BlindSpot - Object Detection System for the Visually Impaired

Sahith Bodla
07/18/21

ABSTRACT

***Motivation*:**

This device provides the visually impaired with knowledge of their surroundings. Though the blind generally use canes to map out their path a few steps in front of themselves, this device corroborates some information gathered from touch using hearing. Just like how bats use echolocation to discover prey at night rather than vision, blind people tend to rely on sound to comprehend and acknowledge their surroundings when, for instance, crossing the street or walking through a mall.

***Criteria*:**

The device must detect the presence of objects in front of it and determine when or when not to signal the buzzer to beep based on a condition or threshold. This threshold is defined by a certain distance in front of the device, which can be changed depending on how the user defines “nearby” (by default, the buzzer will beep for objects 0.5 meters and closer).

***Hardware*:**

An ultrasonic wave sensor, which both transmits and receives ultrasonic waves (sounds that are of a frequency that human ears can’t perceive), is used. An active buzzer is used to signal the user. Both the sensor and the buzzer are connected to the Arduino Uno via pins. The ultrasonic sensor continually transmits and receives waves. The wavelength of the returning wave is used to calculate the distance it traveled, which is then evaluated by a conditional. If the conditions determine the distance to be under 0.5 meters, the buzzer is then activated, resulting in a loud noise. If the distance exceeds 0.5 meters, no sound is made.

***Conclusion*:**

The device worked as intended, producing noise for objects 0.5 meters in distance as well as recording and letting the user know the number of times an object obstructed the ultrasonic detector within a minute via buzzer noise. The sensor, however, was limited to a certain FOV, where it would only detect the presence of objects directly in front of it.

INTRODUCTION AND BACKGROUND

Idea 1

Bats are nocturnal creatures who hunt for food during the night, despite having poor eyesight. The reason for their survival is their echolocation, where they produce ultrasound waves (sounds higher than what humans can hear) that bounce off nearby objects, revealing to a bat if food or obstacles are nearby, kind of like a radar [1]. Though most humans don't do this since they can see well, people afflicted by blindness or low vision require using other senses for navigation, especially hearing.

The following design is purposed to help those suffering from blindness or vision loss by creating a similar system to what is used in echolocation. Essentially, an ultrasonic sensor is used to transmit a wave, which will be received if the transmission rebounds in the same direction. In the case that the receiver notices an ultrasonic transmission, the passive buzzer generates a sound, signifying to the user that there may be a potential obstruction in their path or nearby vicinity. While this sensor would be useful in identifying the presence of objects nearby, it has tunnel vision, meaning it only buzzes for objects within the narrow line of sight. Furthermore, external ultrasonic signals from other devices in the surroundings (dog whistles, for instance), may throw off the sensor and provide the user false information.

