F1 Lights Out Reaction Tester: A Driver's Reflex Challenge

TEAM MEMBERS:
N.CHANDANA [23BEC1219], HARINI KAMATCHI VELRAJAN [23BEC1136]

PROBLEM STATEMENT:

Let's see if you can be an F1 driver!
As soon as the buzzer sounds, you must press the touch sensor. Your reaction time will be calculated. (Note: The average reaction time of an F1 driver is between 0.2 and 0.3 seconds.) The red light will turn off as soon as the sensor detects your touch.

BOARD: Aries Development Board v3

PIN CONNECTIONS :

Component	VEGA Aries Pin
Touch Sensor	GPIO3
Buzzer (+)	GPIO0
Buzzer (-)	GND
Red LED (+)	PWM0 (GPIO0)
Green LED (+)	PWM2 (GPIO2)
LEDs (-)	GND

CODE:

```
const int TOUCH_PIN = 3;     // Touch sensor on GPIO3
const int BUZZER_PIN = 0;     // Buzzer on GPIO0
const int RED_LED = 0;       // Red LED on PWM0 (GPIO0)
const int GREEN_LED = 2;     // Green LED on PWM2 (GPIO2)

// Game variables (all time-related as unsigned long)
unsigned long startTime = 0;
unsigned long reactionTime = 0;
#include <climits>
unsigned long bestTime = ULONG_MAX; // Initialize with maximum value
unsigned long totalTime = 0;
int testCount = 0;
bool waitingForTouch = false;

void setup() {
  pinMode(TOUCH_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);

  Serial.begin(115200);
  while (!Serial); // Wait for serial connection

  Serial.println("\n\n=== VEGA Aries Reaction Tester ===");
  Serial.println("Instructions:");
  Serial.println("1. Wait for RED light + beep");
  Serial.println("2. Touch sensor immediately");
  Serial.println("3. See your reaction time");
  Serial.println("- - - - - - - - - - - - - - - - -");

  digitalWrite(GREEN_LED, HIGH); // Ready state
  digitalWrite(RED_LED, LOW);

  randomSeed(analogRead(A0)); // Better randomness
}

void loop() {
  // Random test start (1-4 second delay)
  if (!waitingForTouch && random(1000) == 0) {
```

```
    startTest();
  }

  // Check for touch response
  if (waitingForTouch && digitalRead(TOUCH_PIN)) {
    endTest();
  }

  delay(10); // Small delay to prevent debounce
}

void startTest() {
  waitingForTouch = true;
  startTime = millis();

  // Visual and audio cue
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(RED_LED, HIGH);
  tone(BUZZER_PIN, 1500, 100);

  Serial.println("\n[GO!] Touch NOW!");
}

void endTest() {
  reactionTime = millis() - startTime;
  waitingForTouch = false;
  testCount++;
  totalTime += reactionTime;

  // Update best time
  if (reactionTime < bestTime) {
    bestTime = reactionTime;
    tone(BUZZER_PIN, 3000, 300); // High-pitch victory sound
  } else {
    tone(BUZZER_PIN, 800, 300);   // Normal sound
  }

  // Reset lights
  digitalWrite(RED_LED, LOW);
  digitalWrite(GREEN_LED, HIGH);
```

```cpp
  // Display results
  Serial.println("- - - - - - - - - - - - - - - -");
  Serial.print("Your time: ");
  Serial.print(reactionTime);
  Serial.println(" milliseconds");

  Serial.print("Average: ");
  Serial.print(totalTime/testCount);
  Serial.print("ms | Best: ");
  Serial.print(bestTime);
  Serial.println("ms");

  Serial.print("Tests completed: ");
  Serial.println(testCount);

  // Prevent division by zero
  if (testCount >= 5) {
  Serial.print("Last 5 avg: ");
    Serial.print(calculateRecentAverage());
    Serial.println("ms");
  }

  Serial.println("- - - - - - - - - - - - - - - -");
  Serial.println("Waiting for next test...\n");
}

// Helper function for recent average
unsigned long calculateRecentAverage() {
  // For more advanced statistics tracking
  return totalTime/testCount; // Simple implementation
  // Add circular buffer for last 5 tests if needed
}
```

OUTPUT: