MPMC Project

Title-Obstacle Avoiding Car using Arduino

Group Members:

1) Rasika Bonde – BE20F04F006
2) Vaishnavi Deshmukh- BE20F04F013
3) Vaishnavi Kale- BE20F04F031
4) Anushka Suradkar- BE20F04F058
5) Namrata Kasabe- BE21S04F007

Introduction: The obstacle avoidance car is used for detecting obstacles and avoiding the collision. This is an autonomous car. The design of the obstacle avoidance car requires the integration of many sensors according to their task. Obstacle detection is the primary requirement of this autonomous car. The car gets the information from the surrounding area through mounted sensors on the car.

Components:

1) Arduino Uno

2) Motor Driver

3) Wheels(4x)

4) TT Gear Motor(4x)

5) Servo Motor

6) Ultrasonic Sensor

7) 18650 Li-on Battery (2x)

8) Male and Female Jumper Wires

9) Acrylic Sheet

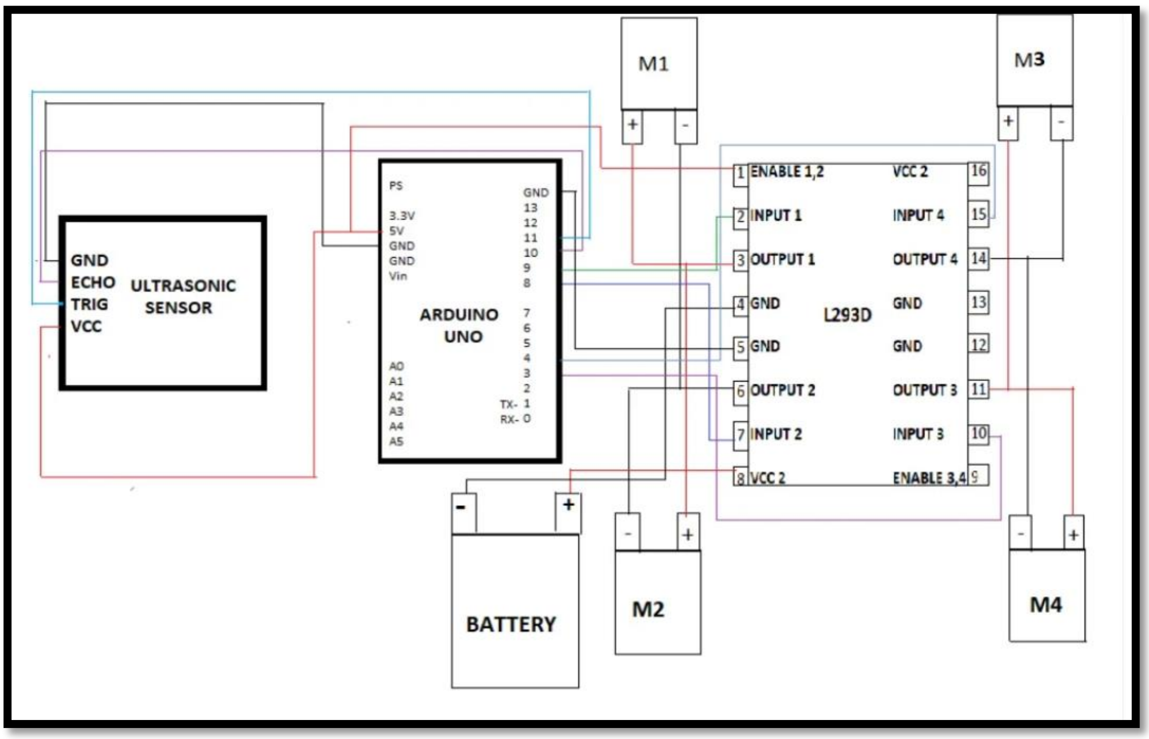
10) DC Power Switch

- Motor Driver- The Arduino L293D motor driver shield guide is a robotics project that involves driving various types of motors. The most common types used for robotic applications include DC, servo, and stepper motors. However, these motors typically cannot be driven directly by Arduino or another microcontroller. This is because of their higher current and power ratings, so motor shields or driver ICs are used instead. These shields or ICs isolate a motor's power supply and use control logic from the microcontroller circuitry. One of the most popular motor driver shields used with Arduino is the L293D. The full-featured L293D motor driver shield can control up to four bi-directional DC motors with 8-bit speed selection, two stepper motors, and two servo motors. The L293D motor driver shield includes two L293 motor driver ICs and a 74HC595 shift register IC. The shield has several important components. The L293D is a dual-channel H-bridge motor driver that can control two DC motors or a stepper motor at one time. As there are two L293D ICs on the shield, it's technically capable of controlling a total of four DC motors. This is ideal for two and four-wheel robot platforms. The IC consists of two H-bridge to control the motors, each delivering p to 0.6A to a motor.

