# Design and Implementation of a Line Following and Obstacle Avoidance Robot


MD Rafiul Islam[*], MD Azadul Islam[*], Umut Faruk Karakaya[*]

[*]Systems Engineering and Prototyping, BSc Electronic Engineering

Hochschule Hamm-Lippstadt, Lippstadt, Germany

{rafiul.islam, azadul.islam, umut.karakaya}@stud.hshl.de



***Abstract*—This paper presents the design, development, and testing of an autonomous robot that can follow a black line on a white surface and avoid obstacles detected within its path. The robot is built using Arduino and multiple sensors including infrared (IR), ultrasonic, and color sensors. The project integrates hardware prototyping, systems engineering principles, and timed behavior modeling using UPPAAL. The robot adjusts its behavior based on environmental feedback, such as light variation and battery voltage, enhancing robustness and real-world usability.**

***Index Terms*—Line follower robot, obstacle avoidance, Arduino, sensor fusion, autonomous system, prototyping, UPPAAL, systems engineering.**


## I. Introduction

Autonomous navigation systems are rapidly becoming critical in robotics. This project aims to implement a small-scale, autonomous mobile robot capable of line following and intelligent obstacle avoidance. The development follows a structured engineering approach, combining system modeling, electronic design, simulation, and real-world testing.

## II. System Design and Engineering

The foundation of the robot was based on systems engineering principles. Initially, use case and requirement diagrams were created using SysML to define the functional and non-functional aspects of the robot. These models were refined into block definition and internal block diagrams which provided a modular structure. Each module—sensing, control, and actuation—was individually designed for clear interface and responsibility.

Fig. 3 illustrates the SysML Use Case Diagram developed during the first phase of system modeling. The diagram describes key interactions of external actors (e.g., the user or operator) and the robot system. Key use cases include making the robot initiate, triggering line-following mode, triggering obstacle avoidance, and executing zone-based actions in color detection. This model provides an overall notion about functional robot scope and user-driven scenarios in order to guide further system decomposition.

Fig. 4 is the SysML Requirements Diagram, which captures the functional and non-functional requirements required for the robot to function correctly. They include performance requirements (e.g., response time of obstacle detection), operational reliability under various lighting conditions, hardware modularity in integration, and extensibility in software. The

