ESP32-based GPS and Satellite SOS System

```
#include <TinyGPS++.h>
#include <HardwareSerial.h>
#include <Wire.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>

TinyGPSPlus gps;
HardwareSerial gpsSerial(1); // GPS on Serial1 (ESP32)
Adafruit_MPU6050 mpu;

// Satellite modem (TX=17, RX=16)
HardwareSerial satSerial(2);

const int buttonPin = 4; // Manual SOS button

bool accidentDetected = false;
float accelThreshold = 2.5; // g-force threshold for crash

void setup() {
  Serial.begin(115200);
  gpsSerial.begin(9600, SERIAL_8N1, 18, 19); // RX=18, TX=19
  satSerial.begin(19200, SERIAL_8N1, 16, 17); // Satellite modem
  pinMode(buttonPin, INPUT_PULLUP);

  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1);
  }
  Serial.println("MPU6050 initialized");
}

void loop() {
  while (gpsSerial.available()) {
    gps.encode(gpsSerial.read());
  }

  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);
  float gForce = sqrt(a.acceleration.x * a.acceleration.x +
                      a.acceleration.y * a.acceleration.y +
                      a.acceleration.z * a.acceleration.z);

  if (gForce > accelThreshold) {
    accidentDetected = true;
    sendEmergencyMessage("AUTO");
  }

  if (digitalRead(buttonPin) == LOW) {
    sendEmergencyMessage("MANUAL");
    delay(1000); // Debounce
  }
```

```
  delay(200);
}

void sendEmergencyMessage(String triggerType) {
  String message = "AERSVS ALERT [" + triggerType + "]\n";

  if (gps.location.isValid()) {
    message += "Lat: " + String(gps.location.lat(), 6);
    message += ", Lon: " + String(gps.location.lng(), 6);
  } else {
    message += "Location: UNAVAILABLE";
  }

  satSerial.println(message);
  Serial.println("Sent via satellite: ");
  Serial.println(message);
  accidentDetected = false;
}
```

Implementation Steps

1. Setup Hardware:

   - Connect GPS and satellite modem to ESP32.

   - Connect MPU6050 via I2C.

   - Add a push button to pin 4 for manual SOS.

2. Upload the Code:

   - Use Arduino IDE.

   - Select ESP32 board and correct COM port.

3. Test:

   - Simulate crash by shaking the device.

   - Press the button to manually trigger SOS.

4. Expansion:

   - Add real-time sensors, LoRa modules, or visual alerts for robustness.