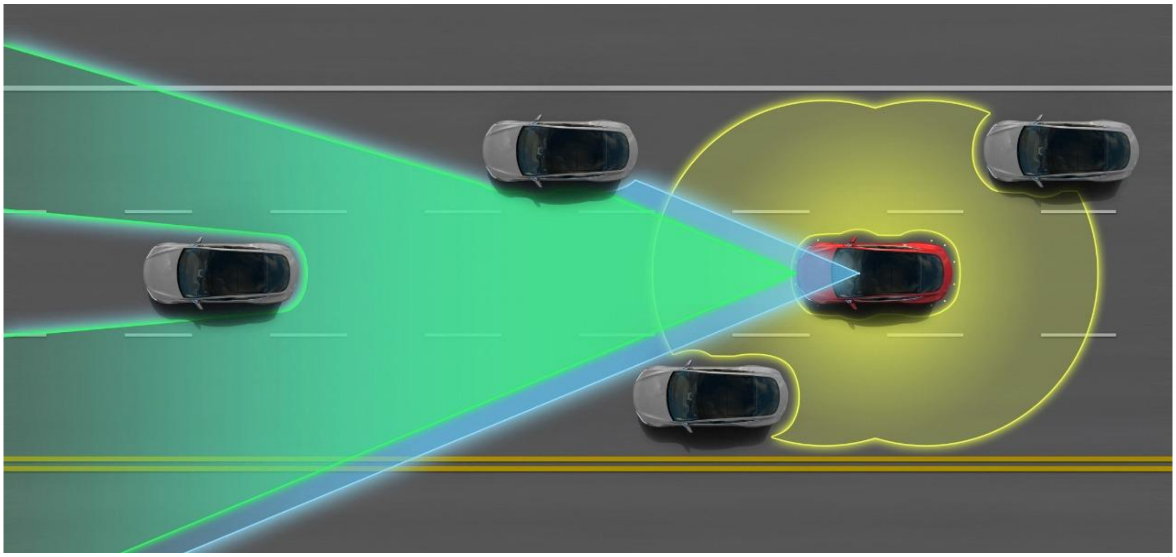
Indian Institute of Technology Jodhpur

Clearing Blind spots Using Ultrasonic sound sensors



Source:https://medium.com/self-driving-cars/tesla-enhanced-autopilot-overview-l2-self-driving-hw2-54f09fed11f1

Engineering Design 1

Team: B

December 2021 - March 2022

1. Motivation:

- According to data furnished by traffic police, all accidents are caused by drivers' fault. Since 1996, bad weather, bad roads, mechanical failure, and pedestrians' fault accounted for zero percent of all road accidents in India.
- These figures are definitely far from the realistic ones. There is no doubt that drivers' fault could be a major cause of road accidents, but the above-mentioned figures reveal that there is a need for improvement in the data collection process to get a more realistic picture of the causes of road accidents in India.
- What we need, is a technologically assisted improvement in the senses/awareness of drivers. The solutions being implemented currently are very expensive and unavailable to the mass public of India.
- It is often the poor, especially male road-users of working age that constitute the vulnerable road users (VRU) in India where VRUs shared road space with other less vulnerable users with their income level having a direct bearing on the mode of transportation used and resultant risk faced by them on that account. For example, truck drivers, especially those involved in transportation of goods across the country have to drive for a long duration at a time leading to lack of awareness. Such people can be benefitted by some technology directed towards helping them drive such a big vehicle.
- The various road features such as straight stretches, curved roads, location of bridges, etc. can also be major factors that causes a damage, and it would be very sensible to help large vehicle drivers drive in these roads. What we might need is a form of an assisted driving tool, which can potentially replicate features of expensive self-driving technology, but at an affordable price for the mass population.

