UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

Wireless Remote Motor Controller

Electrical & Computer Engineering

Group #20 Aaron Chen, Boon Sheng Lee, Kyungha Kim

12/05/2023

Agenda

1. Problem Statement
2. Design Overview
3. Project Build and Testing
4. Successes and Challenges
5. Conclusions
6. Recommendations for Further Work

1. Problem Statement

Purpose of the Project

The Importance of an Efficient Motor Controller

- **Various applications** like robotics, drones, and remote-controlled vehicles
- **Simplicity** in the design systems
- Wireless control through **user-friendly interface** web design

2. Design Overview

High Level Requirements & Block Diagram

High Level Requirements

1. Wireless Control

- **Wi-Fi remote control** of a DC motor within 5-10 meters.

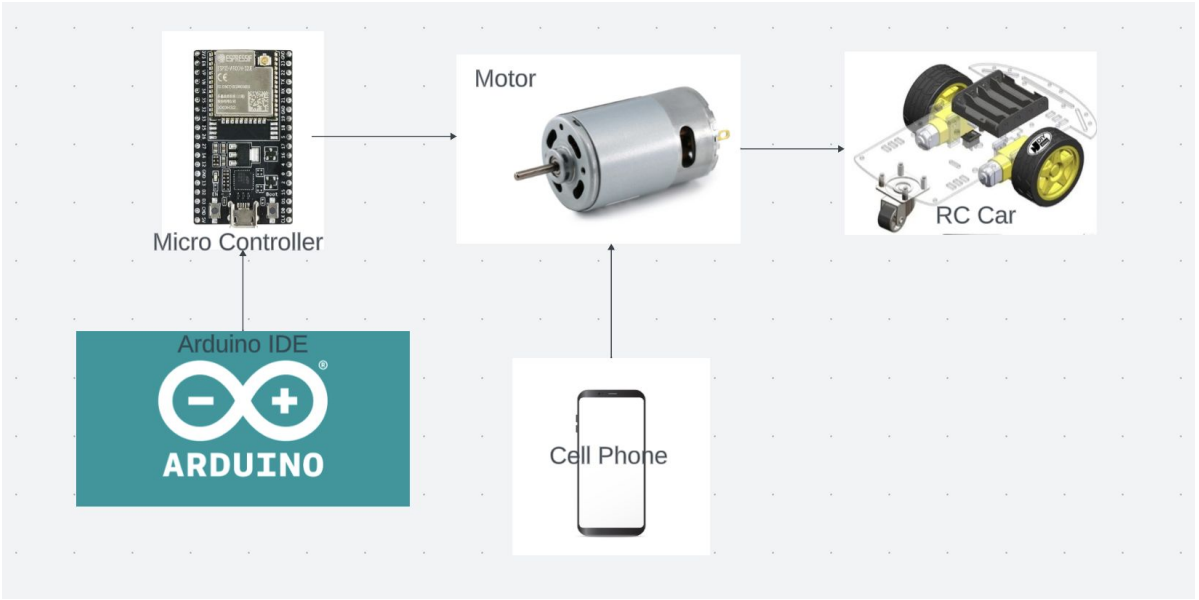
2. Voltage & Current Control

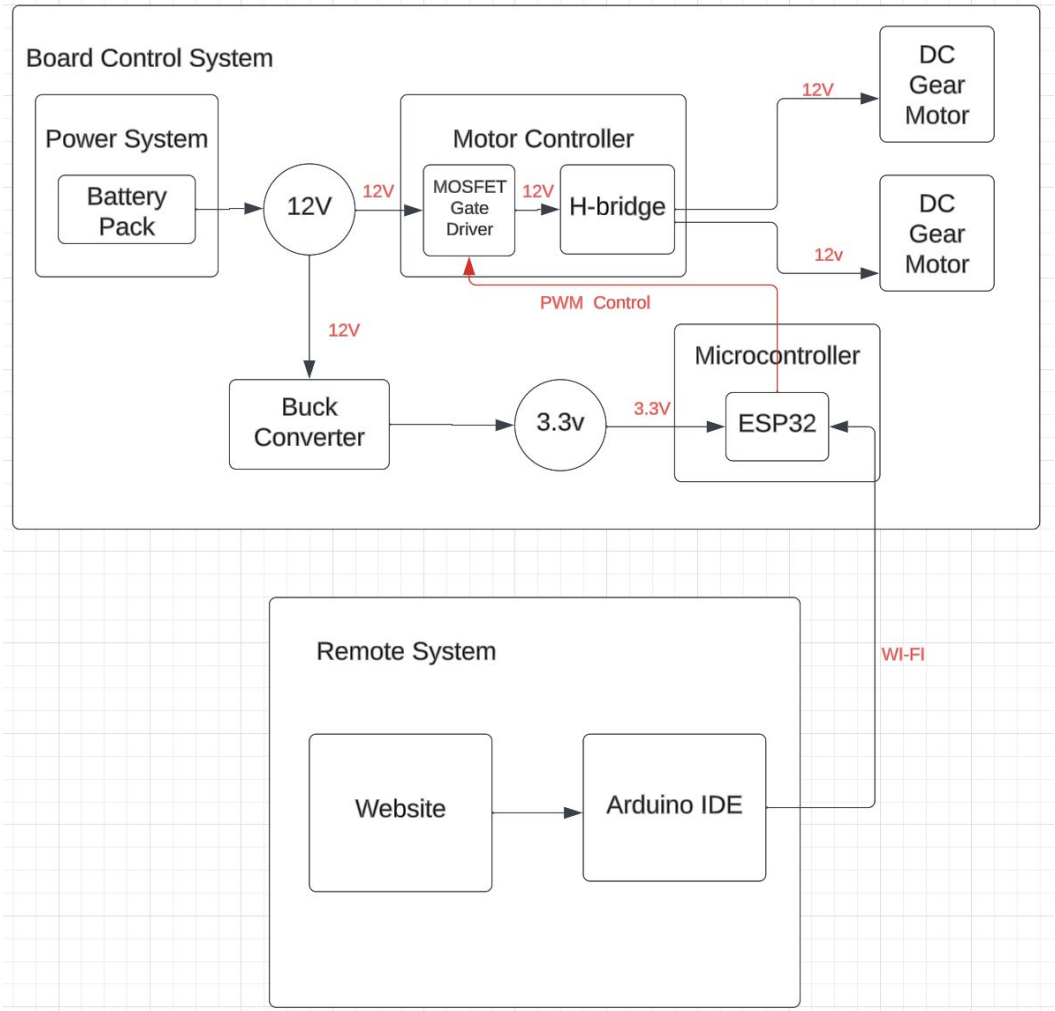
- supports **12-24V DC** and handles up to **10A current.**

