PROCEDURAL PROGRAMMING WITH C

**Objective** –

The goal of this project is to design and implement an autonomous car system that uses ultrasonic sensors for obstacle detection and avoidance. The car is controlled using motors and a steering servo, and it can be operated manually through a serial interface or autonomously navigate around obstacles.

**System Overview**

The autonomous car consists of:

* **Ultrasonic Sensor**: Measures the distance to obstacles in front of the car.
* **DC Motors**: Control the movement of the car (forward, backward, stop).
* **Servo Motor**: Controls the steering direction of the car.
* **Microcontroller (Arduino)**: Processes sensor data, controls motors, and manages user inputs.

**Inputs and Outputs**

* **Inputs**:
  + Distance measurement from the ultrasonic sensor.
  + Serial commands from the user (e.g., "start", "exit").
* **Outputs**:
  + Motor actions (moving forward, stopping, turning).
  + Steering commands (left, right).

**System Behavior**

1. **Idle State**: The car is stopped until a "start" command is received.
2. **Obstacle Detection**:
   * If the distance is greater than 20 cm, the car moves forward.
   * If the distance is less than or equal to 20 cm, the car stops and assesses the distance:
     + If the distance is between 10 cm and 20 cm, it turns right.
     + If the distance is less than or equal to 10 cm, it turns left.
3. **Manual Control**: The user can start or stop the car using serial commands.

**Flow Diagram**

[ Start ] --> [ Wait for User Input ]

|

[ User Command? ]

/ \

Yes No

| |

[ Start ] [ Read Distance ]

| |

[ Move Forward ] ------> [ Check Distance ]

| |

[ Obstacle? ] -------> [ Turn/Stop ]

