OBSTACLE AVOIDING CAR

by

KAVYA PALIWAL 20BRS1111
ANIKET KUMAR PAUL 20BRS1116
SAVIO SAJAN MOLOPARAMBIL 20BRS1161

A project report submitted to

Dr. B. Chanthini

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

CSE2006 –MICROPROCESSOR & INTERFACING

in

B. Tech. COMPUTER SCEINCE & ENGINEERING

Vandalur – Kelambakkam Road

Chennai – 600127

APRIL 2022

BONAFIDE CERTIFICATE

Certified that this project report entitled "**OBSTACLE AVOIDING CAR"** is a bonafide work of **Kavya Paliwal-20BRS1111, Aniket Kumar Paul-20BRS1116 and Savio Sajan Moloparambil-20BRS1161** who carried out the Project work under my supervision and guidance for **CSE2006-MICROPROCESSOR & INTERFACING.**

Dr. B. CHANTHINI

Assistant Professor

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

This project involves the design and implementation of an intelligent obstacle-avoiding robot car. The objective of this project is to implement a robot car, which while moving should have the ability to detect obstacles in its path and change direction where obstacles are present without any form of external influence. The new direction to be taken to avoid collision is the direction that has the most distance between the obstacle and the sensor and this is determined by the robot based on sensor inputs. This implementation was done using an ultrasonic wave sensor/ ir sensor, which measures distance by sending pulses. Also, the movement of the servo motor (for sensor movement) and the DC motors (for wheel movement) are controlled by the motor driver shield in order to enable the obstacle avoidance function. The commands are sent to the Arduino microcontroller chip which serves as the main control of the robot car, as it controls the sensor and car movement. The implemented robot car was able to successfully detect and avoid obstacles within the line of sight of the Ultrasonic sensor used.

ACKNOWLEDGEMENT

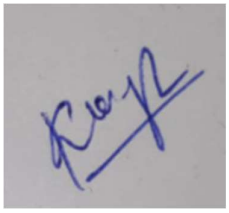
We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. B. Chanthini,** Assistant Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to Dean of School of Electronics Engineering & Computer science Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

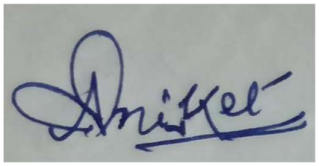
We express our thanks to our Head of the Department for her support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.



KAVYA PALIWAL



ANIKET KUMAR PAUL

SAVIO SAJAN MOLOPARAMBIL

TABLE OF CONTENTS

1. INTRODUCTION

1.1 OBJECTIVES AND GOALS

A necessary requirement of every autonomous mobile robot is obstacle avoidance. This obstacle avoidance feature is of high importance in a robot's navigation system in an unknown area so as to prevent collisions during its operation. It is necessary for an autonomous robot to avoid collisions in order to prevent damage to the object or to the robot itself. Application areas where obstacle avoidance is necessary include automatic vacuum cleaners and helicopters.

The robot will be designed to detect the presence of any object within the specified threshold distance. If any object is found within this distance, it is designated as an obstacle and the robot will turn away from it. The sensor emit an ultrasonic pulse every 300 ms which echoes from the neighboring objects. Using time difference between the input and echo, the Arduino calculates the distance to the obstacle from which the echo is coming by using the constant speed of sound 340 m/s. When one of the sensors detects obstacle within the threshold distance, the robot changes its direction.

The robot car should have the capacity to detect obstacles in its path based on a predetermined threshold distance. After detection of an obstacle, the robot should be able to change its direction to a relatively open path by making an autonomous decision. The robot car should not require any external control during its operation. The robot car should be able to measure distance between itself and an obstacle in real time. The robot car should be able to operate effectively in an environment which is unknown to it.

1.2 APPLICATIONS

- Especially military applications
- It can be used for city wars
- Automated lawn mover
- Smart room cleaner
- Obstacle avoiding robots can be used in almost all mobile robot navigation systems

