Microprocessor & Assembly Language Laboratory

Course Code: CSE-358

Project Report on

Obstacle Avoiding Robot (Car)

Date of Submission: 30.03.2024

Project Supervisor

Dr. Md. Golam Moazzam

Professor

Department of Computer Science And Engineering

Jahangirnagar University

Jahangirnagar University

Savar , Dhaka

Submitted by -

Name	Class Roll	Exam Roll
Suraiya Mahmuda	364	202176
Lamia Binta Latif	366	202178
Choyon Sarker	392	202204

Title

An Obstacle Avoiding Robot (Car) using ultrasonic sensor

.

Abstract

An obstacle avoidance robot car made by using an ultrasonic sensor for its movement. A microcontroller is utilized to achieve the intended functionality. A robot is a device capable of autonomously executing tasks or operating under guidance. Robotics involves the fusion of computational intelligence and physical machinery (motors). Computational intelligence executes predetermined instructions.

The obstacle avoiding car directs itself whenever an obstacle comes in its path. This vehicle is constructed using a microcontroller from the ATmega8 series. An ultrasonic sensor detects obstacles ahead and transmits instructions to the microcontroller. Depending on the received input, the microcontroller alters the vehicle's direction by activating the motors connected through a motor drive interface to it through a motor drive.

Introduction

Obstacle avoidance is a primary requirements of any autonomous robot . Robots designed for obstacle avoidance are crafted to enable navigation in unfamiliar surroundings while sidestepping collisions. These robots perceive obstacles in their path, circumvent them, and then continue their course.
Several prevalent techniques exist for robot navigation, such as wall-following, edge detection, and line-following. For instance, a commercial system employs the wall-following approach for a floor cleaning robot operating in lengthy corridors. However, a widely utilized method for obstacle avoidance relies on edge detection. One drawback of edge detection-based obstacle avoidance is the requirement for the robot to halt in front of an obstacle to obtain a more precise measurement.
All mobile robots incorporate some form of collision prevention, varying from basic algorithms that detect obstacles and halt the robot to avoid collisions, to more advanced algorithms that enable the robot to navigate around obstacles. The latter algorithms are more complex, since they involve detection of an obstacle as well as some kind of quantitative measurements concerning the obstacle's dimension. Once these have been determined, the obstacle avoidance algorithm needs to steer the robot around the obstacle and resume motion toward the original target.

System Specification

The obstacle avoidance initiative employs a microcontroller, such as Arduino or Raspberry Pi, coupled with distance sensors like ultrasonic or infrared sensors to detect obstacles. Motor control systems are incorporated to maneuver around obstacles using servo or DC motors. The algorithm of the system interprets sensor data, utilizing PID controllers or analogous algorithms for accurate motion.

Optional features include wireless communication modules for remote control and user interfaces for manual operation or system monitoring. A robust chassis accommodates the components, and rigorous testing and calibration guarantee optimal performance, with safety measures like emergency stop mechanisms installed to avert accidents during operation.

Proposed System

The project proposes a autonomous robotic car, in which no remote is used for controlling the robotic actions. It intelligently detects obstacles present on its path through the sensors, operates without the need for external control devices.

The detail information is given in the following subtopics which will help you to understand the whole system and its design.

Hardware Specification:

- Arduino Uno R3
- 4WD Smart Robot Chassis Kit 4 Wheeler Robot Car Kit
- 12V DC motor
- SG90 Tower Pro Server Motor
- L298N H-Bridge Dual Motor Driver, Stepper Motor Driver
- Ultrasonic Distance Sensor HC-SR04
- 3.7V 18650 Li-ion Rechargeable Battery 4300mah
- 3.7V Li-ion Battery USB Type-C Charger for 18650 battery
- 2s 18650 Battery Holder Case
- Switch
- HCSR04 BRACKET Mounting Bracket For Ultrasonic Distance Sensor

Software Specification:

- NEWPING Library

- Micro Controller Programming Language : Arduino Code (the code is written is C++ with an addition of special methods and functions).

