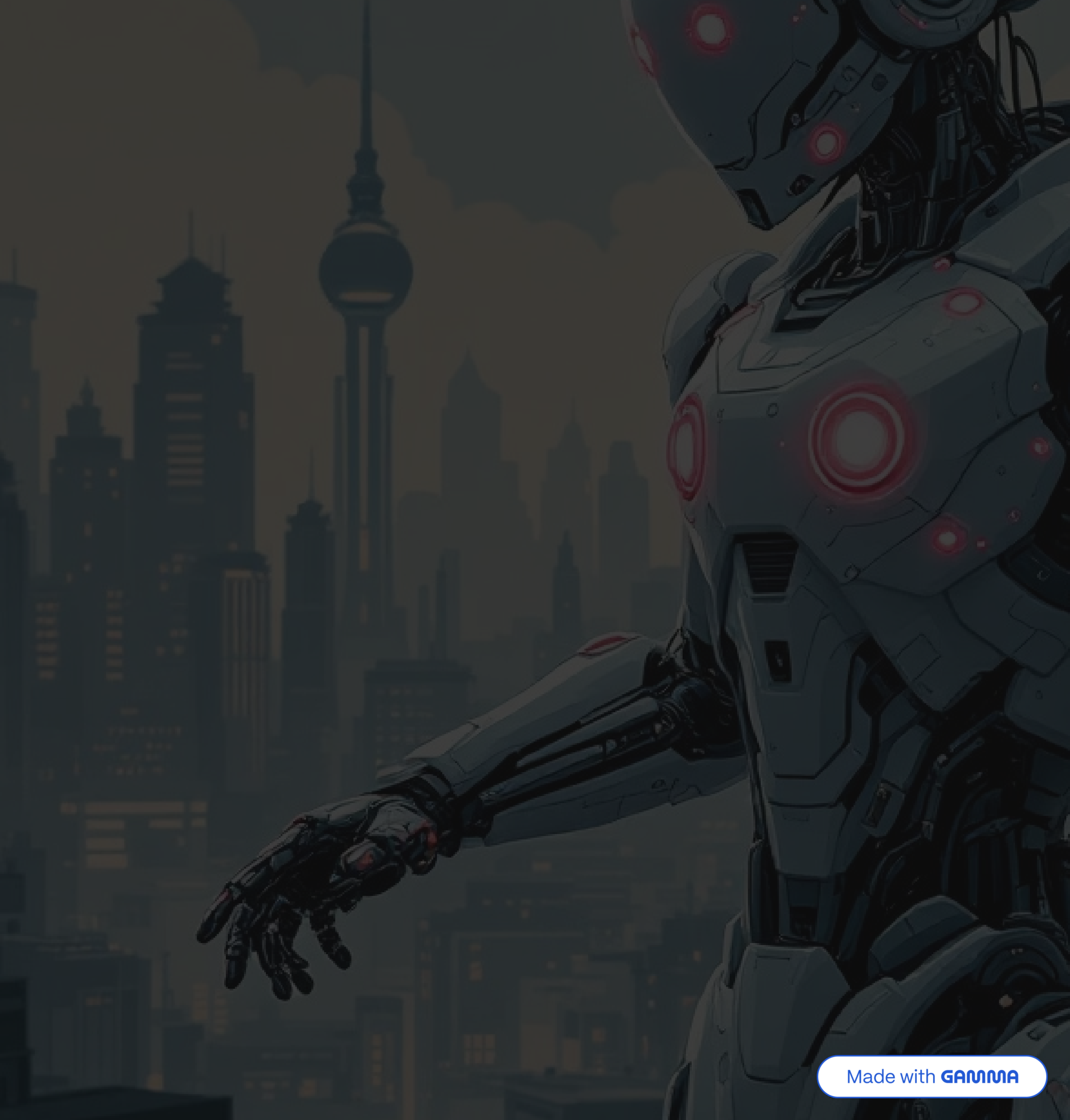
Obstacle-Avoidance & Telematics Robot: Comprehensive Tinkercad Simulation Report

This report details the design, functionality, and operation of an advanced Arduino UNO-based mobile robot, thoroughly simulated within Tinkercad Circuits. It covers robust collision avoidance, real-time environmental monitoring (telematics), and dynamic user feedback, providing a complete overview for demonstration or educational purposes.



