DLD PROJECT REPORT

SUBMITTED TO

Sir Adnan Jelani

SUBMITTED BY

Group Members	Enrollment #
Hasnain Ali	01-135232-024
Hamza Masood	01-135232-023
Babar Hussain Butt	01-135241-061

Contents

Abstract:

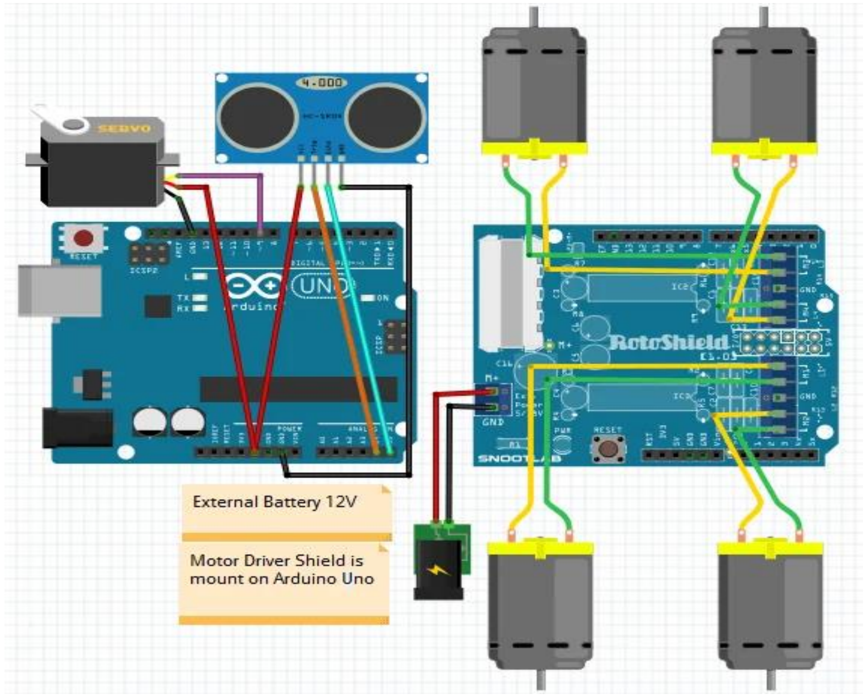
A four-wheel robotic car with object detection combines remote control and autonomous navigation. Equipped with sensors, it avoids obstacles and demonstrates hardware-software integration for educational and entertainment purposes.

Introduction:

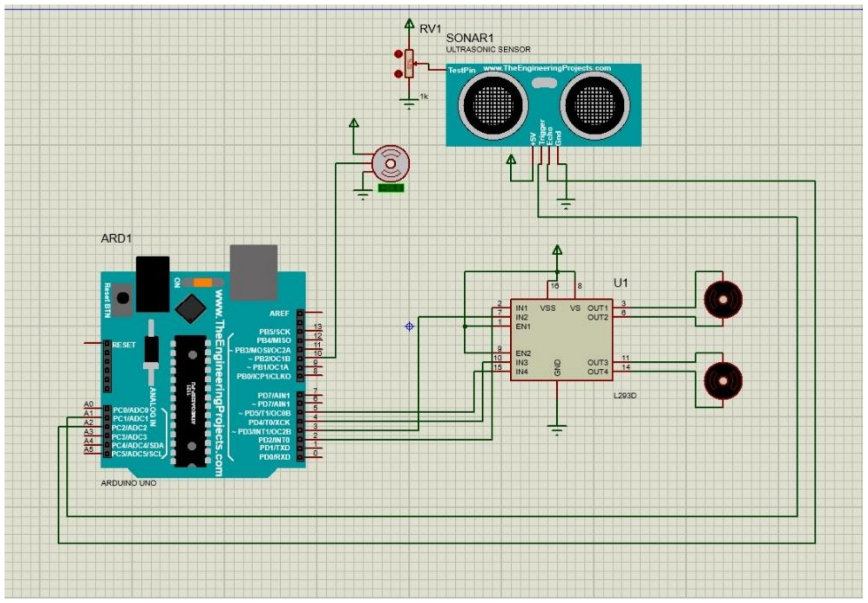
A four-wheel robotic car with object detection is an innovative project that blends remote control with autonomous navigation. It leverages sensors and programming to detect obstacles, showcasing the synergy of hardware and software in practical robotic applications.

