ARCANE GAME BOARD

GROUP 22

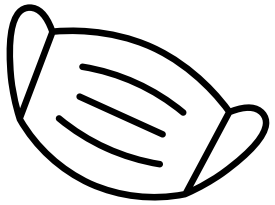
- LUCAS LAGE, CpE
- FERNANDO VALDES-RECIO, CpE
- J. ANTON STRICKLAND, CpE
- KAYLA FREUDENBERGER, EE

INTRODUCTION

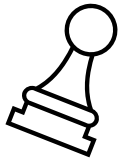
The Arcane Game Board

- Chess Board that moves pieces “magically” without human intervention
- User interacts with the game via web application
- Includes piece “graveyard” for captured pieces
- Unique and fun spin on the centuries-old game

MOTIVATION



- Create a system for long-distance physical board games
  - Physical board with benefits of online gameplay
- Chess is well-known and widely accessible
- Inspired by scenes from media such as *Harry Potter and the Sorcerers Stone*
- Create a fun, new way to play

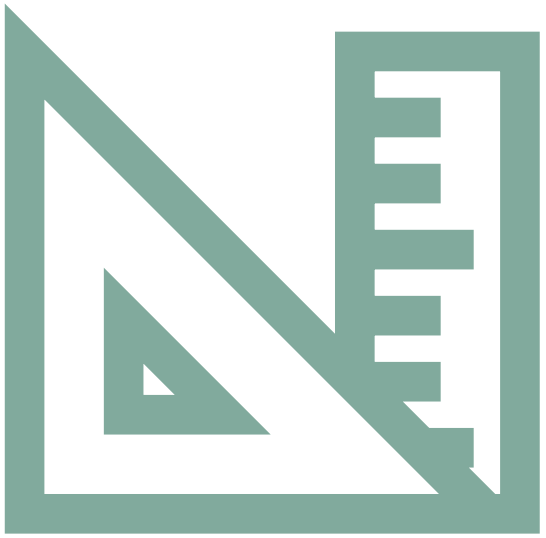


GOALS AND OBJECTIVES

- Accurate and timely piece movement
  - Precise movement to avoid collision
- Operate without human intervention
  - Web application interfaces with game board
- User Friendly
  - User Interface
  - Board setup
- Seamless and enjoyable gameplay experience

SPECIFICATIONS

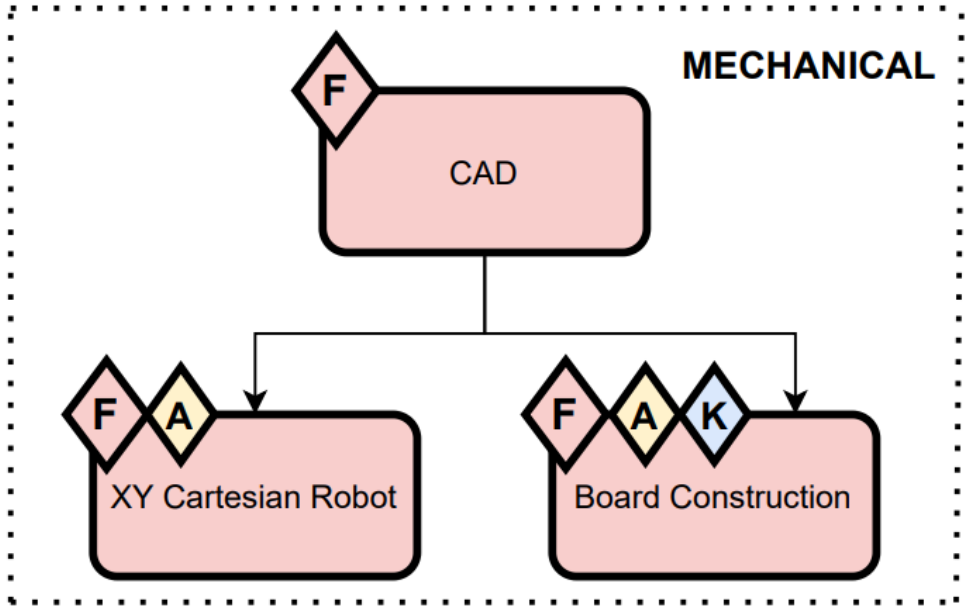
Arcane Game Board Specifications	
Size	<=24inx24in
Weight	<20lbs
Cost	<$500
Responsiveness	~2secs
Duration of piece movement	<=15secs
Range of movement	64 Tiles (1”x1”)
Input Voltage	12VDC



PROJECT DESIGN APPROACH

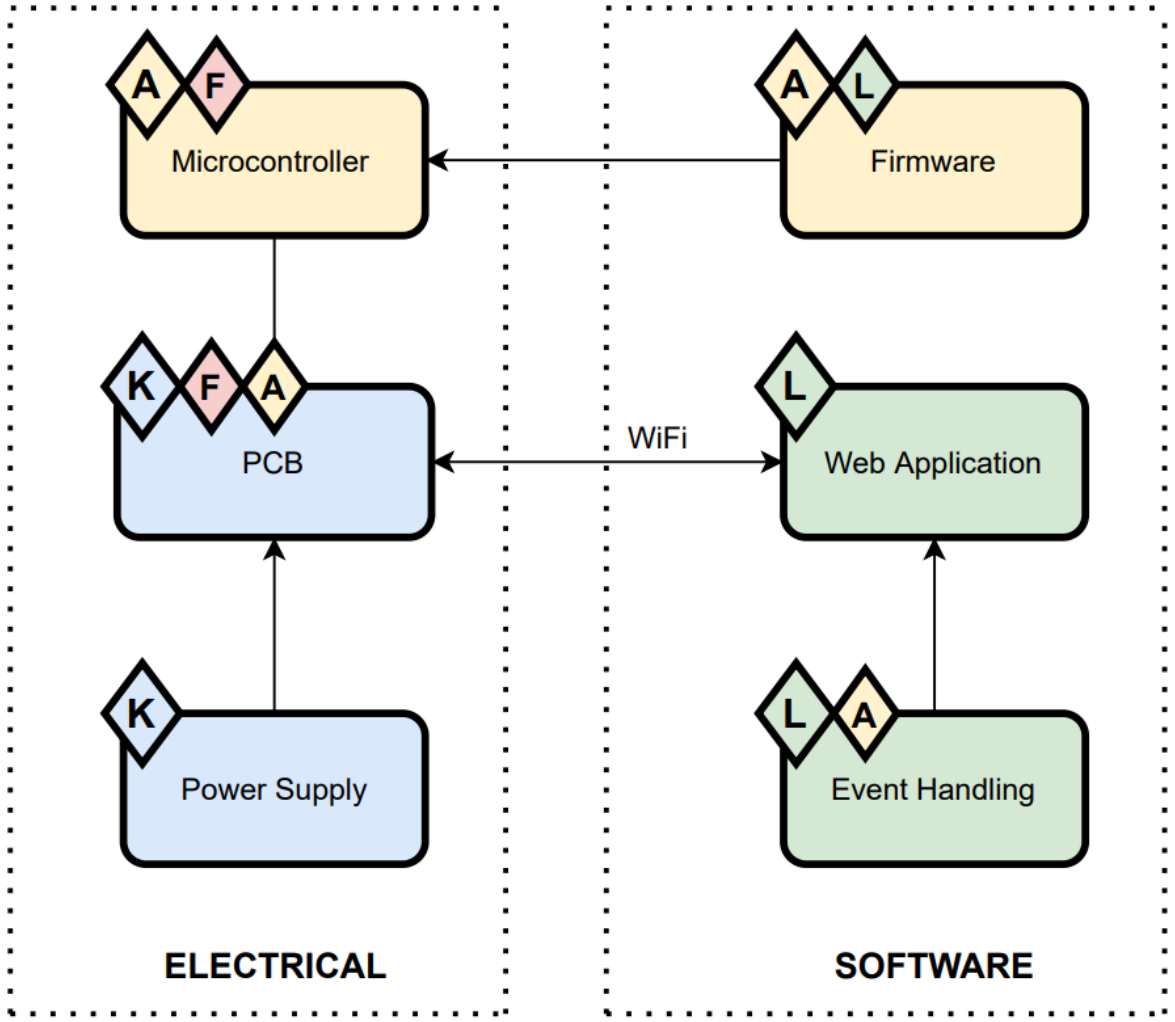
TEAM ROLES, BLOCK DIAGRAM, COMPONENT SELECTION

PROJECT ROLES

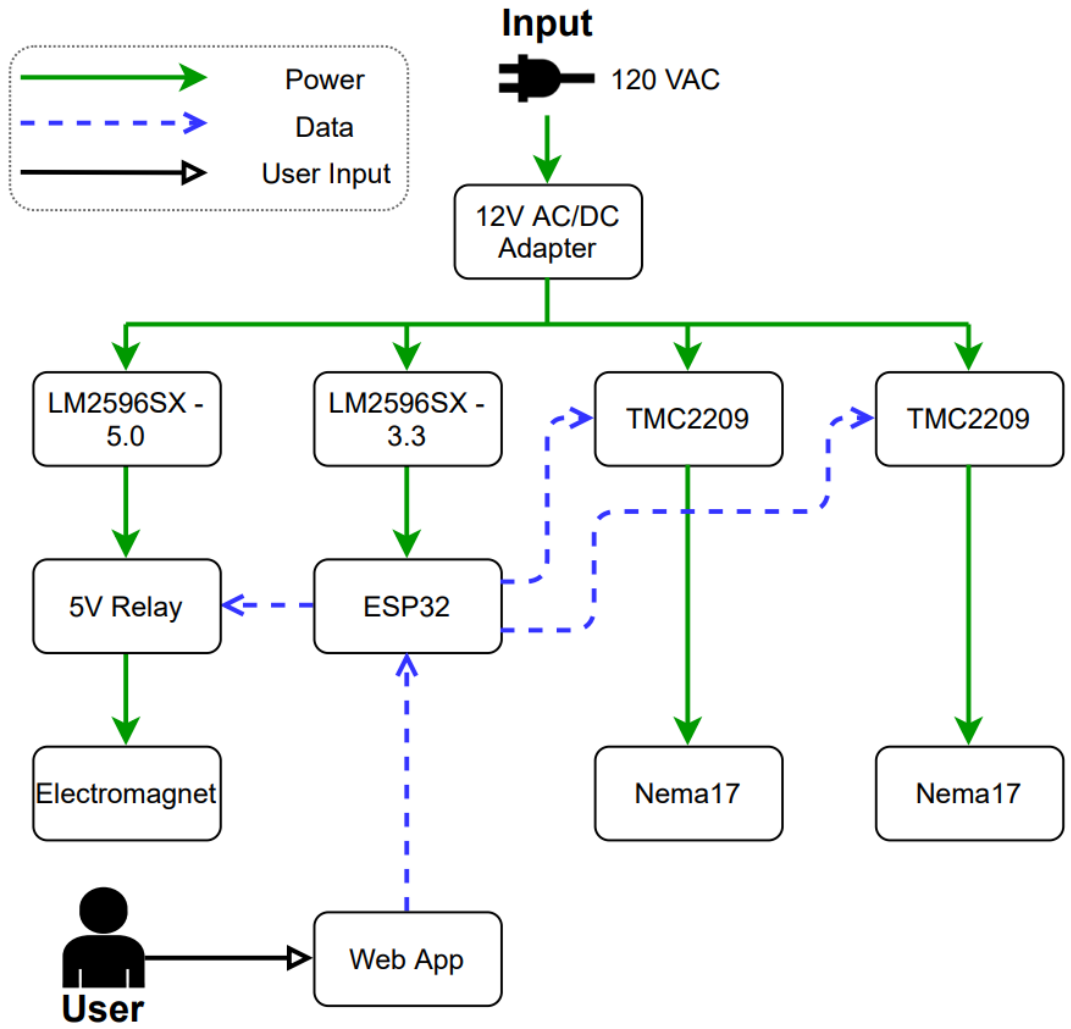


AREA LEADS

- F — Fernando Valdes-Recio
- K — Kayla Freudenberger
- L — Lucas Lage
- A — Anton Strickland