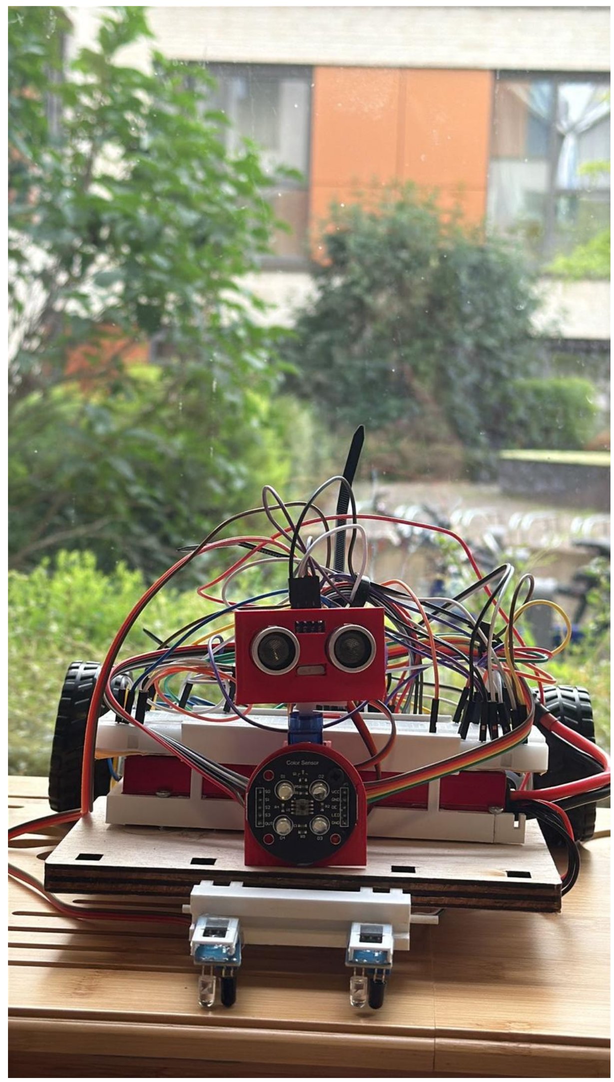

Fig. 1. Front view of the autonomous robot during testing

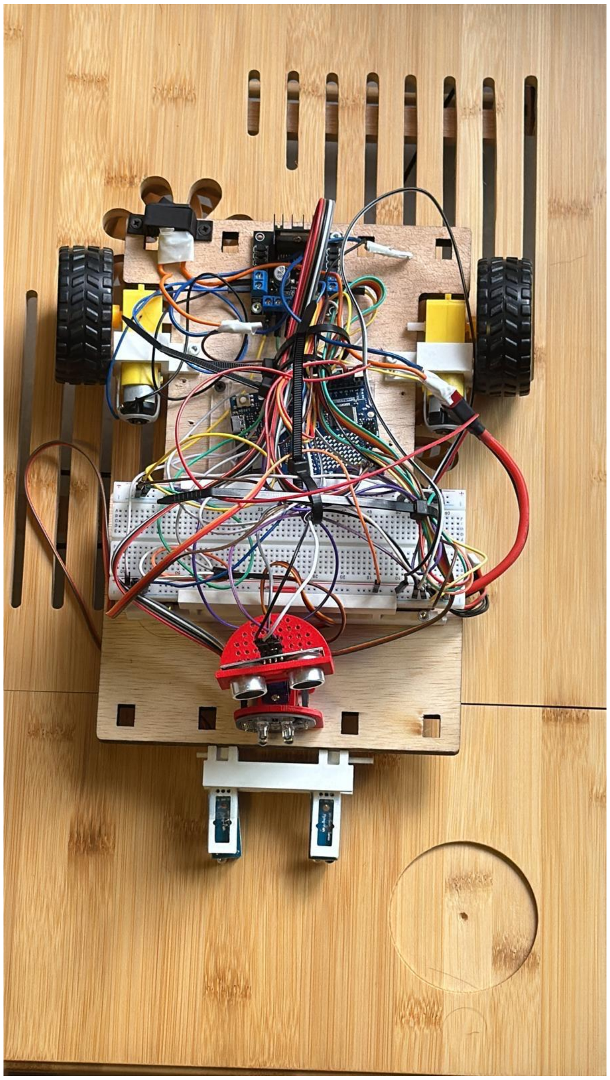

Fig. 2. Side angle showing full robot assembly

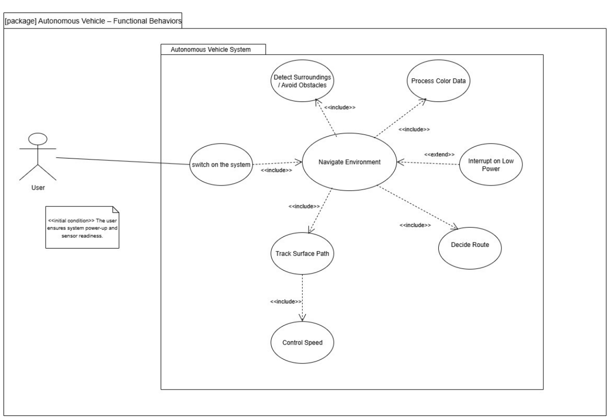


Fig. 3. SysML Use Case Diagram

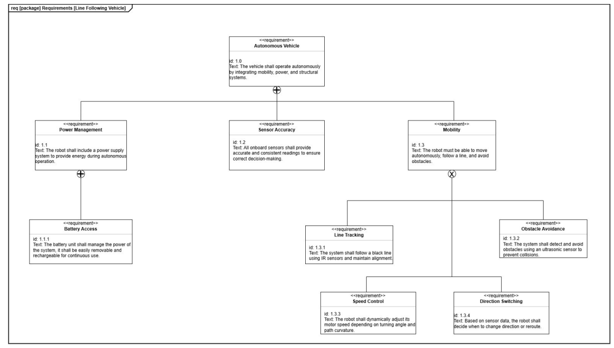


Fig. 4. SysML Requirements Diagram

diagram traces the user needs and system capabilities and serves as a development and validation contract.

