**Obstacle Avoidance and Bluetooth RC Bot:**

**Section-12(Group 9):**

* **Mohan Naik**
* **Durgesh**
* **Pawan Raj**
* **Shah Emaad**

**INTRODUCTION:**

**(Obstacle avoidance)** We proposed a robot that avoids the obstacle which comes in its path this robot is introduced because in many of the industries we have seen that many heavy components which they have to move for one place to another place which is not possible without the help of machines. With this we got idea and we introduce the robot named as Obstacle avoidance robot using Arduino.

Obstacle avoidance robot is design to allow robot to navigate in unknown environment by avoiding collisions. Obstacle avoiding robot senses obstacles in the path, avoids it and resumes its running.

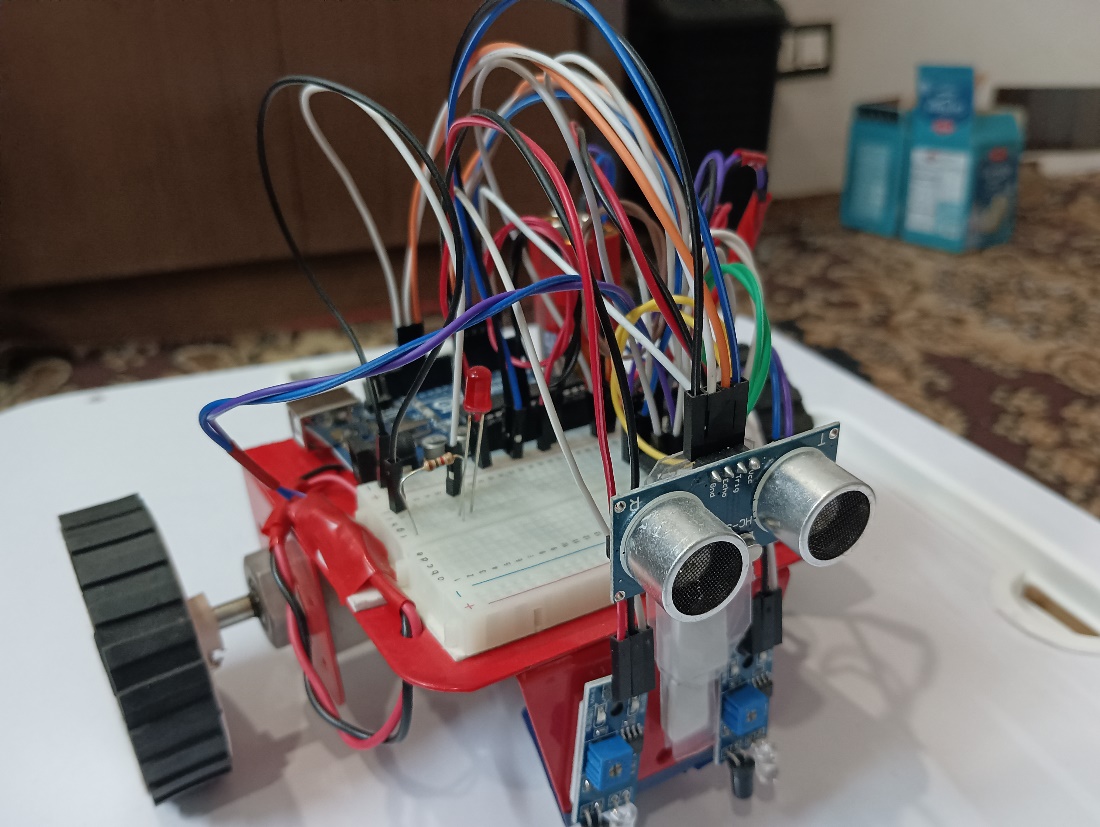
We have made use of sensors to achieve this objective. We have used two D.C. MOTORS i.e., battery operated motors. The reason behind using BO motors is it consumes less power supply and can work properly on 9-volt battery. The construction of the robot circuit is easy and small. The main component behind this robot is L29D3 microcontroller which is a brain of this robot.

