# Table Cleaning Robot: Autonomous Elevated Surface Cleaner

By
Ann Luo(hluo12@illinois.edu)
Bolin Pan(bolinp2@illinois.edu)
Yening Liu(yeningl2@illinois.edu)

Final Report for ECE 445, Senior Design, Spring 2025
TA: Jason Zhang

6 May 2025
Project No. 33

## Abstract

This report describes the design and development of an autonomous table-cleaning robot. The robot can operate on an elevated table surface without manual control. It is able to detect and avoid the edges of the table, move around obstacles, and clean the area beneath it. The system uses ultrasonic and infrared (IR) sensors, along with a servo motor, to sense both obstacles and table edges. A small cooling fan is used to collect dust and debris from the table surface. This report explains the overall design, the main components, and the testing and verification steps taken to make sure the robot works as expected.

# 1. Introduction

## 1.1 Background

In residential, commercial, and dining environments, table cleaning is a repetitive task that must be performed frequently and can be time-consuming. Dust, food particles, and spills accumulate quickly, and cleaning is typically done manually. Although floor-cleaning robots have improved the efficiency of floor cleaning, there are currently few automated solutions available for cleaning elevated surfaces such as tables. In places like offices and restaurants, where tables require regular cleaning, an autonomous table-cleaning robot could significantly reduce manual effort, lower labor costs, and improve overall efficiency.

However, designing a robot for table cleaning has greater challenges than floor-cleaning robots. A table-cleaning robot must be capable of detecting and avoiding edges to prevent falls, moving around objects such as cups and plates, and cleaning effectively without colliding with the items. Due to the limited space and frequent presence of objects on tables, more precise control and obstacle detection are required.

To address these challenges, we developed an autonomous table-cleaning robot capable of operating safely on elevated surfaces. The robot uses ultrasonic and infrared sensors to detect table edges and obstacles, allowing it to avoid collisions and prevent falls. A small vacuum fan collects dust and debris into a sock. By combining edge detection, obstacle avoidance, and debris collection, the robot provides a reliable and efficient solution for maintaining clean tabletops with minimal human effort.

## 1.2 High-Level Requirements

- Edge Detection and Fall Prevention: The robot must detect table edges using ultrasonic sensors and respond by stopping or changing direction within 2 cm of the edge to prevent falling.

- Debris Collection Efficiency: The cleaning system must remove at least 90% of dust, crumbs, and small debris from a 60 cm x 60 cm table surface in a single cleaning cycle and store the collected debris in a removable bin that holds dirt for at least three full cleaning cycles.

- Obstacle Avoidance and Object Interaction: The robot must detect and navigate around objects as small as 5 cm in diameter and as large as 20 cm in diameter with at least 95% success. It must avoid collisions and prevent pushing objects off the table.

# 2. Design

## 2.1 Block Diagram

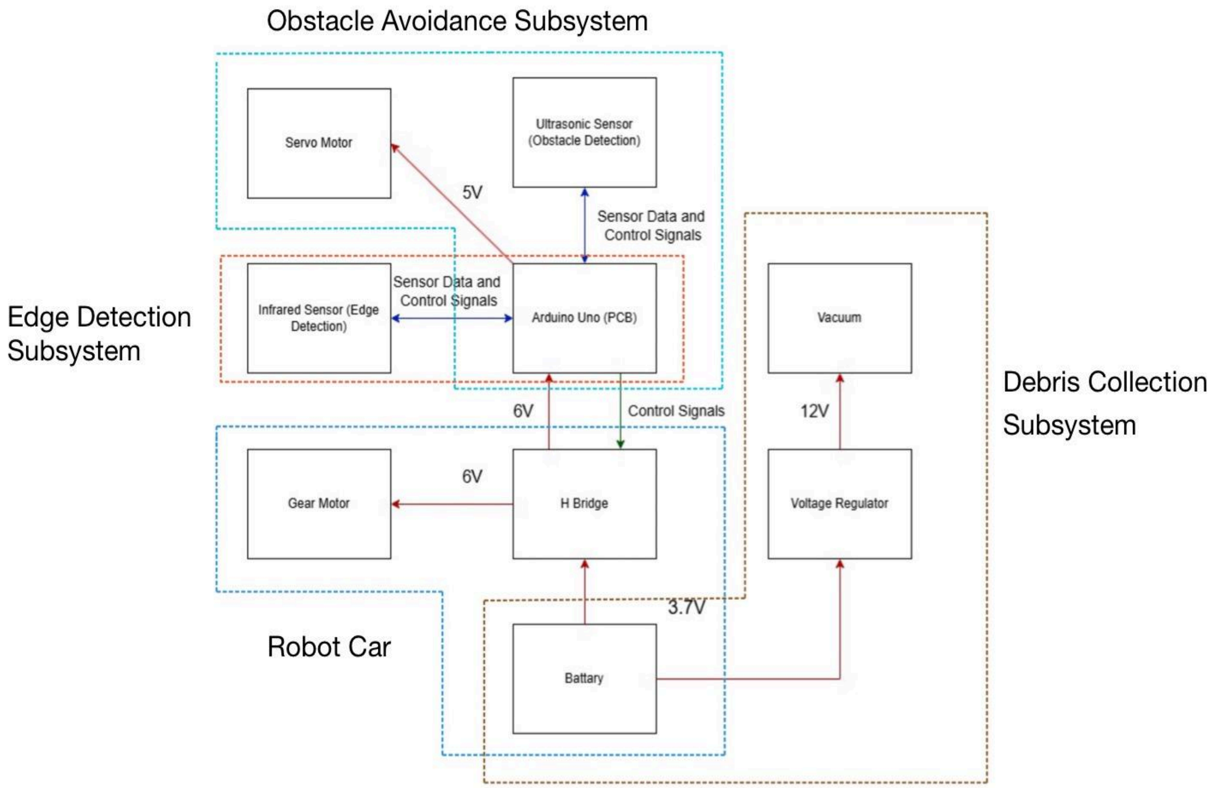


Figure 1. Block Diagram of the Automated Table Cleaning Robot

## 2.2 Subsystem Overview

Our design includes three primary subsystems: edge detection and fall prevention, debris collection, and obstacle detection. An Arduino Uno [1] is used as the development board to control the motion and behavior of the robot.

The edge detection and fall prevention subsystem is essential for ensuring that the robot operates safely on elevated surfaces. It uses two infrared (IR) sensors [2] to continuously monitor and detect the edges of the table in real time. When an edge is detected, the system immediately stops the robot and initiates a turn to prevent it from falling. If no edge is detected, the robot continues moving forward. This subsystem plays a crucial safety role, as failure to detect edges accurately could lead to the robot falling and becoming damaged.

The debris collection subsystem is responsible for cleaning the table surface by removing dust, crumbs, and other small particles. It consists of a mini cooling fan paired with a sock that functions as a lightweight and flexible dust collection bin. The fan runs continuously while the robot is in motion, allowing for effective and consistent debris collection during the cleaning process.

