passant is another special chess rule in the sense that it is the only move to capture a piece that isn’t on the square you move to. What this would mean for the display is that instead of highlighting purple where the piece is directly on top of it, it will highlight purple where you move the pawn to do en passant, and then it can highlight red underneath the pawn you capture during en passant after the previous move was selected. With also detecting this move, en passant is only possible after the opponent on their most recent move moved their pawn two squares forward, and your pawn is in the right position. This opportunity passes though after your move, so if en passant isn’t taken at that moment it can’t happen again with that specific pawn. When determining if en passant is a legal move to make, it will have to consider the previous move made.

Lastly, promotion is a special rule in that as a pawn is pushed to the end of the board, it needs to be replaced with another piece, which is the only circumstance you would replace one of your own pieces. The way this would have to be implemented in this project is by having the pawn move to the last row result in the square being lit up red, and then blinking red until replaced with the promoted piece. Another the player may do it though is simply place down the piece being promoted to, so that needs to be a valid option for the player, by lifting their pawn for promotion and placing a different piece at that square.

3.0 Research

3.1 Existing Products, Past Projects, Similar Work

Currently there are multiple versions of this idea in the market. Some of the companies of these projects include Square Off, ChessUp, and Chess House. The prices overall range from $139 to $349.

The least costly of the three products we looked into is the Chess Genius PRO from ChessHouse. This board currently costs $139. Compared to what we want to implement into our project, this board does not have a lot of the aesthetic characteristics we are aiming for. The ChessHouse board does not make use of lights to provide information to the players, instead it makes use of a small LCD screen that is embedded into the board to display the piece movements to the player. This board is suited more towards single players because of its built-in chess engine (ChessGenius by Richard Lang). This board detects the piece movements due to pressure sensors built into the chess board squares.

The Square Off Pro chess board from Square Off is the best portability wise. Currently priced at $239, this board is flat and can simply be rolled up when it is done being used. This board only makes use of small white lights to convey which squares a player can move to. These lights do not light up the entire square though. The lights are on the line between squares so when a square turns on, the lights on the sides of it will turn on. The use of lights is simple, but it is effective enough to still relay information to the player. The chess engine this board uses is Stockfish. With this built in chess engine, players have the opportunity to play against an AI that can be adapted to different difficulties so beginners will not be overwhelmed. Two in-person players can still face off against each other on this board and the statistics about a game can actually be saved so that they can be analyzed later.

The ChessUp board is the closest to what we want our final project to accomplish. This board is currently priced at $349. The ChessUp board makes use of 3 different colored lights to relay information to the player as a game is played. These colors are used so that all possible squares a selected piece can move to are lit up, but the color used determines how strong moving to that square is. A companion app is also available for this board so the amount of assistance a side can receive from the board can be programmed in to even the playing field between two players with a wide gap in experience. This board ensures players play by the book and will not allow any illegal moves to be made. The board detects selected pieces as soon as a player touches any of the pieces on the board.

There was a senior design project done in the past at UCF from summer 2018 to fall 2018 that heavily inspired our project. The method this team used to detect pieces is actually the biggest difference between the two projects. This past team used a combination of reed switches and resistors to determine the pieces. Each chess piece contained a unique resistor value so that 12 different values of resistors were used in total, with each side having 6 unique resistors. When the pieces were moved, the equivalent resistance of the board would change and that was the value that was monitored to see how the chess pieces were moved. The microcontroller monitored these changes by using an ACDC converter. Even though they were able to accomplish the task of identifying the different pieces, their system was still not perfect. Some of the main issues they dealt with lied in their piece identification and detection system. The reed switches that were used were not consistent and they faced challenges of getting consistent readings of which piece was moved due to noise or poor contact.

Our goal with this project is to improve upon the previous projects' issues and limitations. We believe that incorporating optics into the project will enable us to create a chess board that is more consistent with detecting and identifying pieces. Using optics will also enable faster detection of the pieces and a quicker response from the programming and software to provide the users with a smoother gameplay experience.

3.2 Chess Piece Identification System

One of the most important systems for our project is the chess piece identification system. Our project's main goal is to teach new players how to play the game of chess by showing them where each piece can move to on the board. We needed to develop a chess piece identification system that would enable players to learn how each of the different pieces moves around the board, no matter the state of the game and where the pieces are on the board. Our chess piece identification system needs to be accurate, consistent and reliable or else the players won't be able to learn how to play the game of chess properly. Getting this chess piece identification system to properly work is of utmost importance because failure to do so will result in an unusable chess board and teaching tool.

For our chess piece identification system, we believe that using optics and photonics as the main principle for the system would create the best results possible. After doing extensive research on other "smart" chess boards or chess trainers, we believe that optics and photonics can fix and improve on some of the shortcomings and weaknesses in other systems that rely on magnets or pressure or switches. In our chess piece identification system, there are two main components involved: 1) the optical sensor and 2) the key to differentiate each piece. Understanding both of these components and the relationship they have was important in order to choose the best parts for this project. As outlined in upcoming sections, we will explain the different options we had and explain why we chose the specific parts that we did.

3.2.1 Optical Sensors

3.2.1.1 IR Sensors/Sensing

One of the first ideas we thought of for our chess piece identification system was to use infrared (IR) sensors. IR sensors are an optoelectronic device that can measure and detect infrared radiation. IR radiation is below visible light on the electromagnetic spectrum and therefore cannot be seen by humans. IR radiation ranges from the end of the red wavelengths from visible light all the way to microwaves, typically from 0.75 µm to 1mm. IR sensors are typically used as motion detectors or proximity sensors. We first thought that using IR sensors and putting one under each square for our piece identification system would have worked, but we encountered a few issues with that idea. The main issue with using the IR sensors as the optics for our piece identification system is that being able to differentiate each different type of chess piece would have been difficult to achieve using the IR sensors. We would have had to adjust each different piece's IR emittance, whether that was through different LEDs or other type of source. That would have become too complex and tricky to properly figure out since each piece would have required a power source and circuit design to fit inside the pieces. IR sensors are relatively cheap, going from anywhere between $1 - $10 on average, but the components that would have been needed to go inside each piece to differentiate the types of pieces would have been expensive and much more difficult than necessary. For those reasons, we decided not to use IR sensors as the sensing mechanic for our project.

3.2.1.2 Color Sensors/Spectrometers

Another idea that we had for our chess piece identification system was to use color sensors. Color sensors are optoelectronic devices that can determine the color of an object. The color sensor emits light, typically by an LED, and then uses photodiodes to collect the reflected light from an object. The light received by the photodiodes is converted into an intensity value that can be split into a ratio of colors, typically on an RGB scale and can accurately determine the color of the object. For example, if the object is yellow, then the reflected light received by the color sensor would break down into red and green intensity values, but not blue. Using color sensors would make differentiating the different kinds of chess pieces very simple and straightforward as we would just have to color code the pieces accordingly and provide a key for reference. The main issue with using color sensors as the optics for the chess piece identification system is the cost. Color sensors typically range in price from $5-$20 per sensor. Since we need 64 sensors, one for each square of the board, that would become very expensive to do. The cost of the sensors needed for the project could range from $320-$1280. Since our project is not sponsored by anyone and all of us are in college and don't have large bank accounts, we decided that we couldn't use color sensors for this project.

We did however try to come up with a different approach to still use color in our chess piece identification system as color coding each of the different pieces would be the most simple and straightforward way to help teach brand new players how to play the game and learn how the pieces move. Another idea we had for the optical sensing of our chess piece identification system was to create a spectrometer system. A spectrometer is a scientific device that is used for detecting and separating wavelengths of light on the electromagnetic spectrum. Spectrometers are very sophisticated machines and therefore are very expensive. Spectrometers typically cost in the range of hundreds to thousands of dollars, sometimes even tens of thousands of dollars. We obviously knew we couldn't buy and use a spectrometer in our project, but we tried to develop a way to make a "homemade" spectrometer that would still allow us to differentiate each of the pieces using color coded bottoms. We thought of an optical sensing system that would use photodetectors and color filters to achieve the same result as using a spectrometer. We thought that we could use a white light source aimed at the bottom of each square, underneath the board, and point towards the bottom of a piece that would be on the square. Then the white light would be reflected from the bottom of the chess piece and go into the photodetector. The photodetector would convert the reflected light into an electrical signal that would then be related to a specific color.

We encountered a few problems with this system design. The first problem we encountered with this design is that a photodetector cannot distinguish between various wavelengths of light when it receives the reflected light and converts it to an electrical signal. We would have needed to use color filters on the bottom of the chess pieces to differentiate them and also needed to put the color filters on top of the photodetector to only allow that specific color and its associated wavelengths through into the photodetector. That wouldn't be a major problem if there was some way to have all of the different colors related to the different chess pieces that could be filtered through one photodetector. But we are not able to do that as stacking all of the color filters onto the photodetector would just allow every wavelength of visible light to come through and it would just read the reflected light like it was just the white light. Since there are 6 different pieces in chess, there would be 6 different colors to differentiate through the photodetectors. That means we would need 6 photodetectors under each and every square on the board. Our project would need 384 photodetectors for the entire chess board. Each of the photodetectors would have a color filter on it so that it would only receive that particular range of wavelengths of light coming from the bottom of a certain kind of chess piece. Photodetectors are relatively cheap, ranging in price from $0.50-$5, but since we need 384 photodetectors for the project, the price for our piece identification system would drastically increase. The cost for all of those photodetectors could have ranged from anywhere between $192 all the way up to nearly $2000. We decided not to go with this design of the chess piece identification system as the price and complexity was just too much when considering the other options available.

3.2.1.3 Photodiodes

The last chess piece identification system design that we came up with is using photodiodes. Photodiodes are a semiconductor device that takes in light and converts it into an electrical current. Photodiodes are very similar to photodetectors and sometimes they get confused and mistaken for one another. Photodetectors are a more general term for optical sensors whereas photodiodes are a specific type of photodetector because of how it works. Photodiodes typically use a P-N junction in order to produce a photoelectric current. The photons (light) get absorbed into the P-N junction and they create an electron-hole pair that moves through an electric field, which creates an electrical current. The design of this system using photodiodes is very similar to the design of the system that would have used photodetectors and color filters. In this version of our chess piece identification system, we will have one photodiode under each square of the chess board and it will measure the intensity of the reflected light from the bottom of the piece on top of the board. This intensity value of the reflected light will then be converted into an electrical current and we will determine what piece it is based on the current value we read.

There are a few advantages in this system design over our ideas about using photodetectors. For example, when using photodiodes, we are only concerned about intensity of light. We don't have to worry about relating the different kinds of chess pieces to a specific color. This gives us some flexibility when determining how we will differentiate the six different kinds of chess pieces, which will be further explained later on. We also only need one photodiode under each square of the board instead of six photodetectors with the previous system. This will reduce the cost of the piece identification system overall and also give us more room to be able to adjust the size of our chessboard and the dimensions of each square, piece, etc . Photodiodes are very similar in price to photodetectors, ranging anywhere from $0.50-$5. The realistic price for all sixty-four photodiodes will be in the range of $30-$50 and shouldn't be anymore than $64 ($1 per photodiode).

Another distinct advantage for using photodiodes is the simplicity of the entire system overall. Since photodiodes will convert the reflected light into an electrical signal (current), we just need to design a simple circuit to be able to properly determine the six different kinds of pieces. We will be able to control the resistance, input voltage and overall design of the circuit, which will enable our group to just focus on measuring the current produced by the photodiodes. Since the current produced by the photodiodes is classified as analog (continuous-time varying), we need to convert that into a digital signal so that the microcontroller can read the value and determine if/if not there is a piece on a specific square. To convert the signal from analog to digital, we will need to buy and incorporate into our system an analog-to-digital converter. An analog-to-digital converter (ADC) is a device that converts an analog signal measurement into a digital value in the form of binary code. Since an analog signal is continuously changing, the ADC has to first "take a snapshot" and then quantify that data into binary digits, or bits. Once the signal is quantified and given a resolution in bits, then the data can be read in binary values and sent to the microcontroller as a digital signal.

TABLE 3: OPTICAL SENSORS TECHNOLOGY COMPARISON

	IR Sensor	Color Sensor/Spectrometer	Photodiodes
Function	Measures IR radiation changes	Measures and determines specific wavelength ranges related to color	Measures intensity of light
Wavelength Range	0.75 µm to 1 mm	≅ 380 nm to 750 nm (Visible Light)	Any wavelength range (specific to diode specs)
Extras/Other Components	Light source and circuit inside each chess piece along with some filter or other method to differentiate them	Color filters for each different kind of chess piece	Analog-to-Digital Converter (ADC)
Implementation	Medium to Hard	Hard	Easy
Number of sensors	64	64 - 384	64
Cost (1 sensor)	≅ \$1 to \$10	≅ \$5 to \$20 (color sensor) ≅ \$100s to \$1000s (spectrometer) ≅ \$0.50 to \$5 (photodetectors)	≅ \$0.50 to \$5

After considering all options and determining the pros and cons of each system design, we have decided to create our chess piece identification system using photodiodes as our optical sensors. Using photodiodes enables us to have great flexibility with how we determine the different kinds of chess pieces; it is the most cost-effective option, and it is the simplest option for data collection and implementation into our chess board.

TABLE 4: PHOTODIODES PART SELECTION

Name	Photosensitive Diode	VTP9812FH	SFH 203 P
Brand/Manufacturer	Uxcell	Excelitas Technologies	ams-OSRAM USA INC.
Seller	Amazon	DigiKey	DigiKey
Peak Wavelength	940 nm	580 nm	850 nm
Spectral Range	400 nm - 1100 nm	400 nm - 700 nm	400 nm -1100 nm
Receiving/Viewing Angle	40°	50°	150°
Active Area	N/A	1.55 mm^2	1 mm^2
Price	\$0.65 (per diode) \$6.49 (10 pcs)	\$0.67 (per diode) \$59.14 (100 pcs)	\$1.00 (per diode) \$6.50 (10 pcs) \$42.60 (100 pcs)

After deciding that we will use photodiodes as the optical sensors for the chess piece identification system, we needed to determine the best photodiode to use for our project that fits our requirements and needs. The most important qualities about photodiodes that affect our project specifically are the spectral range that the photodiode can read, the peak response wavelength for the photodiode and the receiving/viewing angle that the photodiode can detect light. After doing extensive research, we determined that the photodiode we will use for the chess board is model VTP9812FH from Excelitas Technologies. We chose this specific photodiode because it best fits our parameters for the project. Since we will be using light in the visible wavelength spectrum to go into the fiber optic cables that will illuminate the bottom of each square on the board, we needed a photodiode that can not only read visible light wavelengths but also have a peak response wavelength in that spectrum. The photodiode we chose has a spectral range of 400 nm to 700 nm, which is roughly all the visible light spectrum, but it also has a peak response wavelength of 580 nm. Currently, we plan on using green light as the color of light that will go through the fiber optic cables, so our choice of photodiode has a peak response wavelength very close to wavelength range that we are using. That is very good as it will lead to an increased responsivity to measuring the reflected light from the bottom of the board. Another reason we chose this specific photodiode is that the receiving/viewing angle is 50°. For our project, using a photodiode with a large viewing angle was important as the photodiode will be measuring reflected light from the bottom of the chess board. The larger the viewing/receiving angle, the better our measurements will be, and it will help us determine more accurate values when differentiating the different types of chess pieces, which will result in higher correct piece detection accuracy during a game. The “cherry on top” for choosing this specific photodiode is active area size. Our photodiode has an active area size of 1.55 mm^2, which was much larger than most of the other photodiodes we researched and considered for this project.

3.2.2 Differentiating each piece

3.2.2.1 Colors/Color Filters

The first idea we thought of when trying to figure out how to differentiate each of the six different kinds of chess pieces was to use color. We thought that color coding the pieces would be the most simple and straightforward way to classify the piece types. We believed that color coding the pieces would help the players learn each piece faster as there would be a visual stimulus to relate each piece to and it would also make setting up the board before a game easier to understand and memorize. The only way we could color code the bottom of the pieces is by using color filters. Color filters are a type of filter that can affect the color of light either through absorption, reflection, conversion and/or balancing of any range of wavelengths in the visible spectrum. We would have needed to use color filters on the bottom of each piece and on top of all of the photodetectors. These color filters would only allow specific ranges of wavelengths of light through to the photodetector and classify each piece type based on each specified wavelength. Since we would have used a white light source under each square to illuminate the bottom of the chess board, the reflected light would change from the broad spectrum that white light has to a very small range of wavelengths corresponding to a specific color. This would have enabled the microcontroller to easily identify the different piece types and allow us to have a high accuracy for determining pieces during any game.

Color filters are relatively cheap and usually come in a pack with a bunch of different colors. These filters are typically sold as big sheets that can be cut to specific shapes and sizes, depending on the needs of the customer. The filters can cost anywhere from $10 to $30, but that usually includes most of the basic colors of the rainbow and multiple sheets for each color. If we decided to use color filters as the way to differentiate the different chess pieces, we would have only needed 1 pack of filters since the bottom of our chess pieces will be small in size when compared to the dimensions of the filter sheet. The reason we won’t be using color filters is not because of the color filters themselves, but because of the cost and setup of the optical sensors for the piece identification system. Color filters are our backup plan to use in case the other kind of filters we use to differentiate piece types doesn’t go according to plan.

3.2.2.2 Materials

Another way we determined how we could identify and differentiate each of the chess pieces was to use different materials. We thought that using various materials that vary in reflectivity could work since we are using photodiodes which just measure intensity of light. Choosing materials based on how reflective they are to identify piece types would work with our choice of photodiodes and optical sensor setup. We could have used any sort of material in the world and they wouldn't have had to relate to each other. We thought of using a mirror, shiny reflective tape that is typically used in construction, aluminum foil, a penny, and many more random objects could be used as they all are reflective at various levels.