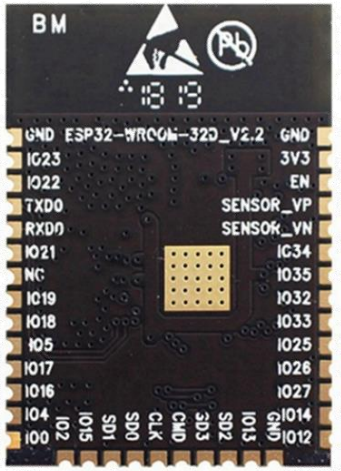


HARDWARE BLOCK DIAGRAM



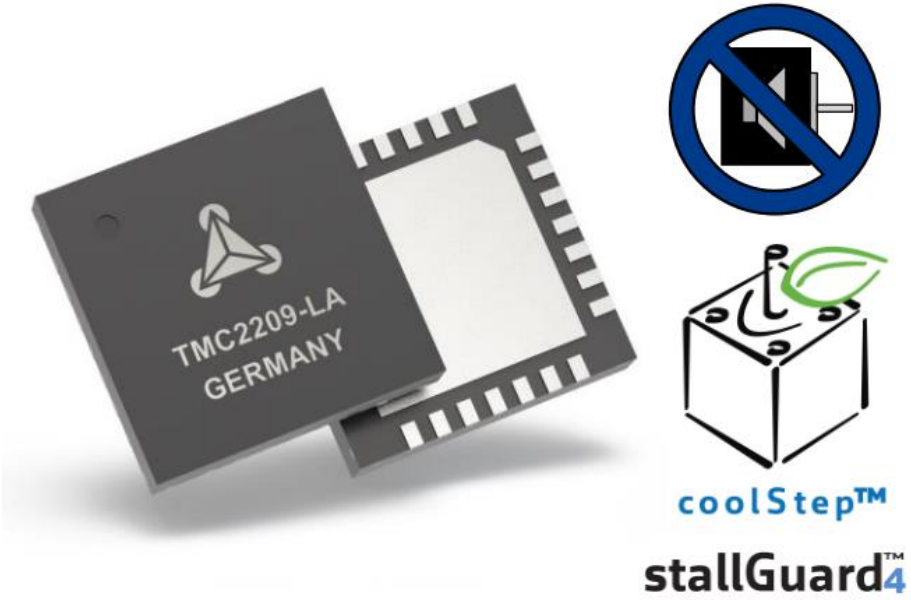
ESP32-WROOM-32D

- 240MHz Core Clock
- 4MB Flash
- Dual Core
  - 32-bit LX6 microprocessor
  - 160 MHz
- Integrated Blu-Fi Module
  - Bluetooth 4.2
- Memory: 520 KiB SRAM



TMC2209 STEPPER MOTOR DRIVERS

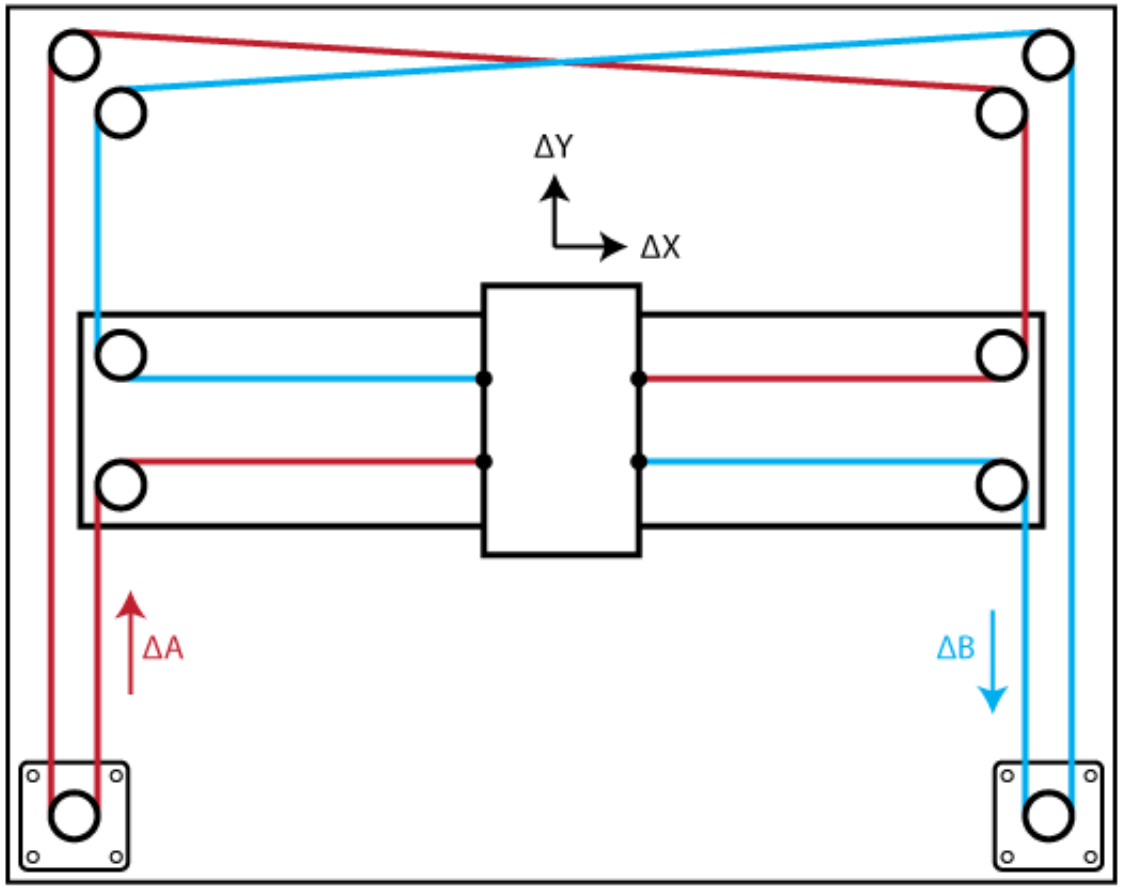
- Commonly found in 3d-Printers
- StallGuard™ feature allows Sensorless homing
- Configurable via UART for greater control
- StealthChop™ feature allows near-silent stepping



HARDWARE DESIGN

LEAD: FERNANDO VALDES-RECIO

X-Y CARTESIAN MOVEMENT

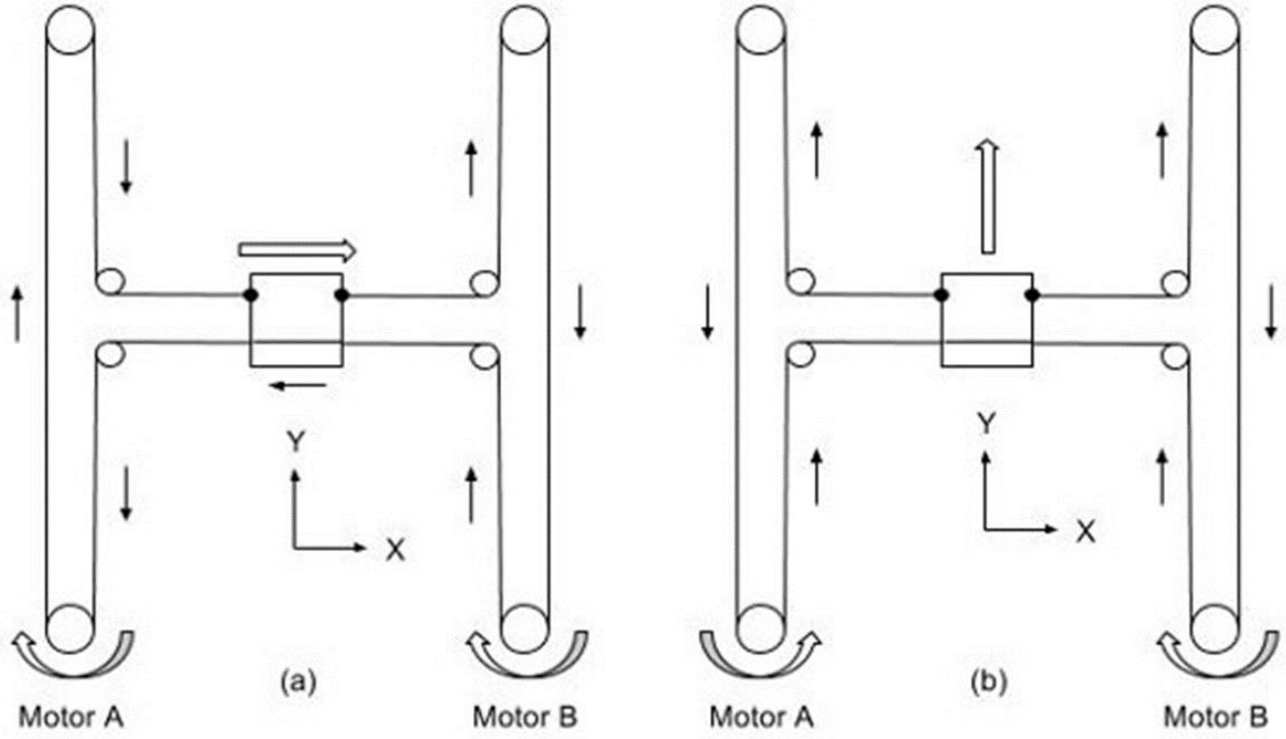


Equations of Motion:

$$\Delta X = {}^1/_2(\Delta A + \Delta B), \quad \Delta Y = {}^1/_2(\Delta A - \Delta B)$$

