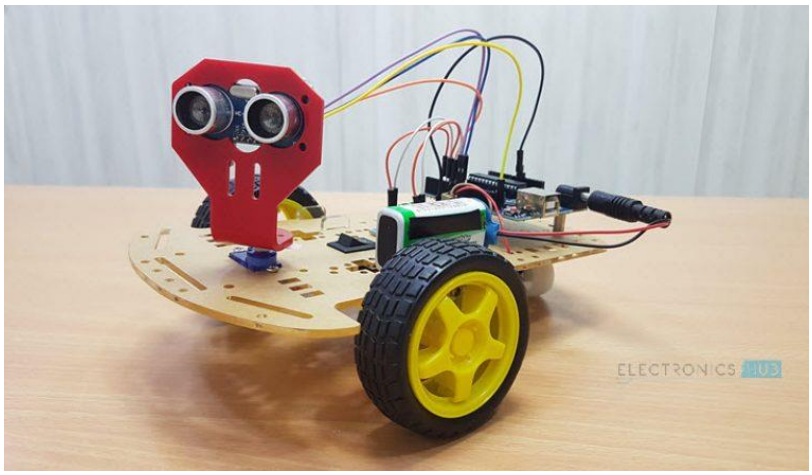
DC MOTOR TO WORK BASED ON OBSTACLE DETECTOR AND USER COMMANDS



ABSTRACT:

INTRODUCTION:

The obstacle detection field is a very broad one and a lot of obstacle detection systems have been developed in the last years. Accidents can be unintentional and sometimes can be random but are generally found to occur due to the unexpected obstacles on the moving path. We describe the design of the obstacle detector irrespective of their movements. It detects the discontinuities in terrain and alerts users of potential hazards like open manholes, potholes, protrusions, etc. which are common on Indian roads. Hence, automobile safety can be improved by anticipating a crash before it occurs and thereby providing additional time to deploy safety technologies. Warnings can be like alarms and beeps, if the vehicle is approaching a pothole, or any moving obstacle, driver can be warned in advance regarding what the road entails.

The distance calculation is used to control the movement of the robot either by one of these 4 commands (forward, backward, left or right turn and stop) through the DC motor. The motive of the project is to avoid any obstacles in travelling path. In this project, Arduino UNO has been taken with the view that, it has an in-buillt ATmega 328p. The arduino send signal to the DC. The movement of the robot is ensured by the sending and receiving signals through HC-SR04(Ultrasonic) sensor based on the varying of distance.

The ultrasonic sensor design on the obstacle robot placed at the front of the robot with the obstacle position in front. From the data analysis, the obstacle robot can determine the accuracy level of the detected distance and can stop according to the detected obstacle distance.

OBJECTIVE:

The main aim of this project through embedded C is to design, develop and to build an "Obstacle Detection System" for the vehicles on Road. The project mainly insists on reducing the increasing number of vehicular accidents in our highways, schools and metropolitan areas.

BENEFITS:

- Obstacle Detection robot will sense the obstacles in the path, avoids it and resumes its normal forward running.
- In real time, minimizing repair cost of a system, vehicle downtime, injuries avoidance through collision(prevent accidents).
- It overcomes the visibility issues for the drivers during snow and rain.
- It is very quick to compute the detect the object.
- Detects object within a certain radius.
- It has a quick response time.
- It can also be used in dark environments.

4W's and 1H:

Who:

With view of sensing the obstacle detector using Ultrasonic sensor to measure the distance between the two entities. It can used by people driving vehicles in terrain, roads, hill station and highways.

What:

Obstacle detection can be visualized through ultrasonic sensor, Infrared sensor, Camera, LIDAR and so on. This project has accompanied with Ultrasonic Sensor.

When:

When an person/ object detects any foreign object or any obstacle while following the movement of car/robo.

Where:

It can used in terrain, roads, highways, hill station while driving.

How:

It can be achieved through incorporating IC(L298p-53) along with HC-SR04(Ultrasonic) sensor.

SWOT ANALYSIS:

STRENGTH:

a) Automatic control of the functioning motor
b) Independancy, rather than relying on user's instruction
c) Adaptability
d) Can prevent faults, accidents and detect any object.
e) Easy to construct and cheaper in cost with long durability.
f)

WEAKNESS:

a) May not detect all objects
b) Can produce false reading at certain occasion
c) Its has special physical limits (like covers only a certain range).
d) The robot needs to stop in front of an obstacle with the view of more accurate measurements.

OPPORTUNITIES:

The robotic car to monitor if any obstacle is to be detected in the travelling path is to be ensured to avoid any accidents. This paves the opportunity to and develop robotics car to avoid the obstacle using certain sensor (like HC-SR04) in the pathway.