The robot moves forward until an obstacle is detected to be closer than a certain distance and then turns to avoid it

**Literature survey:**

We reviewed different obstacle detecting robot mechanisms that have been built by a lot of students and other practitioners that are in existence. For an autonomous mobile robot performing a navigation-based task in a vague environment, to detect and to avoid encountered obstacles is an important issue and a key function for the robot body safety as well as for the task continuity. Obstacle detection and avoidance in a real-world environment that appears so easy to humans is a rather difficult task for autonomous mobile robots and is still a well topic in robotics. In many previous works. researched topic in a wide range sensor and of sensors various methods for detecting and avoiding obstacles for mobile robot purpose have been proposed. Good references related to the developed sensor systems and proposed detection and avoidance algorithms can be found. Based on these developed sensor systems, various approaches related to this work can be grouped.

**SYSTEM OVERVIEW:**



**THE CHASSIS AND THE BODY:**

We are using a basic Chassey which is completely made to aluminum providing a strong and lightweight frame

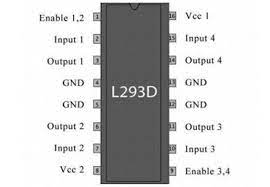
**ARDUINO UNO R3:**

Arduino is an open-source platform used for building electronics projects. Arduino consists of robot a physical programmable circuit board and IDE that runs on your computer, used to write and upload computer code to the physical board. The Arduino IDE uses a simplified version of C++, making it easier to learn to program.

It is used for controlling whole process of obstacle avoidance. Arduino reads every pin from each component and acts accordingly.

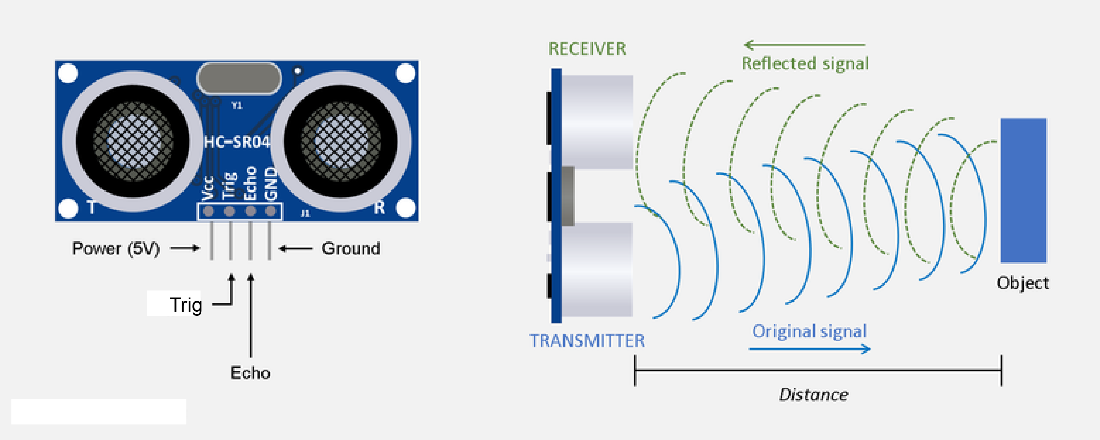
**MOTOR DRIVER (H-BRIDGE):**

Driver section consists of Motor driver and two DC motors. Motor driver is used for driving motors because Arduino does not supply enough voltage and current to motor. Arduino sends commands to this motor driver and then it drives motors in any direction as we want. Working of obstacle avoidance robot is very interesting. Then Arduino drives the motor according to sensors' output. L293D motor driver is used. L293D can rotate the motor in the forward and reverse direction.

