System Design for Chess Connect

Team #4,
Alexander Van Kralingen
Arshdeep Aujla
Jonathan Cels
Joshua Chapman
Rupinder Nagra

January 18, 2023

1 Revision History

Date	Version	Notes
2023-01-11	Arshdeep Aujla	Introduction, Purpose, User Interface, Other Considered Designs
2023-01-16	Joshua Chapman	Scope
2023-01-18	Joshua Chapman	Project Overview, System Variables

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

symbol	description
Chess Connect	Explanation of program name
[... —SS]	[... —SS]

Contents

List of Tables

List of Figures

3 Introduction

This document outlines the system design portion of this project's design documentation. Design documentation is intended to separate the project into modular components to increase the project's understandability and reusability.
Other useful documents for this project are the following:

- SRS
- HA
- VnV

4 Purpose

The purpose of this document is to outline a detailed system design. The system design includes system variables, user interfaces, a timeline for completion, and designs of many different components such as hardware, electrical componets, and communication protocols. This document also includes references to an intesive list of the project's mechanical and electrical components and a reflection in the appendix.
Other documents relating to design are the following:

- Software Architecture Document
- Detailed Design Document

5 Scope

The Chess Connect system includes a physical chess board and associated software application to aid in the learning and sharing of in-person chess games. Mechanical design of the chess board and electrical systems are included in the scope. Design of the user interface and communication between the systems is also included. Not included is the design of the chess AI being used as a learning tool. Exisiting chess websites and platforms are also out of the scope of Chess Connect.

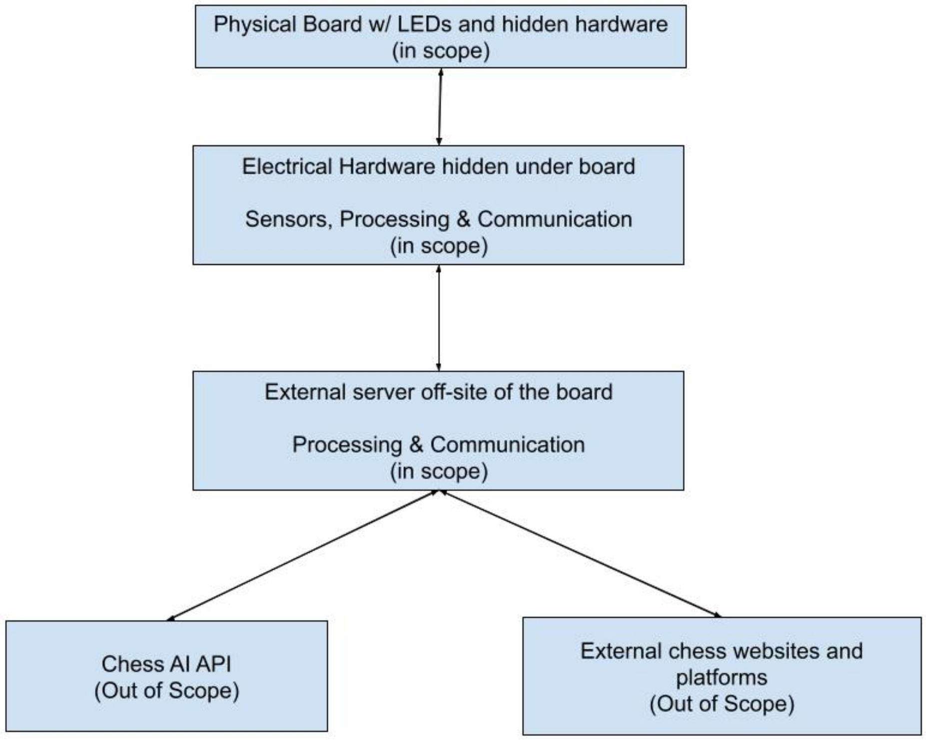

Figure 1: System Scope

6 Project Overview

6.1 Normal Behaviour

The normal operation of the chess board involves configuring game mode and settings, reading the physical positions and identifiers (magnetic strength) of the pieces, lighting up corresponding LEDs and determining legal moves on the micro-controller. There will be three game modes: Normal Mode (no LED feedback), Engine Mode (best moves calculated by a chess engine and displayed by an LCD display) and Beginner Mode (legal moves displayed by LED when a piece is picked up). The micro-controller will also be simultaneously transmitting data to the server via Bluetooth and receiving responses in the form of "best" moves from the server. The server will be calculating these best moves in real time depending on the configuration of the pieces on the board sent from the chess board, and sending it back over Bluetooth every time it is queriedwhile the game is in "Engine Mode". The server will also be communicating all of this information to a web application where users can tune in and watch the pieces and seegame stats in real time.

6.2 Undesired Event Handling

When an unexpected event occurs, chess connect will always return to the most appropriate safe state. When an action during a game, the on-board LCD screen displays the error and provides instructions to correct the mistake and proceed normally. When an error occurs in the communication between sub-systems, all activememory is stored and a default error screen appears on the LCD screen to inform the players.

