Project Report On

Obstacle Avoiding Car: (using ARDUINO UNO)

"A dissertation submitted in partial fulfillment of the requirements of 6th Semester 2022 examination in Information Technology of the Maulana Abul Kalam Azad University of Technology"

Submitted by:

AVISHAK BAGCHI (10200220045)

Department of Information Technology
(Kalyani Government Engineering College)

(Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal)

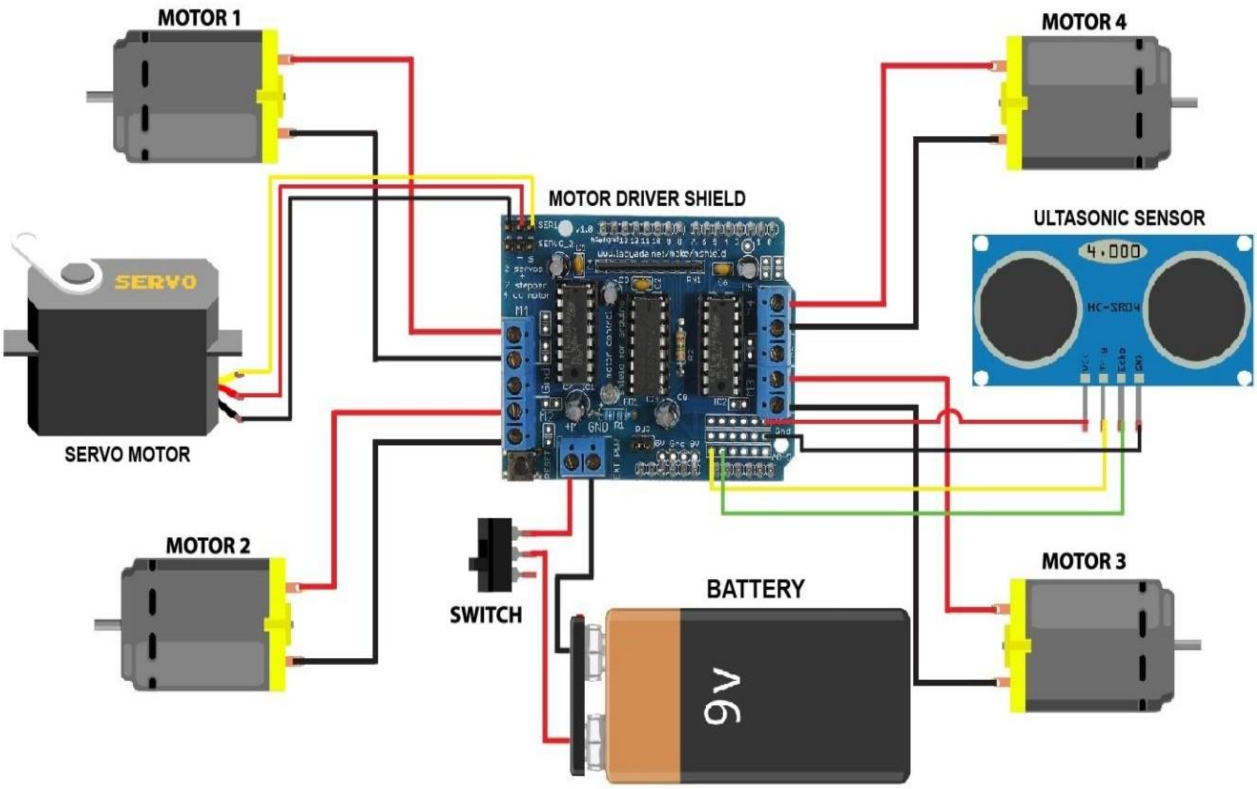
Kalyani - 741235, Nadia,

Abstract :-

Almost all basic electronics robot car work based on a circuit such as microcontroller. Another way to say all type of robot operate by a various type of controller circuit like (Arduino uno, Arduino nano, simple 8051 microcontroller, etc…). Here this robot car which can Sense obstacle around it run way and avoid this obstacle to find a free or open way.

Initially switch on the car due 9v dc supply turn on the Arduino as well as motor driver shield. Then 4 motor can rotated and car goes to forward direction and also ultrasonic sensor get a pulse (+5v). that time if a obstacle came in front of car then car will be stop Now servo start to rotated about 120 degree across and the sensor try to find a free way to car move right or left direction .otherwise the car move forward cover some distance according to uploading code into Arduino module .

