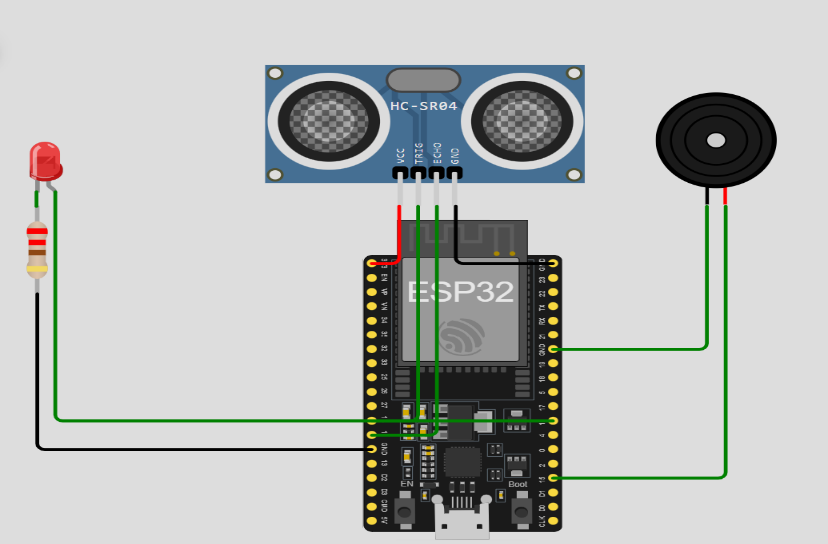
**Wrist-Worn Multimodal Obstacle Detection System For the Visually Impaired**

## ABSTRACT

This paper presents the design and implementation of a Wrist-Worn Multimodal Obstacle Detection System catering to the specific needs of visually impaired individuals. The system integrates an HC-SR04 ultrasonic sensor for precise obstacle detection, a vibration motor for tactile feedback, and a buzzer for auditory alerts. The device, designed to resemble a watch, offers a compact and wearable solution for enhancing the mobility and safety of visually impaired users. The user-friendly interface includes a simple on/off switch for ease of operation, ensuring accessibility for individuals with visual impairments in addition to a rechargeable battery. The system prioritizes smooth transitions while operating it and empowers users to adapt the device to their preferences and varying environmental conditions. Safety features, including fail-safes, are implemented to prevent false alarms, ensuring a reliable and comfortable user experience. The project aims to empower visually impaired individuals by providing a versatile and intuitive tool for enhancing their awareness of surroundings, promoting independent and confident navigation. The Wrist-Worn Multimodal Obstacle Detection System stands as a promising solution for addressing the challenges faced by the visually impaired, offering a customizable, compact, and wearable device that aligns with the principles of inclusivity and accessibility.

**Virtual Image:-**



**Code:-**

const int trigPin = 12;

const int echoPin = 14;

const int buzzerPin = 15;

const int ledPin = 16;  // LED in place of the vibration motor

void setup() {

**Serial**.begin(115200);

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  pinMode(buzzerPin, OUTPUT);

  pinMode(ledPin, OUTPUT);

}

void loop() {

  long distance = getDistance();

**Serial**.println("Distance: " + String(distance) + " cm");

  if (distance < 50) {

**Serial**.println("Activating Buzzer and LED");

    tone(buzzerPin, 2000, 1000);  // Buzzer on

    digitalWrite(ledPin, HIGH);  // LED on

    delay(1000);

    noTone(buzzerPin);  // Buzzer off

    digitalWrite(ledPin, LOW);  // LED off

  }

  delay(500);

}

long getDistance() {

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  return pulseIn(echoPin, HIGH) \* 0.034 / 2;

}

**Real Image:-**