Components Description

Sensor for Obstacle avoidance-

Varieties of sensors are available which can be used for the detection of obstacles. Some of the very popular sensors are : Infrared sensors (IR), Ultrasonic sensors, Cameras, which can be used as a part of Computer Vision, Sonar. It can measure the distance in its field of view of about thousands to hundreds points.

1. Arduino Uno R3:

Arduino is a popular programmable board used to create projects. It consists of a simple hardware platform as well as a free source code editor which has a “one click compile or upload” feature. Hence it is designed in way that one can use it without necessarily being an expert programmer. Arduino offers an open source electronic prototyping platform that is easy to use and flexible for both the software and hardware. Arduino is able to sense the environment through receiving input from several sensors. It is also able to control its surrounding through controlling motors, lights and other actuators. The Arduino programming language that is based on the wiring and the Arduino development environment that is based on the processing are used to program the microcontroller found on the board. Due to its open-source environment, one is able to easily write and upload codes to the I/O board.

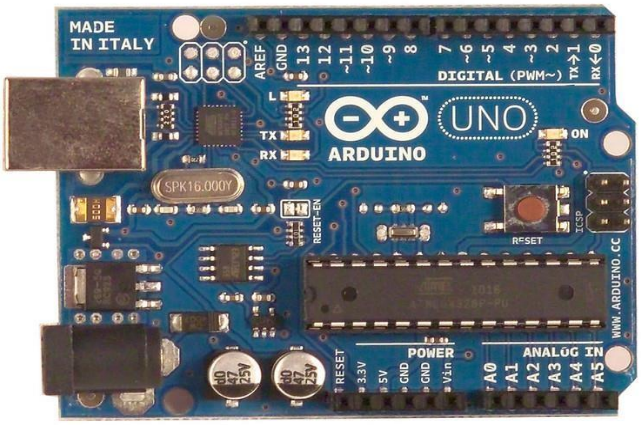

Figure-2: Arduino Uno R3 Board

2. Ultrasonic Sensor:

In the design of robot, we are using ultrasonic sensors (HC-SR04) for obstacle detection and avoidance. The ultrasonic sensors continuously emits the frequency signals, when obstacle is detected this signals are reflected back which then considered as input to the sensor. The ultrasonic sensor consists of a multi vibrator, which fixed at its base. The multi vibrator is combination of a resonator and vibrator. The ultrasonic waves generated by the vibration are delivers to the resonator. Ultrasonic sensor actually consists of two parts : the emitter which produces a 40khz sound wave and detector which detects 40khz sound wave and sends electrical signal back to the microcontroller.

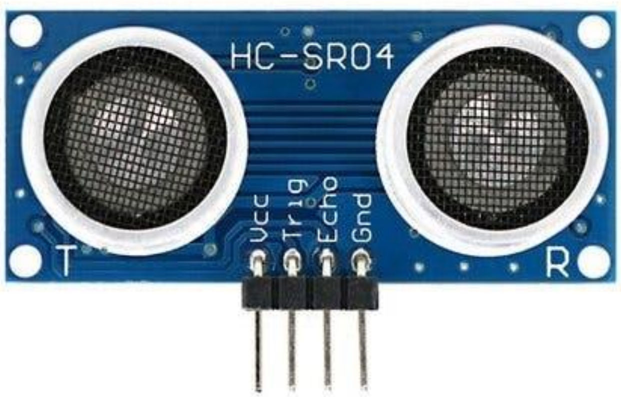

Figure-1: Ultrasonic sensor

3. Motor Sheild:

The L298N motor driver operates on principles of H-bridge configuration for bidirectional control of DC motors. It features two H-bridge circuits, allowing control of two motors independently. It accepts PWM signals for speed control and utilizes four input ports: IN1, IN2, IN3, IN4, for controlling motor direction. Logic HIGH/LOW signals on these ports determine motor rotation direction. The enable pins, ENA and ENB, control motor speed via PWM. Additionally, it incorporates built-in flyback diodes for protection against back EMF. The motor driver efficiently drives motors with higher current requirements, making it suitable for various robotic and automation applications.