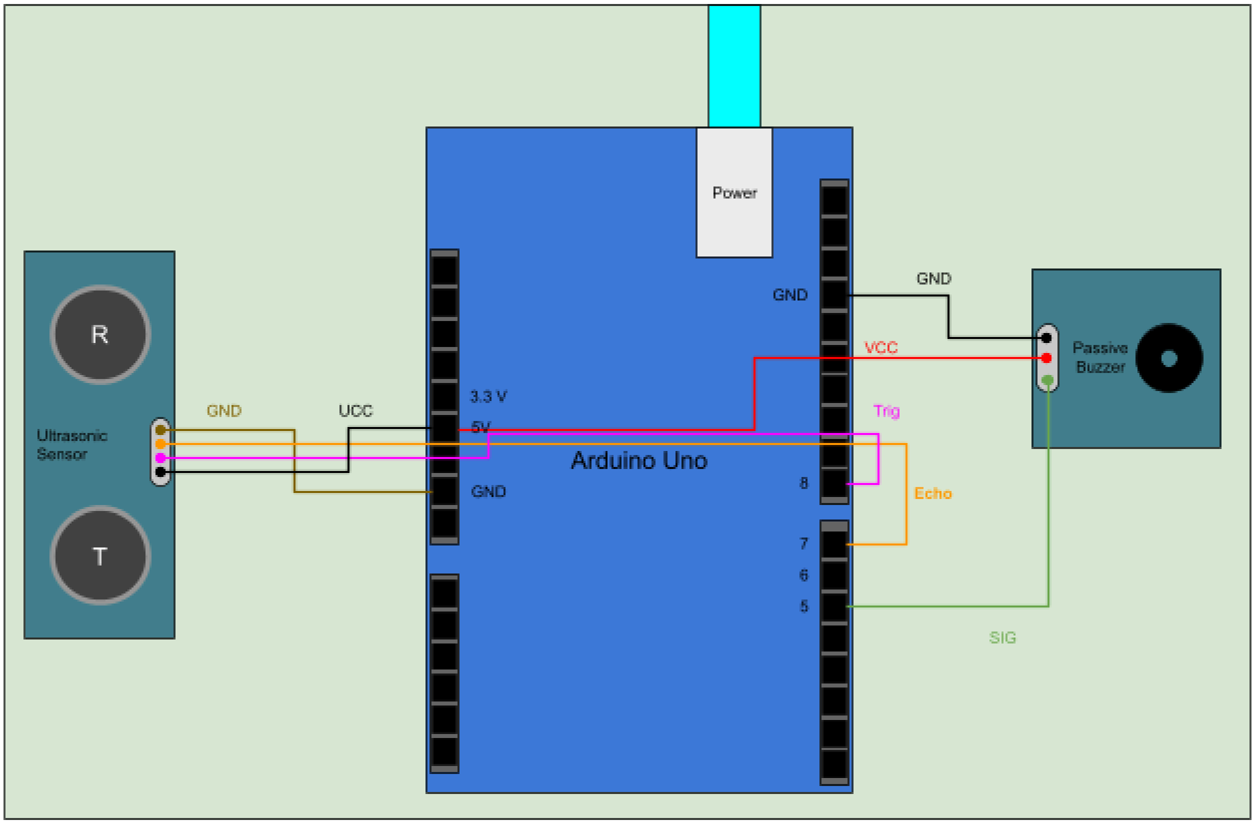


Figure 1.1 Ultrasonic Detector Schematic: Hooks up an ultrasonic transmitter to the Arduino that also receives rebounding ultrasonic waves. The passive buzzer then is coded to produce a sound depending on a certain threshold, while tells the user whether there is an object at a certain distance in their forward path.

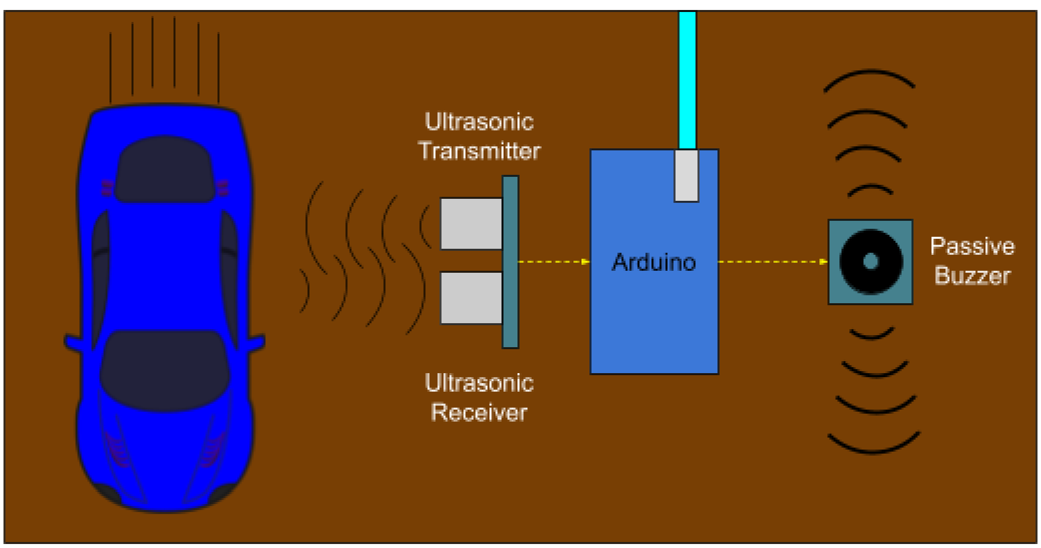


Figure 1.2 Ultrasonic Detector Functionality (birds-eye view)): The ultrasonic chip sends an ultrasonic wave that rebounds off a close obstruction (moving car). The wave is then observed by the receiver, resulting in a signal sent to the Arduino. This signal is then analyzed by code to determine whether the signal is close enough for a buzzer signal. The conditional will then tell the passive buzzer whether or to produce a sound or not.

Idea 2

In many medical treatments, lasers are used to either heal or shape certain tissues. For instance, in LASIK, or laser eye surgery, a laser is used to shave off parts of the cornea or the eye's outer layer. It improves eyesight and vision by reshaping the cornea, so myopic, hyperopic, and astigmatic people no longer need to wear glasses [2]. Furthermore, they assist in cancer treatment, skin treatment, and cosmetics [3].

Many consumer lasers are found in stores as pointers, having a pushbutton or switch to activate or turn off the light. Though lasers are very precise and last a long distance before the wavelength shifts into infrared, they have a narrow exposure radius and are harmful and blinding when shone on unprotected eyes.

