OBSTACLE AVOIDING ROBO CAR

MINI PROJECT REPORT

Submitted by

INISH RAJ B (202009017)

BHARATH D (202009007)

in

19AD752 – INTELLIGENT SYSTEMS FOR IOT LABORATORY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

MEPCO SCHLENK ENGINEERING COLLEGE

SIVAKASI

OCTOBER 2023

MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI

AUTONOMOUS

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

BONAFIDE CERTIFICATE

This is to certify that it is the bonafide work of "**Inish Raj.B (Reg.No.:202009017)** and **Bharath.D (Reg.No.:202009007)"** for the mini project titled **"OBSTACLE AVOIDING ROBO CAR"** in 19AD752 – Intelligent Systems for IoT Laboratory during the seventh semester, July 2023 – October 2023 under my supervision.

SIGNATURE
Dr.A.Shenbagarajan,
Associate Professor,
Artificial Intelligence and Data Science,
Mepco Schlenk Engg. College, Sivakasi.

SIGNATURE
Dr. J. Angela Jennifa Sujana,
Professor & Head of the Department,
Artificial Intelligence and Data Science,
Mepco Schlenk Engg. College, Sivakasi.

ABSTRACT

An obstacle avoiding robot is an intelligent device that senses any object in front of it and avoids it by turning in a different direction. In this project we are using Arduino to control an ultrasonic sensor, four DC motors and a servo motor. The "Obstacle Avoiding Robo Car" project addresses the growing need for autonomous navigation systems in various industries and applications. With the advent of Industry 4.0 and the Internet of Things (IoT), there is an increasing demand for intelligent robotic systems capable of safely and efficiently maneuvering through complex environments. This project leverages cutting-edge technologies and utilizes an Arduino-based platform to create a smart, obstacle-avoiding robotic car. The primary objective of this project is to design and develop an autonomous robotic vehicle that can navigate autonomously while avoiding obstacles in real-time. The project utilizes ultrasonic sensors to detect obstacles in the vehicle's path and employs an Arduino microcontroller to process sensor data and make rapid decisions for steering the car away from obstacles. The system is designed to be cost-effective, making it accessible for educational purposes, prototyping, and practical applications. Key components of the project include ultrasonic sensors, Arduino microcontroller, motor drivers, and a chassis. The vehicle's intelligent decision-making capabilities are made possible through programming, where the Arduino processes data from sensors and controls the motors to maneuver the car. The project has the potential to find applications in various fields, including robotics education, automation in industries, and even in the development of self-driving cars.

CONTENTS

1. INTRODUCTION

1.1. Obstacle Avoiding Robot using Arduino

In the realm of robotics and automation, the development of autonomous navigation systems has witnessed remarkable advancements in recent years. One such innovative project is the "Obstacle Avoiding Robot using Arduino," which serves as a testament to the intersection of electronics, programming, and robotics. This project addresses the pressing need for intelligent, self-navigating robotic systems capable of autonomously avoiding obstacles in real-world scenarios.

The project leverages the versatility and accessibility of Arduino, a popular open-source electronics platform, to create a cost-effective yet highly efficient robotic solution. The fundamental objective is to design and construct a robot that can maneuver through its environment while intelligently detecting and circumventing obstacles in its path. This autonomous navigation capability has far-reaching implications, from enhancing safety in industrial settings to inspiring the next generation of engineers and robotics enthusiasts.

The core of this project lies in the integration of ultrasonic sensors and an Arduino microcontroller. These sensors act as the robot's "eyes," constantly emitting and receiving ultrasonic waves to calculate distances to nearby objects. The Arduino, equipped with suitable algorithms, processes this sensor data in real-time, allowing the robot to make swift decisions about changing its direction to avoid collisions. The result is a responsive and adaptable robotic system that can operate in dynamic and unpredictable environments.

This project not only offers valuable insights into the principles of robotics and electronics but also has practical applications in fields such as surveillance, logistics, and automation. By navigating obstacles intelligently, it showcases the potential for robotics to enhance efficiency, safety, and precision in various industries. In this project, we delve into the intricacies of creating an "Obstacle Avoiding Robot using Arduino," providing detailed instructions and insights to empower enthusiasts, students, and professionals alike in their journey to explore and implement autonomous robotic navigation.

1.2. Objective

The Objective of our project is follows:

- **Autonomous Navigation:** Develop a robot that can navigate autonomously without human intervention, using sensor data and decision-making algorithms.

- **Obstacle Detection:** Implement ultrasonic sensors to detect obstacles in the robot's path accurately.

- **Real-time Decision Making:** Enable the robot to process sensor data in real-time and make instant decisions to avoid collisions with detected obstacles.

- **Arduino Integration:** Utilize the Arduino microcontroller platform for controlling and programming the robot's movements and actions.

- **Cost-effectiveness**: Create a cost-effective robotic solution to make it accessible for educational purposes and practical applications.

- **User-friendliness**: Design the project in a way that is easy for enthusiasts, students, and beginners to understand and replicate.

- **Robotic Learning**: Serve as an educational tool to help individuals learn about robotics, electronics, programming, and sensor integration.

- **Adaptability:** Ensure that the robot can operate in various environments and adapt to different types of obstacles.

- **Versatility:** Explore the potential for the robot's application in scenarios such as surveillance, logistics, and automation.

- **Safety Enhancement:** Highlight the project's potential to enhance safety in industries where autonomous robotic systems can reduce the risk of accidents and collisions.

2. DESIGN PROCESS

2.1. Block diagram of Obstacle Avoiding Robo Car

The basic block diagram of the obstacle avoiding car is shown in the figure. Mainly this block diagram consists of the following essential blocks.