3. Speed Control

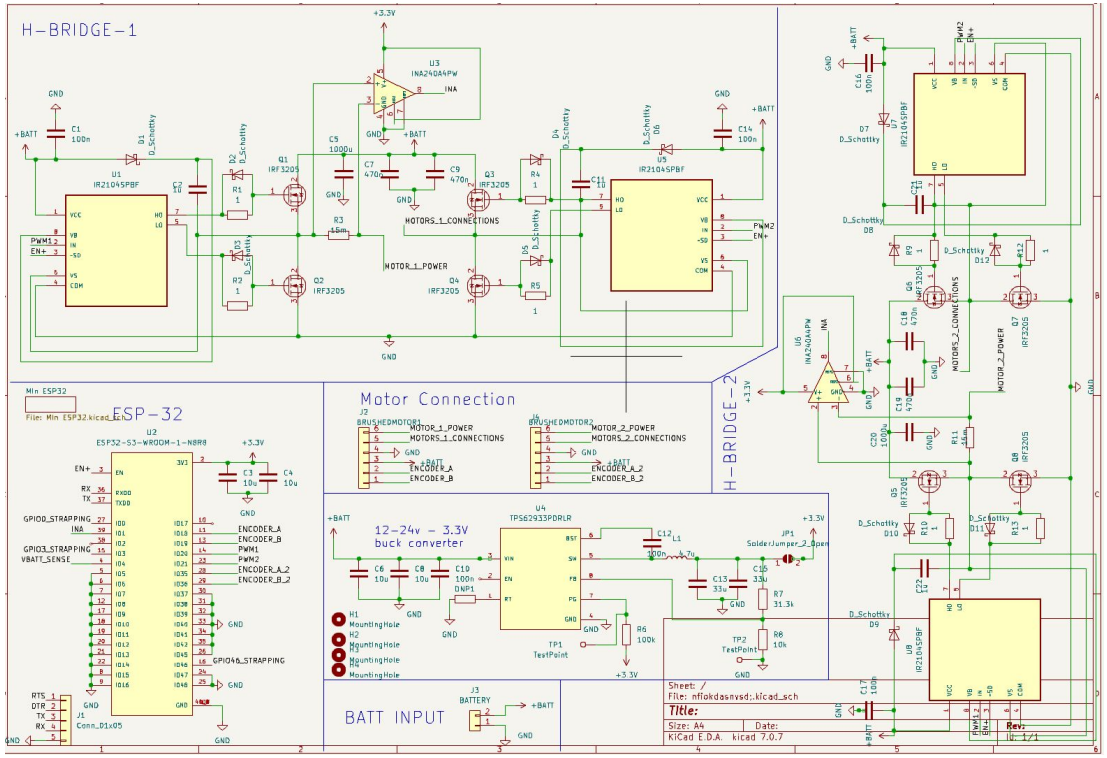
- Users can adjust motor speed from **0 to 100%**.

