## **TITLE OF THE PROJECT**

## SMART ALERT SYSTEM FOR HEARING IMPAIRMENTS

**Team Members:**

| **Sl. No** | **Name** | **SRN** |
| --- | --- | --- |
| 01 | N Riddhi samitha | PES1UG23AM175 |
| 02 | N V MANYA | PES1UG23AM176 |
| 03 | Navyashree J | PES1UG23AM180 |

## Components used:

* PIR Sensor
* Arduino Uno
* Sound Sensor
* 4\*4 matrix keypad
* Led
* 1000 ohm Resistor

**Explanation about the project :**

### Smart Alert System for Hearing Impairments

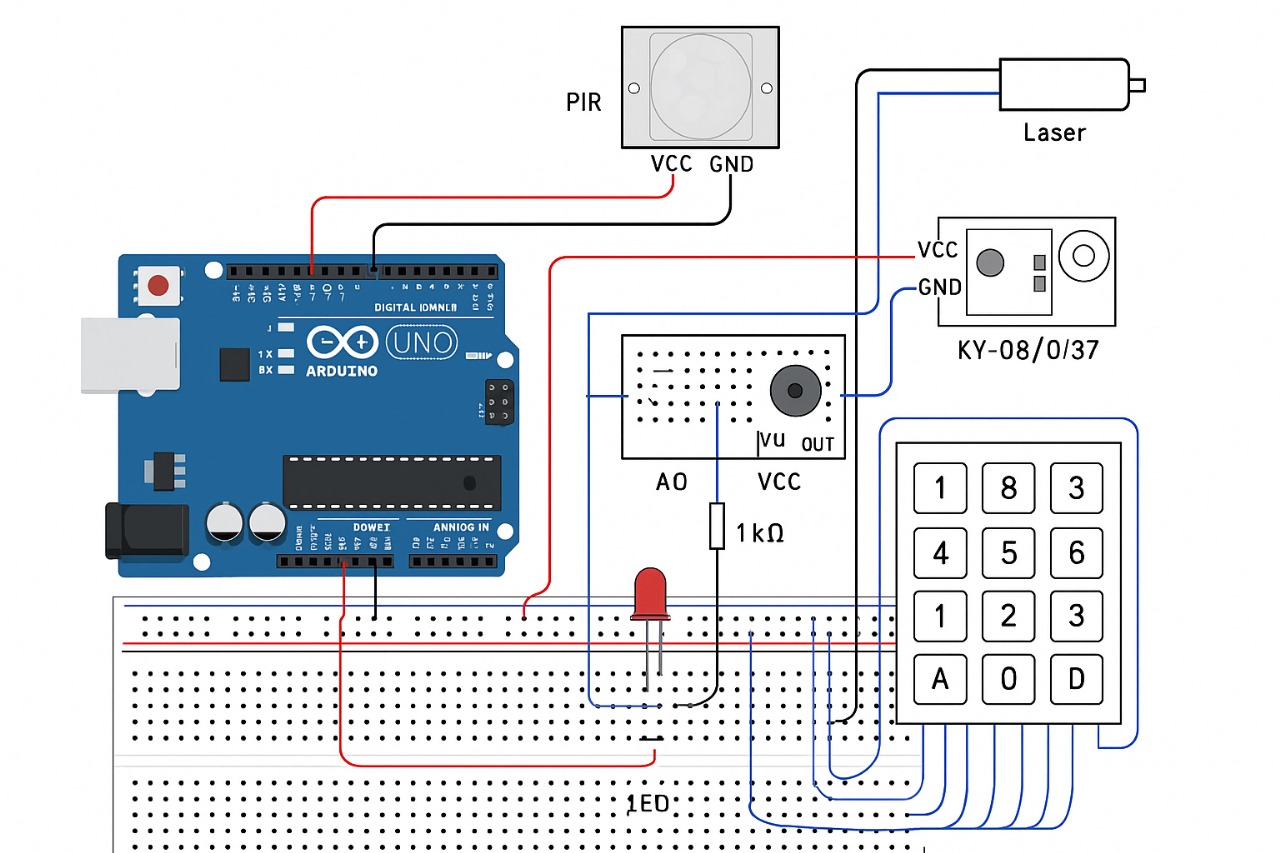
It is a **smart alert system** designed to assist individuals with hearing impairments by providing **wireless alerts**. The system integrates **motion detection**, **sound detection**, and **user interaction** to notify the user about certain environmental events, ensuring they are aware of potential changes or hazards in their surroundings without needing to hear sound-based alarms.

