DEC13-10: Team AutoChess

DEC13-10

DEC13-10
Riley Chaplea
Matt Dresser
Jordan Jacobson
Riley Swindell
Allison Thongvanh
Blake Vermeer

Client

Joseph Zambreno

Table of Contents

Introduction

Project Summary

The purpose of this project is to create a chess board that can play against a person autonomously. A microcontroller interacts with the engineered chess surface and individual pieces, detecting human movements, moving the computer pieces at the appropriate time using artificial intelligence (AI), and determining capture and end-game conditions. A configuration application will allow capture and broadcast individual games and allow the board to be remotely reset.

System Overview

Functional Requirements

After discussion with our client, the following items are the project's functional requirements:

- Chess piece movement system must be unobservable by players.
- Base must contain piece movement system such that the system can be revealed.
- Piece movement system must move fluidly, quickly and quietly.
- User interface must allow players to choose AI skill level.
- User interface must allow players to undo moves.

Non-Functional Requirements

After discussion with our client, the following items are this projects non-functional requirements:

- Base and pieces must not be overly large.
- User interface must be comparable to computer/game console chess games
- Pieces controlled by AI and piece movement system must move seemingly magically.

Overall Design Decisions

Based on the functional and non-functional requirements we decided to use magnets to physically move pieces and infrared light emitting diodes (IR LED's) to identify piece location. Also, we implemented a wireless power scheme to power the chess pieces, eliminating the need for batteries. This approach had several benefits and a few drawbacks.

- Pros
  - The movement system can be entirely hidden from the user.
  - Chess piece movements can be detected without any user intervention.
  - The maintenance of the chess board would be minimal due to no battery requirements.
- Cons
  - This was a fairly complex design with many interdependent parts.
  - This design was fairly costly, although our advisor told us to not worry about the budget.

Design Decisions Conclusion

Based on the requirements for the project, our group felt that this was the best approach despite the complexity and cost of this design.

Schematics/Architectures

Below is a diagram showing how all the components of the chess board and those to be added to the chess board fit together.