/ \

Yes No

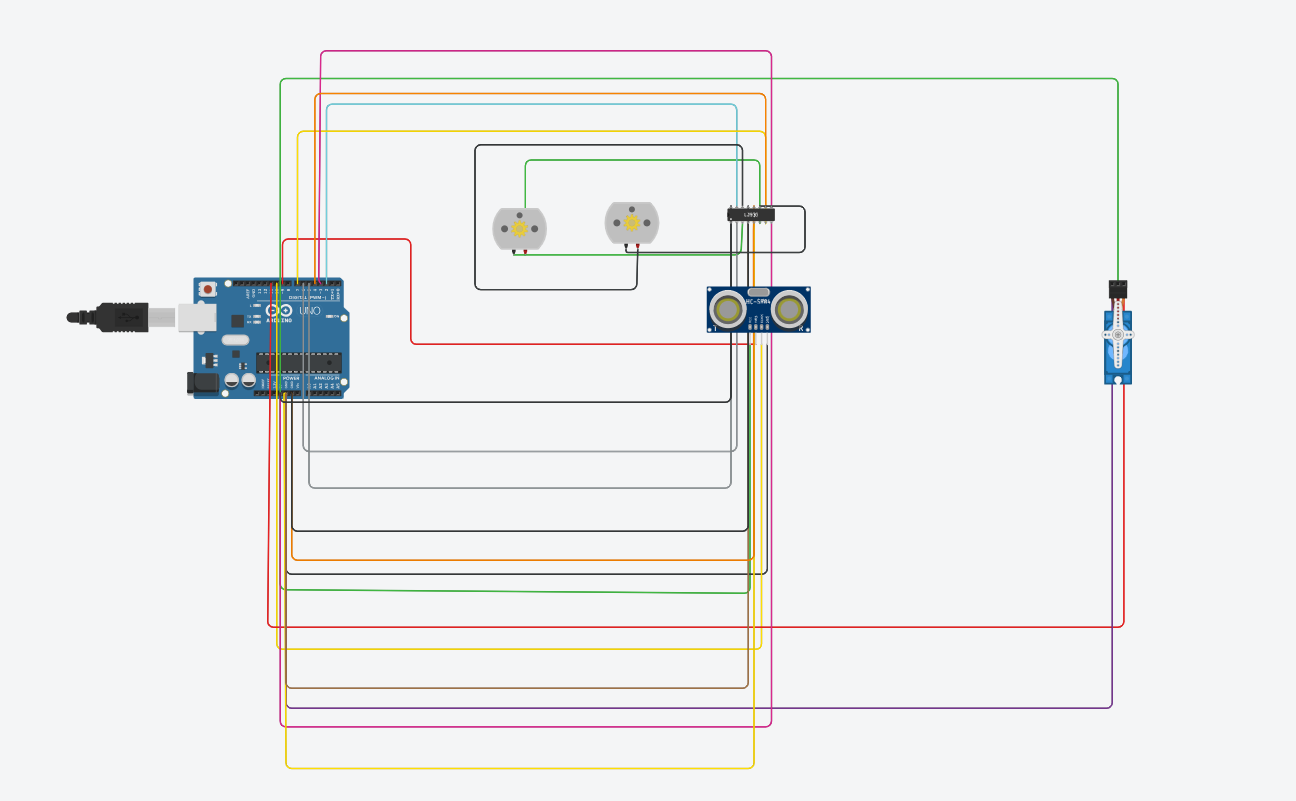
| |

[ Stop ] [ Move Forward ]

**Circuit Design**

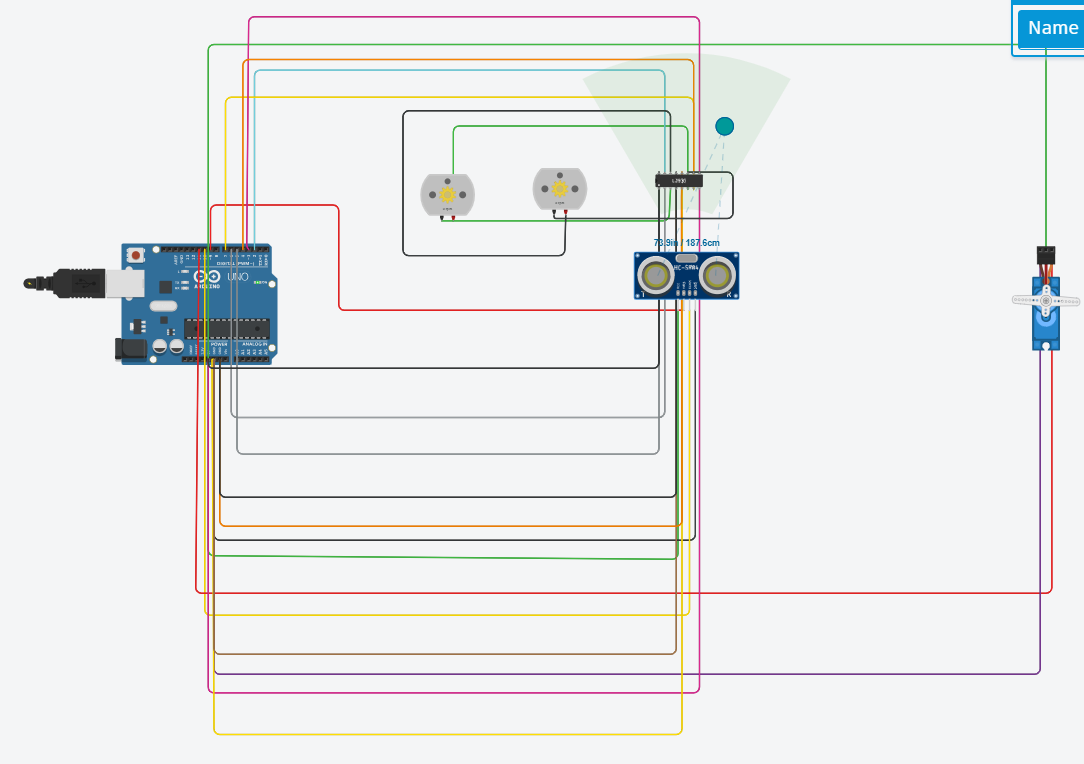
| **Sensor Distance (cm)** |  |  |  |  | **User Command** |  |  |  |  |  |  |  | **Motor State** |  |  |  |  |  |  | **SteeringAction** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| > 20 |  |  |  |  | start |  |  |  |  |  |  |  | Moving Forward |  |  |  |  |  |  | Straight |
| <= 20 and > 10 |  |  |  |  | start |  |  |  |  |  |  |  | Stopped |  |  |  |  |  |  | Turn Right |
| <= 10 |  |  |  |  | start |  |  |  |  |  |  |  | Stopped |  |  |  |  |  |  | Turn Left |
| - |  |  |  |  | exit |  |  |  |  |  |  |  | Stopped |  |  |  |  |  |  | - |

