A Synopsis Report On

OBSTACLE AVOIDING ROBOT USING ULTRA SONIC SENSORS.

SUBMITTED BY:

P.REVANTH	U101115FCS265	S5
BOYINA SANTOSH	U101115FCS 140	S6
N.VISHNUVARDHAN	U101115FCS202	S1
I.LOHIYA	U101115FCS100	S4

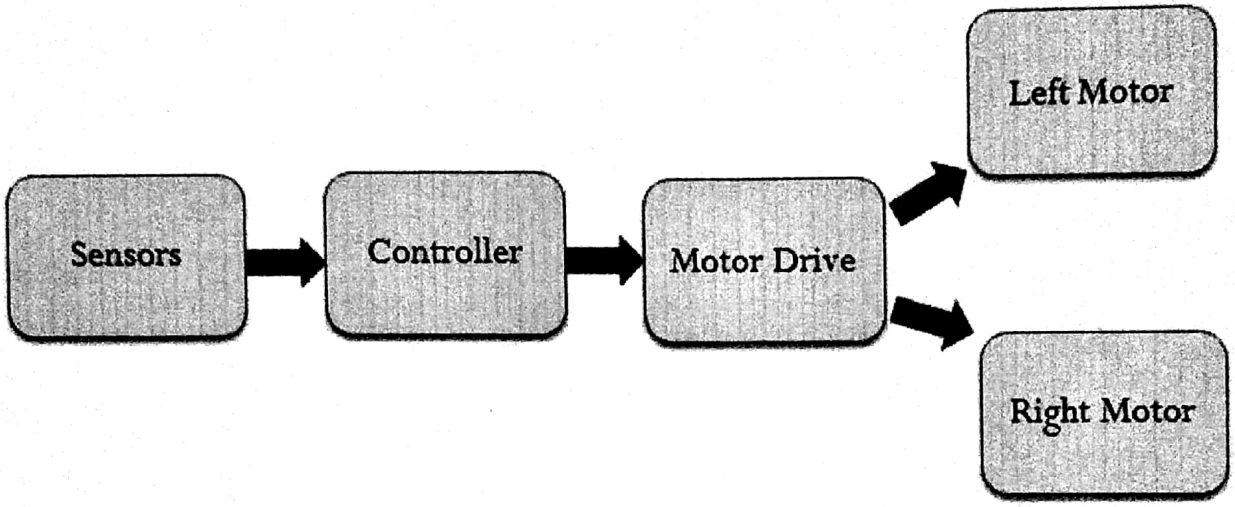
ABSTRACT

****

Our project OBSTACLE AVOIDING ROBOT is one which can avoid an obstacle by using ultrasonic sensor and navigate in its own path. With a arduino attached to the robot you can play fun within a short period of time. One such is what we are going to discuss here. This project can teach you how a sensor can be used to process some data.

An Ultrasonic sensor generate high frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object. A ultrasonic module has been used here for the demo purpose. It has a power, ground and two data pins (Trigger, Echo).

