A Project report on

“Navigation Alerts for the Blind”

Submitted for partial fulfillment of Engineering Explorations Course
For Bachelor of Technology
Degree in

Computer Science and Engineering

For the academic year 2023-24
By,

Sl	USN	Name
1	1RVU23407	Sanjana Kulkarni
2	1RVU23407	Samarth Setty
3	1RVU23407	Sanidhya Tiwari
4	1RVU23407	C S Raksha
5	1RVU23407	Risha N
6	1RVU23407	Prathyush Menon

Title of the Domain: Navigation Alerts for the Blind
Name of the Domain Lead: Dr Mydhili & Prof. Paul

Date of Submission (Faculty signature):10-05-2024

School of Computer Science and Engineering

RV University

RV Vidyanikethan Post, 8th Mile, Mysore Rd, Mailasandra,
Bengaluru, Karnataka 560059

Date of Submission (Faculty signature):10-05-2024

School of Computer Science and Engineering

CERTIFICATE

This is to Certify that the project work “Navigation Alerts for the Blind” carried out by Students Sanjana Kulkarni, C S Raksha, Samarth Setty, Sanidhya Tiwari, Risha N & Prathyush Menon USN: 1RVU23407, 1RVU23128, 1RVU23401, 1RVU23406, 1RVU23374 and 1RVU23352 bonafide students of SoCSE in partial fulfillment of the Engineering Explorations course of Bachelor of Technology in Computer Science and Engineering of the RV, Bengaluru, during the academic year 2023-24. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in this report. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said course.

Title of the Domain: Navigation Alerts for the blind

Signature of the Domain Lead

Name of the Domain Lead: Dr Mydhili & Prof Paul

Date of Submission (Faculty signature):10-05-2024

INDEX

Date of Submission (Faculty signature):10-05-2024

Abstract

While talking signs, guide canes, and echolocation devices can be useful in helping visually impaired people navigate their environment, they are not always sufficient to provide complete spatial awareness, especially when it comes to merging into traffic. We suggest a sophisticated navigation alert system that makes use of a range of sensors and feedback systems to provide visually impaired people more safety and autonomy. Using ultrasonic detection technology, the system locates impediments in the user's immediate area and uses a vibratory feedback mechanism to provide real-time guidance via safe navigation. This technique allows the user to determine how near they are to potential risks by using different frequencies to show the closeness of barriers. This technology is used with a microprocessor to provide the system with accurate and timely information.

Introduction

This project is an improving navigation alert system using a combination of sensors and feedback mechanisms designed to enhance the safety and autonomy of blind people. The concept originated from the use of talking signs, guiding canes, and echolocation equipment, which help the blind and visually challenged find their way about. However, we noticed that these aids fall short of offering a comprehensive spatial awareness, particularly when it comes to integrating safety in dynamic environments.

This project uses the stereophonic mechanism which is abided by the use of ultrasonic detection technology to identify obstacles in the user's immediate vicinity. The sensory feedback provides real-time guidance. The frequency of the vibrations used depends on the user's proximity to the obstacle. The closer the obstacle, the more intense the vibrations, thus offering a clear idea of the upcoming obstacle.

The entire system is housed on a cap, allowing for a more convenient use. The system's simple design makes it practical to use for daily use and allows for quick and easy wearing and removal. With a microprocessor, the cap-mounted device processes data precisely and quickly, giving users signals that they can text to help them navigate their surroundings securely.

Our navigation alert system gives people who are blind a more independent and secure way to explore their environment by offering this comprehensive and simple approach. This technology does not only enhance personal safety but also makes navigation more effective and efficient in various dynamic situations. We expect that by developing this system, we will be able to address the disadvantages in the currently available assistive technology and offer visually impaired individuals a greater level of autonomy and independence in their daily lives.