The main issues with using materials and relating their reflectivity values to differentiate piece types is that there is no source or large amount of data about different reflectivity values for various materials and sources. There is not much information gathered about the exact reflectance values of random objects in the world, which means we wouldn't be able to provide an exact and precise reflectance value without lots and lots of testing. We would need to have tested many different materials and objects dozens if not hundreds of times so that we could get an accurate average value for the reflectivity. That wouldn't have been an issue if we only needed a couple of different values, but since there are six different kinds of chess pieces, we would need at a bare minimum six distinct reflectivity values of materials that wouldn't interfere with each other. For our project specifically, we would like to get as many differentiating values as possible to make it easier for the programming and coding of our board to be as smooth as possible. To have done that would have required dozens and dozens of tests with many various materials such as tapes, metals, mirrors, plastics, etc. and the values we would come up with would only be estimates and averages as there is not truly a source that we could fact check our numbers with. It could still be done, but because of the complexity and extensive amount of testing needed to be done, we decided to not differentiate our chess piece types based on the reflectivity of various materials.

3.2.2.3 Light Filters

The last concept we thought of to differentiate the piece types is to use filters. There are many kinds of light filters that can change light in many various ways. Light filters are designed to narrow the range of wavelengths of light to achieve a specific goal or objective. There are some light filters that specifically absorb certain ranges of wavelengths or reflect certain wavelength ranges. There are also filters that can refract or diffract certain wavelengths as well. For our project, we would be focusing on neutral density (ND) filters. ND filters are light filters that are designed to reduce light intensity evenly across a specified wavelength spectrum. ND filters are typically neutral gray in color and act as a mask to control the intensity of any illumination source. ND filters are characterized by the value at which they reduce the intensity of light. The value that each ND filter has is tied to a percentage that denotes the intensity of light that is transmitted through.

TABLE 5: DIFFERENTIATING EACH PIECE COMPARISON

	Colors/Color Filters	Materials	Light Filters
Function	To differentiate the chess piece types based on color and wavelength associated to colors	To differentiate the chess piece types based on intensity of reflected light due to the reflectivity values of various materials and objects	To differentiate the chess piece types based on overall intensity of light
Wavelength Range	≅ 380 nm to 750 nm (Visible Light)	Any wavelength range (depends on wavelength of light source)	Any wavelength range (specific to wavelength of light source)
Implementation	Medium	Hard	Easy
Cost	≅ $10 to $30	≅ Free to the price of whatever material is used	≅ $10 to $35

The main reasons why we decided to use ND filters for our project to differentiate each of the 6 kinds of chess pieces is because of the simplicity of ND filters. Using ND filters allows us to associate a specific number to a certain kind of piece and we will know how much light is transmitted through. ND filters are just as simple as color filters, but they don't require the same level of setup with the optical sensors. We only need to use one photodiode under each square of the board to measure the intensity of the bottom of a piece. ND filters are also relatively cheap and equivalent to color filters, ranging anywhere from $10 to $35. When buying ND filters, they usually come in a pack with three to four different values of filters in there so there are different options depending on the purpose for the filters. Overall, we decided to use ND filters because they are the most simple and easy to use when compared to the other options for how we could have differentiated the chess piece types.

TABLE 6: FILTERS COMPARISON

Name	Square Filter Kit	Gel Filters, CTO Transparent Light Sheets	Lighting Neutral Density Gels Filter Sheet
Brand/Manufacturer	SIOTI	Meking	RENIAN
Seller	Amazon	Amazon	Amazon

# of filters	4	4	6 (2 per each kind)
Value of filters	ND2 (½ transmission) ND4 (¼ trans.) ND8 (⅛ trans.) ND16 (1/16 trans.)	1 ½ ¼ ⅛	ND3 (≅48% trans.) ND6 (≅24% trans.) ND9 (≅12% trans.)
Type of filters	Neutral Density (ND)	Color Correcting/Enhancing	Neutral Density (ND)
Material	Pmma	Polyester	Polyester
Price	$22.99	$18.99	$16.98

Once deciding that ND filters would be the best way to properly differentiate the chess piece types, we needed to figure out exactly how we will implement that into our project. The main concern for differentiating each chess piece type is determining how many different values we need the photodiodes to measure so a proper game of chess can be played. There are 6 different chess piece types. Ideally, we would want to have 12 different values to be associated with each of the 6 different types of pieces for both colors, white and black. Unfortunately, getting 12 distinct values measured from the reflected light from the bottom of the chess board will be extremely difficult and has increased risk of overlapping that will cause the microcontroller to read the wrong values and therefore light up the wrong moves that a player could make. Currently the plan is to get 7 distinct values to differentiate the pieces on the board: 2 different values for each set of pawns (white and black) and 5 values related to the other chess piece types. We want to differentiate between each set of pawns as pawns are the only piece that cannot move backwards. If we differentiate between them, then that will allow the microcontroller to more easily track and determine which pawn is what color and what direction they should move in. We don't need to worry about what the color is for the 5 other types of pieces as they can all move backwards. Their direction of movement isn't strict like it is for pawns.

To achieve 7 different values that will differentiate the chess pieces on the board, we chose 2 different sets of ND filters. The first set of ND filters has 4 different filters: ND 2,4,8, and 16. Each different filter reduces the intensity of the reflected by a certain factor. The second set of ND filters we chose has 3 different filters: ND 3,6, and 9. By using these 2 sets of filters, this provides us with 7 possible options to differentiate the pieces without even counting the base layer that each piece will have of just the mirror. The base layer that doesn't have any filter and only the mirror will act as 1 value as it should show exactly how much light is being reflected from the bottom of the pieces on the chess board. These 2 sets of filters provide us with options and flexibility when determining what values, we want to control each piece with. It gives us the chance to test different combinations and precisely choose 7 distinct values for our project.

3.3 Fiber Optic Illumination System

3.3.1 Fiber Optic Cables

Optical fibers, which are very similar to wires, are typically made from different types of optical glass or plastic. They enable light to travel through them and can be thought of as similar to electrons in normal electrical wire. Optical fibers are flexible, which enables them to be bent to a certain point and be easy to use since they aren't rigid. They are mostly used for communications since they allow information to travel at the speed of light and have less loss in signals than wires. This isn't to say they have no loss, but they have significantly less loss than signal travel through an electrical medium. Fibers are also used for illumination. They illuminate objects that can be hard to light up due to them not being in line of sight. This is done when light enters through one end of the fiber and exits the other. It will not exit in a straight line, as a laser light ray does, the light typically disperses when it exits the fiber from the opposite end. For this reason, they can be used to illuminate in confined places which is only enhanced by their flexibility. In addition, we can group them together into bundles that enables them to transport the light to the target with a greater intensity.

3.3.1.1 Fiber Optic Properties

Optical fibers are made up of a core that is enclosed by a typically clear cladding material that has less refractive index than the core. Light then is enabled to travel in the core due to total internal reflection since the cladding has a smaller refractive index. Total internal reflection keeps all the light in fiber. This allows the fibers to function like a waveguide, which is an optical device that guides electromagnetic waves (light) through a medium.

When light travels through any medium or bounces off any surface, absorption and scattering occur. These two effects cause losses in a fiber which will reduce output power. Attenuation in a fiber is determined by the wavelength of the light meaning if there are losses due to absorption or scattering the color of the light that is emitted at the end will be different than the light from the illumination source. Longer fibers will cause the light to be marginally redshifted. In addition, because of absorption and scattering loss there will be less light output at the end of the fiber, which results in lower intensity.

Fibers with larger core diameters can couple, which is combining two or more inputs into a single output, more light. Doubling the core diameter would increase the coupling power by four since the cross-sectional area would increase by four.

Another way to improve coupling power is to increase the density of fibers in a bundle. The ends fiber have to be very clean as well as contaminants such as dirt, dust, and finger prints can cause light to be absorbed which would cause output light to be lost. In our application, a larger core diameter would mean more light transmitted through because the fiber would let in more light from the input source.

Modes, as applied to optical fibers, can be thought of as paths that a ray of light can take. When a fiber’s diameter is increased so that its V number, the normalized frequency parameter of a fiber, is greater than 2.405, then the fiber can support more than one mode of light. Fibers that have one transverse mode are called single mode fibers and those with multiple transvers modes are called multimode fibers. The fundamental differences between single and multimode fibers will be explained further below.

Single mode fibers usually tend to have smaller core diameter than multimode fibers. The only mode a single mode fiber can support is the fundamental mode or the LP_{01} mode (the subscripts describe the radial and azimuthal variations). This happens because the fiber diameter is so small that the light travelling through it interferes with itself and will always result in a gaussian beam given the correct angle input.

The main dispersion mechanisms in single mode fibers are from material dispersion and waveguide dispersion. Since refractive index is wavelength dependent, different wavelengths travel at different speeds through different materials. This phenomenon is called material dispersion. An example of waveguide dispersion is that longer wavelengths are less confined, since they have a lower frequency, they will experience a lower refractive index. Because of this property, single mode fibers are usually designed with one wavelength in mind that is optimized for that specific wavelength. A way to minimize dispersion in single mode fibers is to control the relation between refractive index and core radius. We can do this by choosing a different material for the fiber or choosing a different core diameter.

An advantage of single mode fibers is that intermodal dispersion can’t occur. Intermodal dispersion is when different modes travel at different speeds. Not having intermodal dispersion is good for optical fiber communication for high data rates, especially for longer distances. In addition, single mode fibers have lower propagation losses which is another reason why long-distance data transmission and outdoor uses are all done using single mode fibers. This can be seen in data centers or even fibers that are used to transmit data across an ocean or country.

Multimode fibers however suffer from multimode dispersion. In a multimode fiber many modes can form at once and different modes can propagate at different or similar times. When these modes interfere with each other, this causes the superposition of all these modes. Multimode fibers experience many dispersion mechanisms such as chromatic dispersion, intermodal dispersion, and polarization mode dispersion. Chromatic dispersion is inclusive to material dispersion and waveguide dispersion. Polarization mode dispersion is when different polarization components travel at different speeds, such as linearly polarized light in opposite directions (vertical and horizontal) or circularly polarized light (left or right circularly polarized). Dispersion in multimode fibers can be minimized by using a graded index profile, which is a gradual change in the refractive index from the core to the cladding rather than a sudden change, such as a step index fiber.

Multimode fibers usually have a higher numerical aperture (NA) which allows for greater guidance of light even when bending. However, higher propagation losses occur when not bending the fiber. Getting light into multimode fiber is easier than in a single mode fiber because the location and propagation angle of the light don't have to be as precise. However, the spatial coherence of the output is diminished, and the resulting field pattern could be hard to manage. Multimode fibers are mostly used for short distance indoor applications because they allow for cheaper multimode data transmitters such as LEDs rather than laser diodes.

The max angle a ray can hit the core of the fiber, which is determined by the numerical aperture, can allow that light to travel in the fiber more effectively. This is called the acceptance angle. The numerical aperture is defined as the value of the sine of the maximum acceptance angle times the refractive index of the surrounding medium, which is typically air. The numerical aperture can also be determined by taking the square root of the squared individual refractive index difference between the core and the cladding. Increasing the NA of a fiber would also increase the cone of acceptance of the incident light enabling it to get into the fiber easier.

3.3.1.2 Illumination Fibers

Fibers used for illumination are similar to fibers used for communications except they have larger cores and smaller claddings to increase the amount of light received from the illumination source. Most illumination fibers are multimode fibers since they have bigger diameters to allow light to get in easier. Lighting fibers that are used purely for illumination purposes can be made out of glass or plastic.

Glass fibers, when compared to plastic, are more efficient at carrying light because they're more transparent and also more resilient to heat from the illumination source. This enables the illumination source light to be more intense, which gives more light at the end of the fiber. Plastic fibers are more common since they are very cheap due to material availability and drawing time and are easier to set in place. The drawback of plastic fibers when compared to glass fibers is that they have a higher loss of light and can't handle high heat which sometimes limits the amount of light the illumination source can output. The material that the core and cladding are made up of determines the amount and which modes appear.

Modes are defined by the numerical aperture (NA). Light leaves fibers at the end as cones and the higher your NA number is the wider the cone of illumination will be. Fibers with higher NA couple light more efficiently since it collects light at higher angles from the illumination source. Fibers usually have a NA of 0.3 to 0.6 which would accept light in cones of 30 to 60 degrees.

There are two types of illumination fibers, end emitting fibers and side or edge emitting fibers. End emitting fibers use total internal reflection to make almost all the light come out of the end of the fiber. Side emitting fibers are designed to not be as efficient by letting some light to be scattered into the cladding, causing it to become visible, typically for art or display purposes. This will cause the whole fiber to light up while in an end emitting fiber only the end will light up. A lot of light is lost by edge emissions throughout the fiber, causing side emitting fibers to have high attenuation. These losses can limit the length of the fiber that can be used. There are a few solutions to this such as using two illumination sources to illuminate the fiber from both ends, having the fibers come back to the same illumination source, or using reflective end caps to send the light back into the fiber.

Cleaving a fiber is a controlled break in a fiber, this is done to purposefully try to make a flat surface at the end of the fiber perpendicular to the longitude of the fiber. This process is typically done in order to shorten the fiber or to have a cleaner edge in the event of scratching or contamination. In order to properly cleave a fiber, it must be stripped and cleaved using a specialized tool. Stripping entails removing the protective coating off the end of a fiber using a fiber stripper. A quality cleave is needed for low loss splice. A cleaving strategy that is used is known as the scribe and break strategy. This is done by using a cutting tool made of a hard material or diamond to make a crack in the fiber and then add tensile force near the crack. Because of the difference in hardness, it causes a cleaner break. Fiber cleaving is important to know in case we use fiber that is longer than what is needed. We will need to cleave the fiber to reduce the travel distance of the light because this will reduce the amount of loss we will experience.

In regard to this project, we are using optical fibers since they would require less optical alignment rather than putting a light source under each chess board square as well as having less light loss due to free space propagation. In addition, since they are so flexible and able to fit in tight spaces, this allows us to save on space which is needed to make the chess board as portable as possible. These optical fibers will all have to be aligned at an angle that will allow light to hit the internal optical parts of each chess piece for it to be reflected to read the light intensity. They will also have to capture enough light from the illumination source to give us sufficient readings for the intensity of the reflected light off the chess piece.

Considering all the information discussed above, we concluded that the best fiber to use would be a multimode end emitting fiber. We chose an end emitting fiber since it would provide the most light output to illuminate the bottom of the chess pieces. If we had chosen side emitting fiber most of the light would have been lost as light traveled along the fiber which would have given us a weaker light output. Most illumination fibers are multimode fibers which is great since multimode fiber have bigger core diameters than single mode fibers which enables the multimode fiber to receive more light into the fiber more easily. A fiber with higher NA is also desirable since it would have a higher light acceptance angle, again making it easier to receive light. A higher NA would also help with illumination of the chess pieces since the light emitted from the fiber would have a wider cone of illumination. Using plastic fibers would be useful to incorporate since they are more flexible and cheaper than glass fibers. The initial purchase of a longer fiber would be beneficial because they would give us more fiber to use in case of emergency and gives us more leeway in cleaving and angle adjustment. However, this comes with the caveat of longer fibers in our chess board increasing light losses due to dispersion effects caused by the material of the fiber. This would decrease our power output as well as introduce more noise into what's transmitted. We would also like for the fiber to be as cheap as we can get it without sacrificing performance.

3.3.1.3 Part Selection: Fiber

TABLE 7: FIBER MARKET PARTS

Brand	AZIMOM	CHINLY	AKEPO	AZIMOM
Quantity	1	1	100	1
Material	Plastic	Plastic	Plastic	Plastic
Length	100 M (328 ft)	100 M (328 ft)	2 M (6.5 ft)	50 M (164 ft)
NA	Not listed	Not listed	0.5	Not listed
Core Diameter	0.25 mm (0.01 in)	0.75 mm (0.03 in)	0.75 mm (0.03 in)	1.5 mm (0.06 in)
Fiber type	Multimode	Multimode	Multimode	Multimode
Temperature Range (°F)	-58°F - 167°F	-58°F - 167°F	-58°F - 167°F	-58°F - 167°F
Cost	$11.88	$8.44	$9.99	$14.20

Taking all of this into consideration, we have decided to use the AKEPO fiber. The AKEPO fiber isn't the cheapest or has the biggest Core Diameter but has the biggest total length out of all the fibers listed by doubling the AZIMOM 100 M fiber and CHINLY fiber lengths. The AKEPO fiber is the second cheapest fiber only beat out by the CHINLY fiber which has the same core diameter but is half the total length. The AZIMOM 50 M fiber has the biggest core diameter meaning it would be the easiest to get light into from the illumination source, but it is the most expensive and has the smallest total distance. Another advantage of the AKEPO fiber is that it comes as a bundle of 100 fibers that are 2 meters long which would reduce the amount of cleaving that we would have to do. This is because we would not need all 100 fibers and we would use way less than the 2 meters.

3.3.2 Illumination System

The purpose of the illumination system is to generate light so that it can be coupled into the fibers so that they can illuminate the bottom of the pieces for later reflection readings. In order to accomplish this, we will need a light source that will generate enough optical power to make it easier to distinguish between each piece type. Light will also have to be distributed evenly to all the fibers so that each square will reach the same amount of light from the fiber.

