Team BO: Seamless Autonotator

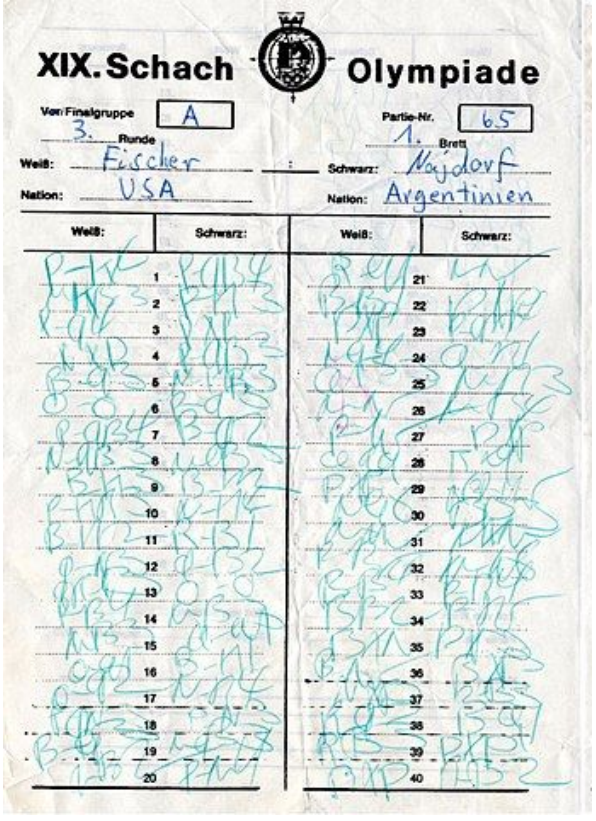
Vikram Marmer, Patrick Joyce, Ryan Guan

Problem Statement/Use Case

- Notation is the way chess moves are recorded

  (Nc3, Bxf6)

- Notation is useful
- Possibility of errors and time-consuming
  - Handwriting
  - Forgetting to notate
- Solution: Create a system that makes it easier for chess players to notate

