OBSTACLE AVOIDING ROBOT

MENTOR :

Dr.MAGESHWARAN.M, HOD of Mechatronics

ARUNKUMAR.S, Assistant Professor

BY,

BARANIDHARAN S

SANJEEV U

JEEVAN PRABAKARAN S

JUBINNATH R

OBSTACLE AVOIDING ROBOT

- An Obstacle Avoidance Robot is an intelligent robot, which can automatically sense and overcome obstacles on its path. It contains of a Microcontroller to process the data, and Ultrasonic sensors to detect the obstacles on its path.
- Obstacle avoidance is one of the most important aspects of mobile robotics. Without it robot movement would be very restrictive and fragile (easily damaged or broken).
- It ensures that the robot does n’t have to stop in front of an obstacle which allows robot to navigate smoothly in an unknown environment, avoiding collisions.

COMPONENTS REQUIRED

HARDWARE COMPONENTS

01 ARDUINO UNO

02 HC-SR04 ULTRASONIC SENSOR

03 L293D MOTOR DRIVER MODULE

04 9V DC GEAR MOTORS

05 SERVO MOTOR

06 BATTERY

07 CHASSIS

08 JUMPER WIRES

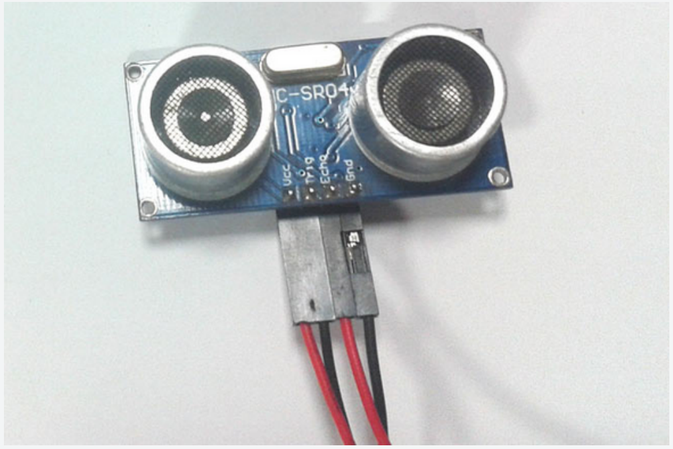
SOFTWARE COMPONENTS

01 ARDUINO IDE

ARDUINO NANO

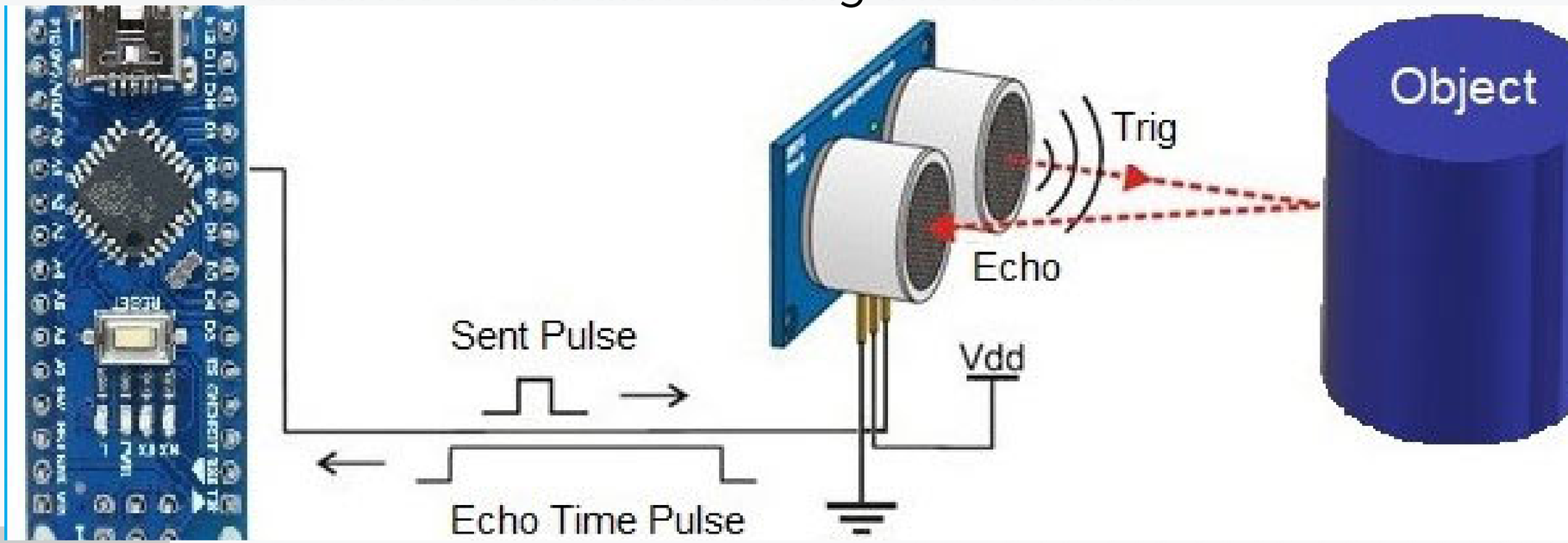
Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays,buttons, motors, switches, LEDs, servos, and motors as an output.