$$\Delta A = \Delta X + \Delta Y, \quad \Delta B = \Delta X - \Delta Y$$

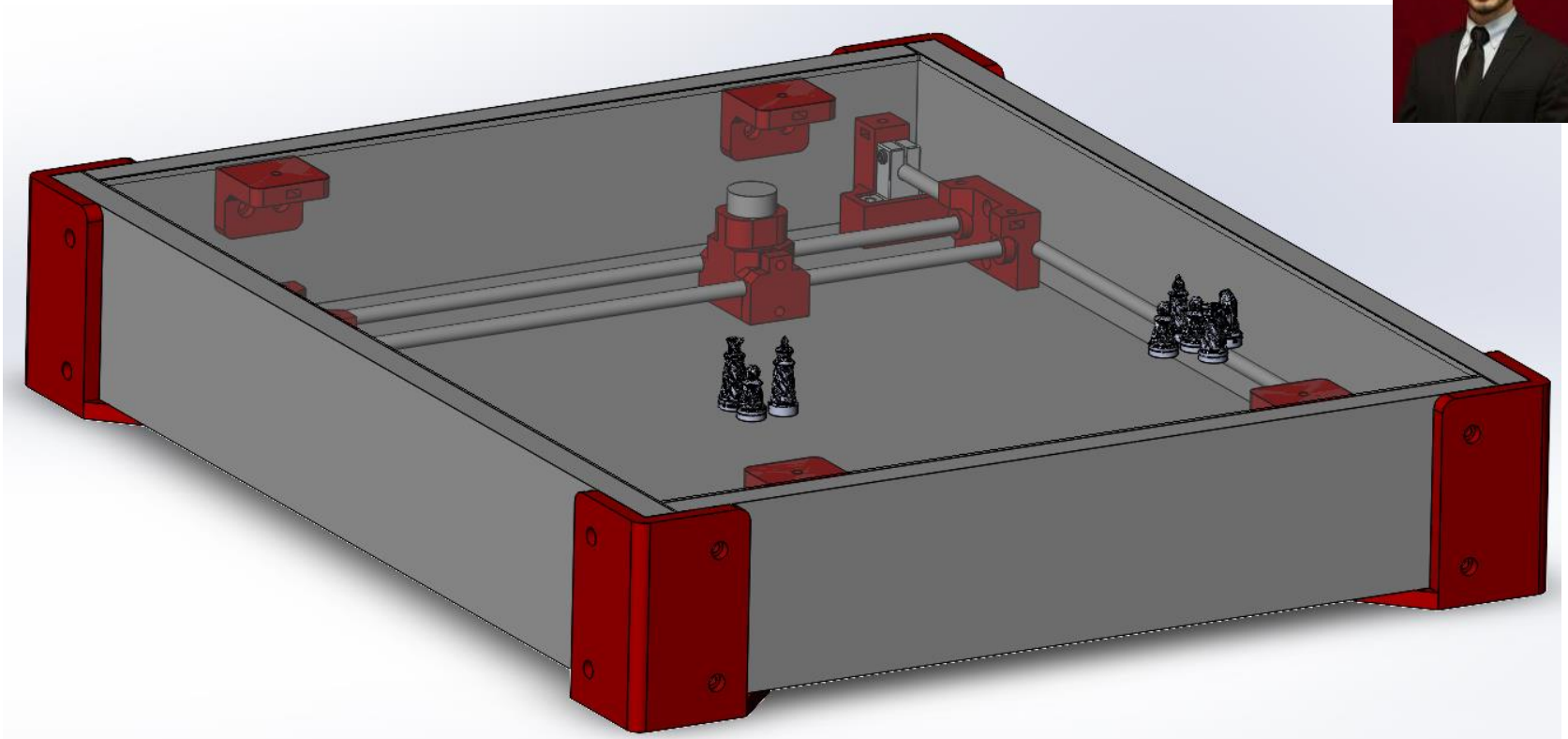
H-BOT

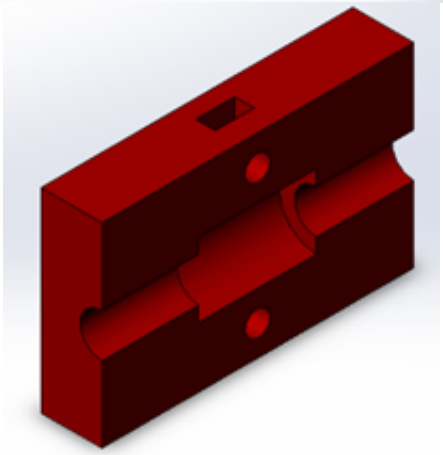


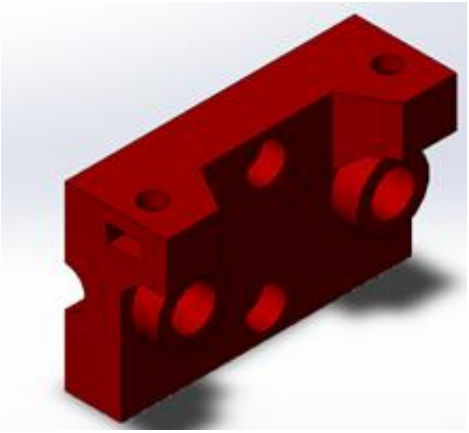
MOTORS

- Nema17
- Mass produced
- Open loop position control
- High positional accuracy

CAD







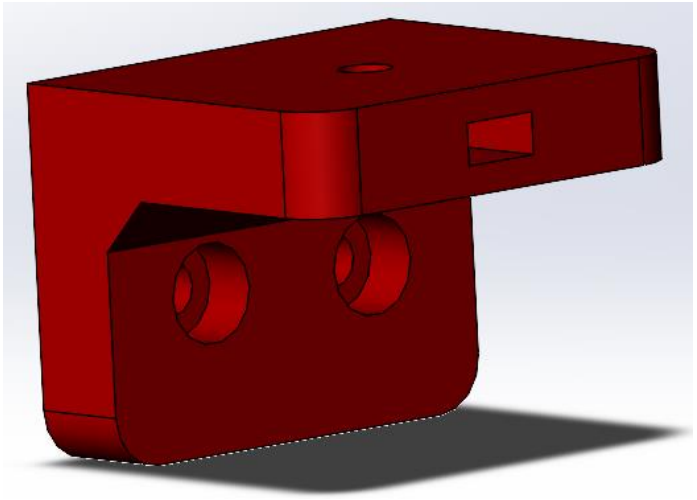
FINAL ASSEMBLY

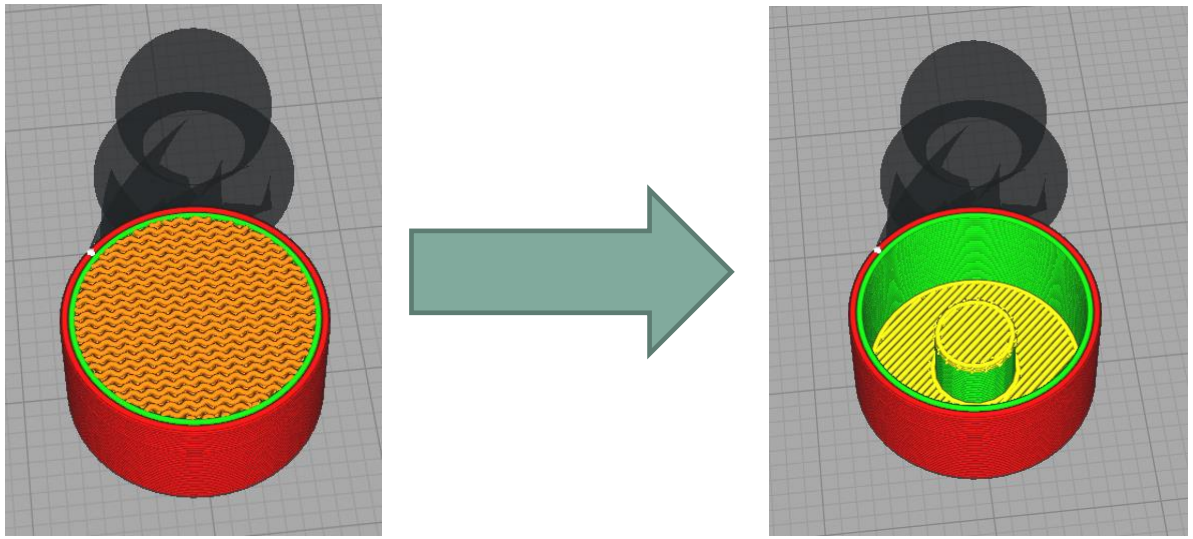
EARLY CHALLENGES

- X-axis wobble when traveling along X (vastly improved)
- Limited metric hardware (mostly acquired)

LATE CHALLENGES

- Acrylic sag (Solved with new mounting brackets)
- Pieces not sliding (Solved by hollowing out 3D prints)
- Inaccurate movement (Solved with belt tension and adjusting motor current)

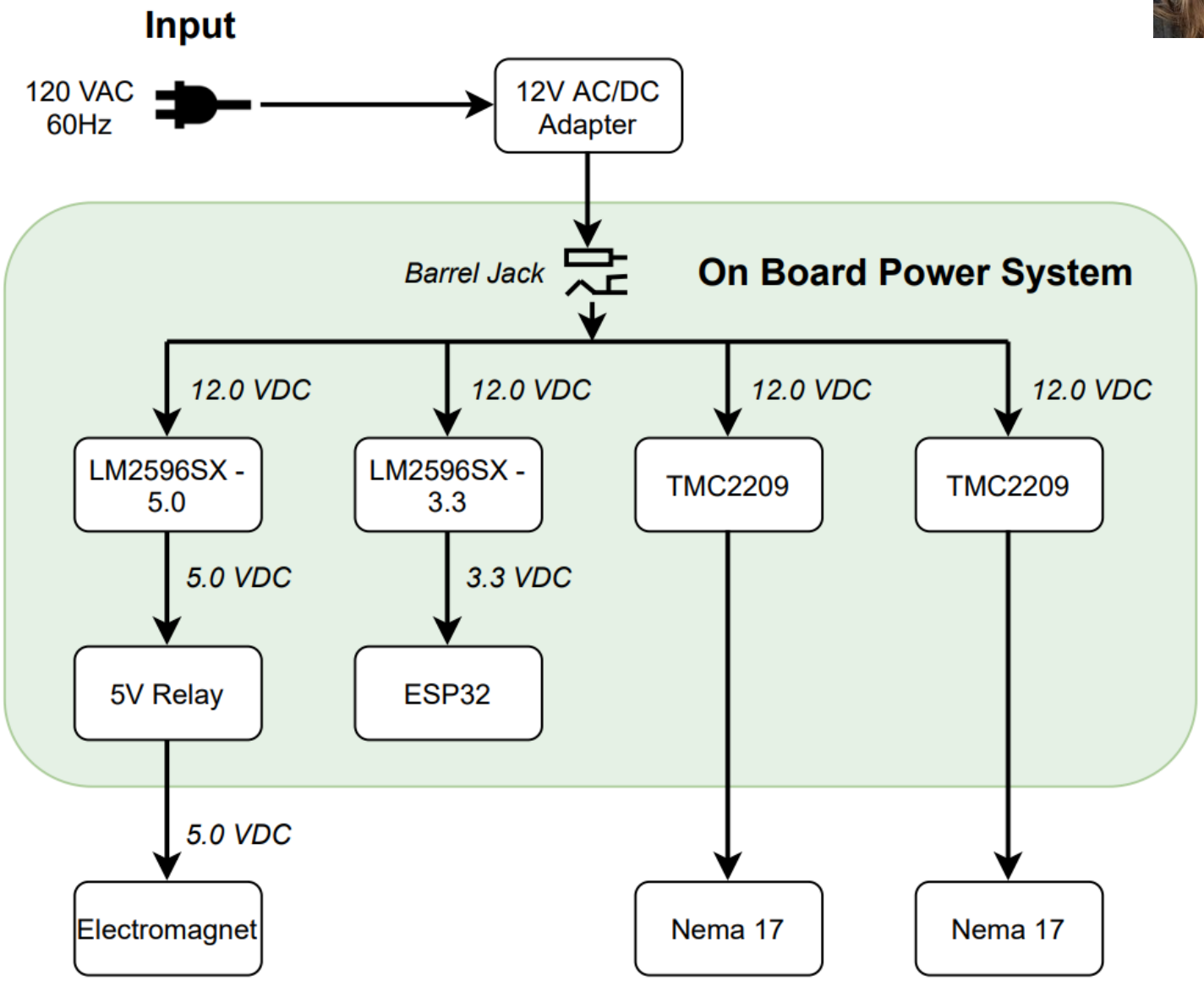




ELECTRICAL DESIGN

LEAD: KAYLA FREUDENBERGER

POWER SYSTEM FLOWCHART



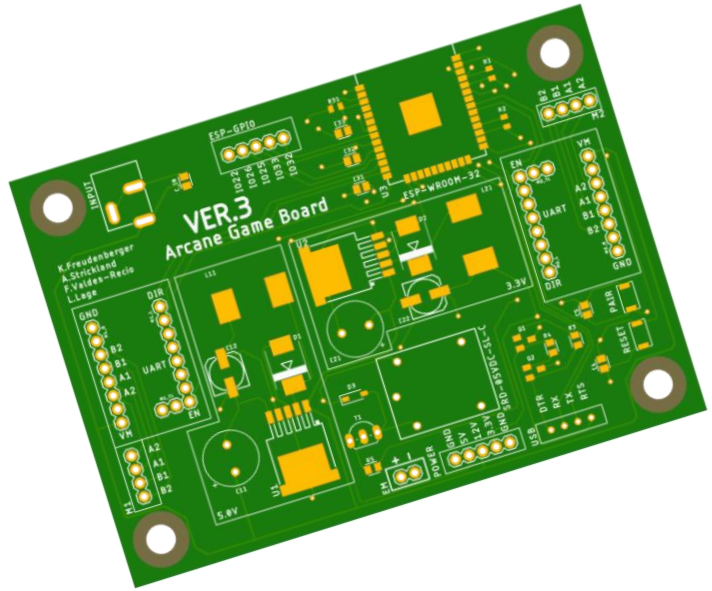
VOLTAGE REGULATOR

- 2 Voltage Regulators
  - ESP32 – 3.3V, 500mA
  - Electromagnet – 5.0V, 300mA
- LM2596SX Switching Step-Down Voltage Regulator
- 3.3V and 5.0V Fixed Output
- High Efficiency (Avg. 76%)