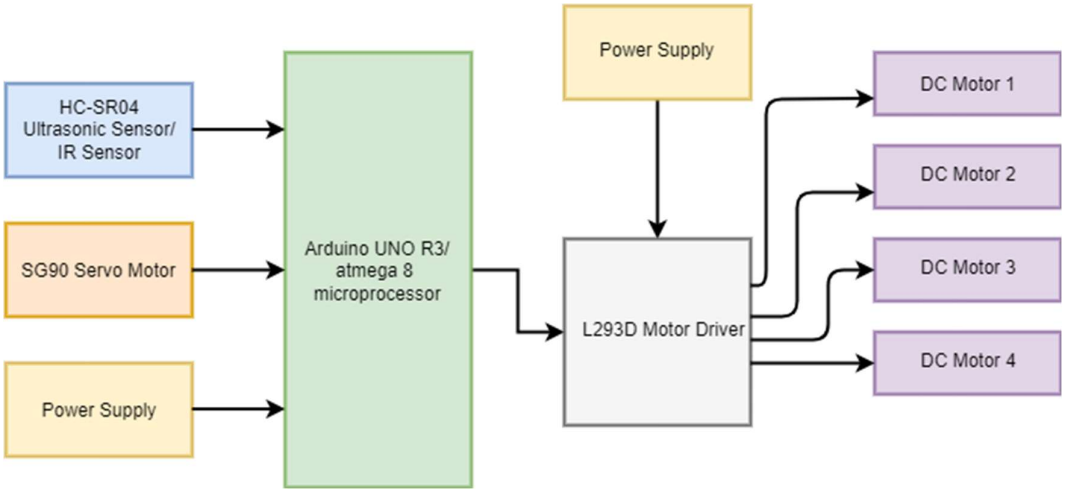
- They can also be used in dangerous environments, where human penetration could be fatal
- Unmanned vehicle driving
- Mining Vehicle that uses Obstacle Detection

1.3 FEATURES

- Cheap
- Uses ultrasonic sensor to determine the distance

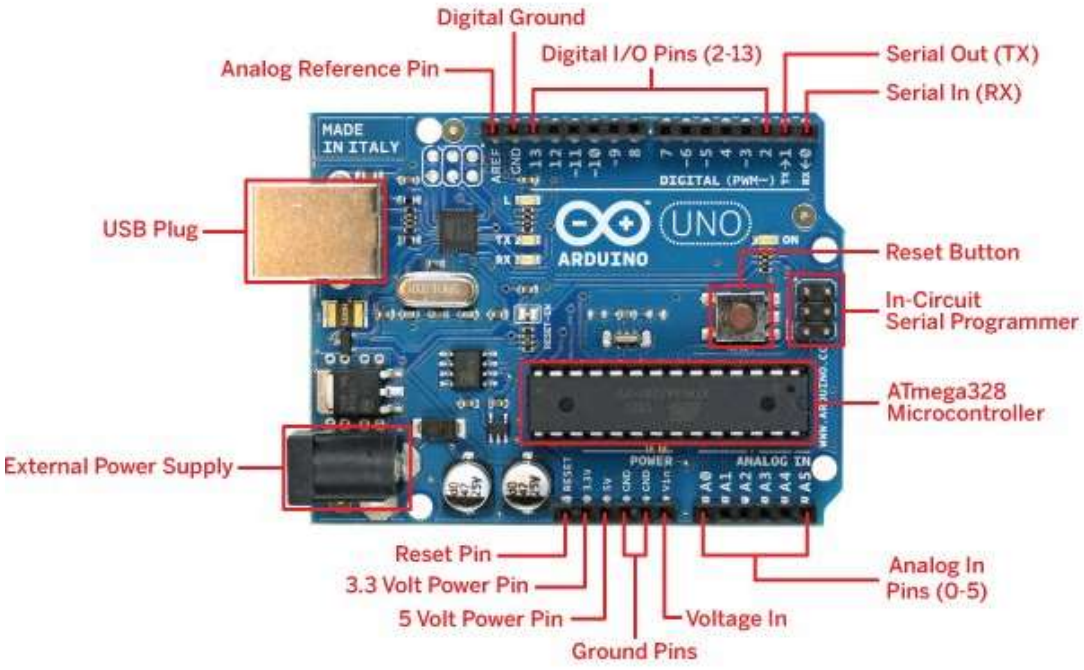
2. DESIGN

2.1 BLOCK DIAGRAM



2.2 HARDWARE ANALYSIS

- **Arduino Board**



Arduino boards are generally based on microcontrollers from Atmel Corporation like 8, 16 or 32 bit AVR architecture based microcontrollers.

The important feature of the Arduino boards is the standard connectors. Using these connectors, we can connect the Arduino board to other devices like LEDs or add-on modules called Shields.

The Arduino boards also consists of on board voltage regulator and crystal oscillator. They also consist of USB to serial adapter using which the Arduino board can be programmed using USB connection.