ULTRASONIC SENSOR (HC-SR04)

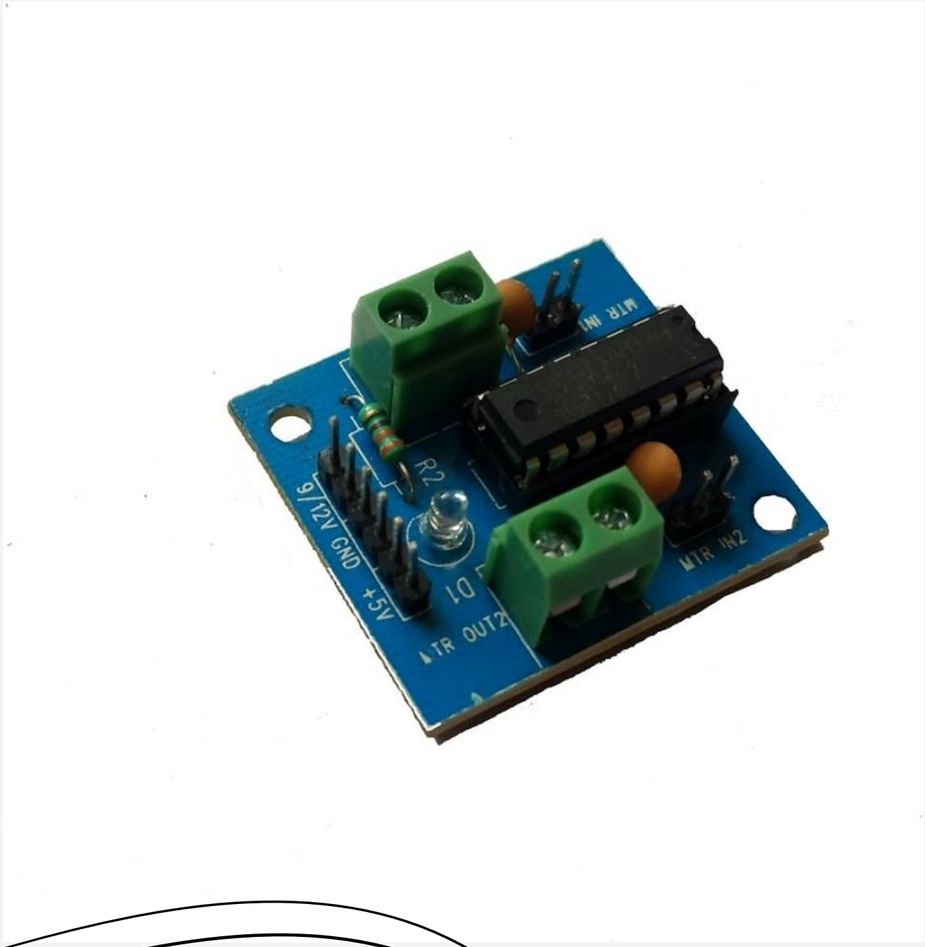


- This sensor will have important role in detecting obstacle.
- The basic principle behind the working of ultrasonic sensor is to note down the time taken by sensor to transmit ultrasonic beams
- Receiving the ultrasonic beams after hitting the surface. Then further the distance is calculated using the formula.