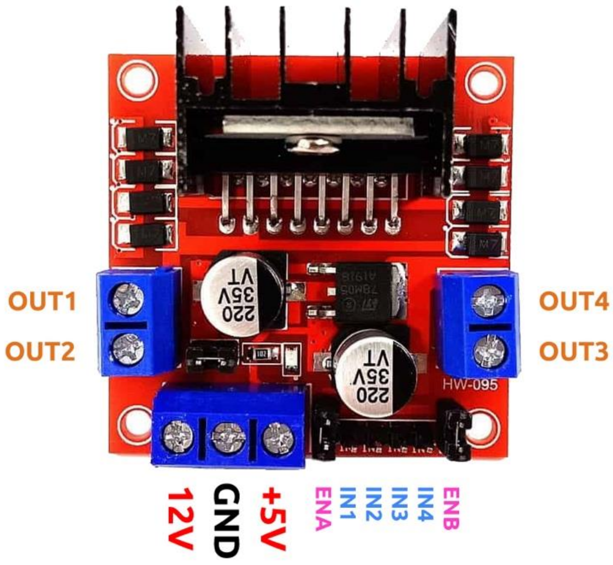

Figure-3:Arduino 2 Driver Sheild

4.SG90 Tower Pro Server Motor:

The SG90 is a popular model of servo motor produced by Tower Pro. Servo motors are commonly used in various applications, particularly in robotics and remote-controlled vehicles, due to their ability to accurately control angular position. The SG90 Tower Pro servo motor is a versatile and cost-effective option for a variety of projects requiring precise control of angular position within a limited range.

5.Motor (100rpm/ 12V DC):

The obstacle detection and avoiding robot uses two 200rpm 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. 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 experience a mechanical force.

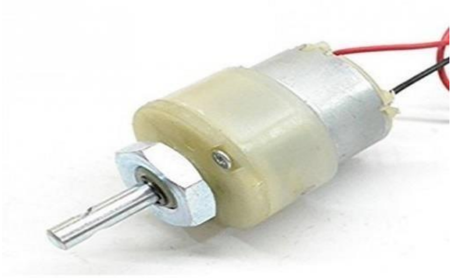

Figure-4: 100rpm/ 12V DC Motor

6. 3.7V 18650 Li-ion Rechargeable Battery

A 3.7V 18650 Li-ion rechargeable battery is a common type of lithium-ion battery cell. The 3.7V 18650 Li-ion rechargeable battery is a versatile and widely used power source suitable for many different applications where compact size, high energy density, and rechargeability are important factors.

