ARCANE GAME BOARD

GROUP 22

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

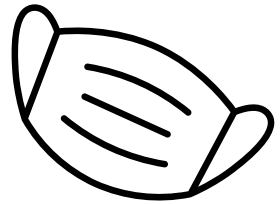
INTRODUCTION

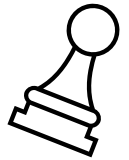
The Arcane Game Board

- 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



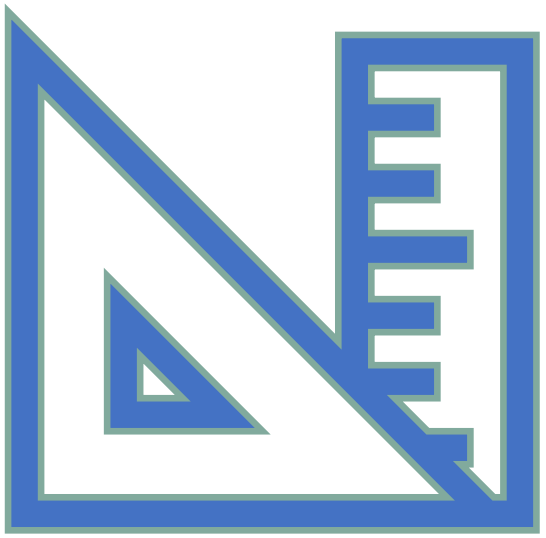


GOALS AND OBJECTIVES

- 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

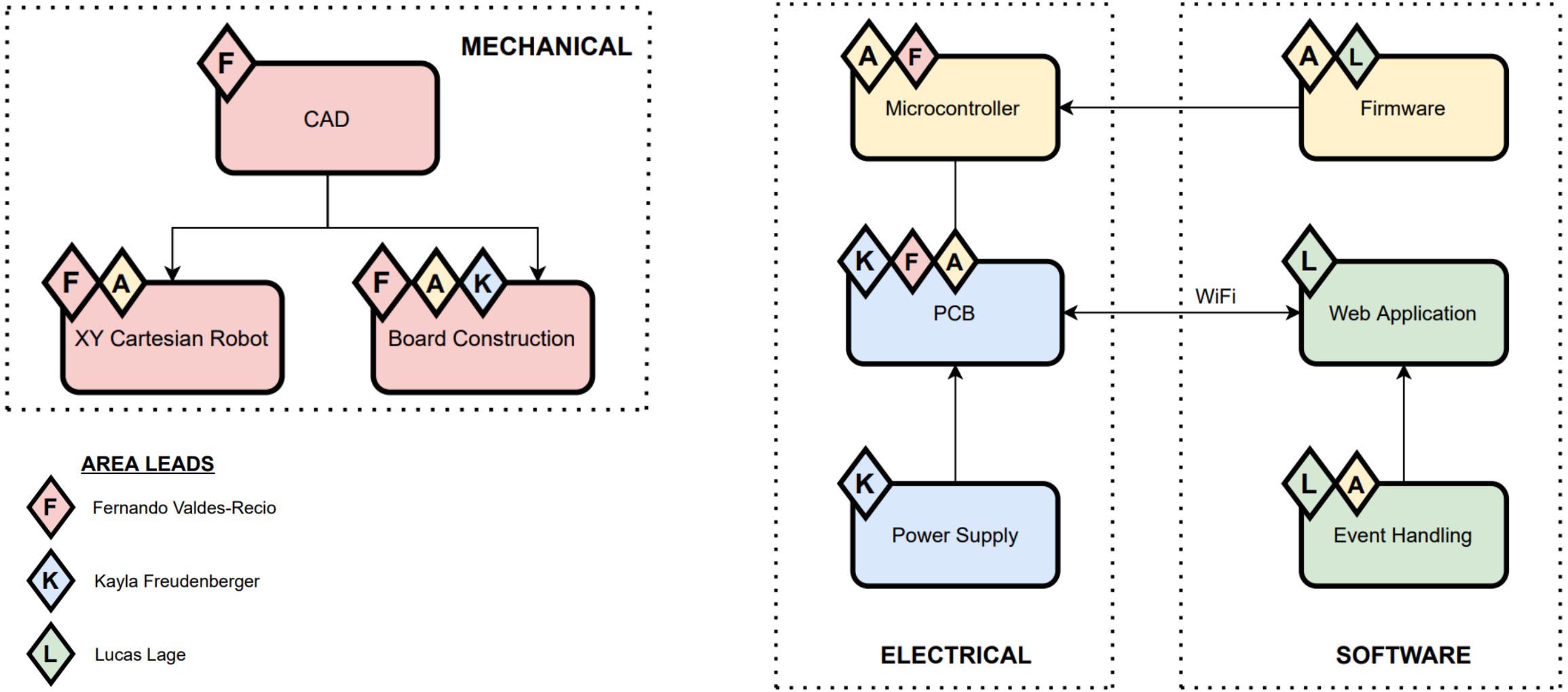
Arcane Game Board Specifications	
Size	<=24inx24in
Weight	<20lbs
Cost	<$500
Responsiveness	~2secs
Duration of piece movement	<=15secs
Range of movement	64 Tiles (1”x1”)
Input Voltage	12VDC



