**VibeNav – A Wearable Tactile Navigation and Safety System for Visually Impaired Users**

**Problem:**

Visually impaired individuals struggle to sense obstacles (especially overhead or moving ones), get lost, or face dangers (like fire, heat, or falling) without a way to alert others.

**Goal:**

Build a **wearable cap/headband** that:

* Senses obstacles and alerts user via **vibration**
* Detects **falls**, **heat/fire**, and sends **emergency SMS with GPS location**
* Is **non-auditory**, **low-cost**, **offline**, and **compact**

**Prototype Thinking:**

We'll break it down into core functionalities and map each one to the components.

| **Feature** | **Component** | **Role** |
| --- | --- | --- |
| Obstacle detection (front/side) | 2× HC-SR04 | Measure distance |
| Tactile feedback | 4× vibration motors | Vibrate based on proximity (direction-based) |
| Fall detection | MPU6050 | Detect sudden tilt/acceleration |
| Emergency alert | SIM800L | Send SMS with alert text |
| Location tracking | NEO-6M GPS | Get current latitude/longitude |
| Heat detection | DS18B20 | Warn if temperature too high |
| Fire detection | IR Flame Sensor | Detect presence of fire nearby |
| Optional control | HC-05 Bluetooth | Manual control or mobile alerts |
| Power | TP4056 + Booster | Charge + Boost battery to 5V |
| MCU | ESP32 or ESP8266 | Brain (reads data, sends alerts, controls motors) |

**Functional Thinking** – **What Each Module Does:**

**1. Ultrasonic Sensors (HC-SR04 x2)**

* Mounted **front & side**
* Detects if any object is < 1m
* Sends signal to corresponding vibration motor

**2. Vibration Motors (x4)**

* Mounted on front, back, left, right of headband
* Vibrate more strongly as the object gets closer

**3. MPU6050 Gyro + Accel**

* Continuously checks for **sudden acceleration or angle tilt**
* Triggers **fall alert**

**4. SIM800L + GPS**

* If fall is detected → gets location from GPS
* Sends **SMS** to predefined number (e.g. parent/guardian)

**5. DS18B20 + Flame Sensor**

* If **temp > 45°C** or flame detected → triggers a **vibration alert**
* Can also send SMS if configured

**6. ESP32 / ESP8266 / NodeMCU**

* Collects sensor data
* Makes decisions: when to vibrate, when to send SMS
* Stores emergency number in code (can be later updated via Bluetooth)

**7. Power System**

* TP4056 for charging 3.7V battery
* XL6009 boost converter to get 5V for motors & logic

**Development Plan**

**🛠️ Week-by-Week Breakdown (We can adjust as needed)**

| **Week** | **Task** |
| --- | --- |
| Week 1 | Setup ESP32 + test ultrasonic sensors + vibration motors |
| Week 2 | Integrate MPU6050 + simulate fall detection |
| Week 3 | Interface SIM800L → send SMS from ESP32 |
| Week 4 | Connect GPS + send SMS with location |
| Week 5 | Add temperature + flame sensor logic |
| Week 6 | Integrate everything → build prototype on cap/headband |
| Week 7 | Testing + optimization |
| Week 8 | Documentation, report |

**Prototype Sketch (Concept)**

Imagine a **cloth headband or baseball cap** with:

* **Front**: 1 ultrasonic sensor + 1 vibration motor
* **Left side**: 1 ultrasonic sensor + vibration motor
* **Back**: Vibration motor
* **Right side**: Vibration motor
* **Inside cap**: MPU6050, ESP32, flame/temp sensors
* **At the back of cap**: SIM800L + GPS
* **Power pack**: small Li-ion battery + TP4056

**Hardware Connection Overview**

**✅ Modules Used:**

* ESP32 (Development Board)
* 1-Channel Relay Module
* MB102 Breadboard Power Supply Module
* 9V Battery (to power MB102)
* Ultrasonic Sensor (HC-SR04)
* Vibration Motor (controlled via relay)
* Jumper Wires
* Optional: 2.2kΩ resistor (for signal stability)

**Power Supply**

| **Component** | **Connection** |
| --- | --- |
| **9V Battery** | Connected to the barrel jack of **MB102 Power Module** |
| **MB102 Output Switch** | Set to **5V** |
| **MB102 5V Pin** | Used to power **Relay Module VCC** and other components |
| **MB102 GND Pin** | Shared GND across **ESP32**, **Relay**, and **Ultrasonic Sensor** |

✅ **Important:** Make sure the **GND of MB102** and **GND of ESP32** are connected. A common ground is critical for reliable signal operation.

**📟 Relay Module Connections**

| **Relay Pin** | **Connection** |
| --- | --- |
| **VCC** | MB102 **5V output** |
| **GND** | MB102 **GND** |
| **IN** | **ESP32 GPIO 18** (used to control the relay) |

⚠️ Relay modules are typically **active LOW**:  
digitalWrite(GPIO, LOW) turns **relay ON**  
digitalWrite(GPIO, HIGH) turns **relay OFF**

**📏 Ultrasonic Sensor (HC-SR04)**

| **Sensor Pin** | **Connection** |
| --- | --- |
| **VCC** | MB102 **5V** |
| **GND** | MB102 **GND** |
| **TRIG** | ESP32 **GPIO 4** |
| **ECHO** | ESP32 **GPIO 5** |

**💥 Coin Vibration Motor**

* Powered through the relay’s **NO (Normally Open)** and **COM** pins
* External 5V from MB102 powers the motor
* When the relay is **ON**, the circuit completes and vibration motor runs
* When **OFF**, motor disconnects completely

| **Relay Terminal** | **Connection** |
| --- | --- |
| **NO (Normally Open)** | One end of vibration motor |
| **COM** | MB102 **5V** or external 5V |
| **Motor GND** | MB102 **GND** |

**🧠 Optional: Pull-Down Resistor**

If relay doesn’t switch cleanly (stays dim when OFF), add a **2.2kΩ resistor**:

* One end of resistor → **GND**
* Other end → Relay **IN pin** (same node as GPIO 18)

This ensures the IN pin is pulled LOW clearly when needed.

**How to Run**

Step-by-step:

1. Upload code to ESP32 using Arduino IDE.
2. Power MB102 using a 9V battery.
3. Connect everything as per the diagram.
4. Open Serial Monitor (baud rate: 115200).
5. Test the sensor by placing an object within range.