Circuit Design

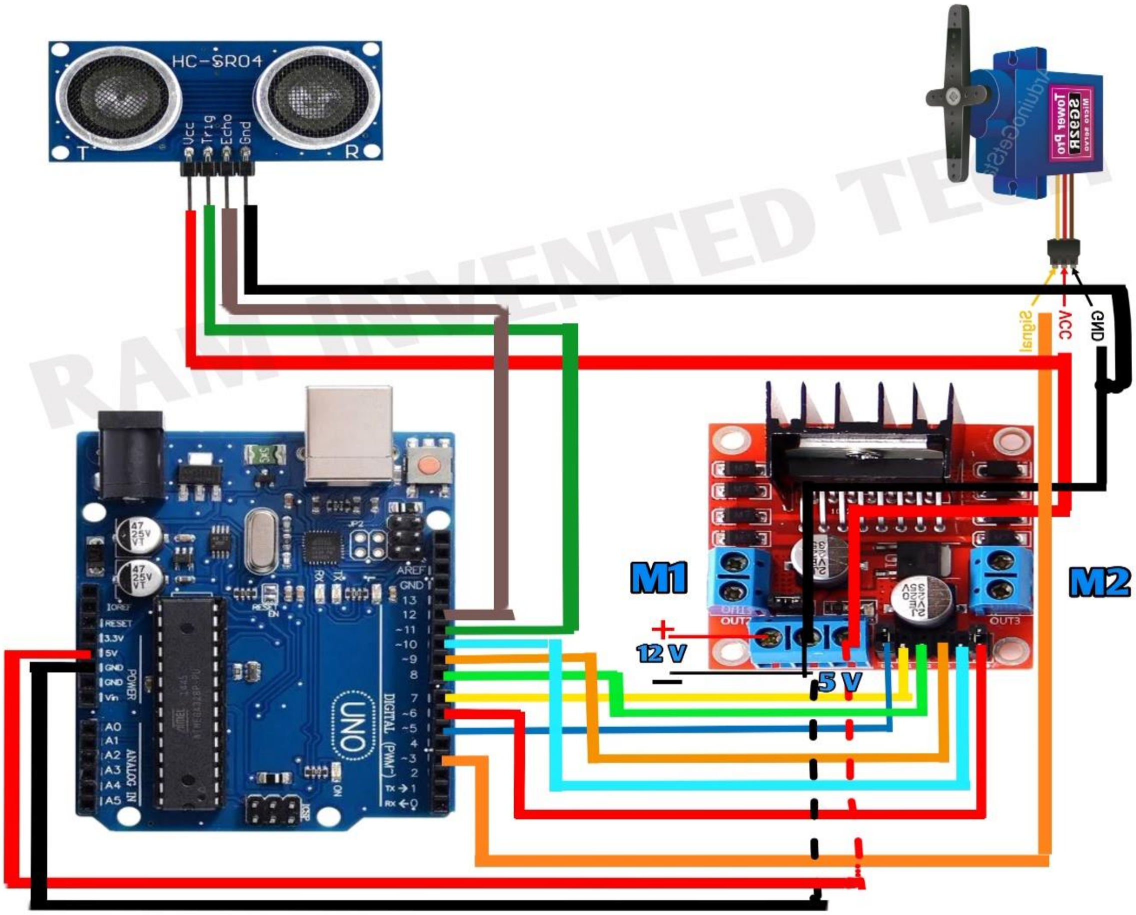

Figure-5: Circuit Diagram of obstacle avoiding robot project

Wiring Connection

Build the Arduino based obstacle avoiding car is pretty simple, to make things easy to understand, in the following paragraphs we have explained how each component is connected to Arduino uno.

Servo Motor Connection with Arduino

To connect the servo motor to the Arduino, simply attach the signal wire to any PWM control pin, connect the power wire to the 5V output from a DC-DC converter , and link the ground wire to one of the Arduino’s ground pins.

Connecting Arduino Car to Ultrasonic Sensor

Connect the sensor to the Arduino like this- VCC to 5V , GND to GND, TRIG to a digital pin 4 of. This setup allows the Arduino to interact with the sensor, further we will also be writing an Arduino car ultrasonic sensor code to enable our robot to read the distance of object ahead of it and control the wheels accordingly. More details on that can be found in the code section.

Working Principles

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 attracted 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 is detected 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 motors 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 rotate the motor M1 in forward direction and rotate the motor M2 in reverse direction such way that the car get moving in left direction.

Similarly, every time whenever an obstacle is found to be in the path of a car it will detect it and rotate the car in the left direction to avoid the obstacle.

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 10Âµs. 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. 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 hitting any obstacle.

Application

Mobile Robot Navigation Systems:

Mobile robot navigation systems are designed to enable robots to move autonomously in their environment. These systems utilize various technologies and sensors to perceive the surroundings, make decisions, and control the robot's movements. Common components include:

1. **Sensors:** Robots are equipped with sensors such as cameras, LiDAR, ultrasonic sensors, and inertial measurement units (IMUs) to gather information about the environment.

2. **Path Planning:** Once the robot has a map and knows its location, path planning algorithms are employed to determine the best route from the current location to the target while avoiding obstacles.

3. **Control Systems:** The robot's actuators are controlled to follow the planned path, and feedback from sensors is used to adjust the robot's movements in real-time.

4. **Mapping and Localization:** Algorithms create maps of the environment, and localization techniques determine the robot's position within these maps. Simultaneous Localization and Mapping (SLAM) is a common approach used for this purpose.