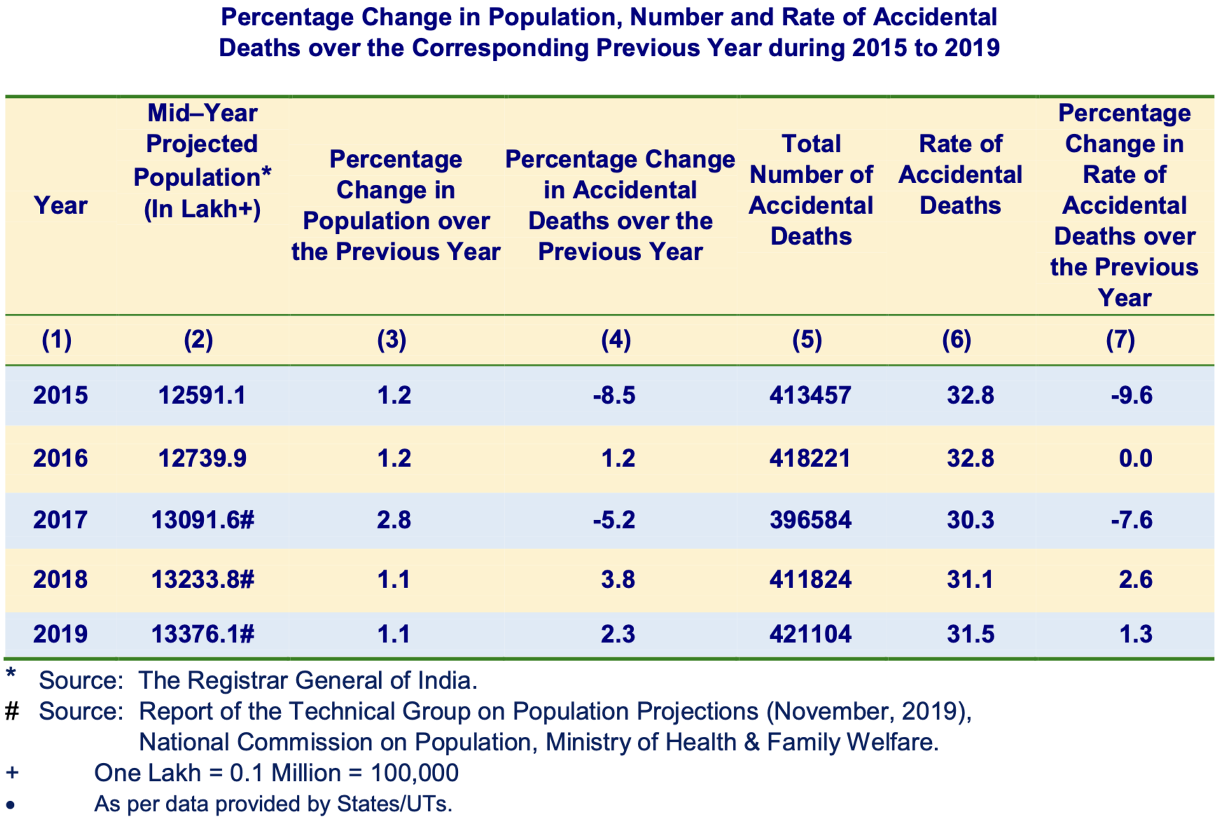
2. Problem Statement:

To assist drivers of any vehicle by clearing blind Spots to prevent accidents without breaking the bank.

3. Background Research:

i) Facts regarding the problem:

According to statista.com, In 2019, there were about 1.5 lakh reported fatalities due to road accidents in India. On average every year 3-5% GDP is expended on road accidents.



Percentage Change in Population, Number and Rate of Accidental Deaths over the Corresponding Previous Year during 2015 to 2019

Year	Mid–Year Projected Population* (In Lakh+)	Percentage Change in Population over the Previous Year	Percentage Change in Accidental Deaths over the Previous Year	Total Number of Accidental Deaths	Rate of Accidental Deaths	Percentage Change in Rate of Accidental Deaths over the Previous Year
(1)	(2)	(3)	(4)	(5)	(6)	(7)
2015	12591.1	1.2	-8.5	413457	32.8	-9.6
2016	12739.9	1.2	1.2	418221	32.8	0.0
2017	13091.6#	2.8	-5.2	396584	30.3	-7.6
2018	13233.8#	1.1	3.8	411824	31.1	2.6
2019	13376.1#	1.1	2.3	421104	31.5	1.3

* Source: The Registrar General of India.

