Harvester Robot

Team 1
Lab 6

Daniel Bigelow, Ben Brown, and Brian Vees

Harvester Robot

Problem Description:
In this lab, we had to write a program that allowed our robot to explore the arena for 60 seconds, finding as much “food” (represented by black dots) as possible, then go “home” (designated by a bright light) once the 60 seconds had elapsed. In this lab, we were able to use four light sensors to find food and to find the light that designated home, and a single bump sensor for obstacle avoidance. We used both the Handyboard and the Lego RCX in our solution to this lab.

Solutions:

Hardware:
We built a base around the Handyboard and the RCX that could support both of the boards, and still had room for four light sensors. We positioned our sensors so that the Lego light sensors were pointing straight at the ground, and the two reflective optosensors were at about a forty-five degree angle to the ground. This allowed us to use all four sensors to find food, as well as still being able to navigate home at the end. We also had one bump sensor at the front of our robot. This kept the robot from getting stuck on walls. All in all, our hardware design was extremely robust, and worked quite well. There were only a few problems in the end, which will be discussed later in this report.

RCX:
We used the Lego RCX board in order to get a better range of values from our Lego light sensors, allowing us to use them to find “food.” We programmed this board to sample the light sensor values, and whenever they reached a certain threshold value, output on one of the motor ports. We connected the two motor ports from the RCX to two of the sensor inputs on the Handyboard in order to facilitate communication between the two boards.

Handyboard:
We designed a program that did the following:

- Once the start button was pressed, the system clock was initialized, and the robot would begin to wander randomly.
    - To wander, it would go forwards for a random amount of time, then turn for a random amount of time, then repeat.
- Whenever one of the light sensors registered black, the Handyboard would beep. Different tones were used, depending on which sensor had seen the food.
- Whenever the touch sensor was triggered, the main wander program would be interrupted, and the robot would back up for ½ of a second, then turn for .25 seconds. The direction of the turn was determined by a counter, so that it didn't always turn the same direction.
- Once the system clock reached 60 seconds, the robot stopped, emitted 3 high-pitched beeps, and began to search for home.
    - To search for home, the robot simply saw which of the reflective optosensors registered a higher light value, then sped up the wheel on the opposite side.
    - When the sensors registered similar values, the robot would normalize so that it could go forwards.

Problems and Solutions:

One of the biggest hardware challenges ended up being sensor placement. Originally, our reflective optosensors were too close together. This meant that when the robot was seeking home, the light values seen by the sensors were not different enough to be able to steer towards the light accurately. We were able to fix this problem by moving the light sensors further apart.

Another problem that we had was that our bump sensor did not originally extend far enough to trigger if the robot hit a corner just right. This problem was not evident at first, because it only really mattered in the home-seeking part of the program. If it got stuck in a corner during the random wander part, the randomness of its movements would eventually get it unstuck again. Once we finally discovered this flaw, however, we were able to fix it by extending the bump sensor forwards and to the sides.

The only other problem we had was that since the reflective optosensors were at an angle to the ground, they were activated whenever the robot got too close to the wall, as well as when they actually saw food. We ended up being unable to fix this problem. We could have made it so that the robot would not look for food when it was in the obstacle avoidance function, but that would have reduced the chances of finding food near the walls.

Conclusion:

While working on this project, we were able to come up with a method for making the RCX communicate with the Handyboard, so that we could use the Lego light to their full potential. We also learned the importance of being able to use a sensor for multiple purposes, allowing us to search a wider stretch of the arena for dots, and still be able to find home at the end. Both of these concepts may be quite useful in later projects.