**CS437: Internet of Things**

**Lab1 Part1**

**Name:** David Orona

**NetIDs (include NetID of all group members):** orona2

**Late days used:** 0

**Video Link:**

[**https://youtu.be/kjVOseLAqIw**](https://youtu.be/kjVOseLAqIw)

**PCB:**

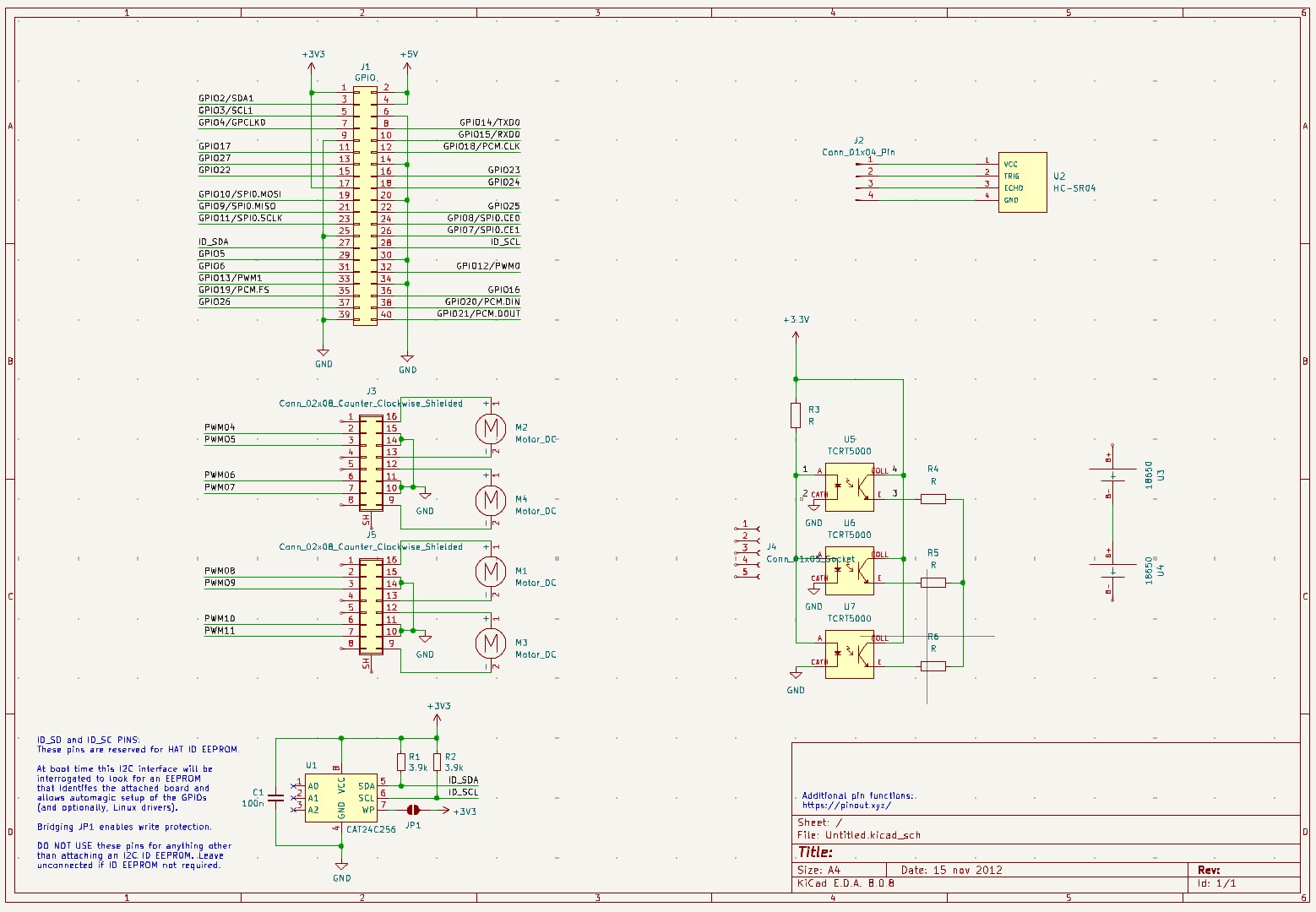
For the PCB portion, I designed a circuit to integrate the required motors, photo-interrupter, battery, and grayscale module. The schematic itself was, more or less, able to take form, as displayed below:   
  


Figure 1: PCB Schematic

However, issues arose when applying wiring and part selection. Specifically, I may not have applied the wires correctly or included too many vague components, leading to an insufficient etching. In addition, other factors and challenges could have applied, including:

* **Human Error in Wiring:** Some wires may not have been assigned to the correct headers, leading to improper connections.
* **Timing Issues:** The etching process may not have been optimized due to incomplete or incorrectly spaced traces.
* **Component Uncertainty:** The lack of specificity in some components caused ambiguity in their PCB footprint, leading to design inefficiencies.