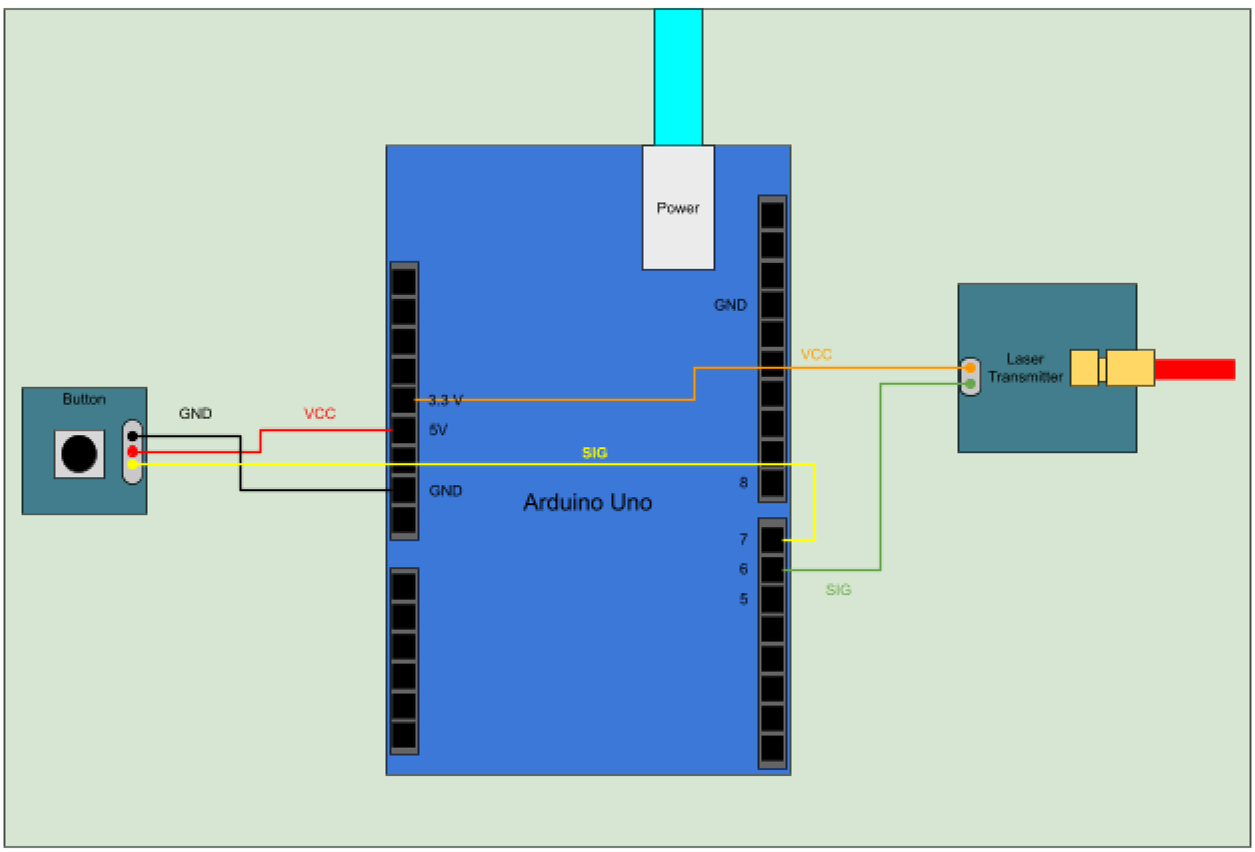


Figure 2.1 Laser Transmitter Schematic: The button will be set up to display a value of 0 when unpressed and 1 when pressed. If the reading is 1, then the code will tell the laser transmitter to turn on, resulting in an emission, and vice versa.

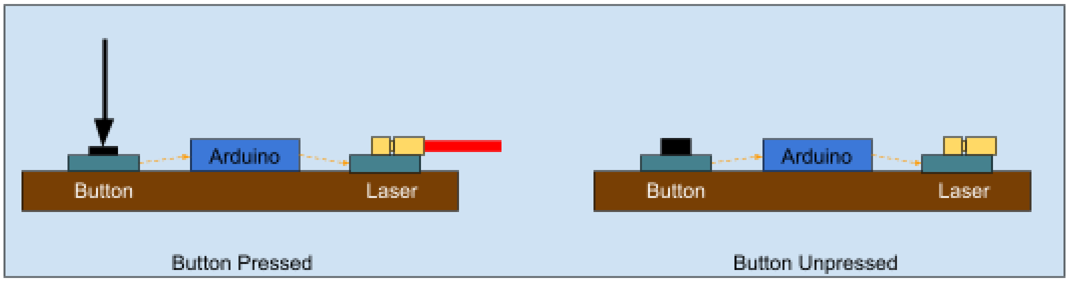


Figure 2.2 Laser Transmitter Functionality (frontal view): The button press will result in the laser being emitted.

Idea 3

Many hospital rooms have to keep a stable and sterile ambiance to prevent bedded patients from feeling uncomfortable. Things like room temperature, ventilation, brightness, and humidity have to be regulated to keep a patient in good health while getting treatment in ICUs for example. Humidifiers, specifically, are used to maintain a certain range of humidity because they can help alleviate cold and flu symptoms, improve sleep, hydrate the body, soothe itchy eyes, and relieve dry skin [4].