- Ultrasonic Sensor- Ultrasonic sensor working principle is either similar to sonar or radar which evaluates the target/object attributes by understanding the received echoes from sound/radio waves correspondingly. These sensors produce high-frequency sound waves and analyse the echo which is received from the sensor. The sensors measure the time interval between transmitted and received echoes so that the distance to the target is known.
  Ultrasonic Sensor Specifications:

- The sensing range lies between 40 cm to 300 cm.
- The response time is between 50 milliseconds to 200 milliseconds.
- The Beam angle is around 50.
- It operates within the voltage range of 20 VDC to 30 VDC
- Preciseness is ±5%
- The frequency of the ultrasound wave is 120 kHz
- Resolution is 1mm
- The voltage of sensor output is between 0 VDC – 10 VDC
- The ultrasonic sensor weight nearly 150 grams

Block Diagram:



Working: The obstacle avoidance robotic vehicle uses ultrasonic sensors for its movements. Arduino is used to achieve the desired operation.

The motors are connected through motor driver IC to Arduino. The ultrasonic sensor is attached in front of the robot.

Whenever the robot is going on the desired path the ultrasonic sensor transmits the ultrasonic waves continuously from its sensor head.

Whenever an obstacle comes ahead of it the ultrasonic waves are reflected back from an object and that information is passed to the Arduino uno.

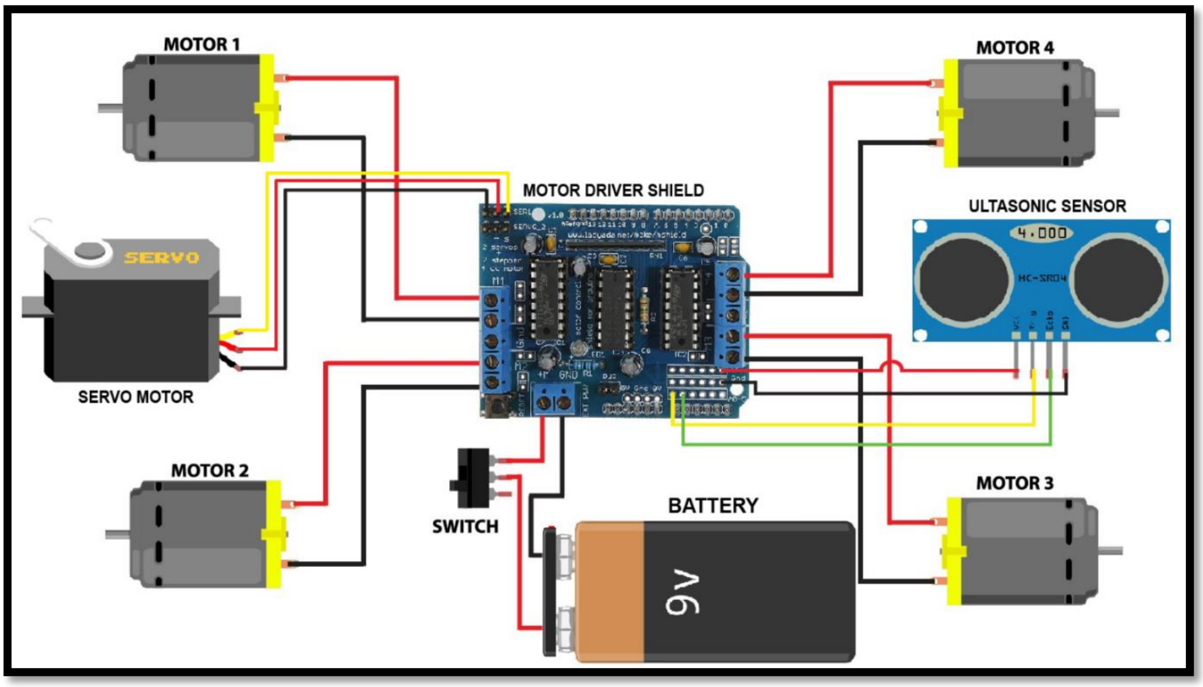
The Arduino controls the motors left, right, back, front, based on ultrasonic signals. In order to control the speed of each motor pulse width modulation is used (PWM).

When ultrasonic sensor detect the object which is kept inside the path it will send the signal toward the Arduino uno and according to that it will rotate the motor.

M3 & M4 in forward direction and rotate the motor MI & M2 in reverse direction such way that the car get moving in left direction.

Similarly in every time when ever an obstacle in found to be in path of car it will detect it and rotate the car in left direction to avoid the obstacle.