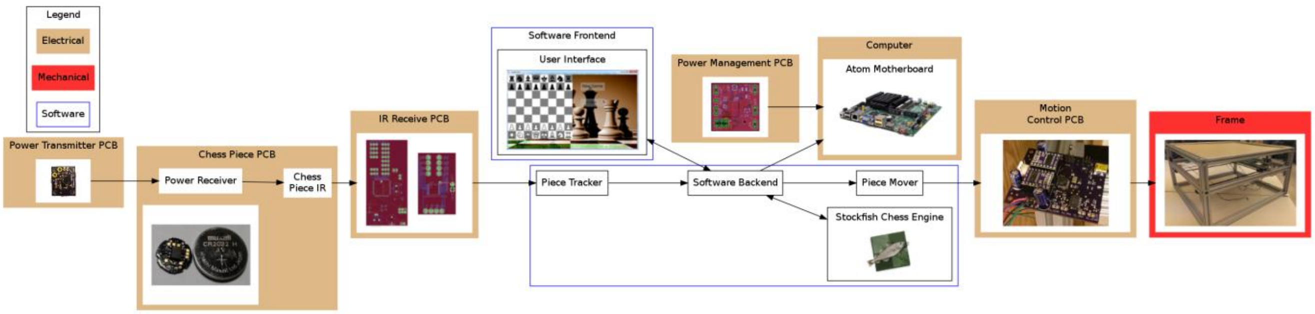

Fig. 1 System diagram of the Chess Board

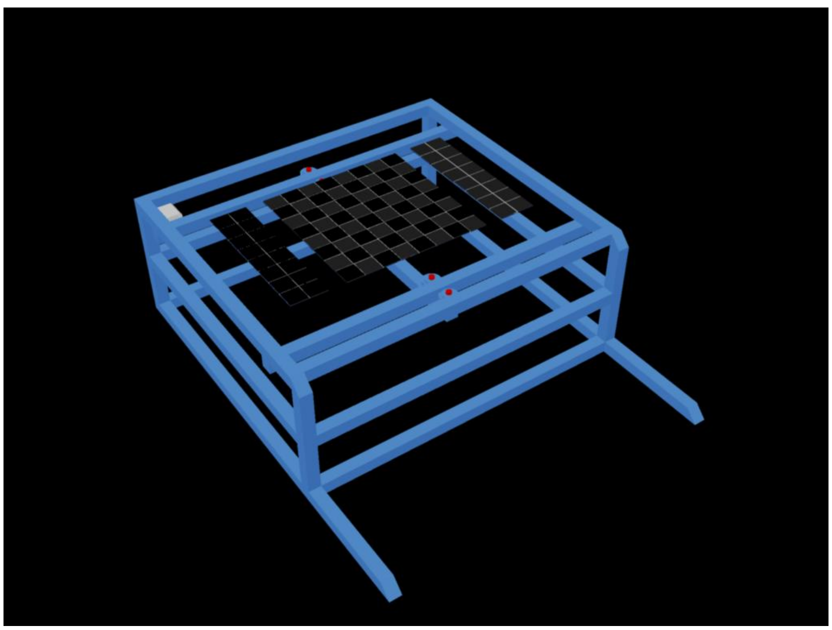

Fig. 2 Early AutoCAD rendering of movement system design

Design Details

This project contained eight major parts. Below is a brief description, a list of requirements, and details of the implementation of each major part.

Motion Controller Specifications

The motion controller is responsible for moving the chess pieces around the board. It consists of two stepper motors and a servo to raise and lower the magnet. The requirements for this part were:

Requirements

- Pieces must move from one corner of the board to the other in under 5 seconds.
- Movement must be relatively silent.

Given these requirements I design a board that used Pololu stepper motor drivers that are controlled by an ATMega328. I then used an FTDI chip to provide a USB interface between the Atom board and the motion controller.

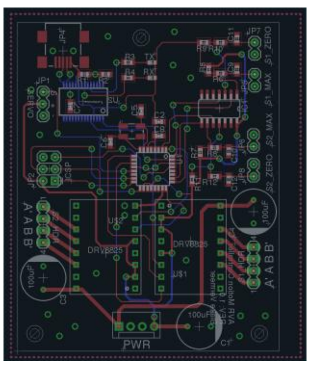

Fig. 4 – Motion Controller PCB Layout

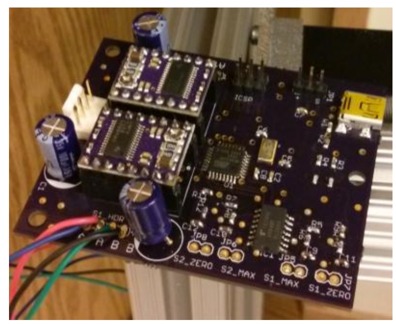

Fig. 4 - Chess Piece PCB

Chess Piece PCB (IR Beacon) Specifications

Our beacon was developed to act as a method for identifying each chess piece. Having the constraint of not being able to visually interact with the internal workings of the chessboard, an infrared LED along with infrared transparent Mylar was our solution. This also fit within the constraint of needing the PCB to be small enough to fit within a 15mm chess piece base. The beacon also must be low power to effectively work with our wireless power system.