# Source: Report of the Technical Group on Population Projections (November, 2019), National Commission on Population, Ministry of Health & Family Welfare.

+ One Lakh = 0.1 Million = 100,000

• As per data provided by States/UTs.

ii) Existing Solutions:

Presently existing solutions include self-driving vehicles (tesla, etc.) also, there are sensors inside a few cars to detect driver drowsiness to prevent accidents (ex- Intel Galileo kit, etc.) The one that we are keen on, is displaying the surroundings of the vehicle to the driver, either through an app on a display, or color coded pattern of LEDs to warn users of a nearby object.

iii) Literary Research

The data shown below from the case study by IIT Kanpur represents the distribution of the number of accidents that have occurred over the years in different types of roads and involving different types of vehicles in Patna.

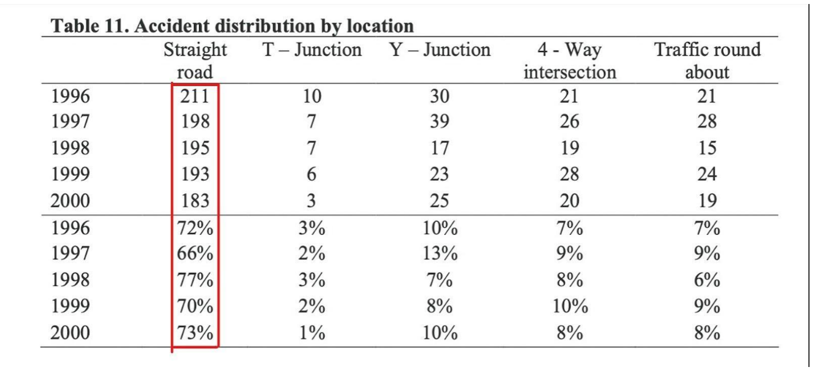


Table 11. Accident distribution by location

	Straight road	T – Junction	Y – Junction	4 - Way intersection	Traffic round about
1996	211	10	30	21	21
1997	198	7	39	26	28
1998	195	7	17	19	15
1999	193	6	23	28	24
2000	183	3	25	20	19
1996	72%	3%	10%	7%	7%
1997	66%	2%	13%	9%	9%
1998	77%	3%	7%	8%	6%
1999	70%	2%	8%	10%	9%
2000	73%	1%	10%	8%	8%

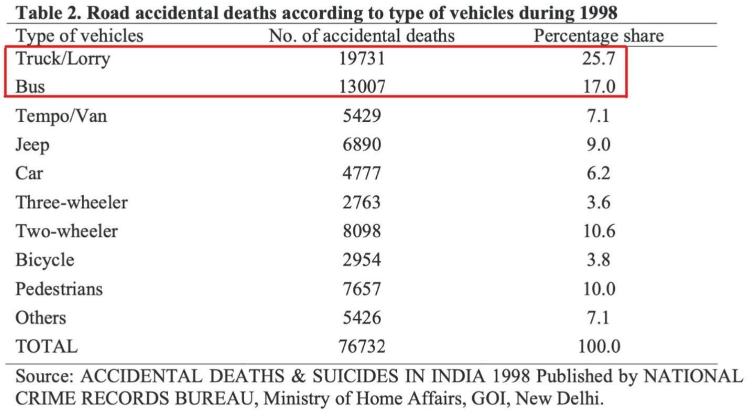


Table 2. Road accidental deaths according to type of vehicles during 1998

Type of vehicles	No. of accidental deaths	Percentage share
Truck/Lorry	19731	25.7
Bus	13007	17.0
Tempo/Van	5429	7.1
Jeep	6890	9.0
Car	4777	6.2
Three-wheeler	2763	3.6
Two-wheeler	8098	10.6
Bicycle	2954	3.8
Pedestrians	7657	10.0
Others	5426	7.1
TOTAL	76732	100.0

Source: ACCIDENTAL DEATHS & SUICIDES IN INDIA 1998 Published by NATIONAL CRIME RECORDS BUREAU, Ministry of Home Affairs, GOI, New Delhi.

