#include <DHTesp.h>

#include <WiFi.h>

#include <ThingSpeak.h>

const int DHT\_PIN = 14;       // DHT22 on pin 14

const int VOLT\_PIN = 32;      // Potentiometer simulating battery voltage

const int VIB\_PIN = 33;       // Simulated analog vibration input

const int TYRE\_PIN = 34;      // Simulated analog tyre wear input

const int BUTTON\_PIN = 12;    // Fault simulation button

const int LED\_PIN = 2;        // Alert LED

DHTesp dht;

WiFiClient client;

// WiFi and ThingSpeak

const char\* ssid = "Wokwi-GUEST";

const char\* pass = "";

const unsigned long channelNumber = 2924892;

const char\* writeAPIKey = "ZL7CKDYRN4KW3U6A";

// Runtime simulation

unsigned long startTime;

void connectWiFi() {

  if (WiFi.status() == WL\_CONNECTED) return;

**Serial**.print("Connecting to WiFi");

  WiFi.begin(ssid, pass);

  unsigned long t = millis();

  while (WiFi.status() != WL\_CONNECTED && millis() - t < 10000) {

**Serial**.print(".");

    delay(500);

  }

  if (WiFi.status() == WL\_CONNECTED) {

**Serial**.println("\nWiFi connected!");

  } else {

**Serial**.println("\nFailed to connect to WiFi.");

  }

}

void setup() {

**Serial**.begin(115200);

  dht.setup(DHT\_PIN, DHTesp::DHT22);

  pinMode(BUTTON\_PIN, INPUT\_PULLUP);

  pinMode(LED\_PIN, OUTPUT);

  connectWiFi();

  ThingSpeak.begin(client);

  startTime = millis(); // Start mileage counter

}

void loop() {

  connectWiFi();

  TempAndHumidity tempData = dht.getTempAndHumidity();

  float voltage = analogRead(VOLT\_PIN) \* (3.3 / 4095.0);      // Simulated battery efficiency

  int vibration = analogRead(VIB\_PIN);                        // Simulated engine vibration

  int tyreWear = analogRead(TYRE\_PIN);                        // Simulated tyre condition

  bool fault = !digitalRead(BUTTON\_PIN);                      // Button press = fault

  // Simulated mileage

  unsigned long mileage = (millis() - startTime) / 1000;      // In seconds

  // Debug output

**Serial**.printf("Temp: %.2f degC, Volt: %.2f V, Vib: %d, Tyre: %d, Mileage: %lu s, Fault: %d\n",

              tempData.temperature, voltage, vibration, tyreWear, mileage, fault);

  // Upload to ThingSpeak

  ThingSpeak.setField(1, tempData.temperature);

  ThingSpeak.setField(2, voltage);

  ThingSpeak.setField(3, vibration);

  ThingSpeak.setField(4, tyreWear);

  ThingSpeak.setField(5,(long)mileage);

  ThingSpeak.setField(6, fault);

  // LED Alert

  if (tempData.temperature > 50 || voltage < 2.5 || vibration > 2500 || tyreWear < 500 || fault) {

    digitalWrite(LED\_PIN, HIGH);

  } else {

    digitalWrite(LED\_PIN, LOW);

  }

  int status = ThingSpeak.writeFields(channelNumber, writeAPIKey);

  if (status == 200) {

**Serial**.println("Data sent to ThingSpeak.");

  } else {

**Serial**.println("ThingSpeak update failed.");

  }

  delay(15000);  // ThingSpeak requires 15 sec delay

}