In order to program the Arduino board, we need to use IDE provided by Arduino. The Arduino IDE is based on Processing programming language and supports C and C++.

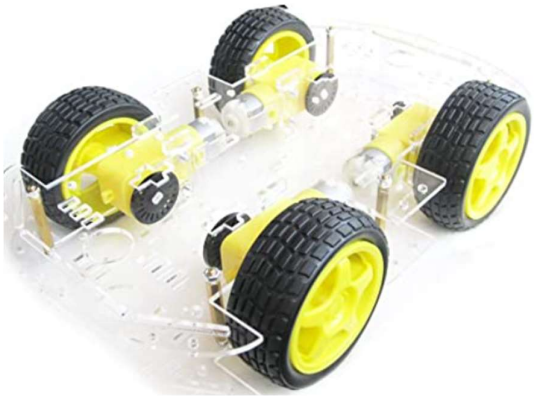
- **Ultrasonic Sensor**

It is an ultrasonic sensor, also known as an ultrasonic transducer that is based on a transmitter and receiver and mainly used to determine the distance from the target object. It mainly depends on the sound waves working on “non-contact” technology. The required distance of the target object is measured without any damage, giving you accurate and precise details.
This sensor comes with a range between 2cm to 400cm and is used in a wide range of applications including speed and direction measurement, wireless charging,humidifiers, medical ultrasonography, sonar, burglar alarms, and non-destructive testing.

- **4 wheel robot car kit**

For making the car move when we give them instructions accordingly. Four wheels smart robot car with independent DC motor,independently drivable. More flexible control.

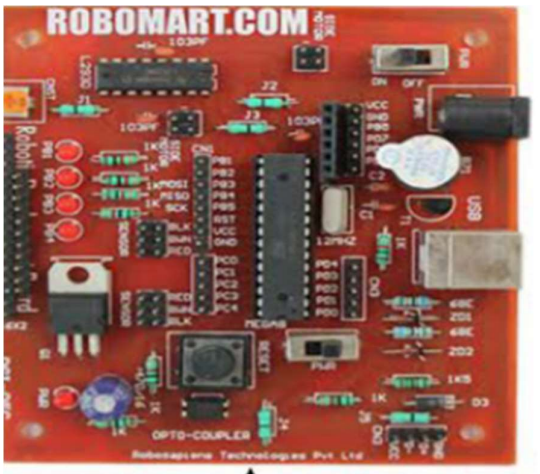
Smart robot car chassis kit for double chassis,more space,more mounting holes for most sensors andmodules. Car chassis easy to install speed measurement module, tracking module,Obstacle Avoidance Module, radar module, etc.



- **SG90 SERVO MOTOR**

We used this to rotate the ultrasonic sensor and to capture the distance from every end

- **ATmega8 mini V4 Development Board.**

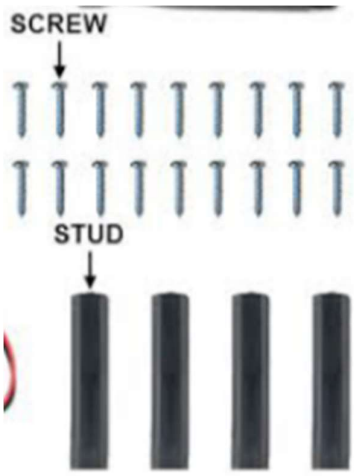


ATmega8 mini V4 Development Board)

