Social Distancing Detector

1.1 Introduction:

1.1.1 Purpose:

This system provides an easy solution to detect distance between people and alert its users if the appropriate distance is not maintained.

1.1.2 Scope:

Today, social distancing has become a norm in any social interaction we have due to the increasing threat of the spread of Covid-19. There are many innovations that have been created to help combat and reduce the spread of the deadly disease, not just with other types of devices but using the power of Arduino as well. This can be seen in other projects across the website such as a Hand wash timer or an automatic hand sanitizer dispenser.

This project aims to help reduce the potential spread of the disease by utilizing an ultrasonic distance sensor. The sensor measures the distance between itself and the object in front of it, whether it be an object or a person. If something comes within 1.5m of the sensor, a buzzer sounds, and an LED lights up, therefore signalling that someone or something is within the range of social distancing.

This Arduino device can be used in real-life situations. For example, if you're in a queue at a coffee shop or at a train station. I have included a hyperlink for a simple 3D rendering of the proposed housing via TinkerCad. With the housing, you could clip this product on your belt or just hold it.

Overall, as individuals, we need to start adapting social distancing more and this is exactly what this device promotes through its visual and audio aids via the buzzer and LED.

1.2 Requirements:

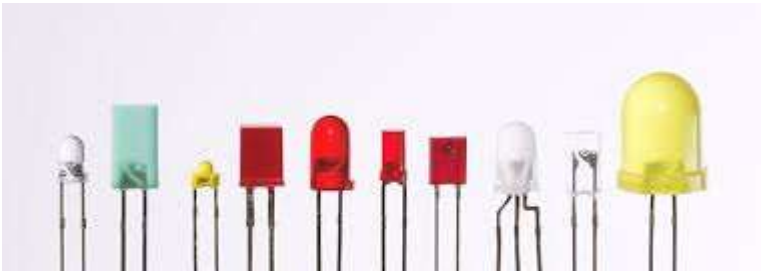
Hardware:

- **Arduino UNO**

  Arduino board is a microcontroller that is used to accept inputs from sensors connected and provide an output action on the desired device connected to it. The sensor inputs can be from light-detecting sensors, motion sensors (Ultrasonic or IR), temperature sensors, etc. The output from this device can be received through other output devices such as LED, Buzzer, Serial monitor, etc.

- **5mm LED: Red, Green**

  A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it.

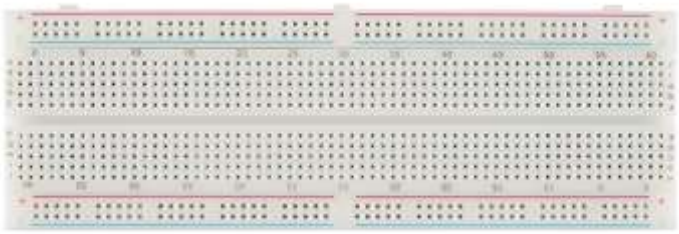


- ## Piezoelectric Buzzer

Piezo buzzers are simple devices that can generate basic beeps and tones. They work by using a **piezo** crystal, a special material that changes shape when voltage is applied to it. If the crystal pushes against a diaphragm, like a tiny speaker cone, it can generate a pressure wave which the human ear picks up as sound.

- ## Breadboard

A breadboard is the basic component of any circuit building process. All components, be it input sensors or output display devices are connected to the power supply, microcontroller using wired connections through a breadboard. The holes in the breadboard are in series. There are various sizes like full-sized, half-sized, and mini breadboard.



- ## Ultrasonic Sensor – HC – SR04

The **HC-SR04 ultrasonic sensor** uses **SONAR** to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1” to 13 feet.

- **Jumper Wires/Cables**

  These are the main components that are used to establish the connections between different devices of the circuit.

- **Resistors**

A **resistor** is a passive two terminal electrical component implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.