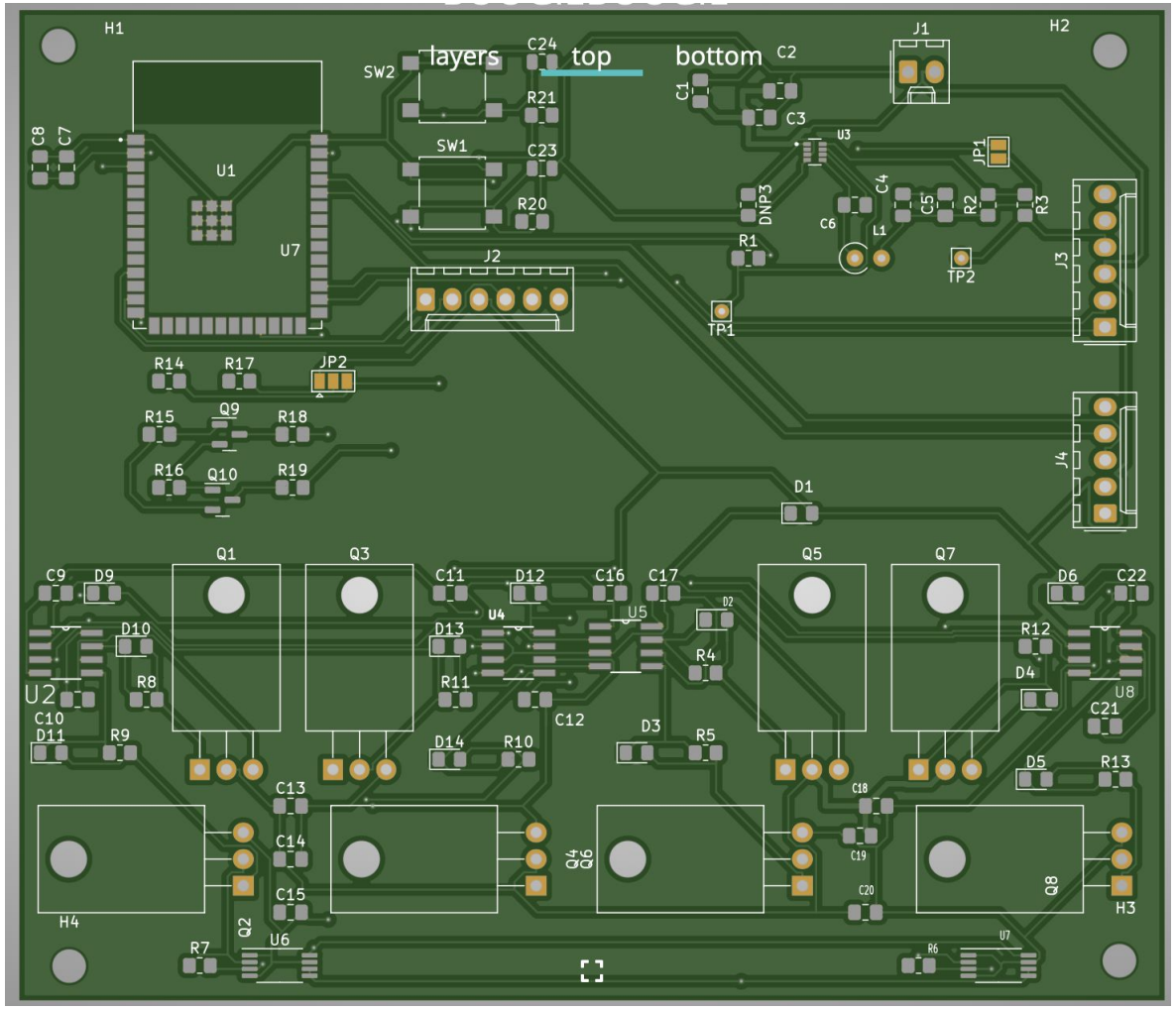
Block Diagram and Visual Aid





Schematic and PCB Design



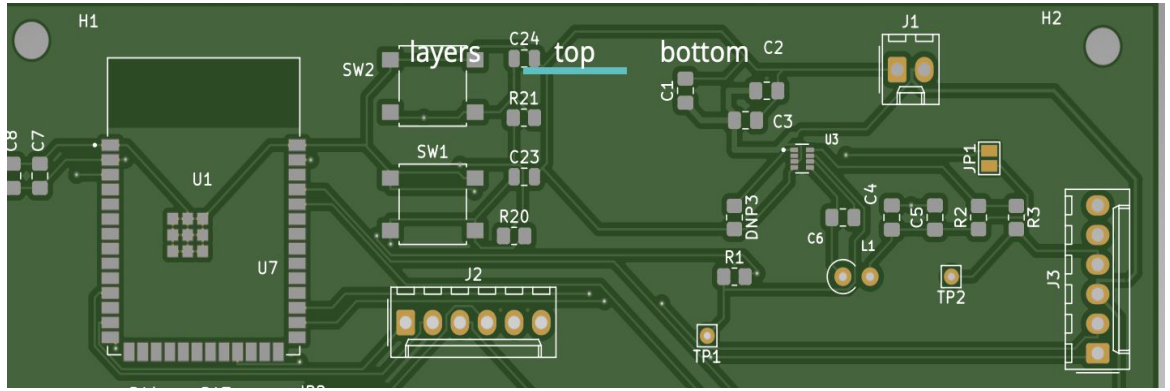


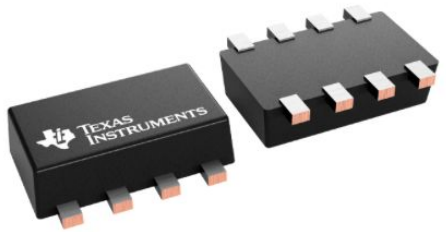
3. Project Build and Testing

Subsystems
Presentation of Project Video

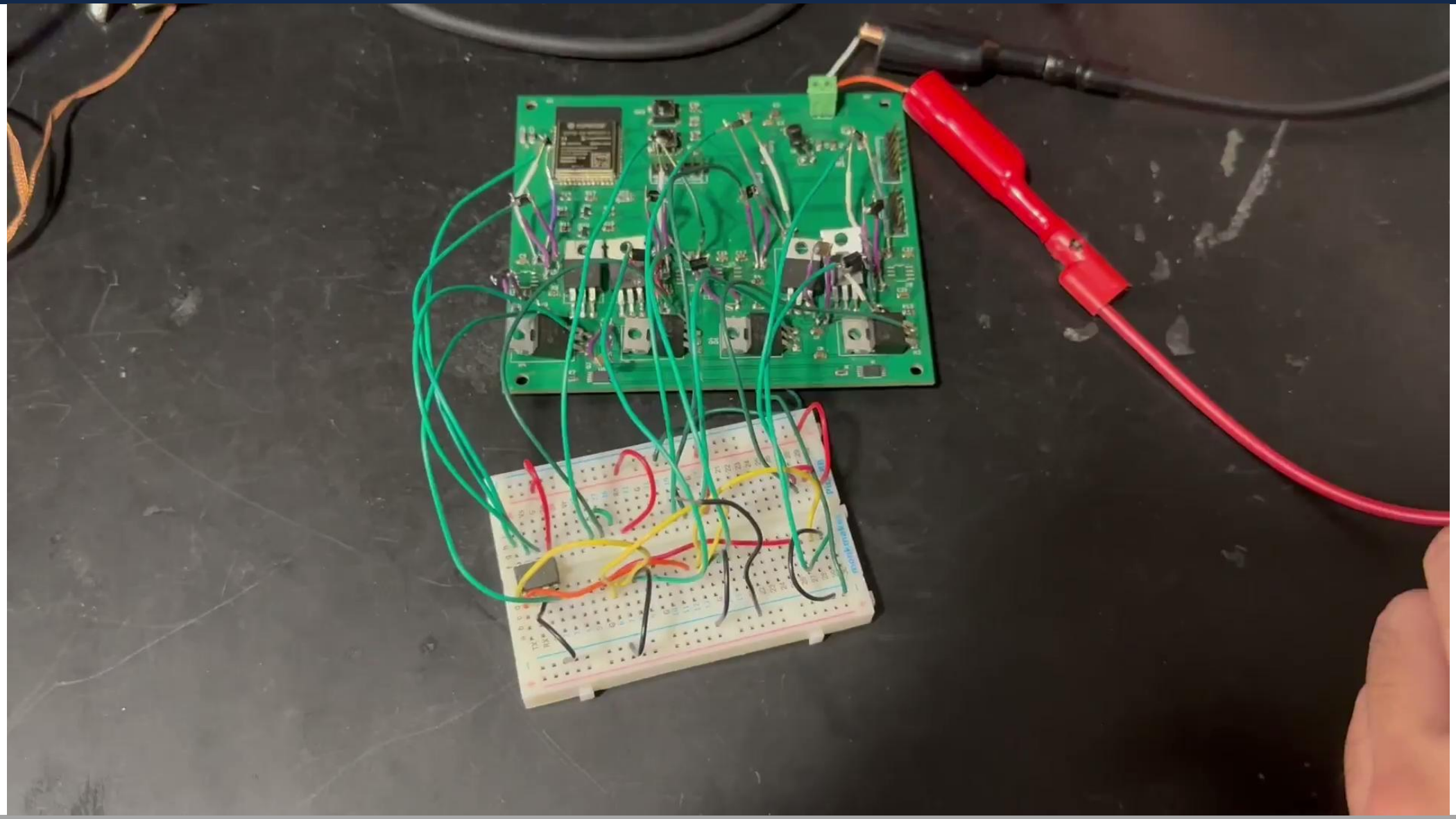
Power Subsystem Overview

Power System Requirement	**Verification**
Limit the input voltage of the ESP 32 to only 3.3 Volts	• Check the buck converter output with a multimeter to ensure a steady 3.3 volts before connecting it to the ESP32.
Organize the voltage and current to each system for when it is needed.	• Test points to monitor subsystem currents and voltages, checking for irregularities





