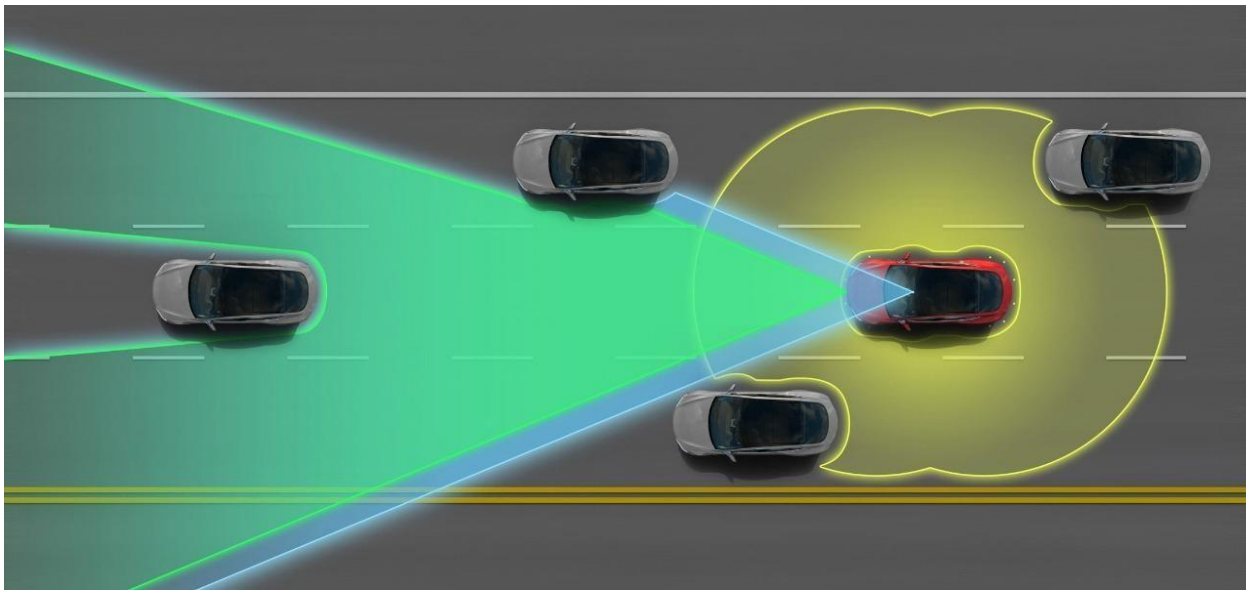
CHARAN SINGH | B21ES008
Indian Institute of Technology, Jodhpur

Clearing Blind spots Using Ultrasonic sound sensors



Source:https://medium.com/self-driving-cars/tesla-enhanced-autopilot-overview-l2-self-driving-hw2-54f09fed11f

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.

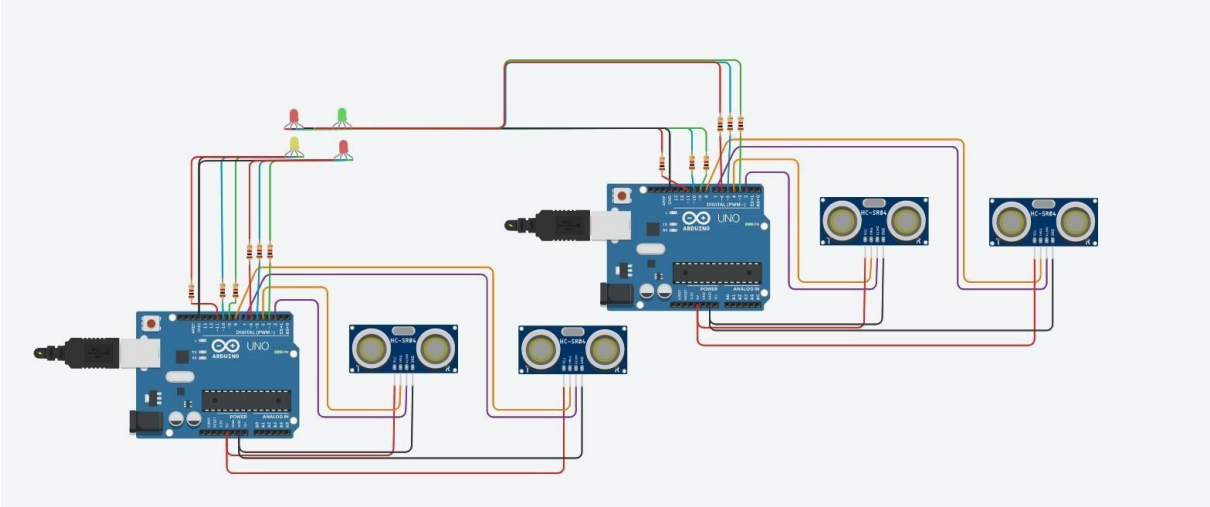
- It is often the poor, especially male road-users of working age that constitute the vulnerable road users (VRU) in India where VRUs shared road space with other less vulnerable users with their income level having a direct bearing on the mode of transportation used and resultant risk faced by them on that account.

