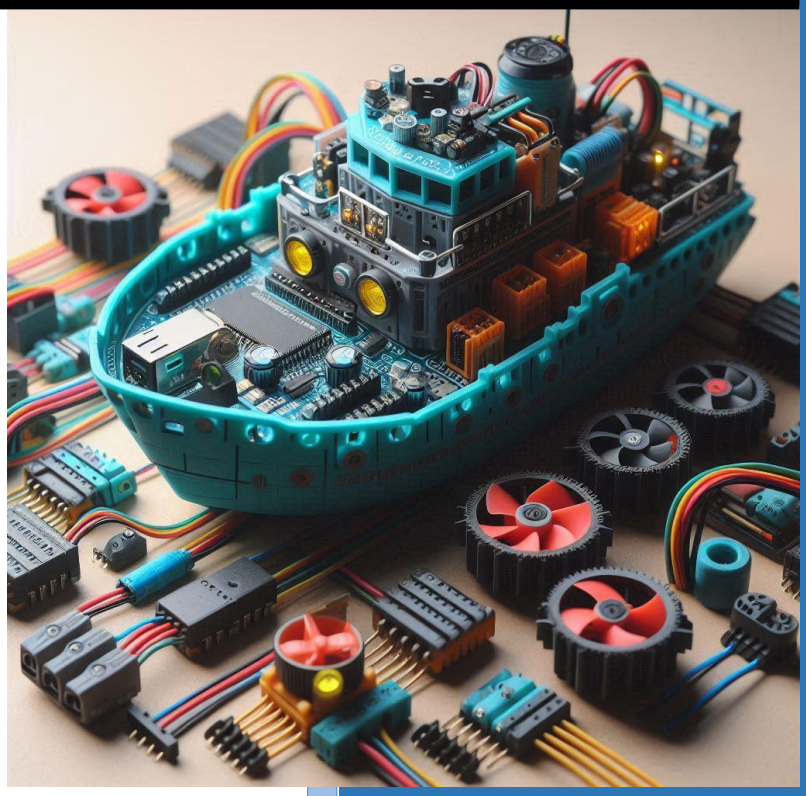
2024

Obstacle-Avoiding ship



Dr.Ahmed Shalaby

5/18/2024

ALAMEIN INTERNATIONAL UNIVERSITY

Arduino-Powered Autonomous Obstacle-Avoiding Ship

Name	Id
Salah ezzat salah Hassan	21100847
Mohamed Saad	21100871
Noor Mokemar	21100861
Zyad Saad	21100868

PROJECT DESCRIPTION

This project involves the design and construction of an autonomous ship powered by an Arduino Uno microcontroller, capable of navigating and avoiding obstacles. The ship utilizes an ultrasonic sensor to detect obstacles in its path and adjusts its movement accordingly to prevent collisions. The primary components of the project include an Arduino Uno, an ultrasonic sensor (HC-SR04), DC motors with propellers for propulsion, a motor driver (L298N), and a power supply. The chassis of the ship is constructed from a lightweight plastic container, housing all the electronic components securely.

Components:

- **Arduino Uno:**
  - The microcontroller that serves as the brain of the ship, processing inputs from the sensor and controlling the motors.
- **Ultrasonic Sensor (HC-SR04):**
  - A sensor that measures the distance to obstacles by emitting ultrasonic waves and measuring their return time.
- **DC Motors with Fan Wheels**:
  - Provide propulsion to the ship, allowing it to move forward, backward, and turn Motor.
- **Driver (L298N):**
  - Interfaces between the Arduino and the DC motors, enabling the control of motor speed and direction.
- **Power Supply:**
  - Batteries that power the Arduino, motor driver, and motors.
- **servo motor:**
  - rotates the ultrasonic sensor to different angles.
- **Chassis:**
  - A lightweight plastic container that serves as the hull of the ship, housing all components.