The obstacle detection subsystem enables the robot to detect and respond to objects such as cups, plates, and utensils. It utilizes an ultrasonic sensor [3] mounted on a servo motor to scan the surrounding area in real time. When an obstacle is detected, the servo rotates the sensor to check distances on both the left and right sides. The robot then turns in the direction with the greater available space. This subsystem enhances cleaning efficiency and allows the robot to navigate safely around objects on the table.

## 2.3 Subsystem Design and Verification

### 2.3.1 Edge Detection Subsystem

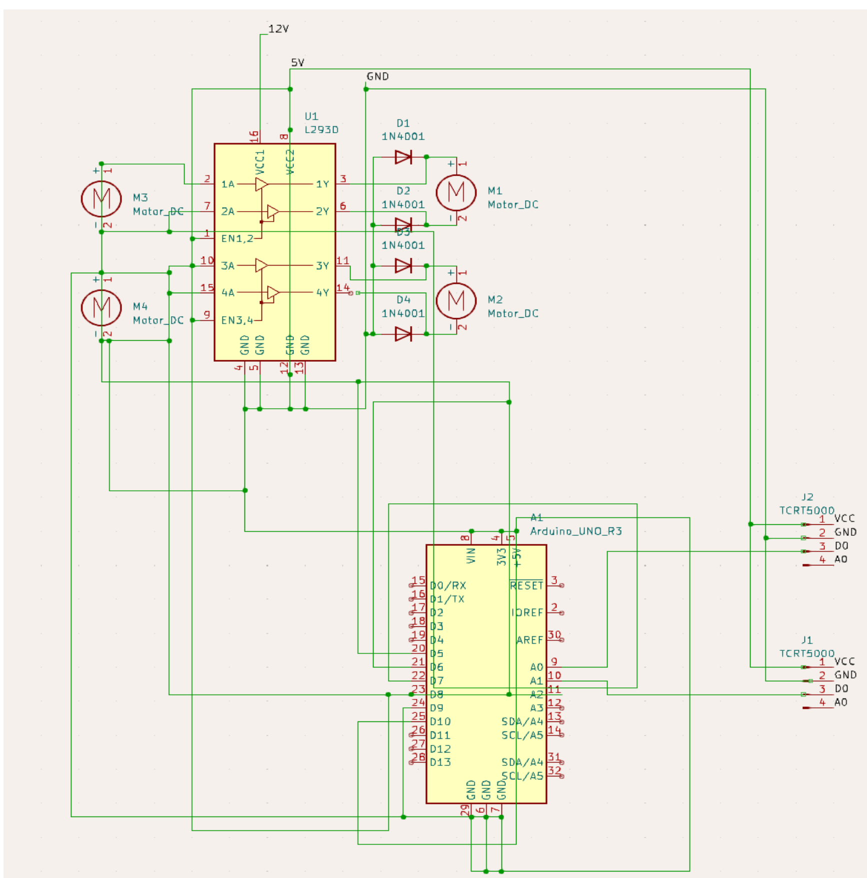


Figure 2. Schematic of the Edge Detection Subsystem

The edge detection and fall prevention subsystem is a critical safety component that ensures the robot does not fall off the table by detecting edges in real time. Failure to accurately detect edges could result in the robot falling and becoming damaged. This subsystem interacts directly with the motion control system, adjusting the robot's movement based on its proximity to table edges.

To achieve reliable edge detection, we use two IR sensors, which detect edges and help control the robot's movement by stopping or changing its direction when necessary. IR sensors were chosen over other options because they are not affected by ambient light, making them reliable in both bright and dark environments. They also provide fast and accurate distance measurements, allowing the robot to quickly detect changes in surface height and identify edges effectively.

The system classifies an edge when the IR sensors detect a sudden increase in distance from less than 5 cm to greater than 20 cm within two consecutive sensor readings, which prevents false triggers. The ultrasonic sensors will continuously send signals to the Arduino Uno microcontroller. Once an edge is confirmed, the microcontroller will send PWM signals to the motor driver (L293D) to either stop or turn when an edge is detected, allowing the robot to stop or change direction.

| Requirements | Verification |
|---|---|
| The IR sensors must detect a table edge when the distance reading suddenly increases within two consecutive sensor readings | Allow the robot to move at a controlled speed from different distances toward the table edge. Use a logic analyzer to measure sensor readings and confirm that the edge is correctly detected. |
| The system must send a STOP or TURN command within 100ms of detecting an edge. | Use an oscilloscope to measure the time delay between edge detection and the execution of the STOP or TURN command. Verify that the command is issued within 100ms to ensure proper functionality. |
| The robot must avoid falling off the table in at least 95% of test cases. | Conduct at least 50 trials. Record the number of successful edge detections |

| | |
|---|---|
| | and confirm that the robot avoids falling in at least 95% of cases. |
| The ultrasonic sensors must function correctly under different lighting conditions. | Test the robot in various lighting environments, including bright daylight and complete darkness, to ensure edge detection remains unaffected by ambient light. |
| The robot must be able to detect edges at different approach angles and movement directions. | Position the robot at various angles and movement directions. Verify that the robot successfully detects the edge and stops or changes direction accordingly. |

Table 1. R&V Table for Edge Detection Subsystem

We ran 50 trials testing the functionality, and 48 of which were successful. The two failure situations are similar, where the robot was stuck at the edge of the table without falling to the ground, but unable to move either, which ensures the basic safety requirements and could be improved through re-designing the mechanics and dimension of the robot.