Voltage Regulator Datasheet Information						
IC	*Manufacturer*	*Vin (Min-Max)*	*Iout (Max)*	*Pins*	*Size (mm)*	*External Parts*
LM2596	Texas Instruments	4.5V – 40V	3A	5	14.85x10.75x5	4

PCB DESIGN

AUTODESK®
FUSION 360™

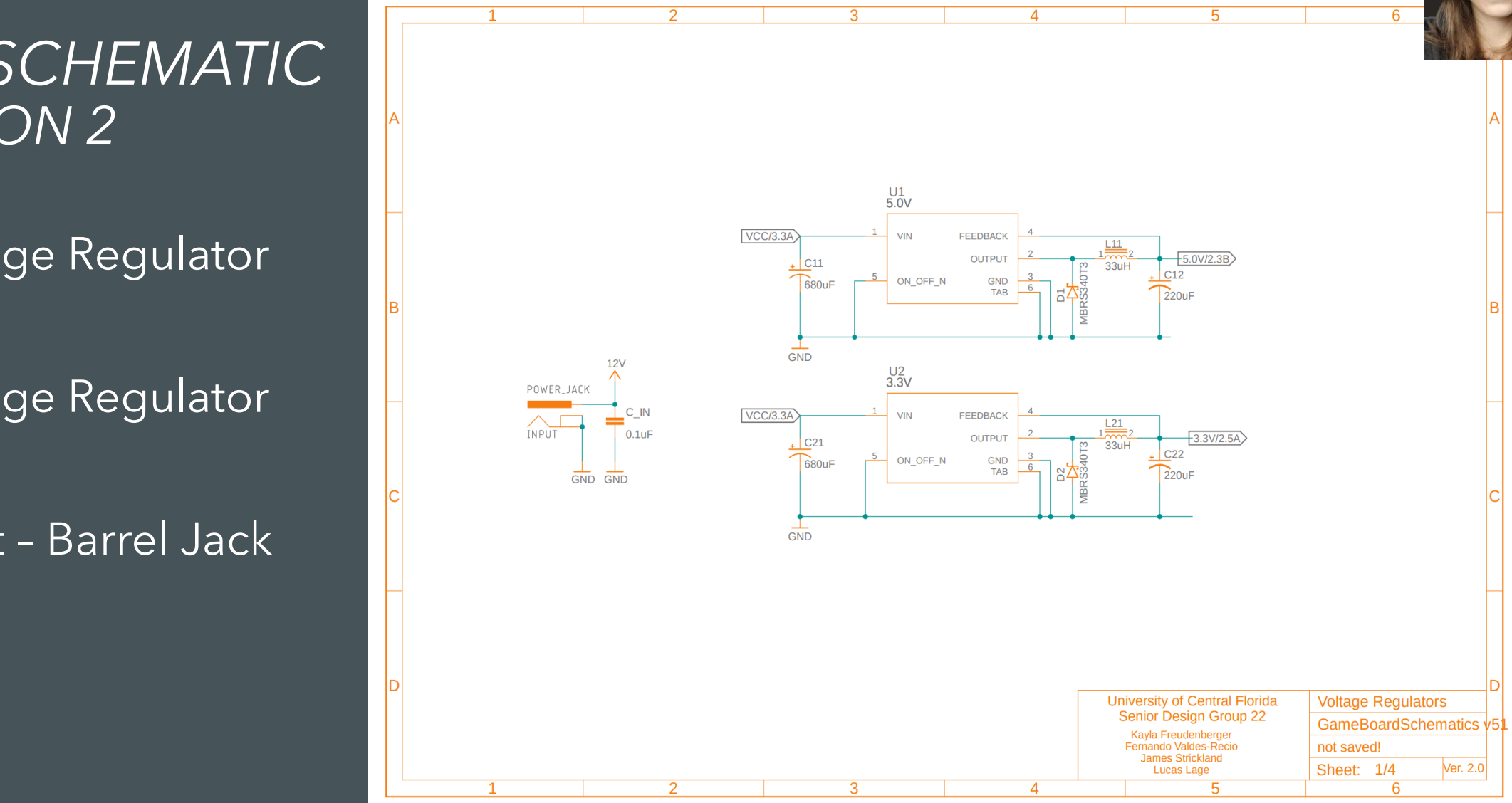


- PCB Design Software: Fusion 360
- Manufacturing: JLCPCB
- Status: Fully tested and functioning
- Final Revision: VER. 3