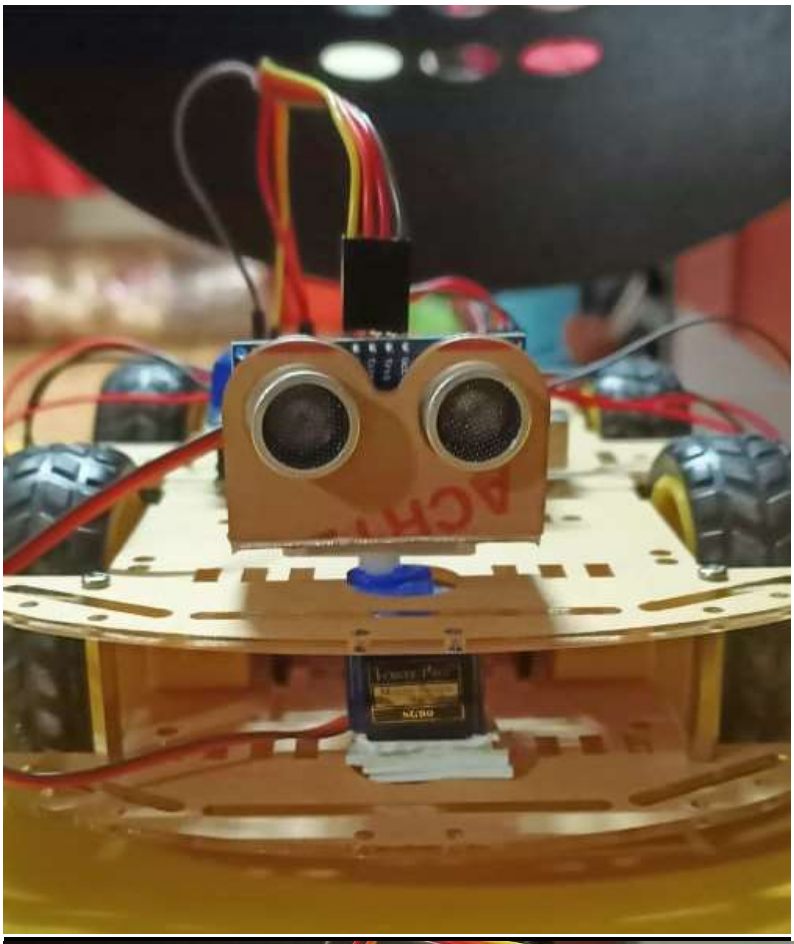
Another type of microprocessor used by us for developing our prototype

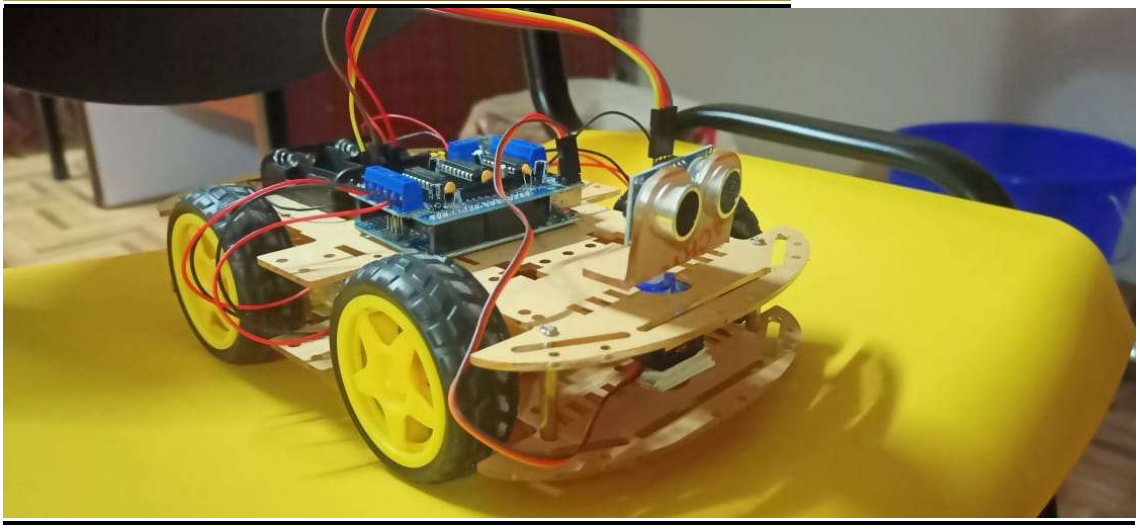
- **Motor**

- **SCREW AND STUDS**



2.3 SNAPSHOTS OF PROJECT AND RESULT





3. SOFTWARE

3.1 CODING AND ANALYSIS

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

#define TRIG_PIN A0 // Pin A0 on the Motor Drive Shield soldered to the ultrasonic sensor
#define ECHO_PIN A1 // Pin A1 on the Motor Drive Shield soldered to the ultrasonic sensor
#define MAX_DISTANCE 300 // sets maximum useable sensor measuring distance to 300cm
#define MAX_SPEED 160 // sets speed of DC traction motors to 150/250 or about 70% of full speed - to get power drain down.
#define MAX_SPEED_OFFSET 40 // this sets offset to allow for differences between the two DC traction motors
#define COLL_DIST 30 // sets distance at which robot stops and reverses to 30cm
#define TURN_DIST COLL_DIST+20 // sets distance at which robot veers away from object
NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE); // sets up sensor library to use the correct pins to measure distance.

