AUTOMATION & ROBOTICS

PROJECT REPORT

OBSTACLE AVOIDING ROBOT

Presented by
Maheen Aslam

PROJECT OVERVIEW

This project is a simple, cost-effective obstacle avoiding robot using an Arduino Uno. The robot uses an ultrasonic sensor to detect obstacles and autonomously navigate around them using two gear motors for movement and a servo motor to rotate the sensor. It is powered by lithium batteries and built without a traditional baseboard, making it lightweight and compact.

COMPONENTS AND SPECIFICATIONS

The robotic system comprises key components that enable its movement, control, and functionality. Two DC gear motors, each costing Rs. 350, are used to provide propulsion to the vehicle, making them essential for mobility. A motor driver shield, priced at Rs. 500, is responsible for controlling the speed and direction of these gear motors, ensuring coordinated movement. The Arduino Uno board, which costs Rs. 850, serves as the central processing unit that executes control algorithms through its digital and analog I/O pins. A standard hobby servo motor, costing Rs. 250, is included for precise control of steering or sensor orientation using PWM signals. To establish reliable electrical connections between all components, four jumper wires (male-to-male and male-to-female) are used, each priced at Rs. 5, totaling Rs. 20. Power is supplied through three rechargeable lithium batteries, each suited to meet the system's voltage and current demands. Although listed as three units, the cost calculation accounts for two batteries, amounting to Rs. 360. Together, these components create a functional robotic system capable of autonomous or semi-autonomous operations.

CODING

```
#include <AFMotor.h> // AFMotor Library
https://learn.adafruit.com/adafruit-
motor-shield/library-install
#include <NewPing.h> // NewPing Library
https://github.com/livetronic/Arduino-
NewPing
#include <Servo.h>  // Servo Library
https://github.com/arduino-
libraries/Servo.git

#define TRIG_PIN A0
#define ECHO_PIN A1
#define MAX_DISTANCE 200
#define MAX_SPEED 120
#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);
Servo myservo;

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

void setup() {

 myservo.attach(10);
 myservo.write(115);
 delay(2000);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
}

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

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

 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;
}
```

CODING

```
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() {
 motor1.run(RELEASE);
 motor2.run(RELEASE);
 }

void moveForward() {

 if(!goesForward)
 {
  goesForward=true;
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  for (speedSet = 0; speedSet <
MAX_SPEED; speedSet +=2)
  {
  motor1.setSpeed(speedSet);
  motor2.setSpeed(speedSet);
  delay(5);
  }
 }
}
```

```
void moveBackward() {
  goesForward=false;
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
 for (speedSet = 0; speedSet <
MAX_SPEED; speedSet +=2)
 {
  motor1.setSpeed(speedSet);
  motor2.setSpeed(speedSet);
  delay(5);
 }
}

void turnRight() {
 motor1.run(FORWARD);
 motor2.run(BACKWARD);
 delay(450);
 motor1.run(FORWARD);
 motor2.run(FORWARD);
}

void turnLeft() {
 motor1.run(BACKWARD);
 motor2.run(FORWARD);
 delay(450);
 motor1.run(FORWARD);
 motor2.run(FORWARD);

}
```

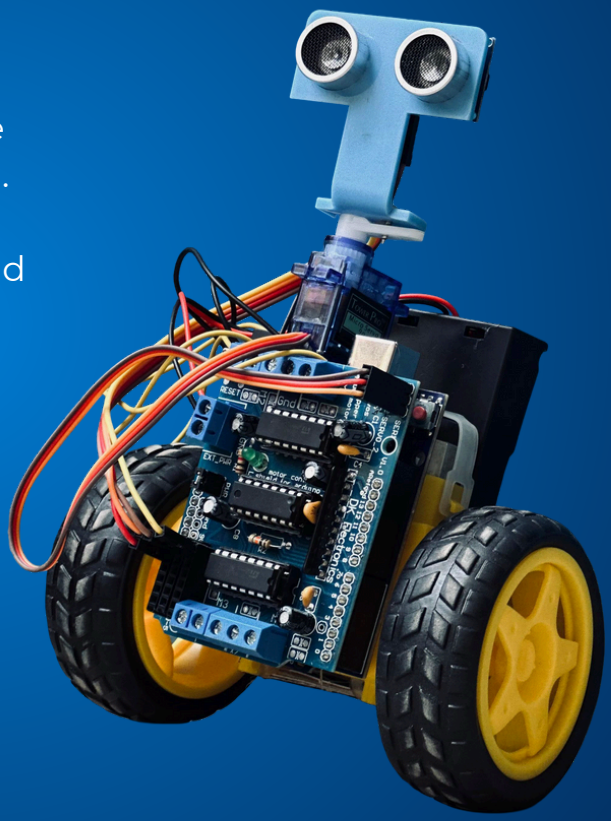
TOTAL COST BREAKDOWN

COMPONENT	QUANTITY	UNIT COST COMPONENT QUANTITY IN (Rs.)	TOTAL COST IN (RS.)
REVENUE	2	350	700
MOTOR DRIVER SHIELD	1	500	500
ARDUINO UNO	1	850	850
SERVO MOTOR	1	250	250
JUMPER WIRES	4	5	20
LITHIUM BATTERIES	2	180	360
CASTOR WHEEL	1	180	180
BATTERY HOLDER	1	200	200
TOTAL			3060

ASSEMBLY & FUNCTIONALITY

FUNCTIONALITY

Assembly and Functionality The two-wheeled robotic vehicle is built without a traditional chassis; all components are directly mounted onto the gear motors and supporting structure. Gear motors are installed symmetrically for balanced movement, and all components are connected using jumper wires to the motor driver shield and Arduino Uno. The power supply is provided by lithium batteries, carefully configured to meet voltage and current requirements. The servo motor, attached to the Arduino, enables sensor orientation.



WORKING PRINCIPLE

The robot uses an ultrasonic sensor mounted on a servo to detect obstacles. When an object is sensed within a defined range, the robot halts and scans left and right for a clearer path. It then turns accordingly, using its motors, to avoid the obstacle. The Arduino Uno manages all logic and control, making the robot fully autonomous.

CONCLUSION

This two-wheeled robotic vehicle project serves as an excellent introduction to robotics and embedded systems. It encompasses mechanical assembly, electrical wiring, and programming, providing a holistic learning experience. The total cost of Rs. 3060 makes it an affordable project for enthusiasts and students interested in robotics.