### 2.3.2 Obstacle Avoidance Subsystem

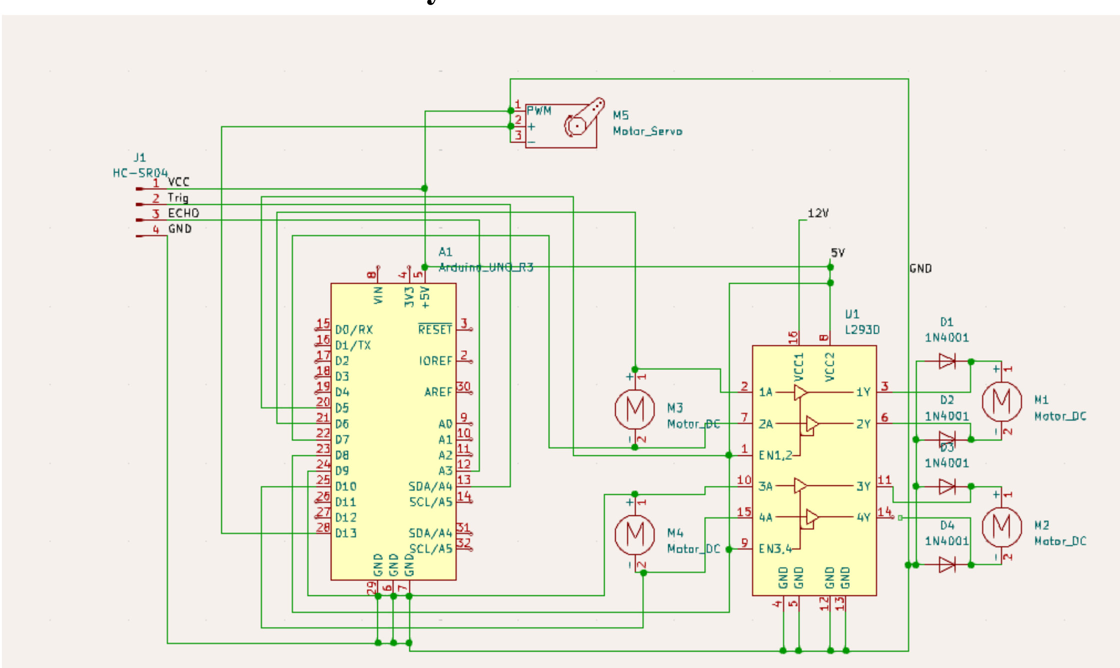


Figure 3. Schematic of the Obstacle Avoidance Subsystem

The obstacle avoidance subsystem identifies objects on the table using an ultrasonic sensor mounted on a servo motor. The sensor has a ranging distance of 2 cm to 400 cm with a resolution of 0.3 cm [4]. It measures distances and sends processed signals to the central control module to prevent collisions. The system determines how to move around the objects to clean area near them.

The ultrasonic sensor is used to help control the robot's movement by stopping or changing its direction when necessary. The ultrasonic sensor provides fast and accurate distance measurements, allowing the robot to quickly detect the obstacles and avoid them effectively.

The system classifies an obstacle when an ultrasonic sensor detects a sudden decrease in distance within two consecutive sensor readings. It will send signals to the Arduino Uno microcontroller, which processes the data in real time. Once an obstacle is confirmed, the microcontroller sends a corresponding command to the motor driver, allowing the robot to stop or change direction.

| Requirements | Verification |
|---|---|
| The ultrasonic sensor needs to detect the obstacle and let the Arduino Uno to stop the gear motor. | Place testing obstacles and set the input voltage to 3.2V and 5.5V respectively to test if the robot can detect and stop before hitting the obstacle under minimum and maximum power. |
| To ensure its stability, the robot must avoid hitting obstacles in at least 95% of the test cases | Repeat tests for 50 times under the regular operating voltage, and record the outcome, including cases of changing angles between the robot and the testing obstacles, and airs with various humidity and temperature. |

Table 2. R&V Table for Obstacle Avoidance Subsystem

We ran 50 trials to test its capability to avoid various obstacles, and the 50 successful trials proved its ability to handle the obstacle avoidance task.

### 2.3.3 Debris Collection Subsystem

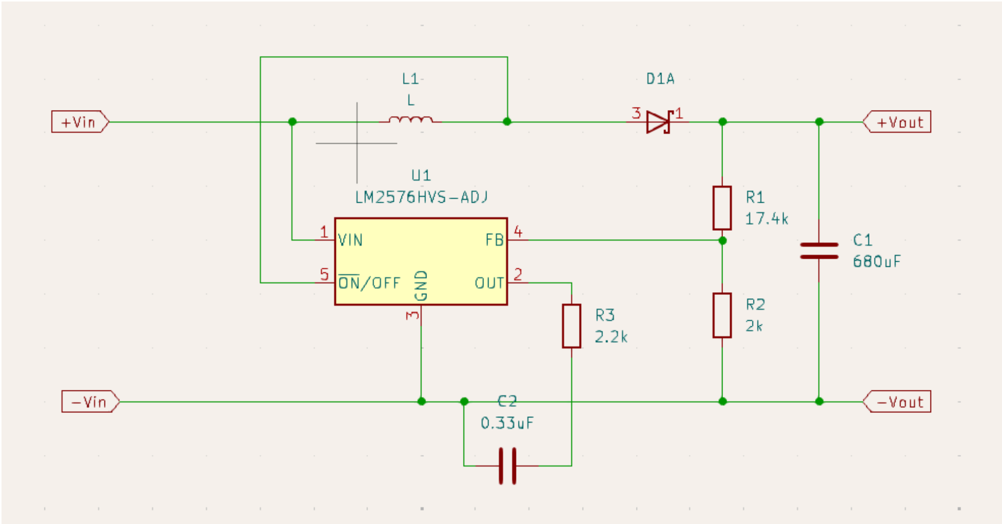


Figure 4. Schematic of the Debris Collection Subsystem

The debris collection subsystem is responsible for cleaning the table by vacuuming dust and crumbs into a sock. It consists of a mini cooling fan and a sock as the collection bin. When the robot is moving, the vacuum will stay on. The motion control subsystem will control the speed and direction of the robot to ensure proper table coverage.

The cooling fan operates at 12V DC, which requires a voltage regulator with our power system, while delivering a strong airflow of 1.2 L/min, ensuring effective debris collection. Additionally, its lightweight structure minimizes the load on the robot, improving maneuverability and battery efficiency. The pump's reliable operation and ease of integration make it well-suited for continuous cleaning while the robot is in motion.

The Debris Collection Subsystem receives a 12V input from the voltage regulator, as the coolingfan operates at 12V while the system's battery voltage is 3.7V. The voltage regulator ensures that the fan receives a stable supply for consistent suction performance. The only electrical input to the vacuum pump is the regulated 12V. There are no outputs from the subsystem, and all debris collected by the vacuum pump will be stored inside the sock. The bin will be cleaned after the cleaning process.

| Requirement | Verification |
|---|---|
| The debris collection subsystem must vacuum at least 90% of debris in a given area. | Measure before and after debris mass to confirm 90% collection efficiency. |
| The input voltage to the Debris Collection System should be 12V (±0.4V). | Measure input voltage using a multimeter and an oscilloscope. |

Table 3. R&V Table for Debris Collection Subsystem

## 2.4 PCB Design

The Printed Circuit Board (PCB) designed for this project replicates the core functionality of the Arduino Uno3, enabling more compact integration in our embedded system. Several approaches were considered and two versions of the design and layout had been made. The initial version closely followed the official Arduino reference design, while the second version introduced layout optimizations, improved power handling, and a more reliable USB-to-serial interface.

In both versions, the microcontroller used is ATmega328P-PU in a DIP package, paired with a 16MHz crystal oscillator and two 22pF capacitors. The USB-to-serial conversion was handled by the ATmega16U2, but we then switched to CH340G converter for simplification and cost reduction. The linear regulators, NCP1117 and LP2985, were used to convert VIN to +5V and +3.3V respectively in the first version, and the reworked power supply section uses LM1117-5.0 for stable +5V output in order to reduce component redundancy. Moreover, the reduced number of decoupling capacitors and pull-up resistors through shared use makes the size of the board closer to the actual industry-produced Uno3 development board.

