LabActivity3 - Report

Problem Analysis

The exercise requires implementing a robot capable of detecting a light source, reaching it as quickly as possible, and avoiding obstacles along the way. Once the robot reaches the area near the light, it must stop.

The arena consists of the usual perimeter walls, a light source, and a corresponding **black spot** that identifies the stopping area. A variable number of obstacles and robots are generated randomly.

The problem must be solved by implementing a **subsumption architecture**.

My Solution

I divided the robot's behavior into 4 hierarchical levels, each with increasing priority:

- **Level 0 →** random walk. Low priority. The robot moves randomly, updating its velocity every *MOVE_STEPS*. This behavior is executed only if no higher-priority behavior is active.
- **Level 1 →** phototaxis. Medium priority. The robot calculates the angular weighted average of light intensity using its circular light sensors, to determine the direction of the light source. The wheels are adjusted accordingly to move toward that direction. When active, the LEDs turn yellow.
- **Level 2 →** obstacle avoidance. High priority. The robot checks its front and lateral proximity sensors (1, 2, 23, 24) to avoid obstacles. If a reading exceeds the *OBSTACLE_THRESHOLD*, the robot rotates in place in the opposite direction of the obstacle. The LEDs turn blue.
- **Level 3 →** stop. Highest priority. Using the *motor_ground* sensor, the robot detects black surfaces. If such a surface is detected, the robot stops completely and sets the LEDs to red. This is the only condition that modifies the global *finish* state, stopping further execution of the *step()* function.

Each level checks whether it should take control of the robot based on predefined thresholds, setting wheel velocities and LED color accordingly. A handler variable is used to avoid conflicts between lower and higher levels.