- Arduino UNO
- Ultrasonic Sensor
- Motor Driver(L293D)

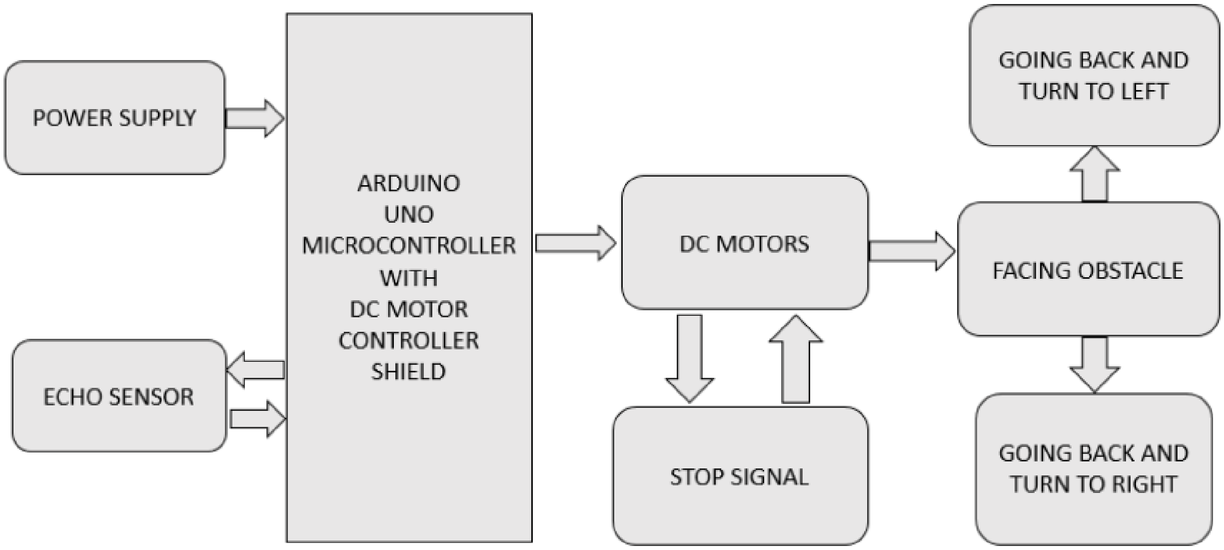

Figure 2.1.1 – Working Mechanism

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 and actuators. It is used for controlling all the operation and assign task to each device.

Ultrasonic Sensor:

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

Motor Driver:

It is a motor driver which can provide bi-directional drive current for two motors.

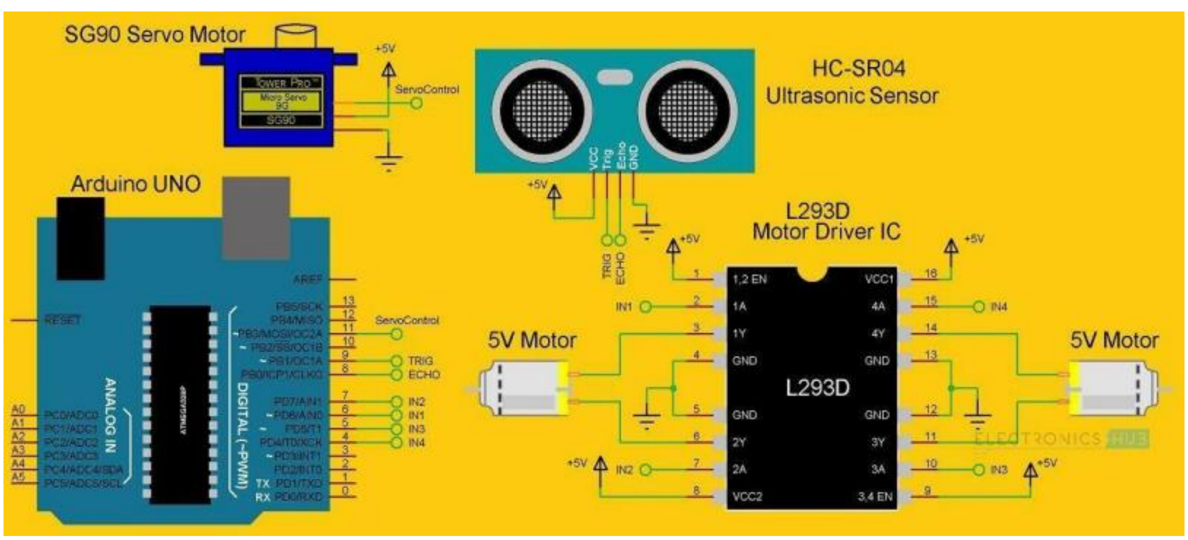

Figure 2.1.2 – Block Diagram

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. 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 it will rotate the motor M3 & M4 in forward direction and rotate the motor M1 & 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.