AF_DCMotor leftMotor1(1, MOTOR12_1KHZ); // create motor #1 using M1 output on Motor Drive Shield, set to 1kHz PWM frequency
AF_DCMotor leftMotor2(2, MOTOR12_1KHZ); // create motor #2, using M2 output, set to 1kHz PWM frequency
AF_DCMotor rightMotor1(3, MOTOR34_1KHZ);// create motor #3, using M3 output, set to 1kHz PWM frequency
AF_DCMotor rightMotor2(4, MOTOR34_1KHZ);// create motor #4, using M4 output, set to 1kHz PWM frequency
Servo myservo;  // create servo object to control a servo

int leftDistance, rightDistance; //distances on either side
int curDist = 0;
String motorSet = "";
int speedSet = 0;

//------------------------------------------ SETUP LOOP ----------------------------------
-------------------------------------
```

```
void setup() {
 myservo.attach(10);   // attaches the servo on pin 10 (SERVO_1 on the Motor Drive Shield to the servo object
 myservo.write(90); // tells the servo to position at 90-degrees ie. facing forward.
 delay(1000); // delay for one seconds
}

void loop() {
 myservo.write(90);  // move eyes forward
 delay(90);
 curDist = readPing();   // read distance
 if (curDist < COLL_DIST) {changePath();}  // if forward is blocked change direction
 moveForward();  // move forward
 delay(500);
}

void changePath() {
 moveStop();   // stop forward movement
 myservo.write(36);  // check distance to the right
  delay(500);
  rightDistance = readPing(); //set right distance
  delay(500);
  myservo.write(144);  // check distace to the left
  delay(700);
  leftDistance = readPing(); //set left distance
  delay(500);
  myservo.write(90); //return to center
  delay(100);
  compareDistance();
 }


void compareDistance()   // find the longest distance
{
 if (leftDistance>rightDistance) //if left is less obstructed
 {
  turnLeft();
 }
 else if (rightDistance>leftDistance) //if right is less obstructed
 {
  turnRight();
 }
  else //if they are equally obstructed
```

```
 {
  turnAround();
 }
}


int readPing() { // read the ultrasonic sensor distance
 delay(70);
 unsigned int uS = sonar.ping();
 int cm = uS/US_ROUNDTRIP_CM;
 return cm;
}
void moveStop() {leftMotor1.run(RELEASE); leftMotor2.run(RELEASE);
rightMotor1.run(RELEASE); rightMotor2.run(RELEASE);}  // stop the motors.
void moveForward() {
   motorSet = "FORWARD";
   leftMotor1.run(FORWARD);     // turn it on going forward
   leftMotor2.run(FORWARD);     // turn it on going forward
   rightMotor1.run(FORWARD);    // turn it on going forward
   rightMotor2.run(FORWARD);    // turn it on going forward
 for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the speed up
to avoid loading down the batteries too quickly
 {
   leftMotor1.setSpeed(speedSet);
   leftMotor2.setSpeed(speedSet);
   rightMotor1.setSpeed(speedSet);
   rightMotor2.setSpeed(speedSet);
   delay(5);
 }
}
void moveBackward() {
   motorSet = "BACKWARD";
   leftMotor1.run(BACKWARD);    // turn it on going backward
   leftMotor2.run(BACKWARD);    // turn it on going backward
   rightMotor1.run(BACKWARD);   // turn it on going backward
   rightMotor2.run(BACKWARD);   // turn it on going backward
 for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the speed up
to avoid loading down the batteries too quickly
 {
   leftMotor1.setSpeed(speedSet);
   leftMotor2.setSpeed(speedSet);
   rightMotor1.setSpeed(speedSet);
```