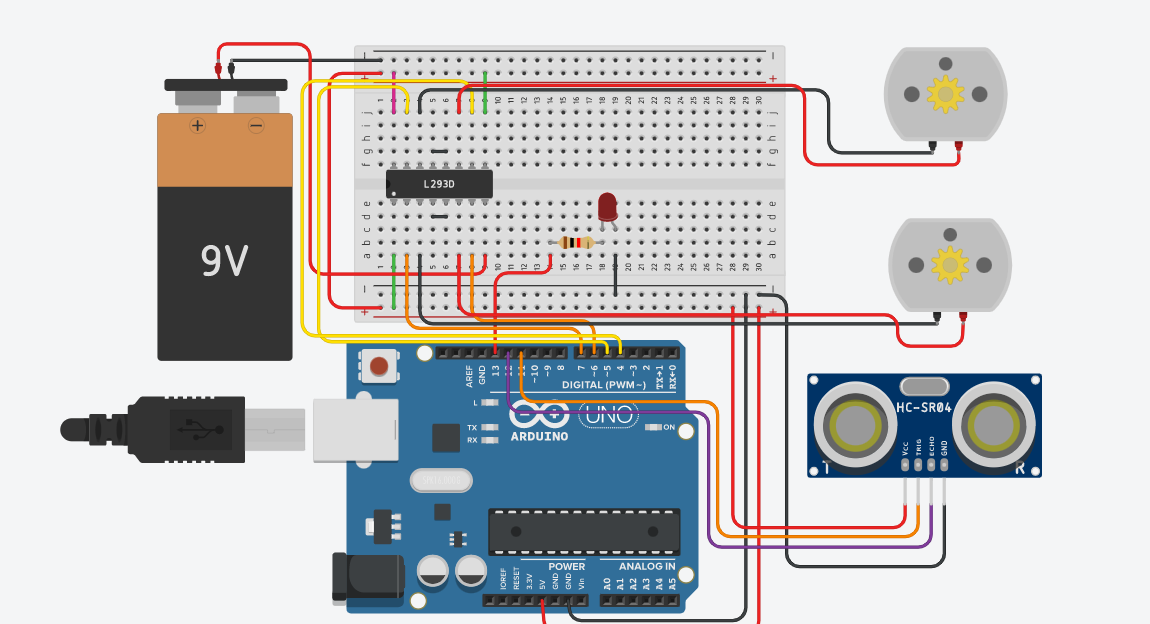


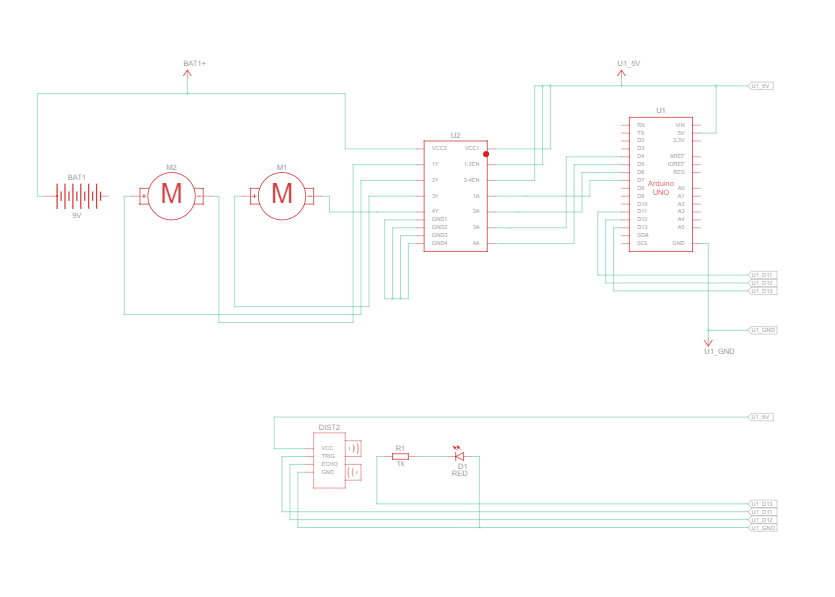
**Sensor section:**

This section contains Ultrasonic sensors. This section is used to sense the particular obstacle which comes in between its path.