Circuit diagram :-



Circuit diagram of obstacle avoiding robot car.

Hardware Required:-

- Arduino Uno
- Ultrasonic Range Finder Sensor – HC – SR04
- Motor Driver Shield (L293D)
- Servo Motor (Tower Pro SG90)
- Geared Motors x 4
- Robot Chassis
- **Power Supply :-**
- Battery Connector
- Battery Holder

Component Description:-

Arduino Uno:-

Arduino Uno is an AT mega 328p Microcontroller based prototyping board. It is an open source electronic prototyping platform that can be used with various sensors and actuators. arduino Uno has 14 digital I/O pins out of which 6 pins are usedin this project.

HC – SR04:-

HC-SR04 Ultrasonic Sensor – Working:-

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4pin module, whose pin names are VCC, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver.
The sensor works with the simple high school formula that

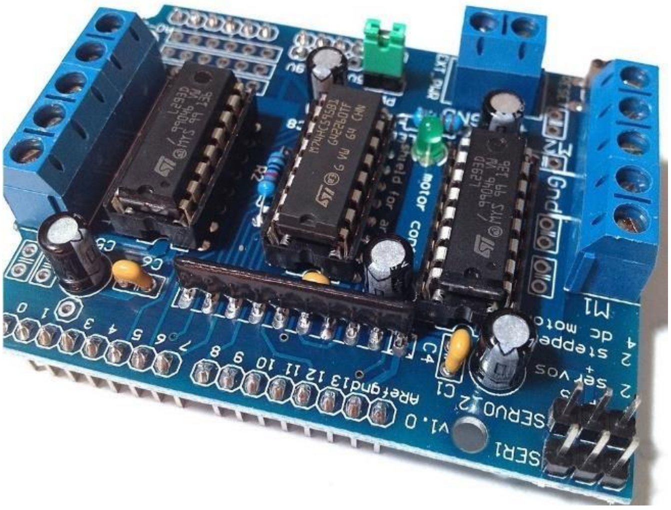
Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below.

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Nowsimply calculate the distance using a microcontroller or microprocessor

Motor driver shield(L293D):-

The L293D is quadruple high-current half-H drive.It is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to driveinductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other



high-current/high-voltage loads in positive-supply applications. All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

Servo Motor :-

The Tower Pro SG90 is a simple Servo Motor which can rotate 90 degrees in each direction (approximately 180 degrees in total).

Design of Obstacle Avoiding Robot using Arduino:-

Arduino is the main processing unit of the robot. Out of the 14 available digital I/O pins, 7 pins are used in this project design.

The ultrasonic sensor has 4 pins: Vcc, Trig, Echo and Gnd. Vcc and Gnd are connected to the +5v and GND pins of the arduino. Trig (Trigger) is connected to the 9th pin and Echo is connected to 8th pin of the arduino UNO respectively.

A Servo Motor is used to rotate the Ultrasonic Sensor to scan for obstacles. It has three pins namely Control, VCC and GND. The Servo Control Pin is connected to pin 11 of arduino while the VCC and GND are connected to +5V and GND.

L293D is a 16 pin IC. Pins 1 and 9 are the enable pins. These pins are connected to +5V. Pins 2 and 7 are control inputs from microcontroller for first motor. They are connected to pins6 and 7 of arduino respectively.

Similarly, pins 10 and 15 are control inputs from microcontroller for second motor. They are connected to pins 5 and 4 of arduino. Pins 4, 5, 12 and 13 of L293D are ground pins and areconnected to Gnd.

First motor (consider this as the motor for left wheel) is connected across the pins 3 and 6 of L293D. The second motor, which acts as the right wheel motor, is connected to 11and 14 pins of L293D.

The 16th pin of L293D is Vcc1. This is connected to +5V. The 8th pins is Vcc2. This is the motor supply voltage. This can be connected anywhere between 4.7V and 36V. In this project, pin 8 if L293D is connected to +5V supply.

How to it’s working:-

Before going to working of the project, it is important to understand how the ultrasonic sensor works. The basic principle behind the working of ultrasonic sensor is as follows:

Using an external trigger signal, the Trig pin on ultrasonic sensor is made logic high for at least 10µs. A sonic burst fromthe transmitter module is sent. This consists of 8 pulses of 40KHz.

The signals return back after hitting a surface and the receiverdetects this signal. The Echo pin is high from the time of sending the signal and receiving it. This time can be converted to distance using appropriate calculations.

