Smart Blind Stick Using Arduino

Prathyusha Baikani

(Robotics Engineer)

1. Introduction

The Smart Blind Stick is an assistive technology designed to help visually impaired individuals navigate their surroundings.

This device uses an **Ultrasonic Sensor** to detect obstacles and then alerts the user using **vibrations** and a **buzzer**. By providing real-time feedback, the stick enhances mobility and safety.

2. Features

- **Obstacle Detection**: Detects obstacles in the path of the user using ultrasonic waves.
- **Alert Mechanism**: Alerts the user using both **sound** (buzzer) and **haptic feedback** (vibration motor).
- **Portable**: The compact design makes it easy to integrate into a regular walking stick.
- **Affordable**: Uses inexpensive components, making it accessible to a wide range of people.
- **Customizable Alerts**: Allows users to adjust the distance threshold for obstacle detection and change the type of alert.

3. Applications

- **Assistive Technology**: Helps visually impaired individuals in walking and avoiding obstacles.
- **Educational Tool**: A hands-on project for learning about sensors, motors, and Arduino programming.
- **Safety**: Can be used in crowded areas to prevent accidents and collisions.

4. Components Required

- **Arduino Nano**: The microcontroller that will control the system.
- **Ultrasonic Sensor (HC-SR04)**: Measures the distance between the stick and obstacles.
- **Buzzer**: Emits sound to alert the user when an obstacle is detected.
- **Vibration Motor**: Provides haptic feedback through vibration when an obstacle is detected.
- **Battery (9V or Li-ion)**: Power source for the system.
- **Jumper Wires**: For making connections between the components.

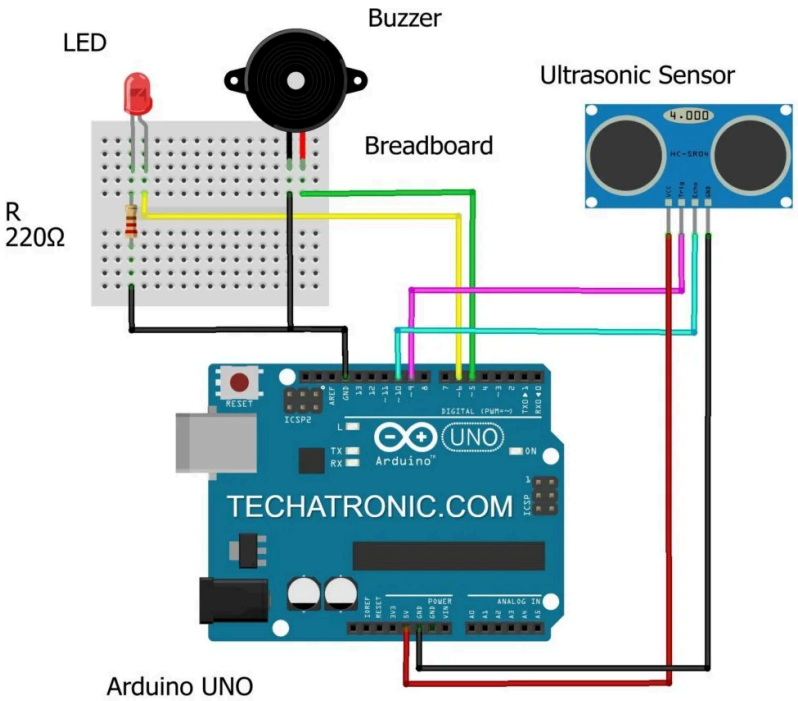
5. Working Principle

1. **Distance Measurement**: The ultrasonic sensor sends a pulse and calculates the time it takes for the pulse to bounce back after hitting an object.
2. **Obstacle Detection**: Based on the returned time, the Arduino calculates the distance to the obstacle.
3. **Alert Activation**: If the detected distance is below a threshold (e.g., 30 cm), the Arduino triggers both the buzzer and vibration motor.
4. **Continuous Monitoring**: The system continuously measures the distance and updates the alerts accordingly.

6. Circuit Diagram

The wiring diagram shows how the ultrasonic sensor, buzzer, and vibration motor are connected to the Arduino Nano. Below is a simplified explanation:

- **Ultrasonic Sensor (HC-SR04)**:
  - VCC → 5V pin of Arduino
  - GND → GND pin of Arduino
  - Trigger Pin → Digital Pin 9 on Arduino
  - Echo Pin → Digital Pin 10 on Arduino
- **Buzzer**:
  - One pin of the buzzer → Digital Pin 3 on Arduino
  - Other pin → GND on Arduino
- **Vibration Motor**:
  - One pin of the motor → Digital Pin 4 on Arduino
  - Other pin → GND on Arduino



7. Code Explanation

This Arduino code reads distance data from the ultrasonic sensor. When an obstacle is detected within a specified range (e.g., 30 cm), it triggers the buzzer and vibration motor.

```
#define trigPin 9
#define echoPin 10
#define buzzerPin 3
#define vibrationPin 4

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(vibrationPin, OUTPUT);
  Serial.begin(9600);  // Start serial communication for debugging
}

void loop() {
  long duration;
  int distance;

  // Sending pulse to the sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reading the time it takes for the pulse to return (echo)
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;  // Calculate distance in cm

  // Output the distance to serial monitor
  Serial.println(distance);
```

```
  // If an obstacle is within 30 cm, alert the user
  if (distance < 30) {
    digitalWrite(buzzerPin, HIGH);  // Activate buzzer
    digitalWrite(vibrationPin, HIGH);  // Activate vibration motor
  } else {
    digitalWrite(buzzerPin, LOW);  // Deactivate buzzer
    digitalWrite(vibrationPin, LOW);  // Deactivate vibration motor
  }

  delay(100);  // Delay before the next reading
}
```

8. Advantages

- **Enhanced Mobility**: Helps the visually impaired avoid obstacles while walking.
- **Low-Cost Solution**: Uses affordable components such as the Arduino Nano and ultrasonic sensor.
- **Simple Design**: Easy to assemble and program.
- **User Customizable**: Users can adjust alert parameters, like distance threshold, as needed.
- **Learning Opportunity**: Great for anyone wanting to learn about Arduino programming and sensor integration.

9. Future Enhancements

- **Multiple Sensor Integration**: Add more ultrasonic sensors to detect obstacles from different directions, providing a full 360-degree awareness.
- **Bluetooth Integration**: Allow users to connect the stick to their smartphone for additional functionality (e.g., app alerts or remote monitoring).

- **Solar Power**: Add a solar panel for charging the battery to make it more energy-efficient for outdoor use.
- **Gesture Recognition**: Use sensors or cameras to recognize hand gestures, providing additional control over the stick.
- **Voice Alerts**: Integrate a voice module for spoken alerts, further enhancing usability.

10. Troubleshooting

1. **Buzzer not working**:
    - Check the connection to the correct pin (Pin 3 on Arduino).
    - Ensure the buzzer is functional by testing it with a simple blink code.
2. **Sensor not detecting obstacles properly**:
    - Ensure the sensor is wired correctly, especially the Echo and Trigger pins.
    - Verify that the sensor is free from any obstruction or dirt.
3. **No vibration feedback**:
    - Ensure the motor is connected to the correct pin (Pin 4 on Arduino).
    - Test the motor separately to check if it's functional.

11. Conclusion

The **Smart Blind Stick** is an excellent project for anyone looking to create a simple assistive device for the visually impaired. It not only helps improve mobility for individuals with vision loss but also provides an educational opportunity to learn about sensors, motors, and Arduino programming.

—----------THANK YOU —----------