MICROCONTROLLER LAB
MINI PROJECT

Obstacle Avoider Robotic Vehicle

SUBMITTED BY:

EC01,BATCH-2,GROUP-5
AVINASH S-B230637EC
BADHON DATTA PROTTOY-B230101EC
AUSTIN BERT AMBOOKEN-B230858EC

1. Objective / Aim

The primary aim of this project is to develop an autonomous robotic vehicle that detects and avoids obstacles in its path without any human intervention. The robot uses an ultrasonic sensor to perceive obstacles and a microcontroller (8051) to process data and make movement decisions. This project introduces automation at a fundamental level and showcases how microcontrollers and sensors can be used for real-world intelligent navigation.

2. Introduction

In the age of automation, obstacle-avoiding robots represent a critical step towards intelligent, self-guided systems. These robots find applications in areas such as driverless vehicles, military patrol bots, warehouse robots, and assistive mobility solutions. The obstacle avoider robot moves forward unless it detects an obstacle in its path. Upon detection, the system evaluates alternative paths and changes direction accordingly.

This project harnesses the processing capability of the 8051 microcontroller and the sensing ability of ultrasonic modules to achieve obstacle avoidance with simple hardware and logic.

3. Theory

3.1 Microcontroller (8051)

The 8051 is an 8-bit microcontroller that consists of built-in RAM, ROM, timers, I/O ports, and serial communication. It is widely used in embedded systems for its ease of programming and reliability.

3.2 Ultrasonic Sensor (HC-SR04)

Ultrasonic sensors measure distance using sound waves. It emits a sound pulse and listens for its echo. By calculating the time taken for the echo to return, the sensor computes the distance between itself and an object.

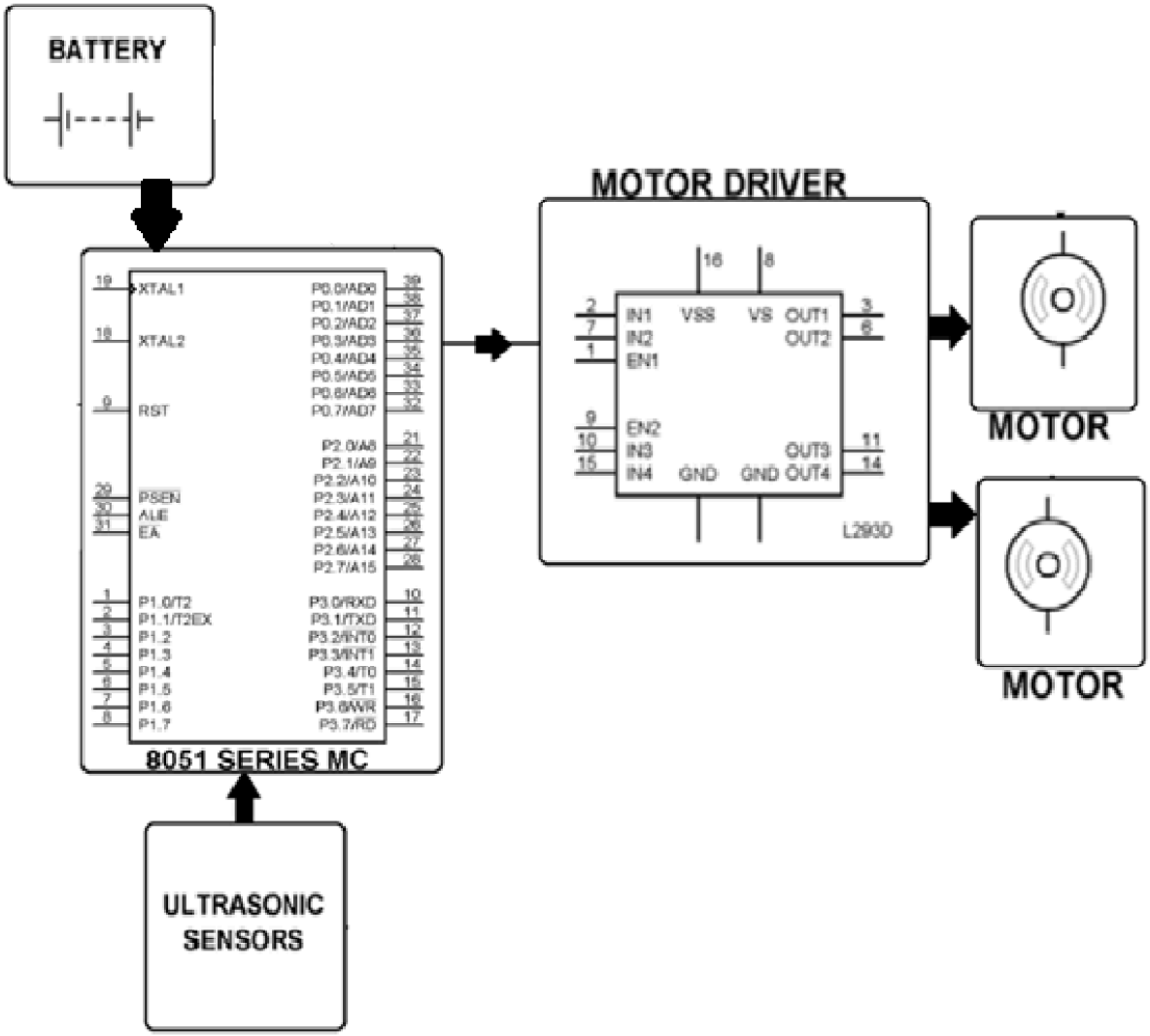
Distance formula:

Distance=Time×Speed of Sound2\text{Distance} = \frac{\text{Time} \times \text{Speed of Sound}}{2}Distance=2Time×Speed of Sound

3.3 Motor Driver (L293D)

The L293D IC is used to control the direction of DC motors. It acts as an interface between the microcontroller and motors, handling higher current and voltage.

4. Block Diagram/Circuit Diagram



5. Working Principle

The Obstacle Avoider Robotic Vehicle functions based on real-time input from an ultrasonic sensor and decision-making logic embedded in the 8051 microcontroller. The system works in a continuous loop to detect obstacles and navigate accordingly. Below is a detailed explanation of how the system operates:

Step 1: System Initialization

Once the power is turned on, the microcontroller initializes all the ports and components, including the ultrasonic sensor and the motor driver. The sensor is set up to begin sending out pulses to detect obstacles.

Step 2: Obstacle Detection Using Ultrasonic Sensor

The HC-SR04 ultrasonic sensor works by sending an ultrasonic pulse (triggered by the microcontroller) and then listening for its echo. The time taken for the echo to return is directly proportional to the distance of the object from the robot.

The 8051 microcontroller calculates this distance using the formula:

$$\text{Distance (cm)} = \text{Time } (\mu s) \times 0.0343/2$$

This data is constantly updated, allowing the robot to “see” its environment in real time.

Step 3: Decision Making by the Microcontroller

The microcontroller continuously checks the distance between the robot and any object in front of it.

- If the distance is greater than a predefined threshold (e.g., 15 cm), the microcontroller sends signals to the motor driver to move the robot forward.

- If the distance is less than the threshold, it interprets this as a potential collision, and the microcontroller immediately halts the forward movement.

Step 4: Directional Maneuvering

Upon detecting an obstacle, the robot performs the following logic:

1. Stop both motors.

2. Check alternative paths:

    - The microcontroller instructs the robot to briefly rotate or move sideways (left/right) to scan for a clearer direction using additional pulses.

3. Decision:

    - If the left side is clear, it turns left.

    - If the right side is clear, it turns right.

    - If both sides are blocked, it reverses slightly and tries again.

This simple but effective algorithm allows the robot to navigate around obstacles without human input.

Step 5: Motor Control via L293D

Based on the decision made, the 8051 sends digital signals to the L293D motor driver IC. This IC controls the direction and rotation of the DC motors connected to the robot wheels. Depending on the input pins (IN1 to IN4), the motor driver either moves the motors forward, backward, or turns the robot.

Step 6: Continuous Loop Operation

The entire process of:

- Sensing the environment

- Processing distance data

- Making movement decisions
- Controlling the motors

...runs in a continuous loop as long as the robot is powered on. This allows the robot to move autonomously through a space, avoiding obstacles dynamically.