REQUIREMENTS:

HIGH LEVEL REQUIREMENTS:

RID	DESCRIPTION	STATUS
HLR1	ATmega 328p	IMPLEMENTED
HLR2	Ultrasonic Sensor	IMPLEMENTED
HLR3	DC Motor	IMPLEMENTED
HLR4	OS WINDOWS	IMPLEMENTED
HLR5	HARD-DISK	IMPLEMENTED
HLR6	4 GB RAM	IMPLEMENTED

LOW LEVEL REQUIREMENTS:

RID	DESCRIPTION	STATUS
LLR1	SimulIDE	IMPLEMENTED
LLR2	Arduino IDE(AVR) compiler	IMPLEMENTED
LLR3	Distance Detection	IMPLEMENTED

LLR4	Forward Movement	IMPLEMENTED
LLR5	Backward Movement	IMPLEMENTED
LLR6	Right Turn	IMPLEMENTED
LLR7	Left Turn	IMPLEMENTED

COMPONENTS:

The basic requirements, components necessary to build an obstacle detector are:

- ATmega 328p
- Arduino Uno
- DC motor
- HC-SR04 Sensor
- Jumper Wires

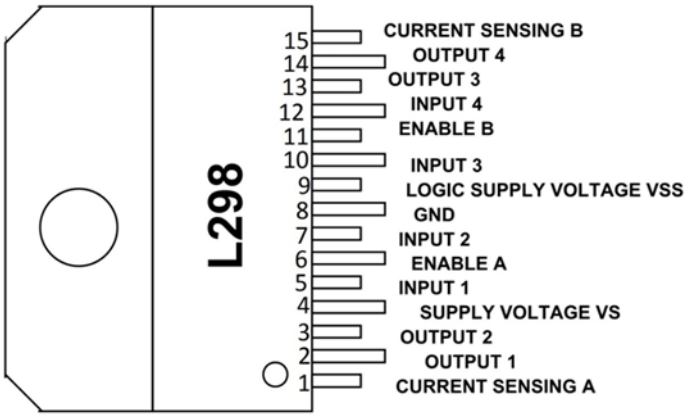
1. ATmega 328p

ATmega328P Features / Technical Specifications:

- High performance design
- Low power consumption
- Total number of Analog Input pins are 6
- Contains 32 kilobytes of flash memory
- Contains 2 kilobytes of SRAM
- Contains 1 kilobytes of EEPROM
- 16 megahertz clock speed
- Minimum & maximum temperature -40 degree centigrade to 105 degree centigrade
- Total number of Digital I/O pins are 14
- Advance RISC

- Lock program functionality for programming code security
- Contains total three timers two 8-bit and one 16 bit
- Total number of I/O pins are 23
- Total number of PWM channels are 6
- Minimum and maximum operating voltage from 1.8V DC to 5.5V DC

2. L298p-53



3. DC motor

A DC motor is an electrical machine that converts electrical energy into mechanical energy. In a DC motor, the input electrical energy is the direct current which is transformed into the mechanical rotation.

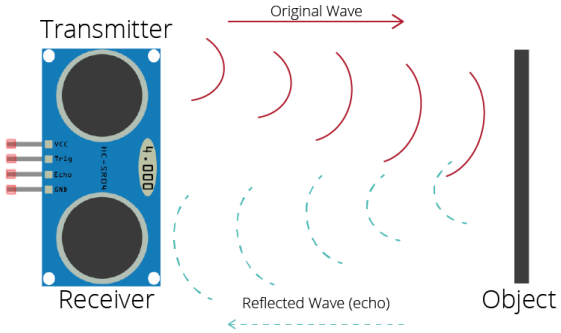
4. Ultrasonic Sensor

FEATURES:

sensor's datasheet:

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1″ – 13ft
- Resolution : 0.3 cm

- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS TTL pulse
- Echo Output Signal: TTL pulse proportional to the distance range
- Dimension: 45mm x 20mm x 15mm



An optical sensor has a transmitter and receiver, whereas an ultrasonic / level sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic / level sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables miniaturisation of the sensor head.