3.3.2.1 Illumination Sources

Illumination sources are the devices that provide light for the fibers to carry. We will need an illumination source since fibers do not create or provide their own light and only transport it. There are many illumination sources that can provide light for fibers such as lamps, LEDs, lasers, laser diodes, quartz halogen lamps, xenon metal halide lamps, and other high and low powered sources.

Each light source has its advantages and disadvantages, but each can be used for purposes that suit them the best. For example, Xenon metal halide lamps have high power output, produce high quality light, and 24 percent of the energy get output as light which is more efficient than incandescent lightbulbs which have an efficiency of 2 to 4 percent but have short lifespans compared to other light sources, require a few minutes to warm up to reach full brightness, and require a high startup voltage to ionize the gas and produce a significant amount of heat. Lamps might need reflectors if they don’t already have them built in as well as lenses to focus the light into the fiber. Lamps can also be used in tandem with optical filters to allow light of specific wavelengths into the fiber.

Quartz halogen light bulbs are incandescent lights which are known to heat materials up so much that they start to produce light, but this causes most of the energy to be lost as heat and causing incandescent lights to be very inefficient. Quartz halogen light bulbs are small and cheap, perfect for compact optical systems. However, quartz halogen light produces high temperatures, consumes lots of energy, anywhere from 55 to 100 watts and is one of the least efficient light sources compared to the many other light sources. In addition, these temperatures, if left unattended for too long, can melt our plastic housing and fibers and ruin the entire system. It would damage the optical setup as well as allow ambient light into the fibers.

Another option we considered was the Helium-Neon (He-Ne) laser. These devices have a stable wavelength, give good gaussian beam profiles, a high-quality beam, and have incredibly simple alignment. Despite this, Helium-Neon lasers have short lifetimes which range from 1000 to 10000 hours, low output power of 35 to 50 mw for the most powerful He-Ne laser, some He-Ne lasers take a while to warm up to allow population inversion, and the big drawback is that they are bulky and tend to be over 1 foot long, which is above the parameters that were decided for this system.

Another light emitting device that can be used is the Laser diode. Laser Diodes have a lot of advantages such as being smaller than most other lasers, they're one of the cheapest devices to produce laser output in the right wavelengths, they have high efficiency, and can be manufactured into arrays easily due to their small size and pinout capabilities. Laser Diodes do come with their drawbacks such as producing smaller optical powers that range from 5 mw to 10 mw which, compared to other lasers, have a heating problem, produce more divergent laser beams, and require enough current to past the threshold current to enter their lasing mode. Lasers and laser diodes tend to have higher optical power concentrated into a small area and produce a beam, but the light isn't dispersed, not enabling them to illuminate multiple objects at a time.

An additional illumination system is the light emitting diode or LED. LEDs have high energy efficiency since they convert about 80 percent of the energy into light and about 20 percent into heat which also mean they produce less heat compared to many other illumination sources but have limited power unlike other light sources. Newer LEDs have been developed to be more efficient and have higher intensity outputs. They have no warmup period and light up almost instantly and have long life spans. LEDs are also directional and don't light up in all directions which is good since light isn't wasted going to places that don't have to be illuminated as well as limiting what can be illuminated. LEDs can be used for multimode fibers since they have larger core diameters. Single mode fibers, however, use lasers or laser diodes due to their smaller core diameter.

Choosing the right wavelength to emit from the illumination source is another important consideration. Light has a wide spectrum of wavelengths and the light which we can see is called visible light. Visible light has many colors which have corresponding wavelength values. For instance, blue has a wavelength range of about 450 to 495 nanometers while red has a wavelength range of about 620 to 750 nm. Wavelength is an important consideration because we will have under each square a fiber, LED, and photodiode all close together. The light from the illumination source cannot be the same wavelength as the light from the LEDs since it could mess with how we will measure the intensity of the light due to interference effects. The light from the LEDs would give false values that would jeopardize our design. To counteract this, we will put a color filter over the illumination source to block out all the other wavelengths except the wavelength of the selected filter. If no filters are used, it would mean that white light could not be used for the illumination system or the LEDs under the square. This is due to white being a combination of multiple wavelengths of different colors. What could be done is to use a single wavelength for the illumination source and not have the LEDs under the squares emit that wavelength or vice versa. For example, the

LEDs under each square will show possible squares to move for piece or if someone is in check. Each of these will have a specific color so if we choose green to show potential moves of pieces, our illumination source cannot emit any green wavelengths since it will mess with our readings and blue or red wavelengths will have to be chosen.

3.3.2.2 Technology Selection: Light Source

The best illumination source would have the most optical power while still being as cheap as possible. This is very important because it would generate a lot of light making finding the intensities of the reflected light to be easier. There are many expensive sources that are bright enough, but those don't fit into the budget constraints of this project. Another requirement is that the illumination source has to be small enough and weigh as little as possible to fit inside the bottom of our chess board or not make the dimensions of the bottom of the chess board to be too big which would risk the portability of the board in addition to making the board too heavy to move around. Heat from the illumination system must be kept to a minimum since we are using plastic fibers and having lots of heat can ruin them. Ideally, we would want to have no warmup time since it would be inconvenient for the player to wait until the illumination source is at full to play. The illumination source also shouldn't consume a lot of power since it would cut down on the battery life of the whole chess board system. A light source with a long lifetime would also be ideal since it wouldn't have to be replaced as often.

TABLE 8: LIGHT SOURCE TECHNOLOGIES MARKET

	LED	HeNe Laser	Laser diode	Quartz halogen lamps	Xenon metal halide lamp
Temperature (Heat)	Low	Medium	Medium	High	High
Warmup time	None	10 min	None	None	15 min
Size	0.98 x 0.2 x 0.2 in	1.74 x 1.74 x 10.70 in	0.23 x 0.55 x 3.54 in	3.07 x 0.32 x 3.07 in	2.1 x 5.3 x 2.1 in
Quantity in package	100	1	30	5	1
Minimum quantity needed	1	64	64	1	1
Operating current	<20 mA	6.5 mA	<20 mA	0.83 A	1.25 A
Weight	1.27 oz	0.92 lbs	1.55 oz	1.48 oz	2.08 oz
Output power	2 mW	2 mW	2 mW	100 W	150 W

Cost	$ 6.75	$ 1732.50	$ 12.99	$ 8.99	$ 20.71

After gathering all this information, the illumination source we decided on is the LED. Since we are not using single mode fibers, any type of laser with a small beam size didn't make sense. Also, we would have needed a laser for each fiber which would have been costly and add bulk and weight to the system. LEDs disperse the light more than lasers however they are still directional, and light won't be wasted as all the light can be directed towards the fibers. LEDs are the cheapest among the sources listed above with the Quartz halogen lamps being the next closest. LEDs are the smallest weight with again the Quartz halogen lamps being the next closest. LEDs are also the smallest in size easily allowing them to fit under the chess board. The LEDS, laser diode, and Quartz halogen lamps have no warmup time which is great since no time will be wasted waiting for them to reach full power. They also generated the lowest amount of heat which would not affect our plastic fiber or housing. Laser diodes and LEDs have the second smallest operating current, only behind the HeNe laser although if we use less than the 20 mA and for example used 10 mA the LEDs will have a longer lifetime than if they used 20 mA. The drawback from this is that this would cause the LEDs to produce less light which wouldn't be the best. However, if more light is needed, we could add more LEDs since they are so small and energy efficient. The small operating current also means that they would consume less power allowing for longer battery life, which is ideal for the longevity of the system.

3.3.2.3 Part selection: LED illumination source

Now that we have chosen LEDs as our illumination source, we will have to choose between which LED is best to use. Since our illumination source is required to be only one color, we only need LEDs that emit at a narrow wavelength range. LEDs of one color should be cheaper than that of multiple color due to semiconductor properties. However, if the price is close enough, we could use multicolor LEDs and just use filters to keep it centered on one wavelength. We would want LEDs without frosting on the outside because the light wouldn't be as dispersed and would be concentrated in one direction. This would be useful to maximize the amount of incident light coupling into the fiber. We would again want the cheapest LEDs while still being as bright as possible. Being as bright as possible is very important as it's the main reason for these LEDs because this will allow for a lot of light to hit the chess piece and be reflected to determine the piece type. Being bright as possible might contradict trying to have longer battery life since the brighter LEDs shine the more energy they would consume. We could use multiple LEDs that are dimmer to conserve more energy or spend more on more energy efficient LEDs to combat this issue.

TABLE 9: LED ILLUMINATION SOURCE MARKET PARTS

Brand	CO-RODE	Chanzon	ALLECIN

Size	4 x 3 x 0.6 in	6.3 x 5.51 x 0.71 in	7.48 x 5.67 x 0.94 in
Weight	0.79 oz	0.634 oz	4.16 oz
Max Luminous Intensity	15,000 – 20000 mcd	15,000 - 18,000 mcd	15,000 - 20,000 mcd
Quantity	100	100	240
Cost	$6.59	$5.99	$9.99

Based on the chart above, the best LED to choose would be the CO-RODE LED. This is because we will get the highest brightness while also choosing the cheapest LED. The ALLECIN also has a max luminous intensity of 15,000 - 20,000 mcd but has only 10 LEDs that can produce this intensity since it's evenly divided between 24 different values. This would mean we would pay more for less LEDs that would work for our project. Even though the Chanzon LEDs are the cheapest they produce the least amount of luminous intensity and ranges from 15000 to 18000 mcd. The color that produces the highest intensity, besides white, for all of the listed LEDs, was green. All of the coatings on these diodes were unfiltered to transmit the maximum amount of light intensity. All 100 of the CO-RODE LEDs are each just green and are not multicolored. The viewing angle was 30 degrees meaning that the light is directional within our desired parameters. The forward voltage is 3 to 3.2 volts which are low voltages and would mean low power consumption. The wavelength for the green light emitted is from 515 to 525 nm.

3.4 LED Array

A significant component of this project is the use of an LED matrix to convey information to the player. By using different colors, we want to convey valid moves and invalid moves by using a specific color. To confirm these good moves versus the bad ones, we would ideally need two colors. We do have stretch goals that make further use of other colors, so to have the flexibility to incorporate that in without much difficulty would be most ideal.

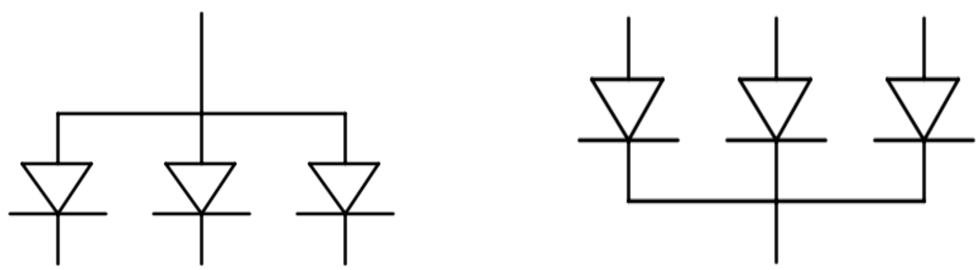
Since a chess board is essentially made up of a grid eight squares wide and eight squares long, an LED has to be under each individual square. The total number of LEDs needed to place one under each square of the chess board will be 64 individual LEDs total. The challenge here would be to find the best way to connect these LEDs and control them to relay information to the players as they play through their game.

3.4.1 Selecting LEDs

Knowing that we wanted to at least communicate valid and invalid moves to the players, at least two colors were needed. We do have stretch goals to implement other colors into our project to serve other purposes, so we wanted to be able to include that option without too much added difficulty. RGB LEDs have built in diodes for red, green, and blue so each can be activated individually to achieve that specific color, but we can turn on combinations of them to achieve other colors. With all the possible combinations, we would have access to seven different colors total.

When researching RGB LEDs, there were decisions that had to be made about the LEDs themselves. The first one was deciding the casing the LEDs would come in. There were only two options: frosted or clear. With the clear casing, the light that would be emitted would shoot straight out from the diodes, so it resembled a laser. Knowing we wanted our squares to be evenly lit as possible, the clear casing was not the best option to accomplish this. With the frosted casing, the light would be diffused by the casing so it would give more of a glow effect. We can use this glow to our advantage when lighting up our board. Another significant detail about the two casings is their ability to mix colors. With the clear casing, when more than one diode is turned on, the colors would not be able to blend. With the frosted casing, the light from multiple diodes would have a chance to mix with the diffusion so we would have a more notable difference with the colors.

Another decision that had to be made about the LEDs were the configuration of the diodes themselves. There are also two ways the diodes could be configured: common anode or common cathode. As the names suggests, the portion that is common would all share a common node. After looking into different LED array configurations, we found it to be easier to implement common anode RGB LEDs. To turn on common anode LEDs, the anode would have to be connected to a voltage source while the cathode of the desired color would be connected to a lower voltage source so current would be pulled through the diode.



(A) (B)

FIGURE 4: (A) COMMON ANODE CONFIGURATION AND (B) COMMON CATHODE CONFIGURATION

The table below shows the specifications of the LEDs that were selected.

TABLE 10: LED CHIP TYPICAL ELECTRIC AND OPTICAL CHARACTERISTICS:(T_A=25°C)

Items	Color	Symbol	Condition	Min.	Typ.	Max.	Unit
	Red	V_F	I_F=20mA	1.8	2.0	2.2	V

Forward Voltage	Green			3.0	3.2	3.4	
	Blue			3.0	3.2	3.4	
Luminous Intensity	Red	I_V	I_F=20mA	---	---	800	mcd
	Green			---	---	4000	
	Blue			---	---	900	
Wavelength	Red	$\Delta\lambda$	I_F=20mA	620	623	625	nm
	Green			515	517.5	520	
	Blue			465	466	467.5	
Light Degradation after 1000 hours	Red	-4.68% ~ -8.27%					
	Green	-11.37%~ -15.30%					
	Blue	-8.23%~ -16.81%					

3.4.2 LED Controller

One of the most difficult parts of the LED is finding a way to turn on and off desired LEDs. In order to simplify the design of the matrix, the LEDs will be connected by row and column in a grid system. Each column would correspond to the cathodes of each RGB LED so there will be 24 columns total. Each row would correspond to the anodes of the LEDs in that row. Below shows a 2 x 3 matrix showing these connections.

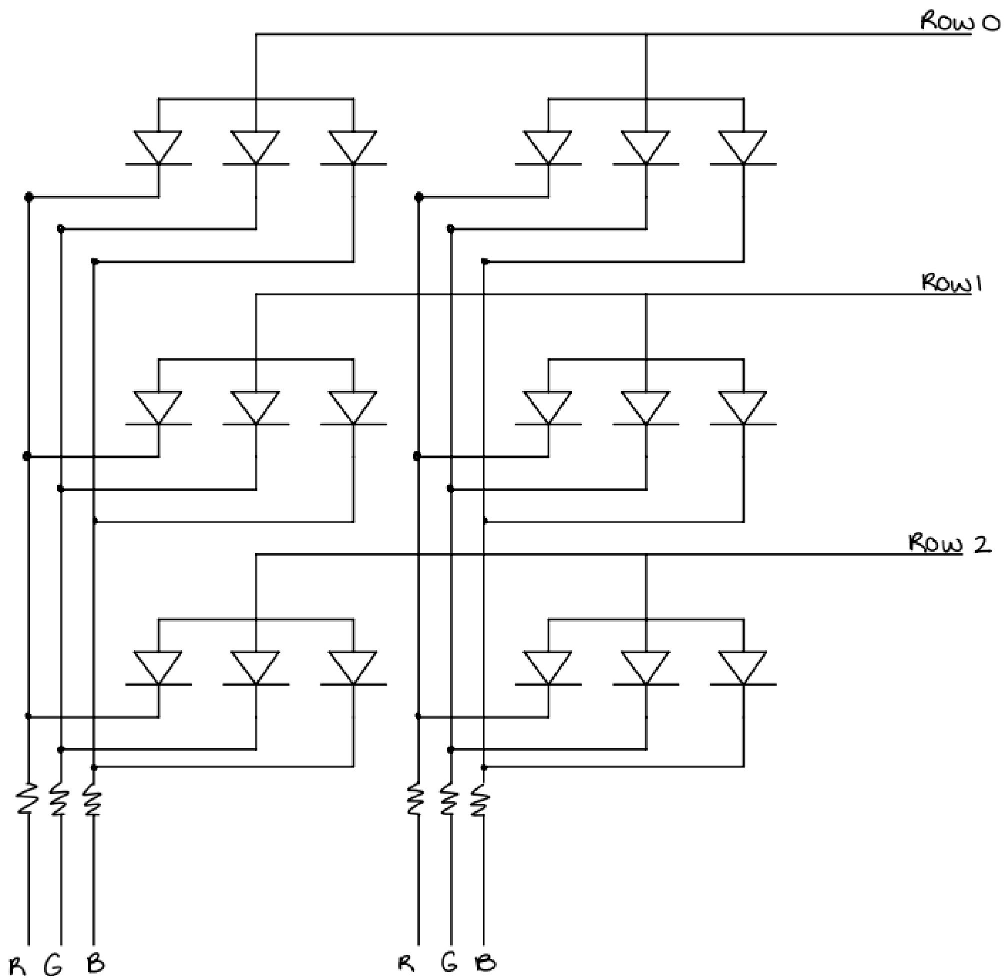


FIGURE 5: 2 X 3 LED MATRIX

Knowing the method to turn on the diodes, we can then make assumptions about the connections that have to be made to this matrix. Each row must be connected to a voltage source while each column must be connected to a ground. With these connections, current can then flow through the diode and it will turn on. There is an issue with this though, all the diodes would turn on without any way to control these rows and columns. To fix this issue, we must implement a way to turn on and off voltages to the desired rows and columns. For our rows, we can use PNP transistors to switch on the power to these rows.

