**Tactile Navigation Stick for the Visually Impaired**

1. **Abstract :**

This project focuses on the development of a wearable assistive navigation device aimed at improving spatial awareness and mobility for visually impaired individuals. The system is implemented using an ESP32 (or Arduino Nano) microcontroller housed within a lightweight and ergonomic walking stick design.

The device is equipped with ultrasonic or infrared proximity sensors positioned to scan the area in three directions—forward, left, and right—providing continuous monitoring of the user’s surroundings within a typical range of 1 to 2 meters. Based on detected proximity, the system delivers real-time haptic feedback through coin-type vibration motors embedded in the handle. The vibration intensity conveys the distance of obstacles—strong vibrations for nearby obstacles, mild for medium-range, and none when the path is clear. Additionally, directional feedback is provided through motor placement: obstacles ahead activate the central motor, while those on the left or right trigger the corresponding side motors. This allows the user to intuitively perceive their environment without relying on auditory cues.

Powered by a compact battery with onboard power management, the device is designed for quiet operation and remains effective in low-light or noisy environments. By avoiding dependence on vision or sound, it ensures uninterrupted spatial feedback with minimal distraction.

The modular structure of the system supports future enhancements, including Bluetooth Low Energy (BLE) connectivity for remote alerts, and integration of lightweight machine learning models for advanced object detection and indoor navigation. Overall, the device offers a practical and scalable solution to enhance the independence, confidence, and safety of visually impaired users across various environments.

**2. Objective**

The primary goal is to design and prototype a multi-sensor smart stick for the visually impaired that:

* Detects nearby obstacles in real time and communicates them via vibration motors.
* Recognizes falls, fire, or high heat, and sends SMS alerts with GPS location.
* Operates silently, without the need for sound-based or screen-based feedback.
* Functions offline, ensuring reliability in all environments.

**Goal:**

Build a **tactical stick** 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**

**3. System Overview**

Core Features and Functions

| Feature | Description |
| --- | --- |
| Obstacle Detection | Uses ultrasonic sensors to detect objects ahead and to the sides. |
| Haptic Feedback | Vibration motors provide direction-based, intensity-controlled tactile feedback. |
| Fall Detection | An accelerometer (MPU6050) monitors sudden changes to detect falls. |
| Emergency Alerts | SIM800L sends SMS messages with real-time GPS data if a fall or fire is detected. |
| Location Tracking | NEO-6M GPS module provides accurate location coordinates. |
| Heat/Fire Detection | DS18B20 sensor for temperature, IR sensor for fire/flame detection. |
| Power Supply | Rechargeable battery with TP4056 and boost converter. |
| Microcontroller | ESP32 (preferred for its wireless and multitasking capabilities). |

**4. Hardware Components**

| Component | Purpose |
| --- | --- |
| ESP32 | Main microcontroller for handling all sensor inputs and control logic |
| HC-SR04 (×2) | Ultrasonic sensors for front and side obstacle detection |
| Vibration Motors (×4) | Provide tactile feedback based on object distance and direction |
| MPU6050 | Accelerometer and gyroscope for fall detection |
| SIM800L | GSM module for sending emergency SMS alerts |
| NEO-6M GPS | GPS module for acquiring current location coordinates |
| DS18B20 | Temperature sensor to detect excessive heat |
| IR Flame Sensor | Detects fire or flame presence nearby |
| TP4056 + Boost Converter | Charges 3.7V Li-ion battery and steps up voltage to 5V |

**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

**5. Circuit and Connectivity**

**Power Supply**

3.7V Li-ion Battery

→ connected to TP4056 charging module

TP4056 OUT+ → 5V step-up module input

Step-up module output (5V)

→ connected to ESP32 VIN (or 5V)

→ power line for sensors/modules

TP4056 GND & Step-up GND → common GND

**ESP32 Microcontroller**

All sensors and modules are connected to ESP32 GPIO pins, with common GND.

**Obstacle Detection – Ultrasonic Sensors (HC-SR04)**

Front HC-SR04

VCC → 5V

GND → GND

Trig → ESP32 GPIO 12

Echo → ESP32 GPIO 13

Left HC-SR04

VCC → 5V

GND → GND

Trig → ESP32 GPIO 14

Echo → ESP32 GPIO 27

Right HC-SR04 (optional, same pattern)

Trig → ESP32 GPIO 26

Echo → ESP32 GPIO 25

**Haptic Feedback – Vibration Motors (via NPN Transistor like 2N2222)**

Use transistor + flyback diode + resistor for each motor control.