Household Automation /Automatic Vacuum Cleaning:

Automatic vacuum cleaners, commonly known as robotic vacuum cleaners, have become popular for household cleaning tasks. These robots incorporate technologies similar to those used in mobile robot navigation:

1. **Sensors:** Robotic vacuum cleaners are equipped with sensors like bump sensors, infrared sensors, and cliff sensors to detect obstacles, navigate around furniture, and avoid falling down stairs.

2. **Autonomous Operation:** Once the cleaning parameters are set, the robot operates autonomously, returning to its charging station when the battery is low.

3. **Mapping and Navigation:** Some advanced models use mapping algorithms to efficiently cover the entire cleaning area, ensuring no spots are missed.

4. **Obstacle Avoidance:** Algorithms help the robot detect and navigate around obstacles, preventing collisions and ensuring effective cleaning.

Dangerous Environments:

Robots are employed in environments that pose risks to human safety. Some applications include:

1. **Search and Rescue:** Robots equipped with sensors and cameras can navigate disaster-stricken areas, locate survivors, and assess the extent of damage without exposing humans to danger.

2. **Hazardous Material Handling:** Robots are employed in environments with toxic substances or chemicals, where human exposure could be fatal.

3. **Nuclear Facilities:** Robots can be deployed in nuclear power plants or radioactive environments to perform inspections and maintenance tasks without endangering human workers.
4. **Bomb Disposal:** Robots are used to handle explosive devices, minimizing the risk to bomb disposal personnel.