2.2. Components

Figure 2.2.1 – Components Used

2.2.1. Arduino UNO R3

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

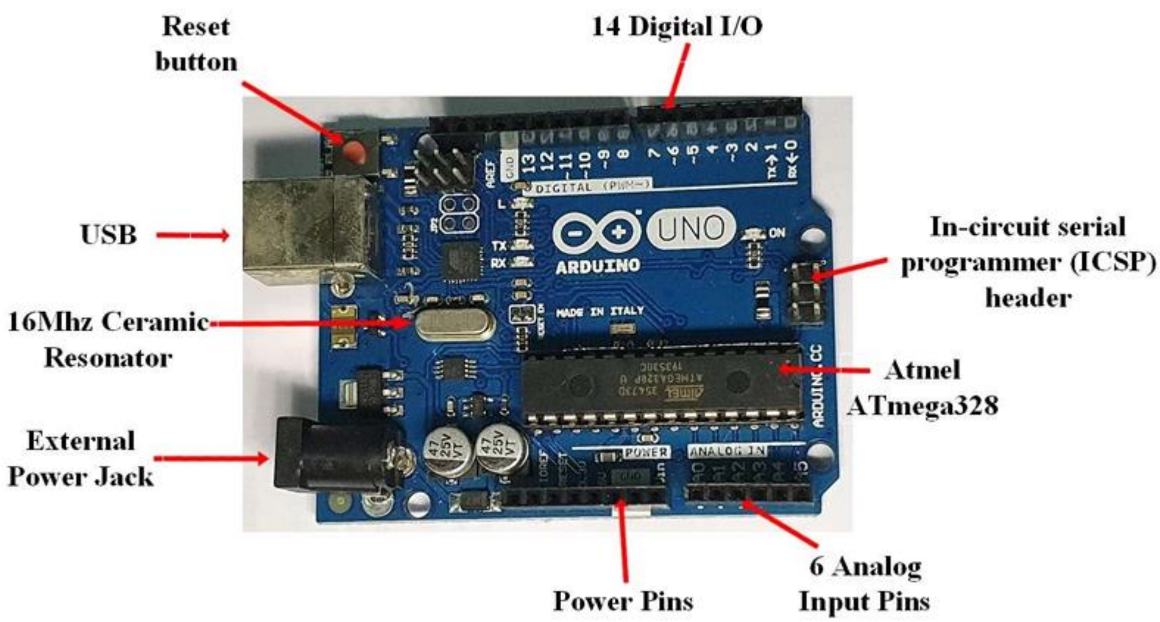

Figure 2.2.1.1 – Arduino Uno Board

Specifications:

- A 16 MHz clock.
- 32 KB of flash memory
- 13 digital pins and 6 analogue pins.
- ICSP connector to re-boot load your chip and for bypassing the USB port
- interfacing the Arduino directly as a serial device
- LED attached to digital pin 13 for and easy debugging of code.
- Reset button to reset the program on the chip.

Pin	Function
Vin	Voltage from External power jack
5V	5V output from on-board Voltage regulator chip
3.3V	3.3V output from on-board Voltage regulator chip
Gnd	3 pins for ground
IOREF	Tied to 5V, tells Arduino shields voltage level from which Arduino board operates
Reset	From RESET pin on MCU, tied to VCC through 10K resistor, pull to GND to reset

Table 2.2.1.1 – Arduino Uno Pin Specification

2.2.2. Ultrasonic Sensor

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

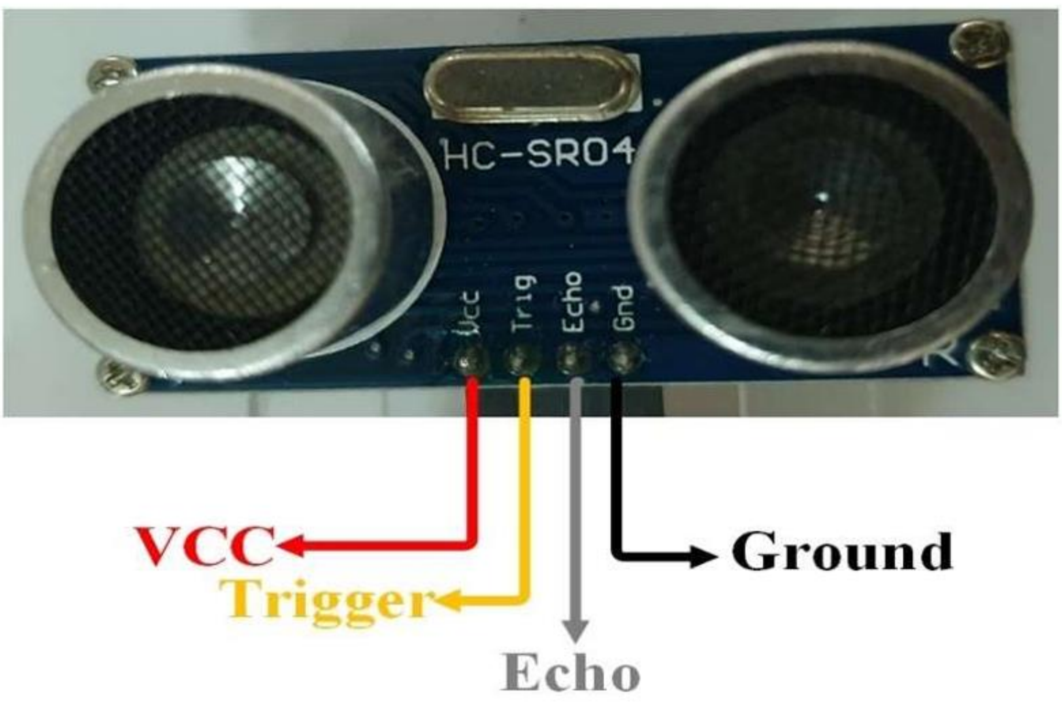

Figure 2.2.2.1 – Ultrasonic Sensor

It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance. The HC-SR04 Ultrasonic Module has 4 pins, Ground, VCC, Trig and Echo. The Ground and the VCC pins of the module needs to be connected to the Ground and the 5 volts pins on the Arduino Board respectively and the trig and echo pins to any Digital I/O pin on the Arduino Board. In order to generate the ultrasound you need to set the Trig on a High State for 10 µs. That will send out an 8 cycle sonic burst which will travel at the speed sound and it will be received in the Echo pin. The Echo pin will output the time in microseconds the sound wave traveled.