Summary of work done:

- We focused on ideation, analysis, data gathering , collection of information and modeling of our proposed solution.We had to collect data on existing technology and how it can be improved.

- To visualize our idea in 3D, we used the blender programme to create 3D models. We divided the group and formed a modeling team. We did modeling of ultrasound sensors, animation of connecting all sensors and avoiding collision.

- We designed a circuit model which consisted of Arduino Uno R-3, 4 ultrasonic sensors, 4-RGB LEDs, resistors, breadboard, and connecting wires



- Arduino UNO R3, acting as an intermediate connection between ultrasonic sound sensors and RGB LEDs

Analysis of Design:

The various hardware components: Arduino Uno R-3(1), Connecting jumper wires(39), A small Breadboard(1),Ultrasonic sound sensors(4), A cardboard box with 4 openings, Arduino to PC USB connector cable, A computer.

Software component:

- Tkinter library, Arduino IDE, Python.
- Data from the sensors is analyzed in the Arduino IDE to Calculate distances of the objects, COM port 4 is used for this communication. The serial plotter feature in the arduino IDE can be used to ensure that the sensors are working properly.
- Serial library is used to import the data from the arduino to the Python IDE.Using the Tkinter library a Graphical User Interface is Created.
- On the GUI, 4 dots representing the obstacles' distances from the vehicle are programmed. If an object is far from the vehicle then the dot is of Green color, when it is at an intermediate distance the color is yellow and when the object or a nearby vehicle dangerously close to the vehicle then the dot is red.

Working :

- A cardboard box has 4 openings in it, this signifies a car with the 4 sensors placed appropriately on the car so as to clear the blind-spots of the driver.
- The sensors are ultrasonic sound sensors which use ultrasonic sound waves to determine the distance of an object placed before them. The beam angle is about 10-15 degrees and the range of distance is about 2 cm -400 cm.
- Each sensor has 4 pins, Trig, Echo, Vcc and GND. Trig pin corresponds to the transmitter which transmits the ultrasonic pulse . Echo pin corresponds to the Receiver, which receives the reflected ultrasonic wave. The Time delay between the transmission and receiving of the pulse depends on the distance between the sensor and the obstacles. The Vcc pin of all 4 sensors is connected to the 5v supply on the arduino chip using the breadboard. Similarly, the GND pin of each sensor is connected to the ground of the arduino. The other two pins are connected to any two of the 13 Digital I/O pins on the microprocessor

CODE (tkinter and python)