This design will be used to display the humidity of the room that it is in. In the case that humidity is above an acceptable level, then an LED will flash red. In the case that humidity is below an acceptable level, then an LED will flash green. This device will resemble a humidifier used in hospitals or the home.

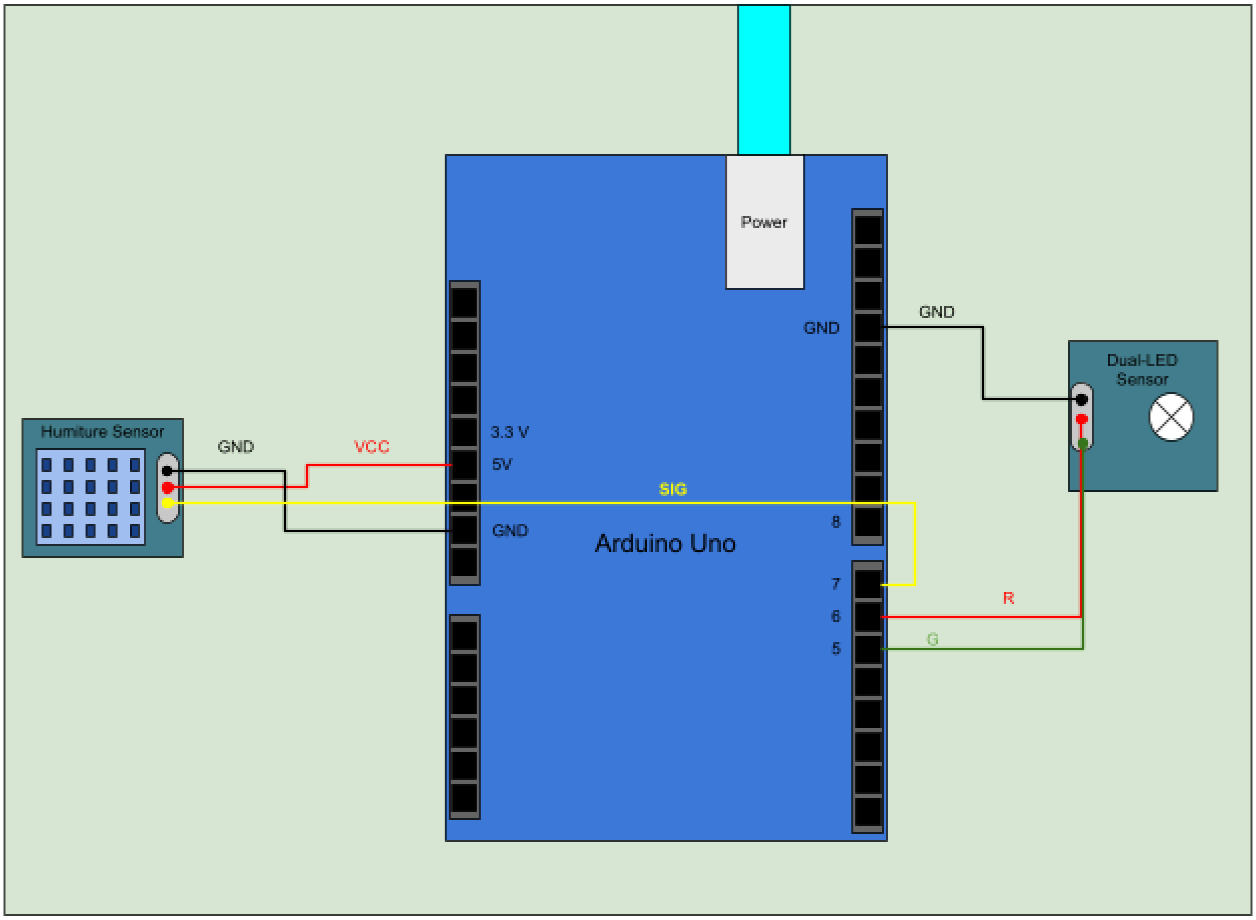


Figure 3.1 Humidity Sensor: The humiture sensor will be used to record both the room temperature and humidity. However, only humidity will be used for this design. The recorded humanity % will be evaluated by a conditional under a certain humidity threshold. If exceeded, the LED flashes red, and if beneath, the LED flashes green.

Chosen Design

I decided to construct Idea 1 because the concept seems like a very good complement to the visual impairment canes that blind people use. Furthermore, the auditory feedback is quite helpful in allowing users to visualize their close surroundings, as the loud, high-pitched sound produced by the buzzer is distinguishable from many other background noises in a street or at a store, for example. The design is also flexible to the user, where the threshold can be changed for the buzzer to identify objects at further or closer distances. Overall, I think this design would be a very useful biosensor for the visually impaired.