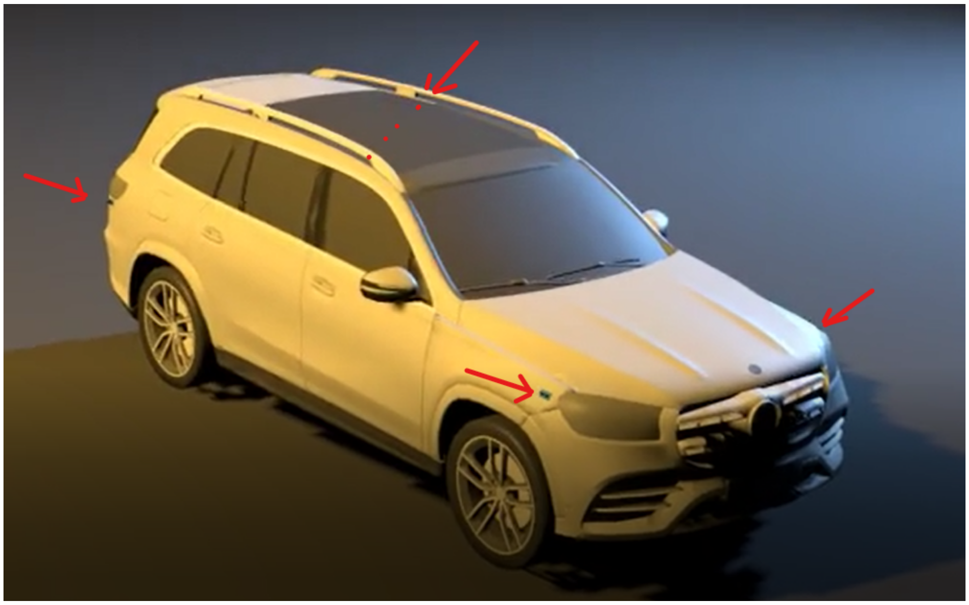
From this data we can observe that a higher number of accidents take place on straight roads compared to road junctions and roundabouts, generally due to larger vehicles like trucks, buses.

4. Ideation and Propose solutions

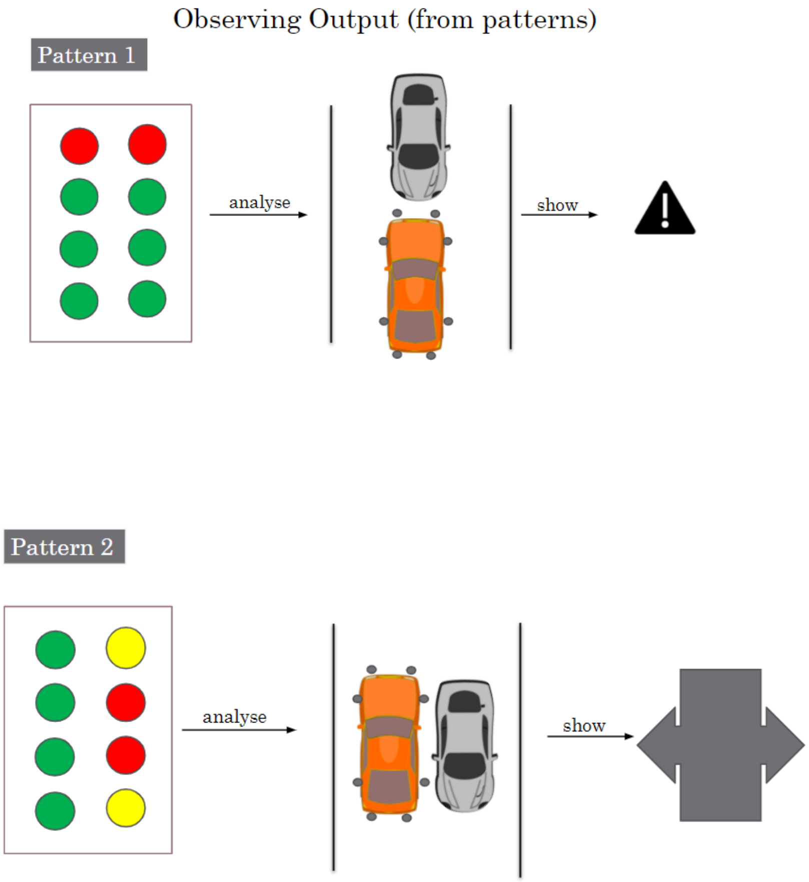
The solution that we have decided to implement is as follows:

- 4-8 ultrasonic sound sensors (depending on vehicle size) to be installed around the vehicle in order to clear blind spots. Each sensor will correspond to an LED which will glow either green or orange or red, depending on whether the detected object is at large, medium or small distance from the vehicle. These LEDs will now be fixed in a panel on the dashboard for ease of viewing by the user.

- This solution is flexible and can be modified by the creativity of the user. For example, in hilly regions on narrow mountain roads, sometimes while turning, a tire could go out of the road hanging temporarily while the vehicle is turning. This could be extremely dangerous and could lead to fatal accidents. To detect this sudden change in vertical elevation, sensors could be installed near the wheels, pointing downwards, and red color could show up if it senses a large distance (indicating that the wheel is hanging). The driver can now accordingly correct for it and be aware of such dangerous turns.



- Here is an example of 2 situations that can be interpreted by the driver as follows. (Note: these examples are trivial cases, but a more complex case could also be shown).



This solution also has a lot of potential for improvement, which will be discussed in a later section.

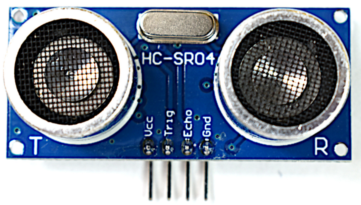
5. Evaluation based on decision matrix and the elements for comparison:

- One of the solutions which was proposed was to use cameras instead of ultrasonic sensors, but after giving it a lot of thought we came to the conclusion that cameras will be useless in the night and in extreme weather conditions such as heavy rain and also from cameras only visual data will be obtained which will not be sufficient to accurately predict the distance of the vehicles and other obstacles which are approaching towards our car.

Reasons for choosing ultrasonic sensors over radar or Lidar is as follows:

- They are the most reliable at the range we are using them in (30cm-3m)
- Cheaper than alternatives (our goal is to keep the product as cheap as possible while still replicating as much functionality as it can).

We also discussed the option of beeping speakers for close range warning, however in India since there are many traffic heavy regions where multiple objects can get within close range, the sound would be too distracting for the driver, so we decided to stick with color changing LEDs.