**Circuit Diagram:**



**IDE PROGRAMM:**

int trig = 11;  
int echo = 12;  
int count = 0;  
int led = 13;  
int motor\_right\_1 = 6;  
int motor\_right\_2 = 7;  
int motor\_left\_1 = 4;  
int motor\_left\_2 = 5;  
int TurnTime = 1000;  
float distance , time\_echo;  
void setup() {  
 Serial.begin(9600);  
 //Pin Modes  
 pinMode(motor\_right\_1,OUTPUT);  
 pinMode(motor\_right\_2,OUTPUT);  
 pinMode(motor\_left\_1 ,OUTPUT);  
 pinMode(motor\_left\_2 ,OUTPUT);  
 pinMode(trig,OUTPUT);  
 pinMode(echo,INPUT);  
 //  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,LOW);  
 digitalWrite(trig,LOW);  
 digitalWrite(led,LOW);  
 //  
 time\_echo = 0;  
 distance = 50;  
  
}  
  
void loop() {  
 // send pulse  
 digitalWrite(trig,HIGH);  
 delay(10);  
 digitalWrite(trig,LOW);  
 // update distance  
 time\_echo = pulseIn(echo,HIGH)

// assumes the pulse of sound travels a straight path twice and returns to the sensor  
 distance = (time\_echo / 2)\*0.0343;  
 // checks if the calculated distance is less than the set parameter   
 if(distance <= 50){  
 digitalWrite(led,HIGH);  
 TurnRight();  
 delay(TurnTime);  
 }  
 else if(distance > 50){  
 Forward();  
 digitalWrite(led,LOW);   
 }   
 }

// Used to move the bot forward  
void Forward(){  
 digitalWrite(motor\_right\_1,HIGH);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,HIGH);  
 digitalWrite(motor\_left\_2 ,LOW);  
 }

// Used to move the bot backwards  
void Reverse(){  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,HIGH);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,HIGH);  
 }

// Used to turn the bot right  
void TurnRight(){  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,HIGH);  
 digitalWrite(motor\_left\_1 ,HIGH);  
 digitalWrite(motor\_left\_2 ,LOW);  
 }

// Used to turn the bot left   
void TurnLeft(){  
 digitalWrite(motor\_right\_1,HIGH);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,HIGH);  
 }

// Used to stop the bot  
void Stop(){  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,LOW);  
 }

**Testing:**

We used a red led to check if an obstacle is being detected and used the serial monitor to view the distance measured while the power supply to the motors was disconnected using the println function while we measured the distance of the sensor to the object with a scale.

**Rc Bot**

**INTRODUCTION:**

In this project we make use of the Bluetooth technology to control our machine

car. We don’t call this as a robot as this device doesn’t have any sensors.