```
    rightMotor2.setSpeed(speedSet);
    delay(5);
  }
}
void turnRight() {
  motorSet = "RIGHT";
  leftMotor1.run(FORWARD);      // turn motor 1 forward
  leftMotor2.run(FORWARD);      // turn motor 2 forward
  rightMotor1.run(BACKWARD);    // turn motor 3 backward
  rightMotor2.run(BACKWARD);    // turn motor 4 backward
  rightMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
  rightMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
  delay(1500); // run motors this way for 1500
  motorSet = "FORWARD";
  leftMotor1.run(FORWARD);      // set both motors back to forward
  leftMotor2.run(FORWARD);
  rightMotor1.run(FORWARD);
  rightMotor2.run(FORWARD);
}

void turnLeft() {
  motorSet = "LEFT";
  leftMotor1.run(BACKWARD);     // turn motor 1 backward
  leftMotor2.run(BACKWARD);     // turn motor 2 backward
  leftMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
  leftMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
  rightMotor1.run(FORWARD);     // turn motor 3 forward
  rightMotor2.run(FORWARD);     // turn motor 4 forward
  delay(1500); // run motors this way for 1500
  motorSet = "FORWARD";
  leftMotor1.run(FORWARD);      // turn it on going forward
  leftMotor2.run(FORWARD);      // turn it on going forward
  rightMotor1.run(FORWARD);     // turn it on going forward
  rightMotor2.run(FORWARD);     // turn it on going forward
}

void turnAround() {
  motorSet = "RIGHT";
  leftMotor1.run(FORWARD);      // turn motor 1 forward
  leftMotor2.run(FORWARD);      // turn motor 2 forward
  rightMotor1.run(BACKWARD);    // turn motor 3 backward
  rightMotor2.run(BACKWARD);    // turn motor 4 backward
  rightMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
```

```
 rightMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
 delay(1700); // run motors this way for 1700
 motorSet = "FORWARD";
 leftMotor1.run(FORWARD);     // set both motors back to forward
 leftMotor2.run(FORWARD);
 rightMotor1.run(FORWARD);
 rightMotor2.run(FORWARD);
}
```

obstacle_avoiding

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

#define TRIG_PIN A0 // Pin A0 on the Motor Drive Shield soldered to the ultrasonic sensor
#define ECHO_PIN A1 // Pin A1 on the Motor Drive Shield soldered to the ultrasonic sensor
#define MAX_DISTANCE 300 // sets maximum useable sensor measuring distance to 300cm
#define MAX_SPEED 160 // sets speed of DC traction motors to 150/250 or about 70% of full speed - to get power drain down.
#define MAX_SPEED_OFFSET 40 // this sets offset to allow for differences between the two DC traction motors
#define COLL_DIST 30 // sets distance at which robot stops and reverses to 30cm
#define TURN_DIST COLL_DIST+20 // sets distance at which robot veers away from object
NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE); // sets up sensor library to use the correct pins to measure distance.

AF_DCMotor leftMotor1(1, MOTOR12_1KHZ); // create motor #1 using M1 output on Motor Drive Shield, set to 1kHz PWM frequency
AF_DCMotor leftMotor2(2, MOTOR12_1KHZ); // create motor #2, using M2 output, set to 1kHz PWM frequency
AF_DCMotor rightMotor1(3, MOTOR34_1KHZ);// create motor #3, using M3 output, set to 1kHz PWM frequency
AF_DCMotor rightMotor2(4, MOTOR34_1KHZ);// create motor #4, using M4 output, set to 1kHz PWM frequency
Servo myservo;  // create servo object to control a servo

int leftDistance, rightDistance; //distances on either side
int curDist = 0;
String motorSet = "";
int speedSet = 0;
```

obstacle_avoiding

```
//-------------------------------------------- SETUP LOOP ----------------------------------------------
void setup() {
  myservo.attach(10);  // attaches the servo on pin 10 (SERVO_1 on the Motor Drive Shield to the serv
  myservo.write(90); // tells the servo to position at 90-degrees ie. facing forward.
  delay(1000); // delay for one seconds
 }

void loop() {
  myservo.write(90);  // move eyes forward
  delay(90);
  curDist = readPing();   // read distance
  if (curDist < COLL_DIST) {changePath();}  // if forward is blocked change direction
  moveForward();  // move forward
  delay(500);
 }

void changePath() {
  moveStop();   // stop forward movement
  myservo.write(36);  // check distance to the right
    delay(500);
    rightDistance = readPing(); //set right distance
    delay(500);
    myservo.write(144);  // check distace to the left
    delay(700);
    leftDistance = readPing(); //set left distance
    delay(500);
    myservo.write(90); //return to center
    delay(100);
    compareDistance();
  }
```