When researching LED matrixes, we found out there are multiple ways to control these arrays. Two of them in particular stood out: LED drivers and shift registers.

A significant component that went into our research on selecting these controllers is the packaging of the component itself. We wanted to be able to test the integrated circuit in a breadboard before finalizing our design, so we had to look for specific package types that allowed us to do just that. There were three different package types that we could choose from: through-hole, surface mount, and chip carrier. In order to test a chip carrier package, it would have to mounted to a PCB that could connect to a bread board for testing. It would take more time to order a PCB to select the specific component and for it to not work in the end. Surface mounts have a similar issue as well. The leads on surface mount components are designed so that they lay directly on top of a PCB so through holes are not needed to solder the component. These special leads are not designed to be inserted into a breadboard for testing. Through-hole components are the best option because of the leads on the component itself. The leads are long enough and spaced enough for it to be placed into a bread board for testing. This way, we can ensure our driver is compatible with the LED matrix before soldering takes place.

3.4.2.1 LED Driver

LED drivers are designed so that constant current and voltage would be supplied to the LEDs so that none of them will overheat, flicker, change color unexpectedly, or not perform as expected. To accomplish this, LED drivers convert alternating current to direct current and some drivers even have the ability to shut off all the LEDs if the temperature reaches above a certain point.

When looking into LED drivers that could control our matrix, we wanted to have enough channels to control at least control the columns which is 24. We found two drivers that could meet this requirement: the LP5024 and the TLC5947. We also wanted to make sure that each channel could handle the maximum current output of each diode which is 20 mA.

Another specification that needed to be met is ensuring that the driver could pull enough current so that the LEDs could turn on. The LEDs we selected draw about 20 mA so we need to make sure that the driver could handle this demand. All three LED drivers that we found meet this requirement.

With the LP5024, it makes use of the current sink feature which would work best for the common anode LEDs that we plan on using. The one downside of the LP5024 is that it is designed to control diodes in parallel. The LP5024 assumes that the anode is already connected to a power source while it connects to the cathodes of the LEDs. By finding a way to control which row receives power, we can trick the LP5024 into thinking it is only controlling one row at a time. This is where using transistors to direct power to specific rows gives us an advantage.

There is one downfall to the LP5024 and other packages in its family: it only is available in surface mount and chip carrier packaging. This would not allow us to directly test it with our LEDs and microcontroller without designing a PCB that would allow us to integrate them all together.

We selected the TLC59210 out of the three LED drivers that we researched due to its ability to control the columns of our matrix and the ability to test it with our other selected components. The TLC59210 can take a clock input speed up to 1 MHz.

Table 11: LED Driver Comparison

	LP5024	TLC5947	TLC59210
Number of channels	24	24	8
LED current per channel (mA)	35	30	200
Vin min-max (V)	2.7-5.5	3-5.5	3.3-5.5
Extra Features	Current sink Constant current Current control I2C control Power-save mode	PWM control Thermal shutdown	Current sink Thermal shutdown PWM control
Price	$1.186	$1.742	$0.697

3.4.2.2 Shift Registers

Shift registers are a type of sequential logic circuit that can be used to store or transform binary data. Each shift register is made up of multiple flip flops designed to hold multiple bits of data. The way a bit of data is sent to the corresponding register is by making use of clock pulses to push each bit individually until it arrives at that register.

One of the features of these shift registers is the ability to daisy chain them together to create a shift register that can control more bits. Daisy chaining is where the output of one shift register is directly fed into the input of the next one. So if two shift registers are daisy chained together and are sent 16 bits of data, the first shift register will take the first 8 bits and process them and send the rest to the other shift register. With the ability to daisy chain, if we cannot find a decent shift register large enough for our project, we can make one that meets the specifications we need. To control the entire array, 24 pins would be needed to control the columns and 8 pins would be needed to control the rows of the array. In total, we would need at least 32 pins to control our LED array.

When selecting shift registers, a key feature is for it to draw enough current for the diodes. Each diode for the LEDs needs about 20 mA to be turned on. Most shift registers out there cannot supply enough current to completely turn on a row of LEDs so we can use these shift registers to turn on switches to power the diodes. There is also another issue with how the shift registers would power the rows and columns. Due to this current power limitation, we can only turn on one LED at a time. When we need to be able to light up multiple squares, this would be a problem. There is a trick to resolve this problem. The human eye can only see about 30 to 60 frames per second. We need to find a way to quickly alternate turning on single LEDs so that it appears to our eyes that all of them are on at once. So, we would need to use a shift register that is fast when processing data.

When researching shift registers, one of the major specifications needed for our project is a serial-in to parallel-out (SIPO) configuration. This configuration takes in a single line of input, let us say 16 bits of data for now, and transforms it so the designated bit goes to the corresponding register. With this configuration, we are able to control all of the rows and columns at once. Ideally, we want a shift register to control both our rows and columns, so 32 bits total would be needed. When looking into 16-bit shift registers we found one that is in the SIPO configuration. The one issue is that a single unit would be $31.592 so for two of them it would be about $63. Compared to the price of getting enough 8-bit shift registers to control 32 bits of information, the price for 4 8-bit shift registers is significantly lower.

With the 8-bit shift registers that we found, they also varied in their specs. We selected two to compare against: the SN74HC595 and the TPIC6A595. The TPIC6A595 is capable of drawing more current, and this value actually goes higher with less outputs on. The SN74HC595 meets the current requirement for our LEDs because it can draw a maximum of 35 mA for a singular output. One of the deciding factors here though is the speed of the clock. We need a high clock speed when controlling the matrix and the SN74HC595 has the best clock speed for its price.

TABLE 12: SHIFT REGISTERS COMPARISON

	SN54LS673	SN74HC595	TPIC6A595
Number of Channels	16	8	8
Max Current (mA)	12	35	350
Vin min-max (V)	4.75-5.25	2-6	4.5-5.5
Clock Speed	20 MHz	24 MHz	10 MHz
Price	$31.592	$0.073	$1.150

3.4.2.3 LED Driver vs. Shift Registers

One of the final decisions that needs to be made is whether to go with the LED driver or the shift registers when controlling our LED matrix. Ideally, we the driver to be quick enough to light up the correct squares for our players almost immediately after a piece is identified. This restriction limit is what will help us decide between these two options. We will not be able to have a final decision until we get time to test all of our components together.

3.5 Display Screen

There are a lot of factors to consider as we start looking for displays to use for our project to select the best option. In addition to showing game time and providing useful information to the players, the display will also act as a general graphical user interface for navigating menus and configuring the game.

3.5.1 Display Requirements

This section will go over the display requirements and characteristics we put into consideration when choosing a display to implement in our design and put forth some available products for comparison.

3.5.1.1 Size and Resolution

The size of our display is high on our list of priorities. Naturally, given the limitations on the device's bulk and dimensions, we must select a size that fits the form factor of our project without sacrificing the display's quality or readability. This also considers the resolution of the display, since displays with higher resolutions offer images that are clearer and more detailed.

3.5.1.2 Refresh Rate and Response Time

The display's response time is still a point of contention, despite being less significant for our design. Since our application won't be displaying any fast-moving images or videos, we don't need exceptionally high refresh rates on our screen. Nevertheless, it's crucial that the display remains responsive to our program's requirements to give the user a positive experience.

3.5.1.3 Power Consumption

The display will probably be one of the most power-intensive parts of the design, so knowing how much power it uses is crucial considering that our project will run on a battery. Most of a display's power consumption goes toward lighting the backlight, so it's critical to select a display with a reasonable power consumption without sacrificing brightness or image clarity.

3.5.1.4 Support

This step serves to ensure that the display is compatible with the microcontroller or processor used in our project. If the display also includes software libraries, development kits, or drivers that make integration with our system easier, we could save a significant amount of development time.

3.5.1.5 Cost

The cost of the display is a significant factor, especially on the budget constraints we have for our project. Careful cost management contributes to the success of the project, making it important to find a display that meets your requirements at a reasonable cost.

3.5.1.6 Touchscreen Capabilities

Finally, one characteristic that is not essential but is also of note is the possibility of using the display as a form of user input. We have planned to use buttons on the design beside the display for controlling user input, however, if the matching display contained a good touchscreen, it would make for much sleeker design.

3.5.2 Display Choice

Given all the requirements described above, we need to analyze the different types of displays available and decide on which one best fits our needs.

3.5.2.1 Seven-Segment Displays & Dot Matrices

There is little reason to consider these two display types, as they are far below our desired specifications. Seven-segment displays are far too limiting on the type and quantity of information we can display, and dot matrices have incredibly low resolution due to their nature, as they are essentially a group of individual LEDs much akin to what we are trying to replicate on a larger scale with our own LED matrix for the chessboard lighting.

3.5.2.2 OLED & AMOLED Displays

OLED stands for Organic Light-Emitting Diode, these use an organic, carbon-based material for emitting light, meaning they lack the necessity of backlighting. This category of displays allows for superior image quality but consumes more power and usually provides worse visibility in bright light. “Hobby-Oriented” OLED and AMOLED displays for microcontroller implementation are also very expensive when compared to other display types and are usually found in smaller-sized packages, which are not exactly ideal for the type or amount of content we plan to display.

3.5.2.3 LCD Displays

Probably one of the most common displays used for general applications, the traditional liquid crystal displays are simple, demand low power, and are considerably cheaper than the OLED displays discussed above. LCD displays vary wildly in their design depending on its intended use, however, displays used for microcontroller integration tend to have limitations, such as only being able to display a small number of characters at a time, and being unable to display full images. For those reasons, LCD will be unfit for this design.

3.5.2.4 TFT LCD Displays

TFT LCDs use a thin film of transistors rather than a single layer of transistors like traditional LCDs. This results in improved image quality, faster response time, and lower power consumption. TFT LCDs are also thinner and lighter than conventional LCDs, making them ideal for use in mobile devices, like our chessboard.

TFT displays are capable of reproducing images and are often cheaper and have a larger screen size over the other contenders. Not only that, but some manufacturers even include touchscreen capability, which, as discussed in the requirements section above, although not something that is obligatory, it is appreciated and would expand the possibilities of our design and overall contribute to the product's presentation.

All these qualities make the TFT displays a great choice for our project.

3.5.3 LCD Market Research

TABLE 13: LCD PART COMPARISON

	HiLetgo 2.8" SPI TFT LCD Display	HiLetgo 3.5" TFT LCD Display	Hosyond 4.0" TFT LCD Display
Screen Size & Resolution	2.8" 240 x 320 pixels	3.5" 480 x 320 pixels	4" 480 x 320 pixels
Operational Voltage	3.3V~5V	3.3V~5V	3.3V~5V
Expected Current Consumption	120mAh	150mAh	180mAh
Driver IC	ILI9341	ILI9488	ILI9486
Communication Protocols	SPI	SPI Capability (Not Implemented)	SPI
Touchscreen	Yes (Resistive)	No	Yes (Resistive)
Cost	$16.39	$18.49	$19.99

All three displays seen on the table were all relatively inexpensive in both their initial hardware and power consumption costs, which are estimated at only a 1Wh power draw for the largest display, less than the expected consumption of the LED grid we will be using to light up the chessboard. For their drivers, all three displays use relatively well-known IC chips with decent library support and are capable of SPI communication, except for the 3.5" display, which has the capability to perform SPI but does not have it integrated in the module by default.

The clear choice among our options was the Hosyond 4.0" Display, however, we did find ourselves a bit limited on the size of the display. A large majority of displays with 5" or bigger screens use HDMI channels and are overall harder to introduce into the design without significant upgrades to additional components, which would drastically impact on our budget. To make up for the smaller size, it is possible to build the display in an angle, as opposed to flat on the chessboard to facilitate reading.

3.6 Power Management

When deciding on a type of battery for our design, we are faced with two options that are the most popular in the market, lithium polymer pouch cells (LiPo) and lithium-ion cylindrical cells (Li-ion). Other types do exist, such as nickel metal hydride batteries, which do not meet our energy needs, and lead acid batteries, which are unnecessarily heavy, bulky, and expensive for use in this application.

3.6.1 Lithium-Ion vs Lithium Polymer Batteries

Since we have already determined that the other kinds of batteries do not attend our project's requirements, in this section we will be analyzing the different characteristics of Lithium-Ion and Lithium Polymer batteries to determine which will be the best fit for our design.

3.6.1.1 Physical Properties

Lithium polymer batteries are a winner in this category. Contrary to the typical steel hard casings of their conventional lithium-ion counterparts, LiPo batteries are usually manufactured in a flexible casing, which are much lighter and have a much smaller footprint.

3.6.1.2 Power Characteristics

Even though, as mentioned above, lithium polymer batteries are usually much lighter than lithium-ion cells, when accounting for their energy density (Wh/Kg), lithium-ion batteries actually pull out ahead, able to store larger amounts of energy for less added weight, allowing for longer runtimes in comparison to LiPo cells.

3.6.1.3 Safety

Safety was a major concern over the choice of batteries, both for the safety of the designers and the users. Lithium-ion batteries tend to be safer than lithium polymer pouches due to their hard metal casing exterior, making them more resistant to impact and puncture, which is going to be great project to make it safer during transport and usage, since we are aiming for portability.

3.6.1.4 Cost

Li-ion wins again. Although comparing individual cells show that lithium polymer is cheaper, we must consider that, to achieve the same capacity we would have with li-ion, we would need a larger amount of battery cells, which would impact on our budget.

TABLE 14: BATTERY COMPARISON

	Lead Acid	Ni-MH	Li-Ion	Li-PO
Cycle Life (Cycles)	200 - 2000	500 - 1000	500 – 2000	> 1200
Efficiency (%)	70 - 90	70	75 - 90	70
Energy Density (Wh/Kg)	30 - 40	30 - 80	100 - 250	130 - 200
Weight	Heavy	Medium	Light	Lightest
Total Cost	Low	Medium	High	Medium
Toxicity	Very High	High	Low	Low

3.6.2 Battery Pack Design

Probably the most crucial part in our power design. Due to the need for convenience and portability of the design, it is important that the board doesn't rely on wired connections to an external power supply to maintain operations. The solution to this issue is simple and effective: to implement a battery pack comprised of 18650 lithium-ion battery cells into the design.

There are several benefits to this approach, but in this case, these cells are particularly advantageous due to their high energy density, that is, their ability to store a large amount of energy in a relatively small package. Not only that, but by grouping together individual cells into a custom power bank makes it simple to alter the design in later stages of development.

Once the voltage (Volts) and capacity (mAh) requirements of the system are established, it is possible to assemble a battery pack that meets the specifications by connecting these cells in parallel to reach the desired amperage and afterwards, connect such parallel groups, which we will call modules, in series to achieve an appropriate voltage.

$$\text{(A)}\quad \# \, of \; Parallel \; Cells = \frac{Desired \; Pack \; Capacity}{Cell \; Capacity}$$

$$\text{(B)}\quad \# \, of \; Serialized \; Modules = \frac{Pack \; Nominal \; Voltage}{Cell \; Nominal \; Voltage}$$

FIGURE 6: (A) BATTERY EQUATION FOR NUMBER OF PARALLEL CELLS (B) BATTERY EQUATION FOR NUMBER OF SERIALIZED MODULES

3.6.3 Power Estimations

The LED matrix used on the board is, by a large margin, the most demanding component power-wise. Assuming an average current draw of 20mAh per LED on a voltage output of 5V, we can estimate, based on average range of 12 to 17 LEDs concurrently lit, that the total power consumption of the LED array will range between 1.2 and 1.7 Wh.

The LCD display comes in second place for power consumption, drawing an expected current of 180mAh at 5V, resulting in an expected power rating of 0.9W.

The remaining components are mostly used in low-power applications, so it's expected that they contribute little to the overall operational demand. However, to account for these other components and give the project some leeway in the upcoming phase of prototyping and development, we will consider the total power consumption of the project to range between 3.5 and 4.5 Wh, for the purposes of designing an initial battery prototype.

To have enough energy on our battery to supply to the project, we need to set expectations on the length of time we want the battery to last. Assuming a minimum duration time of 6 hours, we can multiply that number by the hourly power draw of the circuit to determine that our battery will need a total of 27 Watts. Easily achievable with our battery design.

The initial plan for our battery will be a 2S3P design, meaning it will hold two groups of three 18650 cells each. Considering each cell has a nominal voltage of 3.7 V and a capacity of 3250 mAh, this design will make a 6-cell battery pack with a nominal voltage of 7.4 V and a total capacity of 9750 mAh, which totals for a 72.15 Watts battery, which should easily be able to maintain the demand of the system for much longer than our 6-hour minimum.

3.6.4 Battery Management System (BMS)

A Battery Management System (BMS) serves to ensure safe and efficient operation while running our product on battery power. This system is essential for the type of battery we are using, as it is designed to monitor, control, and safeguard our battery.

In this case, our management system will oversee the balance of charge across all cells, making sure they charge evenly throughout. Without such a system in place, some cells could possibly charge up faster, as not every cell is chemically identical, which would lead misalignment in the voltages of the cells, and ultimately to the destruction of the battery due to overvoltage.

Additionally, other protection mechanisms are included in systems like this one, such as short circuit protection, as well as protection from overcharge, overdischarge, and overcurrent. In essence, a BMS is crucial for this design and will prove itself to be very useful for dealing with any safety concerns regarding our device's power source.

Today, we can find several boards in the market with many of these protection circuits already pre-installed. Displayed in the table below are a few examples of model protection systems made specifically for 2S battery packs like the ones we will be using. All the listed models below possess the same protection circuitry listed above as requirements, however, we decided to go with the JH20 due to its inbuilt charge balancing system, which the others lack.