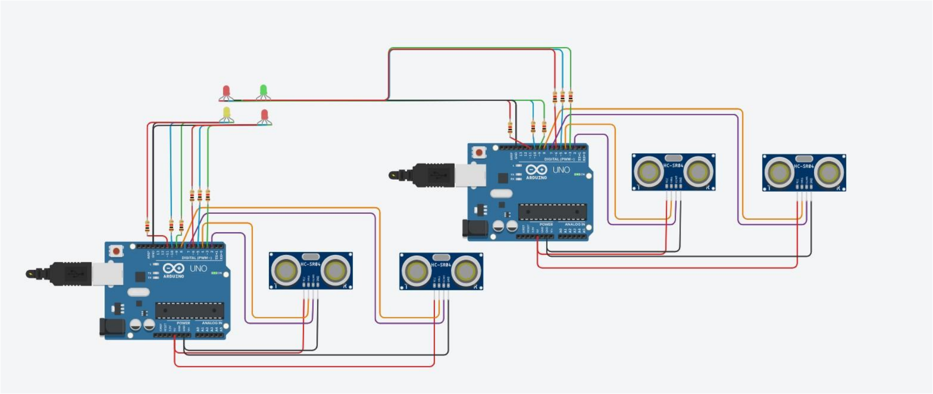


https://makersportal.com/shop/hc-sr04-ultrasonic-distance-sensor

https://www.distrelec.biz/en/lidar-lite-v3-optical-distance-sensor-5v-garmin-010-01722-00/p/30139225

6. Final solution and progress made so far

The technical details of the final solution are as follows:



- Arduino UNO R3, acting as an intermediate connection between ultrasonic sound sensors and RGB LEDs.
- These LEDs will be fixed on a panel on the dashboard for ease of viewing.

Model of 4 sensors d 4 RGB LEDs connected to 2 arduinos:

Approximate pricing:

1. Arduino UNO R3 (2): Rs.650 per piece
2. Ultrasonic sound sensor HC-SR04 (4): Rs.200 per piece
3. RGB LEDs (4): Rs.250 for 25 pcs
4. Resistors (6): Rs.50 for 20 pcs
5. Wires

Approximate Total price: **<u>Rs.2,500- Rs.4,000</u>** depending on the number of sensors required

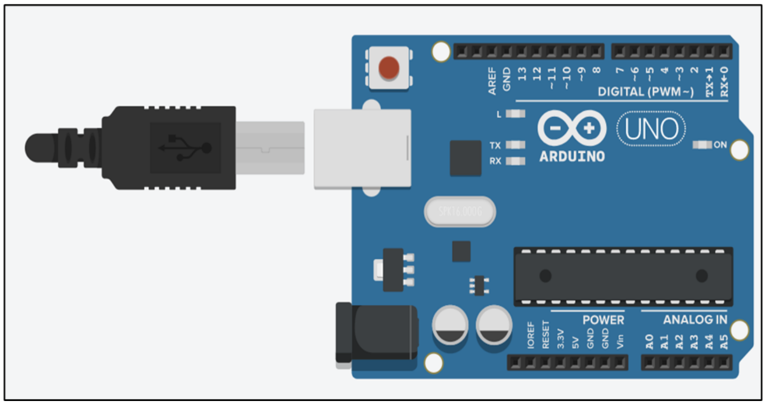
Circuit for LED panel on Tinkercad

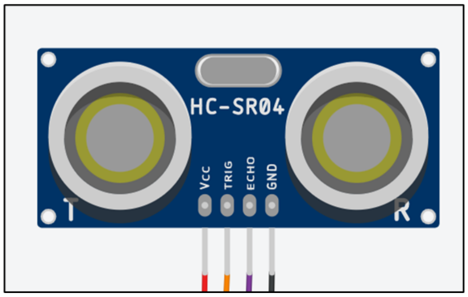
What is Tinkercad?

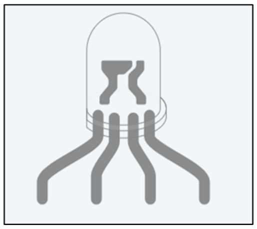
Tinkercad is a web-based 3D CAD software where we can create 3D designs and program, simulate, and assemble electric circuits.

Components we used

- **Arduino UNO R3 x 2** - easy-to-use programmable microcontroller board that can be integrated into a variety of electronic projects.
- **Ultrasonic sound sensor HC-SR04 x 4** - measures the distance to an object using ultrasonic sound waves
- **RGB LEDs x 4**
- **Resistors x 6**
- **Breadboard**
- **Connecting Wires**







Connections

- Drag all required components from the sidebar to the main panel (mentioned in components we used)