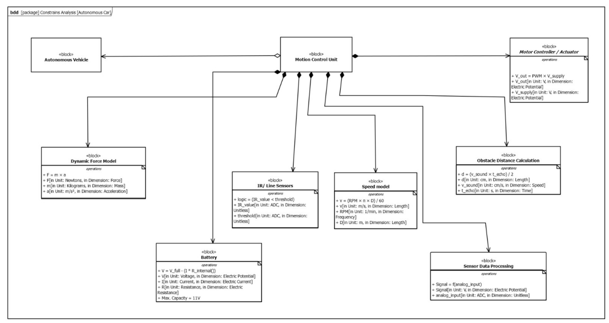


Fig. 5. SysML Constraints Diagram

Fig. 5 shows the SysML Constraints Diagram, which describes the environmental, technical, and resource-based constraints which affect system design choices. Constraints such as limited onboard power supply, accuracy of sensors, hard real-time response times, and processing throughput of the microcontroller are shown to guide hardware and software choices. The physical depiction of constraints enables trade-off choices to be made with expertise, ensuring the final implementation remains within what is feasible.

Fig. 6 is the SysML Activity Diagram, and this is a depiction of the dynamic robot workflow. This diagram decomposes the behavior of the robot into sequential and concurrent activities such as scanning for line deviations constantly, sensing proximity sensors, sensing color zones, and switching control logic. The activity flow is the depiction of how the system moves from one working state to the next based on sensor signals and control logic to enable the refinement of embedded software routines.

Fig. 7 displays the SysML State Machine Diagram representing the state-based robotics behavior. Key states are Idle, Line Following, Obstacle Detected, Obstacle Avoidance, and Color Response. Transitions between these states are triggered by real-time sensor values (e.g., IR sensor for a curve or ultrasonic sensor for an obstacle). The state machine gives deterministic and predictable responses and will be the basis for UPPAAL-based timing verification in later stages.

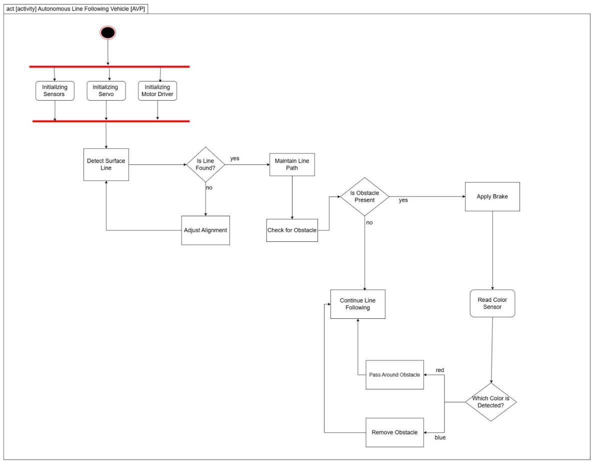


Fig. 6. SysML Activity Diagram

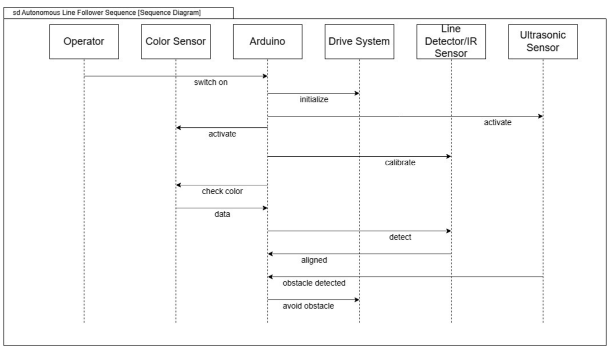


Fig. 7. SysML State Machine Diagram

## III. Behavioral Modeling using UPPAAL

To validate the robot's timing and safety behaviors, formal models were created using UPPAAL. These timed automata diagrams describe the interaction of sensors and actuators over time, including decision delays and critical thresholds for obstacle detection.