Functional Requirements:

- **system Obstacle Detection:**
  - The ship must detect obstacles within a range of 0 to 30 cm using the ultrasonic sensor.
  - The ultrasonic sensor should be able to rotate left and right to cover a wider detection area.
- **Autonomous Navigation:**
  - The ship must navigate autonomously, adjusting its path based on obstacle detection.
  - When an obstacle is detected within 20 cm, the ship should stop, reverse, and turn to avoid the obstacle.
- **Motor Control:**
  - The ship must have two DC motors with fan wheels for propulsion.
  - The motors should be controlled to move the ship forward, backward, left, and right.
- **Servo Motor Operation:**
  - The servo motor must rotate the ultrasonic sensor to 0 degrees, 90 degrees, and 180 degrees to enhance obstacle detection.
  - The servo motor should return to the 90-degree position after scanning.
- **User Interface:**
  - The ship must output distance readings to the serial monitor for debugging purposes.

NON-FUNCTIONAL REQUIREMENTS:

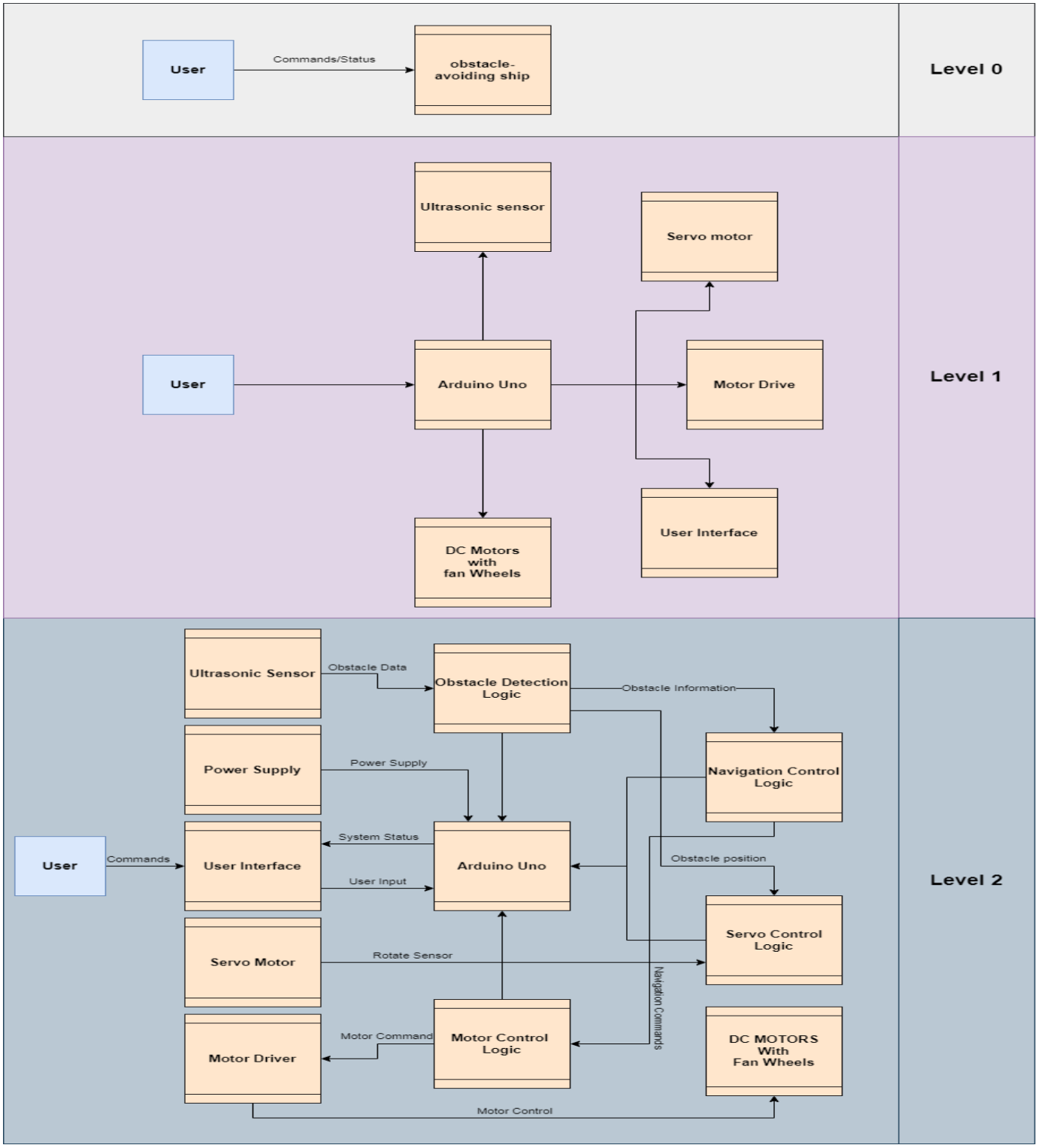
- **Performance:**
  - The ship should respond to obstacle detection and navigation commands with minimal delay.
  - The ultrasonic sensor should provide accurate distance measurements to ensure reliable navigation.
- **Reliability:**
  - The ship should operate continuously for at least 1 hour without failure.
  - All components should be securely mounted to prevent disconnections during operation.
- **Usability:**
  - The ship should be easy to set up and operate by users with basic knowledge of Arduino.
  - The code should be well-documented to facilitate understanding and modification.
- **Safety:**
  - The ship should operate safely without causing harm to users or damage to the environment.
  - All electrical connections should be insulated to prevent short circuits.
- **Portability:**
  - The ship should be lightweight and portable, allowing for easy transportation and deployment in different environments.
- **Power Efficiency:**
  - The ship should use a power supply that can provide sufficient power for all components without excessive energy consumption.
  - The ship should have an efficient power management system to maximize battery life.

