Assistive Wearable Technology

For the Visually Impaired

Submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in

Electronics and Communciation Engineering

By

CHANDRA PRAKASH (2124446)

PRITESH MISHRA (2124469)

Department of Electronics and Communication Engineering

I.K.GUJRAL PUNJAB TECHNICAL UNIVERSITY

DECLARATION

We, Chandra Prakash(Roll no- 2124446), Pritesh Mishra(Roll. No. 2124469), hereby declare that the Interdisciplinary Project Report entitled “Assistive Wearable Technology For the Visually Impaired ” done by us under the guidance of Dr. Rakesh Goyal Sir , Department of Electronics and Communication Engineering is submitted in partial Fulfillment of the requirements for the award of Bachelor of Technology degree in Electronics and Communication Engineering

DATE

PLACE SIGNATURE OF THE CANDIDATE

ABSTRACT

Assistive Wearable Technology is a groundbreaking innovation designed to empower visually impaired individuals by enhancing their mobility, independence, and safety in day-to-day life. This project introduces a comprehensive solution combining a smart cap and smart shoes, integrated with state-of-the-art microcontroller technology, to provide real-time obstacle detection and feedback.

The smart cap is equipped with proximity sensors strategically placed to detect obstacles at head height and provide auditory or haptic alerts, ensuring users can navigate safely in various environments. Complementing the cap, the smart shoes feature embedded sensors that monitor the ground level for potential hazards such as uneven surfaces, potholes, or steps. These sensors are interconnected with a microcontroller that processes data and triggers responsive feedback systems, such as vibrations or audio cues, to guide the user effectively.

By leveraging advanced sensor technology, microcontroller integration, and intuitive feedback mechanisms, this assistive wearable solution aims to bridge the gap between accessibility and technological innovation. The system's user-centric design focuses on delivering a seamless experience that promotes self-reliance and confidence for visually impaired individuals, ultimately improving their quality of life. This project not only demonstrates the potential of wearable technology in assistive applications but also highlights its role in fostering inclusivity and accessibility in society.

Table of Contents

Introduction

Visually impaired individuals often face challenges in navigating their environment safely and independently. This project addresses these challenges by designing a wearable system consisting of a smart cap and smart shoes. These devices use ultrasonic sensors to detect obstacles and provide feedback through haptic and auditory signals.

Design and Implementation

Smart Cap

- Description: A standard cap fitted with an ultrasonic sensor and a buzzer.
- Placement: The ultrasonic sensor is positioned 5 cm above the brim, facing forward for obstacle detection.
- Functionality: Detects obstacles at head height and alerts the user through audible signals.

Smart Shoes

- Description: Regular shoes equipped with ultrasonic sensors and vibration motors.
- Placement: Sensors are attached to the toe area, 2 cm above the ground, angled slightly upward. Vibration motors are embedded in the sole.
- Functionality: Detects obstacles on the ground and provides directional feedback via vibrations.

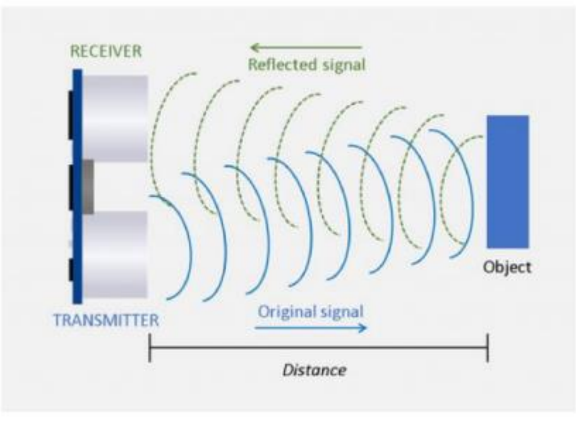
Components and Specifications

Component	Specification
Arduino Uno	Microcontroller for processing input and controlling actuators.
Ultrasonic Sensor	HC-SR04; Detection range: 2 cm - 400 cm, Accuracy: ±3 mm.
Vibration Motor	Operating voltage: 3V-5V; Provides haptic Feedback
Buzzer	Operates at 5V; Produces clear audible signals.
Battery	9V; Powers the system.
Jumper Wires	For secure component connections.
Shoes and Cap	Standard items modified for housing sensors and actuators.
Breadboard	Temporary assembly platform for testing.