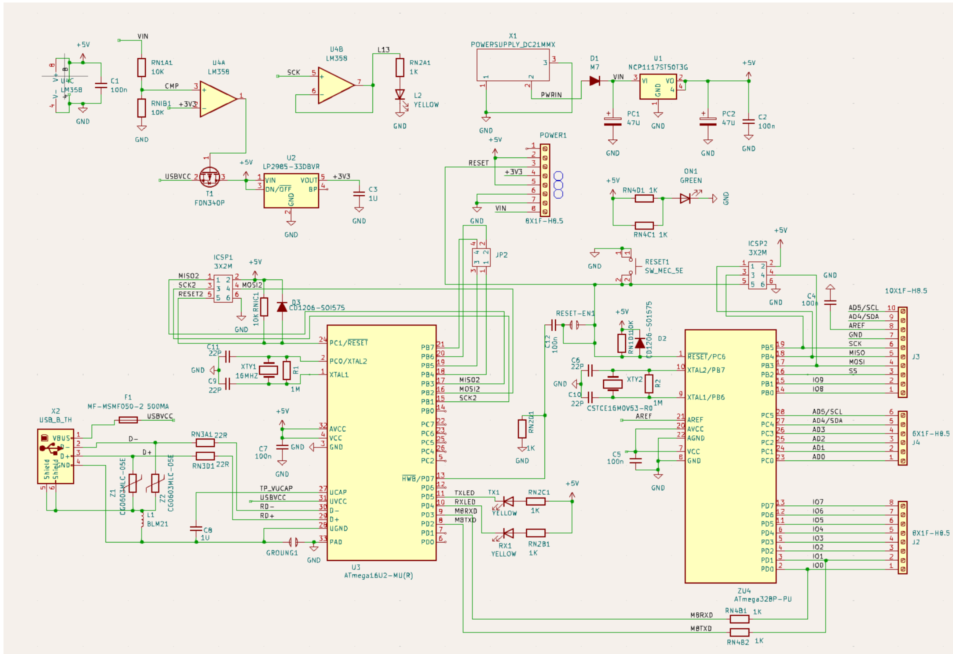

Figure 5. Schematic of Our PCB

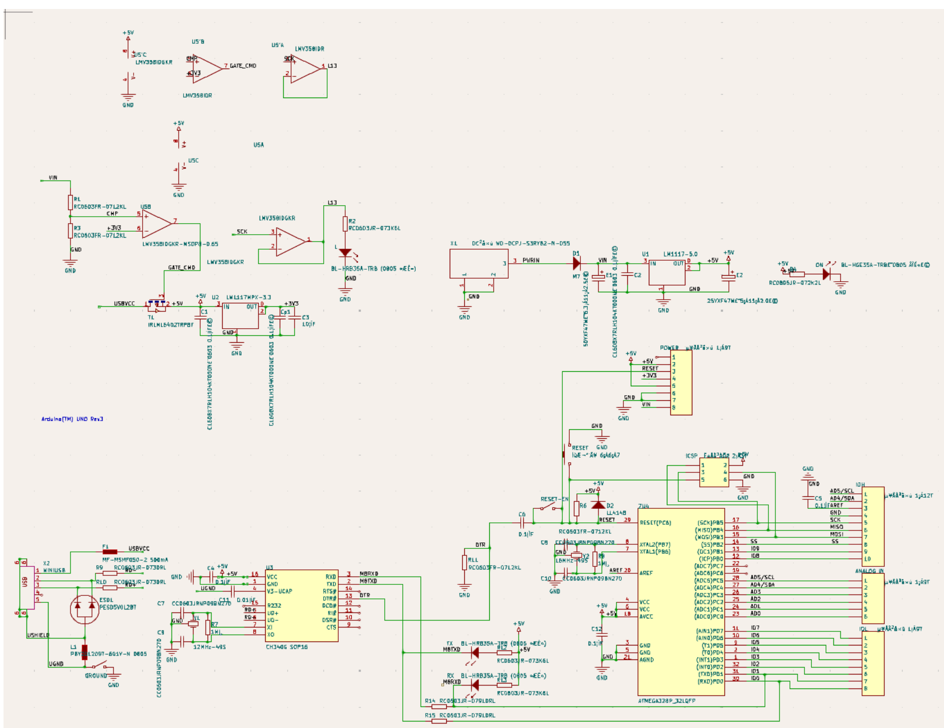

Figure 6. Schematic of Our PCB

## 2.5 Tolerance Analysis

-The robot runs Edge Detection checks at a certain sampling frequency (In this case 40Hz)

-The Ultrasonic sensor can detect a drop-off at nominal $d_s$ = 2 cm from the edge with a typical uncertainty of measurement e = ±3mm.
-The robot moves at v = 5.65 cm/s (0.0565 m/s)
-Reaction time $t_r$ = 0.2s (at 5Hz)
-Deceleration rate a = 0.5 $m/s^2$
With such parameters, the numerical calculation is below:
The distance during reaction time:

$$d_r = v * t_r = 0.0565 m/s * 0.025 s = 0.0014125 m$$

The braking distance:

$$d_b = v^2/2a = (0.0565)^2/(2 * 0.5) = 0.00319225 m$$

The distance needed to stop:

$$d_{stop} = d_r + d_b = 0.0014125 + 0.00319225 = 0.00460475 m$$

Distance of effectiveness:

$$d_{effective} = d_s - e = 0.02 - 0.003 = 0.017 m$$

Since 0. 00460475m<< 0.017m, theoretically, the parameter of parts allows the robot to detect the edge of the table or obstacles and stop to avoid any accidents.

# 3. Cost and Schedule

## 3.1 Cost Analysis

### 3.1.1 Salary

| Team Member | Hourly Rate | Hours | Total |
|---|---|---|---|
| Ann Luo | $38 | 77 | $7315 |
| Bolin Pan | $38 | 77 | $7315 |
| Yening Liu | $38 | 77 | $7315 |
| Total | | | $21945 |

Table 4. Salary Assumption

### 3.1.2 Parts Cost

| Part Number | Quantity | Total Price |
|---|---|---|
| ATMEGA16U2-MUR | 10 | $30.10 |
| SD1206S100S1R0 | 30 | $9.78 |
| EEE-1CA470WR | 20 | $3.72 |
| CG0603MLC-05E | 20 | $5.54 |
| 150060YS55040 | 30 | $4.23 |
| 150060VS55040 | 10 | $1.41 |
| MF-MSMF050-2 | 20 | $2.64 |
| CSTNE16M0V530000R0 | 20 | $3.56 |
| USB-B-S-F-B-TH | 3 | $3.00 |
| C0603C220J5GACTU | 20 | $1.24 |
| LM358DT | 20 | $2.24 |
| NCP1117ST50T3G | 10 | $6.40 |
| ATMEGA328P-PU | 5 | $13.05 |