This system uses various sensors to detect events like movement or noise and provides feedback through a **display screen**. Additionally, the user can interact with the system via a **4x4 matrix keypad** to control settings or activate different features.

1. **Motion Detection (PIR Sensor):**
   * The PIR sensor detects motion (e.g., a person entering the room) and triggers the wristband to vibrate.
2. **Sound Detection (Sound Sensor):**
   * The sound sensor detects loud noises (e.g., doorbell, knock) and triggers the wristband to vibrate if the noise exceeds a set threshold.
3. **Keypad Control:**
   * The 4x4 matrix keypad allows the user to interact with the system, adjust the sound threshold, turn the alert system on/off, or change settings.

This system helps hearing-impaired individuals stay aware of their environment through vibrational alerts, providing a practical, non-hearing-based notification method.

**Circuit Diagram:**



**Arduino Component Connections:**

### Connecting Arduino to Breadboard

#### Powering the Breadboard from Arduino

* Arduino 5V → Breadboard Red Power Rail (+)
* Arduino GND → Breadboard Blue/Black Ground Rail (-)

### LED Wiring

* Anode (+, Long Leg) → Breadboard
* 220Ω Resistor → Digital Pin 5 on Arduino
* Cathode (-, Short Leg) → Breadboard (Ground)

### Matrix Keypad Wiring

* **Rows** → Digital Pins **6, 7, 8, 9**
* **Columns** → Digital Pins **10, 11, 12, 13**

##### PIR Sensor Wiring

* VCC → Breadboard (+)
* GND → Breadboard (-)
* OUT → Digital Pin 2 on Arduino

### Sound Sensor Wiring

* **VCC** → Arduino **5V**
* **GND** → Arduino **GND**
* **DO** → Digital Pin **3** on Arduino

**Pseudocode:**

1. **Start‑up (runs once)**
   * Tell the Arduino which pins are connected to:  
     + the motion sensor (PIR)
     + the LED
     + the sound sensor
     + the 4 × 4 keypad
   * Open the USB/Serial connection so you can read messages on your computer.
   * Print **“System Ready”** to show everything is set up.
2. **Endless loop (runs over and over)** Every pass through the loop the board asks three simple questions:  
     
     A. “Did I see motion?”  
   * Look at the PIR motion‑sensor pin.
   * If it’s HIGH (means movement was detected):  
     + Print **“Motion Detected!”** on the computer screen.
     + Turn the LED on for half a second so someone nearby also sees a flash.
     + Turn the LED off again.

 B. “Did I hear a loud sound?”

* + Check the sound‑sensor pin.
  + If it’s HIGH (means a clap, shout, knock, etc. was heard):  
    - Print **“Sound Detected!”** on the screen.
    - Wait half a second (gives the sensor a moment to settle).

 C. “Did someone press a keypad button?”

* + Ask the keypad object for the most recent key that was pressed.
  + If a key was pressed:  
    - Show **“Key Pressed: X”** (X is the key) on the screen.
    - If that key is a number 0‑9:  
      * Look up a canned phrase in a list:  
         *0→Hello, 1→I need help, … 9→Goodbye*
      * Print that phrase on the screen.
    - If the key is anything else (A, B, C, D, \*, #):  
      * Print **“Invalid Key”**.
    - Pause 2 seconds so rapid key hits don’t flood the screen.

And then the loop starts again, forever, watching for motion, sound, or keypad presses.

