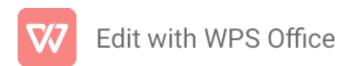
```
#include <TinyGPSPlus.h>
#include <HardwareSerial.h>
#include <WiFi.h>
#include < PubSubClient.h>
#include <DHT.h>
#include "ThingSpeak.h"
#define DHTTYPE DHT11
char ssid[] = "MohammedSohail"; // your network SSID (name)
char pass[] = "12345678"; // your network password
int kevIndex = 0:
                            // your network key Index number (needed only for WEP)
WiFiClient client;
unsigned long myChannelNumber = 2492483;
const char * myWriteAPIKey = "OJRKZAK7C9GB3T8W";
String Lat, Lng;
unsigned long prevMillis, interval = 300; // lockout time in ms
const int vibrationSensorPin = 34; // Pin for vibration sensor
const int fireSensorPin = 33; // Pin for fire sensor
const int dhtPin = 18;
HardwareSerial ss(2); // Use Serial2 on ESP32
HardwareSerial sim800(1); // Use Serial2 on ESP32
// The TinyGPSPlus object
TinvGPSPlus aps:
DHT dht(dhtPin, DHTTYPE);
int count = 1;
void setup()
  Serial.begin(115200);
  ss.begin(9600, SERIAL_8N1, 16, 17);
  sim800.begin(9600, SERIAL_8N1, 4, 2);
  Serial.println("SIM800L serial initialize");
  pinMode(vibrationSensorPin, INPUT_PULLUP); // Set vibration sensor pin as input
  pinMode(fireSensorPin, INPUT); // Set fire sensor pin as input
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client); // Initialize ThingSpeak
  Serial.println("Connected to WiFi");
  dht.begin();
}
void loop()
  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using
```



```
open or WEP network
       Serial.print(".");
       delay(5000);
    Serial.println("\nConnected.");
  }
  // This sketch displays information every time a new sentence is correctly encoded.
  while (ss.available() > 0)
    if (gps.encode(ss.read()))
       displayInfo();
  int vibrationValue = digitalRead(vibrationSensorPin); // Read vibration sensor value
  int fireValue = digitalRead(fireSensorPin); // Read fire sensor value
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  ThingSpeak.setField(1, h);
  ThingSpeak.setField(2, t);
  ThingSpeak.setField(3, vibrationValue);
  ThingSpeak.setField(4, fireValue);
  // Print sensor readings to serial monitor
  Serial.println("Temp " + String(t));
  Serial.println("Humid" + String(h));
  Serial.print("Vibration Sensor: ");
  Serial.println(vibrationValue);
  Serial.print("Fire Sensor: ");
  Serial.println(fireValue);
  if (millis() - prevMillis > interval) {
    if (digitalRead(vibrationSensorPin)) { // if LOW
       Serial.println(count);
       prevMillis = millis(); // reset
       count++:
    }
  if(count >= 1){
    count = 0;
    Serial.println("Cathc");
    sendSMS("Road Accident");
    delay(5000);
  }
  if(fireValue == 0){
    sendSMS("Fire Accident");
    delay(1000);
  }
  if (gps.location.isValid())
     int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    if(x == 200){
        //Serial.println("Channel update successful.");
      else{
      //Serial.println("Problem updating channel. HTTP error code " + String(x));
```



```
delay(5000);
  delay(500);
void displayInfo(){
  if (gps.location.isValid())
    Lat = String(gps.location.lat(), 6);
    Lng = String(gps.location.lng(), 6);
    Serial.print(gps.location.lat(), 6);
    Serial.print(F(","));
    Serial.print(gps.location.lng(), 6);
  }
  else
  {
    Serial.print(F("INVALID"));
  Serial.println();
}
void sendSMS(String mesg){
  Serial.println("Initializing...");
  delay(1000);
  sim800.println("AT"); //Once the handshake test is successful, it will back to OK
  updateSerial();
  sim800.println("AT+CMGF=1"); // Configuring TEXT mode
  updateSerial();
  sim800.println("AT+CMGS=\"+918519806756\"");
  updateSerial();
  sim800.println(mesg
                             +
                                   "Detected
                                                        please
                                                                    tap
                                                                            the
                                                                                     link
                                                                                              to
location:\nhttps://www.google.com/maps/search/?api=1&query=" + Lat + "," + Lng); //text
content
  updateSerial();
  sim800.write(26);
  Serial.println("Sent Successfully");
  delay(5000);
}
void updateSerial(){
  delay(500);
```