Experimental Result

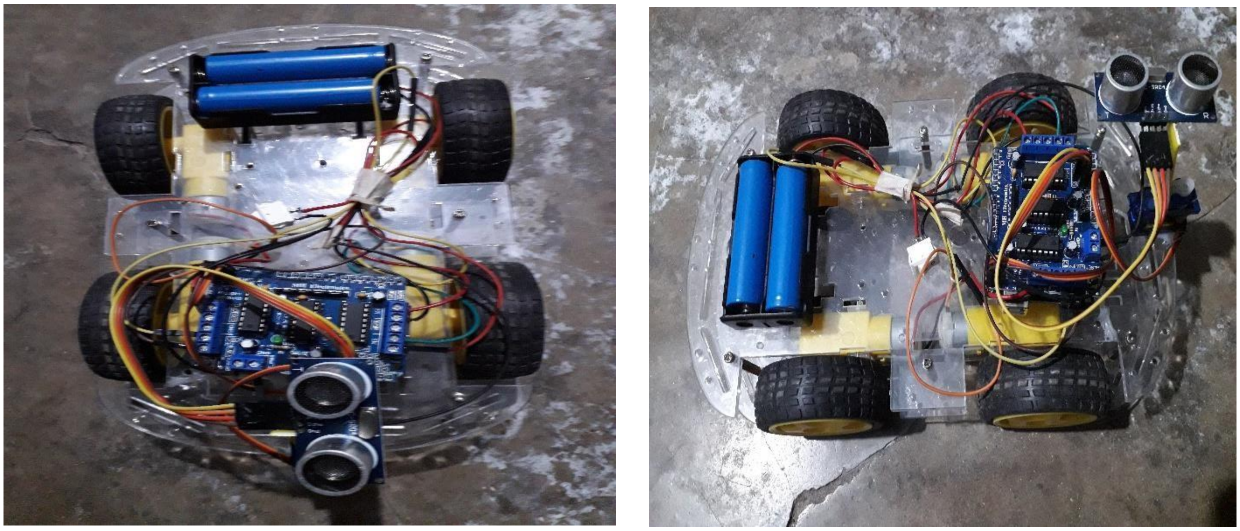

Figure-7: Experimental result of project

Source Code:

```
#include <Servo.h>
#include <NewPing.h>
#define SERVO_PIN 3
#define ULTRASONIC_SENSOR_TRIG 11
#define ULTRASONIC_SENSOR_ECHO 12
#define MAX_REGULAR_MOTOR_SPEED 200
#define MAX_MOTOR_ADJUST_SPEED 200
#define DISTANCE_TO_CHECK 45
//Right motor
int enableRightMotor=5;
int rightMotorPin1=7;
int rightMotorPin2=8;
//Left motor
int enableLeftMotor=6;
int leftMotorPin1=9;
int leftMotorPin2=10;
NewPing mySensor(ULTRASONIC_SENSOR_TRIG,
ULTRASONIC_SENSOR_ECHO, 400);
Servo myServo;
{
//Stop motors
rotateMotor(0, 0);
delay(500);
//Reverse motors
  rotateMotor(-MAX_MOTOR_ADJUST_SPEED, -
MAX_MOTOR_ADJUST_SPEED);
  delay(200);
  //Stop motors
  rotateMotor(0, 0);
```

```
void setup()
{
// put your setup code here, to run once:
pinMode(enableRightMotor,OUTPUT);
pinMode(rightMotorPin1,OUTPUT);
pinMode(rightMotorPin2,OUTPUT);
pinMode(enableLeftMotor,OUTPUT);
pinMode(leftMotorPin1,OUTPUT);
pinMode(leftMotorPin2,OUTPUT);
myServo.attach(SERVO_PIN);
myServo.write(90);
rotateMotor(0,0);
}
void loop()
{
int distance = mySensor.ping_cm();
//If distance is within 45 cm then adjust motor direction as
below
if(distance > 0 && distance < DISTANCE_TO_CHECK)
  else
  {
    rotateMotor(-MAX_MOTOR_ADJUST_SPEED,
MAX_MOTOR_ADJUST_SPEED);
    delay(200);
  }
  rotateMotor(0, 0);
  delay(200);
 }
 else
```

```
   delay(500);
//Rotate servo to left
   myServo.write(180);
   delay(500);
   //Read left side distance using ultrasonic sensor
   int distanceLeft = mySensor.ping_cm();
   //Rotate servo to right
   myServo.write(0);
   delay(500);
   //Read right side distance using ultrasonic sensor
   int distanceRight = mySensor.ping_cm();
   //Bring servo to center
   myServo.write(90);
   delay(500);
   if (distanceLeft == 0 )
   {
    rotateMotor(MAX_MOTOR_ADJUST_SPEED, -
MAX_MOTOR_ADJUST_SPEED);
     delay(200);
   }
else if (distanceRight == 0 )
   {
     rotateMotor(-MAX_MOTOR_ADJUST_SPEED,
MAX_MOTOR_ADJUST_SPEED);
     delay(200);
   }
   else if (distanceLeft >= distanceRight)
   {
     rotateMotor(MAX_MOTOR_ADJUST_SPEED, -
MAX_MOTOR_ADJUST_SPEED);
     delay(200);
   }
```

```
 {
   rotateMotor(MAX_REGULAR_MOTOR_SPEED,
MAX_REGULAR_MOTOR_SPEED);
}
}
void rotateMotor(int rightMotorSpeed, int leftMotorSpeed)
{
if (rightMotorSpeed < 0)
{
digitalWrite(rightMotorPin1,LOW);
digitalWrite(rightMotorPin2,HIGH);
}
else if (rightMotorSpeed >= 0)
{
digitalWrite(rightMotorPin1,HIGH);
digitalWrite(rightMotorPin2,LOW);
}
if (leftMotorSpeed < 0)
{
digitalWrite(leftMotorPin1,LOW);
digitalWrite(leftMotorPin2,HIGH);
}
else if (leftMotorSpeed >= 0)
{
digitalWrite(leftMotorPin1,HIGH);
digitalWrite(leftMotorPin2,LOW);
}
analogWrite(enableRightMotor, abs(rightMotorSpeed));
analogWrite(enableLeftMotor, abs(leftMotorSpeed));
}
```

Conclusion:

The goal of our project is to create a autonomous robot which intelligently detects the obstacle in his path and navigate according to the actions we set for it. This project is been designed and implemented with ATmega328P micro controller in embedded system domain. The experimental procedures have been conducted meticulously. The sensor outputs will be transmitted to the controller. Based on the program coded into the controller, it will issue commands to all connected devices.