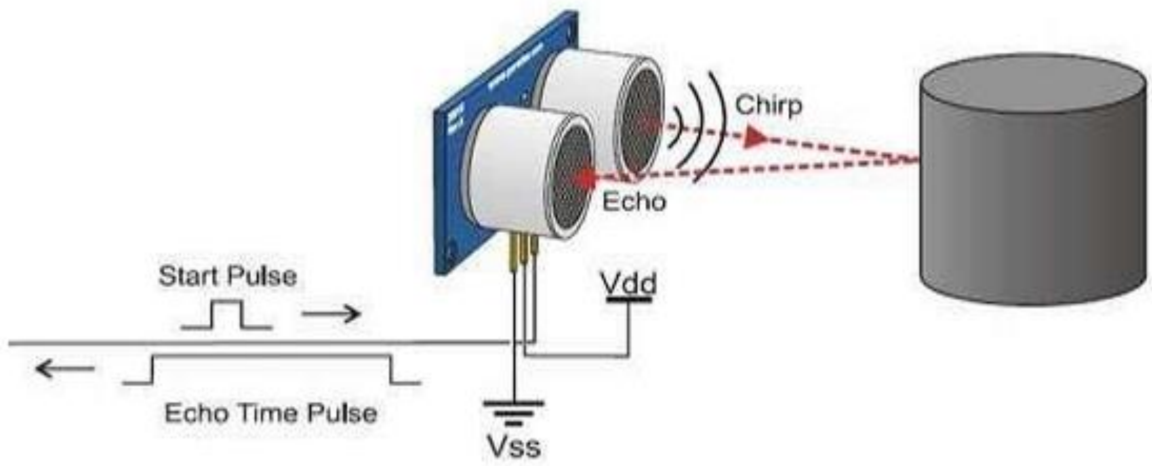

Figure 2.2.2.2 – Working of Ultrasonic Sensor

HC-SR04 Ultrasonic Sensor Features:

- It has a 5V DC power supply
- Its Quiescent current is less than 2mA
- Its working current is 15mA
- Its effectual angle is less than 15 degrees
- Its ranging distance is between 2cm – 400cm/1" – 13ft
- Its Resolution is 0.3cm
- Its measuring angle is 30degrees
- Its Trigger pulse width is 10uS
- Its dimension is 45mm x 20mm x 15mm

Pin Number	Pin Name	Description
1	VCC	The is used to power the sensor with 5V
2	Trigger	This is an Input pin which has to be kept high for 10us to initialize measurement by sending Ultrasound wave.
3	Echo	This is an output pin which goes high for a period of time equal to the time taken for the Ultrasonic wave to return back to the sensor.
4	Ground	This pin is connected to the ground of the system

Table 2.2.2.1 –Ultrasonic Sensor Pin Specification

2.2.3. Servo Motor

A servo motor is an electrical device that pushes or rotates objects with high precision. If there is need for an object to be rotated as a specific angle or distance, then a servo motor is used. A servo motor consists of a motor that uses servo mechanism. The two types of servo motors are the DC servo motors (DC powered) and the AC servo motor (AC powered) where the difference between them is the input power. A very high torque can be obtained from a small and lightweight servo motor which allows these servo motors to be used in applications like robots, toy cars, etc. The main reason why a servo motor is used is because of its high angle precision, i.e. after rotating, it will stop and wait for the next instruction to happen unlike a normal electric motor which rotates as long as it is being supplied power and stops rotating when power supply is turned off.

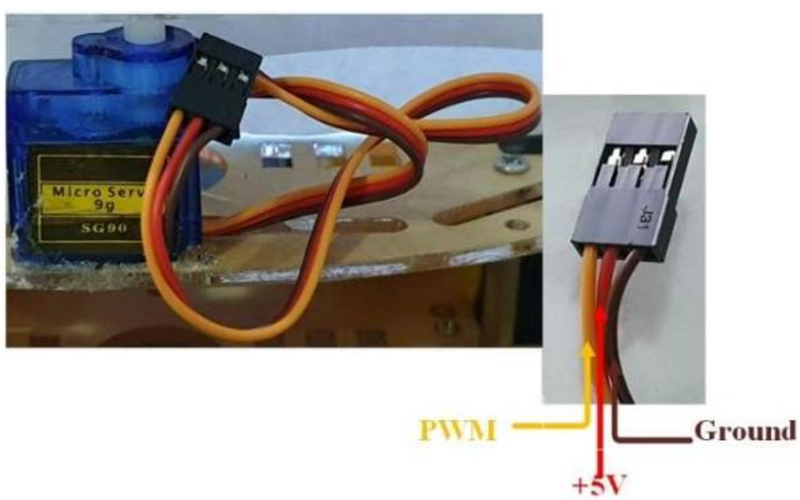

Figure 2.2.3.1 – Servo Motor

A servo motor includes motors (DC and AC), a potentiometer, gear assembly, and a controlling circuit. Firstly, gear assembly is used to reduce the Revolutions per Minute (RPM) and to increase the torque of the motor. At the initial position of the servo motor shaft, the potentiometer knob is in such a way that no electrical signal is generated at the potentiometer output. An electrical signal is passed to another input terminal of the error detector amplifier and the difference between these two signals is that one is coming from the potentiometer and the other is coming from another source. The both signals will be processed in a feedback mechanism and the output will be provided in terms of an error signal. This error is used as the motor input to cause it to start rotating. The motor shaft is connected to the potentiometer which causes the potentiometer to generate a signal as the motor rotates. As the angular position of the potentiometer changes, its feedback output also changes and after a while the potentiometer reaches a position that the potentiometer output and the external signal are the same. Once the outputs are the same, the amplifier will not output anything into the motor since there is no difference between the generated potentiometer signal and the external applied signal which in turn causes the motor to stop rotating. Servo motors are controlled by Pulse width Modulation (PWM) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. A servo motor can turn 90 degrees to either direction from its neutral position. The servo motor expects a pulse every 20 ms which determines how far the motor will turn. In the figure 4-11 below, a 1.5ms pulse will make the motor turn 90 degrees. If the pulse is shorter than 1.5ms, the shaft moves to 0 degrees and if it is longer than 1.5 ms, the shaft will turn to 180 degrees. Pulse with modulation simply means that the angle of rotation is dependent on the duration of pulse applied to the control PIN. Basically, servo motors are made up of a DC motor controlled by a potentiometer (Variable resistor) and some gears which convert the high-speed force into torque.