PROJECT DESIGN APPROACH

TEAM ROLES, BLOCK DIAGRAM, COMPONENT SELECTION

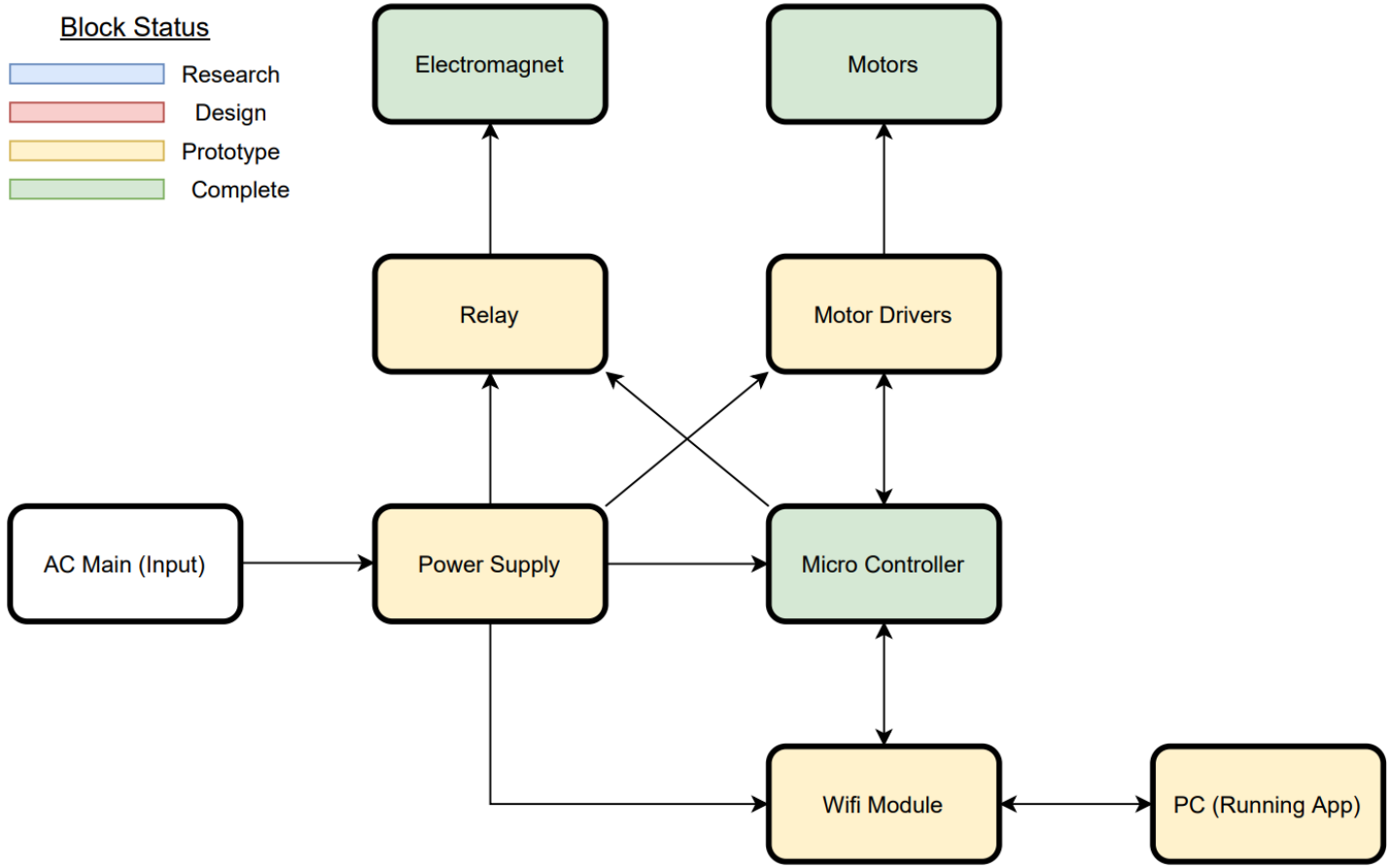
PROJECT ROLES



AREA LEADS

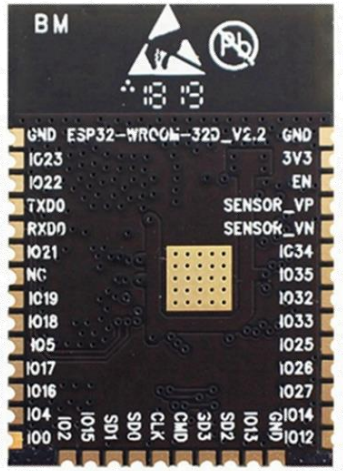
- F — Fernando Valdes-Recio
- K — Kayla Freudenberger
- L — Lucas Lage
- A — Anton Strickland