- Now, connect **Arduino board** with **ultrasonic sound sensor (HC-SR04)**

    a. Connect **Vcc** of HC-SR04 with **power pin of 5V**

    b. **TRIG** -> Digital I/O pins (**pin#4**)

    c. **ECHO** -> Digital I/O pins (**pin#2**)

    d. **GND** -> **GND of Power pins**

- Connect **RGB led** with **Arduino board** (resistor in series)

    a. **Red** terminal -> Digital I/O (**~pin#11**) [pwm – pulse width modulation]

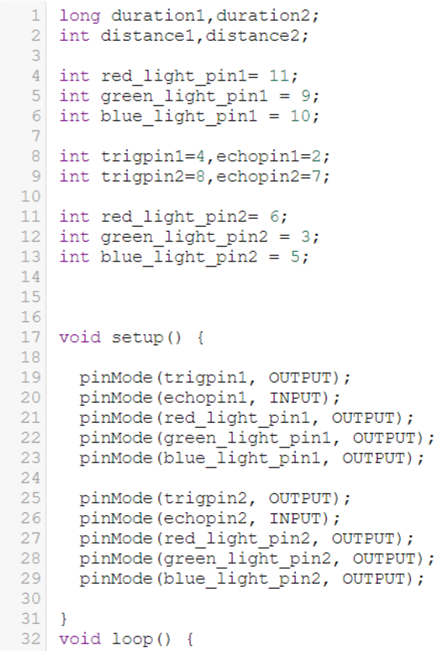
    b. **Blue** terminal -> Digital I/O (**~pin#10**)

    c. **Green** terminal -> Digital I/O **(~pin#9)**

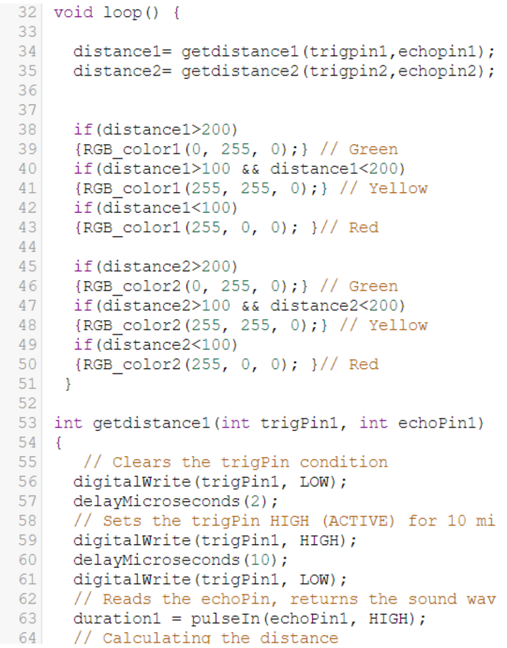
    d. **Cathode** -> **GND** terminal

- Now, repeat these connections for the remaining components (HC-SR04 & RGB led)