TABLE 15: BMS MODELS COMPARISON

	HX-2S-A10	HX-2S-JH20	HX-2S-D01
Charging Voltage	8.4V-9V	8.4V-9V	8.4V-9V
Upper Limit Operating Current	8A	10A	8A
Dimensions	41 x 8 x 2.2 (L x W x H in mm)	46.7 x 23 x 3.15 (L x W x H in mm)	40 x 17 x 3.5 (L x W x H in mm)
Charge Balancing	NO	YES	NO
Cost	$9.49 / 5 ($1.898 ea)	$11.99 / 5 ($2.398 ea)	$6.99 / 2 ($3.495 ea)

3.6.5 Voltage Regulators

Voltage regulators are extremely useful devices. They are a type of component in a power supply unit that ensures a constant voltage supply through all operational conditions. It regulates the device's voltage during cases of power fluctuations and load variations. This project will require consistent and stable voltage levels during the device's operation. This will prevent damage to sensitive electrical components due to overvoltage, prevent undervoltage conditions that can lead to improper operation or data loss, as well as increase overall power efficiency and battery life.

Although the general use for voltage regulators is to maintain the output voltage at a specified amount, there are multiple approaches to achieving this. Thus, there exists two types of voltage regulators, called switching and linear regulators respectively.

Linear regulators are an excellent choice for powering low-power devices. However, despite its simplicity, and low cost, linear regulators are typically inefficient, as they dissipate a great amount of power as heat during operation, making them inadequate for use on a battery-powered device. Meanwhile, switching regulators, are highly efficient and are available in the market in compact and reliable modules that will aid us in the implementation process.

TABLE 16: LINEAR AND SWITCHING VOLTAGE REGULATOR COMPARISON

	Linear Regulators	Switching Regulators
Design Availability	Buck	Buck Boost Hybrid (Buck-Boost)
Efficiency	Low	High
Complexity	Low	Medium-High
Total Cost	Low	Medium to high due to cost of external components
Input Voltage Range	Small Range	Wide Range

In our design we will use a buck converter, which is utilized when the DC output voltage required is lower than the DC input voltage. The ones we will be using are a very commonly used type of switching circuits for voltage regulation. We are using these, as opposed to boost converters because our battery will have a larger voltage than the 5V we will use in the system. The basic topology of a buck converter can be seen in the image below, do note that the finished model contains control circuitry to maintain the proper operation of these circuits.

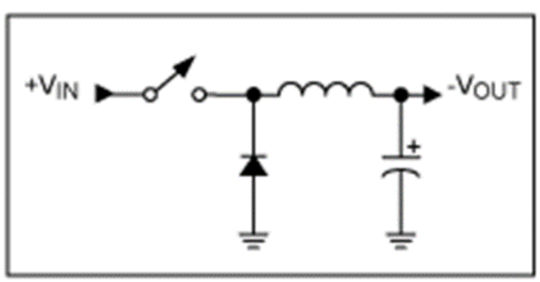


FIGURE 7: BUCK CONVERTER CIRCUIT TOPOLOGY

3.7 Chess Engine

When beginning to research what Microcontroller to select, a few different decisions could be made regarding functionality of the product at the end. A fully implemented chess engine will add features such as best move selection, and an evaluation of the current position, but it may ultimately affect the microcontroller's selection as specifications have to be selected based around this software component, which will be the most CPU intensive. This portion of research was approached with an open mind to see if it was possible to easily implement a chess engine, and not give too many restrictions to the rest of the project if these features are implemented.

3.7.1 Chess Engine ELO

ELO is one term used often to compare the strength of chess engines. ELO is a rating system meant to compare the relative strengths of players in game, and chess often uses this rating to compare the best players. Your ELO rating increases or decreases based upon wins or losses to opponents, with each increase / decrease based on the difference in ELO before the match. Typically, a player with an ELO 100 points above their opponent would win about 64% of their games, while a player with an ELO 200 points above their opponent would win about 75% of their games. Magnus Carlsen achieved the highest ELO ranking ever for classical chess with a rating of 2882 in 2014.

Chess engines also can ELOs associated with them, determined by play with other chess engines or against human players. Ideally for this system, the higher the ELO of the chess engine the better the engine would be at showing the player the best moves. A chess engine with a very low ELO could provide moves that don't help the player and hurt the functionality of it. Of course, with players of a high enough ELO using this board, the ELO of the chess engine would have to be still a few hundred points higher than them to guarantee best move selection, and only select chess engines have that high of an ELO. I chess engine with a lower ELO would still likely be better than a vast majority of its users, but not have as high of an accuracy in best move selection.

3.7.2 Chess Engine Selections

3.7.2.1StockFish

StockFish is the most powerful chess engine to date, winning the "Top Chess Engine Championship" 14-times. It is open source and written in C++, compatible with Windows, Mac, and Linux, which means for this to be placed on a microcontroller, it must be a microcontroller that can have an operating system on it. StockFish exists in 16 versions, where the latest versions generally have greater strength but larger file size and memory usage. Earliest versions of this are under 400 KB, and the most modern versions are about 40 MB. On top of that, I found data sheets with different computer's performance, and it is safe to say that a much less powerful processor, like one that would be in a microcontroller, would struggle to run StockFish at any level. StockFish's latest version's ELO is approximately 3200, and in some cases scoring up to 3400+ in rating, which would be stronger than every human chess player to date.

3.7.2.2 Micro-Max

In opposition to StockFish, Micro-Max is an extremely lightweight chess engine that runs on C, known as the smallest C Chess program in existence. This option would function well on a microcontroller, with only having 1433 characters total. However, the greatest weakness of this chess engine is it's ELO, with recorded ELO of about 1950. This would place it at about the 99.7 percentile out of all USCF (United States Chess Federation) members, meaning that a good majority of all chess players would lose to this engine, but there will still be exceptions. The code provided doesn't initially support being able to load a game at any position other than the starting one, but with some modification that aspect shouldn't be much more difficult to emulate.

3.7.2.3 Tom Kerrigan's Simple Chess Program (TSCP)

Like Micro-Max, TSCP is a lightweight chess engine that runs on C, with a very low ELO of about 1700. It comes in at 2,248 lines of code, but unlike Micro-Max, it is written in a way designed to be very easy to read and teach people how chess engines work. Not only that, but it has an opening book available to emulate an actual player towards the start of the game. Being able to load a position on the board as well as evaluation of any position is also already there and would just have to be slightly modified to be able to use that data ourselves. With its low ELO, it would place at about the 99.3 percentile out of all USCF members.

Through experimentation with this engine itself, its features extend beyond just analysis of a game or a computer opponent, but it can also organize and hold all the information for both sides to play the game. Rather than having to program the entire logic for each move, create systems to store the positions, etc. this program could do that all for us, and easily be modified to work with the rest of the micro controller. This is a huge strength as it will speed up implementation immensely.

One potential issue with TSCP though is that TSCP is copyrighted and needs explicit permission to use the work. However, this issue was quickly resolved by reaching out to the owner of TSCP, Tom Kerrigan, and we got approval to use TSCP if we make it clear that we did not write TSCP. We also cannot be profiting off or redistribute TSCP, but neither of which will happen in the scope of this project.

3.7.2.4 GNU Chess

GNU Chess is a somewhat of an average of the previously found chess engines. It has a very high ELO of 2660, which places it in the 100.00 percentile out of all USCF, and the 99.97 percentile of all non-scholastic USCF. This means that it is extremely unlikely anyone would beat this bot, except for a handful of super grandmasters out of the entire world. This alone will make it extremely accurate for best moves detection and evaluation. In addition to that, the file size isn't too unreasonable with only taking 3.2 MB of space, which shouldn't be too big of an issue on most microcontrollers.

However, it has a few weaknesses, one of which being that it must use a Linux operating system to run. Requiring an operating system will cause many more constraints on the project, as it will become more difficult interacting with hardware and keeping high battery life, so this a major downside to this powerful engine. This engine appears to be one of the best options though for a device with an operating system though, as it is more light-weight than StockFish, and having it run on Linux, the most light-weight operating system, would likely make it the simplest to implement on that type of system.

TABLE 17: CHESS ENGINES

Qualities:	StockFish	Micro-Max	TSCP	GNU Chess
Approximate ELO	3200	1950	1700	2660
File Size	40 MB	6 KB	160 KB	3.2 MB
Requires OS?	YES	NO	NO	YES
*Load Position Feature?	YES	NO	YES	YES

**This quality is true if the given chess engine can load up a chess position rather than only being able to begin from the starting position.*

3.7.3 Chess Engine Conclusions

Based on the research completed on the previous engines, the best choice for a chess engine to be ported on for this project would be Tom Kerrigan's Chess Program. While it has the lowest ELO out of the selections listed, the primary focus of this project is to help beginners learn and play chess, so any ELO as "low as" 1700 is still very high compared to the average beginner. Its readability, and light weight design will make it very easy to implement into this project and become a tool to help implement the many other features outside of just the chess engine aspect of it. Stockfish may have similar features but would be far too demanding on the hardware to be able to run at all. Micro-Max is stronger ELO wise and could fit easily on a microcontroller, but its "minimalist" code style makes it far too difficult to modify for our usage.

I initially began research into this area just to see if this feature was possible to implement, and what restrictions that would impose for the microcontroller selection, but with this engine I see how this would help implement many of the core aspects of this project, and therefore is the best choice and should be implemented regardless of if the analysis features should be added or not. In addition, the specifications of the microcontroller should not be affected much by the selection of this component, if it has storage capable of holding the 160KB file and it can run C code, but the stronger the processing power is the faster and better the engine will run.

3.8 Microcontrollers

3.8.1 Microcontrollers or Single Board Computers?

When beginning to research different microcontrollers to implement for this project, I noticed a large difference in some of the specifications of the devices. Some devices claiming to be at the top end of the industry only had CPU speeds less than a GHz, while I would find other devices from other companies with CPU clock speeds in the multiple GHz. This difference led to the discovery that some of these devices weren't really microcontrollers, but something different known as Single Board Computers (SBC).

SBCs are mainly different from microcontrollers in the sense that they are intended to run Operating Systems, usually Linux. As a result, the specifications for those devices do need to be much higher to be able to support many more libraries of code, and as a result, extended functionality. This would make running more complex code easy, as you can program in a variety of languages that all compile down, with a simpler interface, however this comes with the cost of making access to the hardware components on the device much more difficult interact with. Microcontrollers have very easy access to this without an operating system.

Given the complexity of the hardware stack for this project, that difference alone is enough to make the Microcontroller the clear option for this project. Furthermore, power constraints will also make the microcontroller the winner. Since SBC's have OS running always, the microcontroller can have far more efficient low-power modes, and therefore will allow for longer battery life for our device. Lastly, as researched in the previous section, an operating system can be avoided by selecting a chess engine that runs on C, which was found with TSCP. This project will use a microcontroller.

3.8.2 Microcontroller Specifications

When selecting a microcontroller to use for this project, there are many different specifications to consider that will ultimately cause major issues for the implementation of this project if not selected correctly. Taking into consideration the hardware stack needed, specs such as storage, RAM, CPU speed, GPIO pin count, and other specific functionalities will have certain minimum requirements to make everything work. In addition to that, minimizing cost of this component will also remain a priority, but it must be done so without sacrificing functionality.

Flash storage on the microcontroller is the first benchmark worth considering. This one should not be very restrictive for the project, as the chess engine itself selected from the previous section only will take up a few hundred Kilobytes. With having a display and menus added, there will need to be more data stored, but it is safe to say that everything should be able to fit on the microcontroller with 2MB of flash storage. This is an estimate that will mostly determine on the display graphics and drivers for what needs to be stored. The chess engine will only be a 160 KB file, so it is possible for the program to be stored under 500 KB, but a 2MB max is good realistic upper bound.

RAM size and CPU speed are both important specifications that the stronger they are, the quicker the program can become (and potentially more powerful the chess engine can become). CPU clock rates upon initial research typically range from about 100 up to 600 MHz, and RAM sizes typically range from 200 KB up to 4 MB. The software will likely still be able to run under those minimums provided, but for the sake of minimizing response time, they should be as high as possible.

Low-Power mode should be a feature on this microcontroller to help with battery life on the board. Since there will be many points where the board is just waiting for the user to lift or place a piece, a low-power mode can be implemented to reduce its battery usage. In addition to that feature, the number of GPIO pins is important to have. Depending on the hardware stack, upwards of 30 GPIO pins may needed. Lastly the communication protocols of SPI, I2C, and UART may be used for communicating to these hardware devices, so missing any of these interfaces will restrict what hardware can be selected.

Something worth noting I found after researching these microcontrollers as that there are manufacturers of the processors, and manufacturers of the microcontroller which uses those processors. I found it like buying graphics cards- where a RTX 3070 isn't only sold by Nvidia but buy other manufacturers too who have the hardware components of the Nvidia but with slight modifications. Some of these microcontrollers researched have the same processors in them, but because of changes in the way they are produced on the microcontroller manufacturer specs may differ such, besides CPU clock rate.

3.8.3 Microcontroller Selections

3.8.3.1Teensy 4.0

The Teensy 4.0 microcontroller has high strength with a small size. With an ARM Cortex-M7 CPU, with clock rates of 600MHz and 1MB of RAM, it has some of the best processing power out of any modern microcontroller. It has 2 MB of Flash, which is the bare minimum required for the project, but this may prove to be punishing if my calculated estimation is incorrect. There are 40 digital input/output pins, which should be plenty for this project. For those communication ports, there are 3 I2C and 3 SPI.

One feature of this chip that is amazing is its variety of additional features it hosts, a majority of which aren't initially in the scope of this project but could be expanded for stretch goals. One of which includes its two I2S ports for digital audio. Other features included are Cryptographic Acceleration, Random Number Generator, RTC for date/time, and more. One feature that will be used for this project if this chip is selected is its Pixel Processing Pipeline- this will allow for an accelerated image processing for the display. Depending on what type of display is used, this addition could vastly speed up what and how things are displayed. To top things off, the price and availability of this device is amazing- only $25 and the ability to be delivered in less than two weeks.

Overall, this seems like a solid choice for the project. Its speed will greatly contribute towards our goal of having a short response time and give the chess engine the ability to be very responsive. My primary concerns come with the low flash memory, as this device may run into a situation where the entire program cannot be flashed on, resulting in major changes to the software. In addition, there is no UART communication ports which may make some peripherals unable to be connected at all.

3.8.3.2 ESP32

After finding different microcontrollers, the ESP32 appeared to be one of the most popular microcontrollers currently. It is known for its very cheap price with some strong specs. I manage to find one for $10, with fast delivery of under a week, making it super accessible for starting the project. In addition to that, it has 4 MB of RAM, and 4 MB of Flash, surpassing all the minimum requirements. Its number of GPIO pins isn't too extraordinary at 28 and has communication support for SPI, I2C, and UART. Other additional features this microcontroller has includes Bluetooth connectivity, a 2 MHz – 60 MHz oscillator, Ethernet MAC interface, 5 power modes including an Ultra-Low-Power mode, and a Random Number Generator. Most of those features likely won't be used, but the clock and power mode settings are very helpful for implementing this project efficiently.

Its greatest weakness comes from the CPU speed, at 240MHz. This is not necessarily that slow, but the Teensy 4.0 runs at over double the speed. And this 240MHz is only its *maximum* computing speed, which means realistically it will be running slower than that most of the time. This will be the bottleneck for the system, as the chess engine will directly scale its strength and response time with the CPU speed. However, its many strengths may cause this microcontroller to be the best choice for the project despite its relative slowness.

3.8.3.3 STM32

STM32 does not have nearly as strong specifications as the other options previously found. With a clock rate of 168 MHz, it is about 2/3 slower than the ESP32 and almost 1/4 as slow as the Teensy 4.0. In addition, the RAM comes in at about 200 KB, and the Flash at 512 KB. This is below what I believe should be used for this project, although it is possible it could fit on this small amount of space. Lastly, its price is the highest so far coming in at $28.

However, it does have some strengths. Like the ESP32 board, it has lots of communication with ports for SPI, I2C, and UART. What is greater than that is the fact the microcontroller has about 50 GPIO pins, if not more when configured correctly. This makes it highly adaptable to devices that have many inputs, where the other devices would run out of input space. It is also available very quickly, where it will ship in in less than a week. In addition, it also has the capability to connect to the internet, use an SD card slot, and be easily programmed with the JTAG / SWD debug interface.

The circumstances I could see this project using the microcontroller option is in a dire situation where we need to have a lot of GPIO pins, more than the other options can handle. However, its slow speed and very small amount of RAM and Flash do make it an unlikely contender for the becoming the microcontroller for this project. These specs theoretically could all work for the project, but it will make implementation more difficult, and the final product may have issues with response time.

3.8.3.4 Teensy 4.1

After researching for more similar products to the Teensy 4.0, I found the PJRC Teensy 4.1, which is more expensive, yet more powerful option. Using the same processor as the Teensy 4.0 (an ARM Cortex-M7), it reaches a clock rate of 600 MHz and has installed RAM of 1 MB. However, this version of Teensy improves on its predecessor, with a flash memory of 8 MB (4x greater than version 4.0). This eliminates one of the main concerns with using the Teensy 4.0 as now the flash memory much exceeds my minimum requirements, and it should be comfortably programmed on. In addition to all of this, the are 55 GPIO pins available, which is 15 more than the original version.

One additional special feature this version of Teensy offers is the ability to install additional RAM or Flash Memory. These PSRAM chips can add 8 MB of RAM each, and with two slots available to the Teensy 4.1, you can add up to an additional 16 MB of RAM. The cost of adding these features would total out to be an additional $9.00, where it would be $1.60 per added RAM, but an additional ~$5.00 in shipping. But, if shipping is purchased from the site I provided and not from Amazon, and I purchased the rest of the Teensy 4.1 microcontroller, it will only be about $3.00 more expensive to add these chips as the Amazon option is more expensive.

