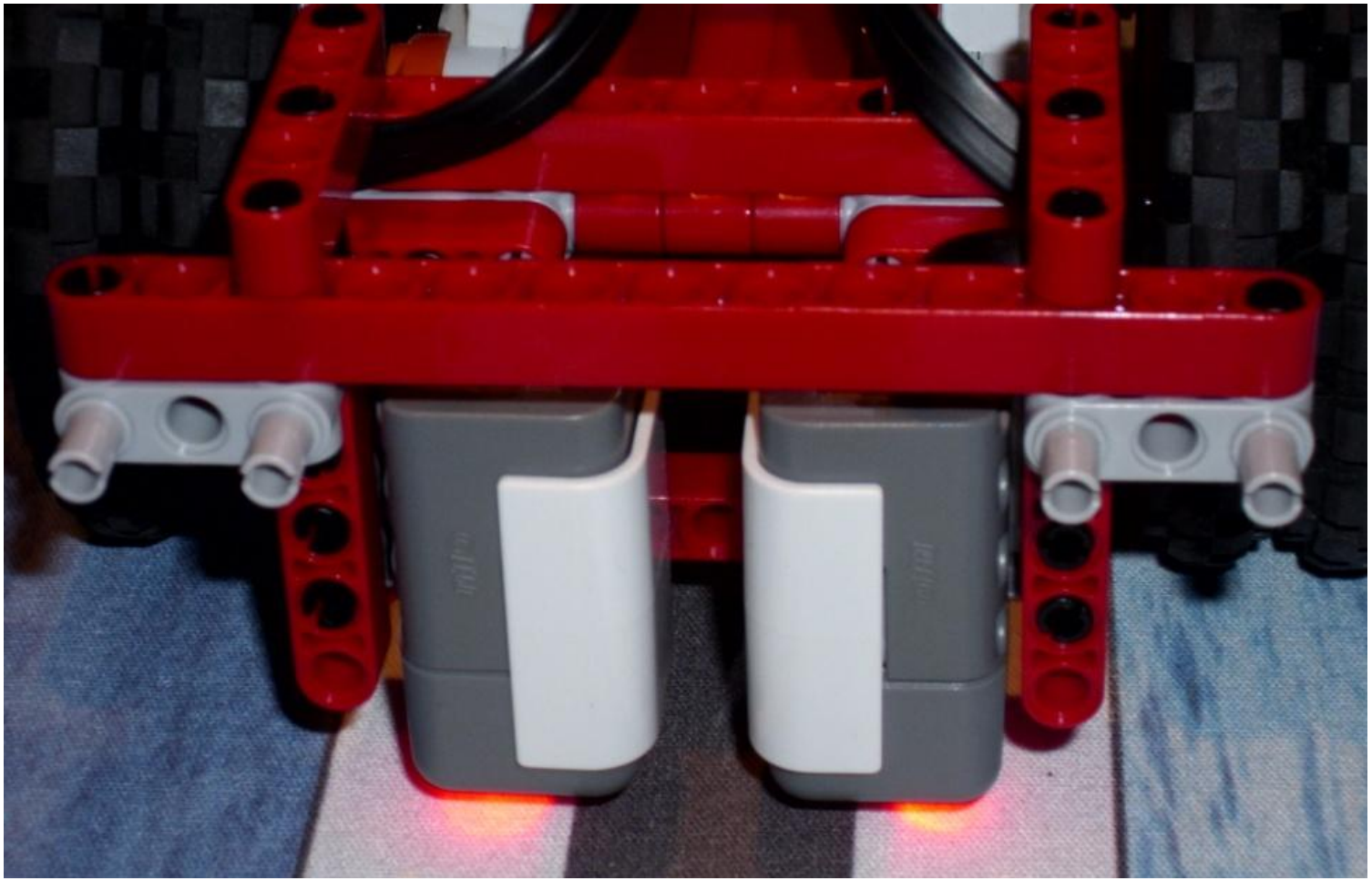
USING LIGHT SENSORS



LOCATING ROBOT

- Basic robot navigation uses “dead reckoning”, which is just a series of movements without precise location knowledge. Positional errors accumulate with each movement and turn. In other words, the farther the robot goes, the greater the likely error from the desired location and orientation.
- Navigating by light sensors allows the robot to redetermine its location and orientation, and remove most of the accumulated errors.