Block Diagram



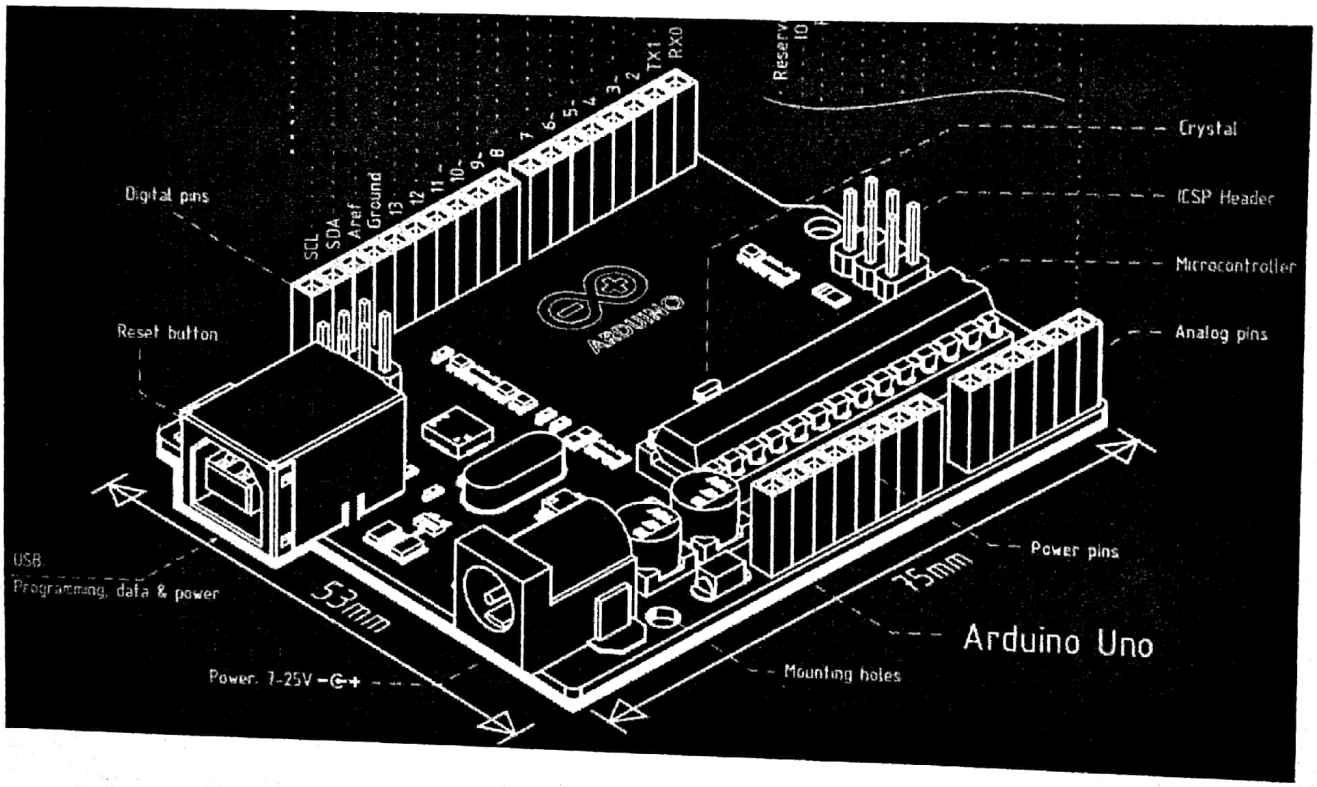
The aim of this project is to implement an obstacle avoiding robot using ultrasonic sensor and Arduino. Using an external trigger signal, the Trig pin on ultrasonic sensor is made logic high for at least 10µs. A sonic burst from the transmitter module is sent. This consists of 8 pulses of 40KHz.

The signals return back after hitting a surface and the receiver detects this signal. The Echo pin is high from the time of sending the signal and receiving it.

ARDUINO UNO:

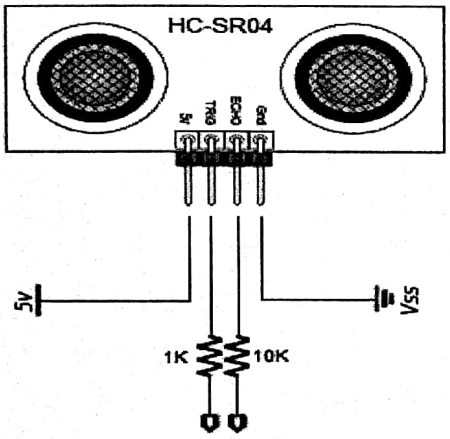
Arduino Uno is an ATmega 328p Microcontroller based prototyping board. It is an open source electronic prototyping platform that can be used with various sensors

Arduino Uno has 14 digital I/O pins out of which 6 pins are used in this project.