Environmental Requirements:

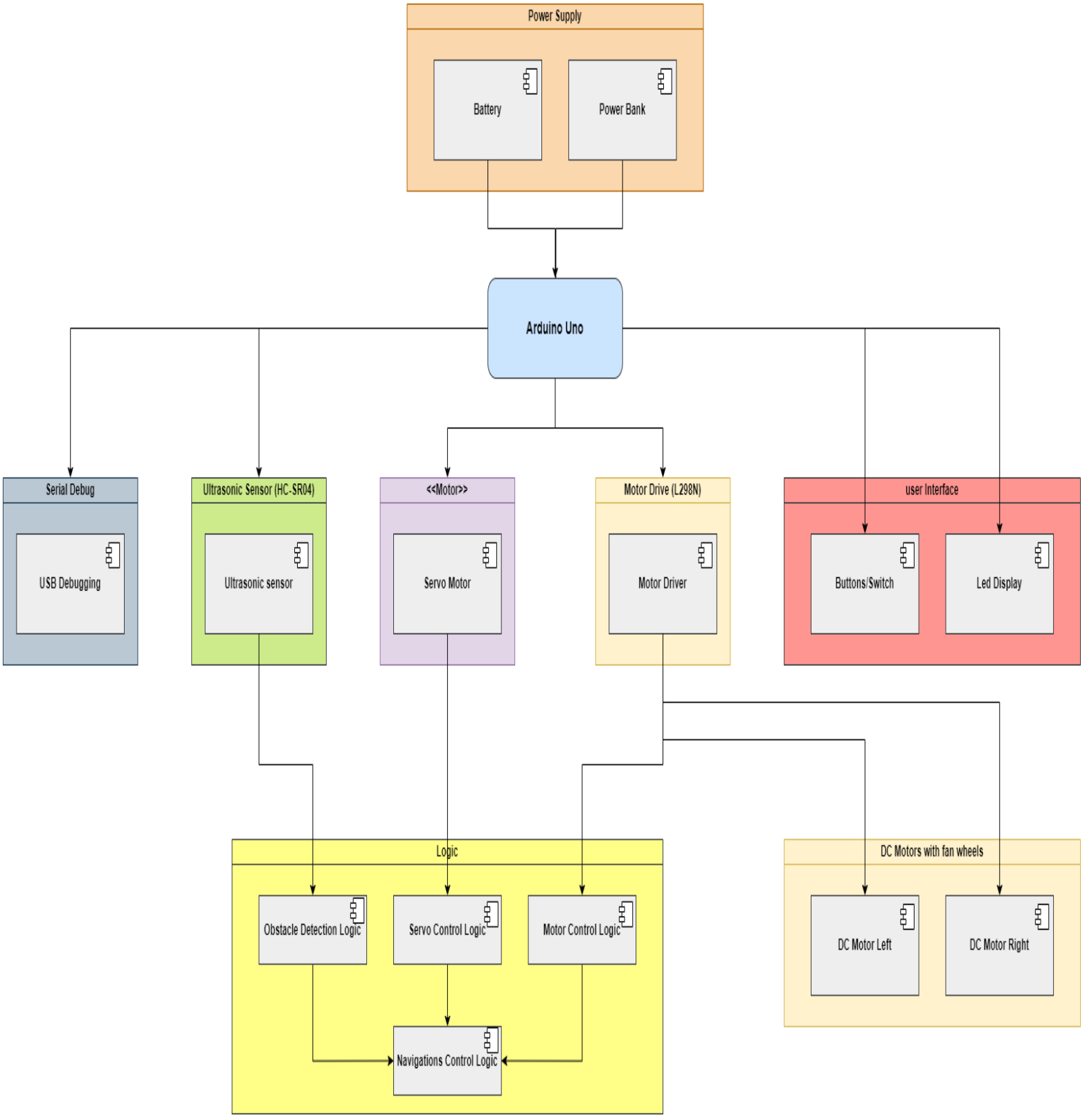
- **Operating Environment:**
  - The ship should be able to operate in indoor environments with flat surfaces.
  - The ship should be resistant to mild impacts and vibrations during navigation.