J@LC JLCPCB

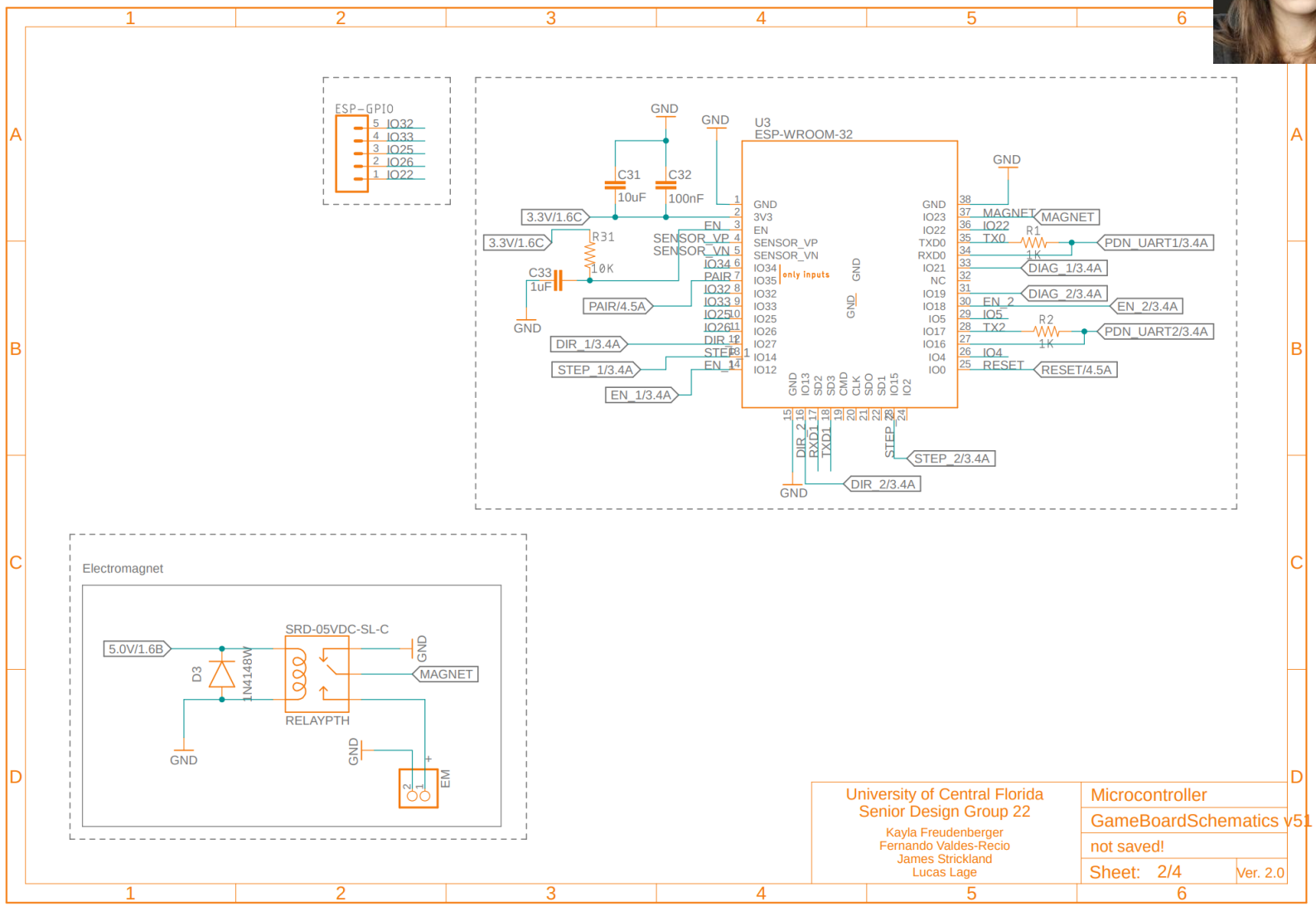
PCB SCHEMATIC VERSION 2

- Voltage Regulator 5.0V
- Voltage Regulator 3.3V
- Input – Barrel Jack



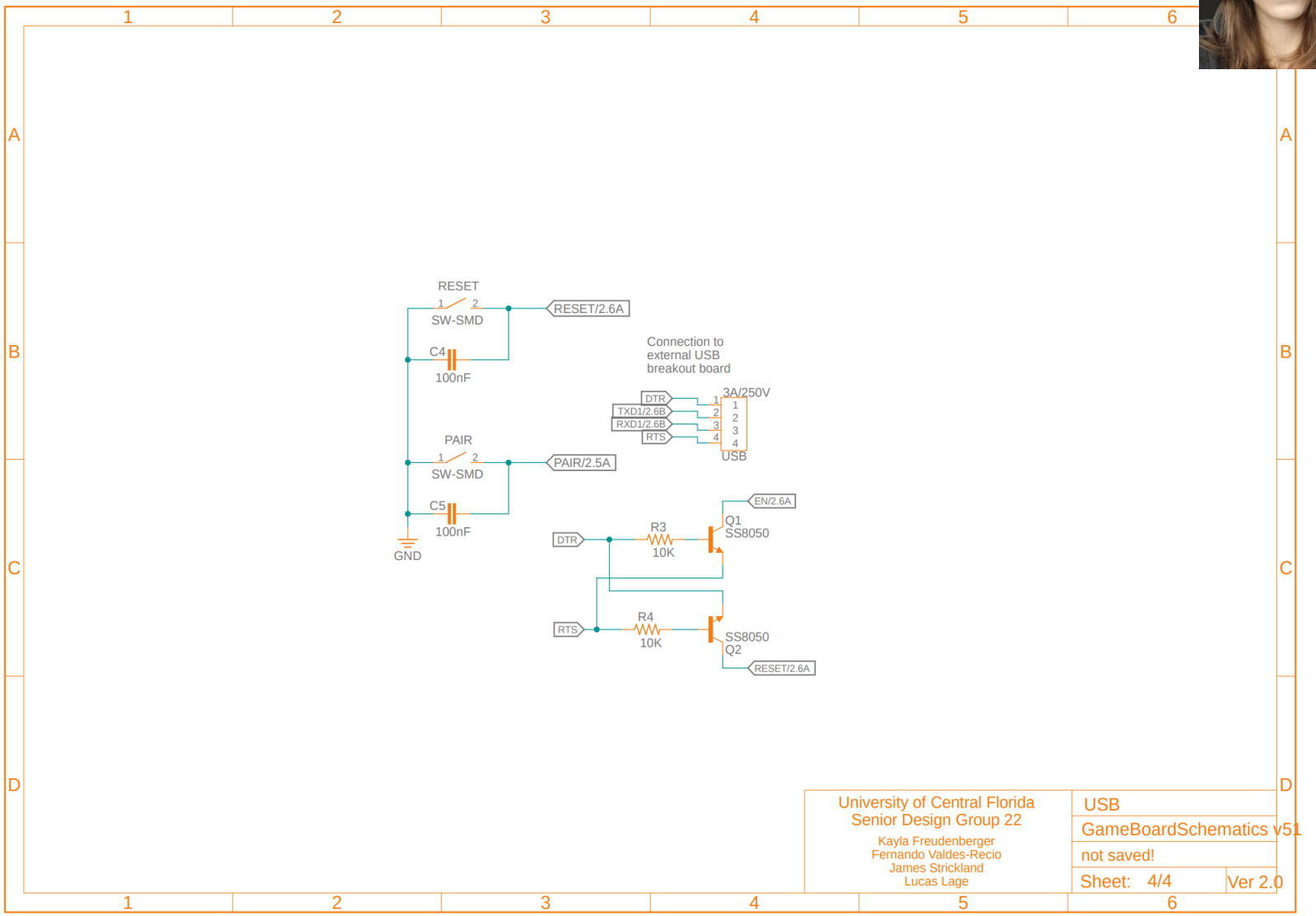
PCB SCHEMATIC VERSION 2

- ESP32-WROOM-32D Microcontroller
- Relay - Electromagnet



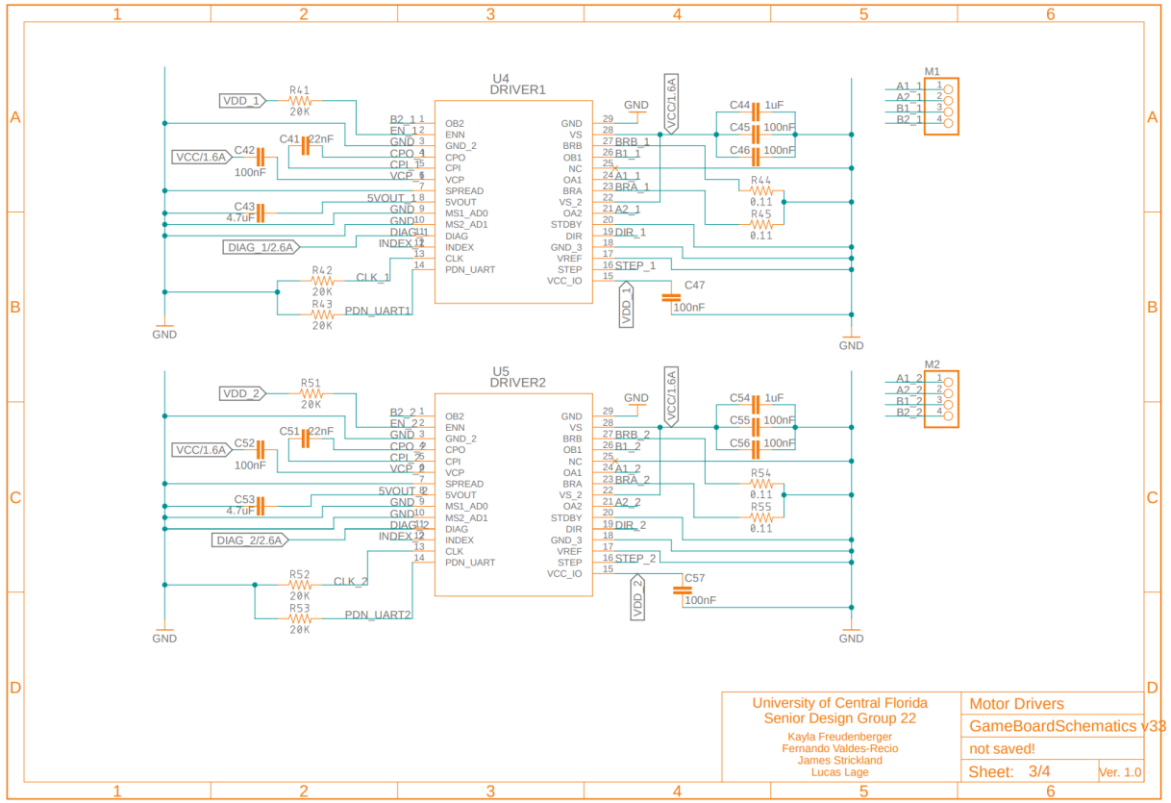
PCB SCHEMATIC VERSION 2

- USB Connection

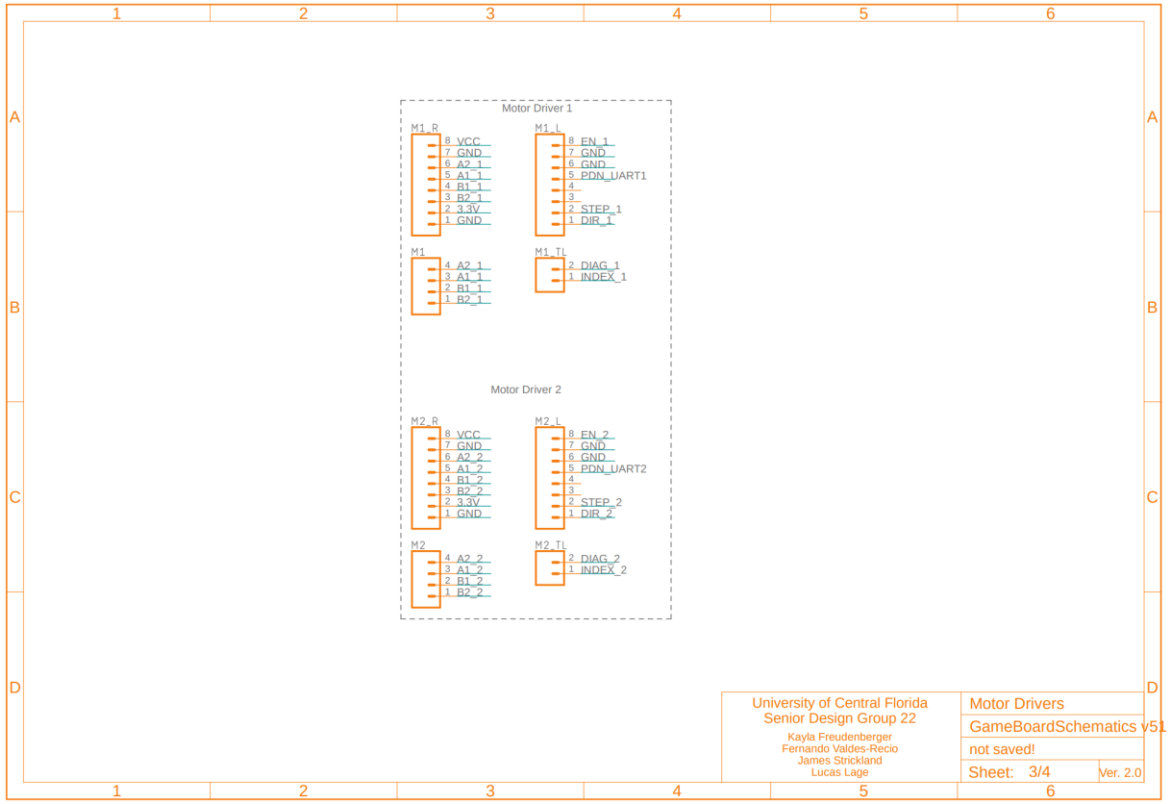


PCB SCHEMATIC – TMC2209 MOTOR DRIVERS

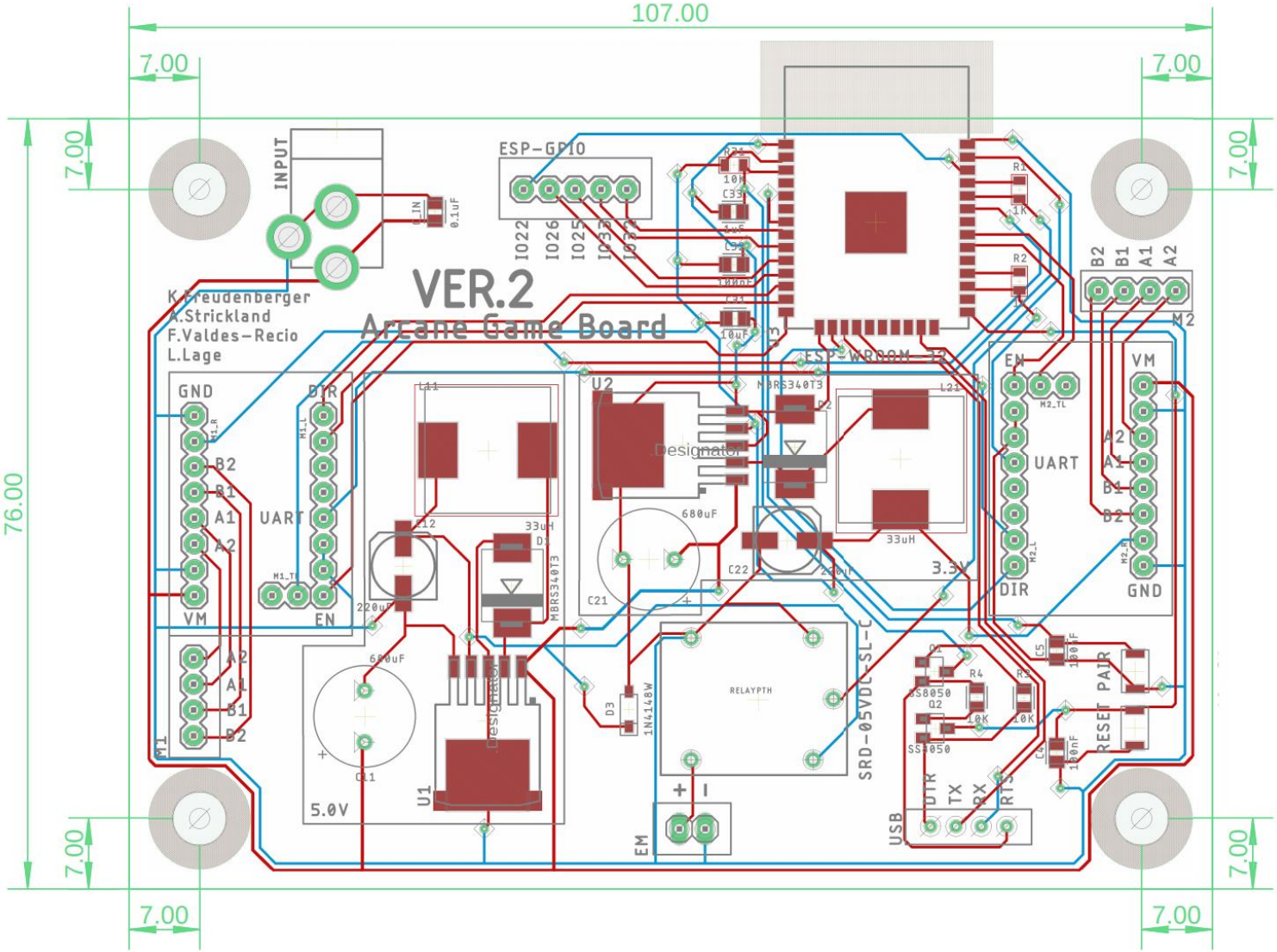
ORIGINAL (*Version 1.0*)