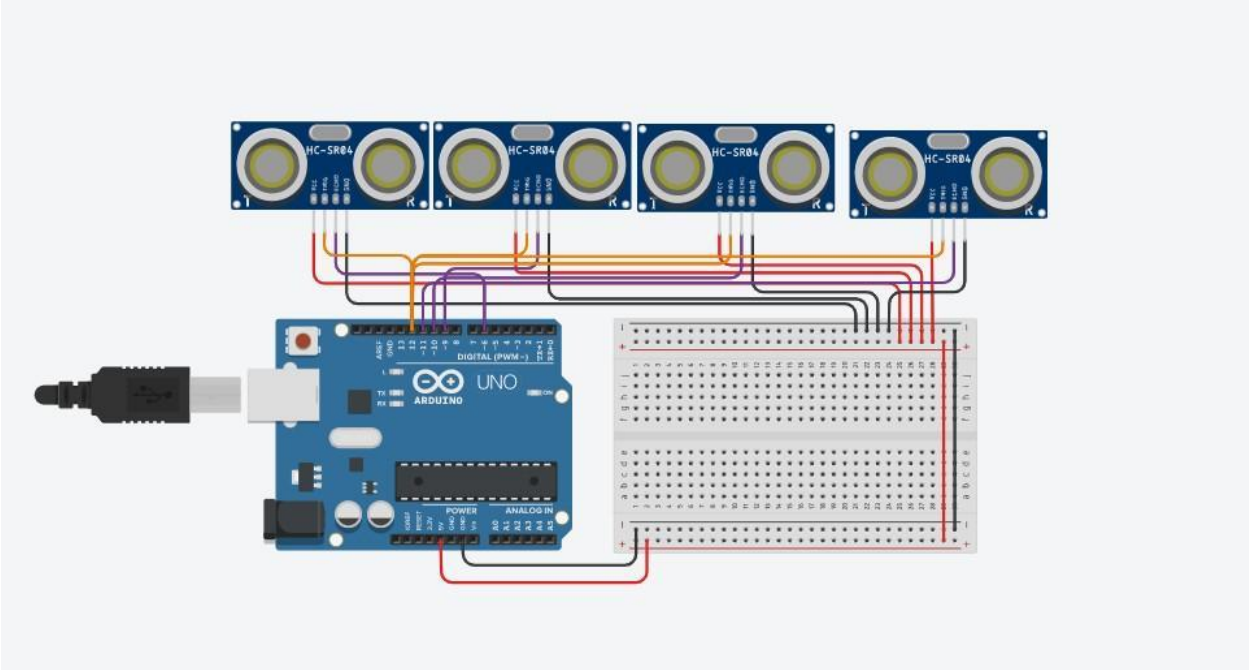
```
import tkinter
import time

import serial

h = 500
w = 1000

def create_animation_window():
    window = tkinter.Tk()
    window.title("Accident Prevention GUI")
    window.geometry(f'{w}x{h}')
    return window


def create_animation_canvas(window):
    canvas = tkinter.Canvas (window)
    canvas.pack(fill="both", expand=True)
    return canvas


def animate(window, canvas):
    colour = ['','','','']
    dist = [0, 0, 0, 0]
    c = 0


    rect_width = 166
    rect_height = 300
    rect_thick = 5


    rect_x0 = (w / 2) - (rect_width / 2)
    rect_y0 = (h / 2) - (rect_height / 2)
    rect_x1 = (w / 2) - (rect_width / 2)
    rect_y1 = (h / 2) - (rect_height / 2)

    image = tkinter. PhotoImage(file="/vehicle.png")
    label = tkinter.Label(animation_window_image=image)
    label.place(x=rect_x0, y=rect_y0)

    X0=rect_x0
    y0=rect_y0
```

```
    x1=rect_x0
    y1=y0+30
    #

    oval1 = None
    oval2 = None
    oval3 = None
    oval4 = None
    vehicle = None

    while True:

        if (dist[0] <= 0):
            c = 3

        if (dist[0] >= 348):
            c = -3

    window.update()

    for i in range(4):
        if (dist[1] >= 200):
            colour[i] = 'green'
        elif (dist[i] < 200 and dist[i] >= 100):
            colour[i] = 'yellow'
        elif (dist[i] < 100):
            colour[i] = 'red'
    W=tkinter.Label(animation_window_text=dist[0])
    W.pack()

    #vehicle = canvas.create_rectangle(rect_x0,rect_ye, rect_x1, rect_y1
rect x1, rect y1, width=rect_thick)

    oval1 = canvas.create_oval(rect_x0 - 30 - dist[0], rect_y0, rect_x0 -
dist[0], rect_y0 + 30, fill=colour[0])
    oval2 = canvas.create_oval(rect_x1 + dist[1], rect_ye, rect_x1 + dist[1]
+ 30, rect_y0 + 30, fill=colour[1])
    oval3 = canvas.create_oval(rect_x0 - 30 dist[2], rect_y1 30, rect_x0 -
dist[2], rect_y1, fill=colour [2])
    oval4 = canvas.create_oval(rect_x1 + dist[3], rect_y1 30, rect_x1 +
dist[3] + 30, rect_y1, fill=colour [3])

    for i in range(4):
        dist[i] += c
```

```
animation_window= create_animation_window()
icon = tkinter.PhotoImage(file="/vehicle.png")
animation_window.iconphoto(True, icon)
animation_canvas = create_animation_canvas(animation_window)
animate(animation_window, animation_canvas
```

Modeling and simulation

https://www.tinkercad.com/things/desTyelwnWf-serial-communication-attempt/edit el?sharecode=NzNrlqZZhKk1uK5iBHZDVh_5k9bap95PaLXEoZBweaU