Video Presentation



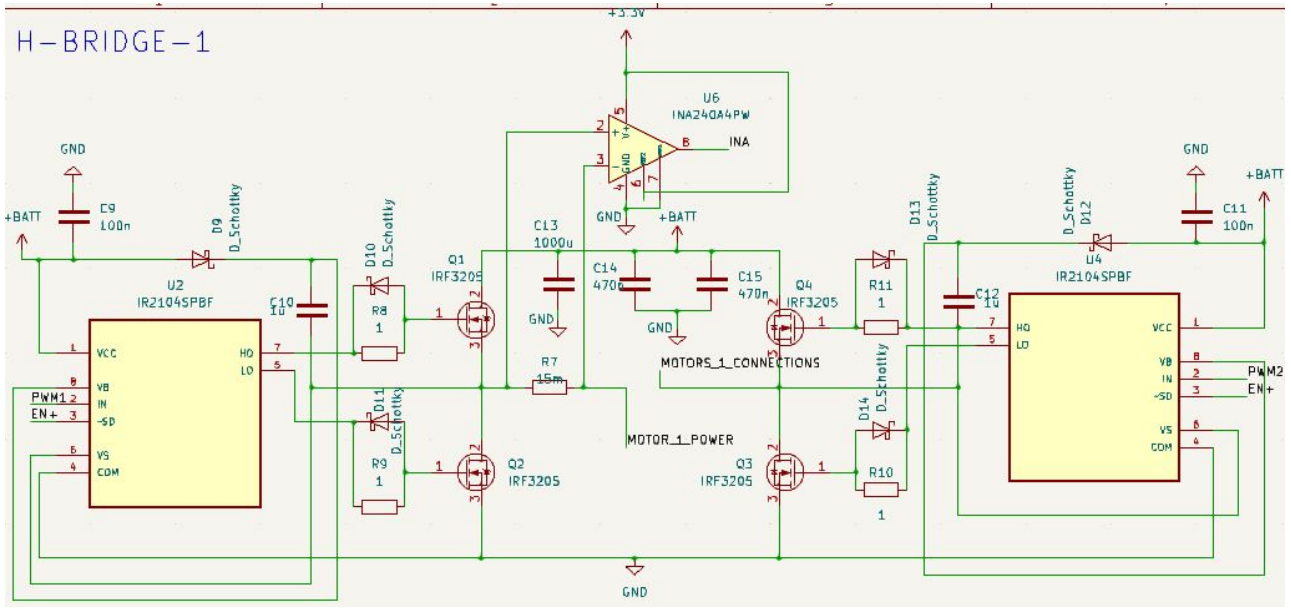
Motor Controller Subsystem Overview

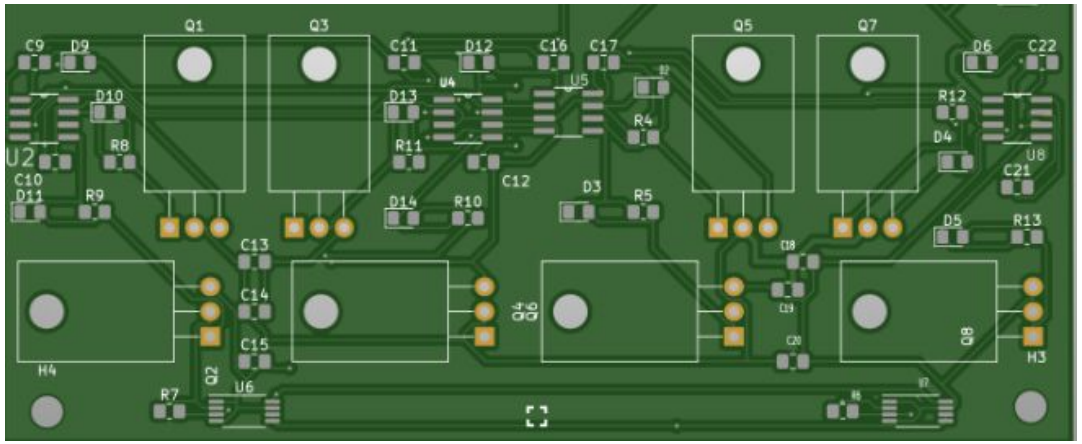
- **Control**ling the **motor**’s speed, direction, and braking
- H-Bridge with four MOSEFTS, two gate drivers, and a current sensor

Requirement	**Verification**
The motor controller can handle 12V	• Verify using a 12V voltage source • Test points in schematic to monitor system at 12V
IRF3205 MOSFETs fully switch on with a ±20V input from the gate driver	• Select the IR2104SPBF MOSFET driver ensures safe operation within a 10 to 600V range for N-channel MOSFETs.

Motor Controller Subsystem Overview

- Schematic for H-Bridge and PCB layout





Remote Website Subsystem

- Manages **DC motor behavior** through Motor Controller Subsystem
  - acceleration
  - deceleration
  - forward and backward movement
  - stopping

Requirement	**Verification**
The ESP32 must interpret **user input status**.	● Verify default website functionality ● Log data sent to ESP32 ● Ensure DC motor responds correctly to commands.