FINAL (*Version 2.0 & 3.0*)



PCB LAYOUT – VERSION 2



INITIAL TESTING – VERSION 2

Positive

Voltage Regulators

- Outputting correct voltage values

TMC2209 Motor Drivers

- Functioning properly

ESP32 Microcontroller

- Functioning properly

Negative

External USB Connection

- Couldn't flash firmware

Relay Circuit

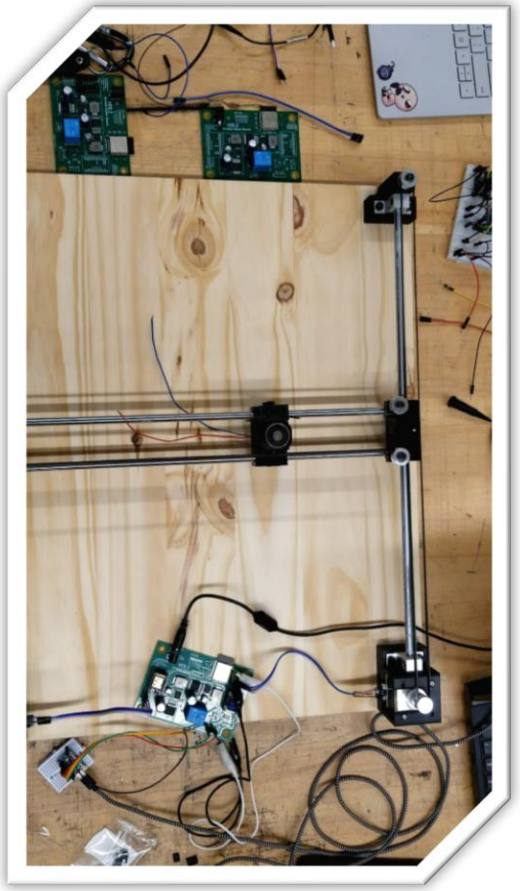
- ESP32 cannot output current necessary to toggle the relay

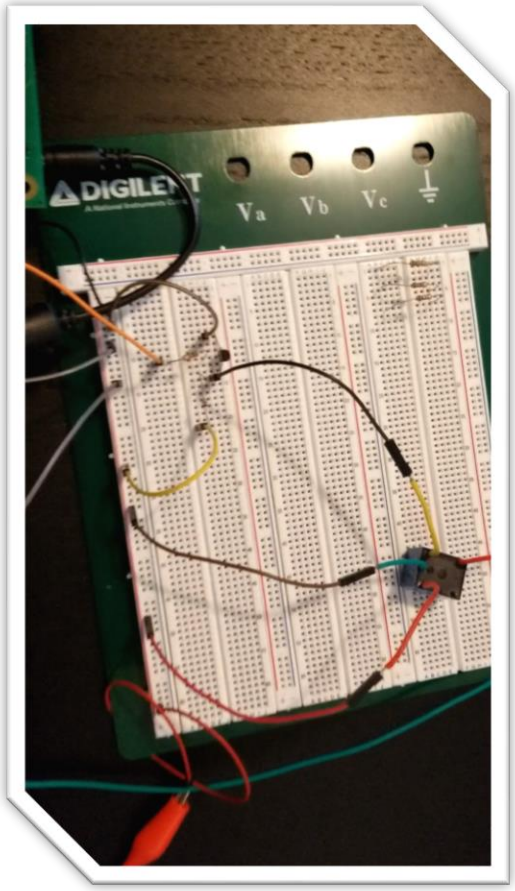
Pair/Reset Buttons

- Footprints differed from actual component size

SOLUTIONS

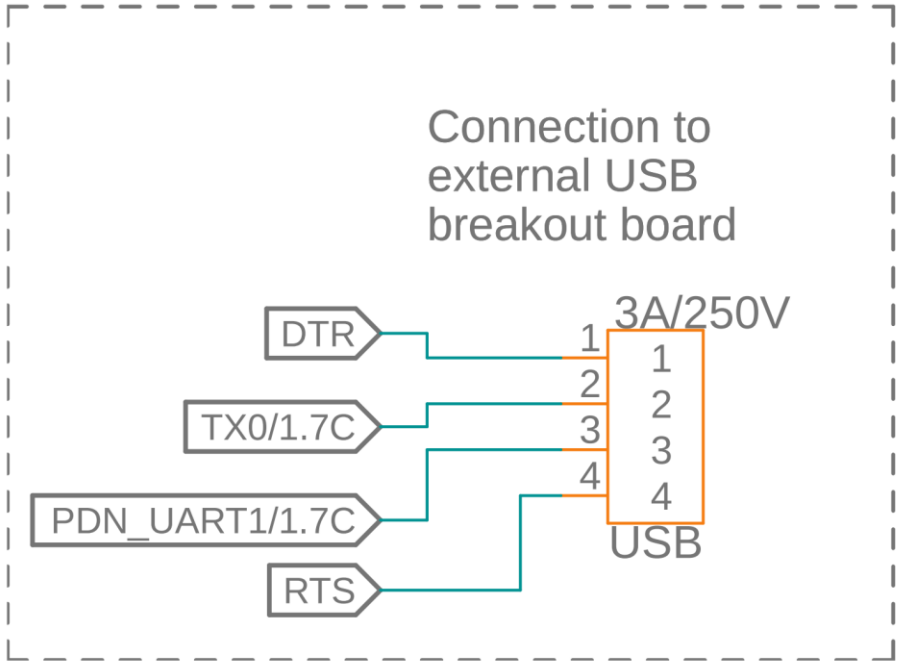
- External USB Connection
  - Jumped TX and RX on breakout board directly to RXD0 and TXD0
- Relay Circuit
  - Added an N-P-N transistor to amplify current coming from the ESP32
- Pair/Reset Buttons
  - Modified button footprints in Fusion 360 to fit datasheet dimensions