HARDWARE BLOCK DIAGRAM



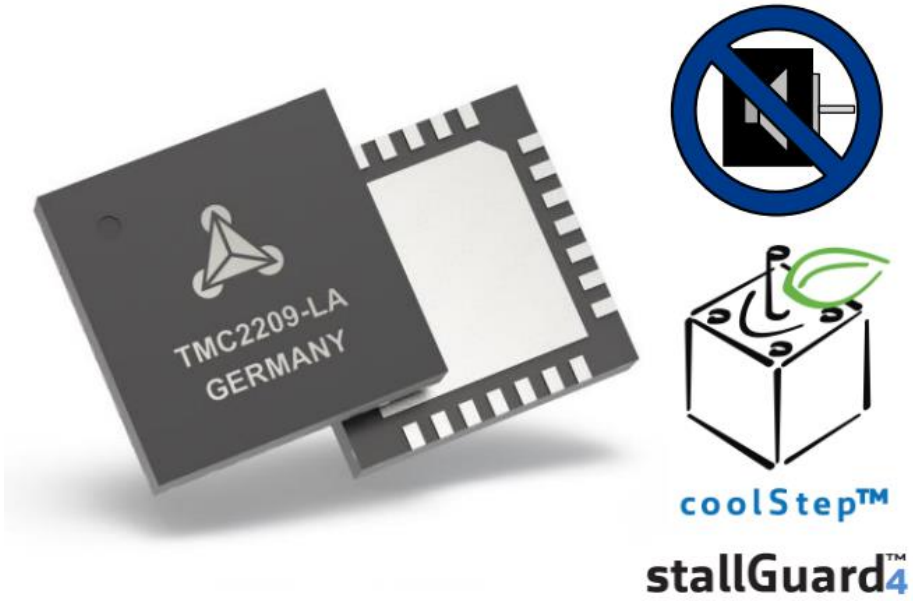
ESP32-WROOM-32D

- 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

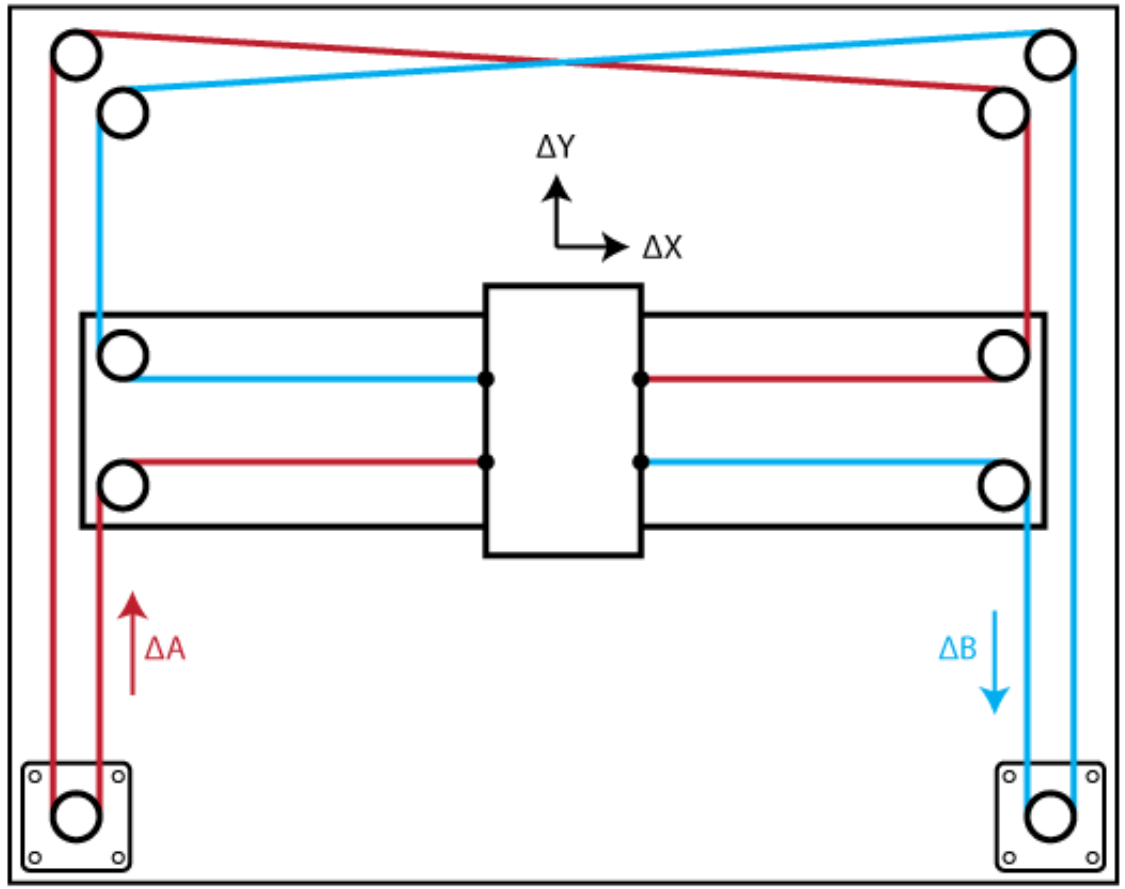


HARDWARE DESIGN

LEAD: FERNANDO VALDES-RECIO

X-Y CARTESIAN MOVEMENT

CORE-XY

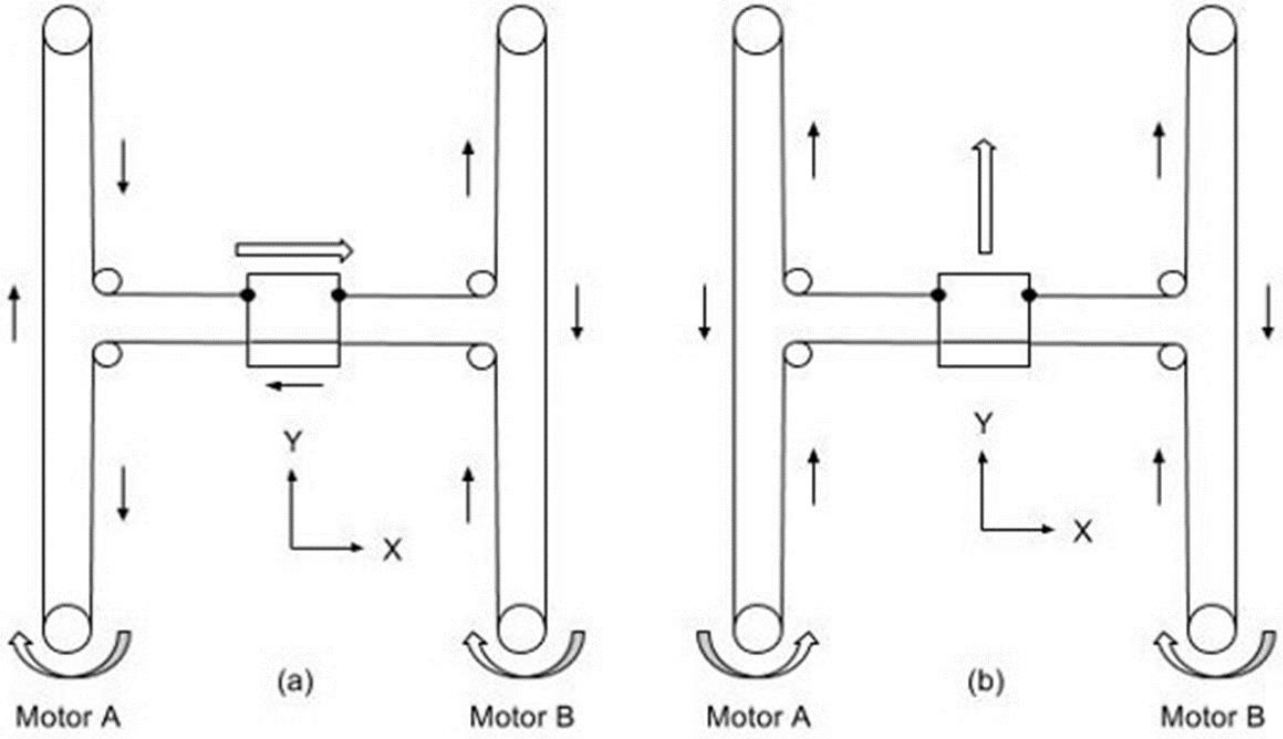


Equations of Motion:

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

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

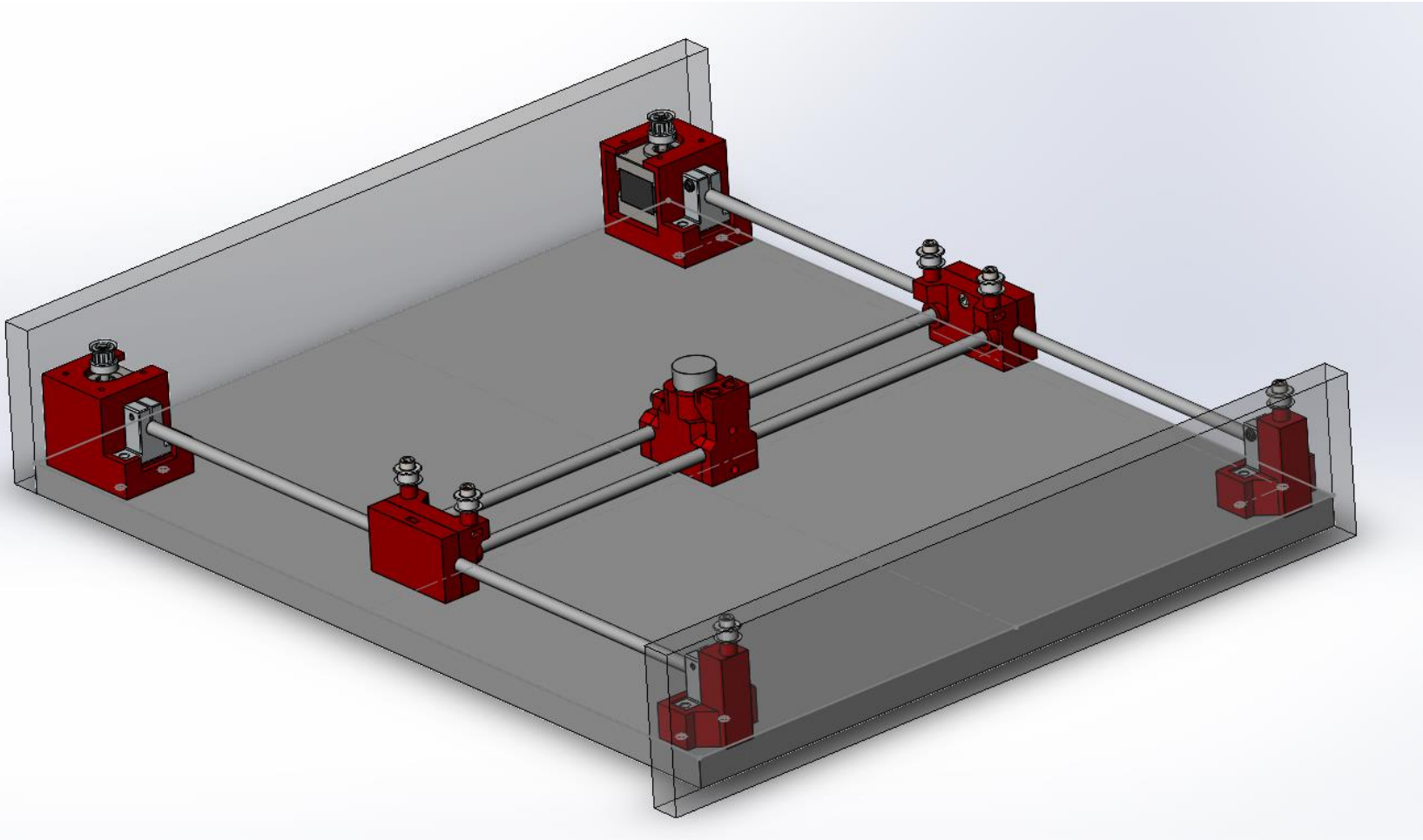
H-BOT



MOTORS

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

CAD



EARLY ASSEMBLY

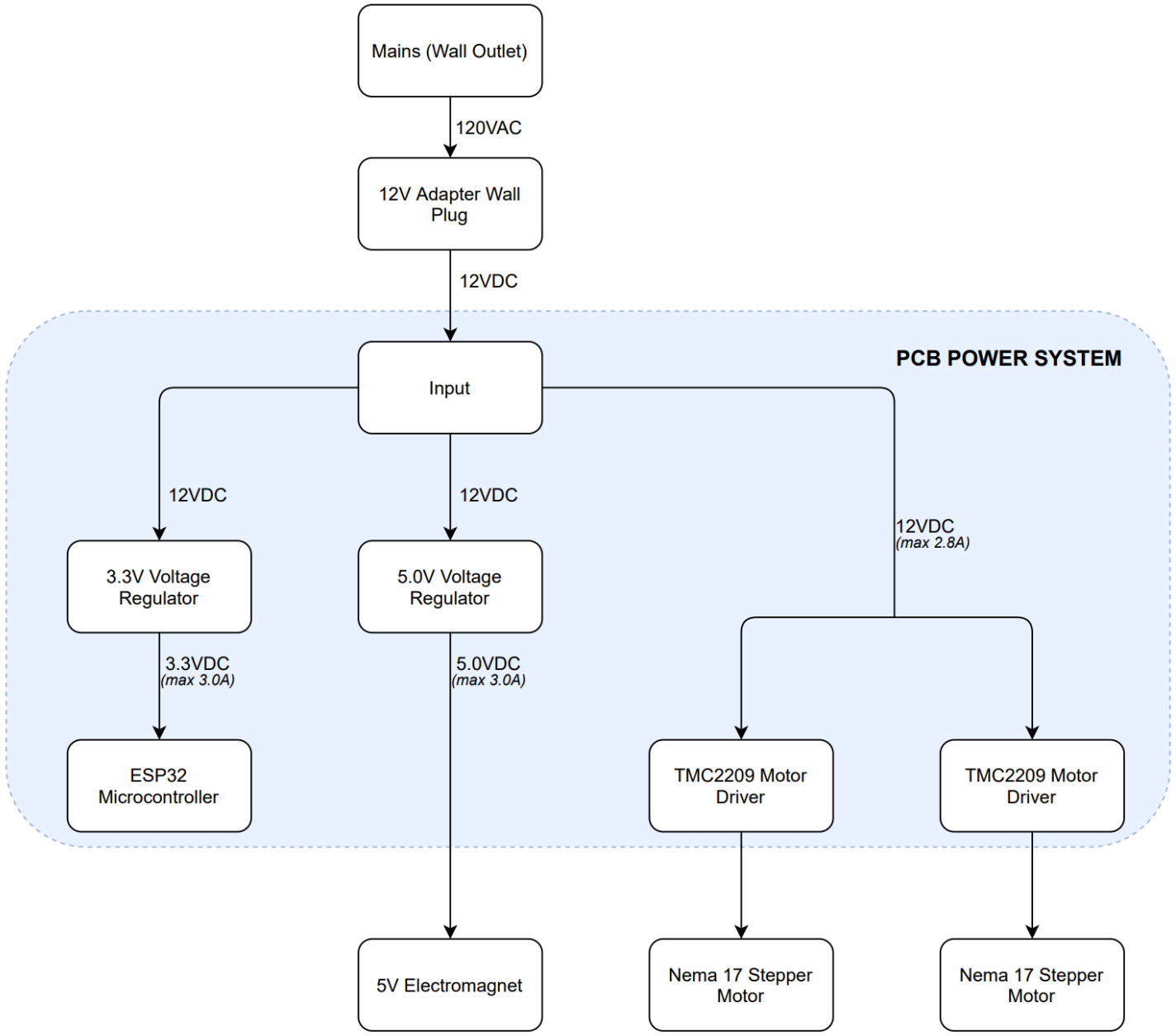
CHALLENGES

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

ELECTRICAL DESIGN

LEAD: KAYLA FREUDENBERGER

POWER SYSTEM FLOWCHART



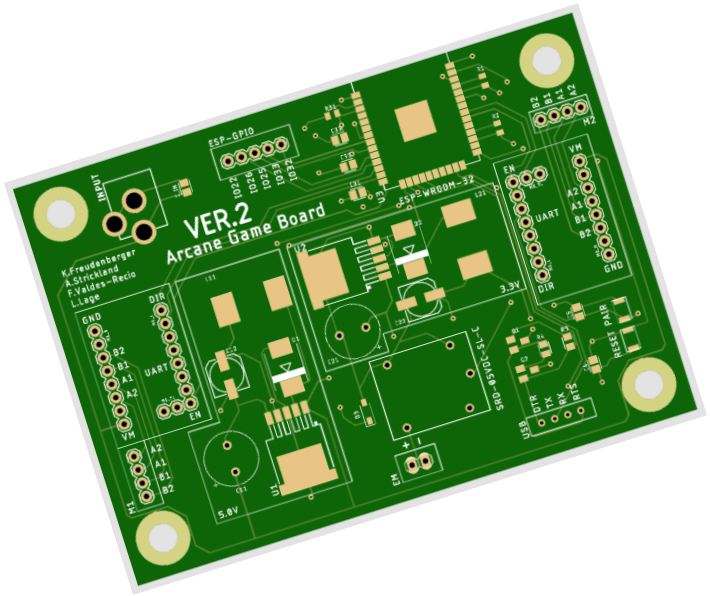
VOLTAGE REGULATOR

- 2 Voltage Regulators
  - ESP32 – 3.3V, 500mA
  - Electromagnet – 5.0V, 300mA
- LM2596SX Switching Step-Down Voltage Regulator
- 3.3V and 5.0V Fixed Output
- Junction Temperature Range -40C to 125C
- 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™

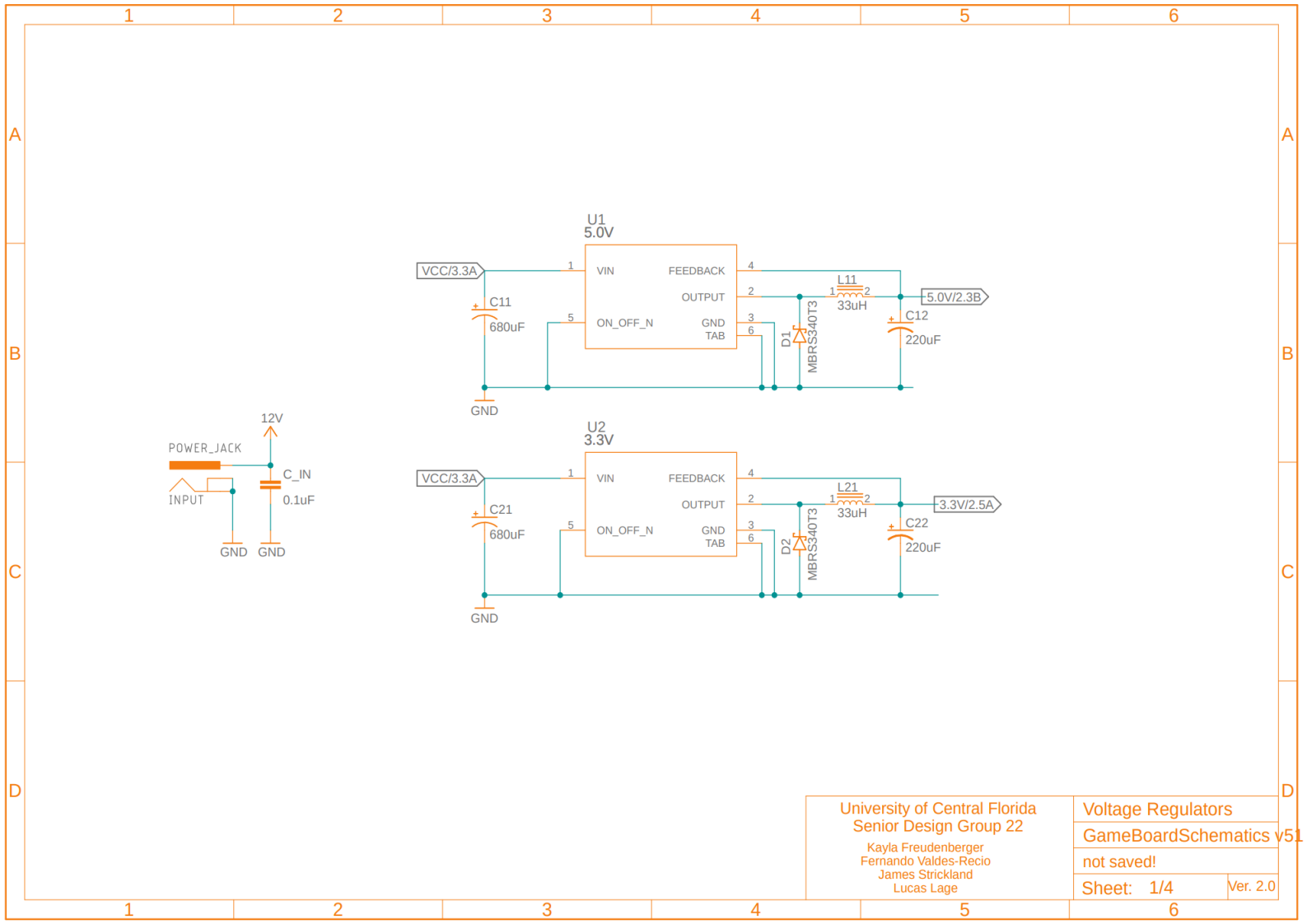


- PCB Software: Fusion 360
- Manufacturing: JLCPCB
- Status: Ordered, waiting for delivery
- Version 2
  - Needed to redesign Version 1 due to component availability

