Internet Connected Chessboard

Spring 2019 Team 61: Jeffrey Ito, Joel Mathews, Ritish Raje
TA: Thomas Furlong
Introduction

- Very Easy to play against someone around the world
- Online chess applications are a great way to play chess against players from all over the world
- A user interface on the computer screen is the traditional way to interact with these applications
Objective

- To regain the physical interface of an actual chess board while maintaining the ability to play against players from all over the world
High-Level Block Diagram
Main Requirements

- Chessboard transmits data to application through WIFI
- Application sends data to chessboard through WIFI
- Move validation
Power

- 9 V, 600 mAh rechargeable battery
- 5 V, 500 mA fixed-voltage LDO regulator
Reed Switches

- Magnetically activated switches
- Detect pieces on chessboard
Reed Switches
Shift Registers

- Facilitate communication between the MCU and reed switches
- Facilitate communication between the MCU and LEDs
Shift Registers

- Parallel-in serial-out shift registers take in parallel data and output serial data
- Serial-in parallel-out shift registers take in serial data and output parallel data
Microcontroller

- ATmega328p
  - UART and SPI communication
  - 32 KB of flash programmable memory
  - Able to operate at 5 V
Microcontroller
Communication Design

Microcontroller

- Used WiFi to allow communication across the globe
- ESP8266-01s WiFi Module
- 4 Input/Outputs
- Cheap
Communication Design

Web Application

- Used FLASK microframework for Python
- Easy to set
- Ideal for prototyping
Data communication

Board to MCU

- Board sends 8 bytes of data to MCU.
- 1 byte corresponds to 1 row on chessboard
  => 1 bit corresponds to 1 square
- 8 bytes * 8 bits = 64 = number of squares on board
Data Communication

MCU to WiFi Module

- Serial Communication is used to send the 8 bytes to WiFi Module
- TX (MCU) -> Logic Converter (5V to 3.3V) -> RX (WiFi Module)
Data Communication

WiFi Module to Flask Python App

- Wifi Module makes POST request to FLASK endpoint with 8 byte data
- Endpoint parses and converts into 8x8 array to change state of chess application
Python Chess Application

- Running on computer through python script
- Brains of whole project
- Saves the game state
Chess Application Implementation

- Python chess application does most data processing
- State is continuously being sent to application
- Could have coded move detection in MCU
- Uses Python-Chess Library
- More concise to use python
Chess Application Communication

- Python script creates its own server
- Python script sends data by connecting to WiFi Module’s servers
Application Data processing

- Gets game state from the chess board
- Sends back opponents move
- Tells chess board which LEDs to light
Ethical Issues

● Cheating by taking advantage of software

● Hackers attacking the website (taking personal information)
Successes

- Fully integrated all communication modules
- Able to send move from Board to the chess application
- Able to send move from Pygame to Chess Board
Challenges

- Wiring
- Serial communication
- Programming microcontrollers
- Trouble getting Wifi module to send data
Conclusion
Future work

● Redesign wiring system

● Fine tune chess application algorithms

● Reduce latency

● Reduce power consumption
Questions?
References