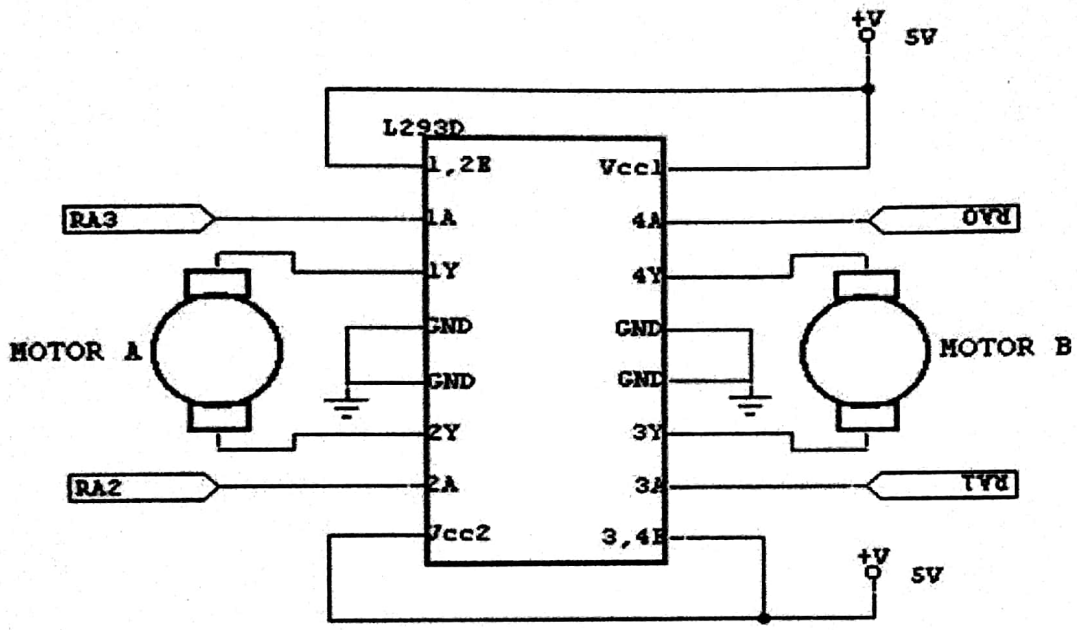
ULTRASONIC SENSOR(HC-SR04):

It is an Ultrasonic Range Finder Sensor. It is a non-contact based distance measurement system and can measure distance of 2cm to 4m.



MOTOR DRIVER IC:

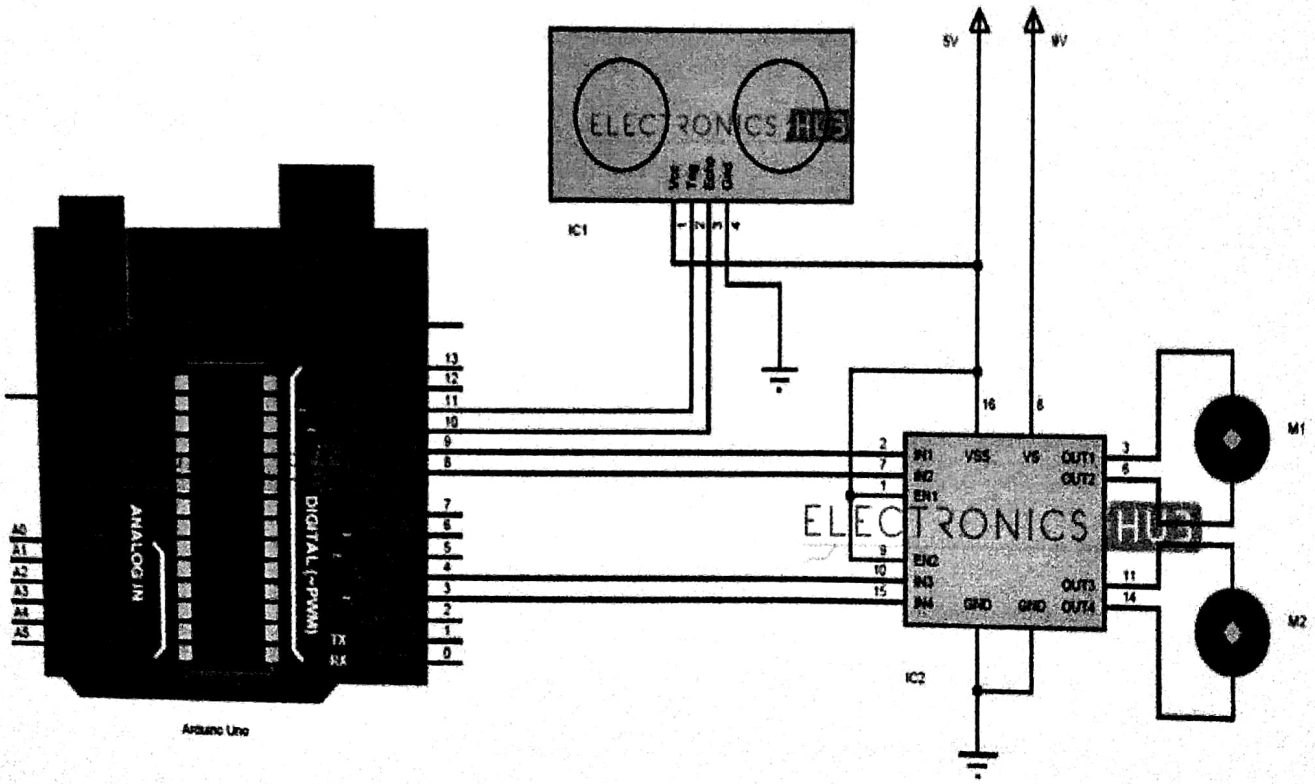
L293D is a typical motor driver or motor driver ic which allows DC motor to drive on either direction. L293D is a 16-pin **IC** which can control a set of two DC motor simultaneously in any direction. It means that you can control two DC motor with a single L293D **IC**. Dual H-bridge motor driver integrated circuit.



It works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. As you know voltage need to change its direction for being able to rotate the motor in clockwise or anticlockwise direction, Hence H-bridge IC are ideal for driving a DC motor.

In a single L293D chip there are two h-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors. Given below is the pin diagram of a L293D motor controller.

CIRCUIT DIAGRAM:



WORKING:

The basic principle behind the working of ultrasonic sensor is as follows:

ARDUINO CODE

```
#include<AF motor.h>
#include<new ping.h>
#include<servo.h>
#define TRIG PIN A4
#define ECHO PIN A5
#define MAX DISTANCE 200
#define MAX SPEED 190// 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 motor3(3, MOTOR12_64KHZ);

Servo myservo;

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

void setup() {

  myservo.attach(9);
  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;
 }

 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);
  motor3.run(RELEASE);
  }

  void moveForward() {

   if(!goesForward)
    {
     goesForward=true;
     motor1.run(FORWARD);
     motor3.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);
      motor3.setSpeed(speedSet+MAX_SPEED_OFFSET);
      delay(5);
     }
     }
    }

     void moveBackward() {
       goesForward=false;
       motor1.run(BACKWARD);
       motor3.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);
   motor3.setSpeed(speedSet+MAX_SPEED_OFFSET);
   delay(5);
  }
}

void turnRight() {
  motor1.run(FORWARD);
  motor3.run(BACKWARD);
  delay(300);
  motor1.run(FORWARD);
  motor3.run(FORWARD);
 }

 void turnLeft() {
   motor1.run(BACKWARD);
   motor3.run(FORWARD);
   delay(300);
   motor1.run(FORWARD);
   motor3 .run(FORWARD);
  }
```