Activity 3: Robot Telephone

1 Overview

The goal of this activity is to build functions which will display the number of hops from your robot to the guide robot by turning on the appropriate number of LEDs, as shown below:

We’re introducing a bunch of new functions. Read the description for each carefully.

1. behControl.nbrGetNbrList(): Returns the list of neighboring robots. Input: None, Output: the list of all robots neighboring this robot.
2. behControl.guideMsgGetHopsFromNbr(nbr): Gets the number of hops of the input neighboring robot. Input: nbr - the tupel for the neighbor taken from the neighbor list (eg. nbrList[3]), Output: hops - the number of hops the neighboring robot has.
3. behControl.ledsSet(color, index): Turns a specific led of a given index and color on. Input: color = ’r’,’g’,’b’, which color to turn on, index = 0-4, in counterclockwise wise direction, Output: None.
4. behControl.nbrGetNbrBearing(nbr): Returns the bearing of the neighbor. Input: nbr - the tupel for the neighbor taken from the neighbor list (eg. nbrList[3]), Output: The bearing in radians.
5. behControl.guideMsgGetMaxHops(): Gets the maximum number of hops allowed in the network. Input: None, Output: HOPS_MAX - the max number of hops in the network

2 Tasks

2.1 Pseudocode

You know the drill. Sketch out your game plan. Proof of pseudo-code before you start programming means +5 point for the design challenge today. It is OK to write pseudo-code for the first function, code and test it, and then write pseudo-code for the second function, and so forth.

2.2 Testing

The behavior hops() is given to you for testing the first two parts of the activity. Just uncomment the parts you want to test.

2.3 Calculate your Hops

You will write the function computeMyHops(). It will take no inputs and returns the number of hops. Use behControl.guideMsgGetMaxHops() to get a starting point for your hops and nbrList = behControl.nbrGetNbrList() to generate you neighbor list.

You are HIGHLY ENCOURAGED to print any variable to gain a better understanding of how to use the data to accomplish tasks. You will use a for loop to iterate through the neighbor list and a nested if statement to update your hops to a lower number if necessary. When your loop finishes, your number of hops should be equal to your lowest neighbor’s hops plus one.

2.4 Turn on LEDs

You will be writing the function ledControl(color, number) which takes two arguments and returns none. You will need to employ a while loop and the function ledsSet(color, index). The LEDs are indexed from 0-4 and go counterclockwise. The parameter color should be ‘r’, ‘g’, or ‘b’. The input for number should be 1-5. Use the while loop to iterate through each LED index.

2.5 Greet Your Neighbor

You need to write two functions that work together to direct your robot to turn until it is facing its lowest hops neighbor.

The first is faceNbr(nbr) which takes a nbr tuple as an input and outputs the behavior tuple (active, tv, rv). You will use the function behControl.nbrGetNbrBearing(nbr) and then employ a series of if/elif statements to make decisions on which direction to turn based on your robot’s bearing compared to its neighbor.

The second function is faceFirstNbr(). It will take no arguments and return a tuple. You will use behControl.nbrGetNbrList() and then a series of if/elif statements to choose between calling moveStop() (given to you) and faceNbr(nbr), based on the number of neighbors.

2.6 Implement ledControl(color, number)

Invoke ledControl(color, number) in your functions such that the LEDs are set when you calculate your number of hops.