Therefore, to improve future iterations, I will ensure:

1. Proper verification of wiring paths using circuit simulation tools before finalizing the design.
2. A thorough review of part specifications to prevent vague component selections.
3. Closer inspection of trace widths and etching parameters to guarantee precision.



Figure 2: PCB "Etching"

## **Design Consideration:**

The design considerations for this lab included structural integrity, power distribution, and sensor placement to ensure reliable obstacle avoidance and movement. The following aspects were taken into account:

* **Chassis Stability:** Ensuring the motor and wheel assembly were correctly aligned to avoid drivetrain issues.
* **Power Management:** Connecting the battery pack properly to the 4WD-HAT to ensure stable power delivery to the Raspberry Pi and motors. Often, my power would go out due to improper seating.
* **Sensor Placement:** Mounting the ultrasonic sensor at an optimal height and angle for effective obstacle detection.
* **Code Implementation:** Structuring the software to provide smooth transitions between movement and obstacle avoidance without abrupt stops.

**Chassis Assembly:**

The PiCar-4WD chassis was assembled following the SunFounder documentation, ensuring that all mechanical and electrical components were securely attached. Key assembly steps included:

1. **Motor Installation:** The four DC motors were mounted and properly wired to the motor driver board.
2. **Battery Pack Setup:** The battery pack was connected to the 4WD-HAT module, ensuring proper voltage regulation.
3. **Ultrasonic Sensor Mounting:** The ultrasonic module was fixed in a forward-facing position with the ability to scan for obstacles.
4. **Raspberry Pi Integration:** The Raspberry Pi was securely mounted and connected to the 4WD-HAT for control signal transmission.

The only main challenge was finding a way to route the camera ribbon (whether under the chassis or over it) so that it did not impede the sonar detector. Other issues like ensuring battery connection and cable management also played a part.