Distance= (Time x Speed of Sound in Air)/2
d= (343 m/s)/2



L293D-MOTOR DRIVER



THE L293D IS A 16-PIN MOTOR DRIVER IC WHICH CAN CONTROL A SET OF TWO DC MOTORS SIMULTANEOUSLY IN ANY DIRECTION. THE L293D IS DESIGNED TO PROVIDE BIDIRECTIONAL DRIVE CURRENTS OF UP TO 600 MA (PER CHANNEL) AT VOLTAGES FROM 4.5 V TO 36 V (AT PIN 8!). YOU CAN USE IT TO CONTROL SMALL DC MOTORS - TOY MOTORS.

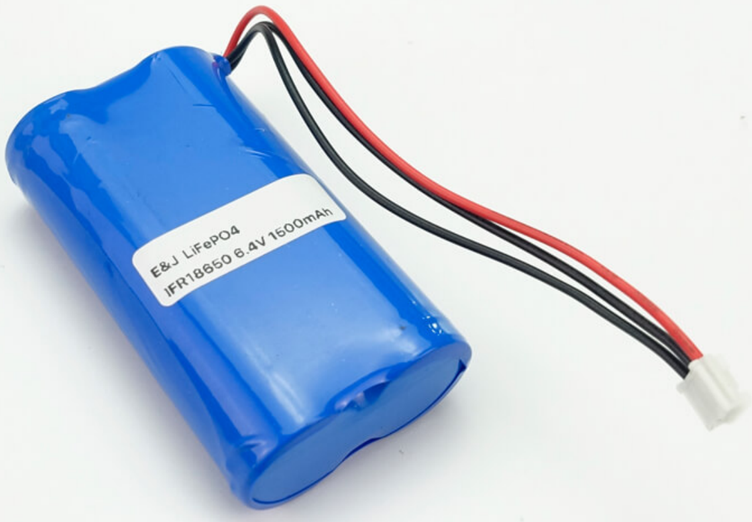
GEAR MOTOR

GEAR motors are electrical motors that convert direct current (DC) electrical energy into mechanical energy. DC motors can be subcategorized as brushed or brushless

SERVO MOTOR

Servo motor - It turns left and right to the ultrasonic sensor or we can say that it helps the robot to sense obstacles in the left and right direction. Gear motor - It helps the robot to move in all directions (forward, backward, left, right). Wheels - It is connected to a gear motor