**Code:**

**#include <Wire.h>**

**#include <Keypad.h>**

**// Sensor pins**

**const int pirPin = 2;**

**const int ledPin = 5;**

**const int soundSensorPin = 3; // Sound sensor digital output**

**// Keypad setup**

**const byte ROWS = 4;**

**const byte COLS = 4;**

**char keys[ROWS][COLS] = {**

**{'1', '2', '3', 'A'},**

**{'4', '5', '6', 'B'},**

**{'7', '8', '9', 'C'},**

**{'\*', '0', '#', 'D'}};**

**byte rowPins[ROWS] = {6, 7, 8, 9};**

**byte colPins[COLS] = {10, 11, 12, 13};**

**Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);**

**// Messages for keypad inputs**

**String messages[10] = {**

**"Hello", "I need help", "How are you", "I am hard of hearing",**

**"Bathroom", "Emergency", "I am fine", "Thank you",**

**"Where are you?", "Goodbye"};**

**void setup() {**

**pinMode(pirPin, INPUT);**

**pinMode(ledPin, OUTPUT);**

**pinMode(soundSensorPin, INPUT);**

**Serial.begin(9600);**

**Serial.println("System Ready");**

**}**

**void loop() {**

**// Check PIR sensor**

**if (digitalRead(pirPin) == HIGH) {**

**Serial.println("Motion Detected!");**

**digitalWrite(ledPin, HIGH);**

**delay(500);**

**digitalWrite(ledPin, LOW);**

**}**

**// Check sound sensor**

**if (digitalRead(soundSensorPin) == HIGH) {**

**Serial.println("Sound Detected!");**

**delay(500);**

**}**

**// Check keypad input**

**char key = keypad.getKey();**

**if (key) {**

**Serial.print("Key Pressed: ");**

**Serial.println(key);**

**if (key >= '0' && key <= '9') {**

**Serial.println(messages[key - '0']);**

**} else {**

**Serial.println("Invalid Key");**

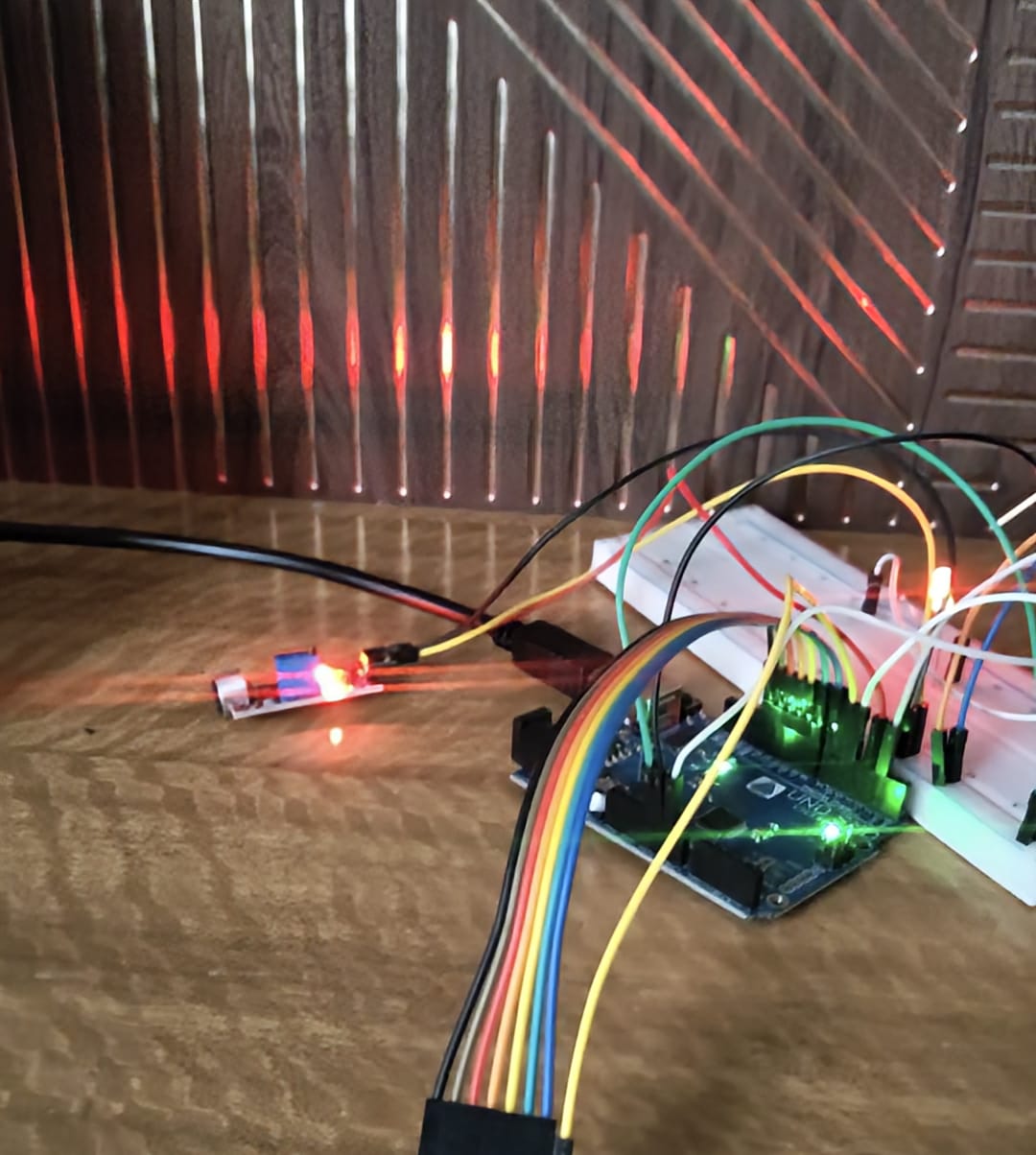
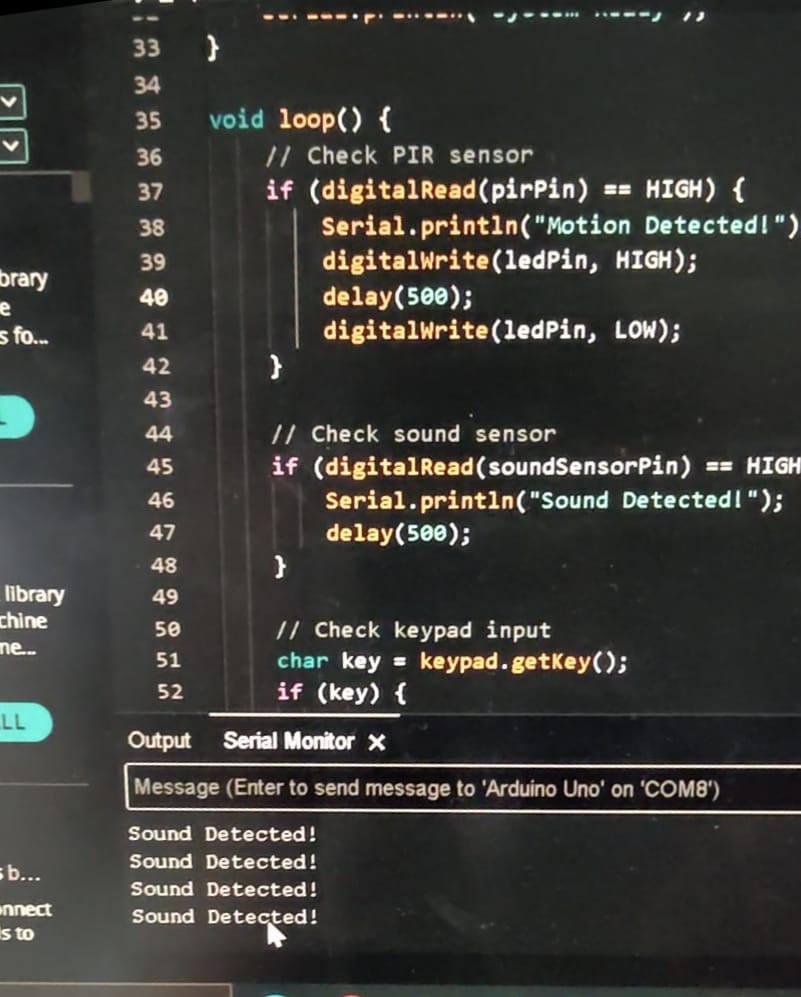
**}**