Other features this microcontroller has are the same as its predecessor, with RTC for date and time, a Pixel Processing Pipeline, and more. These features, particularly the Pixel Processing Pipeline, will help with lowering the response time and allow for the display to be high quality.

Overall, this microcontroller is an amazing contender for this project. It has the highest CPU speed, Flash, GPIOs, and potentially RAM with the additions out of any of the other options listed. A few downsides this microcontroller has though is its price, coming in at $35 or $38 with added RAM, making it the most expensive option. In addition, with manually installing memory to this microcontroller, there is risk of damaging it and needing to replace parts. It also potentially can take the longest out of any other option as well to show up, but that is still within a reasonable range.

3.8.3.5 Arduino Nano

The Arduino Nano has good mix of strengths and weaknesses. This device takes advantage of the ESP32 chip, but the microcontroller itself is manufactured by Arduino. Hence, the CPU speed is at 240 MHz, identical to the other ESP32 option. It has a small amount of RAM with only 512 KB, and very few GPIO pins with only 14. In addition, there are only 4 total communication ports, 1 SPI, 1 I2C, and 2 UART. This may be a major issue when trying to connect to the other peripherals, as communication may be limited.

Its greatest strength however lies in its Flash Memory. Its internal Flash Memory comes in at 8 MB, already equal to the Teensy 4.1, but it also comes with an additional 16 MB of external Flash Memory. This 16 MB external Flash Memory is much slower than the internal one, but with those two combined it can hold 24 MB of Flash Memory. This gives massive flexibility in the software development process, as it is very unlikely that while programming this, we reach the cap of memory. Not only that, but this device is also one of the most affordable, coming in at the cost of $19. Lastly, this device will ship quickly arriving in less than a week.

Overall, this device has some solid strengths, but isn't likely to become to the selected microcontroller for this project based on its low number of GPIO pins and communication ports. With a complex hardware stack, the four communication ports likely won't be enough. It is possible to communicate to multiple devices using one port, but the added latency from doing so will hurt the overall response time of the project.

TABLE 18: MICROCONTROLLERS

Specifications:	Teensy 4.0	ESP32	STM32	Teensy 4.1*	Arduino Nano**
CPU Speed	600 MHz	240 MHz	168 MHz	600 MHz	240 MHz
RAM	1 MB	4 MB	200 KB	1 MB / 17 MB	512 KB
Flash	2 MB	4 MB	512 KB	8 MB	8 + 16 MB
GPIOs	40	28	50	55	14
Communication Protocols	3 SPI, 3 I2C	4 SPI, 2 I2C, 3 UART	3 SPI, 3 I2C, 3 UART	3 SPI, 3 I2C	1 SPI, 1 I2C, 2 UART
Price	$25	$10	$28	$35 / $38	$19
Delivery Time	< 2 Weeks	< 1 Week	< 1 Week	< 1 Week / < 2.5 Weeks	< 1 Week

**Two values are shown for some of these specifications: values on the left are without the added RAM, values on the right are with the added RAM.*

***The two values listed under Flash are for 8 MB of internal flash and 16 MB of external flash. External flash is much slower than internal flash.*

3.8.4 Microco*ntroller Conclusions*

After examining many different microcontrollers, the conclusion was met that the best choice of microcontroller would be the Teensy 4.1. While the Teensy 4.1 is the most expensive option, it remains in the price range of $20 - $40 for this component. Due to a chess engine needing the most computation power available to be the most successful it can be, using the powerful ARM Cortex-M7 chip running at 600 MHz was desired the most. In addition, the ability to reach up to 17 MB of RAM will give this device a high upper bound for performance.

When designing the software, the entire program should realistically fit in under 2 MB, but with Flash Memory of 8 MB, there is plenty of room for adjustment if this estimation was incorrect. The 55 GPIO pins also will permit for very easy and clean hardware integration.

As further hardware components are researched, the communication protocols used needs to be carefully monitored. There is now the added restriction that UART cannot be used to communicate as this chip does not support that communication system. In the circumstance that a hardware component cannot use UART, then a different microcontroller must be selected. In that scenario, the ESP32 would be the best choice.

4.0 Project Standards and Constraints

4.1 Standards about I2C Bus Specification

4.1.1 UM10204 – I2C-Bus Specification and User Manual

In UM10204, the specifications and the uses of the I2C-bus are discussed. I2C is a bus that is both popular and powerful due to how devices communicate on this interface. One device is designated as a master device while all other connected devices are referred to as slaves. This document goes into detail about how the I2C-bus works and specifies the limits that all I2C devices must operate at.

One of the major features is how devices communicate with I2C. Only two bus lines are needed between the master and slave devices: a serial data line (SDA) and a serial clock line (SCL). The master has to always initiate communication with the slave whether it be to send data or to receive it. When sending data, the master sends a start message to the slave, sends the data, and then terminates the connection used to communicate. More steps are necessary when receiving data though. The master still sends a start message, then sends what register of the slave it wants data from, the slave transfers the data, and the master will then terminate the communication line. When there is more than one slave device connected to these two bus lines, the master will then use a unique address to refer to each slave device. The ability to easily add and remove slave devices is one of the major benefits of using I2C because no drastic changes have to be made to the circuit housing these devices.

4.2 Standard About Soldering Standards

Both J-STD-001 and IPC-A-610 are standards that involve the soldering process and includes industry standards for PCB assembly. IPC-A-610 is the overall standard used for soldering because it covers the electronic assembly acceptance while the J-STD-001 is more specific because it covers the materials and processes used for soldering. The LED array in this project require each individual LED to be soldered into the PCB that will contain the wiring for the matrix. The photodiodes that were selected will also need to be mounted to the PCB by soldering as well. These standards will help ensure proper care is taken when soldering these components to the PCB.

4.2.1 IPC-A-610 – Acceptability of Electronic Assemblies

IPC-A-610 is the most widely used electronics assembly acceptance standard in electronics history. This standard deals with the visual aspects of PCBs and ensures that all electronic assemblies meet a certain level of quality. Specifically, this standard deals with component placement, soldering, cleaning, and marking. With this approach in electronic assembly, manufacturers can ensure their products meet acceptable levels of functionality and appearance. This standard covers through-hole, surface-mount components, and mixed technologies.

With component placement, there are certain requirements that need to be met. Since a majority of our project will make use of through-hole components, a lot of detail will be going into designing the PCB. Proper spacing, alignment, and orientation of the components must be considered when designing the PCB. Components must be spaced and aligned properly so that interference between two components does not occur. Proper spacing also allows for an easier time when assembling and testing components on the board. Orientation refers to how polarized components such as diodes and capacitors will be placed onto the PCB so that it follows the design. If the orientation is not correct, the component would not function as intended and will cause further complications. The components have to mounted properly so that they are both secure and the leads of the component are long enough to have it properly soldered to the board. This standard also goes into depth on the soldering requirements. These requirements cover the formation, the fillet, the shape, the cleanliness, and the strength of the solder joint. By covering these specific components of the solder joint, soldered components will be securely attached. Cleaning is required after soldering is completed so that any residing flux is removed so that it does not cause any interference.

4.2.2 J-STD-001 – Requirements for Soldered Electrical and Electronic Assemblies

This standard sets the requirements for the manufacture of electronic assemblies and is used by electronic assembly manufacturers, suppliers, and users. Specifically, this standard covers the necessary materials, methods, and criteria for creating a high-quality electronic assembly. This standard is important to consider because none of the team members have experience in soldering and having basic information about the skill will be very useful.

The standard materials for soldering are covered in J-STD-001. In order to ensure the correct operation and dependability of electronic assemblies, the general composition and qualities of solders are covered. Cleaning agents are covered so that flux residues are properly removed as well as ensuring that the electronic assembly is not damage in the same process.

4.3 Standards for Rechargeable Lithium-Ion Batteries

4.3.1 IEC 62133 - Safety requirements for portable sealed secondary cells, and for batteries made from them, for use in portable applications.

Manufacturers, governmental authorities, and stakeholders involved in the lithium-ion battery industry commonly acknowledge and use IEC 62133 as the benchmark for battery safety. This international standard, developed by the International Electrotechnical Commission (IEC) and last updated in 2021, outlines specifications and tests for the performance and safety of lithium-ion batteries used in a variety of portable electronic devices, such as tablets, laptops, cell phones, and other gadgets. The standard addresses various aspects of battery safety, including electrical, mechanical, and chemical safety.

Adhering to the standard contributes to guaranteeing the safety and dependability of our lithium-ion batteries. IEC 62133 specifically addresses problems like thermal runaway, short circuiting, overcharging, and overdischarging, all of which have the potential to pose a risk to the safety of our users if left unchecked. The standard also specifies testing protocols to confirm standard compliance and labeling and documentation requirements.

IEC 62133 outlines various testing procedures, including checks for electrical abuse such as overcharging and short-circuit conditions, to guarantee that the device is safe for use in its intended applications. One of these tests includes charging a cell utilizing a charging method applicable to cells and batteries that are subjected to external short circuit, thermal abuse, crush, and forced internal short circuit tests before short-circuiting the cell, connecting its positive and negative terminals with a total external resistance of less than 100mΩ. In such abusive electrical scenarios, the battery needs to show a certain level of resilience to avoid thermal runaway or other dangerous reactions, otherwise a failure in the device's operations could pose a great hazard to the users' safety.

However, one of the most pertinent sections for our needs is the one assembly of lithium-ion cells into lithium-ion batteries. It specifies that if several lithium-ion cells are used to build a single battery, then each battery must have precisely matched capacities and be made by the same manufacturer, chemist, and design of cells. The importance of ensuring that each lithium-ion battery, if there are several, has its own independent control and protection is also mentioned in this section. It also emphasizes how important it is for the battery designer to make sure that the design and assembly follow the guidelines provided by the cell manufacturers regarding the maximum limits for voltage, current, and temperature.

IEC 62133 is essential to maintaining the dependability and safety of lithium-ion batteries, which in turn contributes to the general security of the equipment and systems that use them. Manufacturers must adhere to this standard to satisfy regulatory requirements, secure market acceptance, and foster customer trust.

4.3.2 IEC 62115 – Electric Toy Safety

Outside from IEC 62133, another standard that we may want to look at for our project that can impact on our battery design is IEC 62115, which specifies safety requirements for electric toys, that is, any toy that has at least one function dependent on electricity. Any product designed or intended for use in play by children under 14 years of age are considered toys, whether designed exclusively for that demographic or not.

This standard covers the whole range of electric toys from small button battery operated lights to large ride-on electric toys powered by rechargeable batteries. This results in different requirements and tests according to the type of electric toy. Nonetheless, the aim of this standard is to reduce risks when playing with electric toys, especially those risks that are not evident to users, in the case of its electrical function.

4.4 Software Standards

4.4.1 ISO 9899 – C Programming Language

One of the most important standards to be followed for the implementation of our design is the ISO 9899, which standardizes the C programming language. Throughout this document, there are definitions on the syntax, semantic, input specification, output specification, and limits to C. Without this standard, every single device might have its own variation on the language of C, making reusing code virtually impossible as every device would have its own set of commands that differ from the next- meaning either re-writing all the code, translated for that variant, or finding code that someone developed specifically for that device, both of which would cause major restrictions in development.

Some other parts of what this design standard covers includes how it deals with floating-point numbers. In TS 18661, there are five sections listed out describing Binary and Decimal floating-point arithmetic, interchange and extended types, and supplementary functions and attributes. Floating point arithmetic is often one of the most CPU intensive function, so unifying the way the C language uses these will minimize how intensive it will be. Another important section is in TR 18037, which discusses Embedded C programming, and how the C language will interact with all the variable hardware components. This standard is particularly important because programming on any device will use the same command terminology.

4.5 Standard for Chess Board Layout

A standard game of chess is played on a square board that is arranged in an eight by eight grid, with a total of 64 squares. The chessboard alternates the color of the squares between light and dark colors, as related to the color of the pieces. There will be 32 light squares and 32 dark squares. The rows of a chessboard are always referred to as “Files” and typically labeled by letters in the alphabet, ranging from A to H going from left to right. The columns of a chessboard are always referred to as “Ranks” and typically labeled by numbers, ranging from one to eight going from bottom to top. Chess pieces are split into two separate sets, a lighter color (usually white or beige) and a darker color (usually black or brown). Each player will control a set of pieces that contains 16 total pieces. Each side has 6 different kinds of pieces that includes eight pawns, two knights, two rooks, two bishops, one queen and one king.

The setup of the board is the exact same for both players and never changes. The orientation of the board will always be the same for every game. The board will always be oriented so that there is a light colored square in the bottom right corner and a dark colored square in the bottom left corner. A game of chess can never be played if the board is not positioned like that. The setup of the pieces for the game is the exact same for each player as it will always mirror one another. In the back row, the rooks are placed in the corners, followed by the knights placed in the next inside squares, followed by the bishops next to the knights. In the two middle squares is where the queen and king are placed. The queen is always placed on the left middle square while the king is placed on the right middle square. The queen should always be on a square of the same color (the white queen on a light square and the black queen on a dark square). The king will always be placed on an opposite colored square (the white king on a dark square and the black king on a light square). In the row in front of the back row is where all eight pawns will be placed.

The dimensions of the board will vary from country to country, but there are a few standards that are followed, and enforced for professional chess. The dimensions of each square on the board are the exact same as well as the entire board itself. Square size on a chessboard is always proportional to the size of the king’s base diameter. The base diameter of the king should be roughly 75%-80% of the size of the square. This guideline ensures that there will be proper piece spacing on the board during a game. The official World Chess Federation (FIDE) Championship Chess set has a square size of 2 inches and a height for the king at 3.75 inches. Meanwhile, the United States Chess Federation (USCF) has a standard tournament chess board that has 2.25 inch squares and the king’s height is 3.75 inches. Another guideline related to the size of the king piece relates to the king’s base diameter compared to its height. Both FIDE and the USCF follow the guideline that the king’s base diameter should be between 40%-50% of the king’s height.

Once the board is fully set up with 32 pieces altogether, then the game begins. White will always go first followed by black. The objective of the game is for one player to checkmate the opposing king. Once a king is put in checkmate, the game is over.

4.6 Project Constraints

4.6.1 Design Constraints

Design constraints refer to certain physical restrictions that are present when designing a project. The main goal of this project is to create a smart chess board that is portable which causes some constraints to arise when designing our project.

One of the biggest factors that deal with portability is the material the chess board and the box under it will be made out of. We want to use a material that is sturdy enough to protect the hardware it encases but at the same time not be too heavy for a person to carry. There is also a limit to what materials we can use due to economic constraints. None of the hardware components should be disturbed if something were to happen to this shell. Inside of this shell, we want to be able to house two extra layers that the players cannot see: the matrix that contains both the LEDs and the photodiodes, and a PCB that contains the microcontroller and other hardware components. The dimensions of this box should not be too large where it makes it awkward to carry either. We want to aim for a standard chess board size with only a few inches of added depth.

The top layer of our project has some design constraints as well to make it as appeasing as possible to the players. We do not want to give away what is occurring underneath each square so we need to use a material that can allow the LED to emit light but at the same time prevent players from seeing into the chess board. We also want to embed an LCD screen into the side of the project so that it can provide an explanation to players about their selected pieces. We want an LCD screen that is large enough to provide a few sentences but nothing that will add extra bulk to this project.

Another big component is the layer that will be underneath the board the players will play on top of. This layer will contain the LED array along with the optical system that will detect the different pieces. One of the biggest issues with this is to find a way so that the LED does not cause any disturbance to the photodiode and give false values. We have two possible ways to solve this problem. The first is to seal off the optical system under each square so that it can only detect the light that is reflected off the chess piece. This method would then cause the placement of the LED to be shifted so it would be off-centered. The new placement of the LED could cause uneven lighting to occur under a square, so we have to find a way to make it appear even. The second method we can use to solve this problem is using a specific color of light in the optical system. We can use a filter on the photodiode so that it only picks up a certain color of light and make sure the LED array does not light up to that color. In both of these cases, we still have to ensure that the light from each LED does not bleed into neighboring cells and give false information to the players.

There is also a constraint between the two bottom layers of this project. We plan to use fiber optic cables to direct light to each square. The issue with this is the placement of the LED these cables will run to and how these cables will be connected to each square. We want to place the LED on the bottom layer so that it can be in the center and run the cables up through the second layer to attach to the bottom of the top layer. The issue with this is how to implement this in a way where we can still adjust the project without too much disruption to these cables. We can make small holes in the PCB that will hold the LEDs and photodiodes so that these cables can run through to the top layer. The biggest challenge is making all three layers connect.

4.6.2 Economic Constraints

Due to the nature of this project, there is a major economic constraint on the design. Regardless of the scale of what team is designing a project, there is always a tradeoff between minimizing the cost and maximizing the quality of the final product. If there was no economic constraint, then every component would be the highest quality available, and the quality of the final product would be very high- with the cost being equally high, of course. On the other hand, if there was only an economic constraint with no care for the quality of the final product, the result would be very cheap, yet barely function and run at incredibly undesirable speeds.

Because this team is comprised of five undergraduate university students, the tradeoff between cost and quality will have to favor more towards low costs with some reduction to functionality. Each component will have to be selected with this in mind, but ultimately depend on exactly how much value does each one offer for the additional dollar. There are few different groups of components to consider where to distribute cost to. Each component falls into one of the following groups: Microcontroller (MCU), Display (LED array and Display Screen), Piece Identification (Photosensors, Fiber Optic, and Chess Pieces), Power (Battery and Voltage Regulator), or Case + PCB (PCB, Chess Board, Buttons).