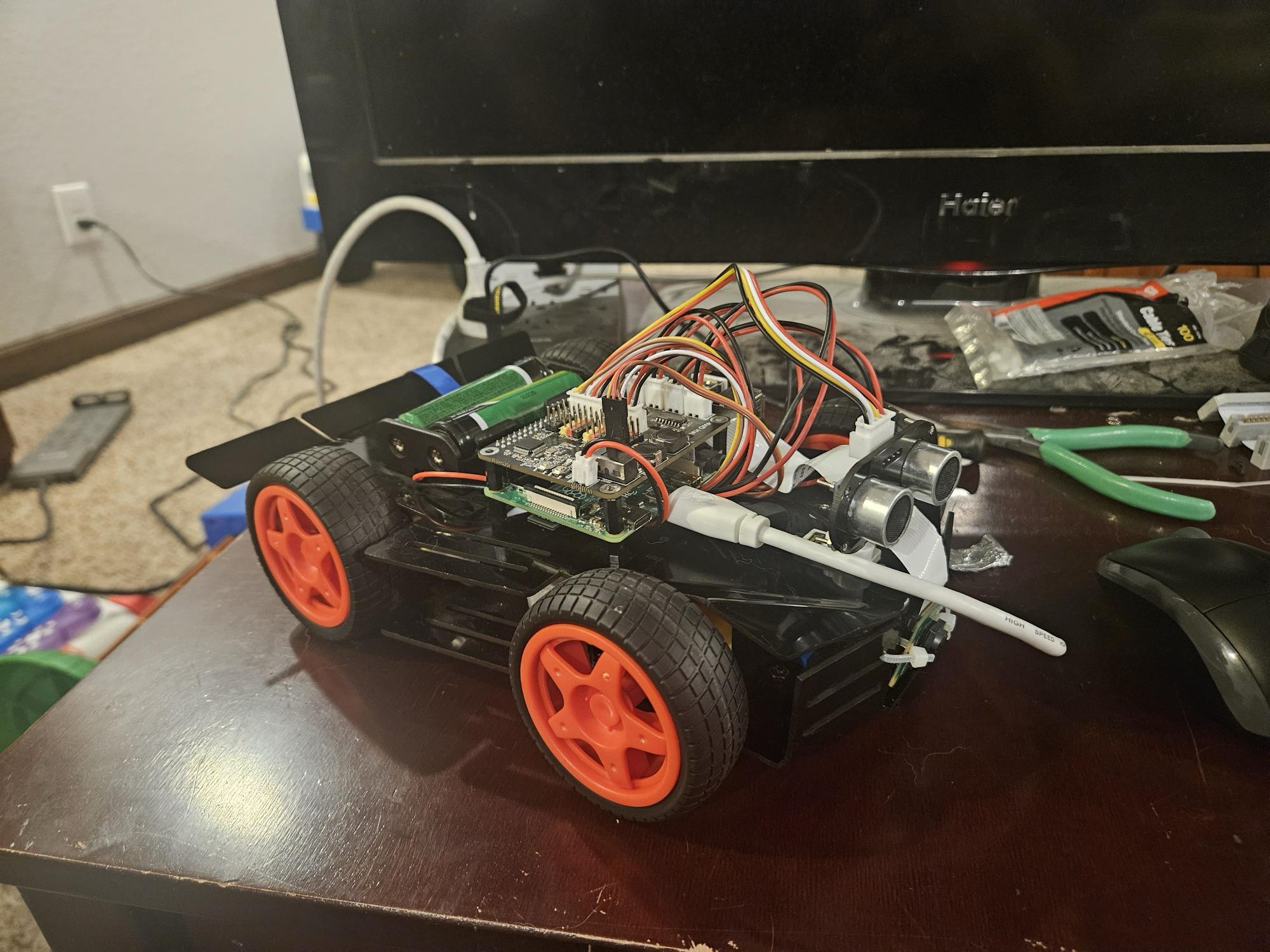
****

Figure 3: The Completed Car

**Naive Mapping:**

To implement naive mapping, the ultrasonic sensor was utilized to detect obstacles and measure distances. The sensor was programmed to continuously scan in front of the car and respond accordingly:

1. **Detection Threshold:** If an obstacle was detected within 10 cm, the car would stop.
2. **Response Mechanism:** Upon stopping, the car would:
   * Reverse for a brief moment.
   * Turn randomly (left or right).
   * Continue moving forward.

Challenges encountered:

* **Sensor Accuracy:** With a 10 cm tolerance, interference in sonar readings often led to false obstacle detection.
* **Motion Smoothing:** Abrupt stopping and reversing sometimes led to jerky movement.
* **Environmental Factors:** Flooring type and leveling affected sensor accuracy.

Future improvements will include:

* Adding noise filtering to improve ultrasonic readings.
* Fine-tuning movement parameters for smoother transitions.

**Naive Self-Driving**

The self-driving implementation focused on basic obstacle avoidance rather than advanced path planning. The logic followed:

1. **Move forward continuously until an obstacle is detected.**
2. **If an obstacle is found:**
   * Stop movement.
   * Reverse for a short distance.
   * Turn left or right at random.
   * Resume forward movement in the new direction.
3. **Loop continuously for ongoing navigation.**

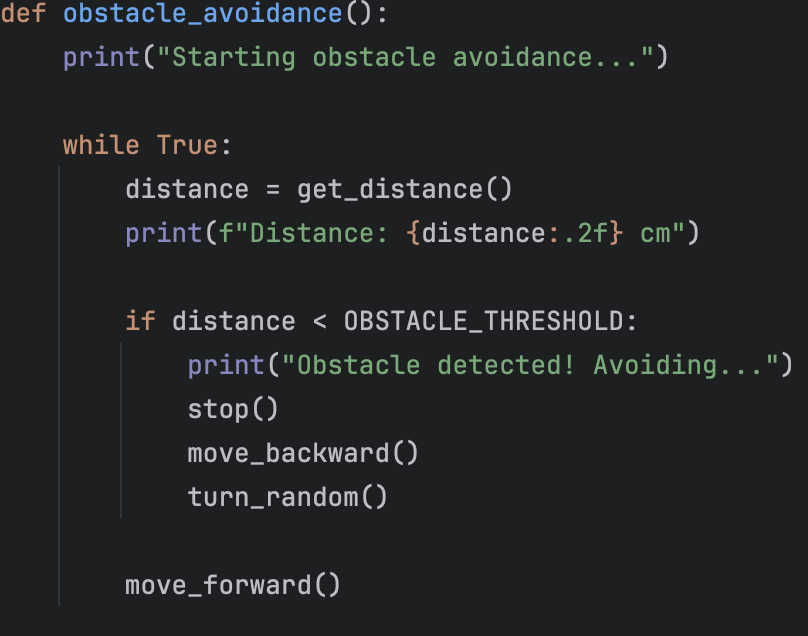


Figure 4: The "Main" Function

**Performance Evaluation:**

The car successfully avoided static obstacles, but struggled slightly with dynamic ones. Occasionally, repeated turning led to getting stuck in confined spaces and/or circling. Improvements in navigation logic, such as tracking past turns, could enhance this already promising performance.

|  |  |
| --- | --- |
| **Name** | **Contribution** |
| David Orona | Sole Contributor |