Center Vibration Motor (for front obstacle)

Base → ESP32 GPIO 32 (via 1kΩ resistor)

Emitter → GND

Collector → Negative of motor

Motor positive → 5V

Flyback diode across motor

Left Motor → ESP32 GPIO 33

Right Motor → ESP32 GPIO 15

(Optional: Intensity levels via PWM)

**Fall Detection – MPU6050 (I2C)**

VCC → 3.3V

GND → GND

SCL → ESP32 GPIO 22

SDA → ESP32 GPIO 21

**Temperature Sensor – DS18B20 (1-Wire)**

VCC → 3.3V

GND → GND

Data → ESP32 GPIO 4

Pull-up resistor (4.7kΩ) between Data & VCC

**Fire Detection – IR Flame Sensor**

VCC → 3.3V or 5V (based on module)

GND → GND

DO (Digital Out) → ESP32 GPIO 18

(Optional: use AO if you want analog readings)

**GPS Module – NEO-6M (UART)**

VCC → 3.3V (or 5V if allowed)

GND → GND

TX → ESP32 RX2 (GPIO 16)

RX → ESP32 TX2 (GPIO 17)

(Use SoftwareSerial if needed on other boards)

**SIM800L – GSM Module (UART)**

VCC → 4.0V (Use AMS1117 or DC buck converter)

GND → GND

TX → ESP32 RX (GPIO 3)

RX → ESP32 TX (GPIO 1)

Note: Needs level shifting (ESP32 is 3.3V, SIM800L is 2.8V safe)

Additional Notes:

* **All GNDs** must be **connected together** (common ground).
* Use **capacitors** (10uF or 100uF) near SIM800L and GPS for power stability.
* If ESP32 pin conflict occurs, change GPIOs in your code accordingly.
* **Ultrasonic sensors** connect via digital pins to the ESP32 and trigger directional motors.
* **MPU6050** and **GPS** use I2C and UART respectively.
* **SIM800L** also communicates via UART.
* **DS18B20** uses 1-wire communication.
* All components are powered via a regulated 5V from the booster circuit.

**6. Working Principle**

1. **Obstacle Detection & Feedback**:  
   The ultrasonic sensors continuously scan in front and side directions. The ESP32 calculates distance values and activates corresponding vibration motors with intensity proportional to proximity.
2. **Fall & Emergency Detection**:  
   Sudden drops or abnormal tilt angles detected by MPU6050 are treated as falls. Combined with flame or high-temperature detection, the system composes a real-time GPS-based alert and sends it via SMS to a predefined number.
3. **Offline & Silent Operation**:  
   Unlike voice-based systems, this device avoids sound entirely, allowing use in noisy environments or with users who prefer non-auditory feedback. All sensing and response are offline.

**7. Software Implementation**

The system is coded using Arduino IDE with libraries:

* Wire.h, SoftwareSerial.h, TinyGPSPlus.h, Adafruit\_MPU6050.h, DallasTemperature.h, etc.

Core features in software:

* Obstacle sensing loop every 200ms.
* Interrupt-driven fall detection.
* SMS formatting: "ALERT! Fall/Fire detected at: Lat, Long"
* Temp/fire checked every 1s with thresholds (e.g., temp > 50°C).
* Vibration motors controlled via PWM or digital write.

**8. Advantages**

* **Affordable**: Uses widely available low-cost modules.
* **Offline support**: No dependency on internet or cloud.
* **Compact**: ESP32 and compact modules make it ideal for embedding in a stick.
* **Silent & discreet**: Does not draw attention, ideal for privacy-conscious users.
* **Expandable**: BLE, AI models, or camera modules can be added later.

**9. Limitations**

* No obstacle classification (just proximity).
* GPS may struggle indoors or in urban canyons.
* SIM800L requires GSM network availability.

**10. Future Enhancements**

* **BLE Integration**: Add a phone app for caregiver alerts.
* **ML Object Detection**: Add visual camera and on-device ML for obstacle classification.
* **Indoor Mapping**: Incorporate RFID beacons or UWB for better indoor navigation.
* **Energy Optimization**: Solar charging or advanced power management.

**11. Conclusion**

The tactile navigation stick offers a reliable, discreet, and modular solution for the visually impaired. Through a combination of ultrasonic sensing, haptic feedback, emergency communication, and environmental monitoring, it significantly enhances the safety and independence of its users. Its design ensures adaptability for future tech integrations while remaining practical for everyday use.