Thereby, sensor less robots are machines. This machine can be controlled by

any human using his android mobile phone, by downloading an app and connecting

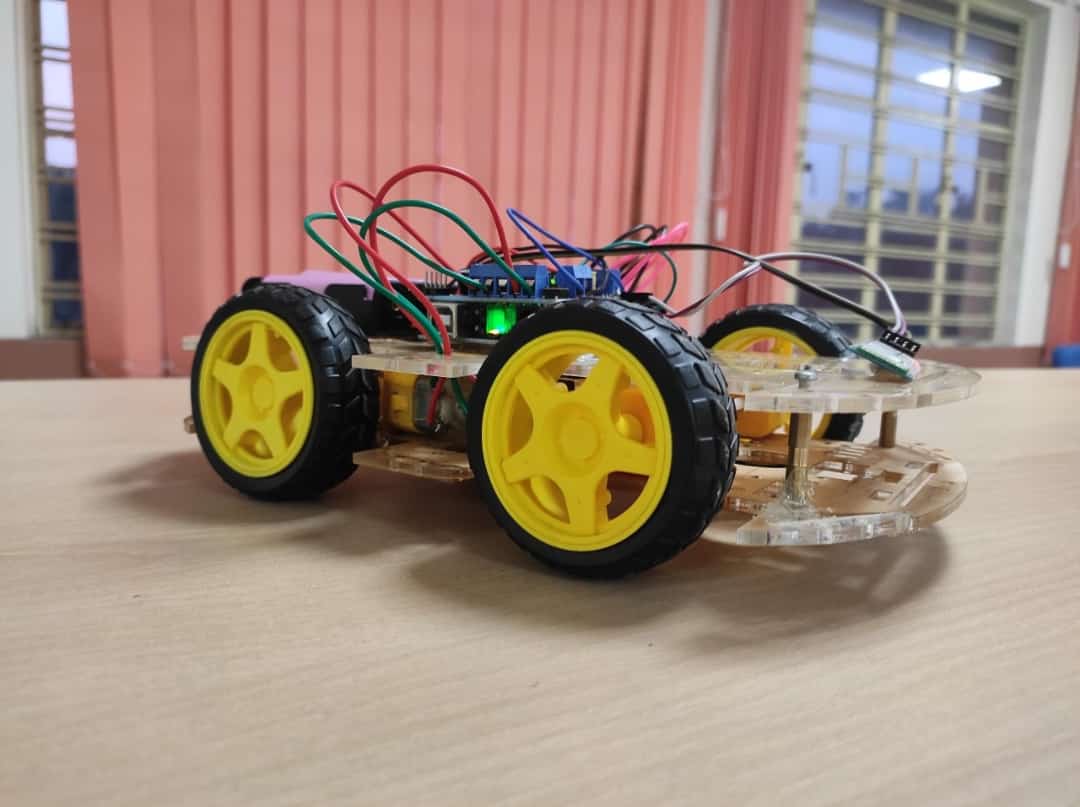
it with the Bluetooth module presents inside our car. User can perform actions

like moving forward, backward, moving left and right by the means of command

using his-her mobile phone app. The task of controlling our car is taken car by

the Arduino UNO with micro controller. Arduino play a major role in the control section and had made it easier to convert digital signals and analogue signals into physical movements. The major reason for using a Bluetooth based tech is that we can change the remote anytime – mobiles phones, tablets and laptops can be used to control the bot.

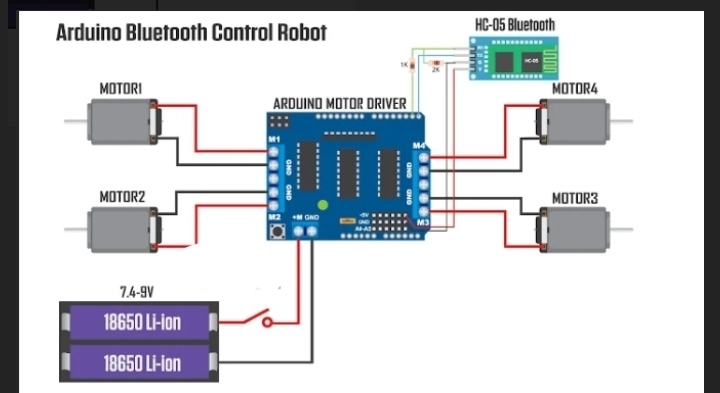
**SYSTEM OVERVIEW:**



* Chassey
* Motor driver
* Bluetooth module
* 4-DC motors
* Jumper wires

**Circuit Diagram:**

**NOTE: There is a minor change in the diagram motoe1&2 and motor 3&4 respectively share the same pins**



**IDE PROGRAMM:**

int trig = 11;  
int echo = 12;  
int count = 0;  
int led = 13;