FUNDAMENTALS OF LIGHT SENSORS

- The Lego light sensor measures the light intensity within its field of view. The field of view is cone-shaped, meaning that the farther the light sensor is from the mat, the larger the elliptical portion of the mat that is within its field of view.
- For line following, the light intensity that is measured is what is reflected off the challenge mat.
- The light sensor can be configured to shine a built-in red LED into its field of vision. If left off, the light sensor will measure the ambient light level.
- KEY POINT: Best performance is when the light source is on and the light sensor is from 5 to 8 mm above mat.
- KEY POINT: For most consistent performance, shading the light sensors will minimize the effects of ambient light on the light sensor readings. However, judicious picking of sensor thresholds and testing with a variety of light levels should minimize the need for shading.

LINE SENSOR MEASUREMENT VALUE RANGES

- The light sensor measurement has an absolute range of 0 to 100.
- However, typical measured values for a light sensor 5 mm from the mat will be between 40 to 65.
- Lego provides the means to calibrate the light sensor. Calibrating the sensor means measuring the sensor at its minimum (over black) and maximum (over white). After calibration, any measurement at or below the minimum will return 0, and any measurement at or above the maximum will return 100. Measurements in between will be normalized to between 0 and 100.
- KEY POINT: Calibrate the light sensors to minimize variability between tables.

FUNDAMENTALS OF LINE FOLLOWING

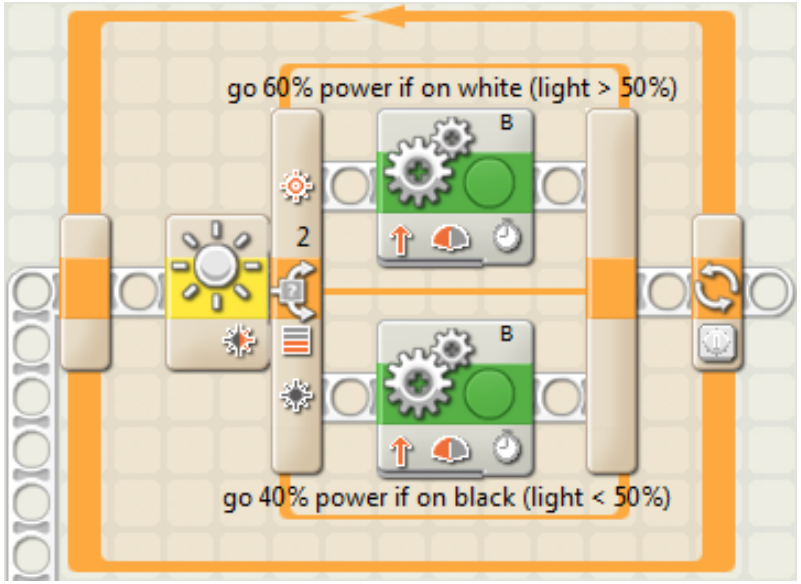
- Line following at it simplest uses a single light sensor to steer the robot along the left or right edge of a line. If the right edge, a calibrated robot will steer
  - left when the light sensor is over a white surface (≥ 50)
  - right when the light sensor is over a black surface (< 50)
- To turn left, the robot will run its right motor at a higher power level than its left motor. Likewise, to turn right, the robot will run its left motor at a higher power level than its right motor.
- Soon after it turns left, the robot will see the opposite color and turn right. The robot will then oscillate between black and white.