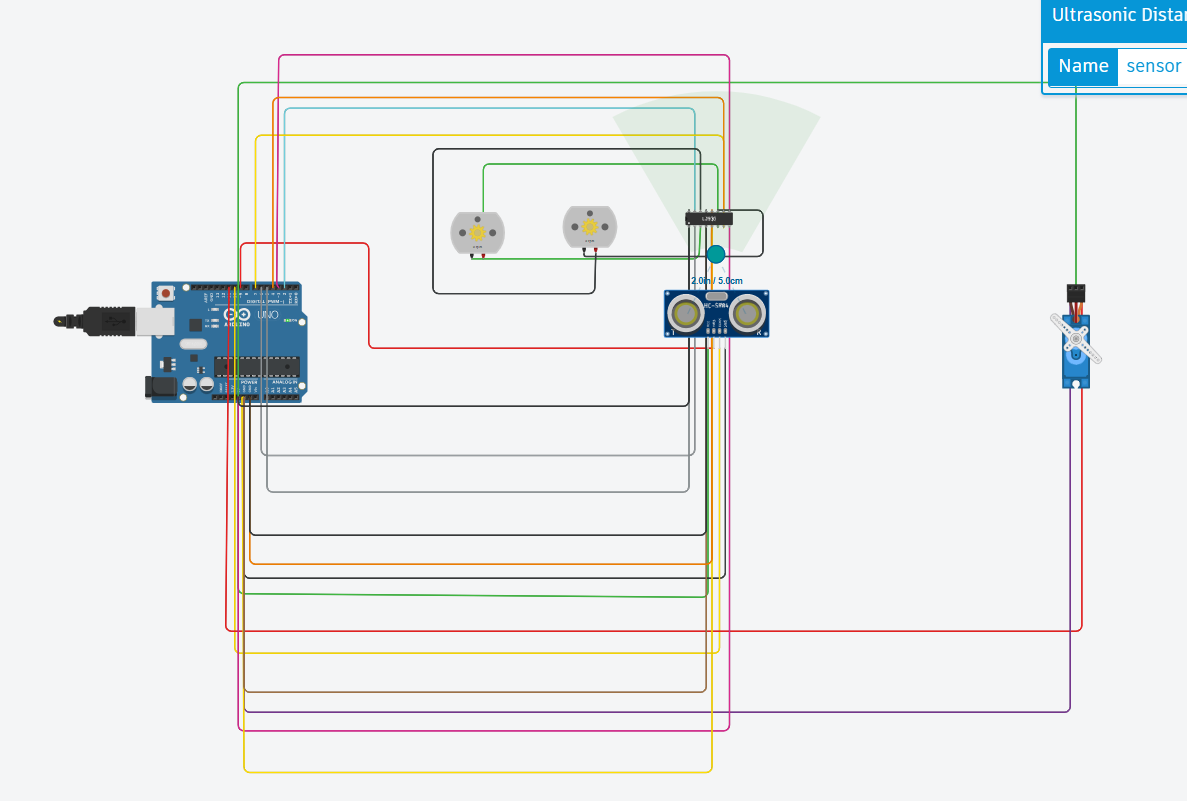
**Circuit Diagram**

****

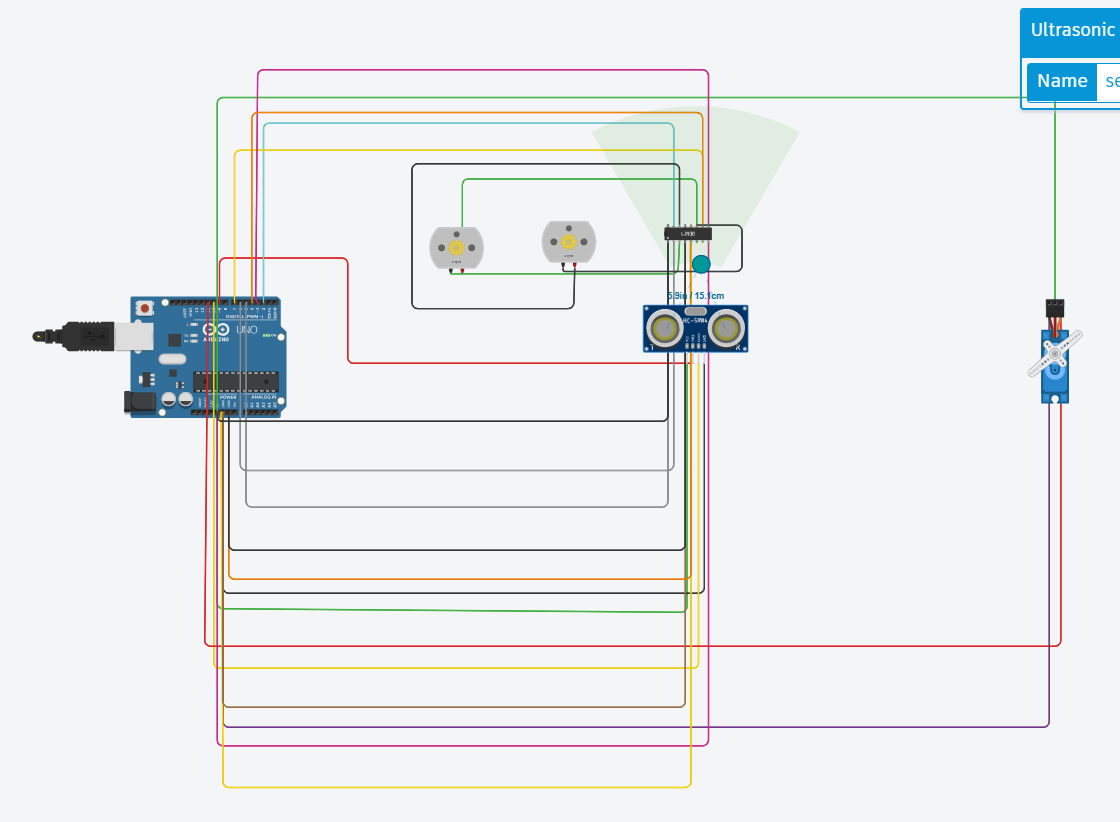
**Simulation Results**

**1.moving forward ( no obstacle detected ) [distance >20.00cm]:**

****

**2. turning left (obstacle detected) [distance<10.00cm]**

**3.turning right (obstacle detected ) [distance between 10 and 20 cm]**

****

**Challenges**

1. **Sensor Accuracy**: Initial readings from the ultrasonic sensor were inconsistent. This was resolved by averaging multiple readings for a single measurement.
2. **Motor Control**: Achieving smooth transitions between moving and stopping required tuning of delay times and motor speed settings.
3. **User Input Handling**: Implementing a robust method to interpret serial commands while maintaining the car's autonomous behavior was challenging. This was addressed by prioritizing sensor checks over user commands in the loop.

**Conclusion**

The autonomous car system effectively detects obstacles and responds appropriately by stopping, turning, or moving forward based on distance readings. Future improvements could include adding more sensors for enhanced navigation, implementing machine learning algorithms for smarter decision-making, or integrating GPS for outdoor navigation. The system can be applied in various fields, such as robotics education, research, and automated transport solutions.