Community Safety Alert System

Overview

The **Community Safety Alert System** is an Intel Galileo-powered project designed to enhance public safety. This system integrates advanced sensors and real-time monitoring capabilities to detect emergencies and provide timely alerts. It also features a directional guidance system to assist individuals in danger by guiding them to safer areas.

Features

1. Accident Detection

- Utilizes advanced sensors to detect accidents or emergencies in real time.
- Analyzes impact severity for accurate response.

2. Directional Guidance System

- Provides real-time instructions to guide individuals away from dangerous areas.
- Uses GPS and other location-based services for accurate guidance.

3. Real-Time Monitoring

- Continuously monitors the environment for potential hazards.
- Updates emergency contacts and individuals as the situation evolves.

4. Sensor Integration

- Equipped with sensors such as accelerometers, GPS modules, and vibration sensors.
- Collects data to identify emergencies effectively.

5. Scalability

- Adaptable for various use cases, including public transportation systems, buildings, and open areas.

Hardware Requirements

- **Intel Galileo Board**
- Accelerometer (e.g., ADXL345)
- GPS Module

- Vibration Sensor
- Buzzer or Alert Module
- Power Supply
- Breadboard and Jumper Wires

Software Requirements

- Arduino IDE
- Python for advanced data processing (optional)
- Libraries:
    - Adafruit_Sensor
    - TinyGPS++ (for GPS module)

Code

Below is the sample code for the project:

```
#include <Wire.h>

#include <Adafruit_Sensor.h>

#include <Adafruit_ADXL345_U.h>

#include <TinyGPS++.h>

#include <SoftwareSerial.h>


// Initialize components

Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);

TinyGPSPlus gps;

SoftwareSerial gpsSerial(4, 3); // RX, TX


void setup() {

 Serial.begin(9600);

 gpsSerial.begin(9600);
```

```
 // Initialize accelerometer
 if (!accel.begin()) {
  Serial.println("Could not find a valid ADXL345 sensor, check wiring!");
  while (1);
 }
 Serial.println("Accelerometer initialized!");

 // Buzzer setup
 pinMode(8, OUTPUT); // Buzzer connected to pin 8
}

void loop() {
 sensors_event_t event;
 accel.getEvent(&event);

 // Accident detection logic
 if (abs(event.acceleration.x) > 15 || abs(event.acceleration.y) > 15 ||
abs(event.acceleration.z) > 15) {
  Serial.println("Accident detected!");
  digitalWrite(8, HIGH); // Activate buzzer
  delay(1000);
  digitalWrite(8, LOW);
 }

 // GPS data reading
 while (gpsSerial.available() > 0) {
  if (gps.encode(gpsSerial.read())) {
```

```
    Serial.print("Location: ");
    Serial.print(gps.location.lat(), 6);
    Serial.print(", ");
    Serial.println(gps.location.lng(), 6);
   }
 }

 delay(100);
}
```

How It Works

1. **Initialization**: The system initializes sensors and begins monitoring for emergency scenarios.
2. **Accident Detection**: When the accelerometer detects sudden movements or high impact values, it triggers the alert system.
3. **Alert Mechanism**: A buzzer or other alert module is activated to notify nearby individuals.
4. **Directional Guidance**: GPS data is used to provide instructions to move toward safer areas.

Installation Guide

1. Assemble the circuit as per the diagram.
2. Upload the provided code to the Intel Galileo board using Arduino IDE.
3. Power the system and test it in a controlled environment.

Future Improvements

- Integration with IoT platforms for remote monitoring.
- Enhanced user interface for real-time feedback.
- Machine learning algorithms to predict potential hazards.