HARDWARES USED

1.**ULTRASONIC SENSOR**:-

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity. High-frequency sound waves reflect from boundaries to produce distinct echo patterns. Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. Ultrasonic sensor uses a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse. IoT ultrasonic sensors are designed for non-contact detection of solid and liquid objects. These sensors are used for a wide variety of functions from monitoring the level of water in a tank to fluid identification/concentration, to detecting object proximity.



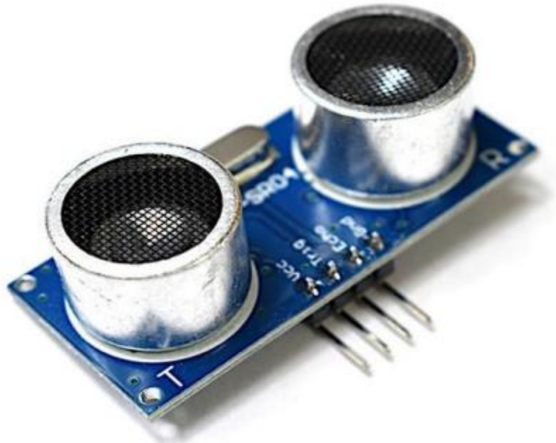


FIG 4.2.2 ULTRASONIC SENSOR

BUZZER

A Buzzer is an audio signalling device. There are many types of buzzer and here 5V passive Buzzer is used, which is used to create the sound and it may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke. The passive buzzer is an electromagnetic squeaker used to generate sound signals of different frequencies. It requires an AC signal to make a sound, where a changing input signal produces the sound, rather than producing a tone automatically. To use this 5v buzzer, connect one pin to ground and the other to a microcontroller programmed to output a square wave or a timer IC.

FIG 4.2.3 5V PASSIVE BUZZER

VIBRATOR

Vibration motor is a compact size coreless DC motor used to informs the users of receiving the signal by vibrating, no sound. Vibration motors are widely used in a variety of applications including cell phones, handsets, pagers, and so on. A vibratory motor is a three-phase motor that is intentionally unbalanced, and is also known as an eccentric rotating mass (ERM) or vibrating motor. This tiny motor produces vibrations by spinning an eccentric shaft at over 900 RPM when power at 1.5V. It is intended for operation around 1.5V, and polarity is not important that is, the motor can run CW or CCW. The main purpose of this vibrator motor is to alerts the user from receiver the call by without sound and vibrating. These motor are applicable for different categories like pager, handsets, cell phones, bluetooth etc. Vibration motor is a coreless DC motor and the size of this motor is compact. The main purpose of this motor is to alert the user from receiving the call by without sound/vibrating. The main feature of this motor is, it has magnetic properties, lightweight, and motor size is small. Based on these features, the motor performance is highly consistent. The configuration of these motors can be done in two varieties one is coin model and another one is a cylinder model.

FIG 4.2.4 VIBRATOR MOTOR

LITHIUM BATTERY

Batteries that have lithium as their anode is called lithium batteries. The charge moves from anode to cathode during discharge and cathode to anode during charging. Lithium batteries were introduced way back in the 1980-the 1990s. These batteries have completely revolutionized the portable electronics market such as cellular telephones and laptop computers. Lithium Battery is a latest technology battery which has 5 features - 2x life with 100% charging efficiency, light weight and no maintenance. Lithium-Ion, or Li-Ion batteries are a type of rechargeable battery that's used in many applications, but most commonly in the electronics industry. Li-Ion batteries provide portable electricity, powering electronic gadgets such as mobile phones, laptops and tablets.