6.3 Component Diagram

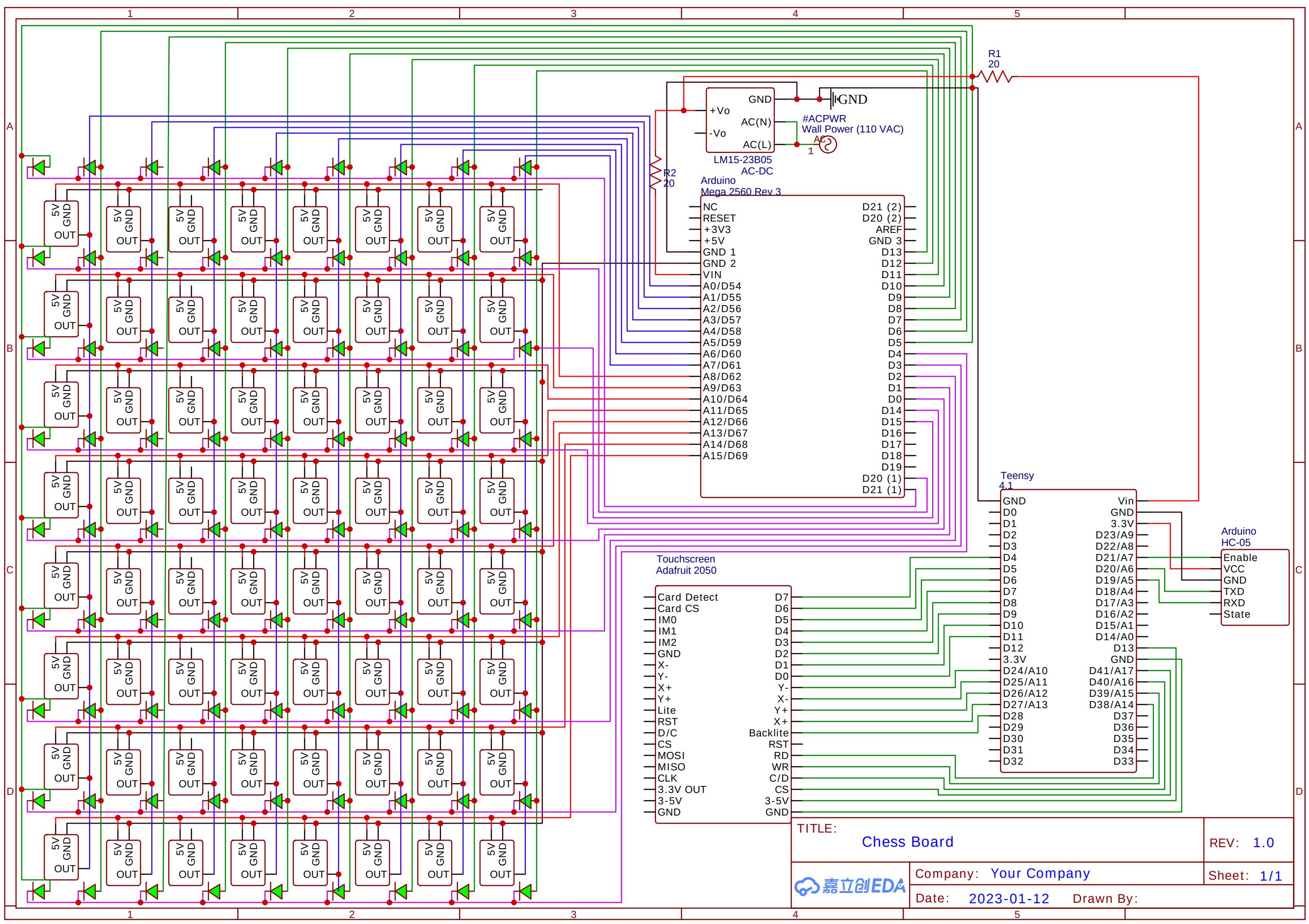

Figure 2: Electrical Wiring Diagram

6.4 Connection Between Requirements and Design

Requirements GA1, GA2, GA3, GA4, GI1, GI2, GI3, NB2, NB3, EB2, EB3, EB4, BB3, BB4, BB5 describe a variety of buttons to support game flow and modes. Too many physical buttons were required so an LCD screen was placed on the side of the board to accomodate the large number of functions.

Requirements GI5, GI6, ND1, NA1, ED1, ED2, EA1, BD1 dictated decisions for specific information to be communicated from the board to the Web Application. Serial Communication was chosen to display a custom string to communicate the necessary game state, errors and best moves between the board and the application.

Requirements NB1, EB1, BB1 describe the memory performance required of the hardware to store game data of entire games. Design satisfies this requirement with the addition of two processors to share the memory load and processing load.

Requirement BB2 outlined the implementation of LEDs requiring fast response times. Due to communication protocols, screens and sensors are already consuming processor time, a second processor was added with faster processing speed to handle the LEDs and sensors exclusively.

