SMART HELMET ACCIDENT ALERT SYSTEM

**Introduction:**The Smart Helmet Accident Alert System is an innovative IoT and Android-based safety solution developed to enhance rider safety in the event of a vehicular accident. It integrates a motion-sensing module, MPU6050 (accelerometer + gyroscope), with an ESP32 microcontroller, mounted inside a helmet. The system continuously monitors the rider’s motion parameters and detects sudden impacts or abnormal acceleration patterns that may indicate an accident.

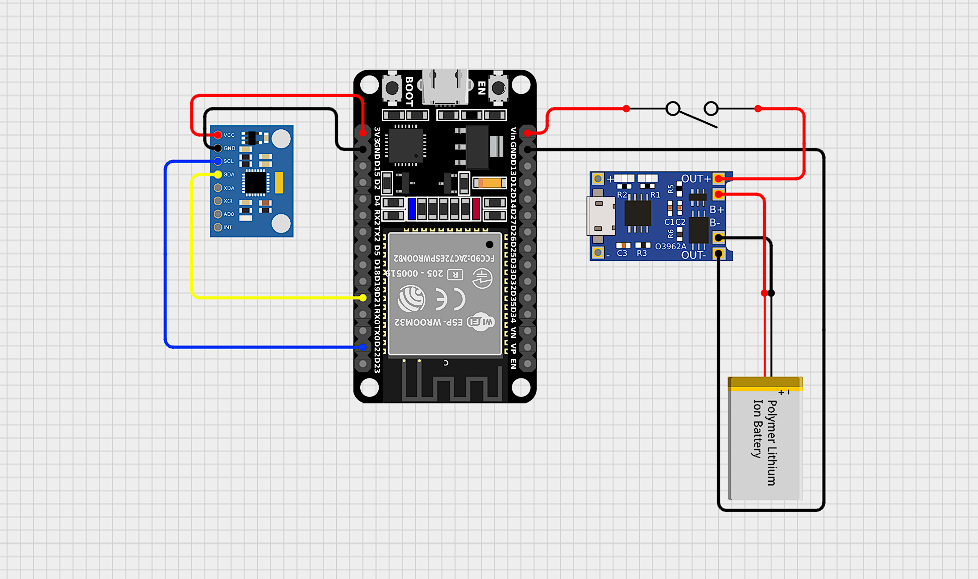
Upon detecting such an event, the ESP32 sends an emergency signal to the rider’s Android smartphone via Bluetooth. The mobile application, upon receiving this signal, automatically retrieves the GPS location of the rider and sends an emergency SMS to a predefined guardian's mobile number. This ensures that help can be dispatched quickly, even if the rider is unconscious or unable to use the phone.

Additionally, a Guardian Android App has been developed, which receives the alert SMS and triggers a siren , grabbing attention immediately. This dual-application architecture ensures that the accident alert is delivered reliably and promptly, improving response time and potentially saving lives.

**Objective:**

* Accurately detect crash impact using the MPU6050 sensor integrated with the ESP32 microcontroller.
* Automatically send an emergency alert SMS containing the rider’s GPS location to a predefined guardian contact.
* Notify the guardian through an audible siren and on-screen pop-up via the dedicated Guardian mobile application.

**System Architecture:**

****

**Hardware Requirements:**

1. ESP32 board
2. MPU6050 Accelerometer + Gyroscope
3. Li-Ion battery
4. Switch
5. TP4056 charging module
6. Helmet (prototype mount)

**Software Requirements:**

1. Arduino IDE
2. Android Studio

**Implementation:**

* **ESP32 code:**
* Continuously reads acceleration and gyroscope data from the MPU6050 sensor.
* Computes the magnitude of acceleration and compares it against a threshold to detect a possible accident.
* When impact is detected, sends a predefined alert message ("ACCIDENT DETECTED") via BluetoothSerial to the connected smartphone.