Distance calculation

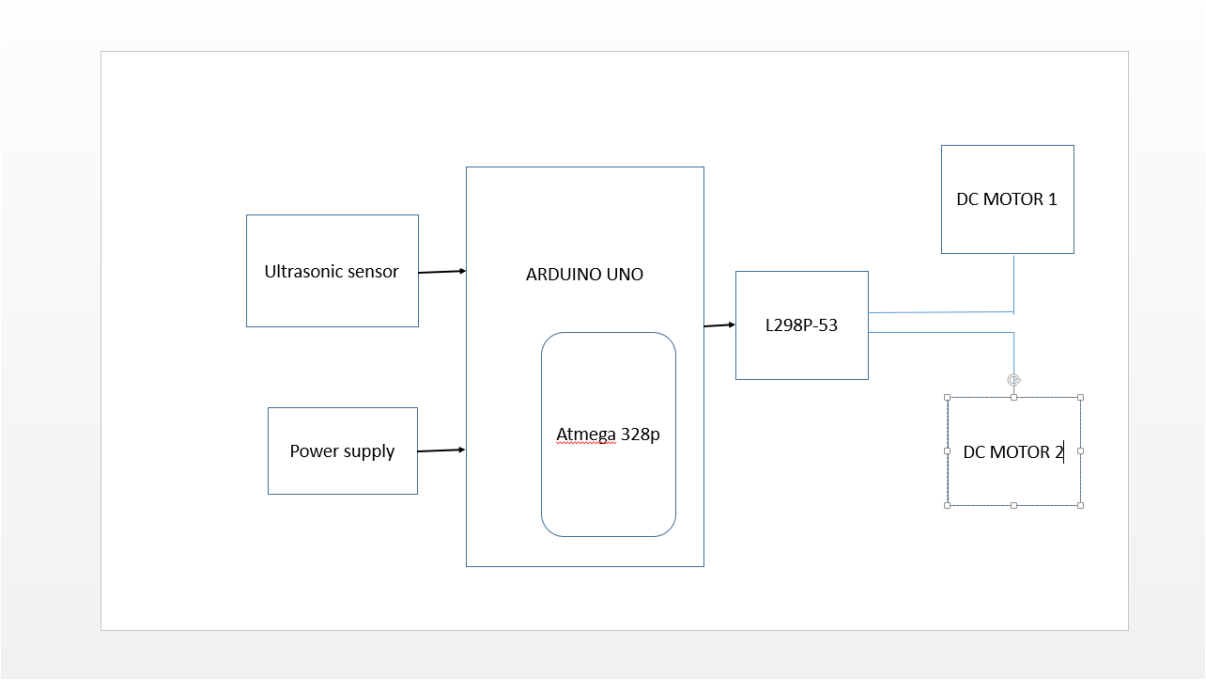
The distance can be calculated with the following formula:

Distance L = 1/2 × T × C

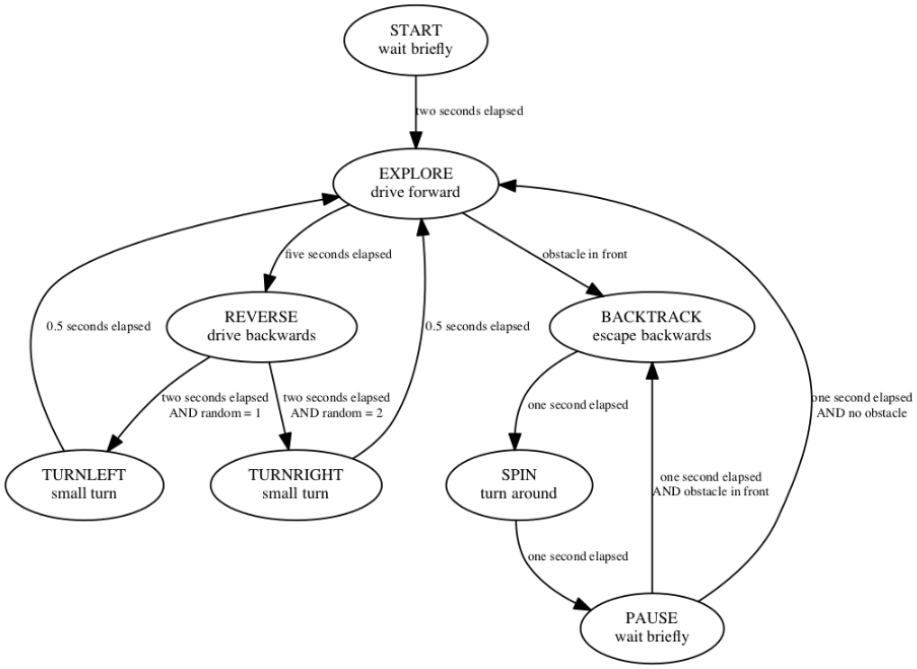
where L is the distance, T is the time between the emission and reception, and C is the sonic speed. (The value is multiplied by 1/2 because T is the time for go-and-return distance.)

DESIGN:

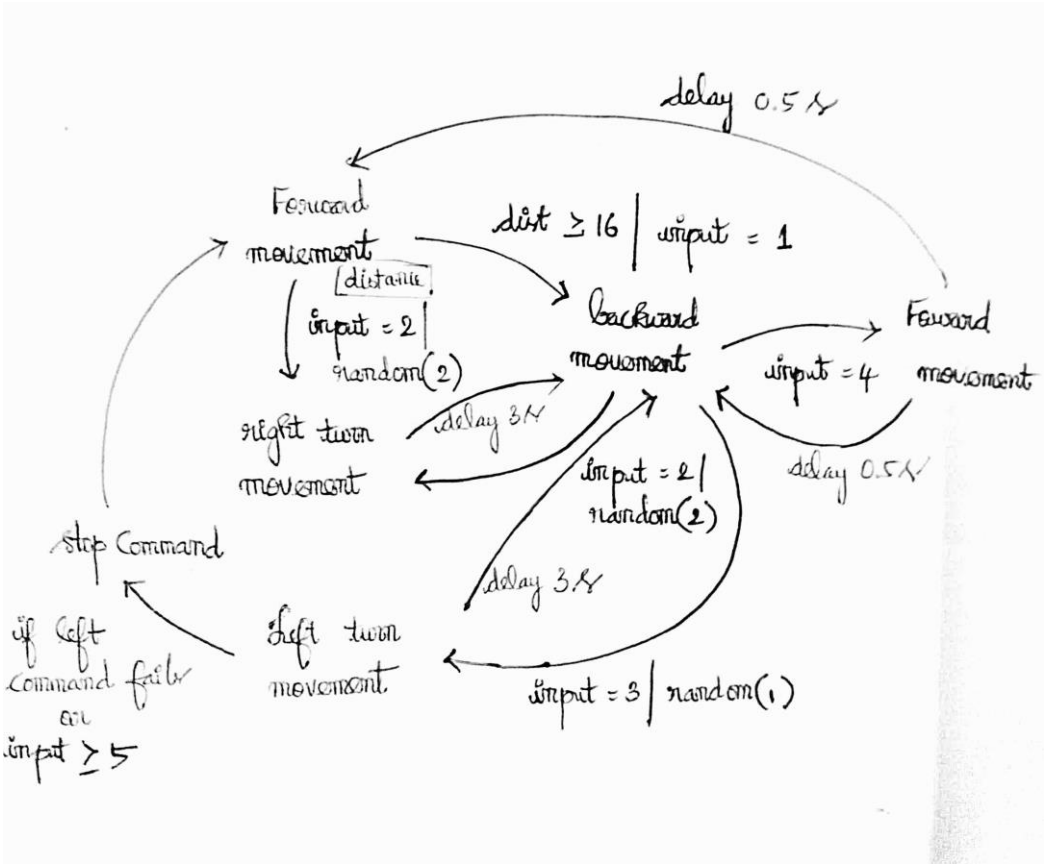
BLOCK DIAGRAM:



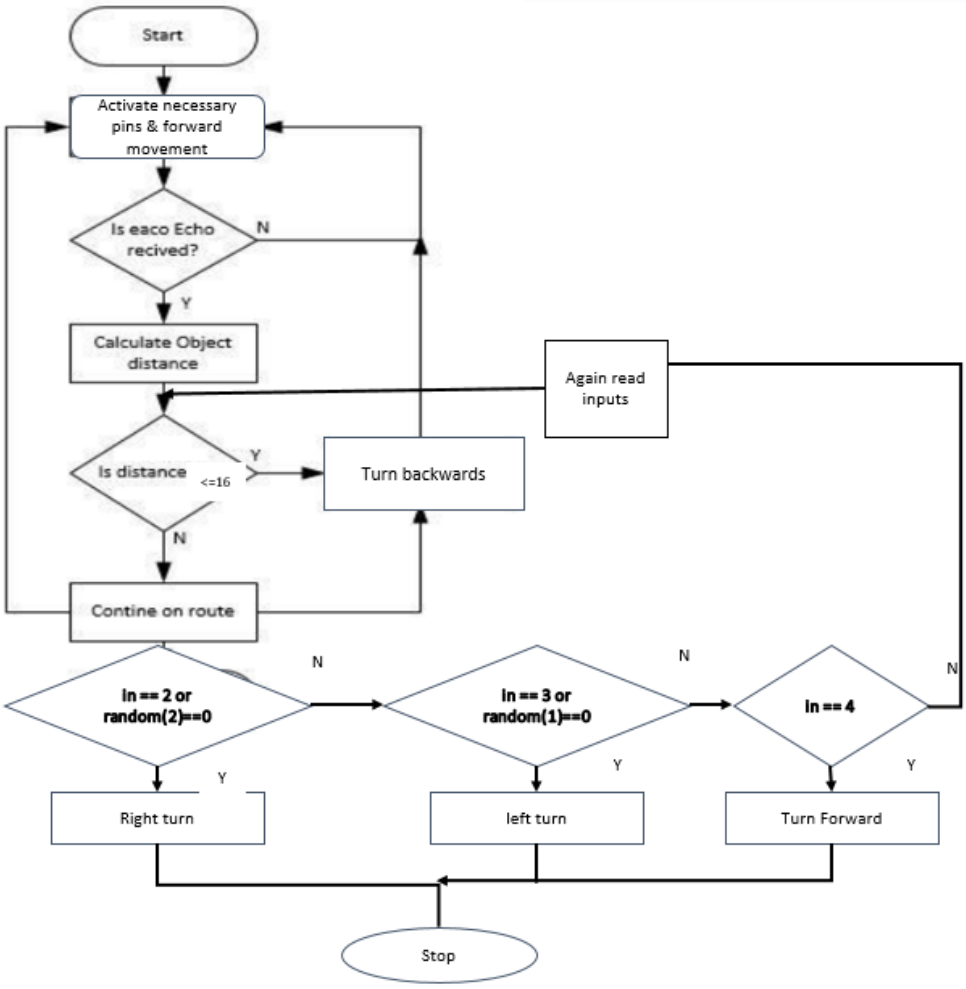
STATE TRANSISTION DIAGRAM:



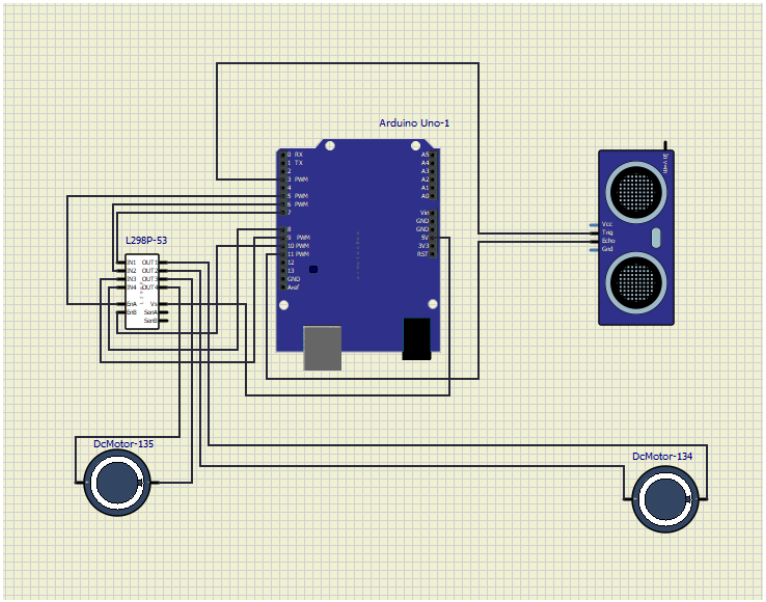
DATA FLOW DIAGRAM:


delay 0.5s
Forward movement
dist ≥ 16 | input = 1
distance
input = 2 | random(2)
backward movement
Foward movement
input = 4
right turn movement
delay 3s
delay 0.5s
input = 2 | random(2)
delay 3s
Stop Command
if left command fail or input ≥ 5
left turn movement
input = 3 | random(1)

FLOW CHART:



BASIC SCHEMATICS:



IMPLEMENTATION:

CODE:

```
#include<avr/io.h>
#include<util/delay.h>
//MOTOR 'a' CONNECTIONS
const int enA = 5;

// MOTOR 'b' CONNECTIONS
const int enB = 10;



// Set the speed (0 = off and 255 = max speed)
// If your wheels are not moving, check your connections,
// or increase the speed.
const int motorSpeed = 60;
int in;
/* Give a name to a constant value before
 * the program is compiled. The compiler will
 * replace references to Trigger and Echo with
 * 3 and 11, respectively, at compile time.
 */
#define Trigger 3
#define Echo 11
  /*
 *  This setup code is run only once, when
 */
void setup(){

  // Set the baud rate to 9600. 9600 means that
  // the serial port is capable of transferring
```

```
  // a maximum of 9600 bits per second.
  Serial.begin(9600);

  // Motor control pins are outputs
  DDRD|=(1<<DDD5) | (1<<DDD6) | (1<<DDD7);
  DDRB|=(1<<DDB0) | (1<<DDB1) | (1<<DDB2);
  //Initially motor is turned OFF
  PORTD&=~(1<<DDD6) & ~(1<<DDD7);
  PORTB|=~(1<<DDB0) & ~(1<<DDB1);

  // Setting the motor speed
  DDD5 <= 0x3C;
  DDB3 <= 0X3C;

  // Define each pin as an input or output.
  DDRB&=~(1<<DDB3);// CLEAR IT AS 0
  PORTB|=(1<<DDB3);//  INPUT SET IT TO 1(TRIGGER)
  DDRD|=(1<<DDD3);// OUTPUT BIT(ECHO)
  // Initializes the pseudo-random number generator
  // Needed for the robot to wander around the room
  go_forward();   // Go forward
  _delay_ms(5000);     // Pause 5 SECONDS
}

/*
 * This is the main code that runs again and again while
 * the Arduino is connected to power.
 */
void loop(){
  int distance = distDetect();
```

```
 // If obstacle <= 16 inches away
 if(Serial.available()>=1){
   in = Serial.read();
   if((in ==1) ||(distance <= 16)){
     //Serial.println("Obstacle detected ahead");
     go_backwards();   // Move in reverse
     _delay_ms(2000);
   }
   /* Go left and right to avoid the obstacle*/
   if((in ==2)||(random(2)==0)){
     // Generates 0 or 1, randomly
     go_right();  // Turn right
      _delay_ms(3000);
  }
   if((in==3)||(random(1)==0)){
     go_left();  // Turn left
      _delay_ms(3000);
   }
   if(in==4){
   go_forward();  // Move forward
   _delay_ms(500);
  }
 else{
   stop_all();
 }
}
}
```

