

# Robot Library Reference

## behControl Library:

`behControl.init(funcList)`: Takes a list of functions as input and executes functions using a heirarchy. If there is no active behavior, red LEDs slowly blink. The first function in the list has the highest priority, while the last function has the lowest priority.

`behControl.robotStart()`: Red LEDs circle until the red button is pushed, then program execution can begin. Velocity, leds, and neighbors are all initialized. Input: None, Output: None.

`behControl.robotEnd()`: All LEDs are turned off and the motors are stopped. Input: None, Output: None.

`behControl.runBehFor(time)`: Updates system for the input amount of time and delays changing it. Input: time - amount of time in microseconds to runBehFor the behavior controller, Output: None.

`behControl.setTv(tv)`: Function used to set the translational velocity. Input: tv - translational velocity, Output: None.

`behControl.setRv(rv)`: Function used to set the rotational velocity. Input: rv - rotational velocity, Output: None.

`behControl.setTvRv(tv, rv)`: Function used to set both the tv and rv part of velocity. Input: tv - rotational velocity, rv - translational velocity, Output: None.

`behControl.setTvRv(tv, rv)`: Function used to set both the tv and rv part of velocity. Input: tv - rotational velocity, rv - translational velocity, Output: None.

`behControl.obsFront()`: Determines if there is an obstacle in front of the robot. Input: None Output: True if an obstacle is in front of the robot, False if otherwise.

`behControl.obsLeft()`: Determines if there is an obstacle to the left of the robot. Input: None Output: True if an obstacle is to the right of the robot, False if otherwise.

`behControl.obsRight()`: Determines if there is an obstacle to the right of the robot. Input: None Output: True if an obstacle is to the right of of the robot, False if otherwise.

`behControl.lightSense(sensor)`: Determines the amount of light at the input light sensor Input: sensor - which light sensor to read I.E. 'fl' = front left sensor, 'fr' = front right sensor, or 'r' = rear/back sensor Output: value between 0 and 1023 indicating the amount of light present at the input sensor.

`behControl.nbrGoal()`: If a goal robot is a neighbor of the robot, return True, else return False Input: None, Output: True if a goal robot is detected, False if otherwise.

`behControl.nbrGetNbrList()`: `behControl.nbrGetNbrList()`: Returns the list of neighboring robots. Input: None, Output -the list of all robots neighboring this robot.

`behControl.guideMsgGetHopsFromNbr(nbr)`: Gets the number of hops of the input neighboring robot. Input: `nbr` - the tuple for the neighbor taken from the neighbor list (eg. `nbrList[3]`), Output: hops - the number of hops the neighboring robot has.

`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..

`behControl.nbrGetNbrBearing(nbr)`: Returns the bearing of the neighbor. Input: `nbr` - the tuple for the neighbor taken from the neighbor list (eg. `nbrList[3]`), Output: The bearing in radians.

`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.