ECE 445: Wireless Remote Motor Controller (Group 20)

Motor 1 Control

< Slower

Stopped - CW at 60%

Faster >

CCW

START

CW

Motor 2 Control

< Slower

Stopped - CW at 60%

Faster >

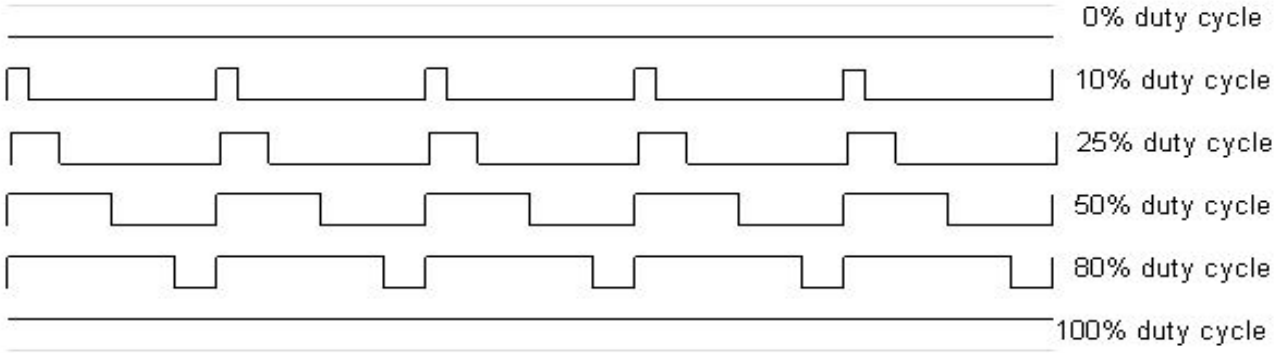
CCW

START

CW

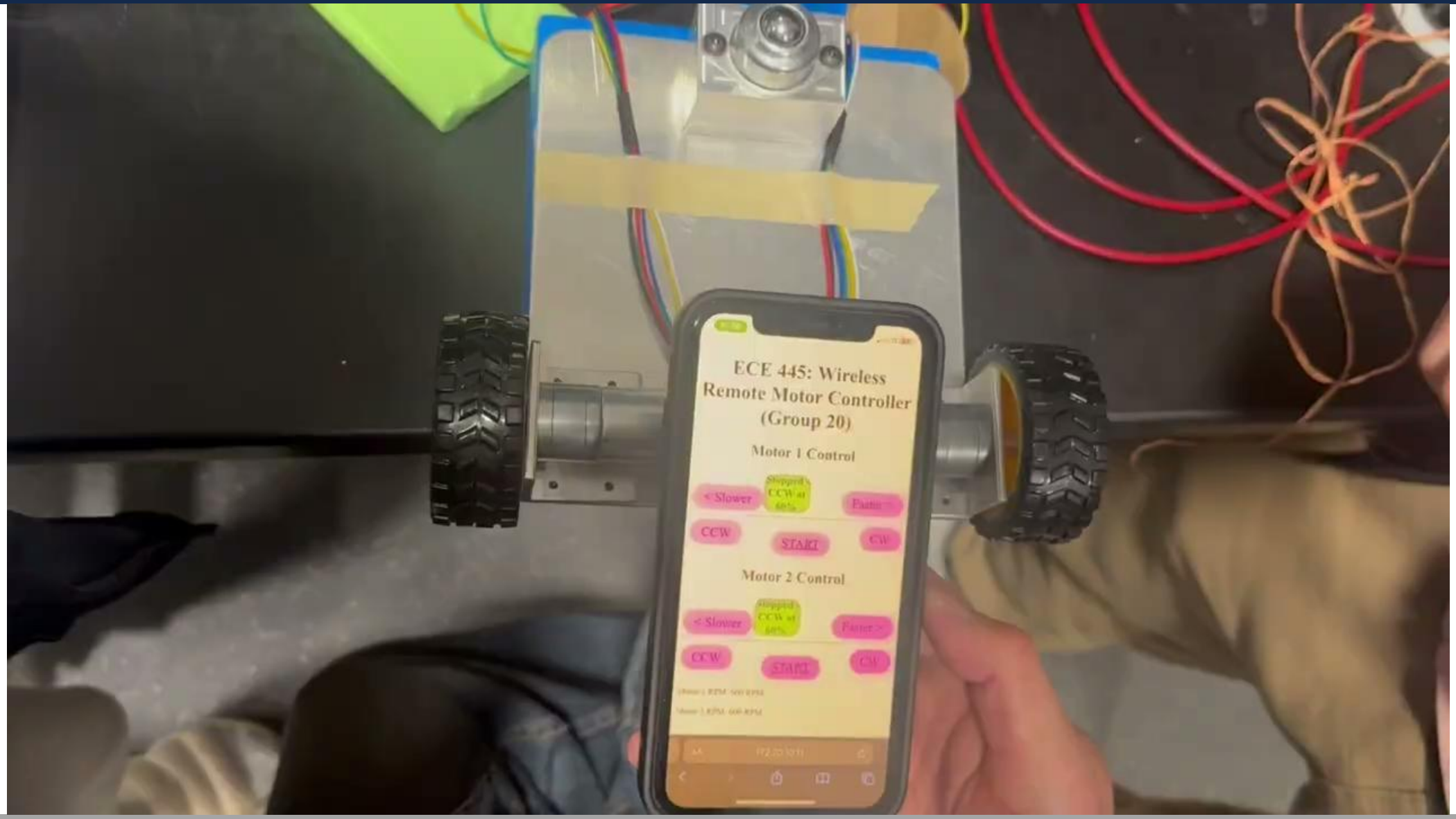
Remote Arduino IDE Subsystem

- Arduino IDE programs the ESP32 to dictate **DC geared motor responses** to website inputs.