1. Introduction

This report serves as a detailed documentation for an advanced Arduino UNO-based mobile robot, meticulously developed and rigorously tested within the Tinkercad Circuits simulation environment. The robot exemplifies sophisticated capabilities including robust collision avoidance, real-time environmental monitoring (telematics), and dynamic feedback mechanisms, making it an ideal platform for both educational demonstrations and practical prototyping. The entire system's functionality is fully simulated, ensuring that all features and operational aspects can be explored and understood without the need for physical hardware.

Motivation

Traditional robots often focus on either obstacle avoidance or basic remote control. However, with the increasing need for **environmental monitoring** (telematics) in smart cities, industries, health safety, and education, combining telematics with reliable, autonomous navigation is both innovative and valuable. The result is a multi-functional, sensor-rich platform that mirrors real-world robotic needs in a low-cost, accessible manner.

Purpose and Significance

The core objectives of this project are:

- **To realize robust, live obstacle avoidance** using three spatially distributed ultrasonic sensors, ensuring the robot detects and navigates around obstacles ahead, on the left, and on the right, rather than blindly following a single forward sensor.
- **To actively monitor the environment**, reporting ambient temperature (using a TMP36 analog sensor) and humidity (simulated via a potentiometer, due to limited sensor availability in Tinkercad), with live data displayed for review.
- **To strengthen operational safety** through a tilt-switch, which instantly halts and alerts the robot in the event of a tip or flip—preventing damage or further collision.
- **To provide real-time, multi-modal feedback** using an RGB LED (for status color coding) and a buzzer (for audio alerts), so that users have immediate sensory cues about the robot's state.
- **To control two independently driven DC motors** via an L298N H-bridge, allowing precise, programmable motion responses: forward, backward, left, right, and stop.
- **To leverage a fully virtual Tinkercad simulation**, eliminating hardware barriers and making the design accessible and replicable for educators, students, and hobbyists alike.

Unique Project Features

- **True multi-directional awareness**: The robot senses obstacles on three sides and reacts dynamically, rather than being limited to simple "front-only" logic.
- **Continuous environmental telemetry**: Temperature and humidity are sampled an time, making the robot not just an agent but a rolling monitoring station.

2. Project Objectives

The development of this robot in Tinkercad Circuits was guided by a set of clear objectives, ensuring a comprehensive and functional simulation. Each objective contributes to the robot's overall intelligence and utility:

1 Autonomous Obstacle Avoidance

To implement a proactive obstacle avoidance system utilizing three ultrasonic sensors for detection in front, left, and right directions, enabling the robot to navigate dynamic environments without collisions.

2 Real-time Telematics

To integrate sensors for continuous monitoring of environmental parameters, specifically temperature and humidity, providing live data relevant to the robot's surroundings.

3 Safety Mechanism

To incorporate a tilt switch as a safety feature, ensuring the robot immediately stops all motion and issues an alert if it is tipped over or flipped, preventing potential damage or unexpected movement.

4 User Feedback

To provide intuitive and immediate feedback to the user through visual cues (RGB LED) and auditory signals (buzzer) for various operational states, alerts, and warnings.

5 Serial Telemetry

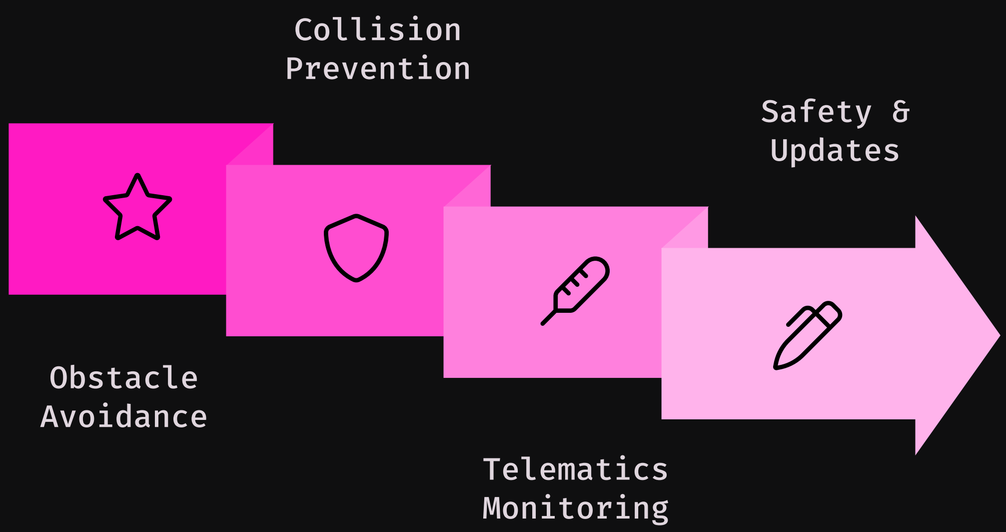
To establish a continuous data stream to the Serial Monitor, displaying all real-time sensor readings and the robot's current operational status, crucial for debugging and monitoring.