FIG 4.2.5 9V BATTERY

SOFTWARE USED

ARDUINO IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment. The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.

The Arduino IDE will appear as:

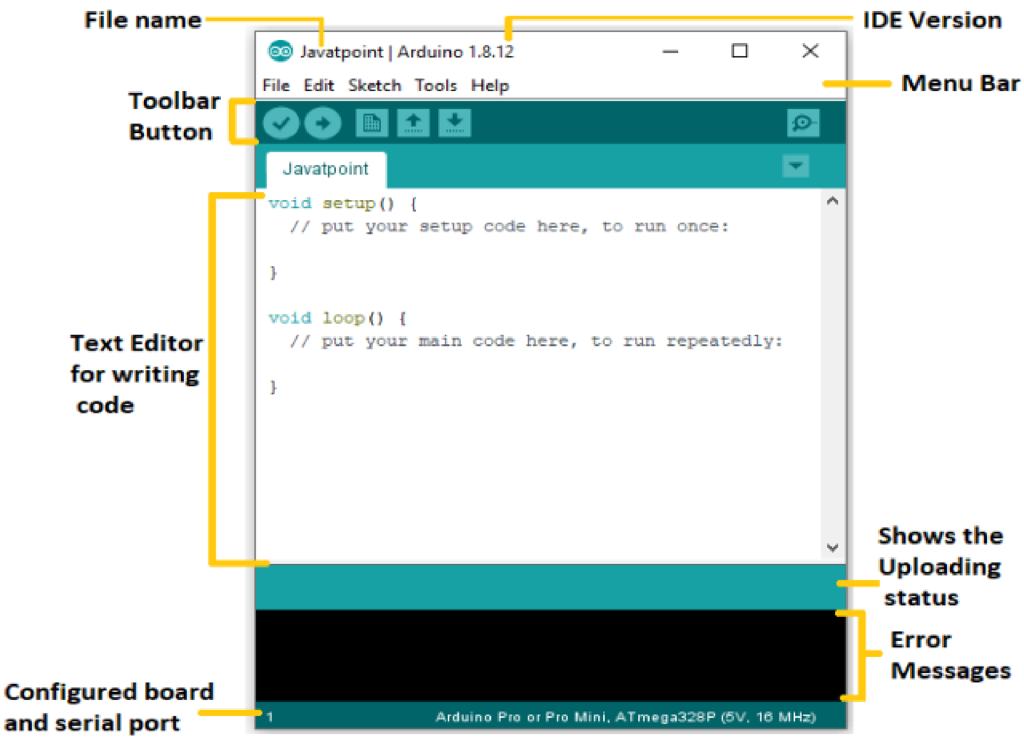


FIG 4.3.1 ARDUINO IDE INTERFACE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems. • Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50 • Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh 24 OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. • Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works. • Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to. Writing Sketches Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

CODE Implementation

Code for Smart Shoes

```
#define TRIG_FRONT 2
#define ECHO_FRONT 3
#define TRIG_DOWN 4
#define ECHO_DOWN 5
#define BUZZER 6

#define OBJECT_THRESHOLD 50
#define HOLE_THRESHOLD 10

void setup() {
  pinMode(TRIG_FRONT, OUTPUT);
  pinMode(ECHO_FRONT, INPUT);
  pinMode(TRIG_DOWN, OUTPUT);
  pinMode(ECHO_DOWN, INPUT);
  pinMode(BUZZER, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int frontDistance = measureDistance(TRIG_FRONT, ECHO_FRONT);
```

```
  int downDistance = measureDistance(TRIG_DOWN, ECHO_DOWN);

  if (frontDistance > 0 && frontDistance < OBJECT_THRESHOLD ||
      downDistance > 0 && downDistance < HOLE_THRESHOLD) {
    activateBuzzer();
  } else {
    deactivateBuzzer();
  }

  delay(100);
}

int measureDistance(int trigPin, int echoPin) {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  return duration * 0.034 / 2;
}

void activateBuzzer() {
  digitalWrite(BUZZER, HIGH);
```

```
}

void deactivateBuzzer() {
   digitalWrite(BUZZER, LOW);
}
```