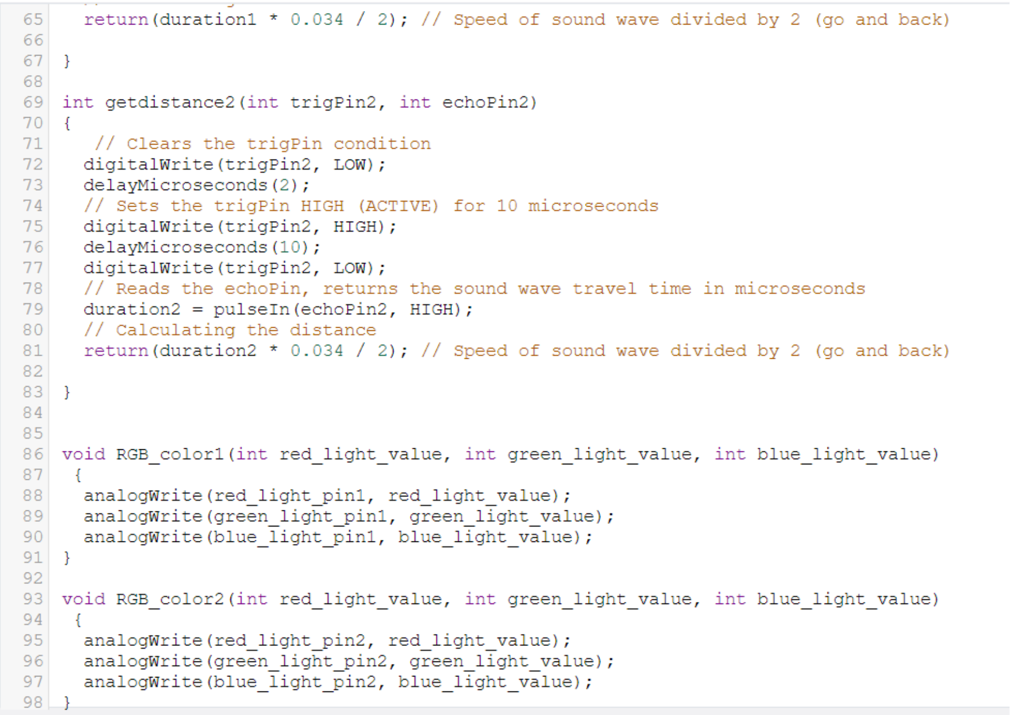
- <u>Code for the schematic is as follows:</u>



```
long duration1,duration2;
int distance1,distance2;

int red_light_pin1= 11;
int green_light_pin1 = 9;
int blue_light_pin1 = 10;

int trigpin1=4,echopin1=2;
int trigpin2=8,echopin2=7;

int red_light_pin2= 6;
int green_light_pin2 = 3;
int blue_light_pin2 = 5;



void setup() {

  pinMode(trigpin1, OUTPUT);
  pinMode(echopin1, INPUT);
  pinMode(red_light_pin1, OUTPUT);
  pinMode(green_light_pin1, OUTPUT);
  pinMode(blue_light_pin1, OUTPUT);

  pinMode(trigpin2, OUTPUT);
  pinMode(echopin2, INPUT);
  pinMode(red_light_pin2, OUTPUT);
  pinMode(green_light_pin2, OUTPUT);
  pinMode(blue_light_pin2, OUTPUT);

}
void loop() {
```



```
void loop() {

  distance1= getdistance1(trigpin1,echopin1);
  distance2= getdistance2(trigpin2,echopin2);


  if(distance1>200)
  {RGB_color1(0, 255, 0);} // Green
  if(distance1>100 && distance1<200)
  {RGB_color1(255, 255, 0);} // Yellow
  if(distance1<100)
  {RGB_color1(255, 0, 0); }// Red

  if(distance2>200)
  {RGB_color2(0, 255, 0);} // Green
  if(distance2>100 && distance2<200)
  {RGB_color2(255, 255, 0);} // Yellow
  if(distance2<100)
  {RGB_color2(255, 0, 0); }// Red
 }

int getdistance1(int trigPin1, int echoPin1)
{
   // Clears the trigPin condition
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 mi
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  // Reads the echoPin, returns the sound wav
  duration1 = pulseIn(echoPin1, HIGH);
  // Calculating the distance
```



```
  return(duration1 * 0.034 / 2); // Speed of sound wave divided by 2 (go and back)

}

int getdistance2(int trigPin2, int echoPin2)
{
   // Clears the trigPin condition
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration2 = pulseIn(echoPin2, HIGH);
  // Calculating the distance
  return(duration2 * 0.034 / 2); // Speed of sound wave divided by 2 (go and back)

}


void RGB_color1(int red_light_value, int green_light_value, int blue_light_value)
 {
  analogWrite(red_light_pin1, red_light_value);
  analogWrite(green_light_pin1, green_light_value);
  analogWrite(blue_light_pin1, blue_light_value);
}

void RGB_color2(int red_light_value, int green_light_value, int blue_light_value)
 {
  analogWrite(red_light_pin2, red_light_value);
  analogWrite(green_light_pin2, green_light_value);
  analogWrite(blue_light_pin2, blue_light_value);
}
```