| | | |
|---|---|---|
| PJ-006B-SMT-TR | 5 | $3.65 |
| 1N4148W-TP | 20 | $1.32 |
| ATMEGA328PB-AU-ND | 5 | $7.35 |
| ATMEGA328PB-MUR | 5 | $7.25 |
| CC0603JRNPO9BN270 | 20 | $0.56 |
| LL4148 | 5 | $0.5 |
| ABLS-12.000MHZ-B4-T | 10 | $1.96 |
| LMV358IDGKR | 5 | $1.70 |
| RC0603FR-0712KL | 20 | $0.18 |
| ABLS-16.000MHZ-B4-T | 10 | $1.96 |
| RC0603JR-073K6L | 20 | $0.16 |
| BL-HGE35A-AV-TRB | 5 | $1.00 |
| RC0805JR-072K2L | 10 | $0.12 |
| 25YXF47MEFCT15X11 | 5 | $1.00 |
| EEU-FR1H470B | 5 | $1.70 |
| RC0603JR-07910RL | 20 | $0.16 |
| PESD5V0L2BT,215 | 5 | $1.80 |
| MF-MSMF050-2 | 5 | $0.72 |
| IRLML6402TRPBF | 10 | $2.30 |
| RC0603JR-0730RL | 20 | $0.16 |
| Total | 428 | $122.46 |

Table 5. Parts Costs

We assume each team member’s salary to be $7,315, based on an hourly rate of $38. The total labor cost amounts to $21,945. Additionally, the total cost of parts is $122.46. Combining both, the overall project cost comes to $22,067.5.
Conclusion

### 3.2 Schedule

| Date | Task | Team Member |
|---|---|---|
| Feb. 24-Mar. 2 | Order components | Ann |
| | Work on schematics of PCB | Bolin |
| Mar.3 - Mar.9 | Schematic design | Ann |
| | Work on PCB wiring, breadboard | Bolin |
| | Write design document | All |
| Mar.10 - Mar.16 | Finish PCB before 2nd round due | Bolin |
| | Breadboard Demo | All |
| | Write code | Ann |
| | Update design document | Yening |
| Mar.17 - Mar.23 | Spring Break | All |
| Mar.24 - Mar.30 | Write code | Ann |
| | Debug PCB | Bolin |
| | Update design document | Yening |
| Mar.31 - April 6 | Individual Progress Report | All |
| | Order 3rd PCB if necessary | Bolin |
| | Test functionality | Ann, Yening |
| April 7- April 13 | Finalize the project | All |
| April 14-April 20 | Team Contract Assessment | All |

| | | |
|---|---|---|
| | Finalize the project | All |
| April 21-April 27 | Mock Demo | All |
| April 28 - May 4 | Final Demo | All |
| | Final Papers | All |
| | Project Video | All |
| May 5 - May 9 | Final Presentation | All |
| | Final Papers | All |

Table 6. Team Schedule

# 4. Ethics and Safety

The design, development, and deployment of the Table Cleaning Robot must adhere to ethical principles to ensure safety, reliability, and responsible engineering practices. This section assesses ethical issues and safety concerns, drawing guidance from the IEEE Code of Ethics.

## 4.1 Ethical Considerations

1. Public Safety and Welfare
   According to IEEE code #1 [5]: "Hold paramount the safety, health, and welfare of the public", public safety and welfare must be ensured. The Table Cleaning Robot must operate safely without causing harm to users, pets, or property. It must be designed to prevent falls and avoid pushing objects off the table so that no one will get hurt during the cleaning process. Therefore, rigorous testing for edge detection, fall prevention, and obstacle detection is required.
2. Honesty and Transparency in Design
   According to IEEE code #5: "Be honest and realistic in stating claims or estimates based on available data", honesty and transparency in design are required. The capabilities of the robot should be accurately presented. If the robots have limitations, the limitations must be clearly disclosed.
3. Privacy Protection
   According to IEEE code #4: "Avoid unlawful conduct in professional activities", we as designers should notice the importance of privacy protection. When we use

visual sensors, we must ensure that data is encrypted, avoid storing unnecessary information, and comply with privacy regulations.

4. Fairness and Non-Discrimination
   The project must not create biased or discriminatory outcomes. We will conduct bias testing and ensure inclusivity in design and deployment.

## 4.2 Safety Considerations

1. Mechanical and Electrical Hazards
   The robot has moving parts that could cause injury if mishandled. Electrical components, particularly lithium-ion batteries, pose risks of overheating or short-circuiting. Implementation of protective casing around moving parts can mitigate the hazards.
2. Fall Prevention and Object Handling Risks
   The robot must reliably detect table edges to avoid falling and prevent damage to itself and surrounding objects. If the robot attempts to move objects, it could accidentally push items off the table. Therefore, high-precision edge detection systems and object detection systems should be strictly tested. Also, implementing redundant safety measures will mitigate risks of failure in mission-critical applications.
3. Safe Charging and Energy Efficiency
   The charging station must be designed to prevent overheating and electrical fires. Therefore, auto-shutoff is recommended.

## 4.3 Regulatory Compliance

1. Battery Safety

   If the robot uses a lithium-ion battery, it must meet UN/DOT 38.3 certification to ensure safe transport and usage.

2. Campus and Lab Safety Policies

   If deployed in university environments, the robot must follow campus safety protocols, especially concerning fire hazards, electrical devices, and laboratory usage.

## 4.4 Mitigation Strategies

To avoid ethical and safety breaches, the following strategies will be implemented:

1. **Regular Ethical Audits**: Conduct periodic reviews of the project's compliance with ethical guidelines.
2. **Risk Assessments**: Identify and mitigate potential hazards through systematic risk management frameworks.
3. **Stakeholder Involvement**: Engage with users, regulatory bodies, and ethicists to ensure accountability.
4. **Transparency and Documentation**: Maintain clear records of decisions, safety tests, and compliance measures to facilitate oversight.

By integrating these ethical principles, safety measures, and regulatory standards, the Table Cleaning Robot will be safe, reliable, and responsibly engineered. Compliance with the IEEE Code of Ethics ensures that the product prioritizes public welfare, environmental responsibility, honesty, and fairness.

## References:


[1] *Arduino Uno R3*, Arduino. [Online].
Available:https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

[2] *TCRT5000 Infrared Reflective Sensor IR*, [Online]. Available:
https://soldered.com/productdata/2022/03/Soldered_TCRT5000_datasheet.pdf?srsltid=AfmBOop2P1qibLg5pNOAAEVvJQYWx2t9zwBiBxEZLxJ1RNyRgRTU_34s

[3] *Ultrasonic Distance Sensor - 5V (HC-SR04)*, Sparkfun Electronics. [Online].
Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[4] *HC-SR04 Ultrasonic Module Distance Sensor*, Sparkfun Electronics. [Online].
Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[5] *IEEE - IEEE Code of Ethics*. [Online]
www.ieee.org/about/corporate/governance/p7-8.html. Accessed 7 Mar. 2025.