Advantages of using Servo Motors:

- Low Cost
- They have a wide range of sizes and torque ratings
- They are simple to control

2.2.4. Motor Driver

The L293D is a high voltage, high current Integrated circuit which is used to drive DC motors with a power supply of up to 36V. This chip is able to supple a maximum of 600mA per channel. This chip is also known as a type of H-Bridge as it enables a voltage to be applied across a load in either direction to an output, e.g. a motor.

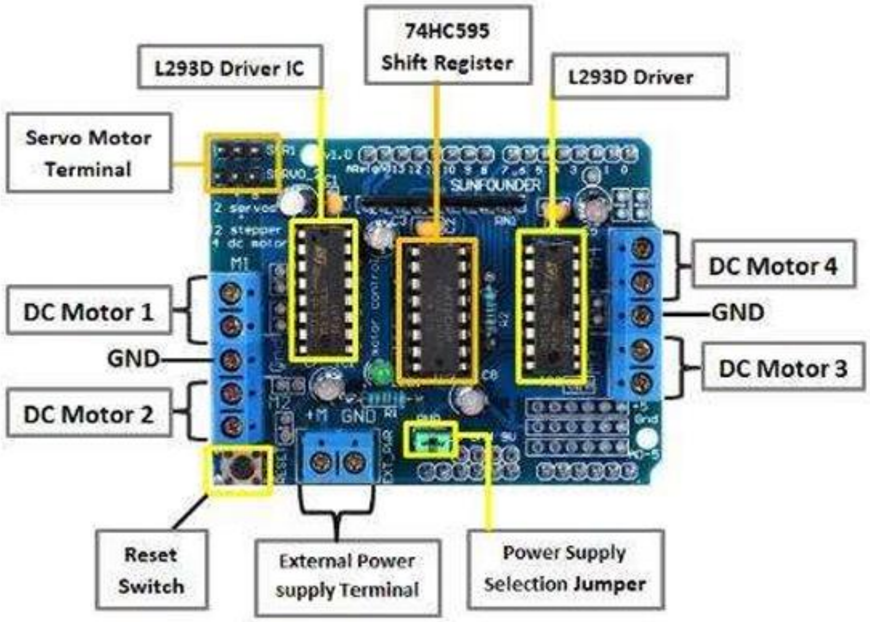

Figure 2.2.4.1 – Motor Driver Board

Specifications:

- It has to connections to allow for 5V servos to be connected to the Arduino's high-resolution
- dedicated timer.
- It has up to 4 bi-directional DC motors with individual 8-bit speed selection
- It has up to 2 stepper motors with single coil, double coil, interleaved or micro-stepping.
- It has 4 H-Bridges
- It has Pull down resistors to disable motors during power up
- The Arduino reset button is brought to the top
- It is compatible with Mega, UNO & Duemilanove
- Its dimensions are: 69mm x 53mm x 14.3mm

2.2.5. DC Motor

The obstacle detection and avoiding robot uses two 200 rpm and 12V DC geared motors. The motor used has a 6mm shaft diameter with internal holes. The internal holes are for easy mounting of the wheels by using screws. It is an easy to use low- cost motor for robotics application.

Figure 2.2.5.1 – DC Motor

An Electric DC motor is a machine which converts electric energy into mechanical energy. The working of DC motor is based on the principle that when a current- carrying conductor is placed in a magnetic field, it experiences a mechanical force. The direction of mechanical force is given by Fleming's Left-hand Rule and its magnitude is given by F = BIL Newton. DC motors are seldom used in ordinary applications because all electric supply companies furnish alternating current.