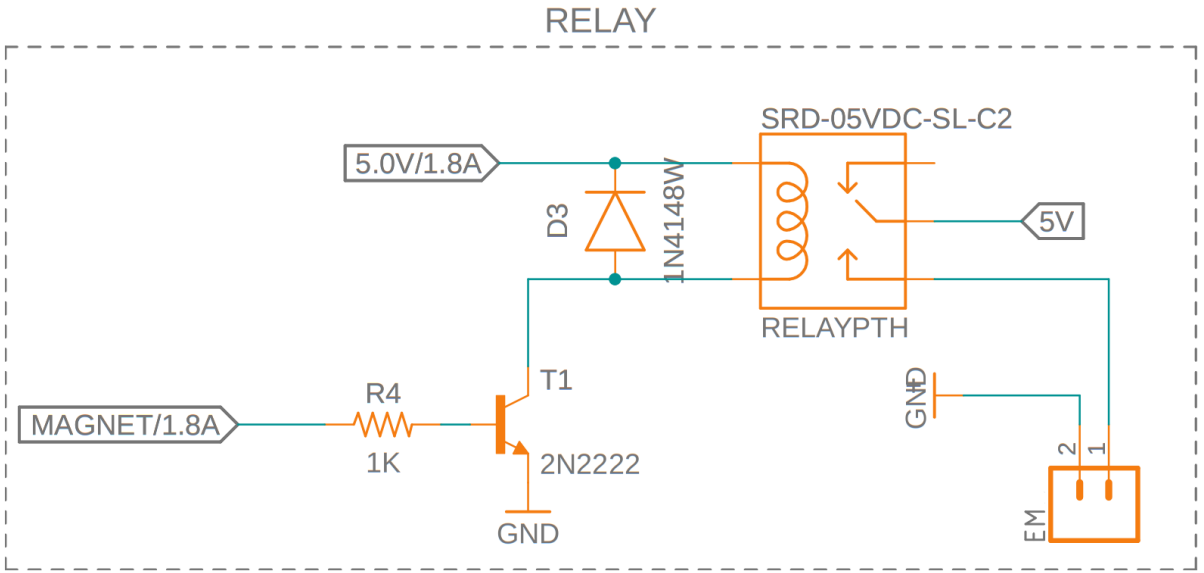


PCB SCHEMATIC – VERSION 3 CHANGES

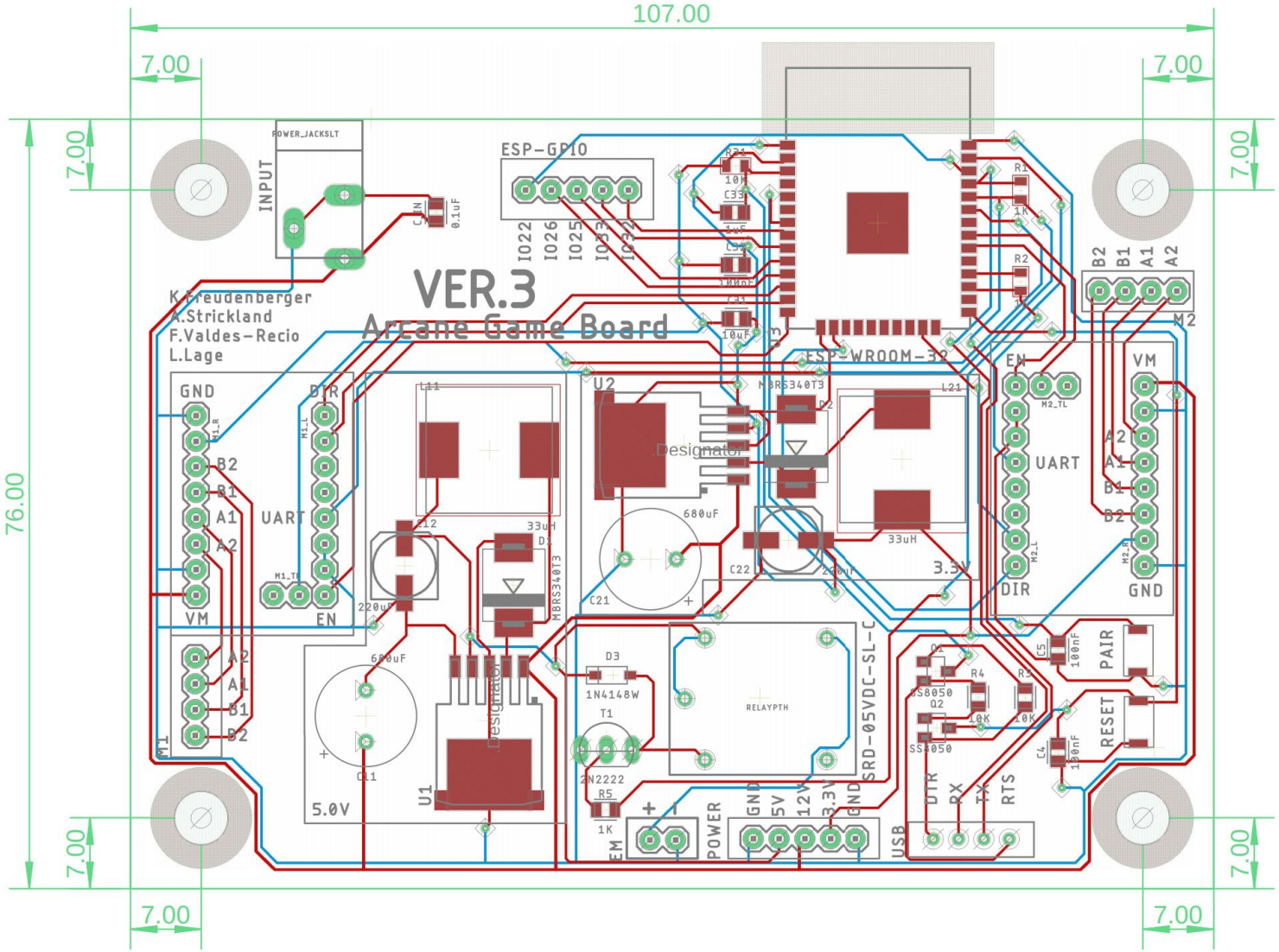
External USB Connection



Relay Circuit



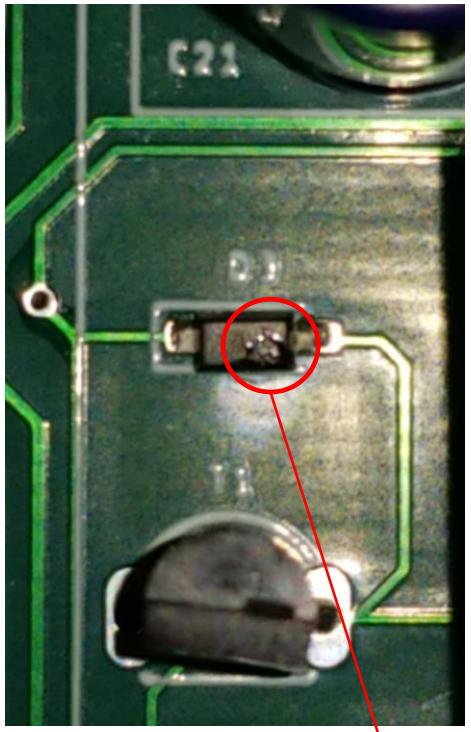
FINAL PCB LAYOUT – VERSION 3



INITIAL TESTING – VERSION 3

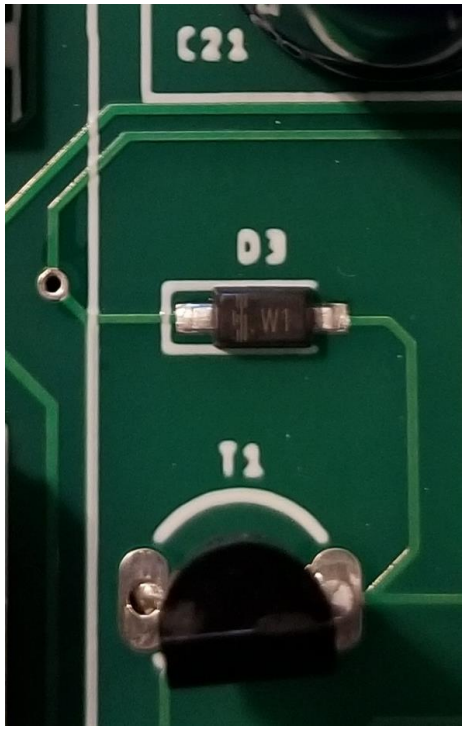
- Manufacturing Errors
  - Out of 5 total boards:
    - 1 had relay flyback diode soldered backwards - fixed
    - 1 had a faulty 5V Regulator – nonfunctional
- Footprint Inaccuracy
  - Relay footprint swapped normally closed and normally open pins

Incorrect Orientation



DAMAGED

Correct Orientation



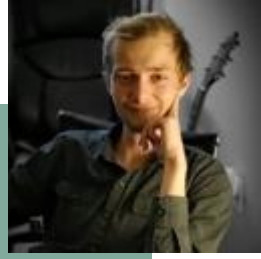
FINAL TESTING

Voltage	Theoretical (V)	Result (V)	Status
Power Supply	12	12.17	Pass
3.3V Regluator	3.3	3.238	Pass
5.0V Regulator	5.0	4.996	Pass
Relay Output	5.0(low) - 0.0(high)	4.93(low) - 0.004(high)	Pass

Component Functionality	Description	Status
ESP32 – WROOM – 32D	Firmware Flashing	Pass
	Communication with TMC2209 – Motor Driver 1	Pass
	Communication with TMC2209 – Motor Driver 2	Pass
	Toggle Relay: Low(on) High(off)	Pass
TMC2209 Motor Driver 1	Drive Motor 1 (M1)	Pass
TMC2209 Motor Driver 2	Driver Motor 2 (M2)	Pass

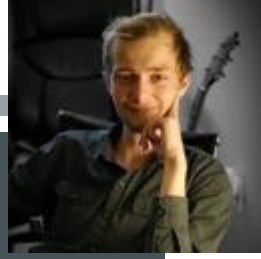
CHALLENGES

- Part placement and size requirements
  - ESP32 placement on board
  - Altered layout due to voltage regulator capacitor size differences
- Component Footprints
  - Non-standard power inductor
  - Out-of-date footprints for relay
- Component availability
  - TMC2209 Motor Drivers out of stock

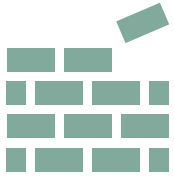


FIRMWARE DESIGN

LEAD: J. ANTON STRICKLAND



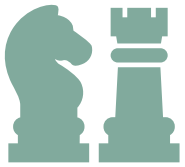
OVERVIEW



Structure

Lightweight

Flexible OOP Design



Class Overview

Controllers

Data Structures

Challenges

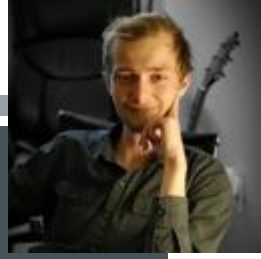
Limitations of Frameworks

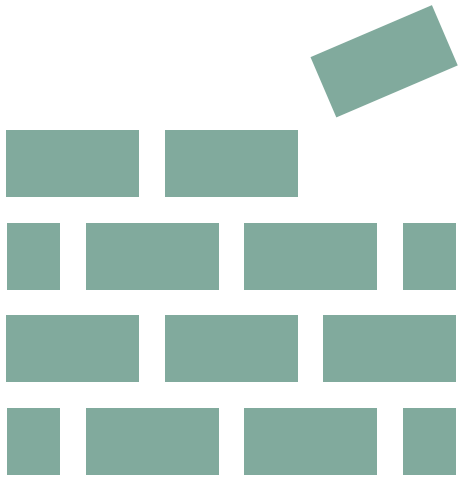
State Logic

Unit Testing

Unity Framework

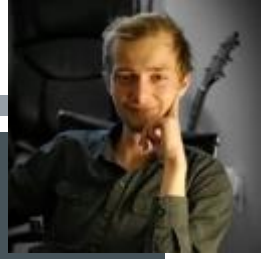
Methodology





STRUCTURAL OVERVIEW

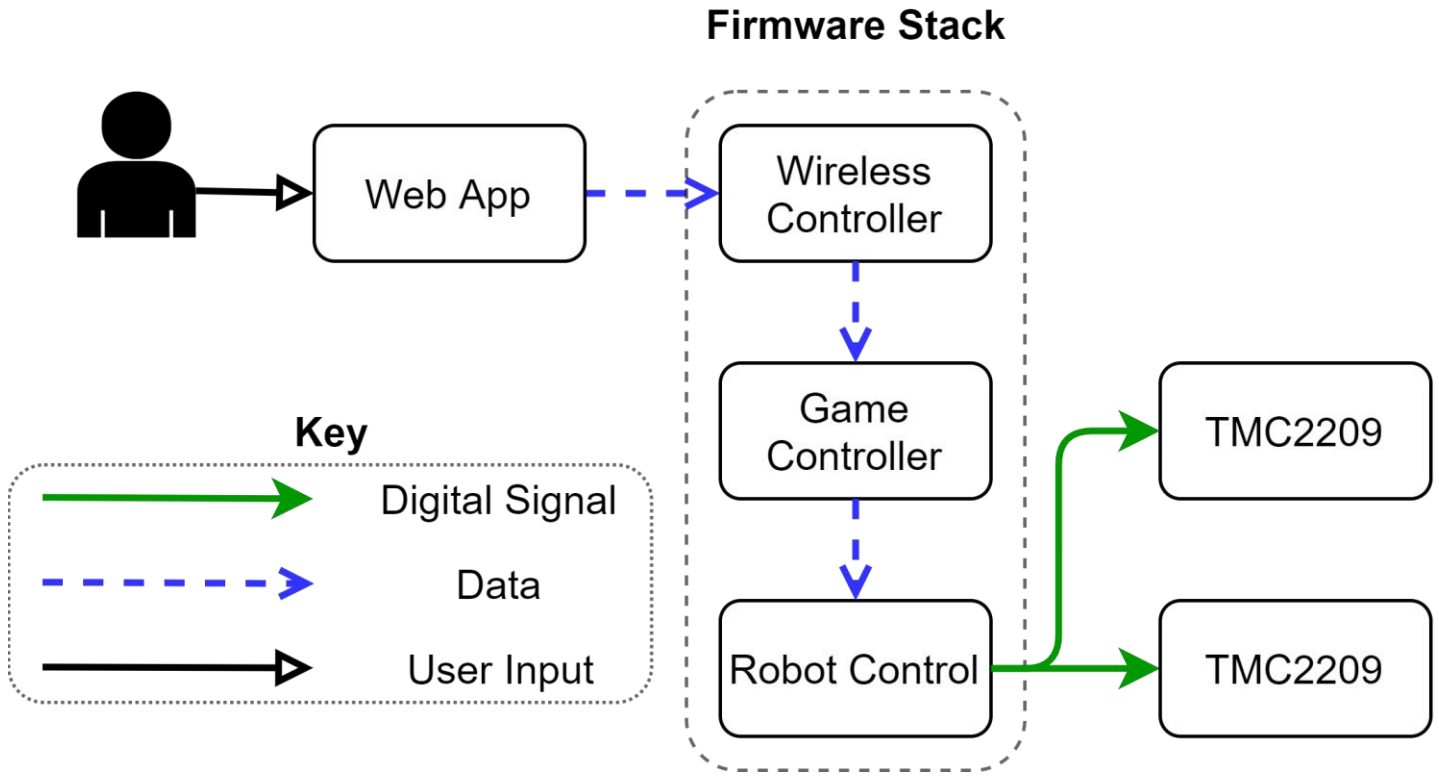
- Broken into 3 Layers
  - Wireless I/O
  - Movement Logic and Planning
  - Hardware Interface
- OOP Design
  - Objects and headers for each function category
  - ArcaneCore.hpp allows for games to be instantiated quickly using flexible code base

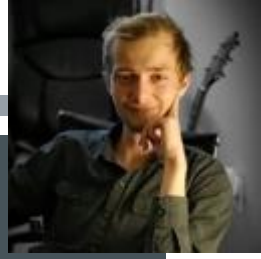


FIRMWARE FLOW DIAGRAM

Notes:

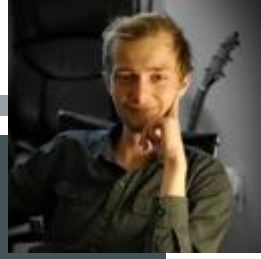
- All classes pass either C++ standard objects or objects of our own design
- Incoming Json files are deconstructed into structs, rather than saved as full strings.