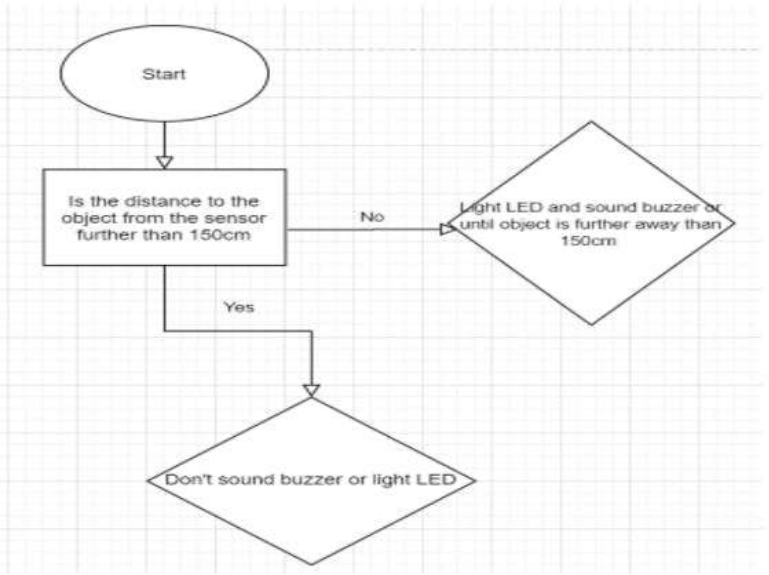
Software:

TinkerCad is a free, online 3D modelling program that runs in a web browser, known for its simplicity and ease of use. Since it became available in 2011 it has become a popular platform for creating models for 3D printing as well as an entry-level introduction to constructive solid geometry in schools. It can be used to construct various circuits

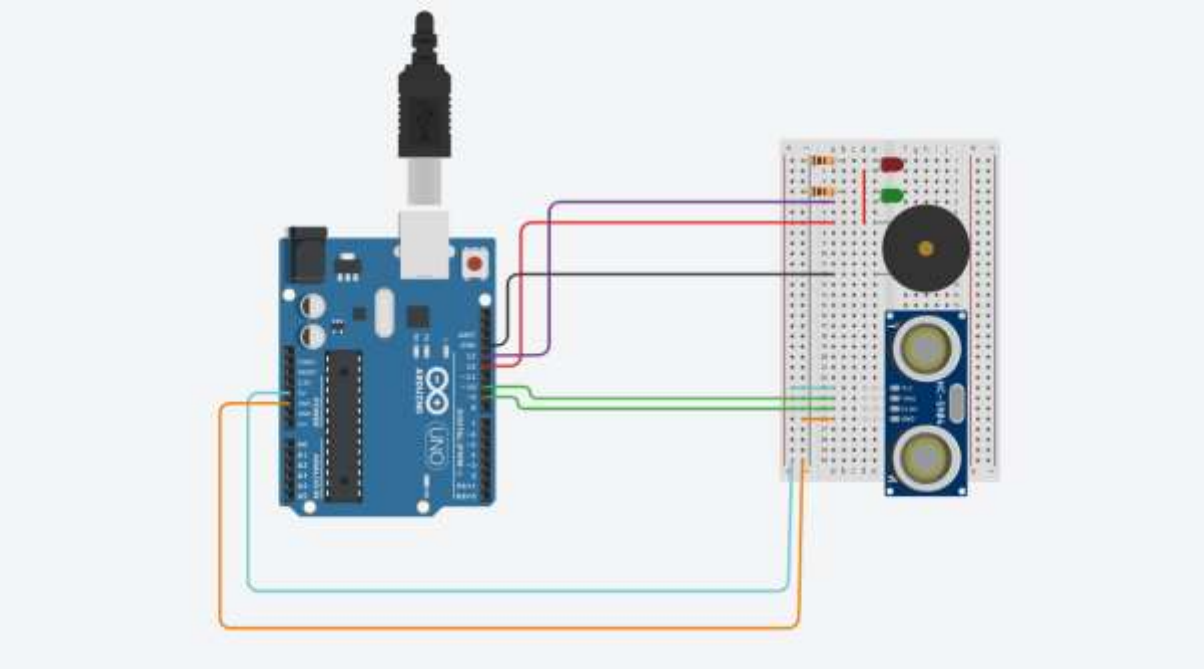
1.3 Pseudocode:

1. Set pin variable for the buzzer, led and ultrasonic sensor
2. Set variables for measurements, duration, and distance
3. Set the buzz, trig, led, echo pins to output or input
4. Sensor measures distance from itself to object
5. If distance > 150cm from object
6. Don’t beep buzzer/ don’t light LED
7. If distance < 150cm from object
8. Beep buzzer/ light LED
9. End

- **Code Flowchart:**



1.4 Circuit:



Explanation:

The circuit connections: Firstly, we need to connect one line of the breadboard to the ground and the other to the power supply. This is done by connecting the 5V pin of the Arduino Board to one line of connection pins on the breadboard. The other line of the breadboard is connected to the ground terminal of the Arduino Board. These lines will be connected to other devices.