The aim of this project is to implement an obstacle avoiding robot using ultrasonic sensor and arduino. All the connections are made as per the circuit diagram. The working of the project is explained below.

When the robot is powered on, both the motors of the robot will run normally and the robot moves forward. During this time, the ultrasonic sensor continuously calculate the distance between the robot and the reflective surface.

This information is processed by the arduino. If the distance between the robot and the obstacle is less than 15cm, the Robot stops and scans in left and right directions for new distance using Servo Motor and Ultrasonic Sensor. If the distance towards the left side is more than that of the right side, the robot will prepare for a left turn. But first, it backs up a little bit and then activates the Left Wheel Motor in reversed in direction.

Similarly, if the right distance is more than that of the left distance, the Robot prepares right rotation. This process continues forever and the robot keeps on moving without hittingany obstacle.

Code will be running through Arduino board (Uno) :-

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


#define TRIG_PIN A0
#define ECHO_PIN A1
#define MAX_DISTANCE 200
#define MAX_SPEED 250// sets speed of DC  motors
#define MAX_SPEED_OFFSET 20
NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);
```

```
AF_DCMotor motor1(1, MOTOR12_64KHZ);
AF_DCMotor motor2(2, MOTOR12_64KHZ);
AF_DCMotor motor3(3, MOTOR34_64KHZ);
AF_DCMotor motor4(4, MOTOR34_64KHZ);
Servo myservo;

boolean goesForward=false;
int distance = 200;
int speedSet = 0;
void setup() {

  myservo.attach(10);
  myservo.write(120);
  delay(1000);
  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<=45)
 {
  moveStop();
  delay(100);
  moveBackward();
  delay(200);
  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(60);
    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(100);
  int cm = sonar.ping_cm();
  if(cm==0)
  {
    cm = 250;
  }
  return cm;
}

void moveStop() {
  motor1.run(RELEASE);
```

```
 motor2.run(RELEASE);
 motor3.run(RELEASE);
 motor4.run(RELEASE);
 }

void moveForward() {

 if(!goesForward)
  {
   goesForward=true;
   motor1.run(FORWARD);
   motor2.run(BACKWARD);
   motor3.run(BACKWARD);
   motor4.run(FORWARD);
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) //
slowly bring the speed up to avoid loading down the batteries too
quickly
  {
   motor1.setSpeed(speedSet);
   motor2.setSpeed(speedSet);
   motor3.setSpeed(speedSet);
   motor4.setSpeed(speedSet);
   delay(5);
  }
```

```
  }
}

void moveBackward() {
    goesForward=false;
    motor1.run(BACKWARD);
    motor2.run(FORWARD);
    motor3.run(FORWARD);
    motor4.run(BACKWARD);
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly
bring the speed up to avoid loading down the batteries too quickly
  {
    motor1.setSpeed(speedSet);
    motor2.setSpeed(speedSet);
    motor3.setSpeed(speedSet);
    motor4.setSpeed(speedSet);
    delay(5);
  }
}

void turnRight() {
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
```

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

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

Conclusion :-

This paper present a simple, cost effective obstacle detection and avoidance system for an unmanned land mover. Two pairs of heterogonous sensors were employed to detect obstacles along the path of the mobile robot. A degree of accuracy and minimum probability of failure were obtained.

The evaluation on the autonomous system shows that it is capable of avoiding obstacles, ability to avoid collision and change its position. It is evident that, with this design more functionality can be added to this design to perform various functions with little or no intervention of humans. Finally, the robot was made to be remote controlled using an IR receiver and a remote controller. This project will be helpful in hostile environment, defense and security sectors of the country.

COMPONENT REQUIRE & COST:-

SL NO.	COMPONEN TREQUIRE	NO OF COMPONEN T	COST
1.	Arduino Uno	1	370
2.	Motor Driver Shield	1	150
3.	TT Gear Motor	4	200
4.	Servo Motor (SG 90)	1	120
5.	Ultrasonic Sensor	1	110
6.	(14500 800mAh) Li-on 3.74v Battery	3	120

7.	14500 Battery Holder	1	50
8.	Male & female jumper wire	4	8
9.	Dc power Switch	1	12
10.	Wire	3ft	8
11.	Wood sheet	1	12
12.	Sensor Mounting Bracket	1	20
13.	Glue Stick	4	40
14.	Rubber wheel	4	120
			Total Cost = 1340