The Microcontroller is one component that has certainly high value for every additional dollar spent. Because one of the main challenges over past similar projects was the response time, having a powerful microcontroller would help solve those issues. However, the constraint exists, and this component can get very expensive when reaching very high specs. Ultimately the constraint of < $40 for this seems reasonable, and there are plenty of powerful microcontrollers that would help address the issue of response time while still being affordable.

The Display components has its tradeoffs. The LED array can be restricted to less colors making it cheaper, which ultimately won’t have too big of an effect on the final product, as two colors per square is the least amount that would be needed. In addition, the display screen is a component that can get very expensive with increased size, but ultimately the design can be built around a small screen used to communicate smaller pieces of information, rather than a lot of information at once that would need a larger display for usability. Generally, these components can still function well with cheaper options selected, meaning that a large economic constraint should be placed on these products.

The Piece Identification System (PID) also needs to have economic constraints but may experience some flexibility like the MCU. One specification for this project is to have a piece identification of > 95%. In general, the cheaper the component, the less reliable it will at detecting the piece. When selecting the fiber optic cable, photosensors, and material for the bottom of the chess pieces, the difference between less than 95% and greater than 95% accuracy of detection will be worth the extra dollars. However, there is still constraint, as with the rest of the project, and at a certain point keeping the costs lower and taking a lower accuracy is appropriate. The difference between a less than 99.9% accuracy and a greater than 99.9% accuracy is less worth the additional dollars than the previous specification.

The power supply system in particular needs strong economic constraint to avoid raising the cost of the entire project significantly. One specification for this project related to this component is the battery life, which should be greater than 4 hours. Beyond that minimum though, there should be strong constraint to avoid higher prices- the battery can get very expensive for greater life. The other components related, such as the voltage regulator, may have less of a tight constraint on it as the battery, as if that component fails then many other components may have to be replaced, significantly raising the cost of the project.

The remaining components with the Case and PCB should be following a strong economic constraint. Apart from existing and providing its basic functionality, increased spending on these components will not improve the achievement of any of the other specifications. It can help with the overall aesthetic design, but for functionality's sake, it is not worth the additional costs. These components should be at a minimum in price.

Overall, the economic constraint exists in every portion of this project. Breaking the budget set for any of the components can lead to troubles implementing, as this project does not have any sponsors and will have to directly come out of the pockets of the students designing it. Most of the components do have some impact upon how successful the final design will be at achieving its specifications, so cost should be appropriately weighed in against the quality of its functionality, but those components without that direct correlation should minimize its cost as much as possible.

4.6.3 Manufacturing Constraints

When creating a product, it's important to consider the many manufacturing constraints so that your product can be as efficient and as cheap to produce as possible. The selection of the material has a very large impact on the manufacturing process and can drive up costs, for example using LEDs to light up the squares and as our illumination source. We will be getting 100 LEDs in one order due to the high availability of this component. To show this further, the 100 LEDs also only cost $6.75, making the cost per unit to be 7 cents each, making them very cheap. LEDs have a very long lifetime meaning that they are very durable and don't require high powers that will wear them out. The availability of material, the cost of each component, and durability of the parts are all considered in the material selection. There were many different types of light sources we could have used, however most of them would have run into one of these problems and not been as efficient.

Additionally, the manufacturing process of how parts are made should be considered. Selecting the right process to make your product and making sure it's capable of creating the right quality and quantity is important to make sure the product lasts a long time and is reliable during that time. For our project we have decided to 3D print the chess pieces because we would need the inside of each piece to be hollow. This is because we will be putting mirrors and filters inside each piece to integrate them into the optical system effectively. This would not affect the quality of the pieces since it would still be one color and would not affect the shape of each piece. 3D printing might take a long time for each type of piece, but it would allow us to customize the sizing if needed and make it hollow inside to fit our purposes. We could try buying plastic pieces to shorten this design process but since we would want them to be hollow it might cost more to customize them as needed. Another method we could have tried is to use wood and to carve the pieces out but that would likely take longer to make than 3D printing and if we made it would likely affect the quality because it would depend on the carving skill of the carver. In addition, wood carving tools are expensive and would be considered an unnecessary expense for this project.

Another manufacturing constraint would be the specifications of your product in terms of its size and the components within it. If we have small specifications that could lead to more precise and costly manufacturing due to tolerances and need for more advanced equipment. If we choose a fiber with the smallest core diameter possible it would be harder to manufacture a core that small due to the drawing process taking longer and using higher heat. This would lead to the cost of the development of this project increasing a lot since we would need a lot of fibers to achieve our goal.

Furthermore, manufacturability, which shows how easy something is to create in a factory setting, is something that needs to be considered. Making designs that would be easy to manufacture by minimizing complex shapes and having hard to reach places would be helpful because they can cause production costs to increase, and the chances of defects would be higher because of the need for precision. An example of this would be our electrical circuit design. By making the electrical setup smaller it would decrease the complexity and the cost of the project but would also make it harder to build. We have tried many different designs to make the electrical setup smaller like putting our components into boxes to have less wiring to reduce cost and reduce material waste.

Moreover, scalability is an important manufacturing constraint because some manufacturing techniques are better for high volume productions while some are better for low volume customized products. In our project we could have made filters for specific wavelengths, but it would have cost more instead of general filters that filter all wavelengths out evenly. LEDS are also very scalable since we could get 100 of them in one order. Scalability as a whole is important because any product has to comply with supply and demand, if our project cannot keep up with these changes in the market it will fail as a whole.

In addition to cost constraints, we have to better understand the actual costs of manufacturing. Things to consider that were factored into our product design are material cost, labor costs, and overhead costs. In our project this would involve our optical and electrical design since making a more complex one would cost more to make because of additional time requirements. Adding multiple layers would increase the difficulty of manufacturing, labor, and money by adding extra steps and more work. This would mean having to spend more money because workers need to be paid, machines need maintenance and repair, and more advanced machines cost more money.

Another manufacturing constraint would be supply chains. We would need to analyze our supply chain to spot potential bottlenecks or risks. Having an alternate source for critical material and components is important. We have many components in our project from different companies with different specifications. While we are choosing what is best for us now, later on we will need to consider alternative sources in case some businesses go bankrupt, products are discontinued, or other circumstances happen without our knowing. We will also need to understand what's happening in the world due to material shortages that could happen or shipping problems that could be caused by terrible world events such as COVID 19. In summary, keeping close tabs on the different factors that would affect us getting the materials we need to make this product a success is important to the product as a whole.

Finally, the transportation and logistics of our manufactured material needs to get back to us to use it. We will also have to consider the cost and delivery time. If our materials are made in another country, they could be shipped to us in a cargo ship, which would be the cheapest option. However, would take a longer time. If we wanted priority, it could be flown to us but will cost more. Another thing to think about is gas prices, as those have been on the rise and have affected ground shipping costs as well as time to ship using this method. Different materials have different lead times for this reason. If there are deadlines we have to meet, such as what's required of the product right now, purchasing will have to be done very ahead of time to make sure we get everything on time and can make the product as needed.

By having these manufacturing constraints in mind during product development, we can reduce the chance of costly development changes and production delays looking forward. Organizing a feasibility study can help make a smoother and cost-effective production process. Having documentation that will outline these problems and how to best handle them will also be very helpful.

4.6.4 Environmental Constraints

Considering environmental constraints when developing a product is becoming very necessary in the world today as environmental sustainability is becoming a bigger worry as climate change continues to take hold. Failing to take this into account can lead to long-term negative environmental impacts and reputational damage within the company that produces the product.

A major environmental constraint would be energy efficiency. Designing our chess board to be energy efficient would reduce energy consumption and avoiding greenhouse gas emissions used to power it would be ideal. An example of energy efficiency in our project would be the use of LEDs on our chess board. LEDs are one of the most efficient light sources since most of the electrical energy is converted into optical energy or light instead of being lost to heat. Incandescent light sources produce light by heating up materials, but this is very inefficient since most of the electrical energy is turned into heat. In fact, less than 5 percent of the energy is turned into visible light while the rest of the energy is loss as heat. When choosing our illumination source, we looked into incandescent light sources, and we quickly realized that it would be too inefficient and would require more energy to use. The lumens efficacy for a 120-volt incandescent bulbs is 16 lumens per watt while an LED is 100 lumens per watt. Incandescent lights are so inefficient that some governments are starting to phase them out to decrease energy consumption. Dealing with energy constraints in product development decreases environmental impacts, increases cost savings, generates a longer lifespan for the product, and increases competitiveness due to energy efficiency is important in the market, as customers are becoming more mindful of environmental needs and considerations. It is important to run simulations and energy inspections to make decisions relating to energy consumption in product developments to ensure that all standards are being met to the best of the company's ability.

4.6.5 Energy Constraints

Energy constraints are important to consider when developing a product because energy usage has economic, environmental, and operational implications. We have already talked about how we are using LEDs which are very energy efficient and minimize energy consumption.

An energy constraint we have is battery life. We are aiming to have a minimum of 4 hours of battery life. To do this we must optimize the energy consumption to extend battery life. To do this we are choosing components that are energy efficient and try to efficiently manage power. For example, in our LED array there is a way to turn on all three red, blue, and green LEDs at the same time but that would require a lot of energy which would lower our battery life. Our solution to this is to turn each LED on and off faster than your eyes can perceive that the LED is off which save more energy than turning all of them on at the same time. This is called flickering and is used in many optical applications.

Another energy constraint is heat generation. The generation of heat can indicate that energy is being wasted. As mentioned before incandescent lights aren't efficient since they lose most of their energy as heat. LEDs produce very light amount of heat since most of the energy is converted into optical light power.

Additionally teaching the user is an energy constraint. The user can be taught to use our chess bord in energy efficient ways and maintenance to keep energy savings as high as possible. We could do this by explaining that holding a piece for a long time would waste energy while keeping the pieces on the board would save energy. Some maintenance that could be done to maximize energy savings is to change LEDs and replace or clean the filters, fibers, and mirrors.

5.0 ChatGPT and AI

5.1 Generative AI

Recent years have witnessed a profound surge in the popularity of AI technology, especially in the field of generative AI. Millions of companies across the globe are utilizing artificial intelligence tools to improve their businesses. According to Reuters, ChatGPT alone has reached over 180 million users as of September 2023.

Originally, traditional AI was designed to complete specific tasks given a predefined set of rules. This type of AI operates in a deterministic manner, following explicit rules and instructions. Contrary to traditional AI, generative AI can create a wide variety of new content by analyzing patterns in existing data and utilizing those to create a unique new output. It can produce complex content that mimics human creativity relatively realistically, making it a valuable tool for many industries such as gaming, entertainment, and product design. In one of their articles, Harvard Business Review states, “AI can not only boost our analytic and decision-making abilities but also heighten creativity.”

Right now, generative AI is being applied and quickly improving in various domains, including natural language processing, image synthesis, text completion, code generation, and more. These advancements have opened new possibilities for the utilization of these tools and will fundamentally change many lines of work.

5.2 Ethical and Bias Concerns

As generative models become more powerful, increasing ethical concerns are being raised due to potential misuse of the technology, as well as raising reasonable fears that AI will ultimately replace human workers throughout the economy. This conflict brings to the surface questions over AI ethics as well as copyright and labor laws.

Many of these concerns stem from the field of creative work, such as that of visual artists. These generative AI models are trained on large datasets, often including copyrighted material scraped from the internet, causing them to directly copy the style and content from the publications of real artists without any compensation or credit. To make matters worse, deep neural networks are often considered "black box" systems because it can be challenging to understand how they arrive at specific decisions or generate certain outputs. The lack of explainability raises concerns over accountability, transparency, and the ability to address errors or biases within the model.

Privacy issues also became a public concern in the field of AI. Its capability to generate synthetic images that closely resemble real people can potentially lead to the unauthorized use of personal likenesses or even identity theft. In the same vein, there are concerns over the spreading of misinformation and the potential for malicious use of AI-generated content for deceptive purposes, which could harm individuals or manipulate public opinion.

These are all valid concerns, and addressing these issues requires a multidisciplinary approach involving technology experts, ethicists, policymakers, and society at large. It is essential to strike a balance to ensure that AI technology contributes positively and serves to augment human work, not replace it.

5.3 AI in Design Process

The multi-step process of designing entails investigating, developing, and perfecting an idea to produce a marketable product. With so many opportunities to improve productivity, creativity, and the design process, artificial intelligence can be a very useful tool in product design.

Every product design starts with a problem or challenge that requires resolution. Since it establishes the tone for the whole design process, identifying this issue and establishing the project's scope and goals are critical steps. With AI, designers now have access to new tools for trend and data analysis, that might assist launch projects much quicker.

Because AI can swiftly analyze large amounts of data, it also provides streamlined techniques for collecting information in research, increasing the efficiency of this process. AI can also be a fantastic source of inspiration by utilizing it to study existing designs and come up with original ones. Artificial Intelligence has revolutionized the creative design process, especially in terms of ideation and problem-solving. Based on human prompts, generative models can create new ideas and innovative solutions for designers. This quickens the ideation process and creates new opportunities for design exploration and innovation.

And lastly, the prototyping stage of a product can be greatly aided by AI. Advanced algorithms enable designers to replicate real-world scenarios, eliminating the need for physical models and facilitating the testing and improvement of product prototypes. This lowers the price of physical prototyping while also quickening the design iteration process. AI-driven simulations offer insightful information that helps designers decide on product performance and design changes. Through the application of AI in user feedback analysis, prototype generation, and design iteration, designers can guarantee that the product fulfills the requirements and anticipations of its intended market.

In summary, AI technologies have become invaluable assets in the design industry, easing jobs and enabling designers to push the boundaries of their imagination. They are used for everything from concept formulation and prototyping to data analysis and design optimization, and designers must adapt the way their work to accommodate the incorporation of AI into the creative design process, as we usher in a new era of efficiency and innovation.

5.4 Research Methods with AI

As generative AI is shown to be very valuable in the design process, its worth going more in depth into how research can be optimized using these contemporary tools. While the information generated by AI isn't always true, when it is interpreted instead as just a reputable opinion on something then the floodgates open with potential uses; It can be used to create recommendations for what to research, used to review what research you have done already, and used to solve any barriers that you may be currently stuck with.

Using AI to create recommendations for what to research is more efficient and plan internet searches. By taking advantage of an AI's ability to generate output with considering the context provided, you can create a list of devices worth researching into that matches that context. For example, in the microcontroller selection research portion of this paper, generative AI was used to help point in the right direction the initial devices to research. When asking it what devices it would recommend using given some of the project specifications, it gave a list of 6 or so devices. From there, to ensure the accuracy of the information found, the rest of research was completed manually by looking up those devices specifically. Rather than reference some random opinionated blog posts with generic lists of microcontrollers, using an AI both removed some of the bias and increased the relevance of those results. Out of those selected by the AI, all of them were great suggestions and considered as candidates for implementation.

Another strategy that could be used when conducting research is to use it as a tool to review what choices you have already selected. By using its ability to derive from context, you can prompt a generative AI with all the research options you have done so far and ask it if you are missing any major areas of research. For example, if I only have done research for a device from one manufacturer, it will be able to cater to my projects needs and recommend me some other manufacturers that can also be considered.

Lastly, using generative AI with research can help providing solutions for barriers you are currently struggling with. By once again taking advantage of the large context provided, it can take in many considerations and do its best to reason out a solution. For instance, let's say that while researching, you can't find any LED arrays that would be compatible with the selected microcontroller. Rather than having to get rid of the microcontroller, using AI to prompt it with the exact issue mentioned might shed insight into some components that would work, or check if the problem does need to come down to changing out the MCU. AI serves as another opinion to help aid in making project changing decisions.

5.5 Programming Assist

With generative AI being trained across billions upon billions of data entries into the internet, some AIs have the ability to create or modify code. Even at the very peak of generative AI, most cases it can't simply just write the entire program for a user, but almost every generative AI can be used to assist the writing of it and be used as a catalyst for faster programming. Using AI as an assistant programmer is a technique that will only grow in use as time passes, until AI is strong enough to do all programming by itself. Until then, this team will be able to develop code much faster than one individual without AI.

When designing a section of code, using AI as programming assistant can go as follows:

- In detail, describe to the AI what a certain section of code should look like.
- Take generated code and modify it to fit the rest of the program.
- Copy and paste errors to the AI and give it the code back that you modified.
- Consider solutions and potential implement them.

This approach to programming allows for every piece of code you write to always be just a modification from a larger piece of code generated for you. Most of the code can be generated instantly for you, but because of limitations in the current strength of the AI, it can't correctly process all the context of the rest of the program, including library restrictions, variable names, etc.

The approach described for debugging is also very helpful in the development process. By simply copy and pasting the errors given, as well as the code that is giving the error, the generative AI will do its best to solve the error on its own- often getting it right. This process can be repeated a few times if not solved immediately, but in some cases the error may not be able to be solved completely from the AI. That's when you ask for suggestions for what to look for or what specifically is likely causing the issue, and it can help the developer look in the right areas first that may be causing the problem. This method will save lots of time programming by having all basic errors and bugs solved by the AI- which is trained on data from millions of issues of a similar nature- and only resorting to using the developer to debug more complicated areas of code. This optimizes what each party is best it- the AI is fast but not necessarily the smartest, while the human developer is the opposite, slow but can theoretically solve any issue.