char t;   
int motor\_right\_1 = 6;  
int motor\_right\_2 = 7;  
int motor\_left\_1 = 4;  
int motor\_left\_2 = 5;  
int TurnTime = 1000;  
float distance , time\_echo;  
void setup() {  
 Serial.begin(9600);  
 //Pin Modes  
 pinMode(motor\_right\_1,OUTPUT);  
 pinMode(motor\_right\_2,OUTPUT);  
 pinMode(motor\_left\_1 ,OUTPUT);  
 pinMode(motor\_left\_2 ,OUTPUT);  
 pinMode(trig,OUTPUT);  
 pinMode(echo,INPUT);  
 //  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,LOW);  
 digitalWrite(trig,LOW);  
 digitalWrite(led,LOW);  
 //  
 time\_echo = 0;  
 distance = 50;  
  
}

void loop () {

//Checks if an input is available and updates the value of t   
if(Serial.available()){  
 t = Serial.read();  
 Serial.println(t);  
}  
   
if(t == 'F'){

//move the bot forward

Forward();  
}  
   
else if(t == 'B'){

// move the bot backwards  
 Reverse ();

}  
   
else if(t == 'R'){

// turn the bot right  
 TurnRight();

}  
   
else if(t == 'L'){

// turn the bot left   
 TurnLeft();  
}  
  
else if(t == 'W'){ //turn led on or off)  
 digitalWrite(led,HIGH);  
}  
else if(t == 'w'){  
 digitalWrite(led,LOW);  
}  
   
else if(t == 'S'){

// Stops the bot

Stop();  
}  
delay(100);  
}

// Used to move the bot forward  
void Forward(){  
 digitalWrite(motor\_right\_1,HIGH);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,HIGH);  
 digitalWrite(motor\_left\_2 ,LOW);  
 }

// Used to move the bot backwards  
void Reverse(){  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,HIGH);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,HIGH);  
 }

// Used to turn the bot right  
void TurnRight(){  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,HIGH);  
 digitalWrite(motor\_left\_1 ,HIGH);  
 digitalWrite(motor\_left\_2 ,LOW);  
 }

// Used to turn the bot left   
void TurnLeft(){  
 digitalWrite(motor\_right\_1,HIGH);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,HIGH);  
 }

// Used to stop the bot  
void Stop(){  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,LOW);  
 }

**Concept:**

Based on these two bots we intended to form a bot which has both obstacle avoidance and Bluetooth control modes with obstacle avoidance as the default to make a multi-purpose bot.

This could not be demonstrated as we were unable to procure some parts in lab and our first motor driver being faulty so we made two separate projects as proof of concept.

This bot is supposed to work by adding a Bluetooth module to the design used for the obstacle avoidance bot.

The code used for this bot would be same as the code for the RC bot while replacing the stop command with an obstacle avoidance mode.

**IDE PROGRAMM:**

int trig = 11;  
int echo = 12;  
int count = 0;  
int led = 13;

char t;   
int motor\_right\_1 = 6;  
int motor\_right\_2 = 7;  
int motor\_left\_1 = 4;  
int motor\_left\_2 = 5;  
int TurnTime = 1000;  
float distance , time\_echo;  
void setup() {  
 Serial.begin(9600);  
 //Pin Modes  
 pinMode(motor\_right\_1,OUTPUT);  
 pinMode(motor\_right\_2,OUTPUT);  
 pinMode(motor\_left\_1 ,OUTPUT);  
 pinMode(motor\_left\_2 ,OUTPUT);  
 pinMode(trig,OUTPUT);  
 pinMode(echo,INPUT);  
 //  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,LOW);  
 digitalWrite(trig,LOW);  
 digitalWrite(led,LOW);  
 //  
 time\_echo = 0;  
 distance = 50;  
  
}

void loop () {

if(Serial.available()){  
 t = Serial.read();  
 Serial.println(t);  
}  
   
if(t == 'F'){

//move the bot forward

Forward();  
}  
   
else if(t == 'B'){

// move the bot backwards  
 Reverse ();

}  
   
else if(t == 'R'){

// turn the bot right  
 TurnRight();