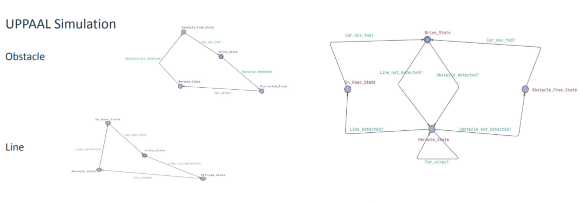


Fig. 8. UPPAAL model of line-following and state transition

## IV. Prototyping and Design

### A. Electronic Components

- Arduino Uno microcontroller
- L298N motor driver
- 2 IR sensors for line detection
- Ultrasonic sensor for distance measurement

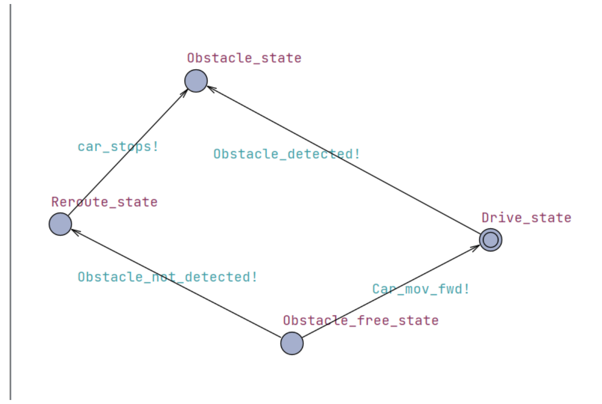


Fig. 9. UPPAAL obstacle avoidance decision automaton

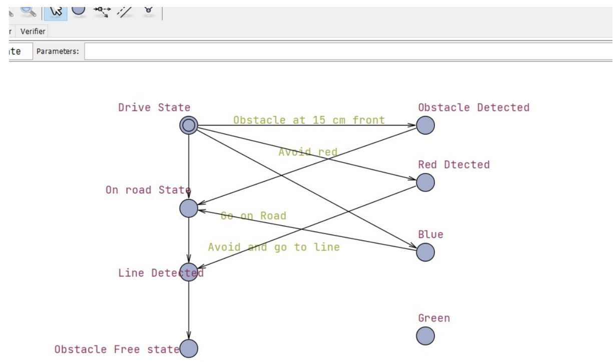


Fig. 10. Updated UPPAAL transition structure

- Servo motor for scanning obstacles
- TCS3200 color sensor for identifying specific zones
- 2 DC motors for driving wheels
- 12V battery (with 5V regulation for logic circuits)

### B. Virtual and Physical Prototypes

Initial logic and wiring were tested using Tinkercad simulation. A physical prototype was developed on a custom chassis. The system was incrementally built and validated using Multisim for circuit design, followed by breadboard implementation and final soldering.

### C. Hardware Assembly

All components were mounted securely on the chassis. Connections were made as per schematic, considering voltage compatibility. A switch and power regulation module were added to safely deliver 12V to motors and 5V to logic.

## V. Hardware Component Overview

**IR Sensor:** The IR module includes an IR LED emitter and receiver. It detects obstacles by measuring reflected infrared light. A potentiometer adjusts detection distance. Pins include VCC (5V), GND, and OUT (digital signal).