Code for Smart Cap

```
// Pin Definitions
#define TRIG_FRONT 2
#define ECHO_FRONT 3
#define TRIG_UP 4
#define ECHO_UP 5
#define BUZZER 6
// Distance Thresholds (in cm)
#define OBJECT_THRESHOLD 50
#define UP_THRESHOLD 50
void setup() {
pinMode(TRIG_FRONT, OUTPUT);
pinMode(ECHO_FRONT, INPUT);
pinMode(TRIG_UP, OUTPUT);
pinMode(ECHO_UP, INPUT);
pinMode(BUZZER, OUTPUT);
```

```
Serial.begin(9600); // Optional for debugging
}
void loop() {
// Measure distances
int frontDistance = measureDistance(TRIG_FRONT,
ECHO_FRONT);
int upDistance = measureDistance(TRIG_UP, ECHO_UP);
// Debugging distances (optional)
Serial.print("Front Distance: ");
Serial.print(frontDistance);
Serial.print(" cm | UP Distance: ");
Serial.println(upDistance);
// Object Detection
if (frontDistance > 0 && frontDistance <
OBJECT_THRESHOLD) {
activateBuzzer();
}
// Hole Detection
else if (upDistance > 0 && upDistance < UP_THRESHOLD) {
activateBuzzer();
}
else {
deactivateBuzzer();
}
```

```
delay(100); // Small delay for stability
}
// Function to measure distance from Ultrasonic Sensor
int measureDistance(int trigPin, int echoPin) {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
long duration = pulseIn(echoPin, HIGH);
int distance = duration * 0.034 / 2; // Convert to cm
return (distance > 0 && distance < 400) ? distance : 0; //
Limit valid range
}
// Activate buzzer
void activateBuzzer() {
digitalWrite(BUZZER, HIGH);
}
// Deactivate buzzer
void deactivateBuzzer() {
digitalWrite(BUZZER, LOW);
}
```

Working Principle

1. ***Smart Cap:***

- Ultrasonic sensor’s TRIG pin connected to Arduino digital pin D2.
- Ultrasonic sensor’s ECHO pin connected to Arduino digital pin D3.
- Buzzer connected to digital pin D5.

2. Smart Shoes:

- Ultrasonic sensor’s TRIG pin connected to Arduino digital pin D6.
- Ultrasonic sensor’s ECHO pin connected to Arduino digital pin D7.
- Vibration motors connected to Arduino digital pins D8 and D9 via transistors for better power management.

Steps to Run

1. Hardware Assembly:

- Assemble the components according to the circuit diagram, ensuring all connections are secure.

- Mount the sensors and actuators on the cap and shoes as described in the design diagram.

2. Code Upload:

- Connect the Arduino Uno to a computer using a USB cable.
- Open the Arduino IDE, paste the code, and upload it to the Arduino Uno.

3. Power Supply:

- Attach the battery to the Arduino Uno's power input.
- Ensure the connections are firm and insulated.

4. Testing:

- Wear the cap and shoes.
- Test the system in a controlled environment to ensure proper detection and feedback.

Applications

- **Mobility Aid**: Enhances navigation for visually impaired individuals.
- **Crowded Areas**: Improves situational awareness in dense environments.

- **Rehabilitation**: Training tool for improving spatial awareness.
- **Educational Demonstration**: Showcases microcontroller applications.

Challenges and Solutions

Fault/Error	Solution
Incorrect sensor connections	Rechecked and corrected TRIG and ECHO connections.
Weak vibration feedback	Added a transistor for current amplification.
High power consumption	Implemented power management circuits.

Conclusion

This project successfully demonstrates a low-cost, innovative solution to enhance the mobility and safety of visually impaired individuals. Its modular design ensures adaptability and scalability for future improvements.