ARDUINO CODE

```
// C++ code
long distance1, distance2, distance3, distance4;
int trigpin1=12,trigpin2=12,trigpin3=12, trigpin4=12;
int echopin1=11, echopin2=10, echopin3=9, echopin4=6;

// declaring all pins

int duration;

// declaring durations

void setup()
```

```
{
  pinMode (trigpin1, OUTPUT);
  pinMode (echopin1, INPUT);
  pinMode (trigpin2, OUTPUT);
  pinMode (echopin2, INPUT);
  pinMode (trigpin3, OUTPUT);
  pinMode (echopin3, INPUT);
  pinMode (trigpin4, OUTPUT);
  pinMode (echopin4, INPUT);
  Serial.begin(9600);

  // for beigining of the connections
}



void loop()
{

  //getting details from the getdistance() functions

  distance1=getdistance (trigpin1, echopin1);
  distance2=getdistance (trigpin2, echopin2);
  distance3=getdistance (trigpin3, echopin3);
  distance4=getdistance (trigpin4, echopin4);

  Serial.print (String (distance1));
  Serial.print(" ");
  Serial.print (String (distance2));
  Serial.print(" ");
  Serial.print(String (distance3));
  Serial.print(" ");
  Serial.println(String (distance4));
  delay(1);

}

long getdistance (int trigPin, int echoPin)
{

  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  // low means - 0
```

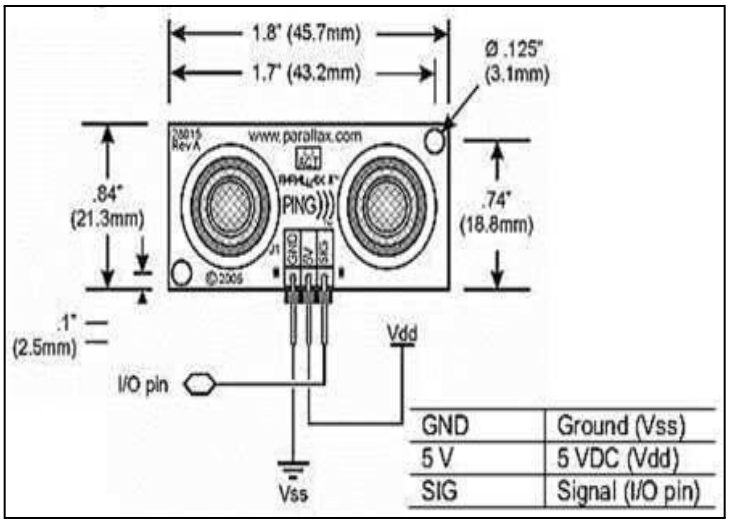
```
  delayMicroseconds (2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  // high means - 1

  delayMicroseconds (10);
  digitalWrite(trigPin, LOW);
  // Reads the echopin, returns the sound wave travel time in microseconds
duration = pulse In (echoPin, HIGH);
  // Calculating the distance
  return (duration * 0.034 / 2); // Speed of sound wave divided by 2 (go and
back)
}
```

Dimensioning and material selection

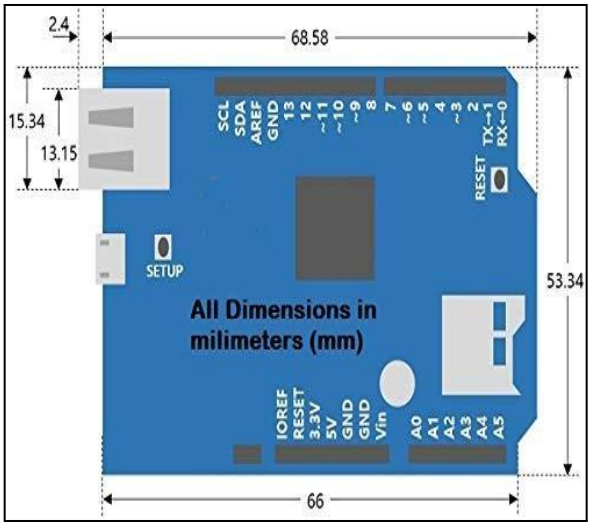
Dimensions of the cardboard box: 23cm by

23cm Dimensions of circuit components:

ultrasonic sensor



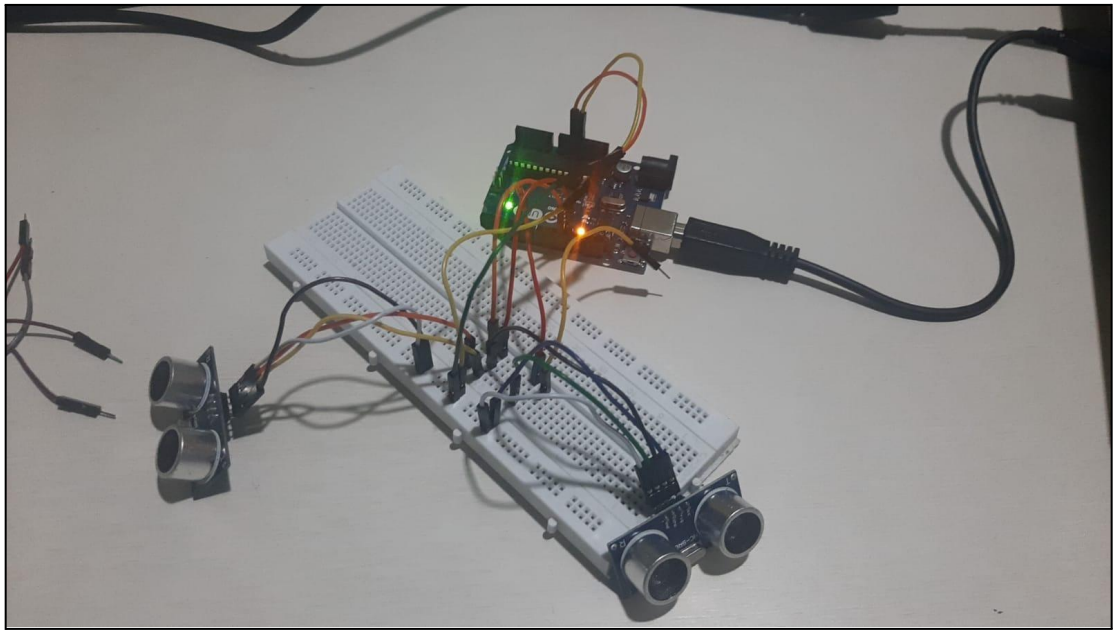
arduino



Prototyping andRefinement:

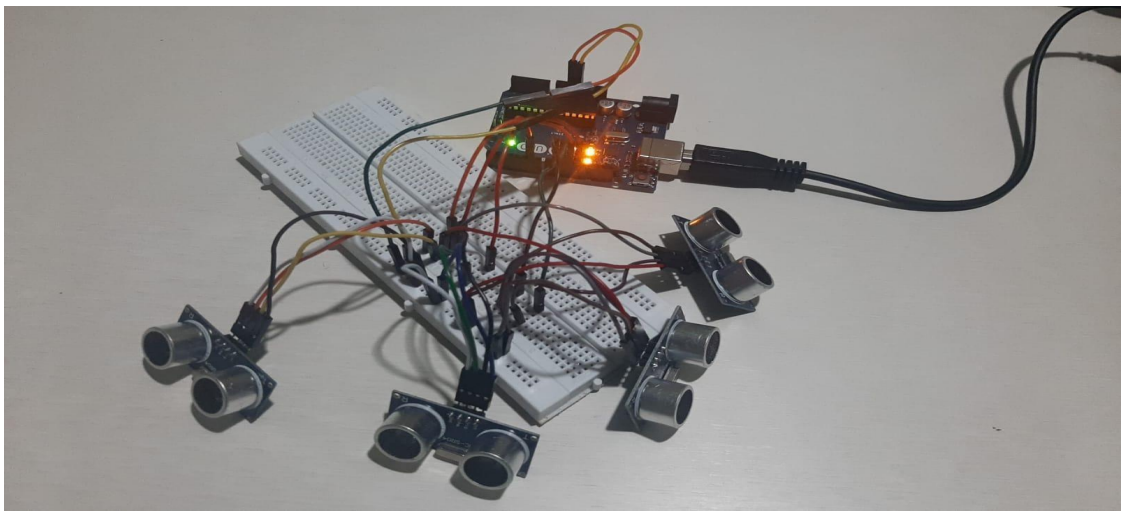
Prototype 1:

2 sensors connected loosely to the breadboard, with separate TrigPins (inefficient)



Prototype 2:

4 sensors connected loosely to the breadboard, with separate TrigPins (inefficient)



Integrating electrical and mechanical elements:

A box has 4 openings in it, this signifies a car with the 4 sensors placed appropriately on the car so as to clear the blind-spots of the driver.

The sensors are ultrasonic sound sensors which use ultrasonic sound waves to determine the distance of an object placed before them. The beam angle is about 10-15 degrees and the range of distance is about 2 cm -400 cm.

Each sensor has 4 pins, Trig, Echo, Vcc and GND. Trig pin corresponds to the transmitter which transmits the ultrasonic pulse . Echo pin corresponds to the Receiver, which receives the reflected ultrasonic wave. The Time delay between the transmission and receiving of the pulse depends on the distance between the sensor and the obstacles. The Vcc pin of all 4 sensors is connected to the 5v supply on the
arduino chip using the breadboard. Similarly, the GND pin of each sensor is connected to the ground of the arduino. The other two pins are connected to any two of the 13 Digital I/O pins on the microprocessor.

Demonstration of Final Design:

4 sensors fixed to the breadboard, with separate TrigPins (Lower Power Consumption).