J@LC JLCPCB

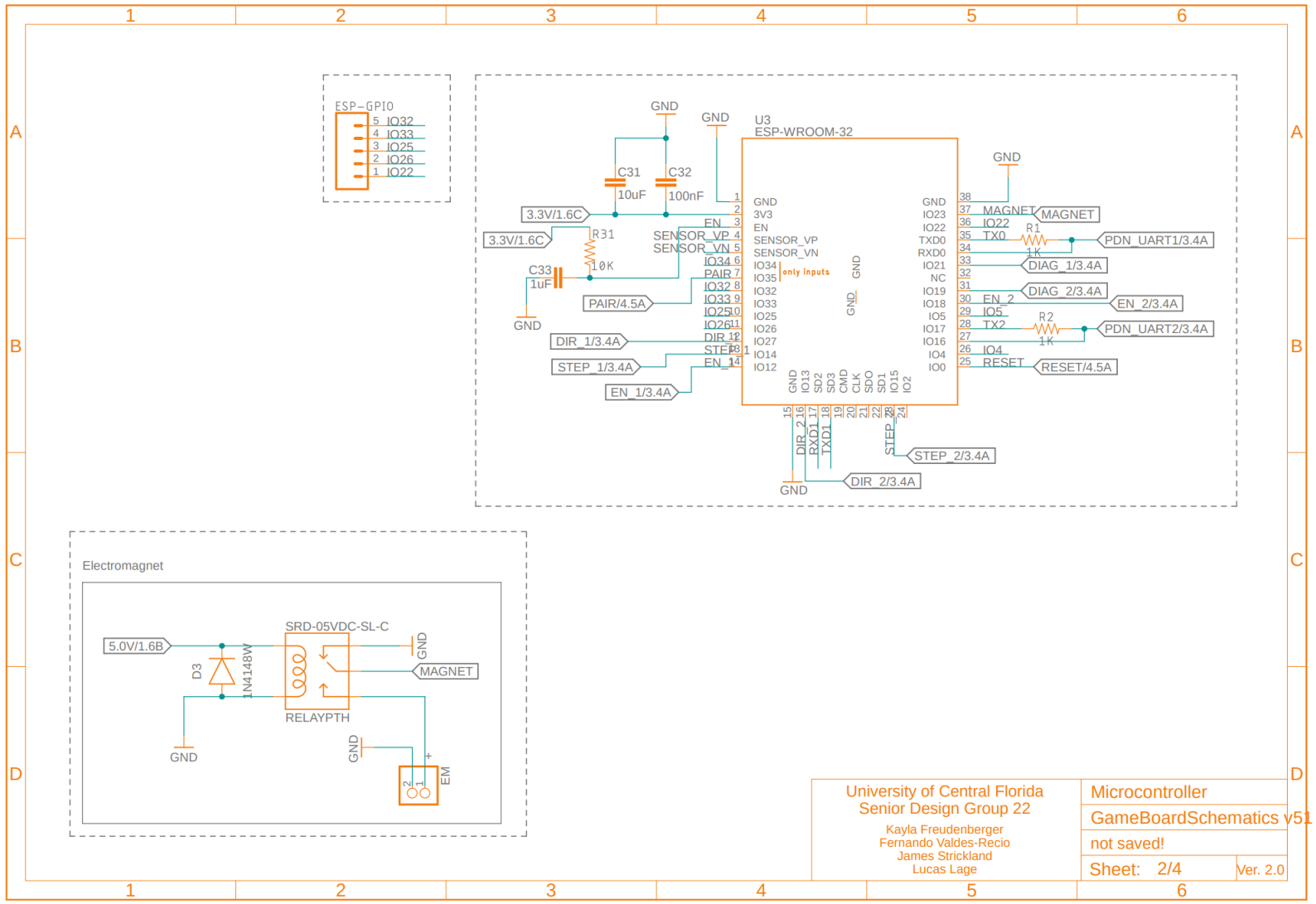
PCB SCHEMATIC VERSION 2.0

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



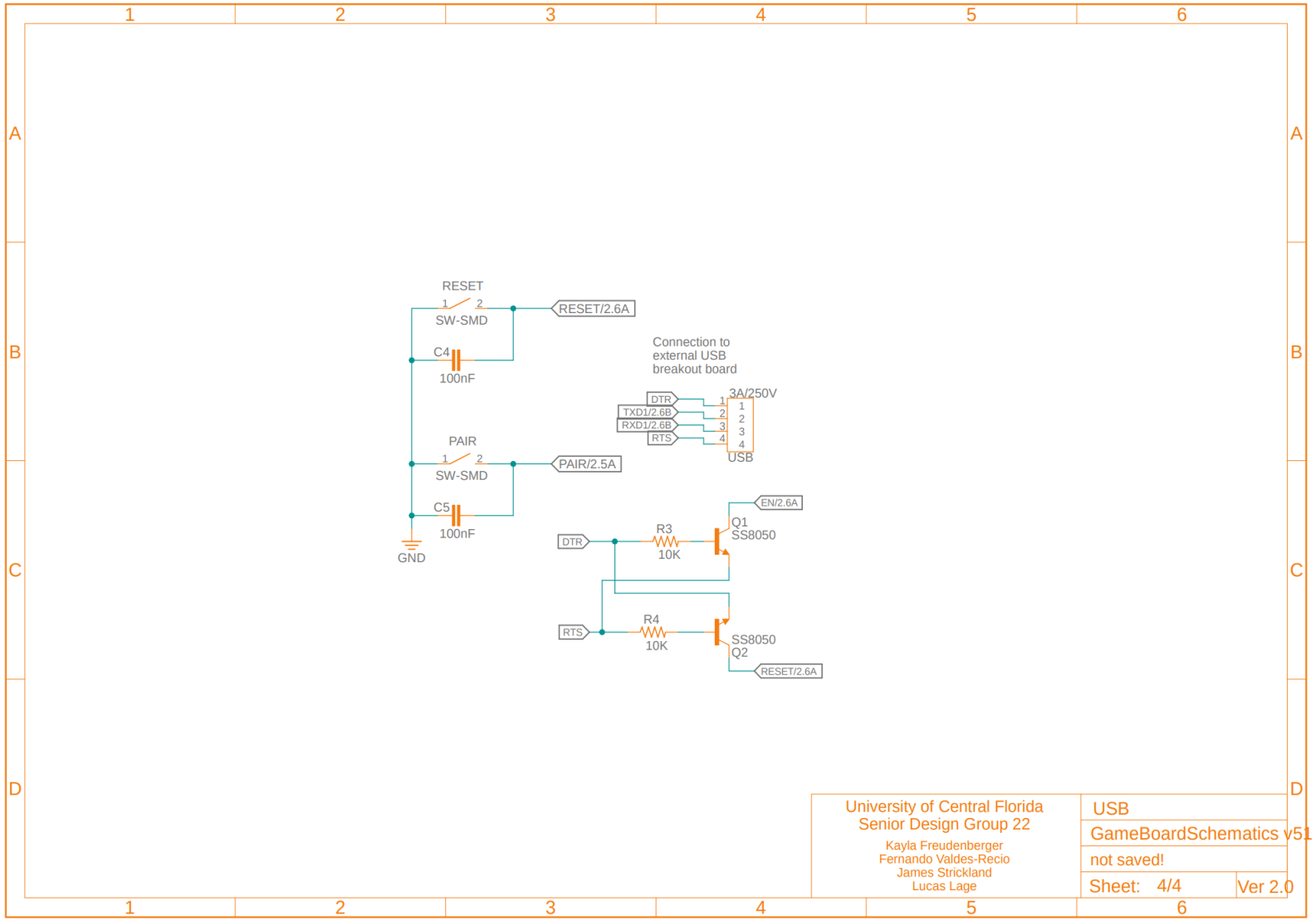
PCB SCHEMATIC VERSION 2.0

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



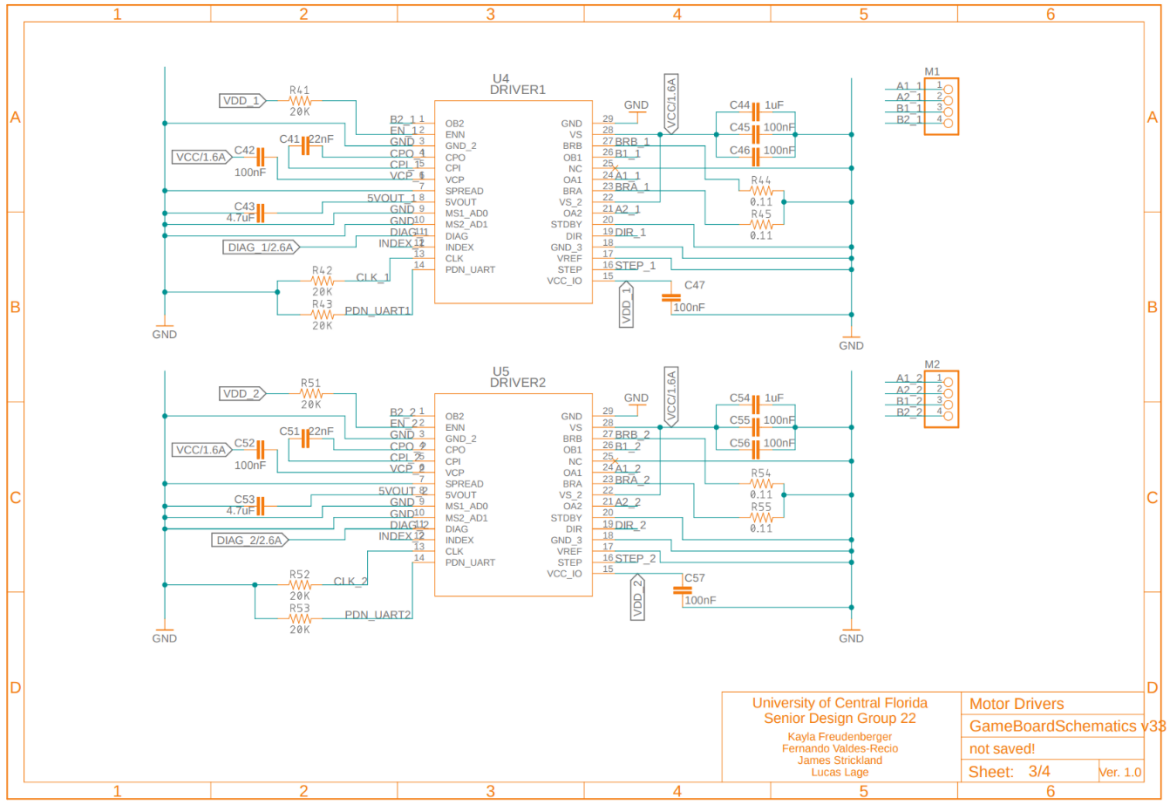
PCB SCHEMATIC VERSION 2.0

- USB Connection

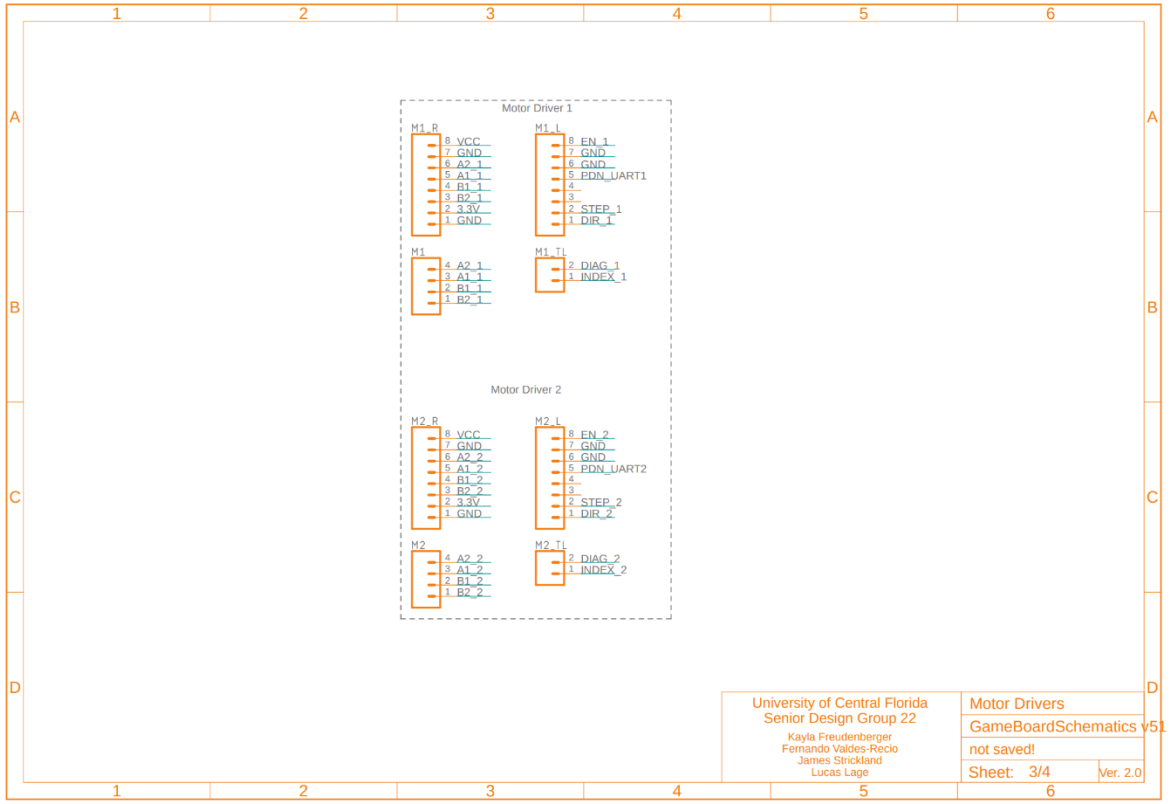


PCB SCHEMATIC – TMC2209 MOTOR DRIVERS

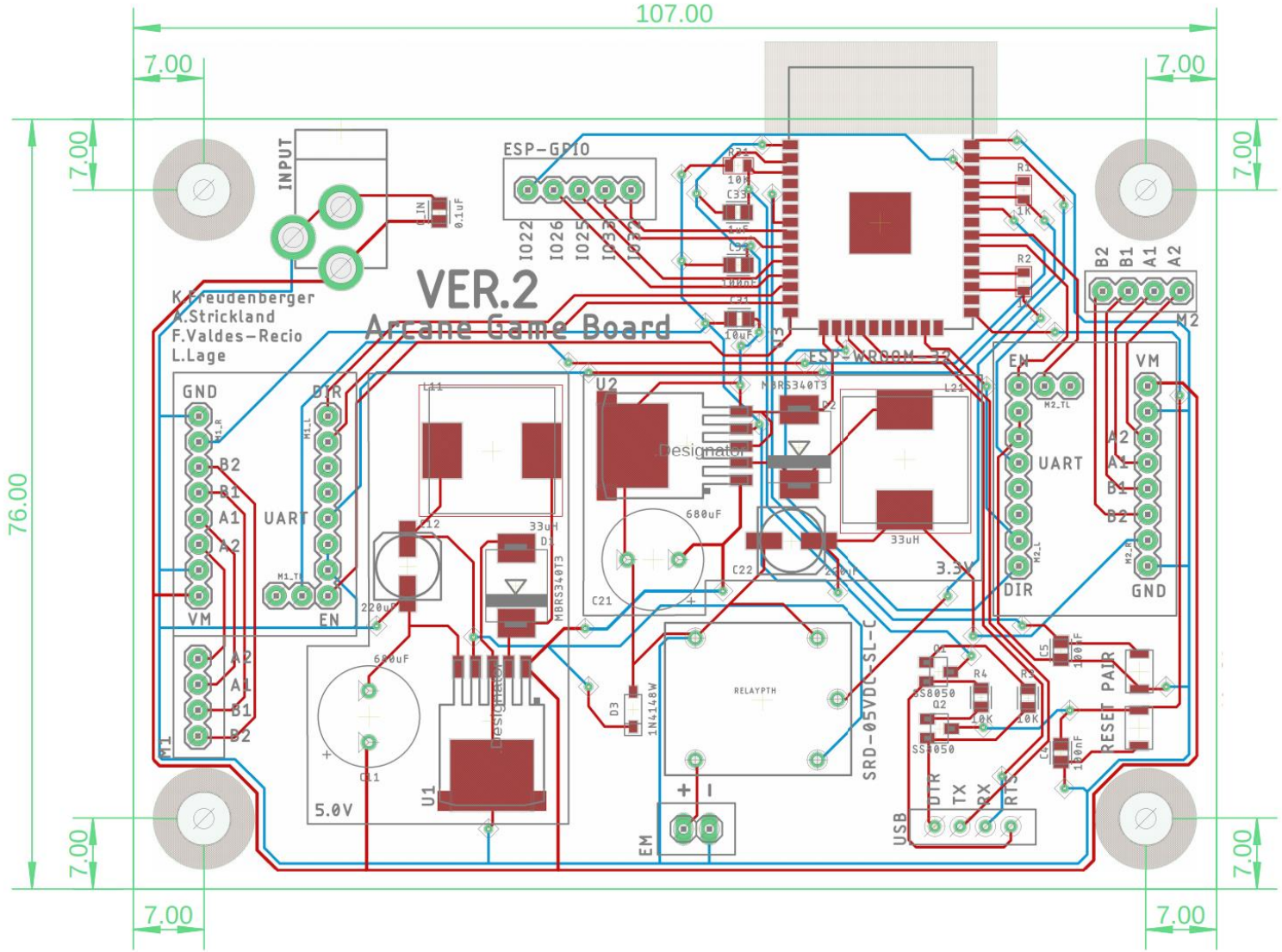
VERSION 1.0



VERSION 2.0



PCB LAYOUT VERSION 2.0



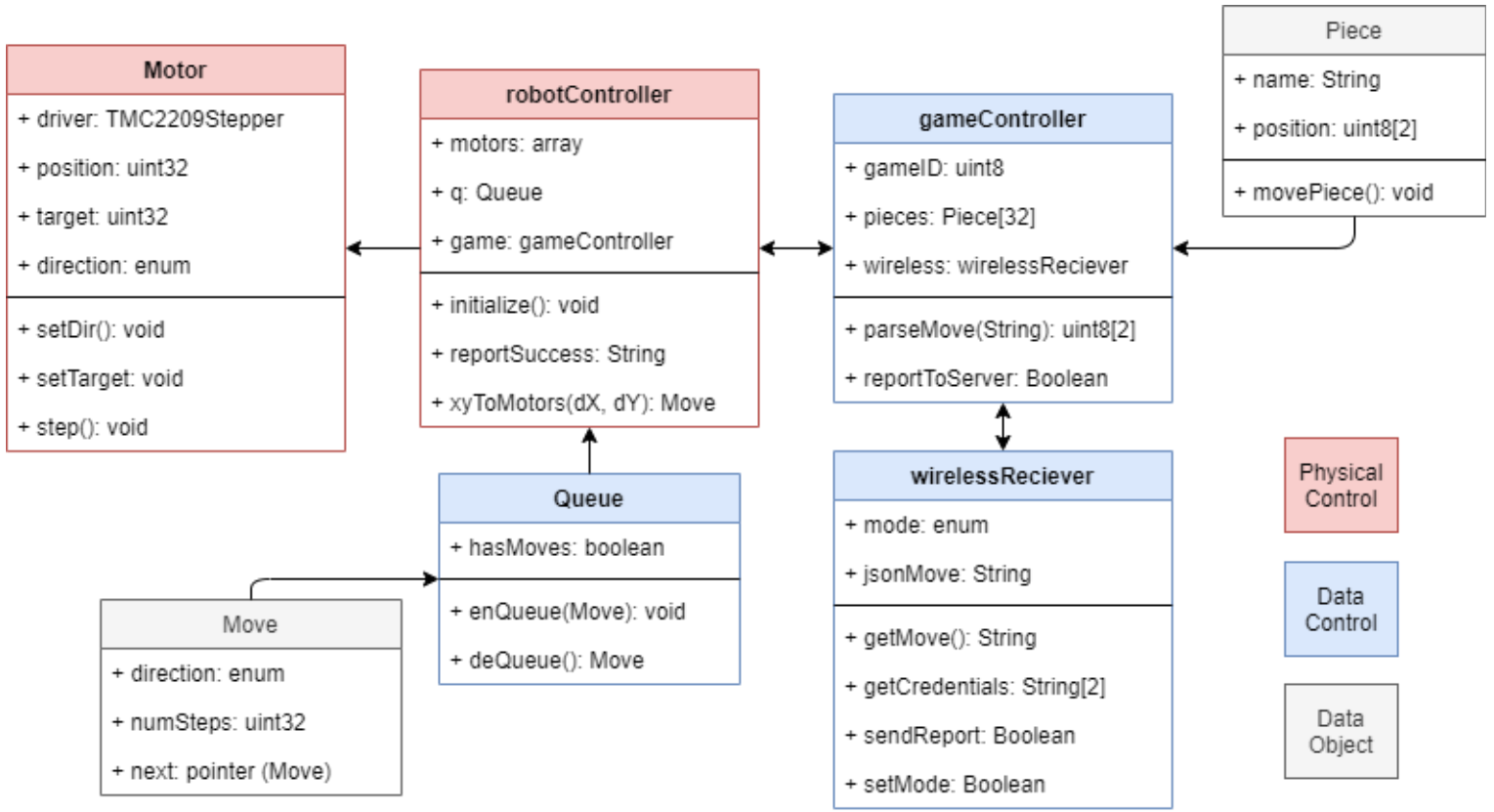
CHALLENGES

- 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
  - New Voltage Regulators needed for Ver. 1.0 design
  - TMC2209 Motor Drivers out of stock

FIRMWARE DESIGN

LEAD: J. ANTON STRICKLAND