7 System Variables

7.1 Monitored Variables

Variable	Units	Description
s_a{1-8}	Volts	States of tiles a1 - a8 on the board. They are analog signals converted to digital and the state of the tile is determined. The possible states of each tile is empty, black/white pawn, black/white rook, black/white knight, black/white bishop, black/white queen, black/white king.
s_b{1-8}	Volts	States of tiles b1 - b8 on the board. " "
s_c{1-8}	Volts	States of tiles c1 - c8 on the board. " "
s_d{1-8}	Volts	States of tiles d1 - d8 on the board. " "
s_e{1-8}	Volts	States of tiles e1 - e8 on the board. " "
s_f{1-8}	Volts	States of tiles f1 - f8 on the board. " "
s_g{1-8}	Volts	States of tiles g1 - g8 on the board. " "
lcd_x_pos	Volts	Positive X coordinate of the resistive touch screen.
lcd_x_neg	Volts	Negative X coordinate of the resistive touch screen.
lcd_y_pos	Volts	Positive Y coordinate of the resistive touch screen.
lcd_y_neg	Volts	Negative Y coordinate of the resistive touch screen.
bluetooth_rx	Volts	Serial communication receive channel via bluetooth to the web application. Best moves from the chess engine are sent to the chess board via this channel.

7.2 Controlled Variables

Variable	Units	Description
hall_power{1-8}	Volts	Eight sources of power to the eight columns of sensors under the board. The power will be pulse width modulated to multiplex the signals of the hall sensors.
LED_row_pos{1-9}	Volts	Nine power supplies are delivered to the nine rows of LEDs. When a corresponding ground is supplied to a column, an individual LED can be lit.
LED_col_gnd{1-9}	Volts	Nine ground supplies are delivered to the nine columns of LEDs. When a corresponding power is supplied to a row, an individual LED can be lit.
LCD_Display{0-7}	Volts	Eight bits of information are sent to the LCD screen to display a variety of images.
bluetooth_tx	Volts	Serial communication transmit channel via bluetooth to the web application. The current game state and errors are sent via this channel.

7.3 Constants Variables

Constant	Unit	Value
Chess board width	inches	12
Chess board length	inches	12
Chess board tile width	inches	1.5
Chess board tile length	inches	1.5
Supply Power to Board	Volts	110 VAC

8 User Interfaces

Hardware Interface

The user will interact with two main components of the hardware.

- Magnetic chess pieces
- Physical chess board containing sensors

They will interact with the chess pieces and chess board as they would in a normal chess game. The chess board reflects a standard chess board, with LEDs in the center of each square. They would move the chess pieces on the board and remove them in according to the rules of chess. If the device is set to beginner mode, the LEDs will light up in according to which available moves are available for that chess piece. They would interact with the LEDs by using them as a guide for potential moves to make.

Software Interface

The users will interact with the software component of this device through a web application. They would need a device with an internet connection and and internet browser to view the application. The user will interact with this interface by visual viewing the chess board status in real time including a visualization of the chess piece locations. They will also be able to turn on and off beginner mode in this interface through clicking an interactive button in the web application.

9 Design of Hardware

[Most relevant for mechatronics projects —SS] [Show what will be acquired —SS] [Show what will be built, with detail on fabrication and materials —SS] [Include appendices as appropriate, possibly with sketches, drawings, CAD, etc —SS]

10 Design of Electrical Components

Aquired Components

Please refer to Appendix C for a detailed list of aquired components.

[Most relevant for mechatronics projects —SS] [Show what will be acquired —SS] [Show what will be built, with detail on fabrication and materials —SS] [Include appendices as appropriate, possibly with sketches, drawings, circuit diagrams, etc —SS]

11 Design of Communication Protocols

[If appropriate —SS]

12 Timeline

[Schedule of tasks and who is responsible —SS]

A Interface

[Include additional information related to the appearance of, and interaction with, the user interface —SS]

B Mechanical Hardware

- Arduino Mega

C Electrical Components

- Arduino Mega 2560
- Teensy 4.1
- 3.5" Touchscreen Display
- HC-05 Bluetooth Module
- AC-DC converter for
- 81 green LEDs
- 81 1000ohm resistors
- 64 HALL sensors
- 64 1mF capacitors
- 30 ft, 20 AWG wire
-

D Communication Protocols

E Reflection

Project Limitations

Other Considered Designs

One problem that we had to overcome in our design is that there are not enough input and output pins in one microcontoller for all of the components. One solution we considered was having multiple microcontrollers for this project to ensure there is one input pin for each input sensor and one output pin for each output. This design would be beneficial in the way of simplicity of code. The tradeoff would be the complexity in cordinating the communitcation between multiple microcontrollers and the web application. A second option to solve this problem is to use multiplexing to reduce the number of input and output pins needed for a large number of compnents. This option requires more complex code, but only requires one microcontroller device. We chose to implement the multiplexing option becasue it only used one device which saves money, as well as eliminates the need to coordinate between multiple microcontrollers.

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)

2. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select documented design? (LO_Explores)