6 Tinkercad Demonstrability

To ensure that the entire project is fully demonstrable and verifiable within the Tinkercad simulation environment, making it accessible for users without physical hardware setups.

3. System Overview

The robot's design strategically integrates two primary subsystems, allowing it to perform both autonomous navigation and environmental monitoring effectively. These subsystems work in concert to provide a comprehensive robotic solution:



A. Obstacle Avoidance Subsystem

This subsystem is the robot's primary mechanism for safe navigation. It leverages three HC-SR04 ultrasonic sensors strategically positioned to scan the environment. One sensor is mounted at the front for direct forward detection, while the other two are placed on the left and right sides to detect lateral obstructions. When an object is detected within a predefined close proximity (e.g., less than 20 cm in front or 15 cm on the sides), the robot's control logic is activated. This triggers a specific avoidance maneuver, which may include executing a precise turn (either left or right) to circumvent the obstacle, or briefly reversing its direction if both forward and lateral paths are blocked. This ensures the robot can autonomously navigate complex spaces.

B. Telematics Monitoring Subsystem

The telematics subsystem provides the robot with environmental awareness. It incorporates a TMP36 temperature sensor for accurate ambient temperature readings and a potentiometer, ingeniously repurposed to act as a proxy for humidity sensing, providing continuous environmental updates. Beyond just data collection, a crucial safety feature is integrated: a tilt switch. This switch constantly monitors the robot's orientation, and if it detects that the robot is not upright (i.e., tipped or flipped), it immediately triggers a safety protocol. This protocol halts all motor activity and initiates an alarm, ensuring the robot's stability and preventing potential damage or unintended movement.

4. Component List & Role

The following components are integral to the robot's functionality, each playing a critical role in its operation and interacting with the Arduino UNO as specified:

Arduino UNO	Core Microcontroller	—
HC-SR04 ×3	Obstacle distance sensing (front, left, right)	7/6, 5/4, 3/2
TMP36 Sensor	Temperature sensing	A0
Potentiometer	Humidity proxy	A1
Tilt Switch	Safety (robot tipped/fallen)	D8
RGB LED	Visual status indication (with 220Ω resistors on each color)	A2 (R), A3 (G), A4 (B)
Piezo Buzzer	Audio feedback	D9
L298N Driver	Motor control interface	D10, D11, D12, D13
DC Motors ×2	Locomotion — (via L298N)	—
Breadboard/wires	Connections	—

5. Circuit Schematic & Wiring

A precise and well-organized circuit schematic is crucial for the proper functioning of the robot. Below is an overview of the main connections, ensuring all components are correctly wired to the Arduino UNO. For a detailed visual representation, please refer to the annotated schematic images in your appendix, where each wire and connection point is clearly labeled.

Main Connections Overview:

- **HC-SR04s (Front/Left/Right):** Each ultrasonic sensor requires two digital pins. The Front sensor uses TRIG on D7 and ECHO on D6. The Left sensor uses TRIG on D5 and ECHO on D4. The Right sensor uses TRIG on D3 and ECHO on D2. Ensure these are connected to their respective VCC (5V) and GND rails.
- **TMP36 Temperature Sensor:** This analog sensor has three pins. Connect its VCC pin to the Arduino's 5V, its OUT pin to Arduino Analog Pin A0, and its GND pin to the Arduino's GND.
- **Potentiometer (Humidity Proxy):** The potentiometer also has three pins. Connect one end pin to the Arduino's 5V, the other end pin to GND, and the wiper (middle) pin to Arduino Analog Pin A1. This setup allows for varying voltage input based on the knob's position.
- **Tilt Switch:** This digital sensor is connected between Arduino Digital Pin D8 and GND. In the code, this pin is configured with `INPUT_PULLUP`, which simplifies the wiring by using the Arduino's internal pull-up resistor.
- **RGB LED:** The common cathode RGB LED requires four connections. The common cathode pin should go to GND. Each color channel (Red, Green, Blue) must be connected through a 220Ω current-limiting resistor to Arduino Analog Pins A2 (Red), A3 (Green), and A4 (Blue) respectively. These analog pins are used for `analogWrite` to control brightness and color mixing.
- **Piezo Buzzer:** The buzzer has two terminals. Connect one terminal to Arduino Digital Pin D9 and the other to GND. This allows the Arduino to generate audio tones.
- **L298N Motor Driver:** This module is crucial for controlling the DC motors. Its VCC and ENA/ENB pins should be connected to the 5V rail (or an external 12V supply for motors if applicable, though 5V is sufficient for Tinkercad simulation). The L298N's GND pin must be connected to the Arduino's GND. The input pins (IN1, IN2, IN3, IN4) are connected to Arduino Digital Pins D10, D11, D12, and D13, which control the direction and speed of the motors. Finally, the output terminals (OUT1/2 and OUT3/4) are connected directly to the respective DC motors.

Ensure all connections are secure on the breadboard and that wire lengths are optimized to prevent clutter. Proper wiring is fundamental for the stable operation of all sensors, actuators, and control systems.

6. Firmware & Code Explanation

The robot's intelligence is embodied in its Arduino firmware, which orchestrates all sensor readings, decision-making, and actuator control. The code is designed for clarity, efficiency, and direct hardware interaction, avoiding external libraries for core functionalities where possible. This section highlights the key features and operational principles embedded within the code.

Key Features of the Code:

- **Direct Ultrasonic Sensor Interaction:** The code directly triggers and reads the HC-SR04 ultrasonic sensors using basic digital writes and reads (`pulseIn`), eliminating the need for bulky external libraries. This approach provides fine-grained control over sensor operation and contributes to a smaller memory footprint.
- **Analog Sensor Readings:** The TMP36 temperature sensor and the potentiometer (for humidity proxy) are read directly using `analogRead()`. The raw analog values are then converted into meaningful units (Celsius for temperature, percentage for humidity) using appropriate scaling formulas.
- **Comprehensive Decision Tree:** The `loop()` function contains a robust decision tree that evaluates the robot's current state based on sensor inputs. It prioritizes safety (tilt detection) over obstacle avoidance, and then handles clear path navigation. This ensures a logical and responsive behavioral pattern.
- **Immediate Feedback Mechanisms:** The RGB LED and piezo buzzer are dynamically controlled by the code to provide instant visual and audio feedback. Different colors and tones correspond to the robot's operational status (e.g., green for clear, yellow for obstacle, red for tipped).
- **Continuous Serial Telemetry:** A crucial debugging and monitoring feature, the code continuously prints all relevant sensor readings (front, left, right distances, temperature, humidity, tilt status) to the Serial Monitor. This provides a live, comprehensive overview of the robot's environment and internal state.