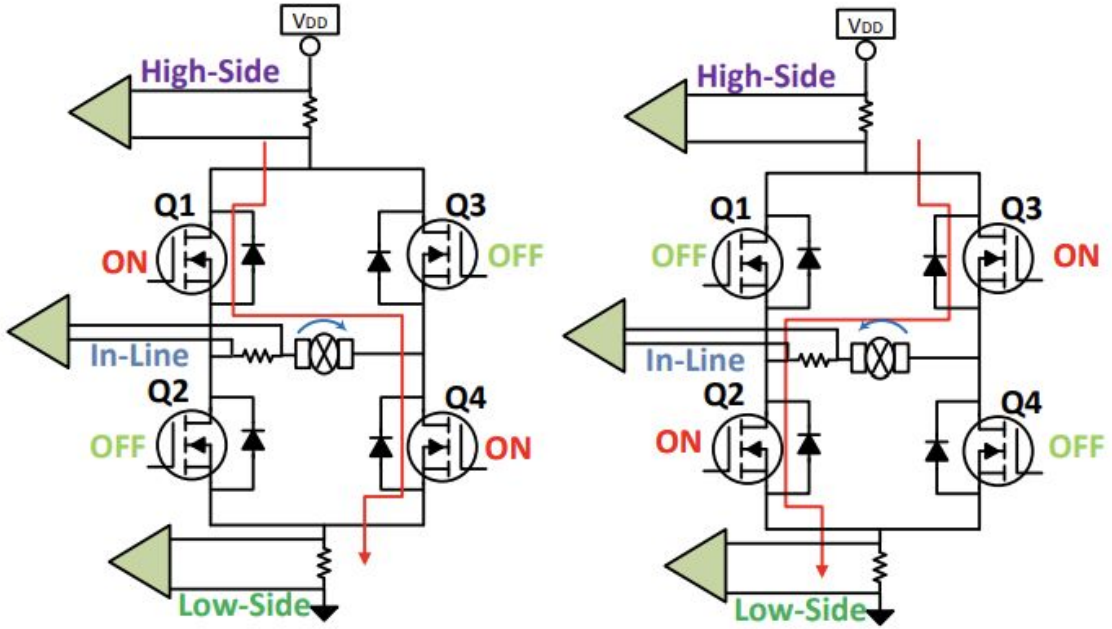
Requirement	**Verification**
Arduino IDE programs ESP32 for **motor direction** via the website commands.	• Test varied commands from the website • Confirm consistent motor response
Arduino IDE programs ESP32 for **speed control** via the website commands.	• Verify PWM signal reduces speed by adjusting duty cycle

Video Presentation



To prevent Short Circuit

1. **Fuse Protection**
   - multiple schottky diodes implemented in the gate driver circuits
   - In-build diodes within the MOSFETs
2. **Dead Time**
   - turn off one set of transistors and turn on the opposite set
3. **Current limiting**
   - in-line current measurement to accurately measure the motor current