}  
   
else if(t == 'L'){

// turn the bot left   
 TurnLeft();  
}  
  
else if(t == 'W'){ //turn led on or off)  
 digitalWrite(led,HIGH);  
}  
else if(t == 'w'){  
 digitalWrite(led,LOW);  
}  
   
else if(t == 'S'){

// Enters the bot to obstacle avoidance for some time depending on the delay here it’s 10 seconds

ObstacleAvoidance()

delay(10000);  
}  
delay(100);  
}

void ObstacleAvoidance () {  
 // send pulse  
 digitalWrite(trig,HIGH);  
 delay(10);  
 digitalWrite(trig,LOW);  
 // update distance  
 time\_echo = pulseIn(echo,HIGH)

// assumes the pulse of sound travels a straight path twice and returns to the sensor  
 distance = (time\_echo / 2)\*0.0343;  
 // checks if the calculated distance is less than the set parameter   
 if(distance <= 50){  
 digitalWrite(led,HIGH);  
 TurnRight();  
 delay(TurnTime);  
 }  
 else if(distance > 50){  
 Forward();  
 digitalWrite(led,LOW);   
 }   
 }

// Used to move the bot forward  
void Forward(){  
 digitalWrite(motor\_right\_1,HIGH);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,HIGH);  
 digitalWrite(motor\_left\_2 ,LOW);  
 }

// Used to move the bot backwards  
void Reverse(){  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,HIGH);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,HIGH);  
 }

// Used to turn the bot right  
void TurnRight(){  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,HIGH);  
 digitalWrite(motor\_left\_1 ,HIGH);  
 digitalWrite(motor\_left\_2 ,LOW);  
 }

// Used to turn the bot left   
void TurnLeft(){  
 digitalWrite(motor\_right\_1,HIGH);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,HIGH);  
 }

// Used to stop the bot  
void Stop(){  
 digitalWrite(motor\_right\_1,LOW);  
 digitalWrite(motor\_right\_2,LOW);  
 digitalWrite(motor\_left\_1 ,LOW);  
 digitalWrite(motor\_left\_2 ,LOW);  
 }

**Conclusion:**

We have assembled and programmed an Obstacle Avoiding and Bluetooth Remote Controlled Car. The obstacle avoidance feature in any car provides safety to all people in driving in roads and prevents accidents by simply using few sensors. Bluetooth feature allows to summon the car from a distance by manually controlling it using a mobile application. Together, by integrating these two features an Autonomous Ground Vehicle can be made which defines the future of the vehicles.

**References:**

// Online Arduino libraries

<https://www.arduino.cc/reference/en/libraries/>

// RESEARCH PAPER ON BLUETOOTH BASED HOME AUTOMATION SYSTEM by Mr. Rishabh Yadav1, Mr. Aryan Singh2 1Raj Kumar Goel Institute of Technology Ghaziabad, India 2Computer Science & Engineering ( <https://www.irjet.net/archives/V7/i4/IRJET-V7I41102.pdf> )

// WORKING PRINCIPLE OF ARDUINO AND USING IT AS A TOOL FOR STUDY AND RESEARCH by Leo Louis1 ( <https://airccse.com/ijcacs/papers/1216ijcacs03.pdf> )

// Research Paper in Ultrasonic Distance Sensor ( <https://www.researchgate.net/publication/3430936_An_ultrasonic_sensor_for_distance_measurement_in_automotive_applications> )

// RRT Algorithm ( <https://theclassytim.medium.com/robotic-path-planning-rrt-and-rrt-212319121378> )

// YouTube Reference 1 ( <https://youtu.be/vyZwYuF-wi0> )

// YouTube Reference 2 ( <https://youtu.be/aE_J7B-O4VQ> )

// Android App used <https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller>

**Video links to our project:**

[Group 9 Subgroup-A Section-12 || Obstacle Avoiding + Bluetooth Controlled Car || IIT KGP](https://youtu.be/avTy2J4Jp14)

