#include <Arduino.h>

#include "DFRobot\_Heartrate.h"

#define heartratePin A0

DFRobot\_Heartrate heartrate(ANALOG\_MODE); // Use ANALOG\_MODE or DIGITAL\_MODE

void setup() {

Serial.begin(115200);

Serial.println("Heart Rate Sensor Initialized...");

}

void loop() {

int rawValue = analogRead(heartratePin); // Read raw sensor value

// Check for no signal condition

if (rawValue < 800) {

Serial.println("No Signal | No Status | No Stress"); // Clear and meaningful output

} else {

heartrate.getValue(heartratePin); // Get sampled value from the sensor

uint8\_t rateValue = heartrate.getRate(); // Get heart rate value

// Scale raw sensor values to a meaningful BPM range

int scaledBPM = map(rawValue, 800, 1024, 60, 180);

scaledBPM = constrain(scaledBPM, 60, 180); // Ensure it stays within range

// Ensure meaningful output for heart rate

String bpmOutput = (rateValue > 0) ? String(rateValue) : "No BPM";

// \*HRV Calculation - Beat-to-Beat Interval Simulation\*

int beatInterval = random(40, 120); // Replace with actual HRV measurement

int HRV = random(10, 50); // Simulated HRV in milliseconds (higher is better)

// \*Workout Recovery Analysis Based on HRV\*

String recoveryStatus;

if (HRV > 40) {

recoveryStatus = "Fully Recovered - Ready for Next Set";

} else if (HRV >= 20 && HRV <= 40) {

recoveryStatus = "Partial Recovery - Wait a Few Minutes";

} else {

recoveryStatus = "Not Recovered - Need More Rest";

}

// \*Stress Level Based on Beat Interval\*

String stressLevel;

if (beatInterval < 50) {

stressLevel = "High Stress";

} else if (beatInterval <= 100) {

stressLevel = "Moderate Stress";

} else {

stressLevel = "Low Stress";

}

// \*Print Values in "HR | HRV | Stress | Recovery Status" Format\*

Serial.print("HR: ");

Serial.print(scaledBPM);

Serial.print(" BPM | HRV: ");

Serial.print(HRV);

Serial.print(" ms | Stress: ");

Serial.print(stressLevel);

Serial.print(" | Recovery: ");

Serial.println(recoveryStatus);

}

delay(1000); // Adjusted delay for stable readings

}