Materials required

Ultrasonic Sensor:

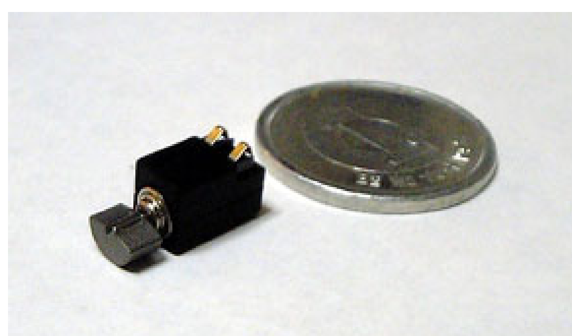
- Model: HC-SR04
- Measuring Range: 2 cm to 400 cm
- Accuracy: ±3 mm
- Operating Voltage: 5V
- Trigger Input Signal: 10 µs TTL pulse
- Echo Output Signal: TTL level signal
- Working Principles:

The HC-SR04 ultrasonic sensor sends out an ultrasonic pulse and measures the time taken for the pulse to return after reflecting off an object. The time delay is used to calculate the distance to obstacles in the user's path, providing critical spatial awareness information for safe navigation.

Vibratory Feedback Mechanism:

- Type: Vibration motor
- Operating Voltage: Typically 3V to 5V
- Frequency Range: Adjustable, depending on proximity to obstacles
- Control: PWM signal for frequency variation
- Working Principles:

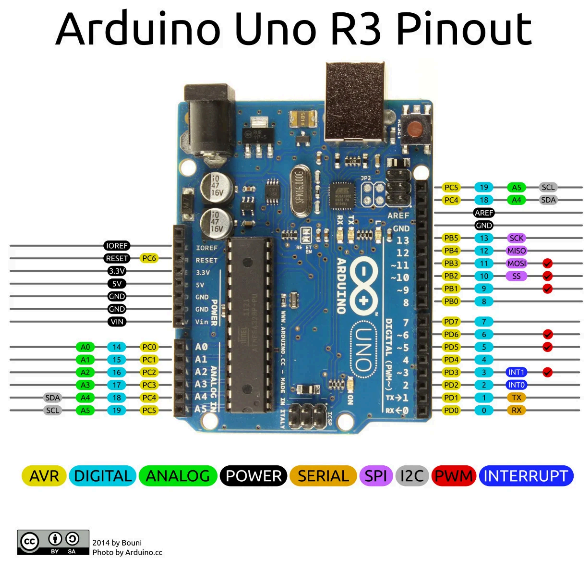
The vibratory feedback mechanism uses varying frequencies of vibrations to alert the user to obstacles. As the user approaches an object, the frequency of vibrations increases, providing intuitive guidance about the proximity of obstacles in their path.



Microcontroller:

- Model: Arduino Uno
- Processor: ATmega328P
- Operating Voltage: 5V
- I/O Pins: 14 digital, 6 analog
- Clock Speed: 16 MHz
- Memory: 32 KB flash, 2 KB RAM, 1 KB EEPROM
- Working Principles:

The Arduino Uno microcontroller processes data from the ultrasonic sensor and controls the vibratory feedback mechanism. It translates the sensor data into meaningful feedback for the user and manages the system’s overall logic.



Wearable Cap:

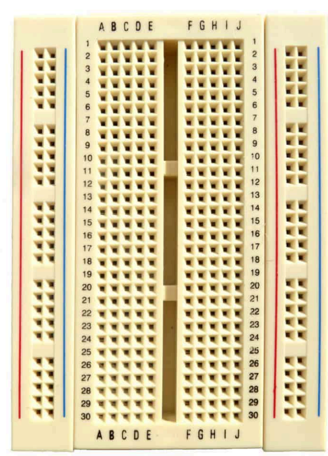
- Material: Lightweight, comfortable, and durable
- Attachment: Designed for easy removal and putting on
- Working Principles:

The system's components are integrated into a wearable cap, providing a convenient form factor for the user. The cap's design allows for quick attachment and detachment, making the system easy to use and user-friendly.

Breadboard:

- Type: Solderless prototyping board
- Working Principles:

The breadboard serves as a platform for assembling and testing the circuit without the need for soldering. It provides a convenient and flexible way to connect and experiment with the project's electronic components.