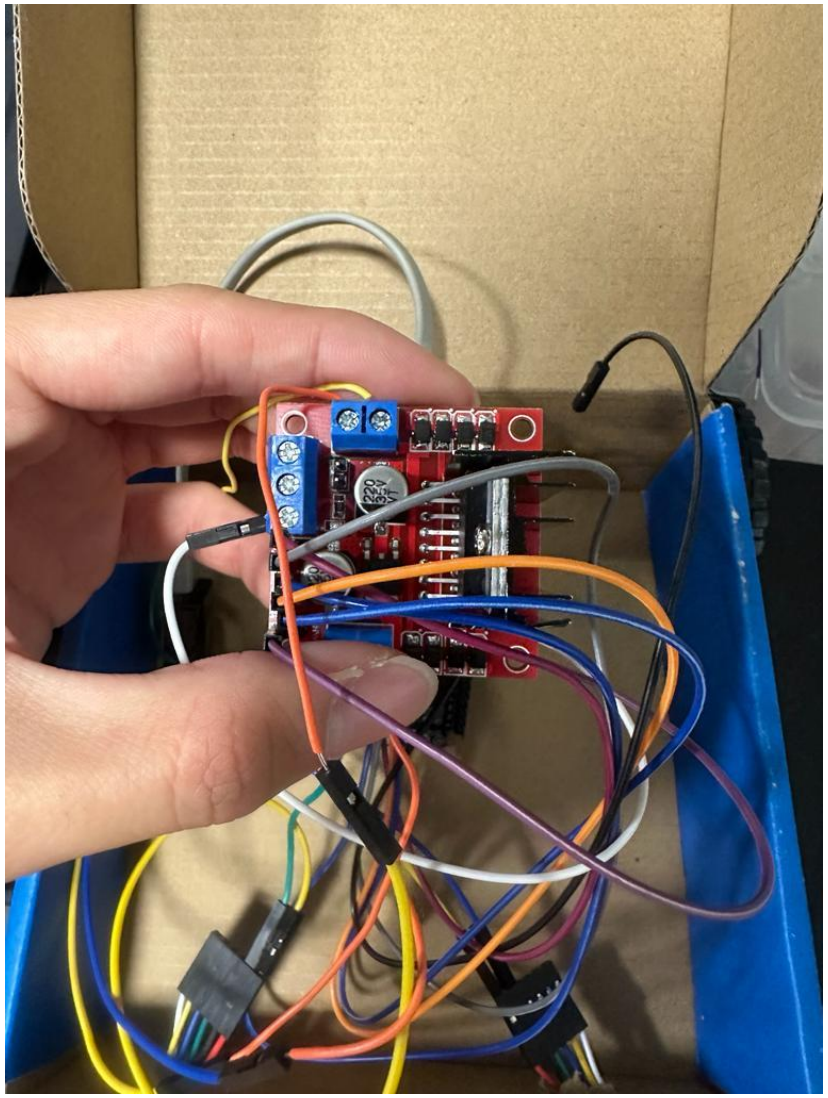


H-Bridge Circuit

4. Successes and Challenges

Achievements
Obstacles Faced

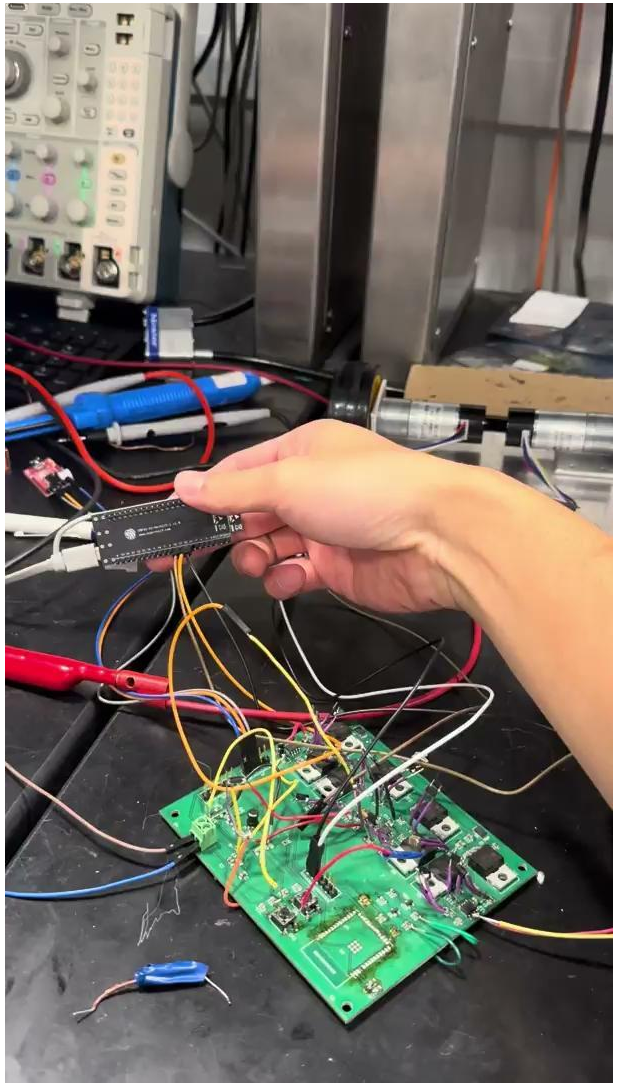
Achievements



Positive Aspects

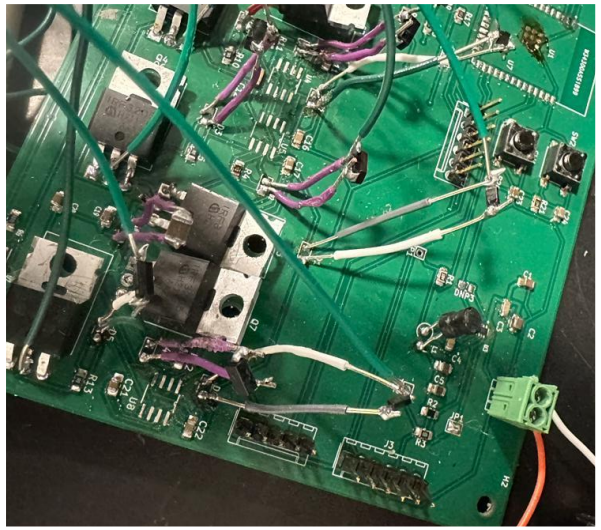
- **Power** subsystem
  - outputs 12V to Power Line
  - outputs 3.3V to ESP32 & Current Sensors
- **Circuit** functions properly after replacing H-bridge with L298N-module.
- **Website** wirelessly controls motor direction, speed (10 -15m range).

Obstacles Faced



Negative Aspects

- Incorrect **diode** size on PCB
- **Gate driver legs** broke a day before demo
- Unprogrammable **ESP32** chip



5. Conclusion

Lessons Learned
Redesign Considerations

Conclusion

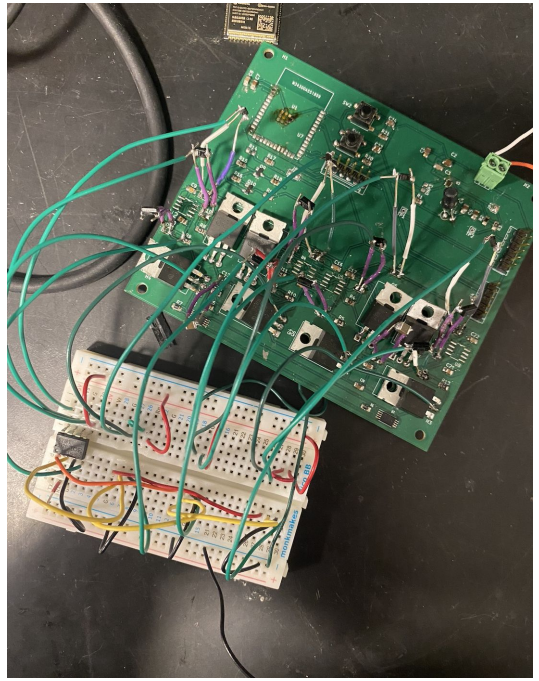
- Our PCB board was working. Unfortunately, the circuit board was shorted.
- Demoed our project with a prebuilt in H-bridge module.

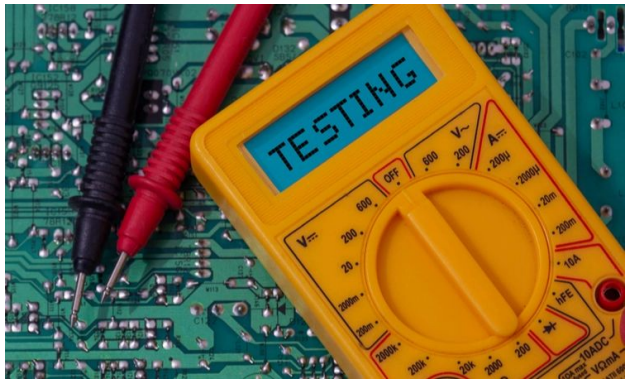
6. Recommendations for Further Work

Areas for Improvement

Areas for Improvement

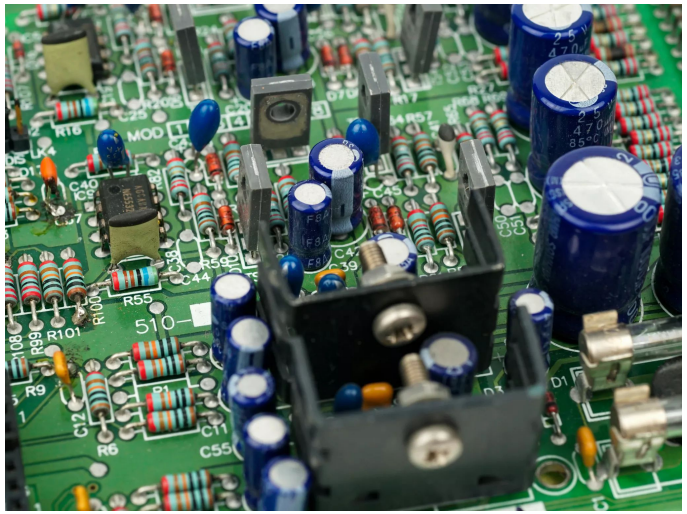
- Substituting the **correct gate drivers**
- **Optimize PCB** component layout
- Adding more **test points**