***Critical Factors**:*

This device must be able to sense the presence of objects in front of the ultrasonic wave detector by capturing rebounded waves that it first emitted. It must also be defined what a nearby object is, meaning that a threshold must be set up where any value underneath would result in a buzz while nay value above would be too far to be considered within the user's vicinity. Thus, the passive buzzer must be able to produce a sound depending on the threshold given, acting as a feedback mechanism to the user about their surroundings. This design can be tested by waving my hand near and in front of the ultrasonic wave detector. If the design is properly built, then the buzzer should create noise when my hand obstructs the two cylinders on the ultrasonic chip. In a realistic scenario, this would assist a visually impaired use when walking on a sidewalk in a residential neighborhood, where the device would beep when a person approaches them from the front or when a car is close to an intersection.

METHODS (Sensing and Measurement)

***Hardware**:*

The ultrasonic wave detector requires being plugged into four pins: ground (brown), echo (white), trig (red), and power (black). The ground (Gnd) pin can be placed in any one of the three GND slots on the Arduino. The echo pin, which receives ultrasonic waves, is placed in slot 7. The trig pin, which emits ultrasonic waves, is placed in slot 8. The power (Vcc) pin is placed in the 5V slot.

An active buzzer was used instead of a passive buzzer when constructing this design. This buzzer requires being plugged into three pins: ground (black), power (red), and signal (yellow). The ground (GND) pin can be placed in any one of the three GND slots on the Arduino. The power (VCC) pin is placed in the 3.3 V slot. The signal (SIG) pin, which tells the buzzer to produce sound, is placed in slot 5.

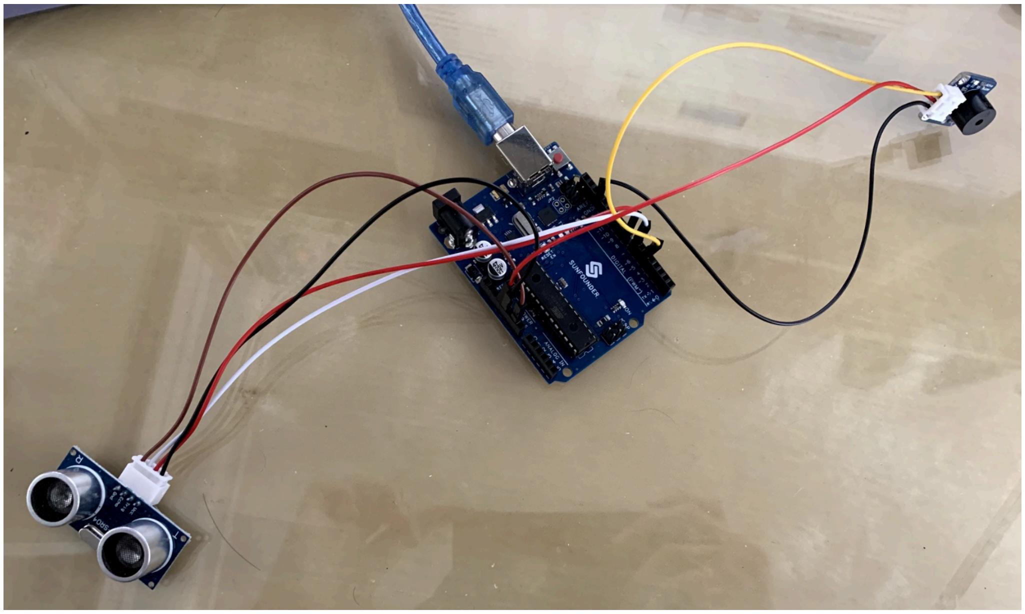


Figure 4.1 Picture of Setup: View of all hardware parts and wiring in the design (Left is ultrasonic wave sensor, Middle is Arduino, and Right is active buzzer).