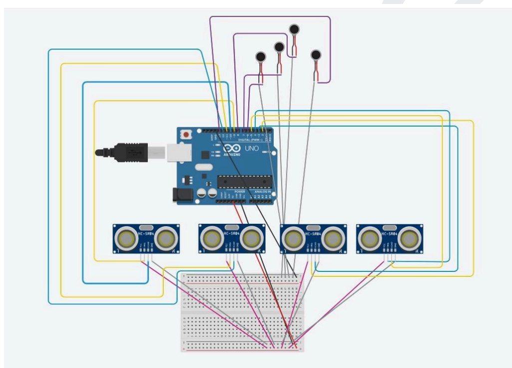
Jumper Wires:

- Type: Connecting wires (typically male-to-male, male-to-female, or female-to-female)
- Working Principles:

Jumper wires are used to create connections between the various components on the breadboard. They enable quick and easy modifications to the circuit.

Diagrams

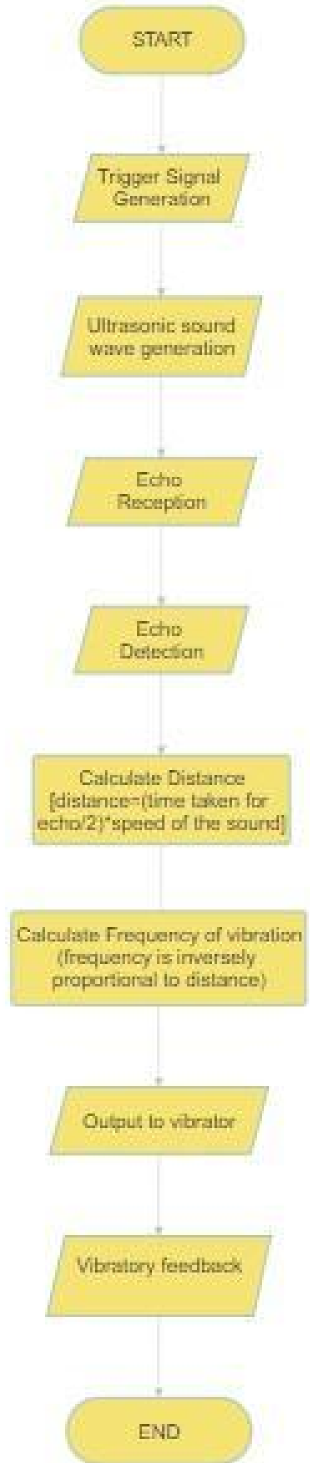
Circuit diagram



1. Ultrasonic Sensor Pins:
   - Left Sensor 1:
     - Trigger Pin: Pin 9 (trigPinLeft1)
     - Echo Pin: Pin 10 (echoPinLeft1)
   - Left Sensor 2:
     - Trigger Pin: Pin 11 (trigPinLeft2)
     - Echo Pin: Pin 12 (echoPinLeft2)
   - Right Sensor 1:
     - Trigger Pin: Pin 2 (trigPinRight1)
     - Echo Pin: Pin 3 (echoPinRight1)
   - Right Sensor 2:
     - Trigger Pin: Pin 4 (trigPinRight2)
     - Echo Pin: Pin 5 (echoPinRight2)

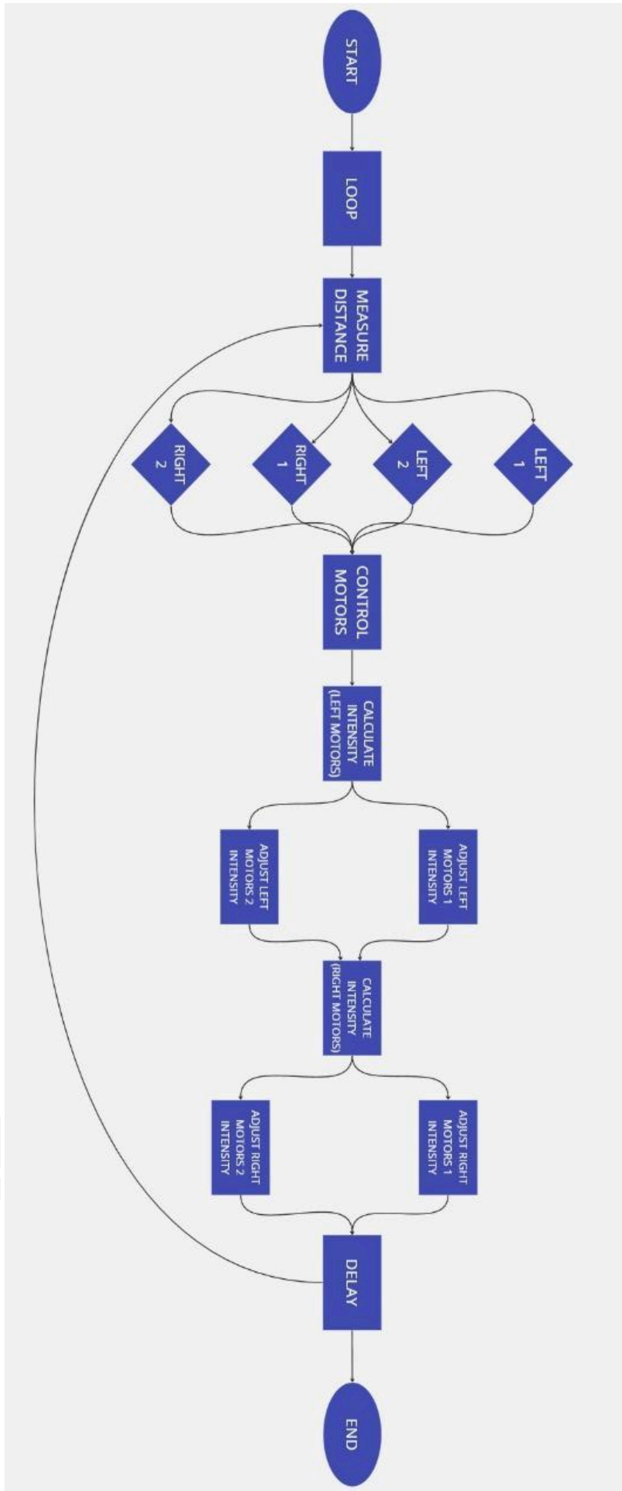
2. Vibration Motor Pins:
   - Left Motor:
     - Motor Pin 1: Pin 6 (motorLeft1Pin)
     - Motor Pin 2: Pin 7 (motorLeft2Pin)
   - Right Motor:
     - Motor Pin 1: Pin 8 (motorRight1Pin)
     - Motor Pin 2: Pin 13 (motorRight2Pin)

Flowchart of working hardware



Date of Submission (Faculty signature):10-05-2024

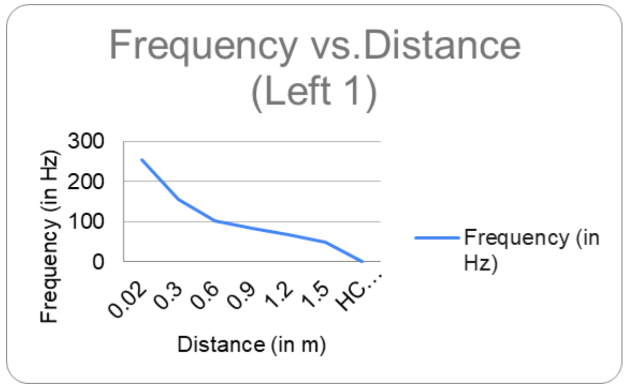
Flowchart of code: Refer to Appendix (i) for the code



Date of Submission (Faculty signature):10-05-2024

Data Analysis

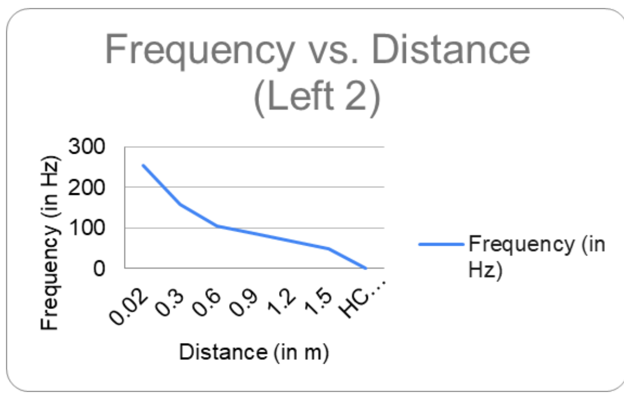
HC SR04 L1	
Distance (in m)	Frequency (in Hz)
0.02	255
0.3	156
0.6	103
0.9	85
1.2	67
1.5	50



Sensor L1

This graph shows a gradual decrease in frequency as the distance increases.
The decline is relatively smooth, suggesting consistent signal attenuation.
The data points align well with the overall trend.

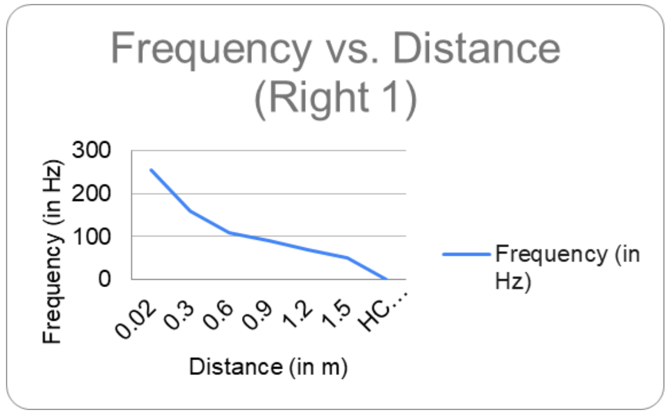
HC SR04 L2	
Distance (in m)	Frequency (in Hz)
0.02	255
0.3	158
0.6	105
0.9	87
1.2	69
1.5	50



Sensor L2

Similar to L1, this graph exhibits a decreasing trend.
However, there are slight fluctuations in the data points.
These fluctuations could be due to environmental factors or measurement noise.

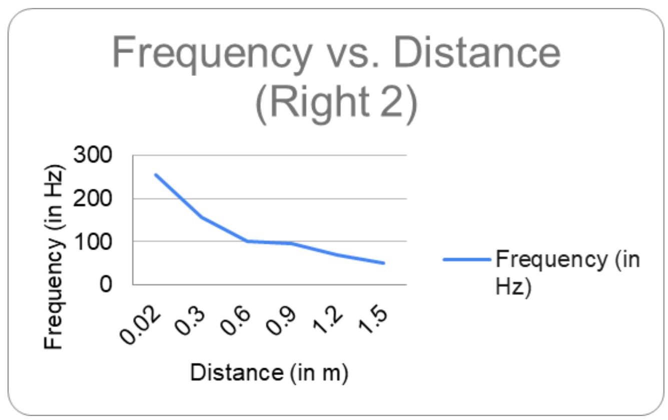
HC SR04 R1	
Distance (in m)	Frequency (in Hz)
0.02	255
0.3	160
0.6	110
0.9	90
1.2	70
1.5	50



Sensor R1:

Again, we observe a negative correlation between frequency and distance.
The graph appears smoother than L2 but has some minor variations.
These variations might be related to the specific setup or conditions.

HC SR04 R2	
Distance (in m)	Frequency (in Hz)
0.02	255
0.3	156
0.6	102
0.9	95
1.2	68
1.5	50



Sensor R2:

Like the other sensors, R2 also displays a decreasing frequency trend.
The graph is relatively consistent, with minimal fluctuations.
The data points align well with the overall pattern.

Key Takeaways:

All sensors experience signal attenuation or damping as distance increases. L1 and R2 show smoother trends, while L2 and R1 have minor fluctuations.
Environmental factors, measurement precision, and sensor characteristics may contribute to these differences.

Date of Submission (Faculty signature):10-05-2024

Discussions

Challenges and solutions

In the course of our project, we initially explored ambitious concepts such as a hydrogen fuel generator and a convertible wheelchair-to-crutches device. While these ideas held significant promise, their implementation was deemed unviable due to the associated risks and costs. Consequently, we redirected our focus towards the development of a cap-based system utilizing stereophonic feedback to aid visually impaired individuals in navigating their surroundings.

The design and execution of this project presented a series of formidable challenges, necessitating a culture of innovative problem-solving and adaptability within our team. Our original choice of the ESP32 microcontroller was swiftly reassessed due to defects, prompting a seamless transition to the Arduino Uno to ensure project continuity.

A critical aspect of our endeavor involved the precise placement of sensors on the cap, considering their 15-degree measurement range and 4-meter coverage distance to optimize functionality. This process demanded iterative experimentation but culminated in a successful sensor configuration.

Reflecting on our journey, the ability to make swift decisions, exemplified by the microcontroller transition, emerged as a pivotal factor in our progress. Attention to detail, particularly in sensor positioning, played a crucial role in enhancing the efficacy of our system.

Ultimately, our project underscored the significance of perseverance and creative problem-solving in overcoming challenges. By remaining steadfast in our pursuit and embracing innovative solutions, we successfully developed a practical and impactful technology aimed at enhancing the mobility and independence of visually impaired individuals.

Application

The innovative cap-based system offers a new approach to assistive technology by providing real-time spatial awareness and ensuring safe mobility, particularly in bustling environments such as busy streets and crowded spaces. This prototype can potentially enhance how individuals navigate their surroundings significantly.

The system prioritizes accessibility through its intuitive design, allowing for easy wear and hands-free operation. This accessibility consolidates the user experience, empowering individuals with greater freedom and independence.

In dynamic environments such as educational institutions, the system can be a reliable source of safety and guidance. Both students and staff can benefit from an additional layer of security and assistance, enabling smooth navigation through changing surroundings.

Overall, the prototype represents a beacon of innovation in assistive technology and accessibility, promising a brighter, safer, and more independent future for those who utilize it.

Future Work

The future work of the project concentrates on enhancing the current navigation aid system for visually impaired individuals to improve usability and effectiveness.

To begin with, incorporating a camera and AI-based object detection can enable the system to recognize and classify obstacles more precisely. This advancement could offer users more detailed and actionable feedback, thereby enhancing their navigation experience.

Additionally, including auditory alerts as an alternative to vibratory cues can provide visually impaired individuals with more options for receiving important navigational information. These auditory signals can be particularly useful in busy or noisy environments.

Integrating GPS technology can further benefit the system by providing real-time route guidance and destination tracking. This feature can significantly improve the system's ability to help users navigate confidently to specific locations.

Wireless connectivity, such as Bluetooth or Wi-Fi, opens up opportunities for the system to communicate with other devices, including smartphones and smart home systems. This connectivity could facilitate remote monitoring and assistance, thereby contributing to a better overall experience.

Finally, optimizing the system's power management will be crucial for extending battery life and ensuring reliable performance. By minimizing power consumption, the system will be able to operate continuously and efficiently.

Focusing on these areas for future development can help evolve the project into a more sophisticated and comprehensive navigation aid, ultimately offering visually impaired individuals greater independence and quality of life.

References

[i] Santos, S. (2021) ESP32 with HC-SR04 Ultrasonic Sensor with Arduino IDE, Random Nerd Tutorials. Available at: https://randomnerdtutorials.com/esp32-hc-sr04-ultrasonic-arduino/ (Accessed: May 10, 2024).

[ii](No date) Researchgate.net. Available at: https://www.researchgate.net/figure/Ultrasonic-sensors-and-Arduino-circuit-diagram-Sensors-echo-and-trigger-pin-are_fig3_332916346 (Accessed: May 10, 2024).

[iii]Movva, N. (no date) Pathfinder: Wearable Navigational Aid for the Blind. Haptics + Ultrasonic + IMU + MCU.

Date of Submission (Faculty signature):10-05-2024

Link for video:
https://drive.google.com/drive/folders/1IWvud58XFp9iKrtMfnEMZW1D2R1KtPF0?usp=drive_link

Github - link:
https://github.com/Sanjana765Kulkarni/Navigation-alerts-for-the-blind