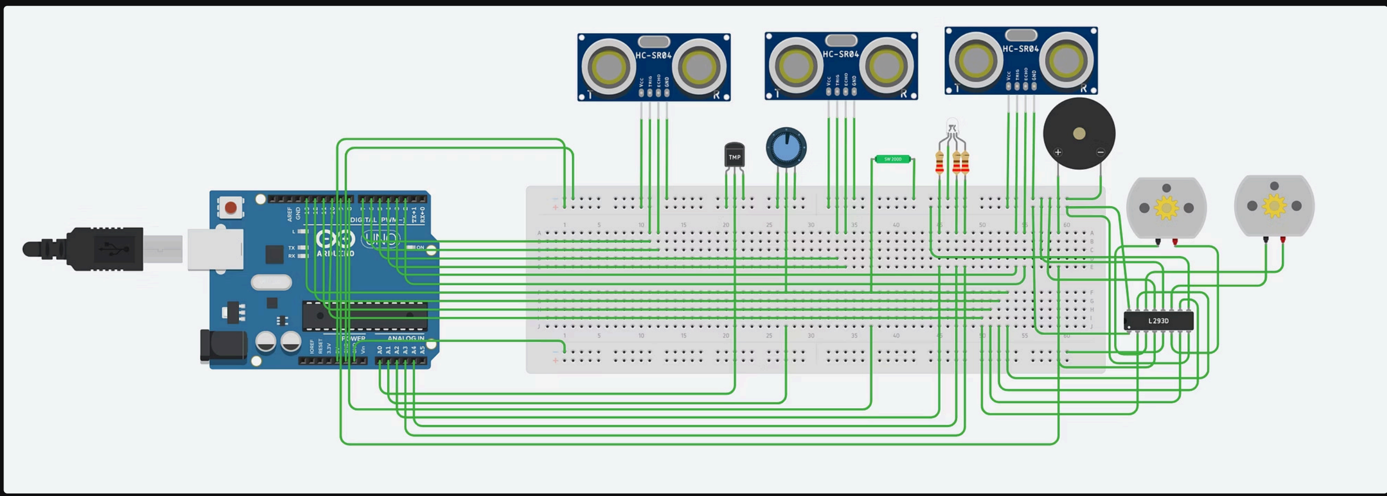
The code structure is modular, with dedicated functions for motor control (`moveForward()`, `turnLeft()`, etc.), LED control (`setLED()`, `dangerAlert()`), and ultrasonic distance measurement (`readDistance()`). This modularity enhances readability and maintainability.

For a complete and detailed review, please refer to the full source code provided in your appendix. The code is thoroughly commented to explain each section, variable, and function, facilitating understanding and potential modifications. Ensure that the final working code in the appendix is formatted clearly for easy readability.