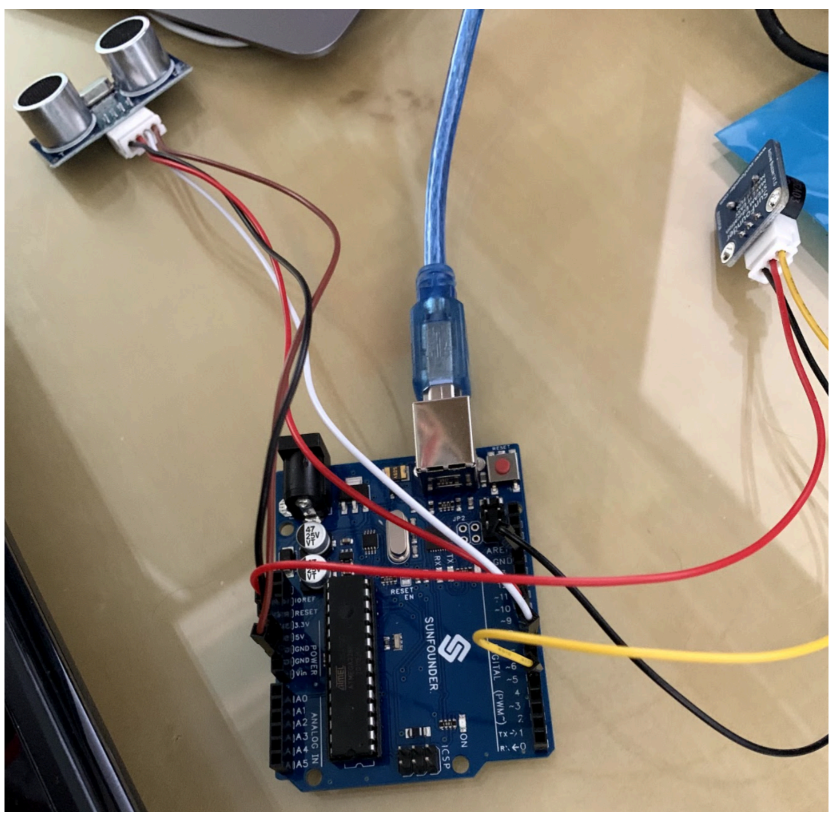


Figure 4.2 Closer View of Setup: Focus implemented on the Arduino and wire placements.

Software:

Ultrasonic

```
/*
  This program uses ultrasonic waves to detect objects 0.5 meters in front of the user
  using a buzzer for feedback
*/

const int trigpin = 8; // the wave transmitter pin attaches to pin 8
const int echopin = 7; // the receiver pin attaches to pin 7
const int buzzpin = 5; // the active buzzer SNG pin attaches to pin 5
int n = 600; // the time taken to complete a minute of ultrasonic wave recordings
int count = 0; // the number of times an object obstructed the ultrasonic sensor

int pulse; // the incoming ultrasonic wave after being transmitted
int distance; // the calculated distance threshold for buzzer activation

void setup() {
  // specifying baud rate
  Serial.begin(9600);

  // set up the pins for the ultrasonic waves trig for sending waves and echo for receiving waves)
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);

  // set up pins for sending signals to the buzzer
  pinMode(buzzpin, OUTPUT);
}

void loop() {
  for (int j = 0; j < n; j++) {
    // tells the ultrasonic chip to transmit a wave
    digitalWrite(trigpin, HIGH);
    // continue transmission for 5 milliseconds
    delay(5);
    // end transmission
    digitalWrite(trigpin, LOW);

    // method "pulseIn" comes from [5]
    pulse = pulseIn(echopin, HIGH);

     // formula for distance comes from [5]
    distance = (pulse / 2) / 29.1;

    // if the objects is within 0.5 meters of the ultrasonic chip, the buzzer will create a sound
    if (distance <= 50 && distance >= 0) {
      digitalWrite(buzzpin, LOW);
      count++; // adds to the number of objects that pass in front of the user
    }
    else {
      digitalWrite(buzzpin, HIGH);
    }

    // program takes a short break before taking the next detection
    delay(95);
  }

  // shows the number of objects passed 0.5 meters in front of the user in the last minute
  Serial.print("Number of objects passed in front: ");
  Serial.print(count);

  // the buzzer beeps the number of times an object passed in front in the last minute
  // (useful for people who can't see, and can be interchangeable for different periods)
  for (int k = 0; k < count; k++) {
    digitalWrite(buzzpin, HIGH);
    delay(50);
    digitalWrite(buzzpin, LOW);
    delay(300);
  }

  // resets counter for the next minute
  count = 0;
}
```

The above code describes how the design detects objects in front of it and how it responds to the presence of these objects. Essentially, the ultrasonic sensor continually sends out and receives ultrasonic waves. However, the buzzer will only make a sound if the object is within a certain threshold distance i.e. the object has to be within 0.5 meters of the field of view of the ultrasonic wave detector. Every time an object is sensed, a counter will go up by one to record how many times an object appears in front of the detector.

After a minute of detection, ultrasonic wave transmission stops and the buzzer starts beeping the number of times that there were an object 0.5 meters in front of the ultrasonic sensor. The display monitor also states how many objects appeared, but won't be visible to the typical user (who will most likely be a visually impaired or blind person).

Measurements:

The number of objects that appeared 0.5 meters in front of the ultrasonic wave transmitter will be recorded over a period of 60 seconds. The average or standard deviation values weren't calculated, however, since the device's practical application would not require these values; an average or a standard deviation would return a decimal value, which is inexpressible using a buzzer. Uncertainty doesn't apply to this design since the buzzer has only two outputs, either on or off.