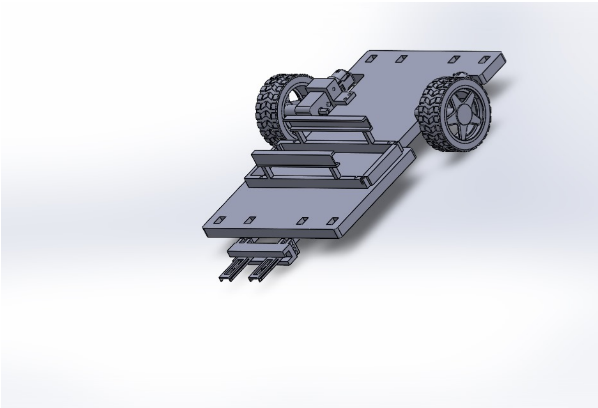

Fig. 11. Physical prototype of the assembled robot

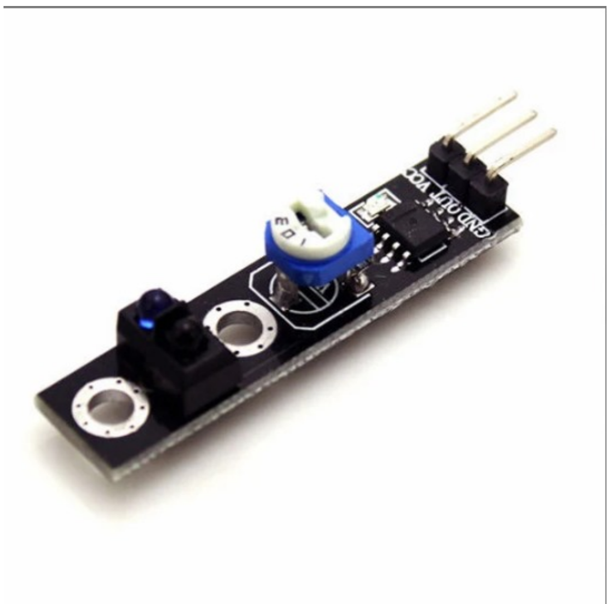


Fig. 12. IR Sensor Module

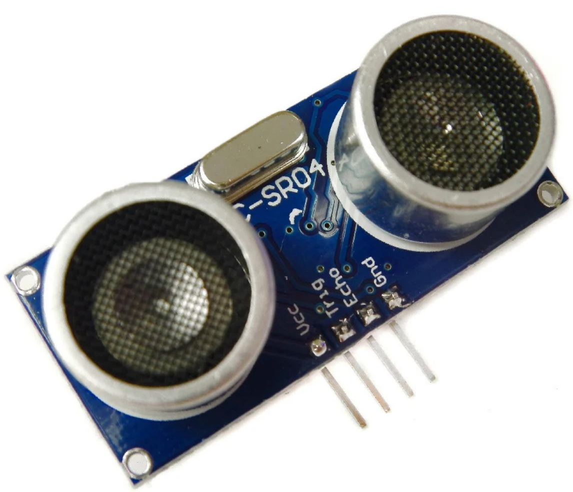


Fig. 13. Ultrasonic Sensor

**Ultrasonic Sensor:** Measures distance by sending a pulse and timing the echo return. Formula used: `distance = (duration * 0.0343) / 2`.


Fig. 14. Servo Motor

**Servo Motor:** Requires 4.8–6V, idle current 6 mA, stall up to 800 mA. Not powered directly from Arduino.


Fig. 15. TCS3200 Color Sensor

**Color Sensor:** Detects color using 8x8 photodiode array. Frequency output based on filter used.

**Arduino Uno:** Based on ATmega328. 14 digital, 6 analog pins. Operating voltage 5V.

**Motor Driver:** L298P dual H-bridge controls direction, braking, PWM.

**DC Motors:** Used to drive the robot. Controlled via motor driver.

**Battery:** LiPo 7.4V 3000mAh. High current output and compact.

Fig. 16. Arduino Uno

Fig. 17. Motor Driver (L298N)

## VI. Circuit Design

## VII. Software Implementation

The Arduino code integrates line following, obstacle detection, and color recognition. It uses a non-blocking loop, servo sweep logic for obstacle scanning, and conditional branching based on sensor input to control motion. Real-time feedback ensures responsiveness and robustness.

## VIII. Git Usage and Collaboration

Project progress was tracked using GitHub. Branches were created per feature. Each member contributed based on assigned tasks, and all commits were documented with messages.
Project Repository: https://github.com/UmutKarakayaHSHL/D3--prototyping

Fig. 18. Yellow DC Motor

Fig. 19. LiPo Battery

## IX. Contributors

- **MD Rafiul Islam:** Responsible for circuit connections and simulations, coding (Obstacle Avoidance and Color Detection), and LaTeX documentation.
- **MD Azadul Islam:** Handled SolidWorks mechanical design, coding (IR Line Following), and UPPAAL simulations.
- **Umut Faruk Karakaya:** Created all diagrams and illustrations including SysML modeling and behavioral representations.

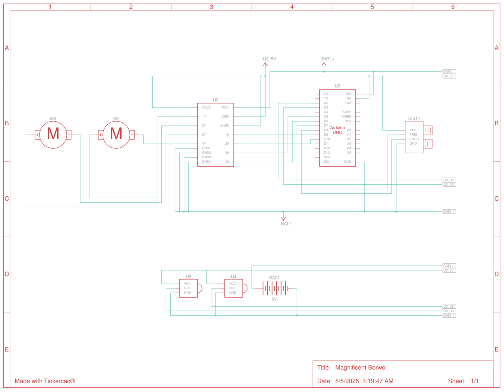


Fig. 20. Complete circuit layout of the robot

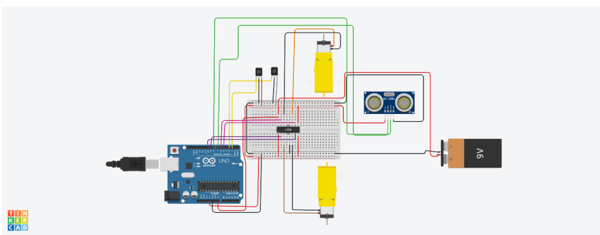

Fig. 21. Tinkercad simulation-based wiring diagram

## X. Key Achievements

- Developed a fully functional autonomous robot
- Validated timing behavior using UPPAAL timed automata
- Created reusable, modular code with real-time sensor feedback
- Demonstrated robust performance under various voltage and light conditions
- Successfully collaborated using GitHub with task-wise contributions