Circuit Diagram:



Program:

```
#include <AFMotor.h>

#include <Servo.h>
```

```
AF_DCMotor rightBack(1);
AF_DCMotor rightFront(2);
AF_DCMotor leftFront(3);
AF_DCMotor leftBack(4);
Servo servoLook;

byte trig = 2;
byte echo = 13;
byte maxDist = 150;
byte stopDist = 50;
float timeOut = 2*(maxDist+10)/100/340*1000000;

byte motorSpeed = 55;
int motorOffset = 10;
int turnSpeed = 50;



void setup()
{
  rightBack.setSpeed(motorSpeed);
  rightFront.setSpeed(motorSpeed);
  leftFront.setSpeed(motorSpeed+motorOffset);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);
  rightFront.run(RELEASE);
  leftFront.run(RELEASE);
  leftBack.run(RELEASE);
  servoLook.attach(10);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
}

void loop()
{
  servoLook.write(90);
  delay(750);
  int distance = getDistance();
  if(distance >= stopDist)
  {
    moveForward();
  }
  while(distance >= stopDist)
  {
    distance = getDistance();
    delay(250);
  }
  stopMove();
  int turnDir = checkDirection();
  Serial.print(turnDir);
  switch (turnDir)
  {
    case 0:
      turnLeft (400);
      break;
    case 1:
      turnLeft (700);
      break;
    case 2:
      turnRight (400);
      break;
  }
}

void accelerate()
{
  for (int i=0; i<motorSpeed; i++)
  {
    rightBack.setSpeed(i);
    rightFront.setSpeed(i);
    leftFront.setSpeed(i+motorOffset);
    leftBack.setSpeed(i+motorOffset);
    delay(10);
  }
}

void decelerate()
{
  for (int i=motorSpeed; i!=0; i--)
  {
    rightBack.setSpeed(i);
    rightFront.setSpeed(i);
    leftFront.setSpeed(i+motorOffset);
    leftBack.setSpeed(i+motorOffset);
    delay(10);
  }
}

void moveForward()
{
  rightBack.run(FORWARD);
  rightFront.run(FORWARD);
  leftFront.run(FORWARD);
  leftBack.run(FORWARD);
}

void stopMove()
{
  rightBack.run(RELEASE);
  rightFront.run(RELEASE);
  leftFront.run(RELEASE);
  leftBack.run(RELEASE);
}

void turnLeft(int duration)
{
  rightBack.setSpeed(motorSpeed+turnSpeed);
  rightFront.setSpeed(motorSpeed+turnSpeed);
  leftFront.setSpeed(motorSpeed+motorOffset+turnSpeed);
  leftBack.setSpeed(motorSpeed+motorOffset+turnSpeed);
  rightBack.run(FORWARD);
  rightFront.run(FORWARD);
  leftFront.run(BACKWARD);
  leftBack.run(BACKWARD);
  delay(duration);
  rightBack.setSpeed(motorSpeed);
  rightFront.setSpeed(motorSpeed);
  leftFront.setSpeed(motorSpeed+motorOffset);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);
  rightFront.run(RELEASE);
  leftFront.run(RELEASE);
  leftBack.run(RELEASE);

}

void turnRight(int duration)
{
  rightBack.setSpeed(motorSpeed+turnSpeed);
  rightFront.setSpeed(motorSpeed+turnSpeed);
  leftFront.setSpeed(motorSpeed+motorOffset+turnSpeed);
  leftBack.setSpeed(motorSpeed+motorOffset+turnSpeed);
  rightBack.run(BACKWARD);
  rightFront.run(BACKWARD);
  leftFront.run(FORWARD);
  leftBack.run(FORWARD);
  delay(duration);
  rightBack.setSpeed(motorSpeed);
  rightFront.setSpeed(motorSpeed);
  leftFront.setSpeed(motorSpeed+motorOffset);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);
  rightFront.run(RELEASE);
  leftFront.run(RELEASE);
  leftBack.run(RELEASE);
}

int getDistance()
{
  unsigned long pulseTime;
  int distance;
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  pulseTime = pulseIn(echo, HIGH, timeOut);
  distance = (float)pulseTime * 340 / 2 / 10000;
  return distance;
}

int checkDirection()
{
  int distances [2] = {0,0};
  int turnDir = 1;
  servoLook.write(180);
  delay(500);
  distances [0] = getDistance();
  servoLook.write(0);
  delay(1000);
  distances [1] = getDistance();
  if (distances[0]>=200 && distances[1]>=200)
    turnDir = 0;
  else if (distances[0]<=stopDist && distances[1]<=stopDist)
    turnDir = 1;
  else if (distances[0]>=distances[1])
    turnDir = 0;
  else if (distances[0]<distances[1])
    turnDir = 2;
  return turnDir;
}
```

Cost Estimation:

Components:

Arduino UNO: ₹865

Servo motor: ₹265

Wheels and gear motors: ₹532

L293: ₹225

18650 Li-ion Battery(7.4v): ₹599

Switch: ₹40

Jumpers and screws: ₹100

Ultrasonic sensor: ₹185

Body: ₹200

Total Cost- ₹ 3000