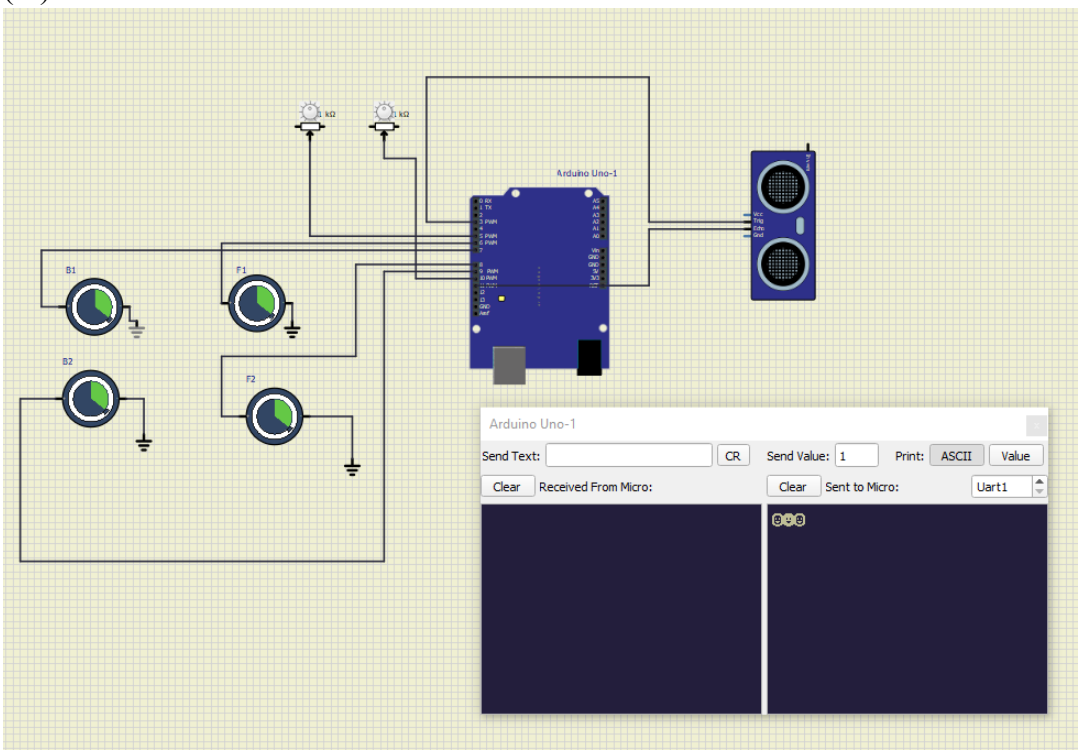
```
/* Returns the distance to the obstacle as an integer
 */
int distDetect () {
  int distance = 0;
  int average = 0;
  // Grab four measurements of distance and calculate
  // the average.
  for (int i = 0; i < 4; i++) {
  // Make the Trigger LOW (0 volts)
    // for 2 microseconds
    PORTB&=~(1<<DDB3);
    _delay_ms(2000);


    // Emit high frequency 40kHz sound pulse
    // (i.e. pull the Trigger)
    // by making Trigger HIGH (5 volts)
    // for 1 microseconds
    PORTB|=(1<<DDB3);
    _delay_ms(1000);
    PORTB&=~(1<<DDB3);

    // Detect a pulse on the Echo pin 8.
    // pulseIn() measures the time in
    // microseconds until the sound pulse
    // returns back to the sensor.
    distance = pulseIn(Echo,1);

    // Speed of sound is:
    // 13511.811023622 inches per second
```

```
    // 13511.811023622/10^6 inches per microsecond
    // 0.013511811 inches per microsecond
    // Taking the reciprocal, we have:
    // 74.00932414 microseconds per inch
    // Below, we convert microseconds to inches by
    // dividing by 74 and then dividing by 2
    // to account for the roundtrip time.
    distance = distance / 74 / 2;

    // Compute running sum
    average += distance;

    // Wait 10 milliseconds between pings
    _delay_ms(10);
  }

  // Return the average of the four distance
  // measurements
  return (average / 4);
}
/*
 *  Forwards, backwards, right, left, stop.
 */
void go_forward() {
  PORTD|=(1<<DDD7);  //F1
  PORTD|=(1<<DDD6);   //B1
  PORTB|=(1<<DDB1);   //B2
  PORTB|=(1<<DDB0);   //F2
}
void go_backwards() {
```