## Acknowledgment

We thank Prof. Dr. Stefan Henkler and the Prototyping staff at HSHL for guidance throughout the project.

## References

## References

[1] Arduino Documentation, *Arduino.cc*. [Online]. Available: https://www.arduino.cc/en/Guide [Accessed: Jul. 5, 2025].

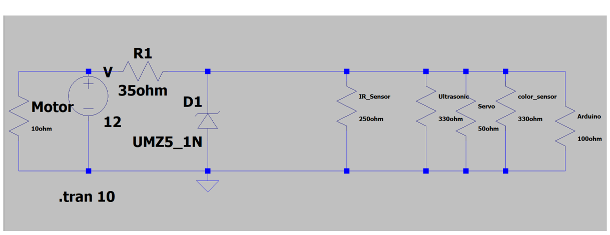


Fig. 22. Voltage regulation and power distribution simulated in LTspice

| Team Member | Number of Commits |
|---|---|
| MD Rafiul Islam | 53 |
| MD Azadul Islam | 53 |
| Umut Faruk Karakaya | 53 |
| **Total** | **159** |

TABLE I
GitHub Commits by Team Members

[2] Tinkercad Simulation, *Autodesk*. [Online]. Available: https://www.tinkercad.com [Accessed: Jul. 5, 2025].

[3] Uppaal Model Checker, *Uppaal.org*. [Online]. Available: http://www.uppaal.org/ [Accessed: Jul. 5, 2025].

[4] LTspice - Analog Circuit Simulator, *Analog Devices*. [Online]. Available: https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html [Accessed: Jul. 5, 2025].

[5] "Obstacle Detector using IR Module," *Arduino Project Hub*. [Online]. Available: https://projecthub.arduino.cc/aboda243/obstacle-detector-using-ir-module-tutorial-101320 [Accessed: Jul. 5, 2025].

[6] "Servo Motors," *Arduino Documentation*. [Online]. Available: https://docs.arduino.cc/learn/electronics/servo-motors [Accessed: Jul. 5, 2025].

[7] "Color Detection Using TCS3200," *Arduino Project Hub*. [Online]. Available: https://projecthub.arduino.cc/SurtrTech/color-detection-using-tcs3200230-a1e463 [Accessed: Jul. 5, 2025].

[8] "What is Arduino Uno? A Getting Started Guide," *RS-Online DesignSpark*. [Online]. Available: https://www.rs-online.com/designspark/what-is-arduino-uno-a-getting-started-guide [Accessed: Jul. 5, 2025].

[9] "Motor Shield Rev3: Controlling DC Motor," *Arduino Documentation*. [Online]. Available: https://docs.arduino.cc/tutorials/motor-shield-rev3/msr3-controlling-dc-motor/ [Accessed: Jul. 5, 2025].

[10] "Arduino Two-Wheel Self Balancing Robot," *Arduino Project Hub*. [Online]. Available: https://projecthub.arduino.cc/mircemk/arduino-two-weel-self-balancing-robot-edef61 [Accessed: Jul. 5, 2025].

[11] "Conrad Energy LiPo Battery 7.4V 3000mAh," *Conrad Electronic*. [Online]. Available: https://www.conrad.nl/nl/p/conrad-energy-lipo-accupack-7-4-v-3000-mah-aantal-cellen-2-20-c-stick-hardcase-t-b html [Accessed: Jul. 5, 2025].

## Declaration of Originality

We, the undersigned, hereby declare that this report titled Design and Implementation of a Line Following and Obstacle Avoidance Robot is entirely our own work. All sources and external contributions have been properly cited and acknowledged. No part of this document has been plagiarized.

- MD Rafiul Islam
- MD Azadul Islam
- Umut Faruk Karakaya