obstacle_avoiding §

```
void compareDistance()   // find the longest distance
{
  if (leftDistance>rightDistance) //if left is less obstructed
  {
    turnLeft();
  }
  else if (rightDistance>leftDistance) //if right is less obstructed
  {
    turnRight();
  }
   else //if they are equally obstructed
  {
    turnAround();
  }
}



int readPing() { // read the ultrasonic sensor distance
  delay(70);
  unsigned int uS = sonar.ping();
  int cm = uS/US_ROUNDTRIP_CM;
  return cm;
}
void moveStop() {
  leftMotor1.run(RELEASE);
  leftMotor2.run(RELEASE);
  rightMotor1.run(RELEASE);
  rightMotor2.run(RELEASE);}  // stop the motors.
```

obstacle_avoiding §

```
void moveForward() {
    motorSet = "FORWARD";
    leftMotor1.run(FORWARD);      // turn it on going forward
    leftMotor2.run(FORWARD);      // turn it on going forward
    rightMotor1.run(FORWARD);     // turn it on going forward
    rightMotor2.run(FORWARD);     // turn it on going forward
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the speed up to avoid loading down the batteries too quickly
  {
    leftMotor1.setSpeed(speedSet);
    leftMotor2.setSpeed(speedSet);
    rightMotor1.setSpeed(speedSet);
    rightMotor2.setSpeed(speedSet);
    delay(5);
  }
}
void moveBackward() {
    motorSet = "BACKWARD";
    leftMotor1.run(BACKWARD);     // turn it on going backward
    leftMotor2.run(BACKWARD);     // turn it on going backward
    rightMotor1.run(BACKWARD);    // turn it on going backward
    rightMotor2.run(BACKWARD);    // turn it on going backward
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the speed up to avoid loading down the batteries too quickly
  {
    leftMotor1.setSpeed(speedSet);
    leftMotor2.setSpeed(speedSet);
    rightMotor1.setSpeed(speedSet);
    rightMotor2.setSpeed(speedSet);
    delay(5);
  }
}
```

obstacle_avoiding §

```
void turnRight() {
  motorSet = "RIGHT";
  leftMotor1.run(FORWARD);      // turn motor 1 forward
  leftMotor2.run(FORWARD);      // turn motor 2 forward
  rightMotor1.run(BACKWARD);    // turn motor 3 backward
  rightMotor2.run(BACKWARD);    // turn motor 4 backward
  rightMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
  rightMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
  delay(1500); // run motors this way for 1500
  motorSet = "FORWARD";
  leftMotor1.run(FORWARD);      // set both motors back to forward
  leftMotor2.run(FORWARD);
  rightMotor1.run(FORWARD);
  rightMotor2.run(FORWARD);
}

void turnLeft() {
  motorSet = "LEFT";
  leftMotor1.run(BACKWARD);      // turn motor 1 backward
  leftMotor2.run(BACKWARD);      // turn motor 2 backward
  leftMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
  leftMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
  rightMotor1.run(FORWARD);     // turn motor 3 forward
  rightMotor2.run(FORWARD);     // turn motor 4 forward
  delay(1500); // run motors this way for 1500
  motorSet = "FORWARD";
  leftMotor1.run(FORWARD);      // turn it on going forward
  leftMotor2.run(FORWARD);      // turn it on going forward
  rightMotor1.run(FORWARD);     // turn it on going forward
  rightMotor2.run(FORWARD);     // turn it on going forward
}

 void turnAround() {
   motorSet = "RIGHT";
   leftMotor1.run(FORWARD);      // turn motor 1 forward
   leftMotor2.run(FORWARD);      // turn motor 2 forward
   rightMotor1.run(BACKWARD);    // turn motor 3 backward
   rightMotor2.run(BACKWARD);    // turn motor 4 backward
   rightMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
   rightMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
   delay(1700); // run motors this way for 1700
   motorSet = "FORWARD";
   leftMotor1.run(FORWARD);      // set both motors back to forward
   leftMotor2.run(FORWARD);
   rightMotor1.run(FORWARD);
   rightMotor2.run(FORWARD);
 }
```

3.2 WORKING MECHANISM

The robot will be designed to detect the presence of any object within the specified threshold distance. If any object is found within this distance, it is designated as an obstacle and the robot will turn away from it. The sensor emit an ultrasonic pulse every 300 ms which echoes from the neighboring objects. Using time difference between the input and echo, the Arduino calculates the distance to the obstacle from which the echo is coming by using the constant speed of sound 340 m/s. When one of the sensors detects obstacle within the threshold distance, the robot changes its direction.