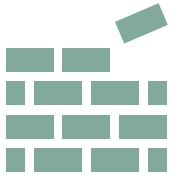
FIRMWARE BLOCK DIAGRAM



Notes:

- Robot Controller contains the FSM logic
- Game Controller allows for future piece tracking expansion

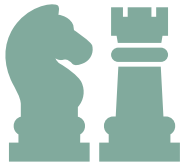
OVERVIEW



Structural Overhaul

Stepping Away From Marlin

Enhanced OOP Design



Introduction of Game and Robot State Control

Acts as a Slicer

Works in Parallel with Web-App

Challenges

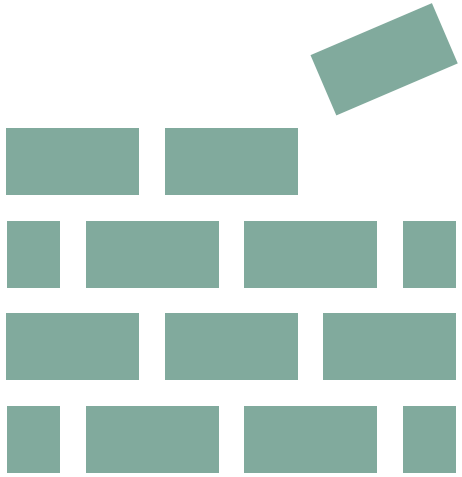
ESP IDF vs Arduino Framework

Library Deprecation and Incompatibility

Unit Testing

Feature Testing Actively Underway

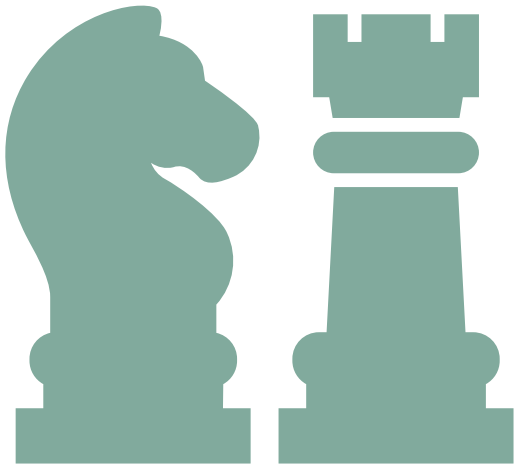
43% completed overall



STRUCTURAL OVERHAUL

- Stepping Away From Marlin
  - Too Complex
  - Many errors when a direct port was attempted
  - Built for different a task
- Enhanced OOP Design
  - Drivers for each function category
  - Opportunities for Multi-Threading instead of ISRs

GAME AND ROBOT STATE CONTROL



- Acts as Slicer
  - Converts chess move to H-bot commands
  - Simplifies HTTP connection by leveraging .JSON
- Works in Parallel with Web-App
  - Basic Navigation Command
  - Avoidance Transposition

CHALLENGES

- ESP-IDF vs Arduino Framework
  - Initially thought to be a non-issue
  - Documentation for both has strengths in different areas
  - Arduino chosen for the sake of rapid development
- Library Deprecation and Incompatibility
  - Major libraries intended to save significant development time
    - WDT Errors causing reboots were common
    - Lack of full ESP32 support