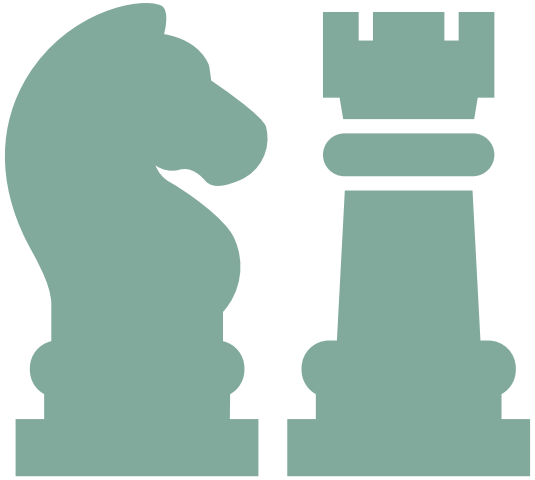
CLASS OVERVIEW – WIRELESS CONTROLLER

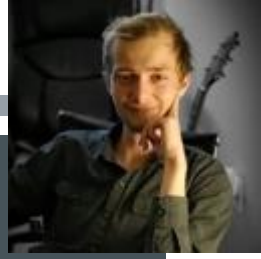
- Responsible for WiFi and Websocket Connection to Web-App
  - Uses Arduino Framework and Libraries
- Converts JSON files into structs containing:
  - Start Position [std::array<int8, 2>]
  - End Position [std::array<int8, 2>]
  - Special Flag [std::string]



CLASS OVERVIEW – GAME CONTROLLER

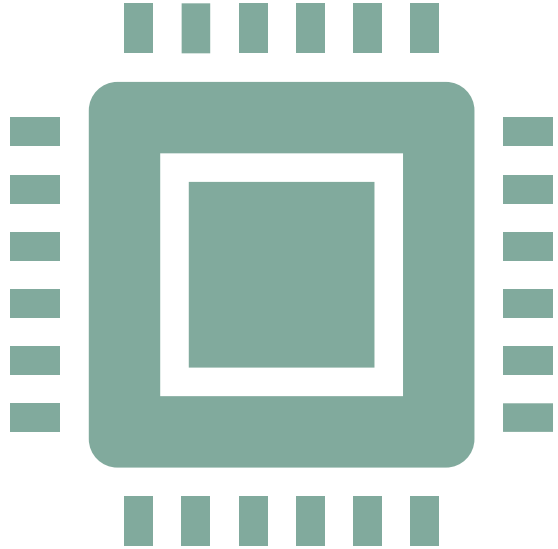
- Responsible for generating Piece paths and maintaining a digital record
  - Contains a master First-In-First-Out (FIFO) Queue to place moves in
  - Contains Piece objects
- Queue is polled by Robot Controller in ArcaneCore (main)

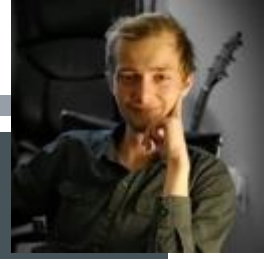




CLASS OVERVIEW – ROBOT CONTROLLER

- Robot Hardware and Software Interface
  - Sends Signals to TMC2209s
  - Loads Moves from internal Queue into a scheduling function
  - Contains code to allow UART control of stepper drivers with more advanced MCUs

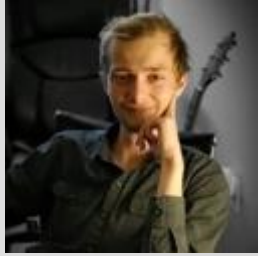




CHALLENGES

- Realities of Embedded System Frameworks
  - Blocking functions and out-of-site delays caused inconsistencies in memory addresses.
  - Watchdog timer resets were common with complex logic

- Path Planning
  - Avoidance Transposition was needed to ensure pieces didn't interfere with one another
  - Retirement needed each piece to have a target position easily set at instantiation without taking up memory.



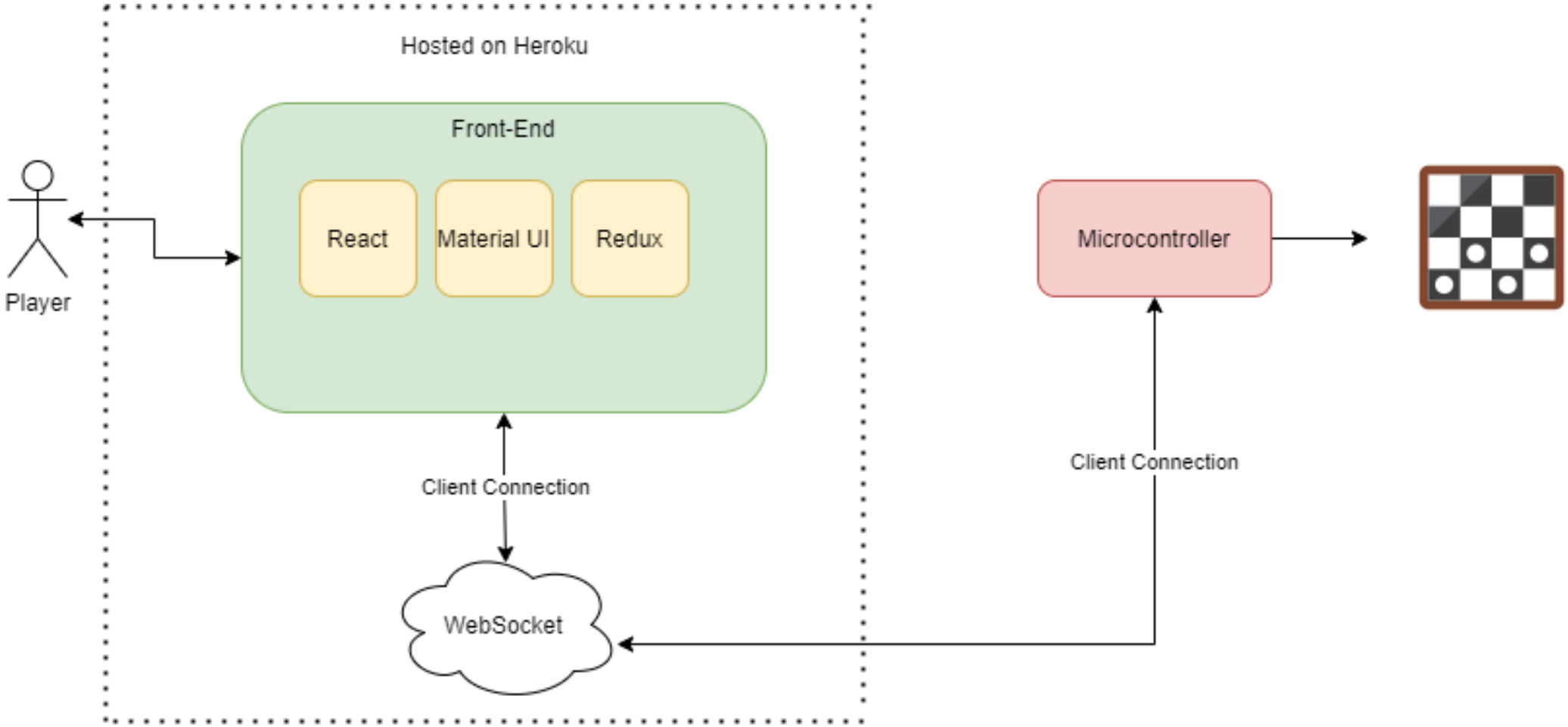
TESTING PROCEDURES

- Test-Driven Development was achieved using the Unity Embedded Testing Framework
  - Integrated with PlatformIO
  - Ran tests natively on the ESP32
  - Tests could be specified at execution, allowing for rapid turnarounds

SOFTWARE DESIGN

LEAD: LUCAS LAGE

SOFTWARE BLOCK DIAGRAM



OVERVIEW

Front-End

Tools, Frameworks, and Dataflow

and Dataflow

Back-End

Tools, Node.js,

and Design

WebSocket

Making connection, Real time messages, and client connections

Deployment

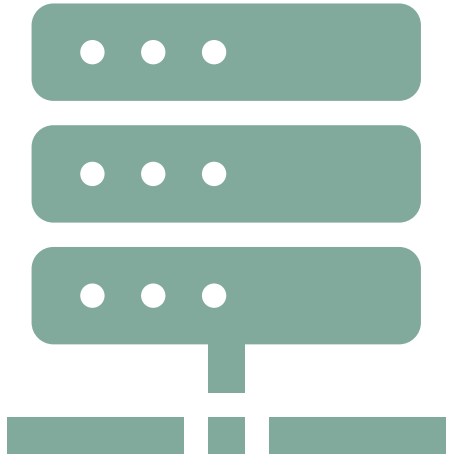
Secure connection, Responsive design, and Heroku

FRONT-END DESIGN

- Main Tools
  - React Framework
  - Material-UI
  - Redux
- Dataflow
  - User inputs
  - Local Storage
  - WebSocket Client Connection

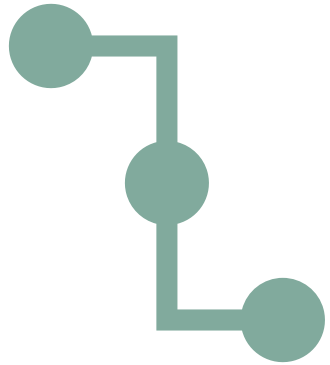
BACK-END DESIGN

- Tools
  - WebSocket
  - Node.js
  - Express.js
- Hosting
  - Heroku



WEBSOCKET CONNECTION

- Socket.IO
  - WebSocket Connection
  - Connecting Web App to Microcontroller
  - Utilizing data on the Front-End
- From moves to movements
  - Communication medians with firmware



DEPLOYMENT

- Front-end
  - Application is hosted
    - https://arcane-gameboard.herokuapp.com/
  - Can be downloaded as PWA
- Back-end
  - Node.js Server

ADMINISTRATIVE CONTENT

BUDGET AND BUILD PLANS

BUDGET

Item (#)	Quantity	Price	Total
400mm Linear Motion Rods	2	$7.19	$14.39
500mm Linear Motion Rods	2	$7.19	$14.39
LM8UU Linear Ball Bearings (*12*)	1	$10.95	$10.95
Electromagnet	1	$9.99	$9.99
Timing Belt + Pulley Wheel	1	$16.99	$16.99
Rail Clamps (*4*)	1	$11.99	$11.99
Nema 17 Stepper Motor (*3*)	1	$25.99	$25.99
PCB Version 2.0	1	$71.41	$71.41
PCB Version 3.0	1	$78.03	$78.03
12V AC/DC Adapter Plug	1	$9.99	$9.99
Threaded Inserts	1	$9.19	$9.19
Bushings	1	$5.99	$5.99
Acrylic Sheet	1	$24.99	$24.99
Wood	2	$14.99	$29.98
PLA Plastic Spool	1	$19.99	$19.99

BUDGET - CONTINUED

Item (#)	Quantity	Price	Total
Relay (*10*)	1	$7.86	$7.86
Felt	2	$1.89	$3.78
Header pins (*40*)	1	$5.29	$5.29
Barrel Jack (*10*)	1	$10.99	$10.99
Neodymium Magnets	1	$22.44	$22.44
		Total:	**$ 404.44**

QUESTIONS?