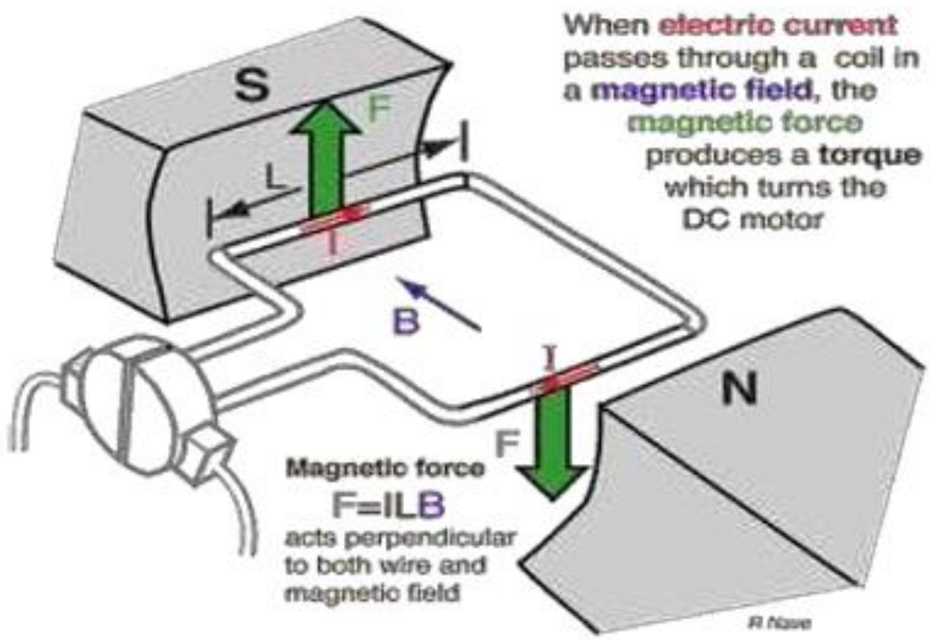

Figure 2.2.5.2 – Working of DC Motor

2.3. Circuit Diagram

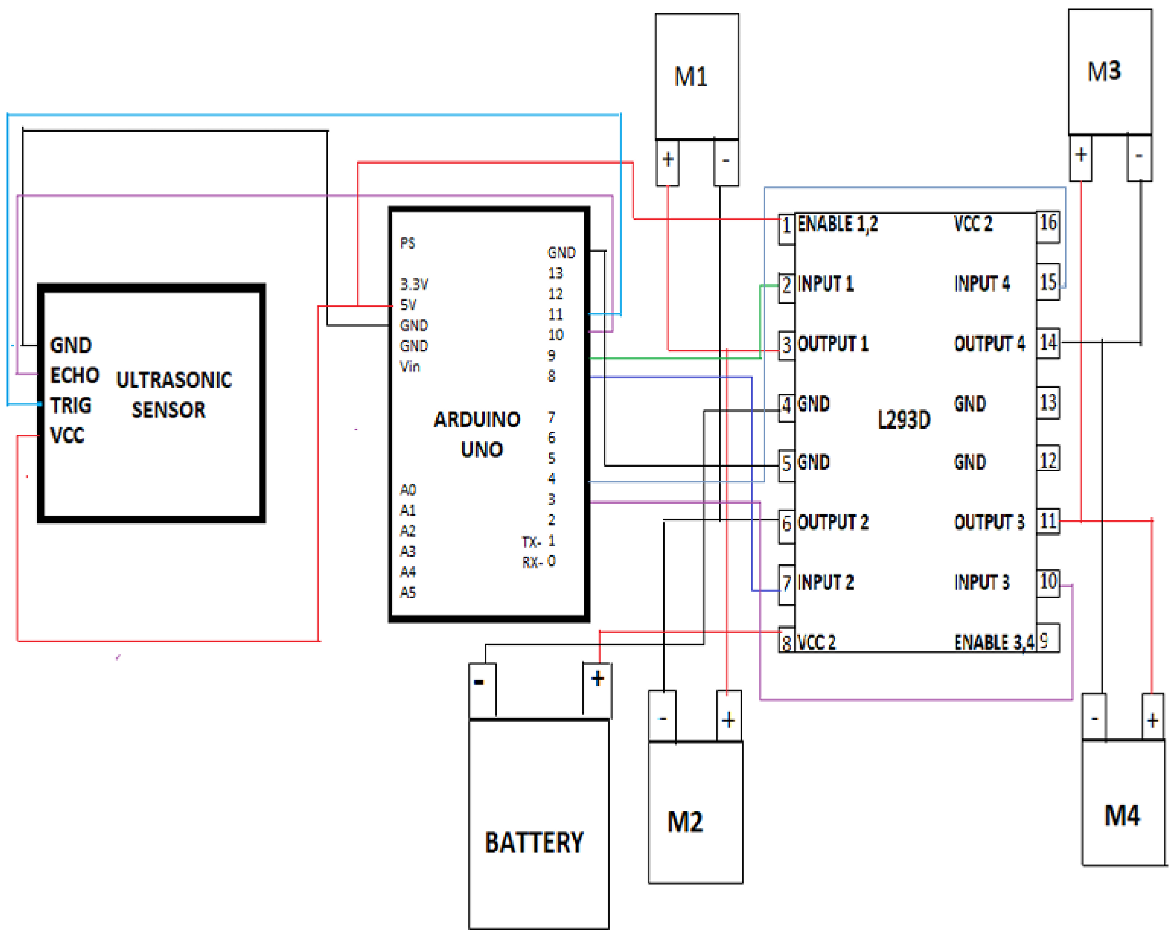

Figure 2.3.1 – Circuit Diagram

2.4. Flowchart

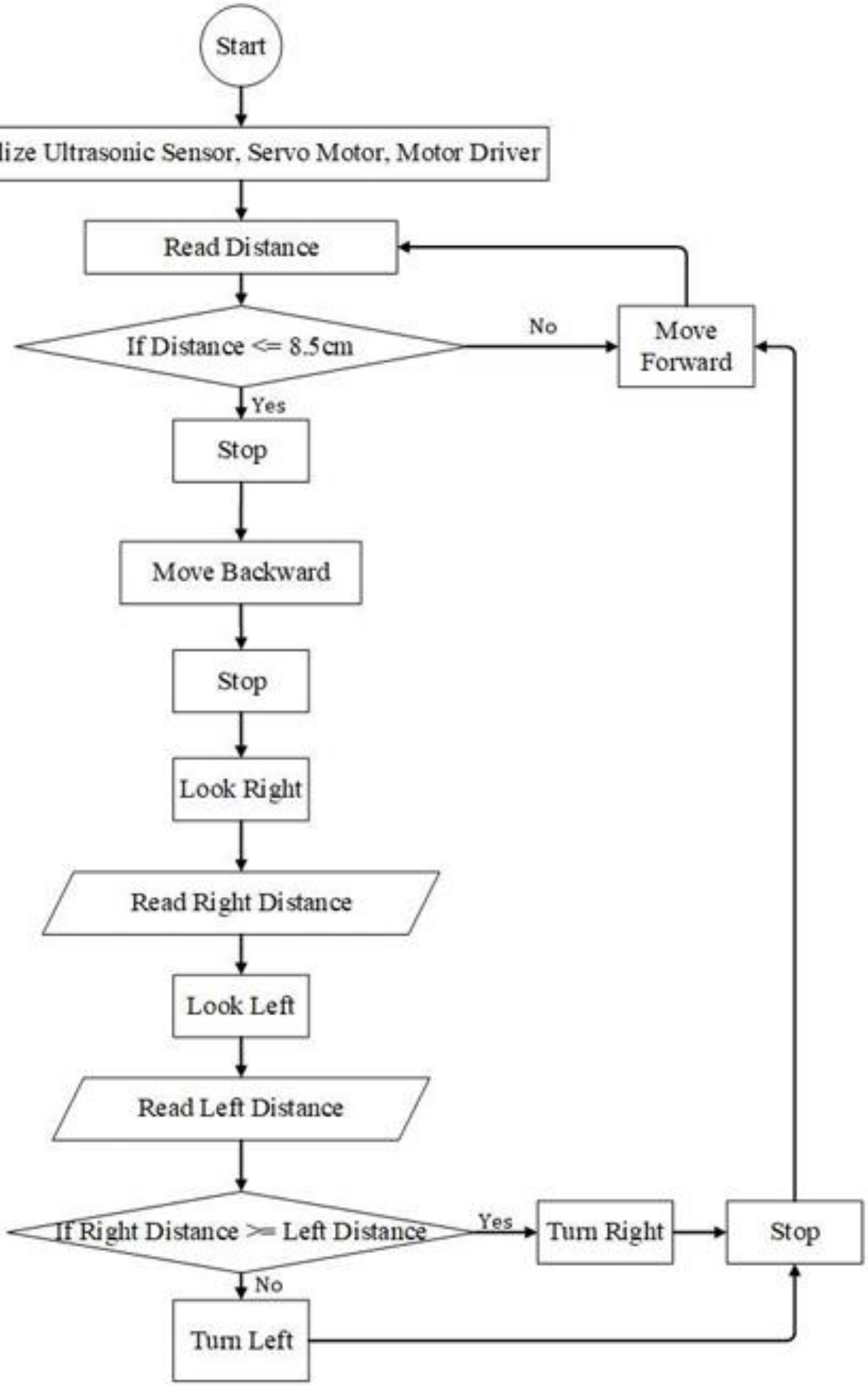

Figure 2.4.1 – Flow Chart

2.5. Source Code

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

#define TRIG_PIN A0
#define ECHO_PIN A1
#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_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);
Servo myservo;

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

void setup() {

  myservo.attach(10);
  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);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  }