System Design:

i. ***Block Diagram***:



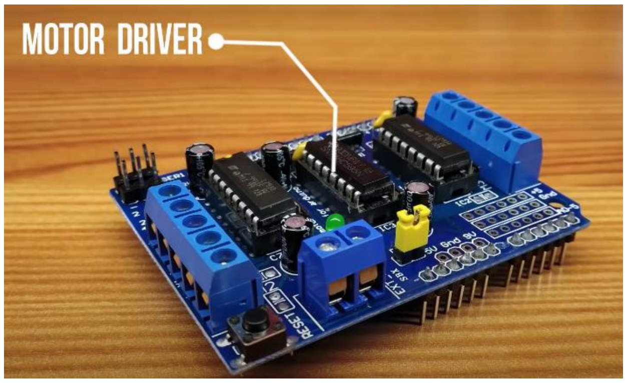
ii. ***Circuit Diagram:***



Material & Equipment:

i. **Microcontroller:** Such as an Arduino to serve as the brain of the car.

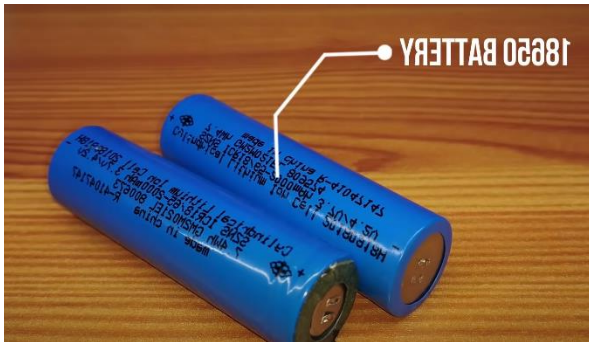
ii. **Motor Driver:** To control the speed and direction of DC motors.



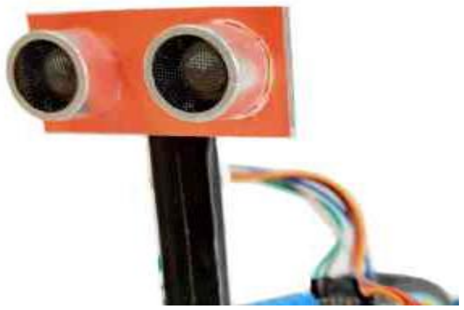
iii. **4 gear motors with wheels:** Motors that can be used to power the wheels of the car.

iv. **Wires:** Wires that can be used to connect the components of the car.

v. **Battery**: To power the microcontroller, sensors, and motors.



vi. **Sensors:** Ultrasonic or infrared sensors for object detection.



Implementation Details:

i. **Connect the Microcontroller and Motor Driver:**
Interface the microcontroller with the motor driver to enable control of the motors.

ii. **Attach Motors to Wheels:**
Secure the motors to the wheels and ensure they are properly aligned for smooth movement.

iii. **Power Connection:**
Connect the battery to the microcontroller and motor driver to provide necessary power.

iv. **Sensor Integration:**
Attach the object detection sensors to the car and connect them to the microcontroller for obstacle detection.

v. **Software Installation:**

Install and configure the necessary software on the microcontroller. This includes setting up the programming the control logic.

vi. **Configuration:**

Configure the car's network settings, including its IP address and the programming of the microcontroller it will connect to.

Results:

It detects obstacles using sensors and adjusts its path smoothly. The system demonstrates effective integration of hardware and software, achieving reliable movement and obstacle avoidance.