6. C-Programming Code

```
#include <reg52.h>

// Define Motor Control Pins

sbit IN1 = P2^0;  // Left Motor Forward

sbit IN2 = P2^1;  // Left Motor Backward

sbit IN3 = P2^2;  // Right Motor Forward

sbit IN4 = P2^3;  // Right Motor Backward

sbit EN1 = P2^4;  // Enable Left Motor

sbit EN2 = P2^5;  // Enable Right Motor


// Define Ultrasonic Sensor Pins

sbit TRIG = P3^0;

sbit ECHO = P3^1;


// Function Prototypes

void delay(unsigned int ms);
```

```
unsigned int measure_distance();
void move_forward();
void move_backward();
void turn_left();
void turn_right();
void stop();

void main() {
    // Initialize Timer0 in Mode 1
    TMOD = 0x01;  // Timer0 Mode 1 (16-bit)
    TH0 = 0;
    TL0 = 0;

    // Enable Motors and Start Moving Forward Immediately
    EN1 = 1;  // Enable Left Motor
    EN2 = 1;  // Enable Right Motor
    IN1 = 1;  // Move Left Motor Forward
    IN2 = 0;
    IN3 = 1;  // Move Right Motor Forward
    IN4 = 0;
```

```
    while(1) {
        unsigned int distance = measure_distance();

        if (distance > 15 && distance < 400) {  // Ensure valid reading
            move_forward();  // Keep Moving Forward
        } else {
            stop();
            delay(500);
            turn_right();
            delay(500);
            move_forward();  // Continue after turning
        }
    }
}

// Measure distance using Ultrasonic Sensor
unsigned int measure_distance() {
    unsigned int time, distance;

    TRIG = 0;
    delay(2);
```

```
    TRIG = 1;
    delay(10);
    TRIG = 0;

    while (!ECHO);  // Wait for echo pulse to start
    TF0 = 0; // Clear Timer Overflow Flag
    TR0 = 1;  // Start Timer 0
    while (ECHO);  // Wait for echo pulse to end
    TR0 = 0;  // Stop Timer 0

    time = (TH0 << 8) | TL0;  // Get the time count
    distance = time / 58;  // Convert to cm

    if (distance > 400)  // If out of range, return max value
        return 400;

    return distance;
}

// Motor Movement Functions
void move_forward() {
```

```
    EN1 = 1; EN2 = 1; // Ensure Motors are Enabled
    IN1 = 1; IN2 = 0; IN3 = 1; IN4 = 0;
}

void move_backward() {
    EN1 = 1; EN2 = 1;
    IN1 = 0; IN2 = 1; IN3 = 0; IN4 = 1;
}

void turn_left() {
    EN1 = 1; EN2 = 1;
    IN1 = 0; IN2 = 1; IN3 = 1; IN4 = 0;
}

void turn_right() {
    EN1 = 1; EN2 = 1;
    IN1 = 1; IN2 = 0; IN3 = 0; IN4 = 1;
}

void stop() {
    EN1 = 0; EN2 = 0;  // Disable Motors to save power
```

```
    IN1 = 0; IN2 = 0; IN3 = 0; IN4 = 0;
}


// Delay function
void delay(unsigned int ms) {
    unsigned int i, j;
    for(i = 0; i < ms; i++)
        for(j = 0; j < 1275; j++);  // Adjusted for better accuracy
}
```

7. Software Used

- Keil µVision IDE
- Flash Magic
- Embedded C Language
- Proteus (optional)

8. Applications

The Obstacle Avoider Robotic Vehicle is a practical implementation of embedded systems and autonomous robotics. Its simple design and effective functionality make it suitable for a wide range of real-world applications:

1. Autonomous Navigation Systems

This robot can serve as a basic model for self-driving systems used in modern autonomous vehicles. Although it operates on simple logic, the

principle of obstacle detection and path correction is fundamental to autonomous driving technology.

2. Smart Wheelchairs

By integrating obstacle avoidance mechanisms, smart wheelchairs can help users navigate safely without bumping into walls or furniture. This improves the independence and mobility of individuals with disabilities.

3. Military and Surveillance Operations

In hostile environments where it's unsafe for humans to venture, robots like these can be deployed to navigate through unknown terrain, avoiding obstacles automatically. They can be equipped with cameras and sensors for surveillance and reconnaissance missions.

4. Rescue and Disaster Response

In disaster-struck areas, rubble or debris can block paths. An obstacle-avoiding robot can be used to move through tight spaces and deliver essentials or scan for trapped individuals, where manual entry is too dangerous.

5. Factory and Warehouse Automation

Autonomous robots are increasingly used in logistics and warehouse operations. Obstacle-avoiding vehicles can safely transport goods and components from one section to another without human supervision, avoiding collisions with people or other machines.

6. Educational and Research Projects

This project is widely used in academic settings to teach students about embedded systems, microcontrollers, sensor integration, and real-time programming. It lays the foundation for more advanced robotics and automation research.

7. Home Automation and Service Robots

Robots that can autonomously navigate a house without bumping into furniture or walls can serve as mobile assistants — delivering items, vacuuming, or even monitoring the environment for security.

9. Advantages

- Fully autonomous operation
- Low cost and power consumption
- Easy to build and modify
- Excellent educational project

10. Limitations

- Cannot detect transparent objects
- Limited pathfinding capability
- Basic obstacle avoidance without advanced logic

11. Future Enhancements

- Integration with AI and machine learning
- Camera for visual navigation
- GPS for location-based movement
- IoT connectivity for remote monitoring

12. Result and Conclusion

The project successfully demonstrates the working of an autonomous robotic vehicle that avoids obstacles using real-time sensor input and microcontroller logic. It acts as a foundation for more complex robotics and automation systems.