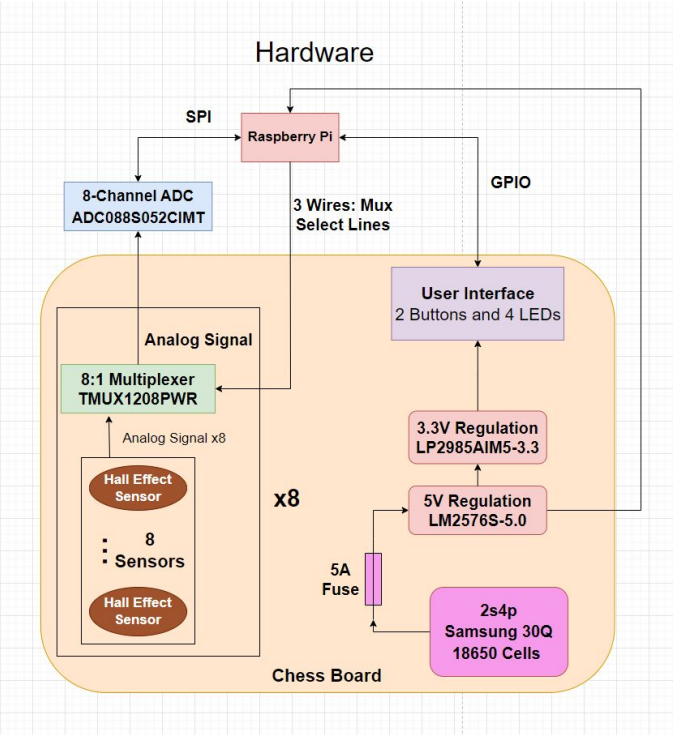


Requirements

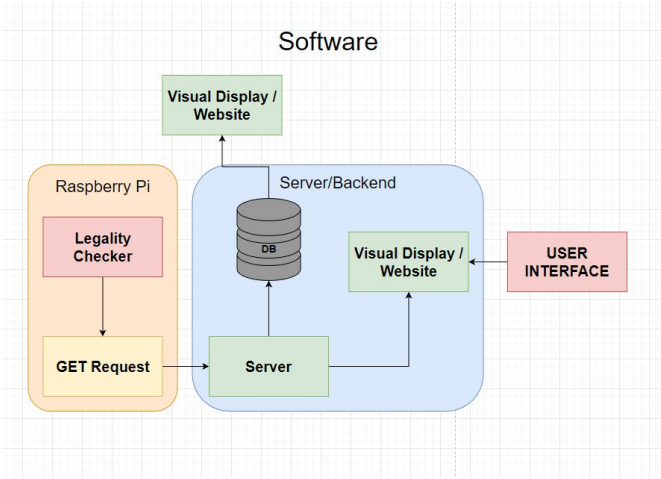
- Record notation with **100% accuracy**
- Report move legality in **300ms** or less
- Provide **10 hours** of battery life
- Access previous **10 games** through website
- Provide exportable PGNs through website for analysis on Chess.com

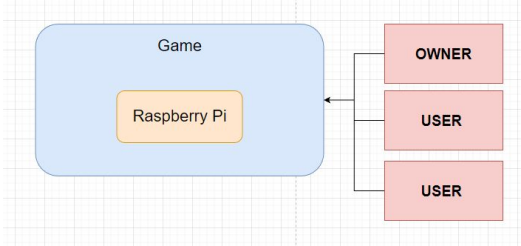
Block Diagram

HW Unchanged



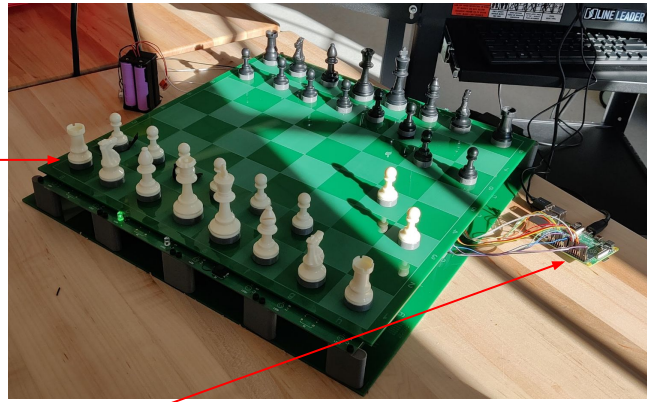
Changed GET Request

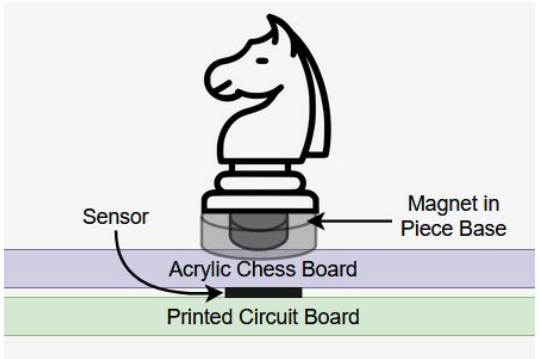




Complete Final Solution-Custom Board

- Chess board made from acrylic
- Plastic pieces
  - Magnet on bottom of each piece in a 3d-printed base
- PCB with hall-effect sensors directly under the board
  - Bipolar sensors for simplification
  - One sensor per square, multiplex into ADC
- Raspberry Pi reads sensors and controls LEDs on the board
- Power
  - Integrated batteries under board
  - Internal voltage regulation



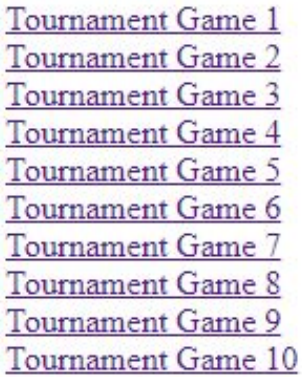


Complete Solution-Legality Check

- When user presses a button, the C++ program will:
  - Read every sensor on the board
  - Compare the new board state to the old board state, and determine the source and destination squares of the move
  - Compare the move against the legal move list (with Stockfish)
    - If legal, translate move to Standard Algebraic Notation (eg. c3d5 -> Nxd5), Light green LED and wait for next move
    - If illegal, light red LED and wait for the move to be corrected

Complete Solution: Software

- Implemented framework using Django
  - Used AJAX for asynchronous updating
  - Viewed past games using SQLite Database
  - Sent data from Pi using Python requests
    - Used GET Requests to transmit data
  - Deployed using repl.it

