

Microprocessors and Microcontrollers CEECCO8

Social Distancing Alert Assistant
using
Arduino Nano



Group 4

Tabishi Singh (2019UCO1506)

Aryaman Sharma (2019UCO1508)

Maanas Talwar (2019UCO1544)

Bhavya Nagpal (2019UCO1561)

Harshit Gupta (2019UCO1580)

Problem Statement

In the wake of COVID-19 that has gripped the entire world, it has become imperative to practice safety measures like social distancing. Available evidence indicates that the COVID-19 virus is transmitted during close contact through respiratory droplets and by fomites. There should be a minimum of one arm's length between 2 people and/or all surfaces to ensure that there is little to no contact.

Solution

According to [cdc.gov](https://www.cdc.gov), social distancing means keeping a safe space between yourself and other people. We need to stay at least 6 feet (about 2 arms' length) from other people who are not from your household in both indoor and outdoor spaces.

The project aims to fluidly incorporate social distancing, into a device that constantly scans the user's surroundings and alerts them of any object/person in their close vicinity. The alerts are in the form of a buzzing sound and a blinking light, which increases its frequency as the user comes closer to an object/ another person. At a distance of 160 cm, the LED's light up and signify that unsafe limits have been reached. Between 160 and 120 cm (approximately 4 feet), the buzzer starts sending signals to maintain a safe distance. Less than 80 cm (2.5 feet), the buzzer emits sound continuously to alert the user to stay away as far as possible. Hence, it helps make both the user as well as the person around them aware to have a hygienic interaction while maintaining the social distancing norms set by the government. This device will thus be a helpful aid for people in these dire circumstances.

Please check out the functioning and demonstration of our device on [YouTube](#).

Components Used

Hardware requirements:

- Arduino Nano board



- 1 10k Ω Resistor



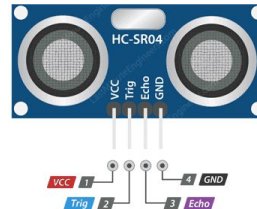
- 1 LED



- Robocraze active buzzer



- HC SR04 Ultrasonic sensor



- 1 9V battery



- Breadboard



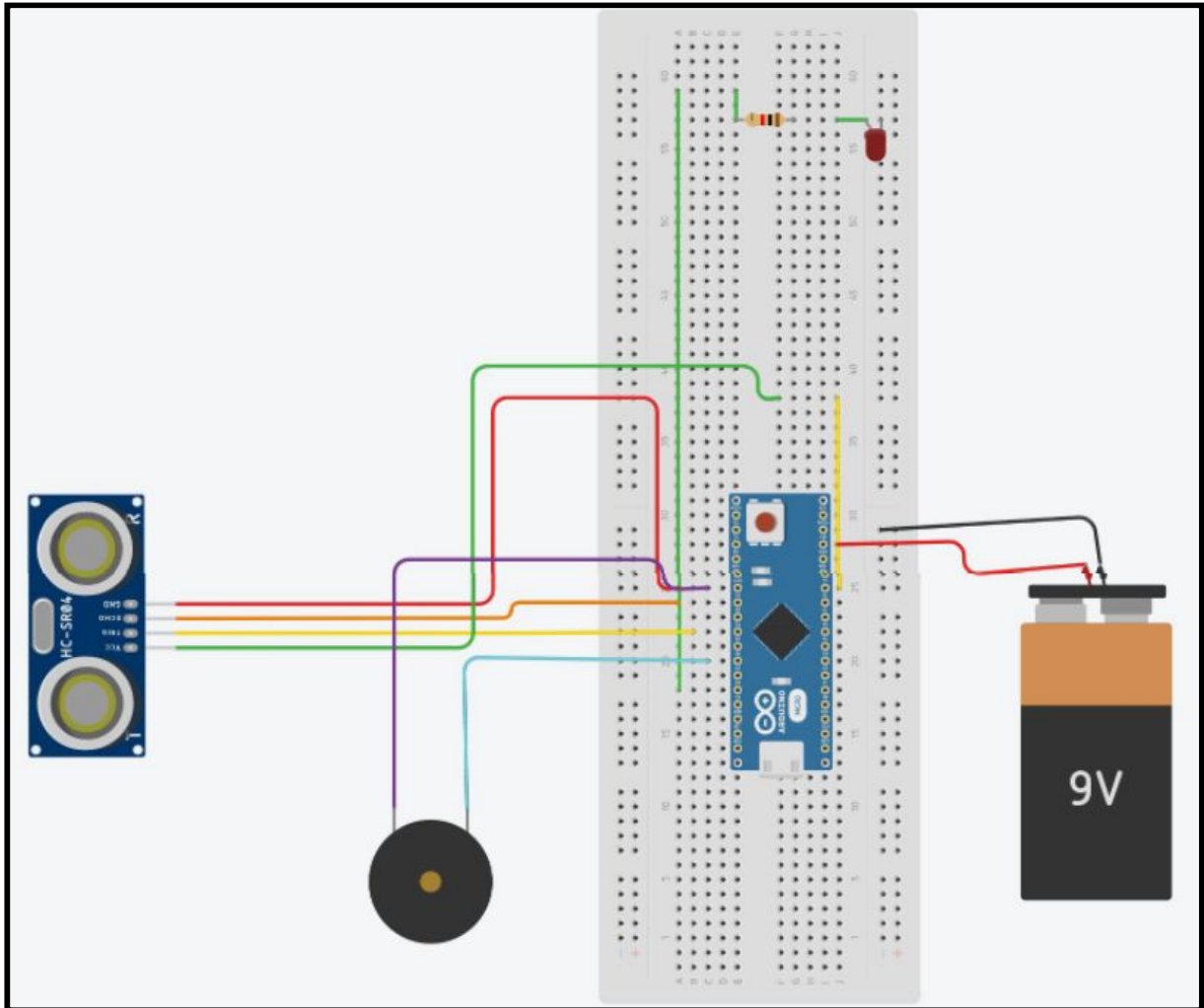
- Jumper cables



Software requirements:

- [Arduino IDE](#)

Circuit



HC SR04 Ultrasonic sensor:

Trig (Trigger) pin is kept high for $10\mu\text{s}$ to initialize measurement by sending an ultrasonic wave.

Echo pin goes high for a period of time which will be equal to the time taken for the ultrasonic wave to return back to the sensor.

Source Code

```
const int red = 8;
const int buzzer = 6;
const int trigger = 4; // the pin which gives a signal for
the sending of the ultrasound.
const int echo = 2; // the pin which receives the signal
after deflection from object

long duration, distance;
void setup() {
    Serial.begin(9600);
    pinMode(red, OUTPUT);
    pinMode(buzzer, OUTPUT);
    pinMode(echo, INPUT);
    pinMode(trigger, OUTPUT);
}

void loop() {
    digitalWrite(trigger, HIGH); //trigpin's output pulse
    delay(10); // waiting for the pulse to return
    digitalWrite(trigger, LOW);
    //calibrate echo pin's pulse input
    duration = pulseIn(echo, HIGH);
    // distanceCm is half the duration multiplied by 0.034
    distance = duration * 0.034 / 2;
    // cases for different distances
    if (distance >= 0 && distance < 80) {
        digitalWrite(red, HIGH);
        digitalWrite(buzzer, HIGH);
        delay(100);
        digitalWrite(buzzer, LOW);
        delay(100);

        digitalWrite(buzzer, HIGH);
        delay(100);
        digitalWrite(buzzer, LOW);
        delay(100);

        digitalWrite(buzzer, HIGH);
        delay(100);
```

```
    digitalWrite(buzzer, LOW);
    delay(100);
    digitalWrite(buzzer, HIGH);
    delay(100);
    digitalWrite(buzzer, LOW);
    delay(100);
    digitalWrite(buzzer, HIGH);
    delay(1000);

} else if (distance >= 80 && distance < 120) {
    digitalWrite(buzzer, LOW);
    digitalWrite(red, HIGH);
    delay(10);
    digitalWrite(buzzer, HIGH);
    digitalWrite(red, LOW);
    delay(200);

} else if (distance >= 120 && distance < 160) {
    digitalWrite(buzzer, HIGH);
    digitalWrite(red, HIGH);
    delay(300);
    digitalWrite(red, LOW);
    delay(400);
} else if (distance >= 160) {
    digitalWrite(buzzer, HIGH);
    digitalWrite(red, LOW);
}
}
```

Note: Please refer to our [GitHub repository](#) for further insights.