# Appendix A

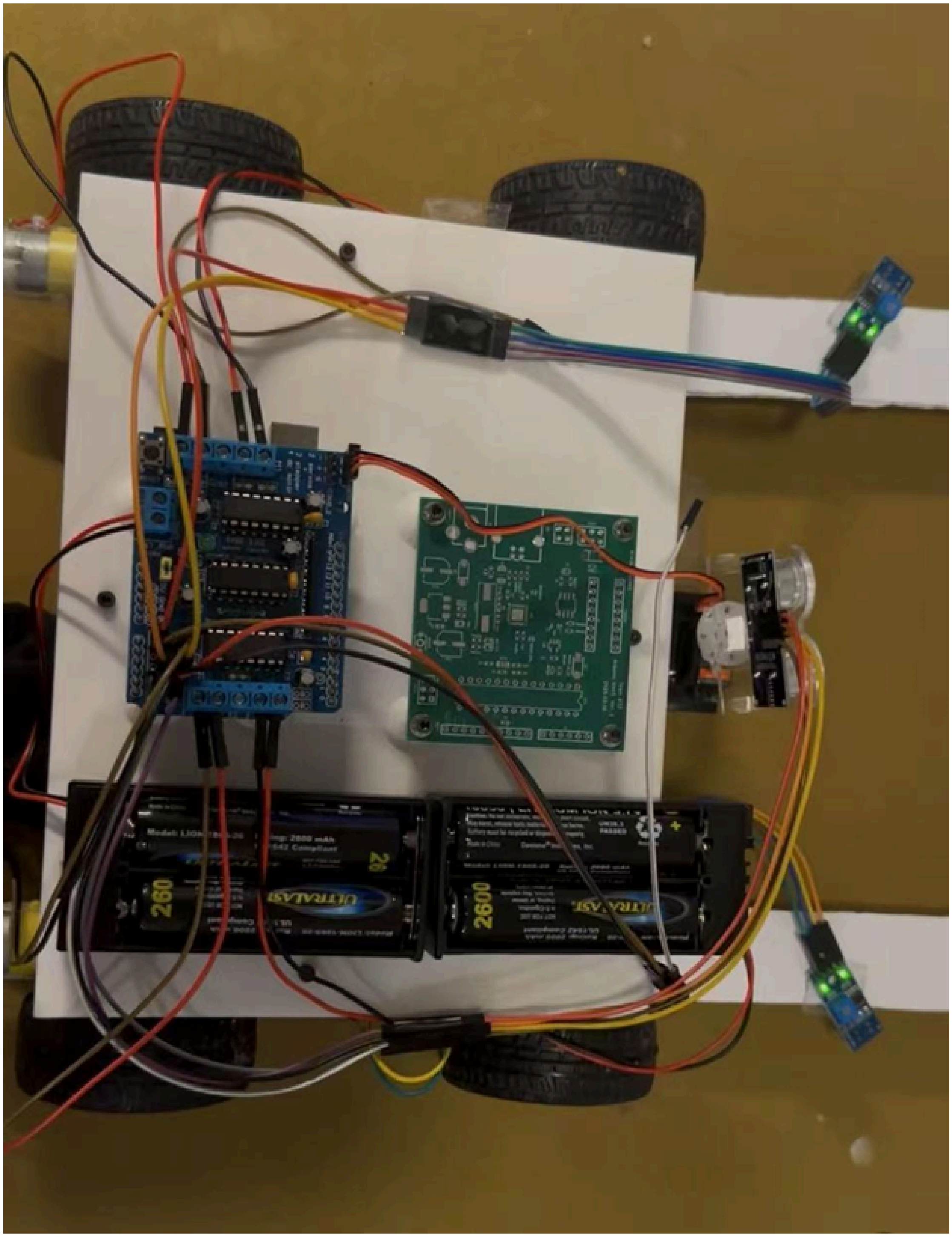

Figure 7. Edge Detection Subsystem

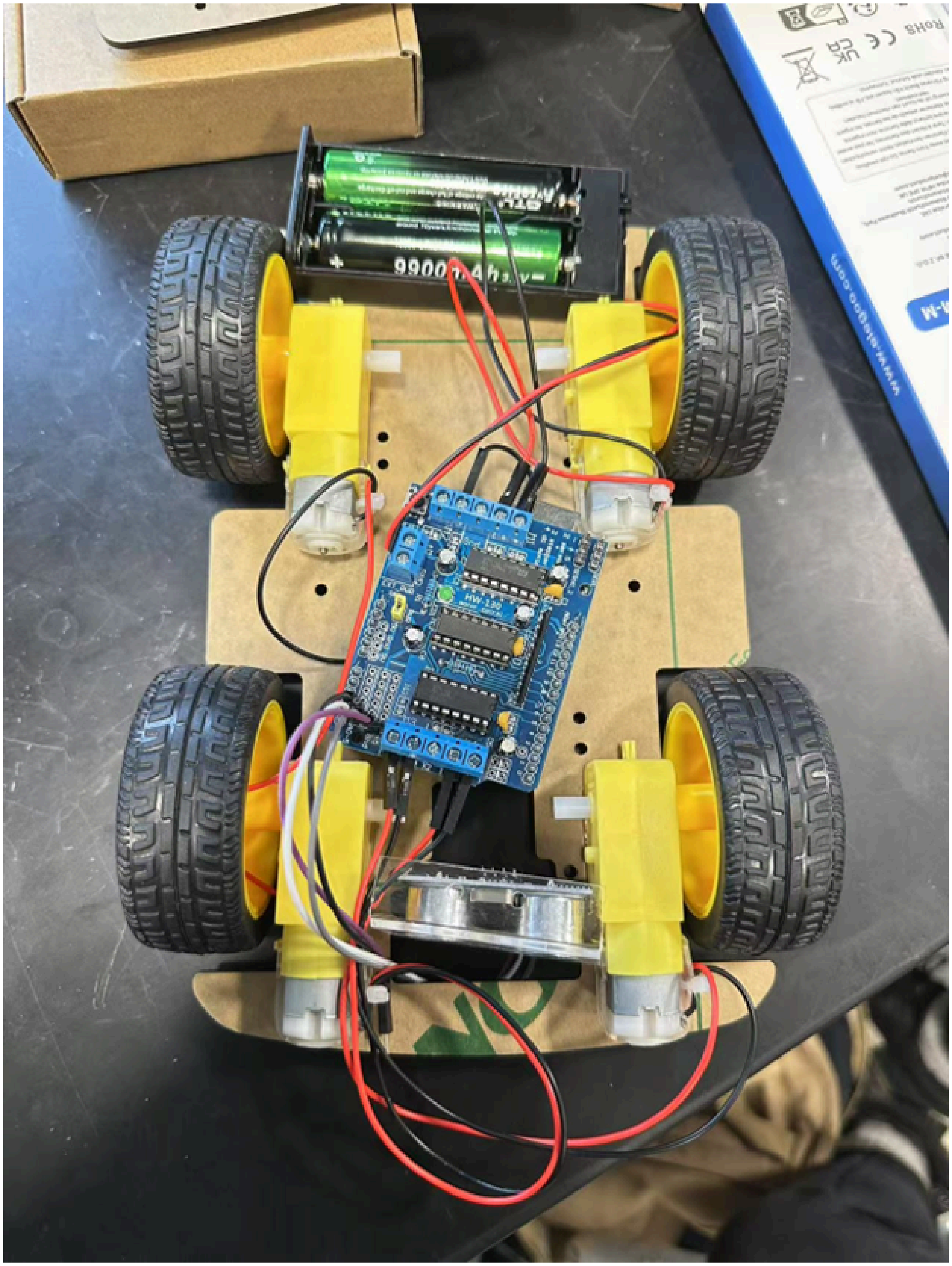