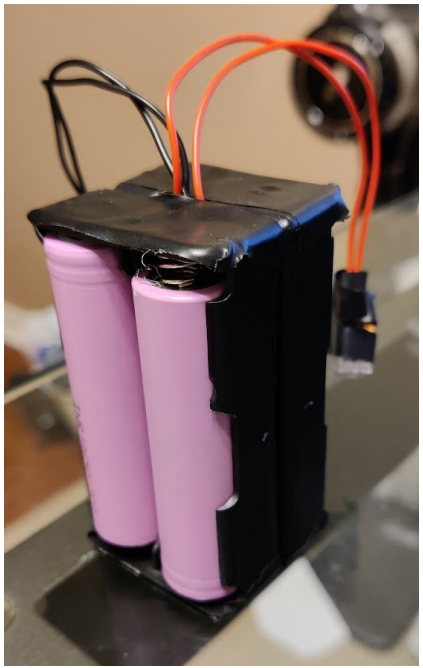


Testing Verification – Notation and Latency

- Requirements (revisited)
  - Move pieces at playing speed and test for accuracy of notation
  - Goal: 100% notation accuracy - if it’s not reliable, it’s useless
- Met this goal in preliminary tests - 3 games, over 30 moves total
  - Sensor thresholds need to be adjusted, as squares are sometimes reported empty when pieces were off center - causing legal moves to appear illegal
- Latency was not tested quantitatively, but feels very fast
- Plan: play through 3 long games that will target specific edge cases in legality and notation
  - Record video, and analyze to determine latency

Power Testing

- Power Usage: <= 8W
  - Tested with a power supply
  - Pi + Sensors + ADC
- Target: 10 Hours
  - 80 Wh minimum capacity
- 2 series 3Ah lithium battery
  - 3.6V nominal
  - 11.1 Ah capacity required
- Therefore: Need 4 parallel
  - Have space for 8 on board



Design Tradeoffs

Magnet Choice vs. Software Complexity

- Larger magnet more accurately detected
  - Hard to find magnet strength variety at size
- Use simple piece color detection over type
- Software tracks piece type

Magnet Name	Error (mT)	Diameter (mm)
9149	-51.33316327	6.35
8019	-78.24633929	6.35
9144	-108.9707283	6.35
8004	-36.95306122	6.35
8176	**-6.338968112**	**12.7**
8005	-34.26718857	6.35

Software Tradeoffs

- Server Framework: Django vs Flask vs NodeJS
  - Many of these features that differentiate these frameworks don't make a difference for our MVP
  - Chose framework based on familiarity
- Database: SQLite vs MySQL Database
- Rest API vs GET Request

Schedule


Task
Start of Week
8/29/2022
9/5/2022
9/12/2022
9/19/2022
9/26/2022
10/3/2022
10/10/2022
10/17/2022
10/24/2022
10/31/2022
11/7/2022
11/14/2022
11/21/2022
11/28/2022
12/5/2022
Chess Board (Circuits and Hardware) - Vikram
Sensor Research and Validation
PCB Schematic
PCB Layout and Ordering
PCB Assembly
PCB Hardware Tests and Validation
Making/Modifying Physical Board
Firmware and State Logic - Patrick
Legal Move Generation
Legality Check
Notation Legality Check
Interface with Hardware
Interface with Software
Output State Logic
Software and Web - Ryan
Creating Chess Class and Object Oriented Structure
Test Python vs C++ Latency Requirements
Create Website
Create Autonotator and interface Autonotator with Website
Allow Website to Read and Write from Database
Allow Website to export notation to chess.com
Create basic analysis tools within website (past games)
Integration - All
Test Data Collection with PCB
Testing Interface with RPi
Testing Web Interface
Validating Final Product - All
Hardware Testing-Power, Accuracy, Latency
"User" Testing

Conclusion

- Tasks before final demo
  - More “User” Testing
  - Record actual latency to confirm calculations
  - Play more games to stress test board
  - Test latency of website display
- Takeaways
  - Planning for slack time is crucial
  - Magnets *really really* like to stick together

Accuracy	100% (given current testing)
Latency for Legality	50 ms (per our calculations)
Battery Life	20 Hours
Board Size	18” x 18”
Stockfish Version	11 (Jan 2020 release)