- **Temperature and Humidity:**
  - The ship should operate effectively within a temperature range of 0°C to 40°C.
  - The ship should be resistant to humidity levels up to 80%.

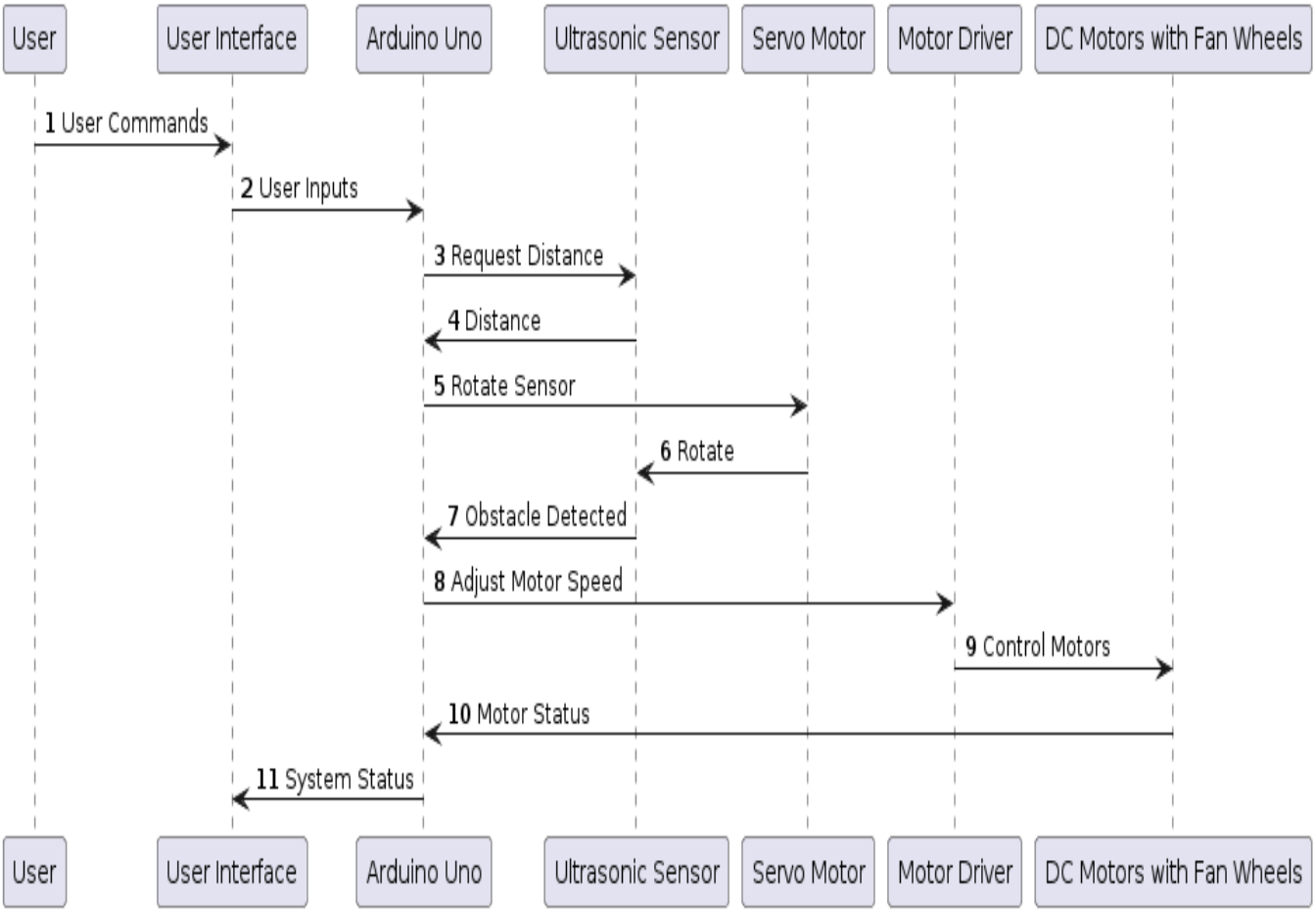
Data Flow Diagram:



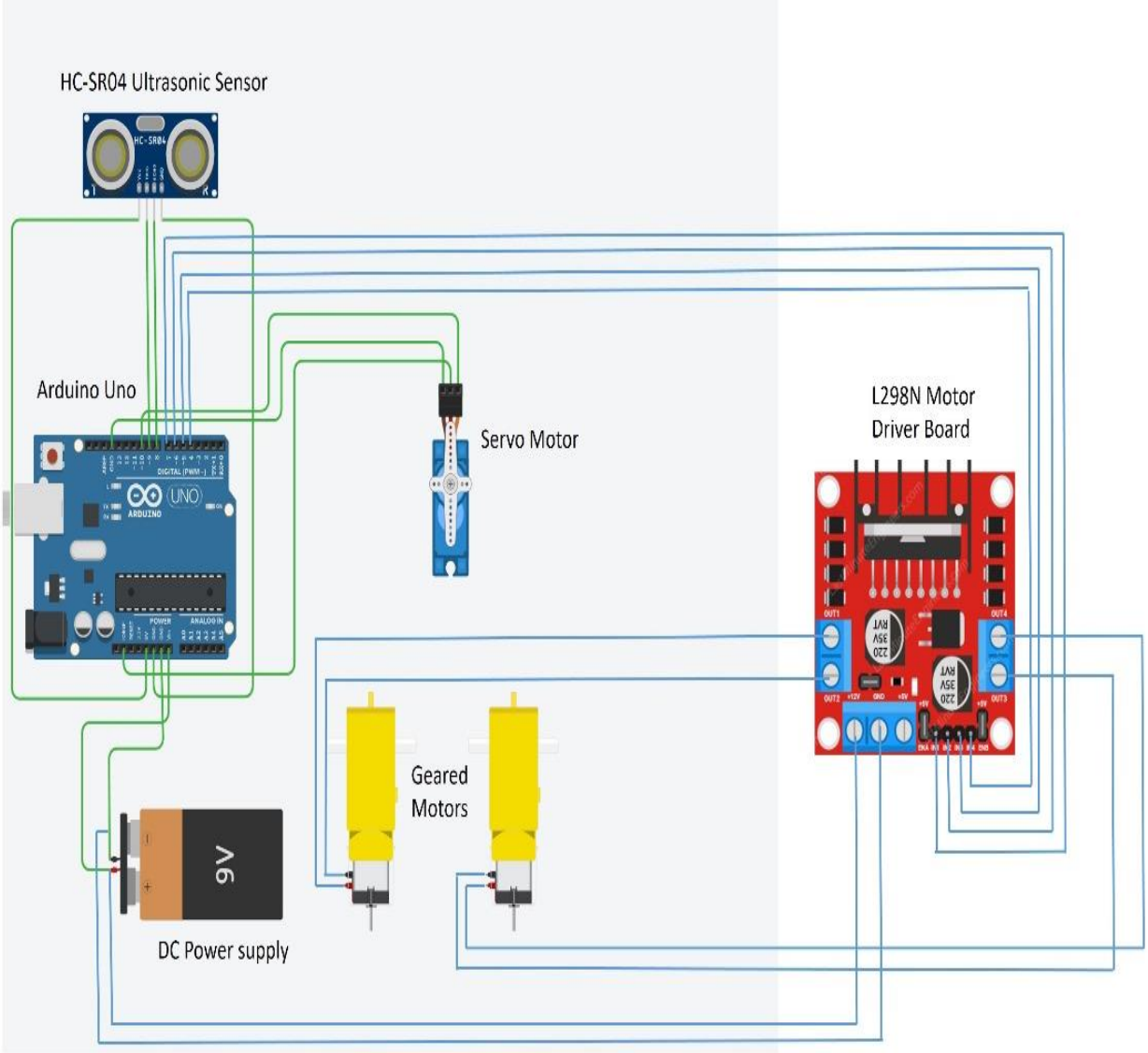
Block Diagram:



Sequence Diagram:



Circuit Diagram:



Obstacle_Avoiding Robot Ship Code:

```
#include <Servo.h>
Servo Myservo;

#define trigPin 9          // Trig Pin Of HC-SR04
#define echoPin 8        // Echo Pin Of HC-SR04
#define MLa 4                 //left motor 1st pin
#define MLb 5                //left motor 2nd pin
#define MRa 6               //right motor 1st pin
#define MRb 7               //right motor 2nd pin
#define servoPin 3         // Servo motor pin
#define brakePin 2         // Red LED pin
#define forwardPin 10      // Green LED pin

long duration, distance;

void setup() {
 Serial.begin(9600);
 pinMode(MLa, OUTPUT);     // Set Motor Pins As O/P
 pinMode(MLb, OUTPUT);
 pinMode(MRa, OUTPUT);
 pinMode(MRb, OUTPUT);
 pinMode(trigPin, OUTPUT);       // Set Trig Pin As O/P To Transmit Waves
 pinMode(echoPin, INPUT);       // Set Echo Pin As I/P To Receive Reflected Waves
```

```
 pinMode(brakePin, OUTPUT);     // Set Red LED pin as output
 pinMode(forwardPin, OUTPUT);   // Set Green LED pin as output
 Myservo.attach(servoPin);
}

void loop() {
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);      // Transmit Waves For 10us
 delayMicroseconds(10);
 duration = pulseIn(echoPin, HIGH);       // Receive Reflected Waves