The HC – SR04 Ultrasonic Distance sensor has 4 pins – VCC(power), TRIG(trigger), ECHO and GND(ground). The ground and power supply of sensor are connected to those of Arduino via breadboard.

The Piezoelectric Buzzer has two pins – positive, which is connected to Arduino pin 12 and negative, which is connected to Arduino ground.

The two LEDs and resistors are connected to the positive end of buzzer, and anode of the LED adjacent to buzzer is connected to Arduino pin 13.

1.5 Code:

```
const int trigPin = 9; //pin for sensor, trig pin transmits signal to object in front of it
const int echoPin = 10; //pin for sensor, echo pin receives the signal the object reflects
and therefore the distance is measured
const int ledPin = 13; //led pin on arduino
const int buzzPin = 2; //buzzer pin on arduino
long duration; //distance variable
int distance;
void setup()
{
Serial.begin(9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(ledPin, OUTPUT);
pinMode(buzzPin, OUTPUT);
}
void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH); //trigpin's output pulse
delayMicroseconds(1000);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH); //calibrate echo pin's pulse input
distance= duration*0.034/2; // distanceCm is half the duration multiplied by 0.034
Serial.println("Distance: "); //print to serial monitor
Serial.println(distance); //print distance in Cm to the serial monitor
delay(500); //so you don't get overloaded by serial monitor outputs


if (distance <= 150 && distance >= 0) { //if distance from sensor to object is less than
150cm and more than 0

digitalWrite(ledPin, HIGH); //light led
digitalWrite(buzzPin, HIGH); //buzz
tone(2, 500, 120); //so we can hear a sound from the buzzer that isn't inaudible
//buzz and light LED
} else {
//dont buzz or light LED
digitalWrite(ledPin, LOW); //don't light led
digitalWrite(buzzPin, LOW); //don't buzz
}
}
```

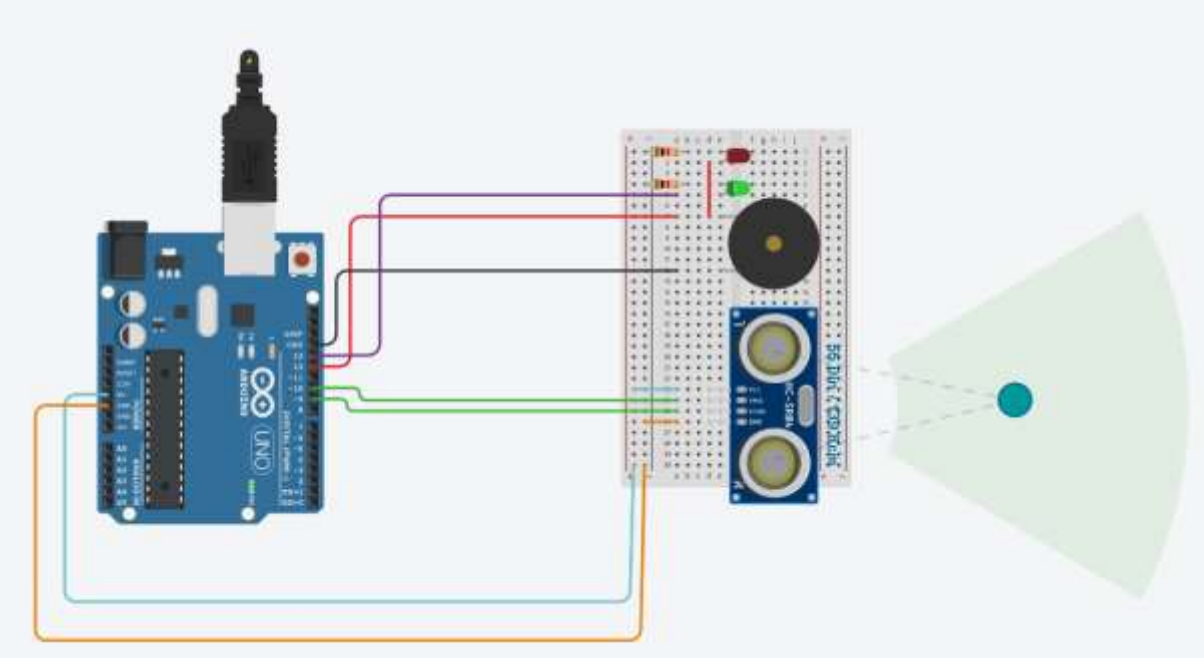
Working:

- The circuit uses an ultrasonic distance sensor to achieve its goal. As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves.
- The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.
- The distance can be calculated with the following formula:

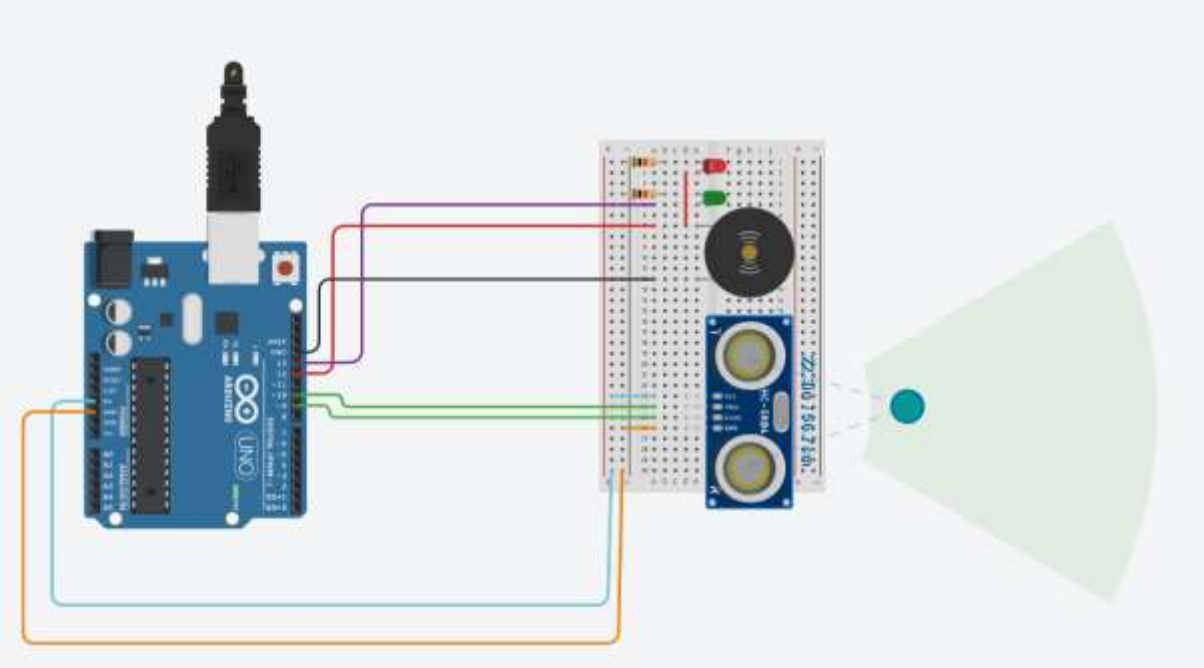
  Distance L = 1/2 × T × C

  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.)
- So a condition is applied while coding, that if an object is in a certain range of the sensor, then it should light up the LED attached as well as buzz the piezoelectric buzzer in order to indicate that you are too close and need to maintain a certain distance.
- Arduino is a great tool for developing interactive objects, taking inputs from a variety of switches or sensors and controlling a variety of lights, motors and other outputs. So these functionalities make it useful for the circuit.
- This is the main principle applied to implement social distancing.

Images:



Green LED is lit up, and piezo is not buzzing, indicating you are at a safe distance.



Red LED is lit up, piezo is buzzing, indicating that you are too close and need to maintain some space.

Conclusion:

In this project we learnt how ultrasonic distancing sensor works and how it can be used with Arduino to create various useful circuits using TinkerCad.

BY:

Yash Maheta - 18IT403

Taksh Shah - 18IT411