Appendix

(i) Code:-

```
// Define pins for left sensors
const int trigPinLeft1 = 9;
const int echoPinLeft1 = 10;
const int trigPinLeft2 = 11;
const int echoPinLeft2 = 12;

// Define pins for right sensors
const int trigPinRight1 = 2;
const int echoPinRight1 = 3;
const int trigPinRight2 = 4; // Changed from 6 to avoid overlap
const int echoPinRight2 = 5; // Changed from 7 to avoid overlap

// Define pins for motors
const int motorLeft1Pin = 6; // Changed from 2 to avoid overlap
const int motorLeft2Pin = 7; // Changed from 3 to avoid overlap
const int motorRight1Pin = 8; // Changed from 4 to avoid overlap
const int motorRight2Pin = 13; // Changed from 5 to avoid overlap

// Define vibration intensity limits
const int minIntensity = 50; // Minimum vibration intensity
const int maxIntensity = 255; // Maximum vibration intensity

// Variables for left sensors
long durationLeft1;
float distance_cmLeft1;
long durationLeft2;
float distance_cmLeft2;

// Variables for right sensors
long durationRight1;
float distance_cmRight1;
long durationRight2;
float distance_cmRight2;

// Delay between scans/waves when an object is far
int longDelay = 2000; // Long delay
int shortDelay = 500; // Short delay

void setup() {
  // Set up sensor pins
  pinMode(trigPinLeft1, OUTPUT);
  pinMode(echoPinLeft1, INPUT);
  pinMode(trigPinLeft2, OUTPUT);
  pinMode(echoPinLeft2, INPUT);
  pinMode(trigPinRight1, OUTPUT);
  pinMode(echoPinRight1, INPUT);
  pinMode(trigPinRight2, OUTPUT);
  pinMode(echoPinRight2, INPUT);

  // Set up motor pins
  pinMode(motorLeft1Pin, OUTPUT);
  pinMode(motorLeft2Pin, OUTPUT);
  pinMode(motorRight1Pin, OUTPUT);
  pinMode(motorRight2Pin, OUTPUT);
```

```
  Serial.begin(9600); // Start serial communication
}

void loop() {
  // Measure distances from all sensors
  measureDistances();

  // Control motors based on proximity readings
  controlMotors();

  // Adjust delay between measurements based on distance readings
  adjustDelay();

  // Delay for stability
  delay(longDelay); // Adjust as necessary
}

void measureDistances() {
  // Measure distance from left sensors
  measureDistance(trigPinLeft1, echoPinLeft1, durationLeft1, distance_cmLeft1);
  measureDistance(trigPinLeft2, echoPinLeft2, durationLeft2, distance_cmLeft2);

  // Measure distance from right sensors
  measureDistance(trigPinRight1, echoPinRight1, durationRight1, distance_cmRight1);
  measureDistance(trigPinRight2, echoPinRight2, durationRight2, distance_cmRight2);
}

void controlMotors() {
  // Calculate vibration intensity for each motor
  int left1Intensity = calculateIntensity(distance_cmLeft1);
  int left2Intensity = calculateIntensity(distance_cmLeft2);
  int right1Intensity = calculateIntensity(distance_cmRight1);
  int right2Intensity = calculateIntensity(distance_cmRight2);

  // Control left motors
  analogWrite(motorLeft1Pin, left1Intensity);
  analogWrite(motorLeft2Pin, left2Intensity);

  // Control right motors
  analogWrite(motorRight1Pin, right1Intensity);
  analogWrite(motorRight2Pin, right2Intensity);
}

int calculateIntensity(float dist) {
  if (dist < 150) { // If distance is less than 1.5 meters
    // Map distance to vibration intensity (linear mapping)
```

```
    int intensity = map(dist, 0,
150, maxIntensity, minIntensity);
    return intensity;
  } else {
    return 0; // No vibration if
distance is greater than 1.5
meters
  }
}

void measureDistance(int trigPin,
int echoPin, long &duration, float
&distance_cm) {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin,
HIGH, 30000); // Timeout after
30ms
  if (duration == 0) {
    // Handle timeout or other
errors
    distance_cm = -1;
  } else {
    distance_cm = duration *
0.0343 / 2;
  }
}

void adjustDelay() {
  if (distance_cmLeft1 < 150 ||
distance_cmLeft2 < 150 ||
distance_cmRight1 < 150 ||
distance_cmRight2 < 150) {
    // If any sensor detects an
object within 1.5 meters, set a
short delay
    longDelay = shortDelay;
  } else {
    // If no object is detected
within 1.5 meters by any sensor,
set a longer delay
    longDelay = 2000;
  }
}
```