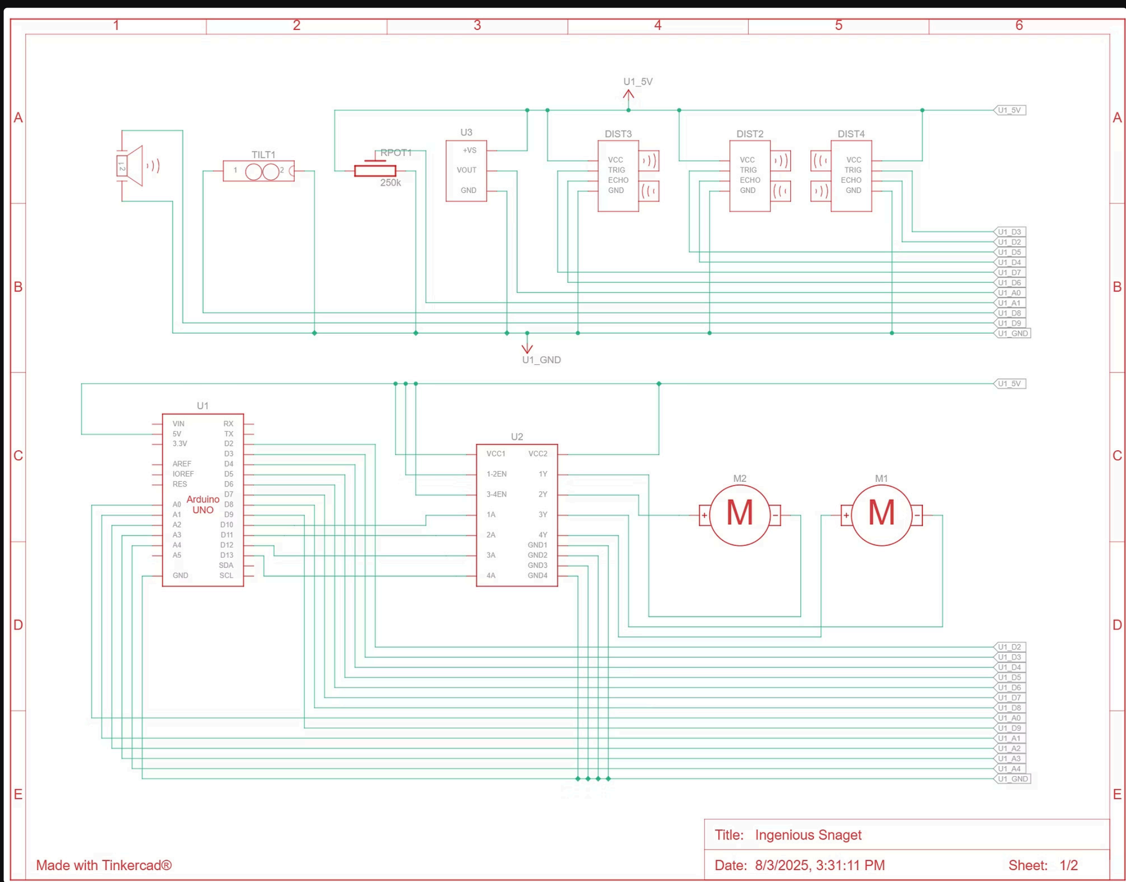
Full Code

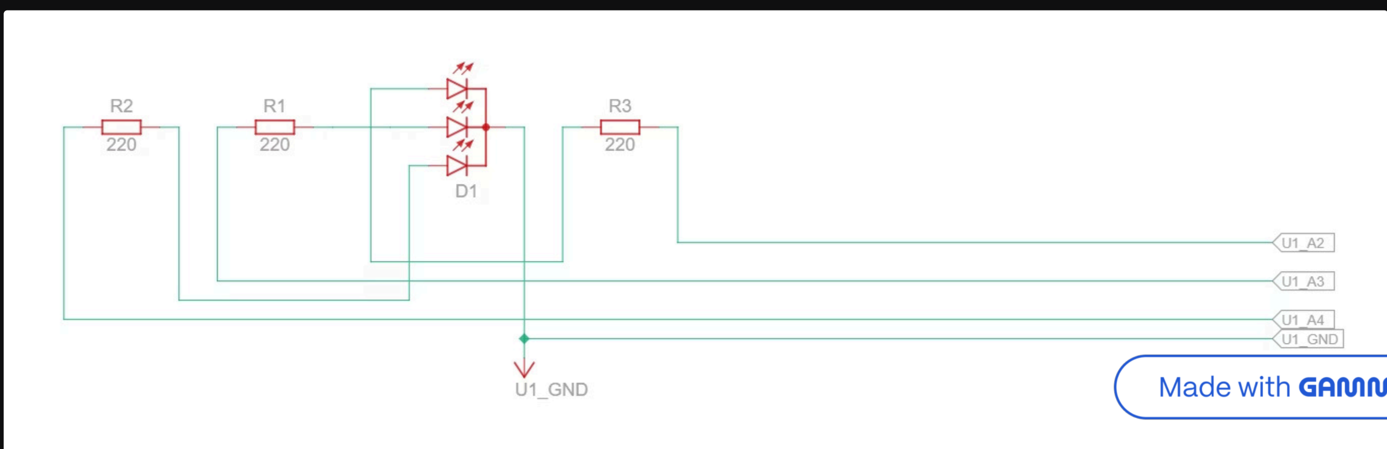
```
// Advanced Obstacle-Avoidance & Telematics Robot
// Fully compatible with Tinkercad Circuits

// Pin Definitions
#define TRIG_F 7    // Front ultrasonic trigger
#define ECHO_F 6    // Front ultrasonic echo
#define TRIG_L 5    // Left ultrasonic trigger
#define ECHO_L 4    // Left ultrasonic echo
#define TRIG_R 3    // Right ultrasonic trigger
#define ECHO_R 2    // Right ultrasonic echo

#define TMP_PIN  A0 // TMP36 temperature sensor
#define HUM_PIN  A1 // Potentiometer for humidity proxy
#define TILT_PIN 8  // Tilt switch (LOW when tipped)

#define BUZ_PIN   9  // Buzzer
#define RED_PIN  A2  // RGB LED Red channel
#define GRN_PIN  A3  // RGB LED Green channel
#define BLU_PIN  A4  // RGB LED Blue channel

#define IN1 10      // L298N motor driver inputs
#define IN2 11
#define IN3 12
#define IN4 13

// Read HC-SR04 distance in cm
long readDistance(int trigPin, int echoPin) {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  return duration * 0.034 / 2;
}

// Motor control
void moveForward() {
  digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
}
void moveBackward() {
  digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);
```

Images



Schematics





Links To Project

https://www.tinkercad.com/things/6Q2PvHlt6mS-ingenious-snaget/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard&sharecode=YX-WKS31Xhx-Afi8R9yPTr9-bQifoNNWt6vcdBuac2A

Full simulation video

https://drive.google.com/file/d/1lQdDG9eChYtEJKY4YWsmHVDn9GsoiC49/view?usp=drive_link