**Code:**

#include <Wire.h>

#include "BluetoothSerial.h"

#define MPU6050\_ADDR 0x68

#define ACCEL\_XOUT\_H 0x3B

#define GYRO\_XOUT\_H 0x43

#define PWR\_MGMT\_1 0x6B

#define ACCEL\_SCALE 16384.0 // ±2g

#define GYRO\_SCALE 131.0 // ±250 deg/s

BluetoothSerial SerialBT;

const int LED\_BUILTIN = 2;

void setup () {

  Serial.begin(115200);

  Wire.begin();

  SerialBT.begin("Helmet-Alert");

  pinMode(LED\_BUILTIN, OUTPUT);

  digitalWrite(LED\_BUILTIN, HIGH); // Turn on LED to indicate system is active

  // Wake up MPU6050

  Wire.beginTransmission(MPU6050\_ADDR);

  Wire.write(PWR\_MGMT\_1);

  Wire.write(0);

  Wire.endTransmission();

  Serial.println("System ON. Monitoring started...");

}

void loop () {

  float ax = readWord(ACCEL\_XOUT\_H) / ACCEL\_SCALE;

  float ay = readWord(ACCEL\_XOUT\_H + 2) / ACCEL\_SCALE;

  float az = readWord(ACCEL\_XOUT\_H + 4) / ACCEL\_SCALE;

  float gx = readWord(GYRO\_XOUT\_H) / GYRO\_SCALE;

  float gy = readWord(GYRO\_XOUT\_H + 2) / GYRO\_SCALE;

  float gz = readWord(GYRO\_XOUT\_H + 4) / GYRO\_SCALE;

  float accMagSq = ax \* ax + ay \* ay + az \* az;

  Serial.printf("AX=%.2f AY=%.2f AZ=%.2f | GZ=%.2f | Acc²=%.2f\n", ax, ay, az, gz, accMagSq);

if (accMagSq > 7.0) {

    Serial.println("ACCIDENT DETECTED!");

    SerialBT.println("ACCIDENT DETECTED!");

    delay (5000); // Prevent spamming

  }

  delay(200);

}

int16\_t readWord(int reg) {

  Wire.beginTransmission(MPU6050\_ADDR);

  Wire.write(reg);

  Wire.endTransmission(false);

  Wire.requestFrom(MPU6050\_ADDR, 2, true);

  return (Wire.read() << 8) | Wire.read();

}

**Note:**  
The condition if (accMagSq > 7.0) checks if the squared magnitude of acceleration exceeds a threshold that corresponds to an impact of approximately **5g**. This threshold value (7.0) can be adjusted to increase or decrease the sensitivity of accident detection depending on the specific use case, rider preference, or testing results.

* **Smart Helmet Android App:**
* Establishes a Bluetooth connection with the ESP32 (device name: "Helmet-Alert").
* Listens for incoming messages from the ESP32 in real-time.
* When it receives "ACCIDENT DETECTED", it:
  + - Retrieves the rider’s GPS location using the phone’s location services.
    - Sends an SMS containing the accident alert and Google Maps location link to the stored guardian number.
* Provides a Settings UI to store or update the guardian’s phone number.
* Automatically starts monitoring once connected to ESP32 (no manual start needed).
* **Guardian Android App:**
* Listens for incoming SMS messages using a ‘BroadcastReceiver’.
* On detecting a message containing "ACCIDENT DETECTED":
* Automatically plays a siren sound and triggers phone vibration.
* Requests and handles Do Not Disturb (DND) permission to override silent or vibrate mode if necessary.

**Working Explanation:**

1. **System Power-On:**

* The Smart Helmet system (ESP32 + MPU6050) is powered by a battery. The ESP32 boots up and begins monitoring data from the MPU6050 sensor.

1. **Accident Detection:**

* The MPU6050 continuously measures acceleration and detects if the acceleration magnitude crosses a set threshold

1. **Bluetooth Communication:**

* On detecting an accident, the ESP32 sends the alert message (“ACCIDENT DETECTED”) over Bluetooth to the rider’s smartphone running the SmartHelmet Android app.

1. **Smart Helmet App Response:**

* The app receives the Bluetooth message, fetches the current GPS location of the phone, and sends an SMS containing the alert and Google Maps location link to the predefined guardian’s phone number.

1. **Guardian App Action:**

* The guardian’s phone receives the SMS. The Guardian app detects this using a Broadcast Receiver.

1. **Guardian Alert:**

Upon receiving the SMS:

* A siren sound plays automatically.
* The phone vibrates.
* A Toast message or popup indicates that an accident alert has been received.