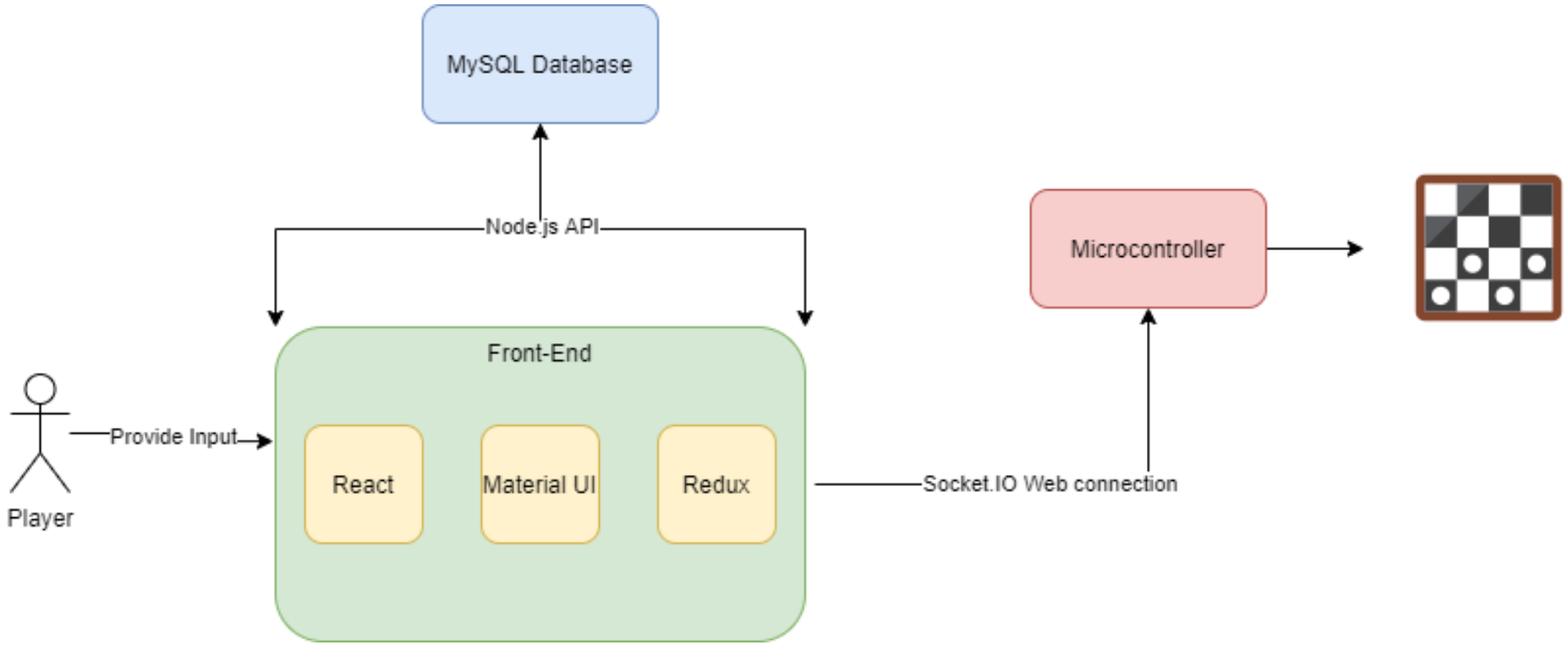
TESTING PROGRESS

Category	Active Feature Dev	Tests Completed/Planned	Testing Progress
Motor Control Driver	None	9/9	100%
Queue/Moves Implementation	Dynamic Memory Allocation	7/13	54%
Robot Control Driver	Motor Wrapper	3/16	17%
Game State	Off-Board Server IO	4/11	36%
UX	None	0/5	0%
		23/54	**43%**

SOFTWARE DESIGN

LEAD: LUCAS LAGE

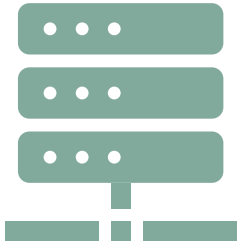
SOFTWARE BLOCK DIAGRAM



OVERVIEW

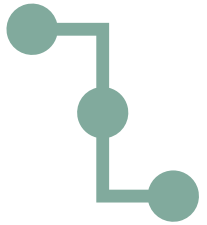
Front-End Design

Tools, Frameworks,
Redux, and Dataflow



Back-End Design

Tools, API endpoints,
and Design



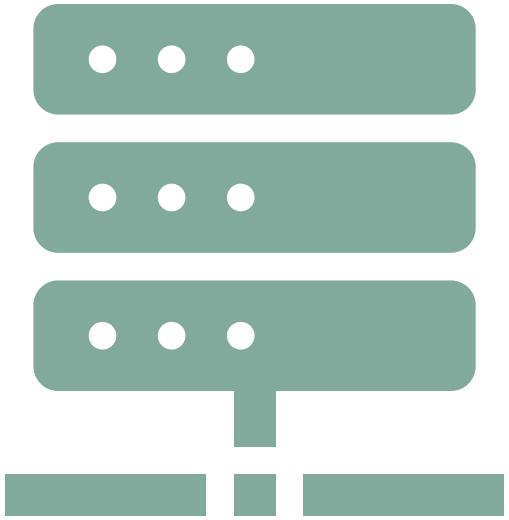
WebSocket Connection

Socket.IO, Making a connection,
Working with firmware

FRONT-END DESIGN

- Main Tools
  - React Framework
  - Material-UI
  - Redux
- Dataflow
  - User inputs
  - Local Storage
  - API Calls

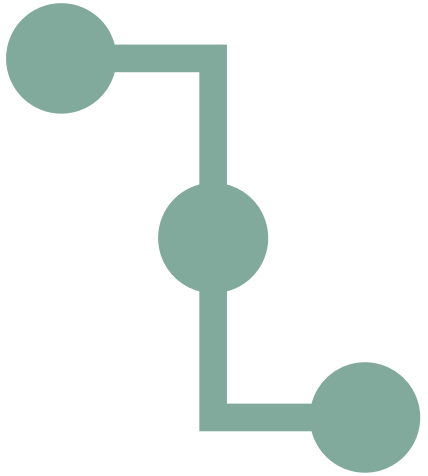
BACK-END DESIGN



- Tools
  - MySQL Database
  - Node.js API
- Design
  - Relational Database
  - REST API

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



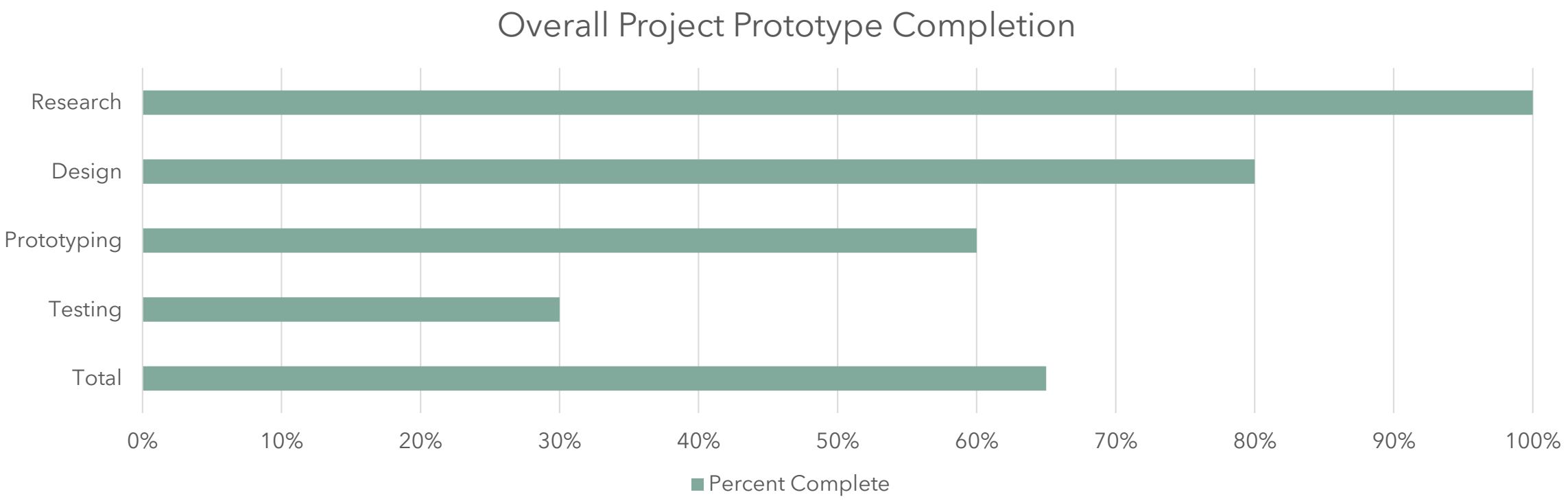
ADMINISTRATIVE CONTENT

BUDGET AND BUILD PLANS

CURRENT 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	$2.99	$11.99
Nema 17 Stepper Motor (3)	1	$25.99	$25.99
PCB Version 2.0	1	$71.41	$71.41
12V AC/DC Adapter Plug	1	$9.99	$9.99
Threaded Inserts	1	$9.19	$9.19
Bushings	1	$5.99	$5.99
Construction Materials	1	$40.00	$40.00
PLA Plastic Spool	1	$19.99	$19.99
Barrel Jack	1	$1.27	$1.27
		Total:	**$262.35**

TOTAL PROJECT PROGRESS



PLANS MOVING FORWARD

- Test PCB Version 2.0 – Make modifications to Version 3.0 design as needed
- Construct the walls of the H-Bot housing
- Finishing firmware MVP by July 7th
- Implementing Blu-Fi module toggling for better UX experience

QUESTIONS?