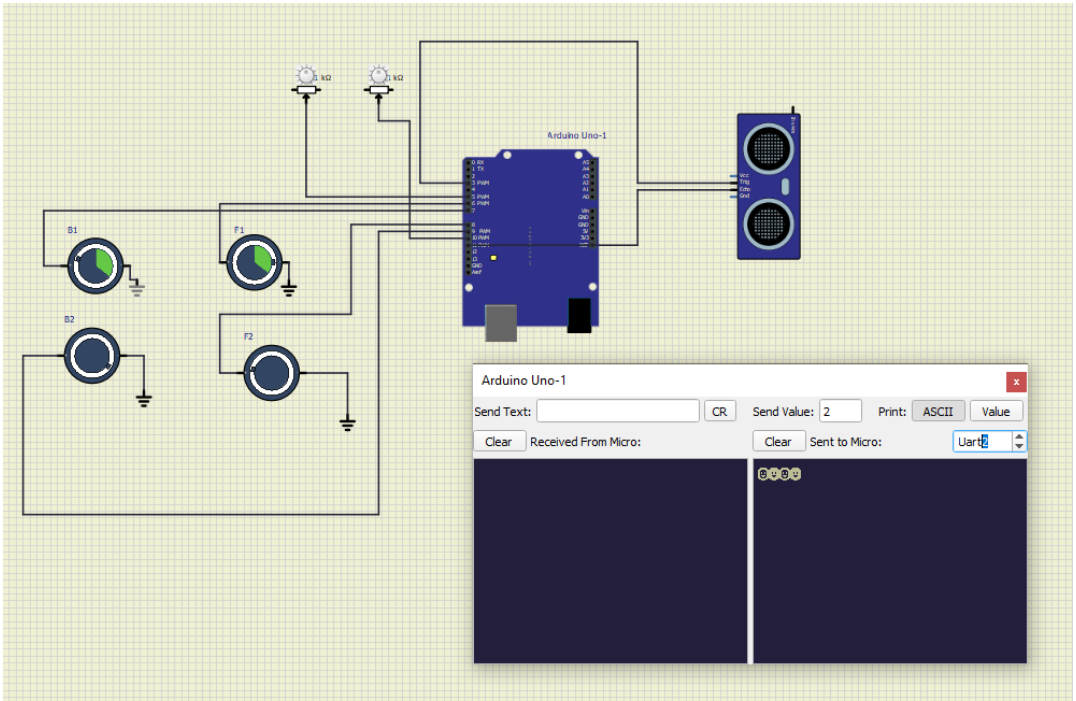
```
  PORTD&=~(1<<DDD7);
  PORTD|=(1<<DDD6);
  PORTB&=~(1<<DDB1);
  PORTB|=(1<<DDB0);
}
void go_right() {
   PORTD|=(1<<DDD7);
  PORTD|=(1<<DDD6);
   PORTB&=~(1<<DDB1);
  PORTB&=~(1<<DDB0);
}
void go_left() {
  PORTD&=~(1<<DDD7);
  PORTD&=~(1<<DDD6);
  PORTB|=(1<<DDB1);
  PORTB|=(1<<DDB0);
}
void stop_all() {
  PORTD&=~(1<<DDD7);
   PORTD&=~(1<<DDD6);
   PORTB&=~(1<<DDD1);
    PORTB&=~(1<<DDD0);
}
```

TEST CASES:

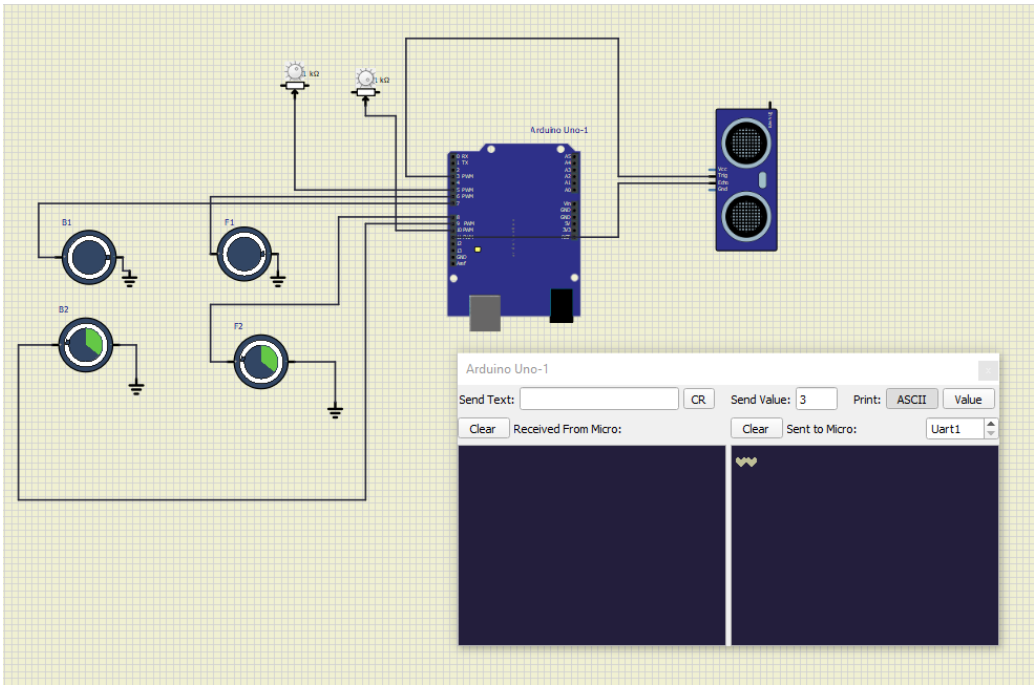
- When the robotic car with Dc motor is within the 16 meter distance from the obstacle (or) User's command as 1 – MOVE BACKWARDS



