Automated Street Light System - C++ Code

```cpp
#include <iostream>
#include <ctime>
#include <thread>
#include <chrono>
#include <cstdlib>
using namespace std;

const int LIGHT_THRESHOLD = 40;
const int NUM_STREET_LIGHTS = 5;

struct StreetLight {
    int id;
    bool isOn;
};

int readLightIntensity() {
    return rand() % 101;
}

bool detectMotion() {
    return rand() % 2;
}

void controlStreetLight(StreetLight &light, int lightLevel, bool motionDetected) {
    if (lightLevel < LIGHT_THRESHOLD || motionDetected) {
        light.isOn = true;
    } else {
        light.isOn = false;
    }
}

void displayStatus(StreetLight lights[], int lightLevel, bool motionDetected) {
    cout << "\nTime: " << time(0) << endl;
    cout << "Light Intensity: " << lightLevel << " | Motion Detected: "
         << (motionDetected ? "Yes" : "No") << endl;

    for (int i = 0; i < NUM_STREET_LIGHTS; ++i) {
        cout << "Street Light " << lights[i].id << " is "
             << (lights[i].isOn ? "ON" : "OFF") << endl;
    }
    cout << "----------------------------------------\n";
}

void delay(int milliseconds) {
    this_thread::sleep_for(chrono::milliseconds(milliseconds));
}

int main() {
    srand(time(0));
    StreetLight lights[NUM_STREET_LIGHTS];
    for (int i = 0; i < NUM_STREET_LIGHTS; ++i) {
        lights[i].id = i + 1;
        lights[i].isOn = false;
    }

    char continueSimulation = 'y';

    while (continueSimulation == 'y' || continueSimulation == 'Y') {
        int lightLevel = readLightIntensity();
        bool motionDetected = detectMotion();

        for (int i = 0; i < NUM_STREET_LIGHTS; ++i) {
            controlStreetLight(lights[i], lightLevel, motionDetected);
        }

        displayStatus(lights, lightLevel, motionDetected);
        delay(3000);

        cout << "\nDo you want to simulate again? (y/n): ";
        cin >> continueSimulation;
    }

    cout << "\nSimulation ended. All street lights OFF for now.\n";
    return 0;
}
```