Li-ion battery



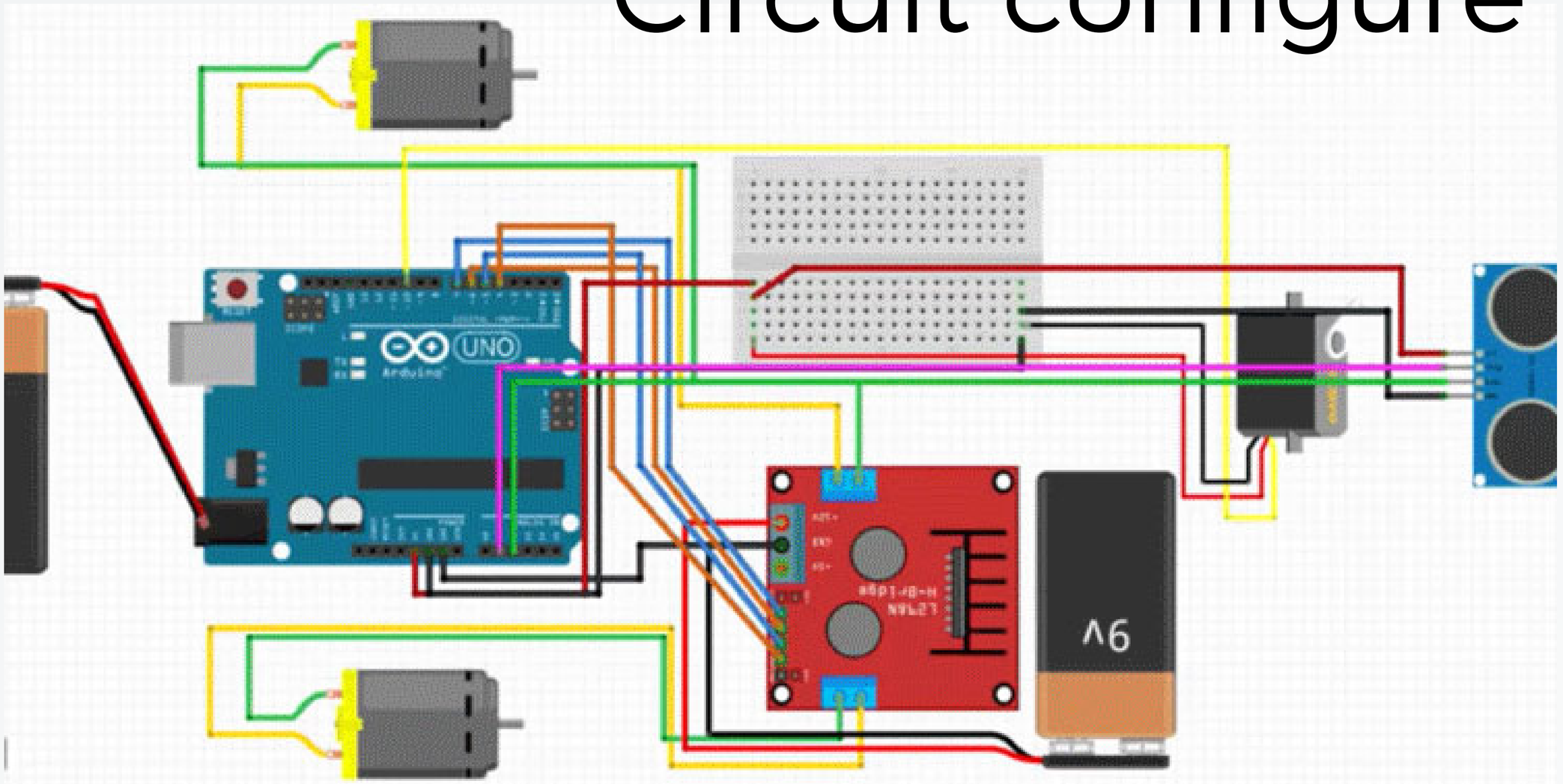
A lithium-ion battery is a type of rechargeable battery that is charged and discharged by lithium ions moving between the negative (anode) and positive (cathode) electrodes. take 2 3.7v battery and connect in serial it is converted into 7.4

CHASSIS AND JUMPERS

A chassis is the load-bearing framework of a manufactured object, which structurally supports the object in its construction and function

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

Circuit configure



WORKING PROCESS

- The robot reads data from its sensors, such as ultrasonic or infrared sensors.
- It processes the sensor data and calculates the distance to the nearest obstacle.
- Based on the sensor data and obstacle distance, the robot's software decides on its next move, such as turning, stopping, or continuing forward.
- The software sends motor control commands to the motor drivers, which drive the wheels accordingly.
- The robot repeats this process continuously, effectively navigating around obstacles and avoiding collisions.