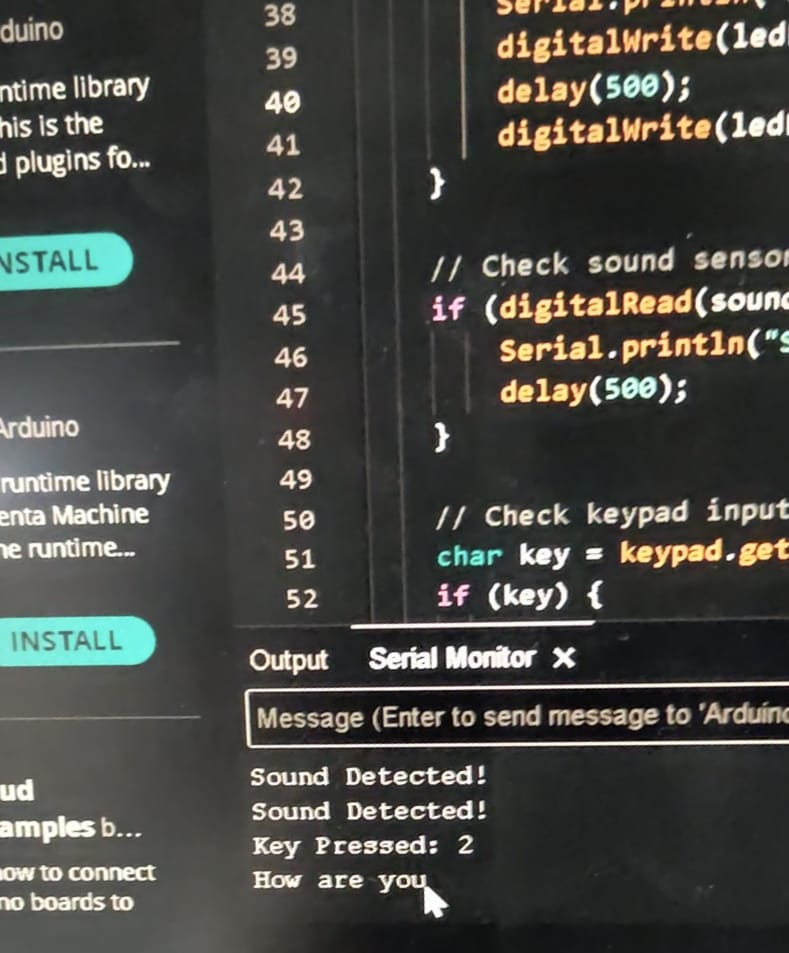
**delay(2000);**

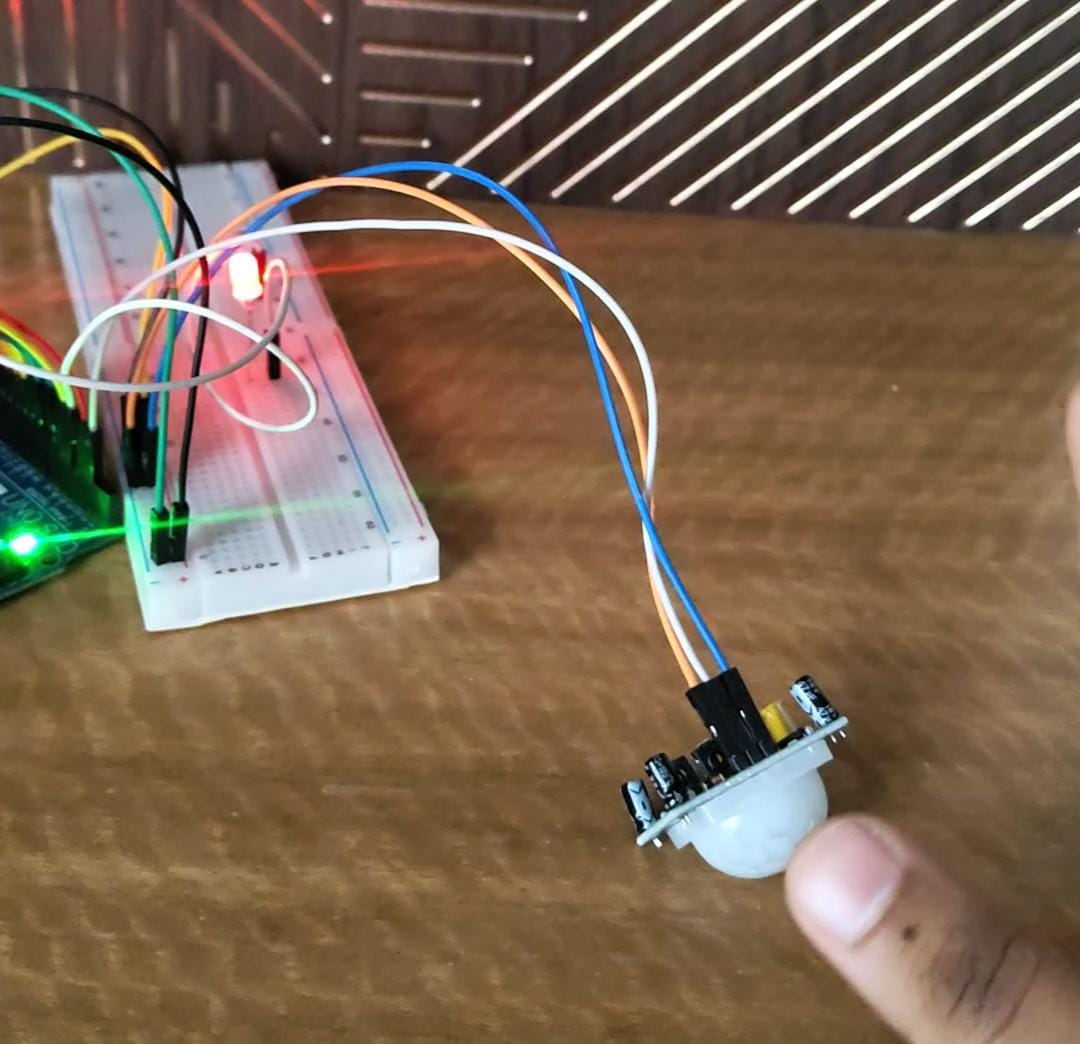