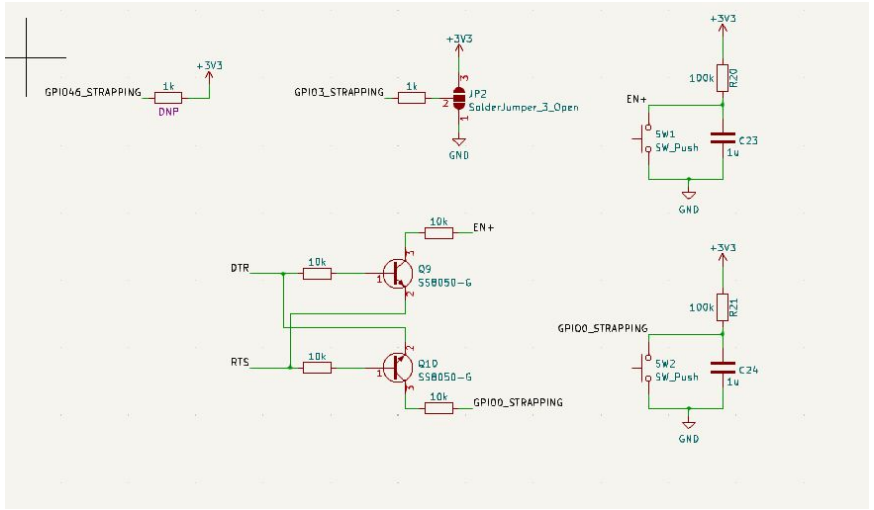




Areas for Improvement

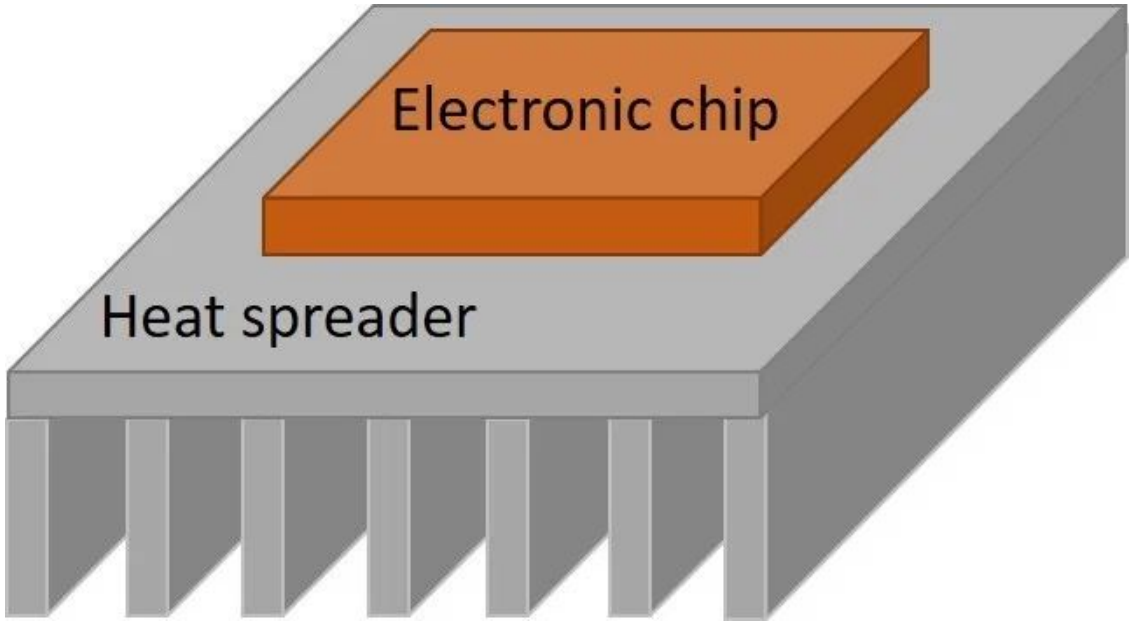
- Ordering a **surplus** of **key components** for back up plans
- Triple checking our PCB design before ordering
- Decoupling Capacitors

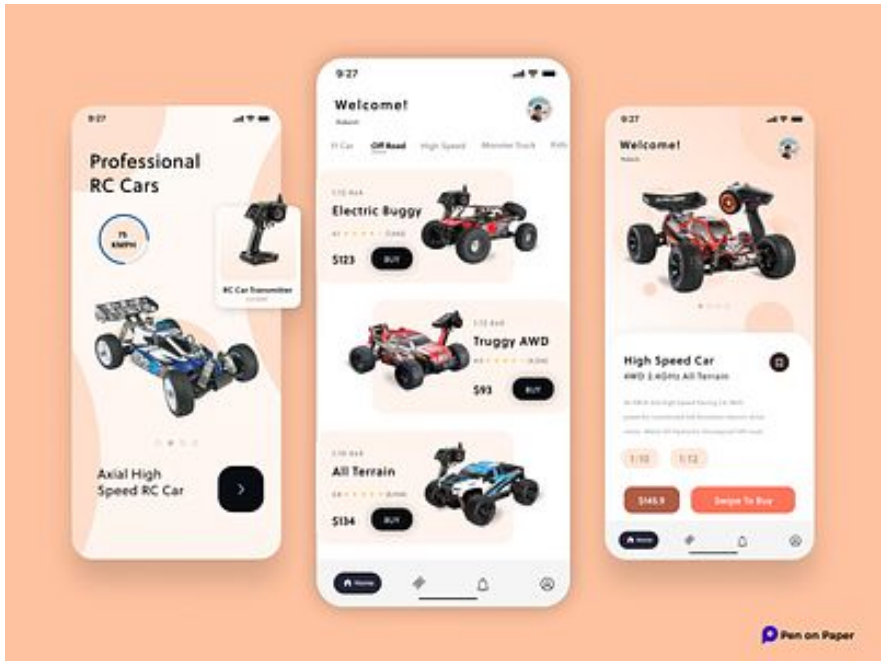




Areas for Improvement

- Improved Thermal Dissipation
- Incorporating **additional functions** onto the **webpage**
  - real-time RPM
  - current readings





Thank You!

Questions?