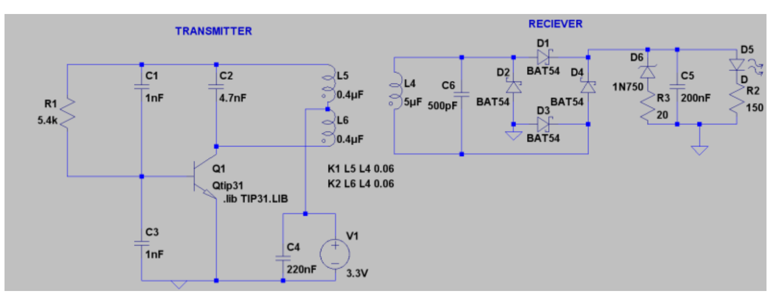

Fig. 3 Power transfer circuit schematic

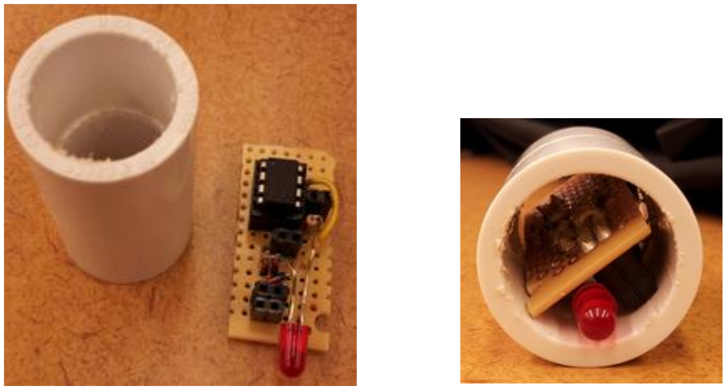

Fig. 4 - Chess Piece PCB Prototype

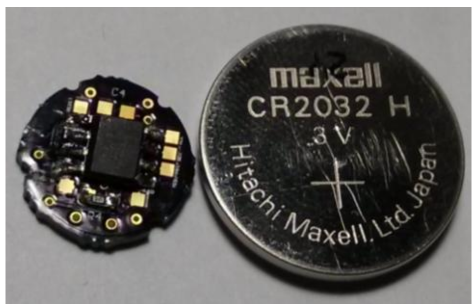

Fig. 5 - Chess Piece Rev. 2 PCB

Power Transmitter Specifications

The power transmitter needed to be as reliable as possible in terms of maintaining a constant frequency. This way we would be able to better refine the IR beacon's RLC characteristics to have a resonant frequency close to that of the transmitter. We also needed to have a low parts count to minimize cost as much as possible.

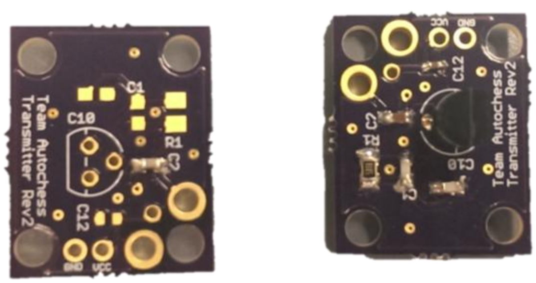

Fig. 6 – Power Transmitter Rev. 2 PCB

IR Receiver Specifications

The purpose for the IR receiver is to detect the IR codes sent by the Chess pieces and determine which spot the chess piece is on. The chess pieces transmit their codes or IR using the UART protocol with a modulation of 36 kHz. To make the firmware easier to implement, an ATMega128 was chosen because it features dual UARTs. This allows the IR receiver to use one UART hardware module to interpret the codes sent by the chess pieces and another UART hardware module to communicate with an FTDI chip to communicate with the Atom board over USB.

The IR receiver board needed to be able to multiplex between all 96 IR sensors located underneath the possible chess piece locations. To accomplish this and multiplexer tree design was used. This also allowed the IR receiver board to be modularized into several smaller boards which allowed the IR receiver to be designed within the constraints of the free version of Eagle CAD.

User Interface

After discussion with our client, the following items are this project's user interface requirements:

- Ability to display a 2D chess board that represents the physical boards current state
- Display menus to allow user to choose difficulty level
- Have ability to give the user an interface to interact with the chess AI and board

Based on the requirements given we have designed our interface to be displayed on a 7" LCD touch screen. On an Atom board running Windows 8, we have an interface on which observers of the touch screen start a game, choose an opponent difficulty level, end a game, and view the current state of the board (as indicated in the figures below).

Fig. 5 – In game user interface with debug output

Fig. 6 - User Interface Start Screen

I/O Specifications

- Chess Piece Identification
  - Pieces transmit their ID by sending a unique code using the UART protocol over IR.
  - Piece identification circuit keeps track of piece positions and report back to the Atom board.
  - Maximum receiving circuit input voltage – 5.5V
  - Typical receiving circuit input voltage - 5V
- Universal Chess Interface
  - Communication protocol that enables the chess AI to interface with the board
- Universal Serial Bus (USB)
- UART

Hardware Specifications

- Chess Piece Circuit
  - Maximum input voltage – 5V
  - Typical input voltage – 3.3V
  - Minimum input voltage – 1.3V
  - Current draw – ~15mA continuous