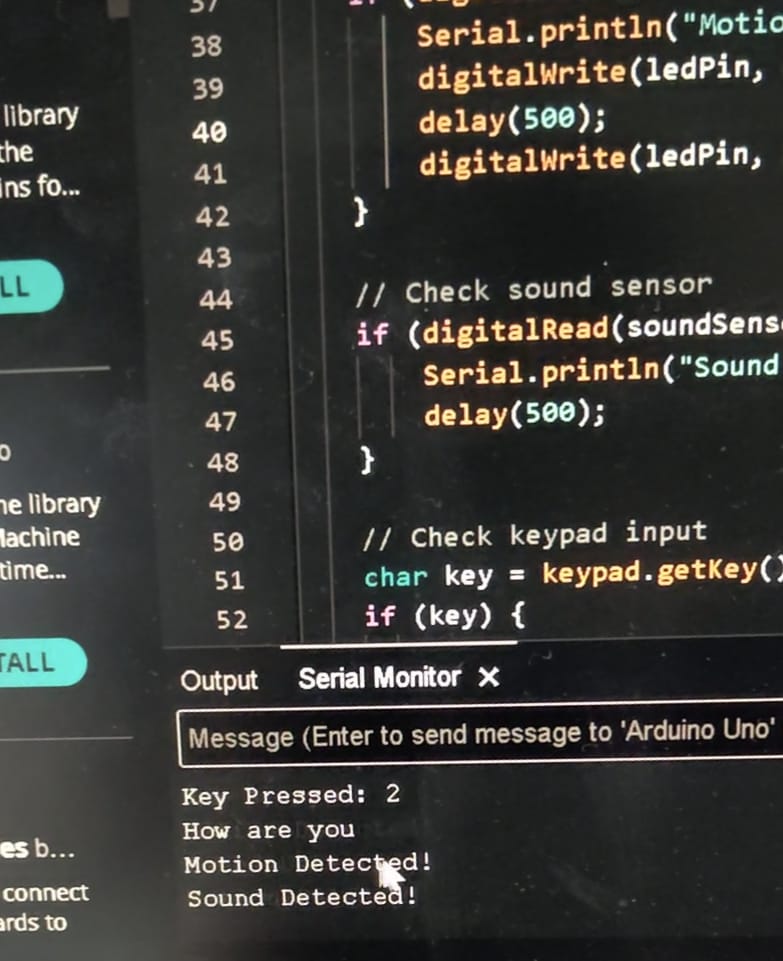
**}**

**}**

**Simulation Output:**

** **

** **

** **