 distance = duration / 58.2;                  // Get Distance
 Serial.println(distance);
 delay(10);
 if (distance > 30) {             // Condition For Absence Of Obstacle
   Myservo.write(90);
   digitalWrite(MRb, HIGH);      // Move Forward
   digitalWrite(MRa, LOW);
   digitalWrite(MLb, HIGH);
   digitalWrite(MLa, LOW);

   // Turn on green LED
   digitalWrite(forwardPin, HIGH);
 }
 else if ((distance < 20) && (distance > 0)) { // Condition For Presence Of Obstacle
```

```
digitalWrite(forwardPin, LOW);
  // Stop the motors
  digitalWrite(MRb, LOW);
  digitalWrite(MRa, LOW);
  digitalWrite(MLb, LOW);
  digitalWrite(MLa, LOW);

  // Turn on the brake LED
  digitalWrite(brakePin, HIGH);

  // Perform other actions (servo movements)
  delay(100);
  Myservo.write(0);
  delay(500);
  Myservo.write(180);
  delay(500);
  Myservo.write(90);
  delay(500);

  // Move backward
  digitalWrite(MRb, LOW);
  digitalWrite(MRa, HIGH);
  digitalWrite(MLb, LOW);
  digitalWrite(MLa, HIGH);
  delay(500);
```

```
  // Stop the motors again
  digitalWrite(MRb, LOW);
  digitalWrite(MRa, LOW);
  digitalWrite(MLb, LOW);
  digitalWrite(MLa, LOW);
  delay(100);

  // Move Left
  digitalWrite(MRb, HIGH);
  digitalWrite(MRa, LOW);
  digitalWrite(MLa, LOW);
  digitalWrite(MLb, LOW);
  delay(500);
  digitalWrite(brakePin, LOW);
  digitalWrite(forwardPin, HIGH);
}
```