- Transmitter Coil Circuit
  - Typical input voltage - 12V
  - Diameter of Coil - ~1.87"in
  - Inductance - ~0.744µH

- Receiver Coil Specifications
  - Diameter of Coil - ~.75"in
  - Inductance - ~55.7µH
  - 50 turn coil

Software Specifications

After discussion with our client, the following items are this projects software specifications:

- Use universal chess interface as our protocol of choice when interacting with the board and chess AI
- Communicate over USB to interface with microcontroller
- Determine current state of the chess board (piece location & possible moves)
- Push chess board information to user interface to be displayed
- Take menu data from user interface and push it to the computer for processing

Implementation Challenges

Inductive Power Transfer

The amount of power transferred is highly dependent on the load resistance. We are using a microcontroller to blink an IR LED on the chess pieces. When the LED is on, the load resistance is decreased, which lower the voltage over the load. When the LED is turned back off, the voltage across the load is increased again. This was causing problems with the operation of our microcontroller and causing voltage spikes. To overcome this challenge we connected a dummy load to another pin of the microcontroller and alternating between switching on the LED and switching on the load. This keeps the load impedance consistent which in turn keeps the voltage across the load constant.

Transmitting Coil Design

To make a high quality inductor, the wire has to be a fairly low gauge and wound in a long spiral. This causes the inductor to physically take up a lot of space. The space where we need to fit the transmitter coil is fairly space constrained. Therefore, our inductor design will create a lower quality coil. This challenge can be overcome by creating a more efficient transmitting circuit and tuning the receiving circuit to resonate at the frequency at which the transmitting circuit is operating at. This will increase the overall efficiency of the power transmission and compensate for the lower quality of the transmitter inductor.

Movement Speed

A major functional requirement given was that the piece movement system must move fluidly, quickly and quietly. The mechanical base design must be built such that the movement system can move a piece diagonally across the board in 4 to 5 seconds. Due to this requirement, the base design has undergone many changes. The first design used stepper motors to rotate threaded rods (like a screw driver to a screw) to move a platform in the X direction. After finding that the threaded rod system was too slow due to the pitch of the rods, the second design used Acme rods—similar to threaded rods but inherently designed for CNC machines. However these rods were made more for precision than the speed desired. Currently the design features an 80/20 aluminum frame, with timing belts in place of rods. These timing belts allow the movement system to move a piece faster than the previous screw-based designs.

Chess Engine Integration

The chess engine utilized was an open source C++ engine called Stockfish. The user interface is fully integrated with Stockfish in order to detect legal moves, keep track of the board state, and play the opposing side of the chess board. Stockfish was an unknown code, and did not provide a simple API for implementation. Several weeks went into learning how the engine worked, and how to properly access the various methods and protocols involved.

Risks

There are a few risks associated with this project. One of these risks is that the magnetic fields generated by the inductive power transmission circuit will interfere with the capacitive touch screen. This could cause the chess board to operate sporadically and not properly respond to user input. To mitigate this risk we can mount the screen in a metal enclosure to help shield it from magnetic interference.

Project Details

Standards

For this project we are following the guidelines of several different standards.

- IEEE 1801-2009 standard for labeling the power components of the circuits.

Expected Deliverables

After discussion with our client, the following deliverables were decided on with the following features:

- An automated chess board
    - Touch-screen user interface for configure and monitoring the game.
    - Enclosure that will have removable side to reveal the inner workings of the chess board.
    - Near silent operation.
    - Aesthetically pleasing.

Appendix A

Cost

The costs of the project so far are as follows:

Component	Cost ($)
Frame	707.66
Fasteners	536.92
Aluminum Extrusions	97.17
Miscellaneous	73.57
Printed Circuit Boards	188.60
Chess Piece (32)	92.16
Power Transmitter (16)	4.80
Motion Controller	57.69
ACPI Power	4.94
IR Receiver Multiplexer	2.03
IR Receiver Main	26.98
User Interface	463.00
Elo 7” Touch Screen	250.00
Software Platform	168.00
Miscellaneous	45.00
Total	$1332.26