Calculations:

Distance of object calculation (Lines 37-41) [5]:

$$\text{pulse} = 698.4 \text{ (measured)}$$

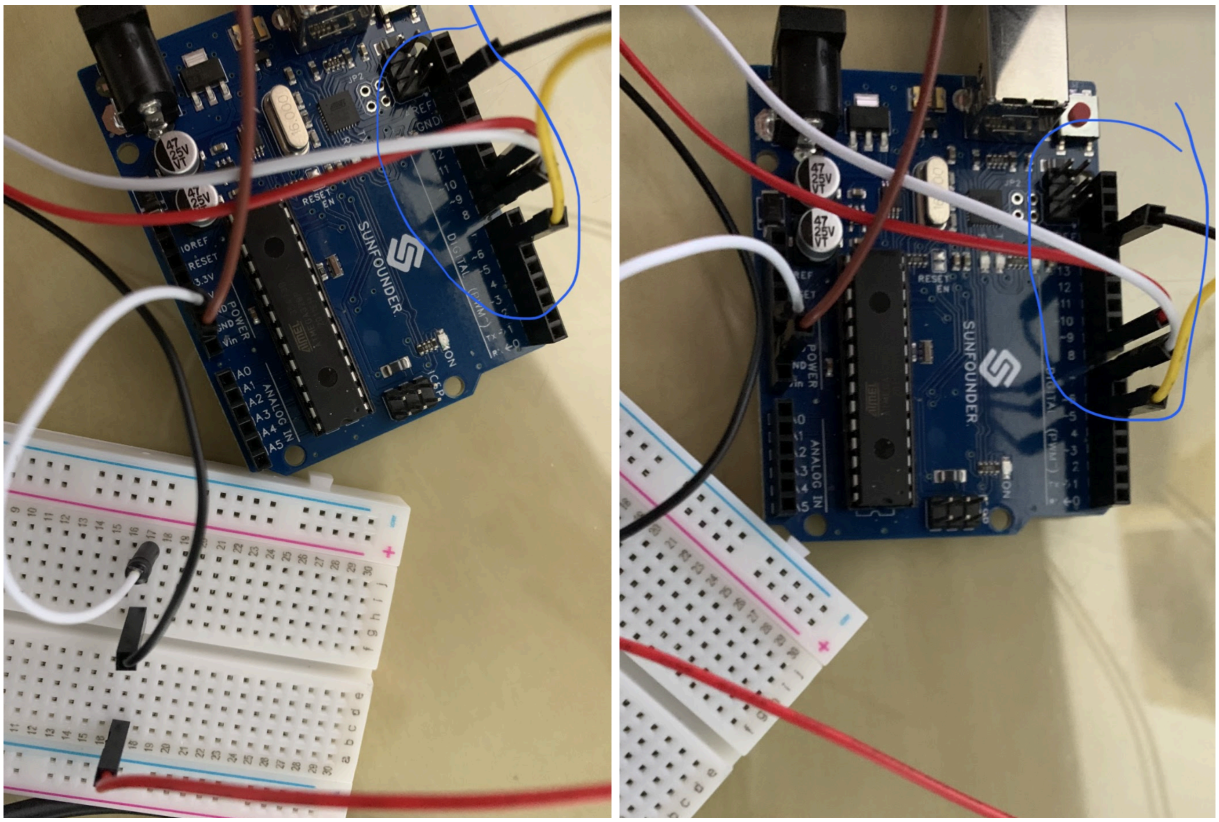
$$\text{distance} = (698.4 / 2) / 29.1 = 48 \text{ cm} = 0.48 \text{ m}$$

DISCUSSION

Modifications:

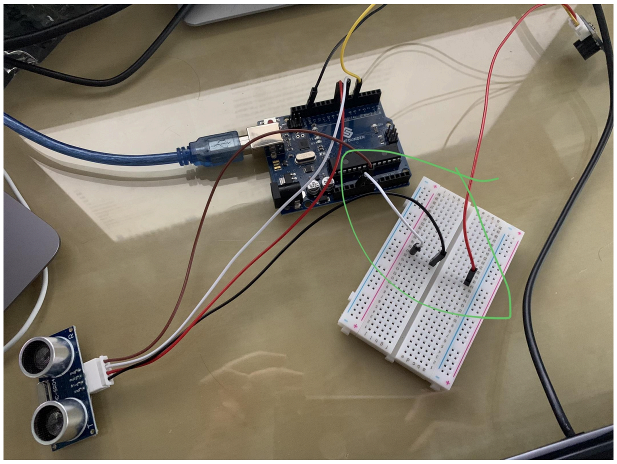
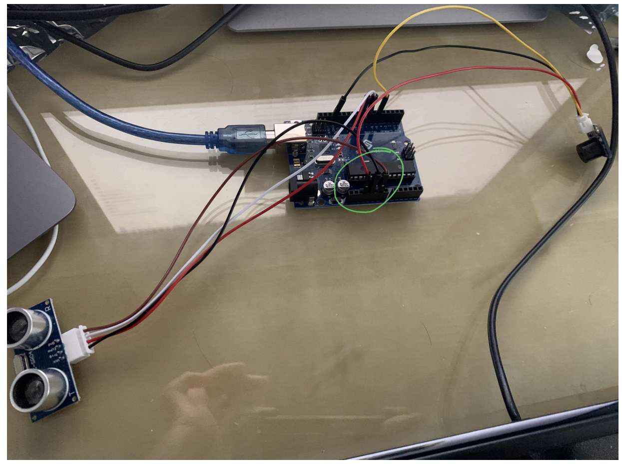
Day 1:

- I spent a lot of time figuring out why nothing worked but realized I simply put the pins in the wrong slots (they were placed one slot above the correct ones)



Day 2:

- I couldn't figure out why the breadboard wasn't allowing both power wires (one from the buzzer and the one from the ultrasonic sensor) to be attached to the same 5V slot. Therefore, I removed the breadboard entirely from the setup and instead hooked up the ultrasonic detector's VCC pin to the 5V slot and buzzer's VCC pin to the 3.3 V slot.