CODES:

```
#include <Servo.h>            //Servo motor library. This is standard library
#include <NewPing.h>          //Ultrasonic sensor function library. You must install
                                this library

                                //our L298N control pins
const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 4;
const int RightMotorBackward = 5;
 //sensor pins
#define trig_pin A1 //analog input 1
#define echo_pin A2 //analog input 2
#define maximum_distance 200
boolean goesForward = false;
int distance = 100;

 NewPing sonar(trig_pin, echo_pin, maximum_distance);          //sensor function
 Servo servo_motor;                                            //our servo name
```

```
void setup(){

 pinMode(RightMotorForward, OUTPUT);
 pinMode(LeftMotorForward, OUTPUT);
 pinMode(LeftMotorBackward, OUTPUT);
 pinMode(RightMotorBackward, OUTPUT);

servo_motor.attach(10);            //our servo pin

 servo_motor.write(115);
 delay(2000);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
  }
```

```
void loop(){
 int distanceRight = 0;
 int distanceLeft = 0;
 delay(50);

 if (distance <= 20){
    moveStop();
    delay(300);
    moveBackward();
    delay(400);
    moveStop();
    delay(300);
    distanceRight = lookRight();
    delay(300);
    distanceLeft = lookLeft();
    delay(300);

    if (distance >= distanceLeft){
      turnRight();
      moveStop();
    }
    else{
      turnLeft();
      moveStop();
    }
 }
     else{
    moveForward();
 }
```

```
distance = readPing();
}

 int lookRight(){
 servo_motor.write(50);
 delay(500);
 int distance = readPing();
 delay(100);
 servo_motor.write(115);
 return distance;
 }
 int lookLeft(){
 servo_motor.write(170);
 delay(500);
 int distance = readPing();
 delay(100);
 servo_motor.write(115);
 return distance;
 delay(100);
}
int readPing(){
delay(70);
 int cm = sonar.ping_cm();
 if (cm==0){
 cm=250;
 }
return cm;
}
```

```
void moveStop(){
digitalWrite(RightMotorForward, LOW);
 digitalWrite(LeftMotorForward, LOW);
 digitalWrite(RightMotorBackward, LOW);
digitalWrite(LeftMotorBackward, LOW);}
 void moveForward(){
 if(!goesForward){
 goesForward=true;

digitalWrite(LeftMotorForward, HIGH);
digitalWrite(RightMotorForward, HIGH);
 digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  }
  }
void moveBackward(){
 goesForward=false;

digitalWrite(LeftMotorBackward, HIGH);
digitalWrite(RightMotorBackward, HIGH);
digitalWrite(LeftMotorForward, LOW);
 digitalWrite(RightMotorForward, LOW);
 }
```

```
void turnRight(){
digitalWrite(LeftMotorForward, HIGH);
digitalWrite(RightMotorBackward, HIGH);
digitalWrite(LeftMotorBackward, LOW);
digitalWrite(RightMotorForward, LOW);
delay(500);
digitalWrite(LeftMotorForward, HIGH);
digitalWrite(RightMotorForward, HIGH);
digitalWrite(LeftMotorBackward, LOW);
digitalWrite(RightMotorBackward, LOW);
        }

void turnLeft(){

digitalWrite(LeftMotorBackward, HIGH);
digitalWrite(RightMotorForward, HIGH);
digitalWrite(LeftMotorForward, LOW);
digitalWrite(RightMotorBackward, LOW);
delay(500);

digitalWrite(LeftMotorForward, HIGH);
digitalWrite(RightMotorForward, HIGH);
digitalWrite(LeftMotorBackward, LOW);
digitalWrite(RightMotorBackward, LOW);

            }
```

ADVANTAGES AND USES:

1.An Obstacle Avoidance Robot is an intelligent robot, which can automatically sense and overcome obstacles on its path.

2.It contains of a Microcontroller to process the data, and Ultrasonic sensors to detect the obstacles on its path.

3.Obstacle avoidance is one of the most important aspects of mobile robotics

APPLICATION:

1.Obstacle avoiding robots can be used in almost all mobile robot navigation systems.

2.They can be used for household work like automatic vacuum cleaning

3.They can also be used in dangerous environments , where human penetration could be fatal.

THANKING YOU.....

https://github.com/Baranidharan333/Obsticle-Avoidance.git