Challenges:

Challenges faced during this were the adjustment of sensor as the servo motor was unable to have a load on this so it was a quit bit challenging.

Conclusion:

An object detection car is a multifaceted project that integrates autonomous navigation, sensor technology, and control systems. It offers a practical and engaging way to learn about robotics and electronics, making it an excellent educational tool. Additionally, its ability to navigate and avoid obstacles autonomously makes it useful for various research and practical applications, preparing students and hobbyists for more advanced projects and careers in technology and engineering.

Application:

i. **Educational Purposes:**

This project can be used to teach students about digital logic design, robotics, and sensor technology, providing hands-on experience in these areas.

ii. **Research and Development:**

The car can serve as a platform for research into autonomous navigation and sensor integration.

iii. **Practical Applications:**
The principles learned from this project can be applied to developing more advanced autonomous systems, such as self-driving cars.

Software and Control

i. **Programming:**
Write code to control the car via Arduino, including routines for moving forward, backward, turning, and stopping. Implement object detection algorithms to autonomously avoid obstacles.

ii. **User Interface:**
Develop a user-friendly interface on the control device to send commands to the car and receive feedback.

iii. **Testing and Calibration:**
Test the car's movements and sensor accuracy, calibrating as necessary to ensure reliable performance.

Code:

```
#include <Servo.h>
#include <AFMotor.h>

#define Echo A0
#define Trig A1
#define motor 10
#define Speed 170
#define spoint 103

char value;
int distance;
int Left;
int Right;
int L = 0;
int R = 0;
int L1 = 0;
int R1 = 0;

Servo servo;
AF_DCMotor M1(1);
AF_DCMotor M2(2);
AF_DCMotor M3(3);
AF_DCMotor M4(4);

void setup() {
    Serial.begin(9600);
    pinMode(Trig, OUTPUT);
    pinMode(Echo, INPUT);
    servo.attach(motor);
    M1.setSpeed(Speed);
    M2.setSpeed(Speed);
    M3.setSpeed(Speed);
    M4.setSpeed(Speed);
}

void loop() {
    //Obstacle();
}

void Obstacle() {
    distance = ultrasonic();
    if (distance <= 12) {
        Stop();
        backward();
        delay(100);
        Stop();
        L = leftsee();
        servo.write(spoint);
        delay(800);
        R = rightsee();
        servo.write(spoint);
        if (L < R) {
            right();
            delay(500);
            Stop();
            delay(200);
        }
        else if (L > R) {
            left();
            delay(500);
            Stop();
            delay(200);
        }
    }
    else {
        forward();
    }
}

// Ultrasonic sensor distance reading function
int ultrasonic() {
    digitalWrite(Trig, LOW);
    delayMicroseconds(4);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
    long t = pulseIn(Echo, HIGH);
    long cm = t / 29 / 2; //time convert distance
    return cm;
}

void forward() {
    M1.run(FORWARD);
    M2.run(FORWARD);
    M3.run(FORWARD);
    M4.run(FORWARD);
}

void backward() {
    M1.run(BACKWARD);
    M2.run(BACKWARD);
    M3.run(BACKWARD);
    M4.run(BACKWARD);
}

void right() {
    M1.run(BACKWARD);
    M2.run(BACKWARD);
    M3.run(FORWARD);
    M4.run(FORWARD);
}

void left() {
    M1.run(FORWARD);
    M2.run(FORWARD);
    M3.run(BACKWARD);
    M4.run(BACKWARD);
}

void Stop() {
    M1.run(RELEASE);
    M2.run(RELEASE);
    M3.run(RELEASE);
    M4.run(RELEASE);
}

int rightsee() {
    servo.write(20);
    delay(800);
    Left = ultrasonic();
    return Left;
}

int leftsee() {
    servo.write(180);
    delay(800);
    Right = ultrasonic();
    return Right;
}
```