Test case :

Test Case: Obstacle Detection and Navigation

Test Case ID:

- SHIP-TC001

Description:

- This test case verifies the obstacle detection and navigation functionality of the Arduino-powered autonomous ship. The test involves simulating obstacles in the ship's path and verifying that it successfully detects and navigates around them.

Precondition:

1. The ship's components are properly assembled and connected.
2. The Arduino code for obstacle detection and navigation is uploaded and running on the Arduino Uno.
3. The ship is placed in an open area with enough space to maneuver.

Test Steps:

1. Power on the Arduino-powered ship.
2. Ensure that the ultrasonic sensor is functioning properly and able to detect obstacles.
3. Place a large obstacle (e.g., a cardboard box) in front of the ship at a distance within the sensor's range.
4. Observe the ship's behavior:
    - Verify that the ship detects the obstacle using the ultrasonic sensor.
    - Confirm that the servo motor rotates the sensor to scan for obstacles.
    - Ensure that the ship adjusts its movement to avoid the obstacle.
    - Verify that the ship successfully navigates around the obstacle without collision.
5. Repeat steps 3-4 with obstacles placed at different distances and angles to test the ship's ability to detect and navigate around obstacles in various scenarios.
6. Repeat the test with multiple obstacles placed in succession to simulate a cluttered environment.
7. Power off the Arduino-powered ship after testing is complete.

Expected Results:

- The ship should detect obstacles using the ultrasonic sensor.
- The servo motor should rotate the sensor to scan for obstacles.

- The ship should adjust its movement to avoid obstacles.
- The ship should successfully navigate around obstacles without collision.
- In a cluttered environment, the ship should navigate around multiple obstacles without colliding with any of them.
- No errors or malfunctions should occur during the test.

Pass/Fail Criteria:

- The test passes if the ship successfully detects and navigates around obstacles in all test scenarios without collision.
- The test fails if the ship fails to detect obstacles, collides with obstacles, or exhibits unexpected behavior during navigation.

Test Environment:

- Arduino Uno microcontroller board.
- Ultrasonic sensor (HC-SR04).
- Servo motor.
- DC motors with fan wheels.
- Motor driver (L298N).
- Power supply.
- Open area for testing with obstacles.

Test Data:

- Distance of obstacles from the ship.
- Angle of obstacles relative to the ship's path.
- Number of obstacles placed during testing.

Test Duration:

- Approximately 30 minutes.