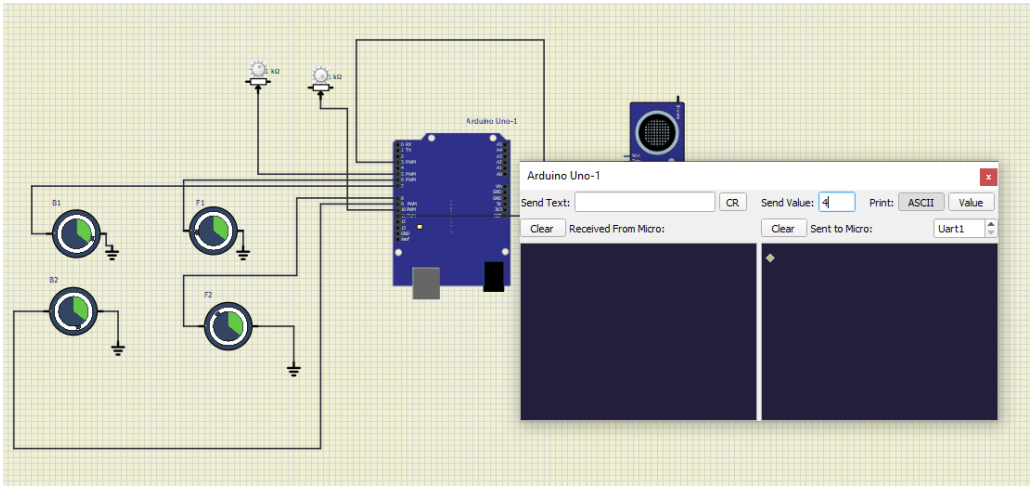
- When the random generating of random numbers between 0 and 1 (or) User's command as 2 – TURN RIGHT



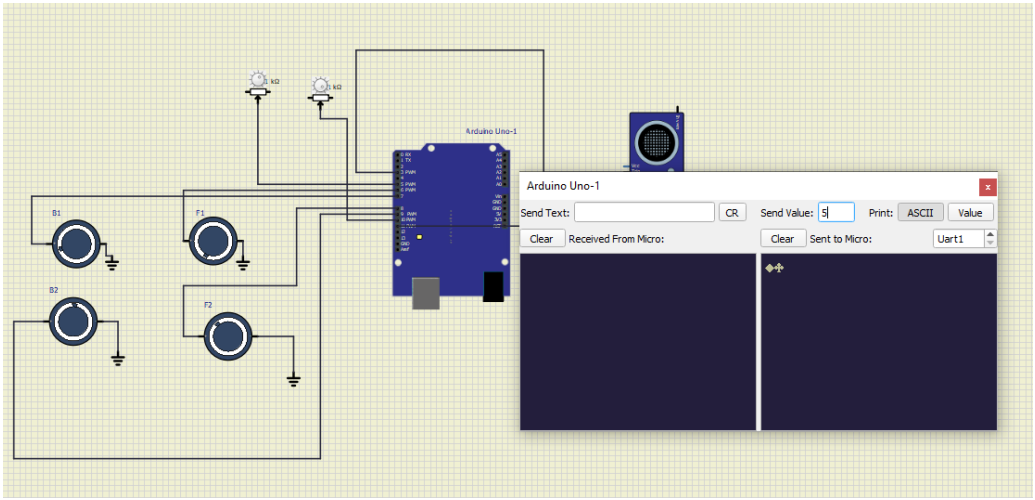
- When the random generator 1 continuously (or) User's command as 3– TURN LEFT



- When the User's command as 4 – MOVE FORWARD



- When the User needs to stop the motor to stop functioning



OUTPUT:

HEX FILE:

file:///C://Users//PERSONAL//AppData//Local//Temp//arduino_build_410181/obstacle-follower.ino.hex

REFERENCES:

[1] Amrutha.S, Raibagi, Surabhi Anand.B, Shwetha.R, "Ultrasonic anti-crashing system for automobiles", International journal of advanced research in computer and communication engineering-volume-2,Issue-4, April-2012.

[2] S.P.Bhumkar, V.V.Deotare, R.V.Babar "Accident avoidance and detection on highways", International journal of engineering trends and technology-volume-3, Issue-2, pp.247-252, 2012.

[3] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles", Proceedings of the IEEE, vol.90, n0. 7, pp.1258-1271, 2002.