Figure 8. Obstacle Avoidance Subsystem

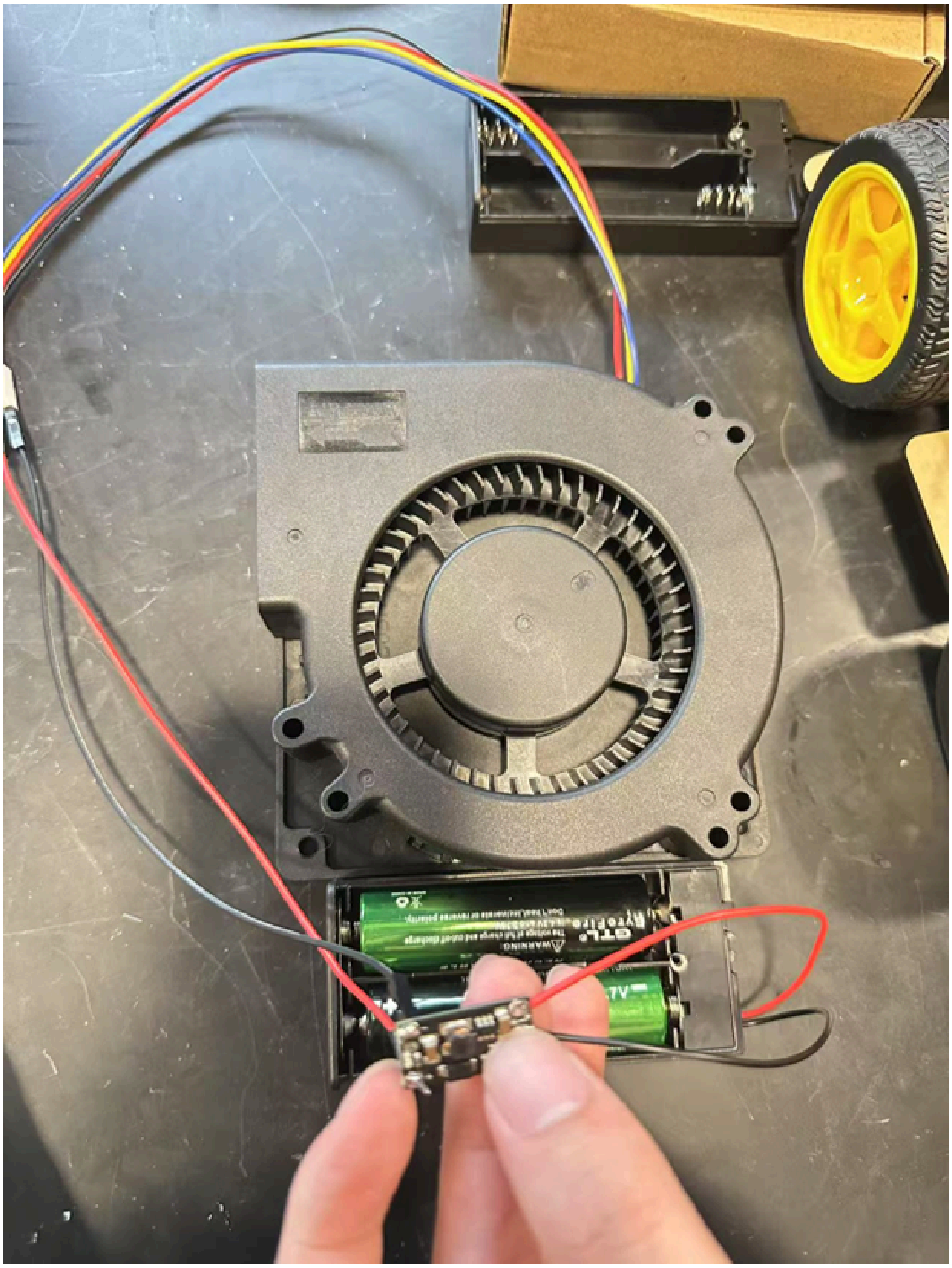

Figure 9. Debris Collection Subsystem

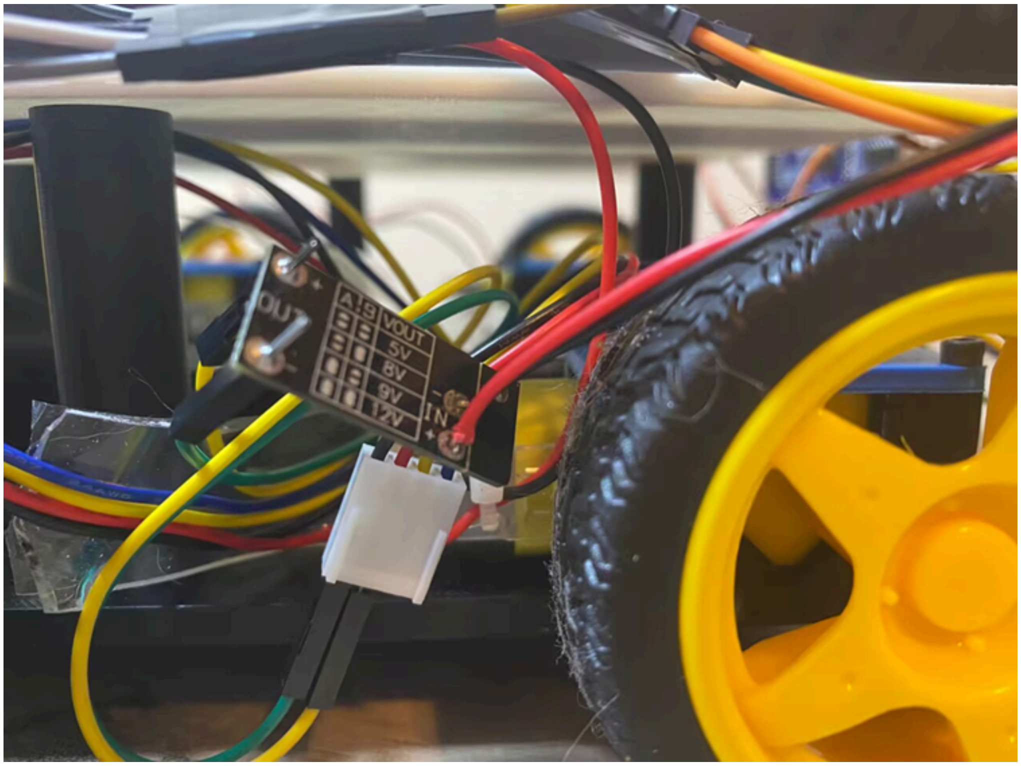