void moveForward() {

 if(!goesForward)
  {
    goesForward=true;
    motor1.run(FORWARD);
    motor2.run(FORWARD);
    motor3.run(FORWARD);
    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(BACKWARD);
    motor3.run(BACKWARD);
```

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

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

Motion/Motor	Left Motor 1	Left Motor 2	Right Motor 1	Right Motor 2
Move Forward	Forward	Forward	Forward	Forward
Move Backward	Backward	Backward	Backward	Backward
Turn Right	Forward	Forward	Backward	Backward
Turn Left	Backward	Backward	Forward	Forward

Table 2.5.1 – Driving System Operation

3. RESULT AND DISCUSSION

The outcome of this thesis is a simple, Arduino-controlled robot car which moves around detecting obstacles in its way and avoiding them. During operation of the robot, the ultrasonic sensor sends out an ultrasound wave to the front position (90 degrees), right position (36 degrees), and left position (144 degrees). When the wave strikes an obstacle, it bounces back and the distance is stored for the front, right, and left position. After this, the microcontroller compares the values based on its algorithm and determines whether to move forward or change path. Tests carried out on the final hardware revealed the limitations of the detection algorithm. The limitations were related to cases of some obstacles not being detected and this was as a result of the sensor not being able to measure obstacles outside the measuring range of the sensor. When an object is in the way of the car and this object is not within the line of sight of the sensor, it will not be detected thereby leading to collision. To avoid this, the testing was further carried out in an enclosed area where the wall is the only obstacle, and the car was able to move freely without collision. To implement a car which will detect multiple obstacles and avoid them, more sensors have to be used in order to cover a wider range for obstacle detection.

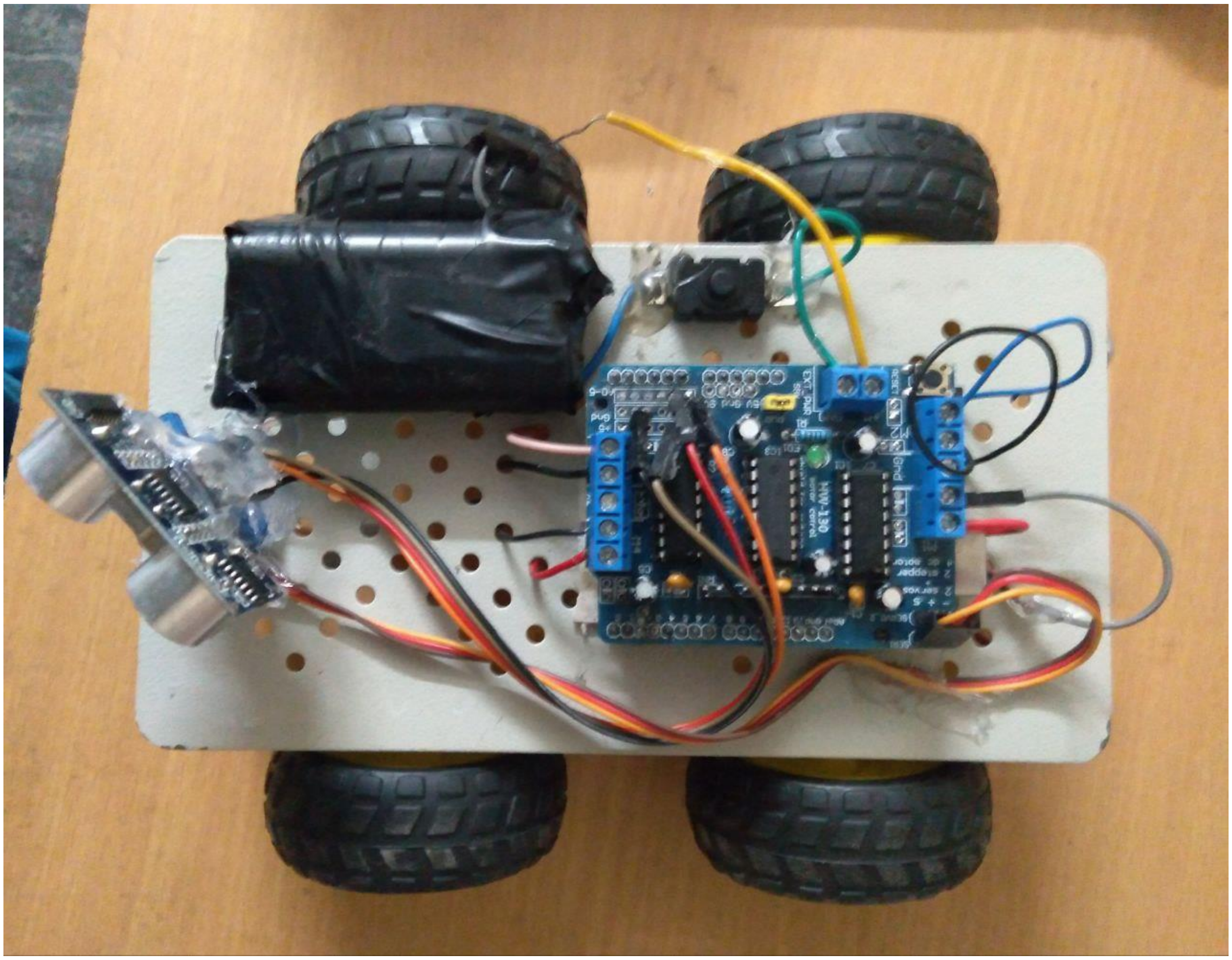

Figure 3.1 – Obstacle Avoidance Car

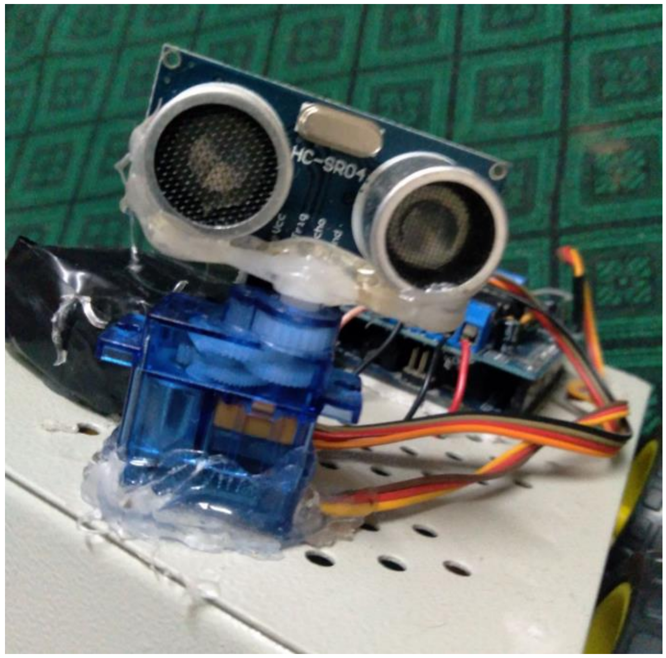

Figure 3.2 – Changes the path based on the Ultrasonic sensors

Sensor Performance: The sensor used is one of the most important components. It is a very important component because the performance of this robot is critically dependent on the accuracy of the sensor. The HC-SR04 has a measuring range of 30 degrees and 300cm. It operates by emitting a high-frequency sound wave (40Khz) through one of its piezo-electric transducers and detects the returning pulses (echo) through another transducer. Aside from the range issue, the sensor was able to detect objects within its range for obstacle avoidance.

Battery Stress: There was a lot of stress on the battery coming from the components, especially the motors. The motors need a lot of energy for their operation, especially cheaper motors because they have less efficiency. The motor driver shield is powered separately from the Arduino. The Arduino requires a 9-12V supply and it was supplied power using a single 9V battery while the motor driver shield requires 5-28V and was supplied power with two 9V batteries connected in series. The motors, servo motor and sensor all required 3-6V each.

Effect of Heat on Motor Driver Shield: The continuous current that the shield can deliver is dependent on how well the shield is kept cool. The printed circuit board is designed in such a way so as to draw heat out of the motor driver chips but performance can still be improved by addition of heatsinks. As the motor driver chip temperature approaches its limit, less power is delivered to the motor which would result in motors not being able to rotate. The shield was tested at room temperature with no forced air flow or heat sinks and it was able to deliver continuous current to the four motor channels before the thermal protection started reducing the current. The shield delivered enough power to move the car for about 15 minutes, after which the power was insufficient for the motors.

4. CONCLUSION

Today we are in a world of robotics, and we use different types of robots daily in our life. 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.

REFERENCES

- https://www.ijcaonline.org/archives/volume56/number5/8885-2882
- https://ieeexplore.ieee.org/document/4725791
- https://www.sciencedirect.com/science/article/pii/S1877050917302107
- http://www.datasheetcatalog.com
- http://www.instructabal.com
- http://en.wikipedia.org/wiki/Artificial_intelligence
- http://science.howstuffworks.com/robot2.htm
- http://arduino.cc/en/Main/arduinoBoardUno