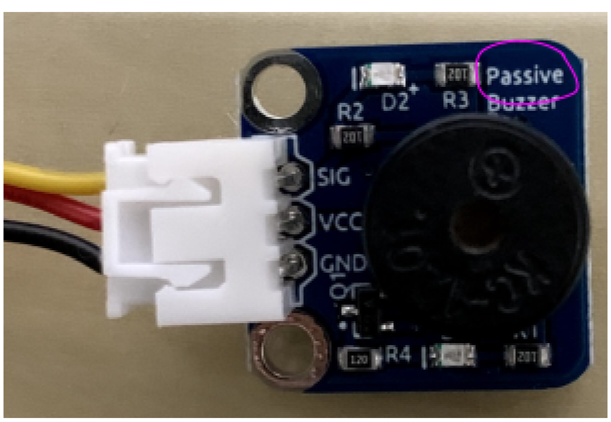
- The incoming pulse needed to be converted into meters from Hz, but I didn't know the formula, so I did some research to find other similar projects and borrowed a formula with citation (Lines 37-41)

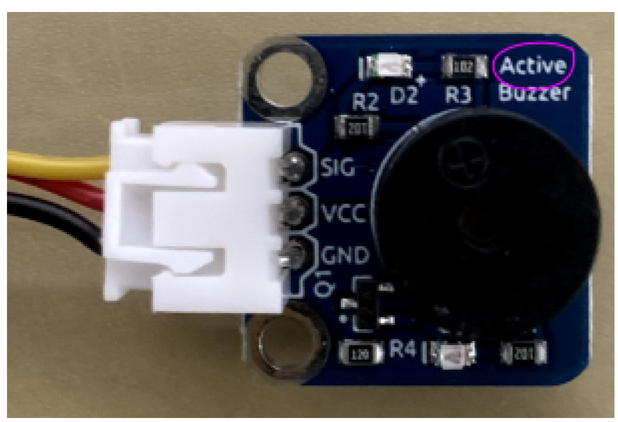
```
// method "pulseIn" comes from [5]
pulse = pulseIn(echopin, HIGH);

// formula for distance comes from [5]
distance = (pulse / 2) / 29.1;
```

Day 3:

- The passive buzzer did not produce the intended effect (it produced less of a buzz and more of a click), so I switched to the active buzzer since it produces a continuous and loud noise





Testing:

The video established that the sensor and the design work as intended. When the ultrasonic wave detector was blocked, the waves rebounded off my hand and resulted in the buzzer making a sound. The opposite was true as well, where if my hand was not within the frontal view or within 0.5 meters of the sensor, the buzzer would not create noise.

Limitations:

Though this design fulfills its purpose, having sufficient precision, it is quite clunky and rugged. The chips and the wiring is slightly unkempt and spread out. However, if encased in a container that can house all of the equipment in a compact space, the device could be used in practical applications. If the ultrasonic sensor was glued to opaque sunglasses with a smaller version of the Arduino, then the design could potentially be used in real-life situations. Nevertheless, the device isn't quite portable or usable in its current design.

CONCLUSION

Just like how ultrasonic echolocation helps bats find their prey with only hearing, this device uses an ultrasonic wave sensor to detect objects in front and nearby it. This assists the optically disadvantage with navigation as it informs the user with a noise produced by an active buzzer that an object half a meter in front of them is present. The device proved to function as intended, though it is limited to a certain FOV and distance (however, the distance threshold can be changed manually through the code).

Literature Cited

[1]
Elizabeth Hagen. (2009, November 04). Echolocation. ASU - Ask A Biologist. Retrieved July 16, 2021 from https://askabiologist.asu.edu/echolocation

[2]
"LASIK Eye Surgery." Mayo Clinic, Mayo Foundation for Medical Education and Research, 8 Nov. 2019, www.mayoclinic.org/tests-procedures/lasik-eye-surgery/about/pac-20384774.

[3]
"Laser Surgery Overview." Johns Hopkins Medicine, www.hopkinsmedicine.org/health/treatment-tests-and-therapies/laser-surgery-overview.

[4]
"Healthy Air Humidity." Nortec, www.condair.com/m/0/importance-of-healthy-air-humidity-in-hospitals.pdf.

[5]
Ali, Ammar Ated. "Detecting Obstacles and Warning - Arduino and Ultrasonic." Arduino Project Hub, 12 Feb. 2016, create.arduino.cc/projecthub/ammaratef45/detecting-obstacles-and-warning-arduino-and-ultrasonic-13e5ea.