Figure 10. Voltage Regulator

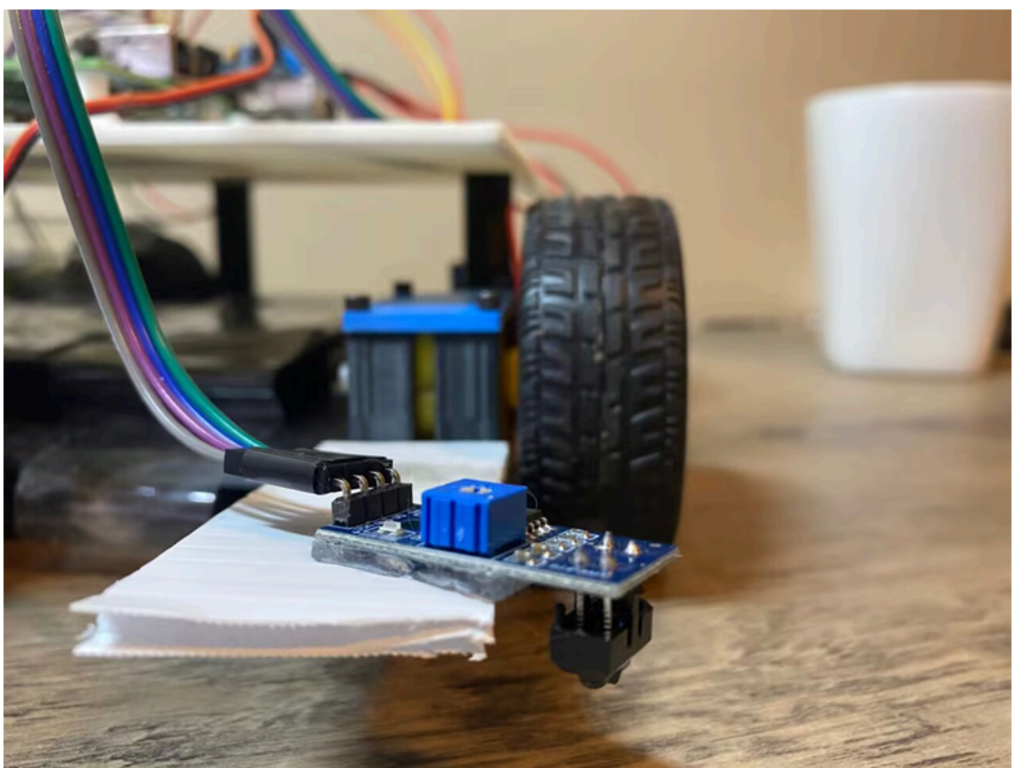

Figure 11. IR Sensor

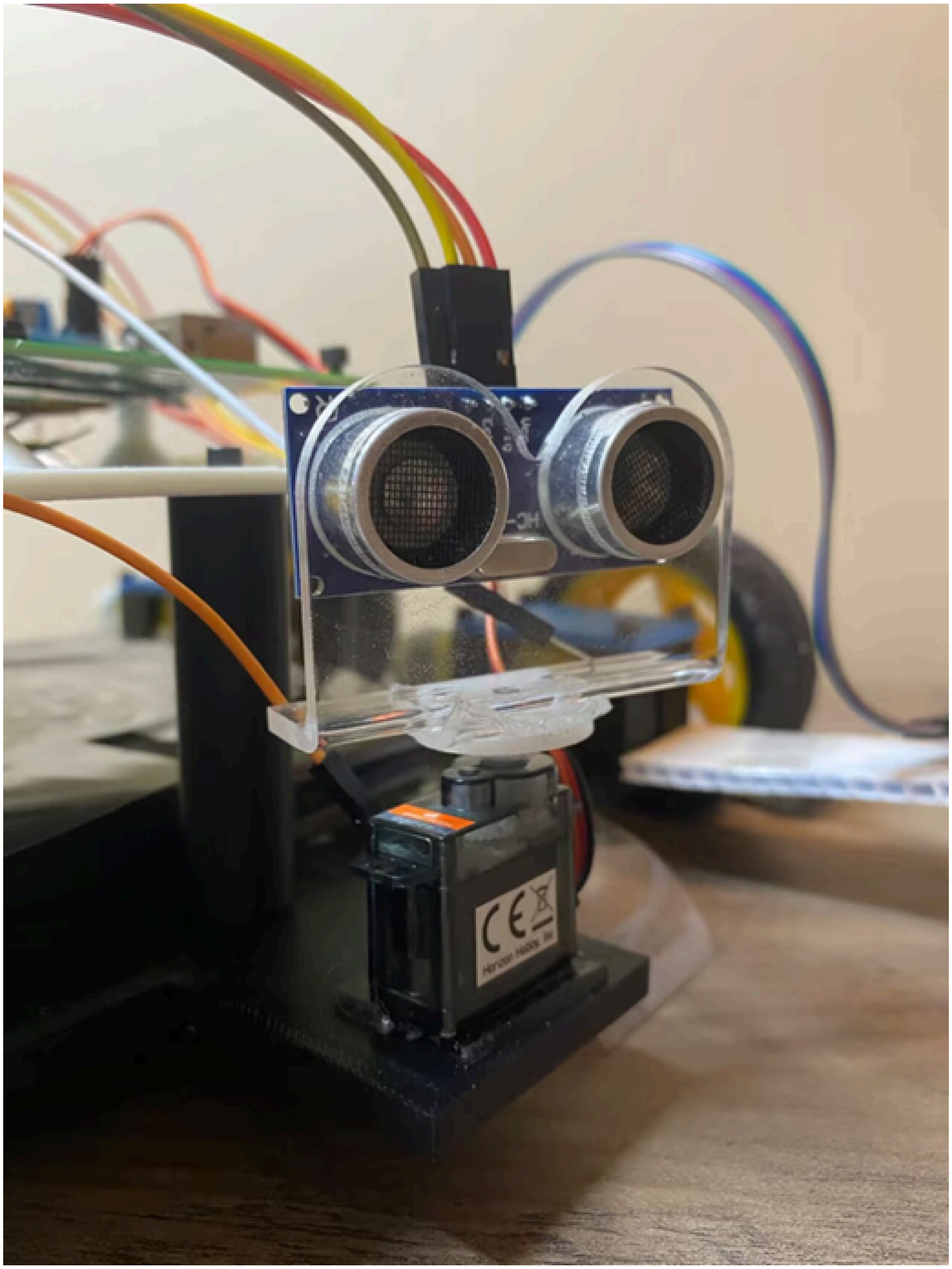

Figure 12. Ultrasonic Sensor