5.6 Prompt Engineering

While generative AI has proven to be a powerful tool for many industries, this tool is greatly weakened without proper use of it. Since all the input you provide for the generative AI is its prompt, the better prompt you create, the better the output you can receive. The collection of strategies used to maximize the output from generative AI is known as prompt engineering. By manipulating the only communication between the user and the machine, result will likely be far more relevant and accurate then what a generic prompt could offer.

For example, let's say you are trying to create a name for a new website you created that offers virtual private math lessons. You would want the best name possible as it will be the first impression it makes on users, which may turn into paying clients. A generic prompt you may yield:

“*List me 5 names for a math tutor website*”

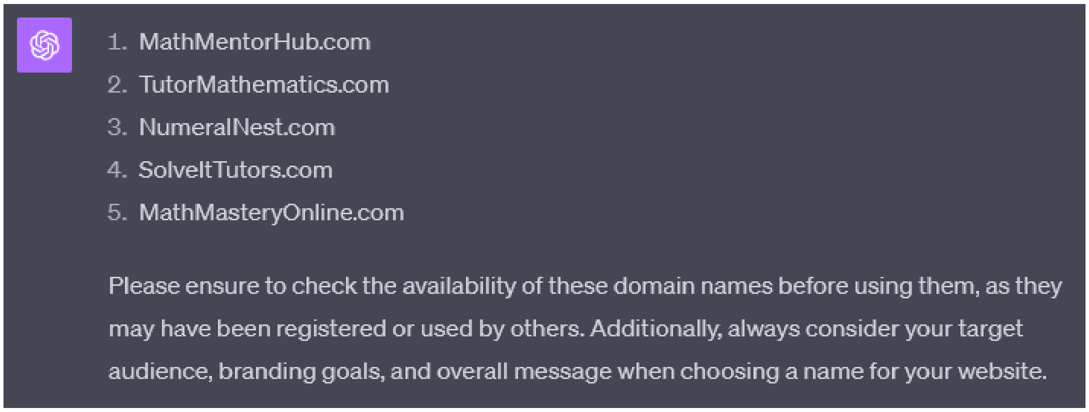


FIGURE 8: OPENAI’S GPT-4 PROMPT 1

Now these five website names provided aren’t too bad! There are some catchy names that generally explain what the site does. However, the names are a bit too generic for the website and won’t really make it obvious what is special about this site.

To improve this, we will use prompt engineering. One basic technique with prompt engineering is to feed in many more details for what names you would want. The generative AI will do its best to consider all these details to craft a suitable list of website names. Here are the results when using a more precise prompt:

“*Generate five creative and engaging website names for a math tutor platform focused on providing affordable and accessible math lessons through Zoom, targeting high school students and those at earlier levels. The lessons cover topics from Algebra 1 to Calculus 2, offered at a rate of $30 per hour. The website's mission is to make math education easily accessible to all. Please ensure the names are catchy and inspire confidence in potential students*.”

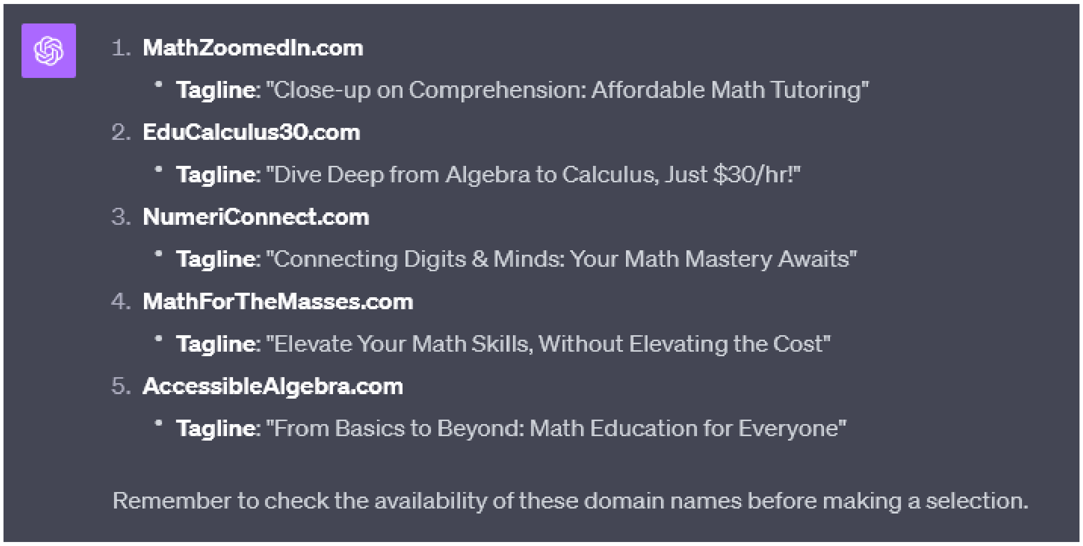


FIGURE 9: OPENAI'S GPT-4 PROMPT 2

Not only did the website names get more relevant, but it one came with its own tagline! Out of all the options, I would have ended up selecting the MathZoomedIn.com one, which wouldn't have been found if I didn't prompt it with something more relevant to the output I wanted.

Another technique used to create very precise prompts, is to turn the generative AI into the prompt engineer for itself. By asking the AI to create a prompt that would give the best possible results for an AI, it can restructure your ideas in a way that makes the AI more responsive to your specifications. For example, in the last example, that prompt was generated by ChatGPT! The prompt used to generate the prompt was:

"*Create me a prompt that would generate me the best possible results for a generative AI to create a list of 5 website names for a math tutor website. The website will be done by setting up private lessons through zoom, running at the rate of $30 per hour. Its goal is to make math accessible for anyone at high school level or earlier, with math lessons ranging from Algebra 1 to Calculus 2*."

The feedback loop from ChatGPT allowed for my requirements to be clearly organized into an effective result. Using my own written prompt might have given similar results, but in the sake of optimizing the output from the AI, this is a highly effective method.

One more strategy that is very helpful when creating prompts is that generative AI works well when given context. When debugging code, including all the code for that file, and explaining the error that is happening gives the AI all the tools it needs to attempt to solve the problem. Missing the additional context means that it will have to predict what context you didn't include, which may lead to further inaccuracy. Including context doesn't have to be in a well formatted text file either, because of the way it reacts to any prompt given, dropping in dozens of messy error lines can still be translated and used by generative AI.

For this project, any usage of generative AI should certainly utilize these prompt engineering strategies. Doing so will save time from the user’s side, save money as some generative AI’s charge per prompt, and increase the accuracy of the task given to it. All the information provided by generative AI will not be correct, as it is only trained off the internet and isn’t the internet itself, but using techniques to modify the prompt will maximize the correctness possible.

10.0 Administrative Content

10.1 Budget Estimates

TABLE 19: BUDGET ESTIMATES

	Parts	Price per part ($)	Quantity	Cost ($)
1	LED's	$0.07-3	65	$4.55-195
2	Sensors	$0.10-3	64	$6.4-192
3	Power supply	$30-110	1	$30-110
4	PCB	$40-120	2	$80-240
5	Microcontroller	$20-40	1	$20-40
6	Chess board material	$7-25	1	$7-25
7	Chess pieces set	$2.50-60	1	$2.50-60
8	Fiber optic cable	$0.30-2	64	$19.2-128
9	Display screen	$10-$90	1	$10-90
10	Charging controller	$20-80	1	$20-80
	Total			**$199.65-1160**

10.2 Milestones

TABLE 20: FALL 2023 SENIOR DESIGN 1 MILESTONES

Fall 2023		
Description	Time	Dates
Project ideas	3 weeks	August 21- September 8
Project Division	1 week	September 9 – September 15
Initial project documentation		September 15
Research on past projects	3 weeks	September 16 – October 7
Individual writing	4 weeks	October 8 - November 2
60-page draft		November 3
Design and Development prototyping	4 weeks	November 4 - November 26
Documentation Review and Purchasing Components	1 week	November 27- December 4
Final Documentation		December 5

TABLE 21: SPRING 2024 SENIOR DESIGN 2 MILESTONES

Spring 2024		
Description	Time	Dates
Testing of component	1 week	January 8 - January 14
Building prototype	8 weeks	January 15 - March 10
Testing prototype	3 weeks	March 11 – March 31
Finalizing the project	2 weeks	April 1 – April 14
Final Documentation and Presentation	2 weeks	April 15 – April 28

10.3 Work Distribution Table

TABLE 22: FALL 2023 SENIOR DESIGN 1 MILESTONES

	Alec	Alex	Cassidy	Nikolai	Vinny
Piece Identification	Primary	Secondary		Secondary	
Fiber Optic System	Secondary	Primary			
Software			Secondary	Primary	Secondary
Microcontroller			Secondary	Primary	Secondary
Power	Secondary	Secondary			Primary
PCB Design			Primary	Secondary	Primary
LED Array	Secondary	Secondary	Primary		
System Fabrication	Primary	Secondary	Secondary	Secondary	Secondary

When deciding on work distribution for our project, one of the main goals was to aim for a majority of the group (minimum of 3 out of 5 people) to have a role in each specific task so that there are at least 2 “backup” people to help the person with the primary role in a specific task. We believe this is the best way to ensure that every task in the project will be created in the best way possible through a “majority rules” idea. Another point of clarification about our work distribution table is that for system fabrication, everyone in the group will have either a primary or secondary role for that task because everyone in the group will have input and make decisions about how our project should be built, how the design/aesthetic of the board will be and how putting together all of the components inside the board will occur. Everyone in the group will be responsible for getting their specific tasks parts and components and we will come together as a group to discuss how to best implement them and create a finished product.

Appendices

Copyright Permissions

Authorization to use Lichess: https://lichess.org/contact#help-authorize

Authorization to use ChatGPT: https://openai.com/policies/terms-of-use

Approval for usage of Tom Kerrigan's Simple Chess Program (TSCP):

TK Tom Kerrigan <tom.kerrigan@gmail.com>
To: Nikolai Coletta

Tue 10/24/2023 12:05 PM

Hi Nikolai,

Sounds interesting.

As long as you aren't taking credit for TSCP, profiting off of TSCP, or redistributing TSCP, then you have my permission.

Just make it clear in the documentation and presentations for the project that you didn't write TSCP.

I'd be interested to see the presentation for the project, if you end up making a slide deck. :)

Good luck,
Tom

References

1. “Learn How To Play Chess.” Chess.com, Chess.com, www.chess.com, www.chess.com/learn-how-to-play-chess.

2. “Stockfish - Open Source Chess Engine.” Stockfish, stockfishchess.org, stockfishchess.org/.

3. “AMD & Intel Chess Benchmark – Stockfish.” IP Man Chess, ipmanchess.yolasite.com, ipmanchess.yolasite.com/amd--intel-chess-bench-stockfish.php.

4. Muller, H.G. “MAX, a MAXimum strength chess program.” Home HCCNet, home.hccnet.nl, home.hccnet.nl/h.g.muller/max-src2.html.

5. “CCRL 40/40.” Computer Chess Rating Lists, computerchess.org.uk, www.computerchess.org.uk/ccrl/4040/index.html.

6. “Elo Rating System.” Chess.com, Chess.com, www.chess.com/terms/elo-rating-chess.

7. “US Chess Rating Distribution.” US Chess, uschess.org, www.uschess.org/archive/ratings/ratedist.php.

8. Kerrigan, T.C. “TSCP: Tom's Simple Chess Program.” TCKerrigan.com, www.tckerrigan.com, www.tckerrigan.com/Chess/TSCP/.

9. “GNU Chess.” Wikipedia, en.wikipedia.org, en.wikipedia.org/wiki/GNU_Chess.

10. “Picking the Right Tool for the Job: MCU, SBC, or FPGA.” Mouser Electronics, Mouser, www.mouser.com, www.mouser.com/blog/picking-the-right-tool-for-the-job-mcu-sbc-or-fpga.

11. “SparkFun Pro nRF52840 Mini - Bluetooth Development Board.” SparkFun, www.sparkfun.com, www.sparkfun.com/products/15583.

12. “Teensy 4.0 Pins.” PJRC, www.pjrc.com, www.pjrc.com/store/teensy40_pins.html.

13. “Pixel Pipelines Definition.” Techopedia, www.techopedia.com, www.techopedia.com/definition/89/pixel-pipelines.

14. “ESP32 Datasheet.” Espressif Systems, espressif.com, www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

15. “HiLetGo STM32F407VET6 STM32 Cortex-M4 MCU Core Board Development Board.” Ubuy Indonesia, www.ubuy.co.id, www.ubuy.co.id/en/product/23G74LD8-

hiletgo-stm32f407vet6-stm32-cortex-m4-mcu-core-board-development-board-nrf2410-fmsg-sd-card.

16. “PSRAM for Teensy 4.1.” PJRC, www.pjrc.com, www.pjrc.com/store/psram.html.

17. “ChatGPT traffic slips for third month in row.” Reuters, www.reuters.com, www.reuters.com/technology/chatgpt-traffic-slips-again-third-month-row-2023-09-07/.

18. “Collaborative Intelligence: Humans and AI Are Joining Forces.” Harvard Business Review, hbr.org, www.hbr.org/2018/07/collaborative-intelligence-humans-and-ai-are-joining-forces.

19. “What is prompt engineering?” McKinsey & Company, www.mckinsey.com, www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-prompt-engineering.

20. OpenAI. “Chat with OpenAI.” Chat OpenAI, chat.openai.com, chat.openai.com/.

21. “Safety of toy electric appliances for children - IEC 62115.” International Electrotechnical Commission, webstore.iec.ch, webstore.iec.ch/publication/28054.

22. “Secondary cells and batteries containing alkaline or other non-acid electrolytes - IEC 62133.” International Electrotechnical Commission, webstore.iec.ch, webstore.iec.ch/publication/32662.

23. “ISO/IEC 9899: Programming languages — C.” ISO, www.open-std.org, www.open-std.org/jtc1/sc22/wg14/

24. “ECSTUFF4U for Electronics Engineer.” *Advantages and Disadvantages of LASER Diode*, www.ecstuff4u.com/2018/02/advantages-and-disadadvantages-of-laser.html. Accessed 3 Nov. 2023.

25. Meyer, Michael. “HeNe Lasers vs Diode Lasers: Hene Laser Pros and Cons.” *Solid State Lasers and Laser Diodes from RPMC Lasers Inc*, 4 Aug. 2023, www.rpmclasers.com/blog/hene-lasers-vs-diode-lasers-hene-laser-pros-and-cons/.

26.“Fiber Optic Lighting.” *FOA*, www.thefoa.org/tech/lighting/lighting.html. Accessed 3 Nov. 2023.

27.ryno419. “The Pros and Cons of LED Lights.” *Brennan Electric*, 26 June 2023, brennan-electric.com/the-pros-and-cons-of-led-lights/.

28.You, Semiconductor For. “What Are the Advantages and Disadvantages of Laser Diode?” *Semiconductor for You*, 27 May 2021, www.semiconductorforu.com/advantages-disadvantages-laser-diode/.

29.Conditioning, Berkeys Air. “Advantage & Disadvantage of LED Lights.” *Berkeys Plumbing, Air Conditioning & Electrical*, 7 Apr. 2020, www.berkeys.com/2016/11/16/advantage-disadvantage-led-lights/.

30.Paschotta, Dr. Rüdiger. “Multimode Fibers.” *Multimode Fibers, Explained by RP; Optical Fiber, Large-Core Fibers, Fiber-Coupled, Laser Diodes*, RP Photonics AG, 2 Nov. 2023, www.rp-photonics.com/multimode_fibers.html.

31.Paschotta, Dr. Rüdiger. “Single-Mode Fibers.” *Single-Mode Fibers, Explained by RP; Launching Light, Monomode Fiber, Cut-off Wavelength*, RP Photonics AG, 2 Nov. 2023, www.rp-photonics.com/single_mode_fibers.html.

32.“What Do You Mean by Mode in an Optical Fiber?” *Quora*, www.quora.com/What-do-you-mean-by-mode-in-an-optical-fiber. Accessed 3 Nov. 2023.

33. Shaik, Asif. “Light Emitting Diode (LED).” *Light Emitting Diode (LED) - Working, Construction and Symbol - Diode*, www.physics-and-radio-electronics.com/electronic-devices-and-circuits/semiconductor-diodes/lightemittingdiodeledconstructionworking.html. Accessed 3 Nov. 2023.

34.“Thorlabs - HNL020RB HeNe Laser, 632.8 Nm, 2 MW, Random, 100 - 240 VAC Power Supply Included.” *Thorlabs, Inc. - Your Source for Fiber Optics, Laser Diodes, Optical Instrumentation and Polarization Measurement & Control*, www.thorlabs.com/thorproduct.cfm?partnumber=HNL020RB. Accessed 3 Nov. 2023.

35. *I2C-Bus Specification and User Manual - NXP Semiconductors*, www.nxp.com/docs/en/user-guide/UM10204.pdf. Accessed 3 Nov. 2023.

36. “IPC-A-610: The Standard for Acceptability of Electronic Assemblies.” *NextPCB*, www.nextpcb.com/blog/ipc-a-610. Accessed 3 Nov. 2023.

37. Kamprath, and Instructables. “RGB Led Matrix.” *Instructables*, Instructables, 30 Dec. 2017, www.instructables.com/RGB-LED-Matrix-1/.

38. “L6: RGB Leds.” *Physical Computing*, makeabilitylab.github.io/physcomp/arduino/rgb-led.html#:~:text=With%20the%20Common%20Anode%2C%20the,levels%20(e.g.%2C%20ground). Accessed 3 Nov. 2023.

39. Team, The Sierra. “IPC J-STD-001 Standard for Soldering.” *Sierra Circuits*, 28 Sept. 2023, www.protoexpress.com/blog/ipc-j-std-001-standard-soldering-requirements/.