Infrared sensors detect the object's distance with infrared radiation. When the beam detects an object, the light beam returns to the receiver with an angle after reflection. The method of triangulation is as. IR motion sensor, which detect the difference in temperature, thermal radiation, human body or an animal, operates with the radiation the hotter the detected object, there will be more emission occurs in IR sensor. IR sensors also act as a transducer since they use infra-red signal as the input and convert it to analogue electrical output signal

4. CONCLUSION AND FUTURE WORK

4.1 RESULT

Using this obstacle avoiding car, static and moving objects are detected. If there is any obstacle in the path then the vehicle will automatically make a sound into their ears by listening to which the blind person can automatically change his path to move. Using this tool they need not use their conventional stick to walk around. They will also get to know which path has the obstacle as there are four different sensors for four different paths and four different sensors produce different sounds for a blind person to understand. The sensors operate for the detection of fast moving objects as they calculate the speed of the object moving and detect if it is fast or not.

CONCLUSION

This “Obstacle Avoidance Robot Car” project is proved using the Ultrasonic sensor for detecting objects, Motor Driver Shield for driving the DC motors, DC motors for movement of the wheels of the robot with the help of the Arduino Microcontroller. The factors which affect the accuracy of the designed robot include the environment the robot was tested and the number of present obstacles in the test space. These factors mainly affected the sensor which means that the accuracy of the robot is dependent on the sensor.

INFERENCE

The objective of this task is to make an independent robot that wisely distinguishes the obstacle in its way and explores as indicated by the activities that we set for it. So what this framework gives is an option in contrast to the current framework by supplanting gifted work with automated apparatus, which thusly can deal with more patients in less time with better exactness and a lower for each capita cost.

4.2 FUTURE WORK

This paper is all about Obstacle Avoidance Robot using Arduino which avoids obstacles which it encounters. In future this project can be enhance by connecting Bluetooth module and a camera so that the user can see the detected obstacle on his screen by sitting at just one place. Further improvement can be achieved by adding sensors on the left and right side of the robot. Besides that, computer vision with camera features can be implemented for monitoring applications. For further improvement, to implement an obstacle avoidance in aerospace, well-suited sensors should be used to gather the accurate information about the environment and obstacles. The laser based (LIDAR) sensor system is robust especially in off-road outdoor environments.LIDAR sensor is considered as an effective solution to the problem of obstacle detection and recognition. However, the obstacle avoidance poses challenges to the image processing using LIDAR sensor.

Video link:

https://drive.google.com/file/d/1DloqJlVBr_pLev12wfmpjTrCeC7wN40w/view?usp=sharing

REFERENCES

LIST OF PUBLICATIONS

1. Amir attar, Aadilansari, abhishekdesai, shahid khan, dip ashrisonawale "line follower and obstacle avoidance bot using arduino" International Journal of Advanced Computational Engineering and Networking, vol. 2, pp. 740- 741.April2017.

2. R. Polvara1, S. Sharma, R. Sutton, J. Wan, A. Manning, "Obstacles avoidance approaches for autonomous navigation of unmanned surface vehicles". J. Navig. **38**, 1–16 (2017).

3. Vaghela Ankit1, Patel Jigar2, Vaghela Savan3 "Obstacle Avoidance Robotic Vehicle Using Ultrasonic Sensor, Android And Bluetooth For Obstacle Detection" International Research Journal of Engineering and Technology (IRJET), vol. A247, pp. 29-32, 2005.

COST

Arduino UNO - Rs.509

Chassis - Rs.705

Servo - Rs.199

Motor driver shield - Rs.404

Jumper wires - Rs.98

Ultrasonic Sensor - Rs.74

Mounted Holder & Battery - Rs.75

BO Motor L-type - Rs.220 (55x4)

Total - Rs. 2284

5. PHOTO GRAPH OF THE PROJECT ALONG WITH THE TEAM MEMBERS

BIODATA

Name : Kavya Paliwal

Mobile Number : 9450358202

E-mail : kavya.paliwal2020@vitstudent.ac.in

Permanent Address: Saket Nagar, Lanka, Varanasi, Uttar Pradesh

Name : Aniket Kumar Paul

Mobile Number : 9093250238

E-mail : aniketkumarpaul1@gmail.com

Permanent Address: Gopimohan more, main road, Jaigaon, Alipurduar, WB

Name : Savio Sajan Moloparambil

Mobile Number : 9061612080

E-mail : saviosajan202@gmail.com

Permanent Address:Thodupuzha-Moolamattom road, Muttom, Thodupuzha, Idukki