Project Report: Smart Arduino Car Using Ultrasonic Sensor

Team Members

- Srijan Shivakumar Kakhandaki - 1CR23EC213
- Sristi Purkayastha - 1CR23EC214
- Srushti Bennehalli Hanumesh - 1CR23EC214

Introduction

Our project involves building a smart car that uses an ultrasonic sensor to detect obstacles and navigate around them. The car has four wheels and a single ultrasonic sensor mounted on a servo motor. The sensor scans the left and right sides to determine the best path for the car to follow, ensuring it avoids obstacles. The objective was to create an autonomous vehicle capable of navigating through an environment without human intervention.

Challenges Faced

1. Weight Distribution

Balancing the weight of the car was one of the significant challenges. An uneven weight distribution caused stability issues, making the car prone to tipping over, especially when making turns. We had to carefully position the battery, motors, and other components to ensure an even weight distribution.

2. Torque

Another challenge was ensuring that the motors provided enough torque to move the car efficiently. Initially, we faced issues where the car struggled to start moving or slowed down significantly when encountering minor obstacles. By adjusting the motor placements and selecting appropriate motors, we managed to achieve the necessary torque.

3. Traction

Getting the right amount of traction was crucial. Without adequate traction, the car's wheels would slip, particularly on smooth surfaces. We experimented with different wheel materials and

surface textures to improve traction, eventually finding a suitable combination that allowed the car to move smoothly and reliably.

4. Motor Placement

Placing the motors in the right positions was vital for the car's movement and stability. Improper motor placement led to difficulties in steering and inconsistent movement. We adjusted the motor positions multiple times to find the optimal setup that allowed for smooth and controlled movement in all directions.

Functionality

The smart car uses an ultrasonic sensor attached to a servo motor to check distances on its left and right sides. The sensor measures the distance to obstacles and the car moves in the direction with more space. Here’s how it works:

- The car moves forward until it detects an obstacle within 15 cm.
- When an obstacle is detected, the car stops, moves backward, and then scans left and right.
- Based on the distance readings, the car turns towards the side with more space and continues moving forward.

Code Implementation

The following is the Arduino code used to control the smart car:

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

#define TRIG_PIN A0
#define ECHO_PIN A1
#define MAX_DISTANCE 200
#define MAX_SPEED 190
#define MAX_SPEED_OFFSET 20

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);
Servo myservo;
```

```
boolean goesForward = false;
int distance = 100;
int speedSet = 0;

void setup() {
  Serial.begin(9600);
  myservo.attach(10);
  myservo.write(115);
  delay(2000);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  Serial.println("Setup complete");
}

void loop() {
  int distanceR = 0;
  int distanceL = 0;
  delay(40);

  Serial.print("Distance: ");
  Serial.println(distance);

  if (distance <= 15) {
    moveStop();
    delay(100);
    moveBackward();
    delay(300);
    moveStop();
    delay(200);
    distanceR = lookRight();
    delay(200);
    distanceL = lookLeft();
    delay(200);

    Serial.print("Distance Right: ");
    Serial.println(distanceR);
    Serial.print("Distance Left: ");
```

```
    Serial.println(distanceL);

    if (distanceR >= distanceL) {
      turnRight();
      moveStop();
    } else {
      turnLeft();
      moveStop();
    }
  } else {
    moveForward();
  }
  distance = readPing();
}

int lookRight() {
  myservo.write(50);
  delay(500);
  int distance = readPing();
  delay(100);
  myservo.write(115);
  return distance;
}

int lookLeft() {
  myservo.write(170);
  delay(500);
  int distance = readPing();
  delay(100);
  myservo.write(115);
  return distance;
  delay(100);
}

int readPing() {
  delay(70);
  int cm = sonar.ping_cm();
  if (cm == 0) {
    cm = 250;
  }
  return cm;
}

void moveStop() {
```

```
  Serial.println("Stopping motors");
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}

void moveForward() {
  if (!goesForward) {
    goesForward = true;
    Serial.println("Moving forward");
    motor1.run(FORWARD);
    motor2.run(FORWARD);
    motor3.run(FORWARD);
    motor4.run(FORWARD);
    for (speedSet = 0; speedSet < MAX_SPEED; speedSet += 2) {
      motor1.setSpeed(speedSet);
      motor2.setSpeed(speedSet);
      motor3.setSpeed(speedSet);
      motor4.setSpeed(speedSet);
      delay(5);
    }
  }
}

void moveBackward() {
  goesForward = false;
  Serial.println("Moving backward");
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet += 2) {
    motor1.setSpeed(speedSet);
    motor2.setSpeed(speedSet);
    motor3.setSpeed(speedSet);
    motor4.setSpeed(speedSet);
    delay(5);
  }
}

void turnRight() {
  Serial.println("Turning right");
  motor1.run(FORWARD);
```

```
  motor2.run(FORWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
  delay(500);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}

void turnLeft() {
  Serial.println("Turning left");
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  delay(500);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
```

Sample Serial Monitor Output

Below is a sample output from the Arduino IDE serial monitor showing the car's distance measurements and actions taken:

```
Setup complete
Distance: 100
Moving forward
Distance: 85
Distance: 70
Distance: 50
Distance: 20
Distance: 10
Stopping motors
Moving backward
Distance Right: 80
Distance Left: 60
Turning right
Stopping motors
Distance: 100
Moving forward
```

Overview

The code controls a small car that can move forward, backward, and turn left or right. It uses an ultrasonic sensor to detect obstacles and a servo motor to help the sensor look around. When the car detects an obstacle, it decides which way to turn to avoid it and continues moving.

Main Parts of the Code

1. **Setup**
    - **Initializing Components**: The code starts by setting up the ultrasonic sensor, servo motor, and the car's four wheels. It also sets up communication with the computer so it can print messages (useful for debugging).
    - **Initial Sensor Readings**: The car takes a few distance readings with the sensor to make sure everything is working.
2. **Loop**
    - The main part of the code runs over and over again. This is where the car checks its surroundings and decides what to do.

How It Works Step-by-Step

1. **Moving Forward**:
    - The car starts by moving forward. It checks the distance to the nearest obstacle using the ultrasonic sensor.
2. **Detecting an Obstacle**:
    - If the sensor detects an obstacle closer than 15 cm, the car stops.
3. **Avoiding the Obstacle**:
    - The car moves backward a little to get some space.
    - Then, it uses the servo motor to turn the ultrasonic sensor to the right and takes a distance reading.
    - Next, it turns the sensor to the left and takes another distance reading.
    - It compares the distances to the right and left.
    - The car turns in the direction where there is more space. If the right side has more space, it turns right; if the left side has more space, it turns left.
4. **Continuing Forward**:
    - After turning, the car stops briefly and then continues moving forward, repeating the process.

Key Functions in the Code

- **readPing()**: Measures the distance to an obstacle using the ultrasonic sensor.
- **lookRight() and lookLeft()**: Turns the servo motor to look right or left and measures the distance.
- **moveForward()**: Moves the car forward.
- **moveBackward()**: Moves the car backward.

- **turnRight() and turnLeft()**: Turns the car right or left.
- **moveStop()**: Stops the car.

Serial Monitor Output

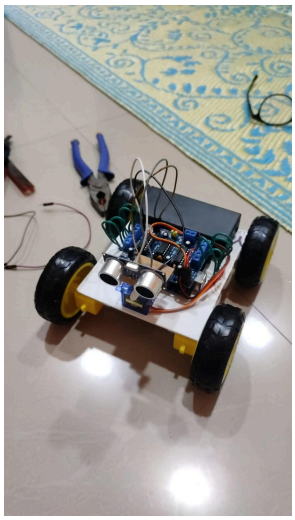
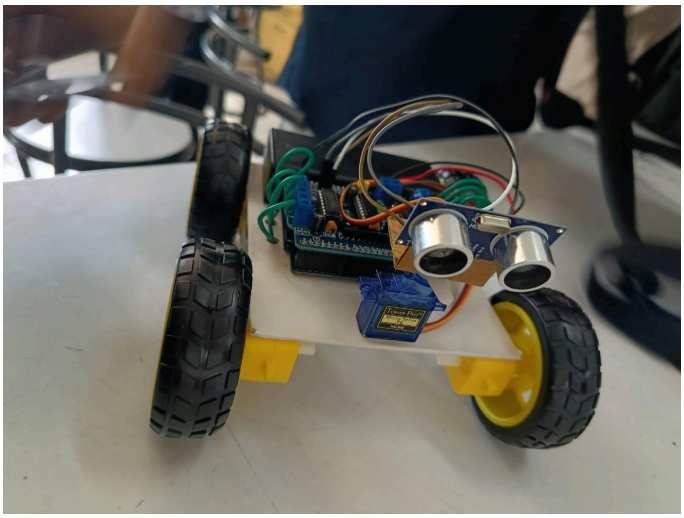
The car also prints messages to the computer screen (serial monitor) to show what it's doing. For example:

- "Distance: 100" means the car measured a distance of 100 cm to the nearest obstacle.
- "Moving forward" means the car is moving forward.
- "Stopping motors" means the car has stopped.
- "Turning right" means the car is turning right to avoid an obstacle.

Conclusion

Building this smart car was a valuable learning experience. We overcame challenges related to weight distribution, torque, traction, and motor placement, which improved our understanding of robotics and autonomous systems. The project demonstrates how an ultrasonic sensor can effectively guide an autonomous vehicle, paving the way for more advanced developments in this field. I am also happy to say that our team has won one of the best mini project awards in the project exhibition at CMRIT.

A Few Clicks Of The Model:


PERFECTLY
Imperfect