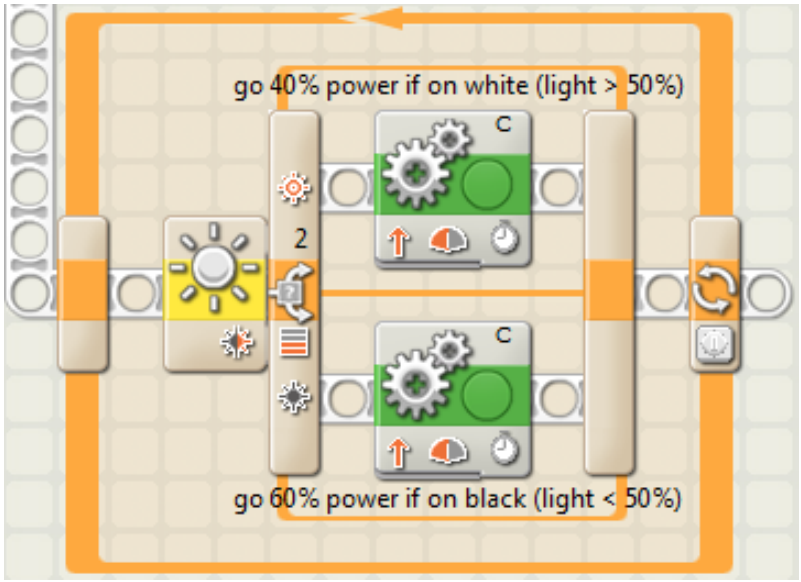
FUNDAMENTALS OF LINE FOLLOWING (continued)

- KEY POINT: The relative difference between the left and right motor power levels have significant effects on performance.
  - A relatively small difference will minimize the oscillation along the line, making it easier to point to the next mission. However, should the robot's orientation from the line get too large, the robot can cross over the line and never recover.
  - With a relatively large difference, the robot will hold onto the line. However, it will have large oscillations, such that when it stops, its orientation relative to the line can be over a wider range. The robot's orientation may be significantly off from the desired orientation to the next mission.

SINGLE SENSOR LINE FOLLOWING PROGRAM

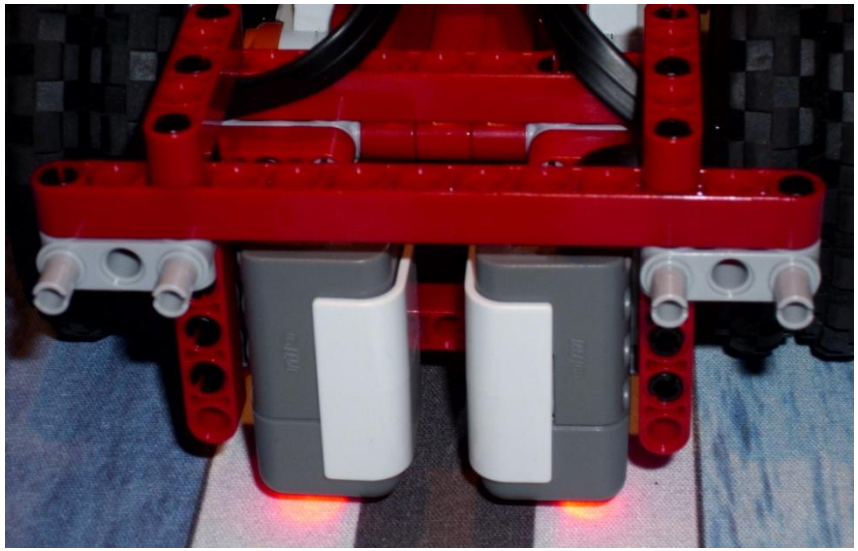
- *LINE-SINGLE.rbt* has the basic single sensor line following logic.





TWO LIGHT SENSORS

- KEY POINT: Use two light sensors to bracket the line.
- With two light sensors, the robot will straighten over the line as both light sensors will measure the white lines that border the center black line.
- The final robot orientation should be close to the line's.



MATCHING LIGHT SENSORS

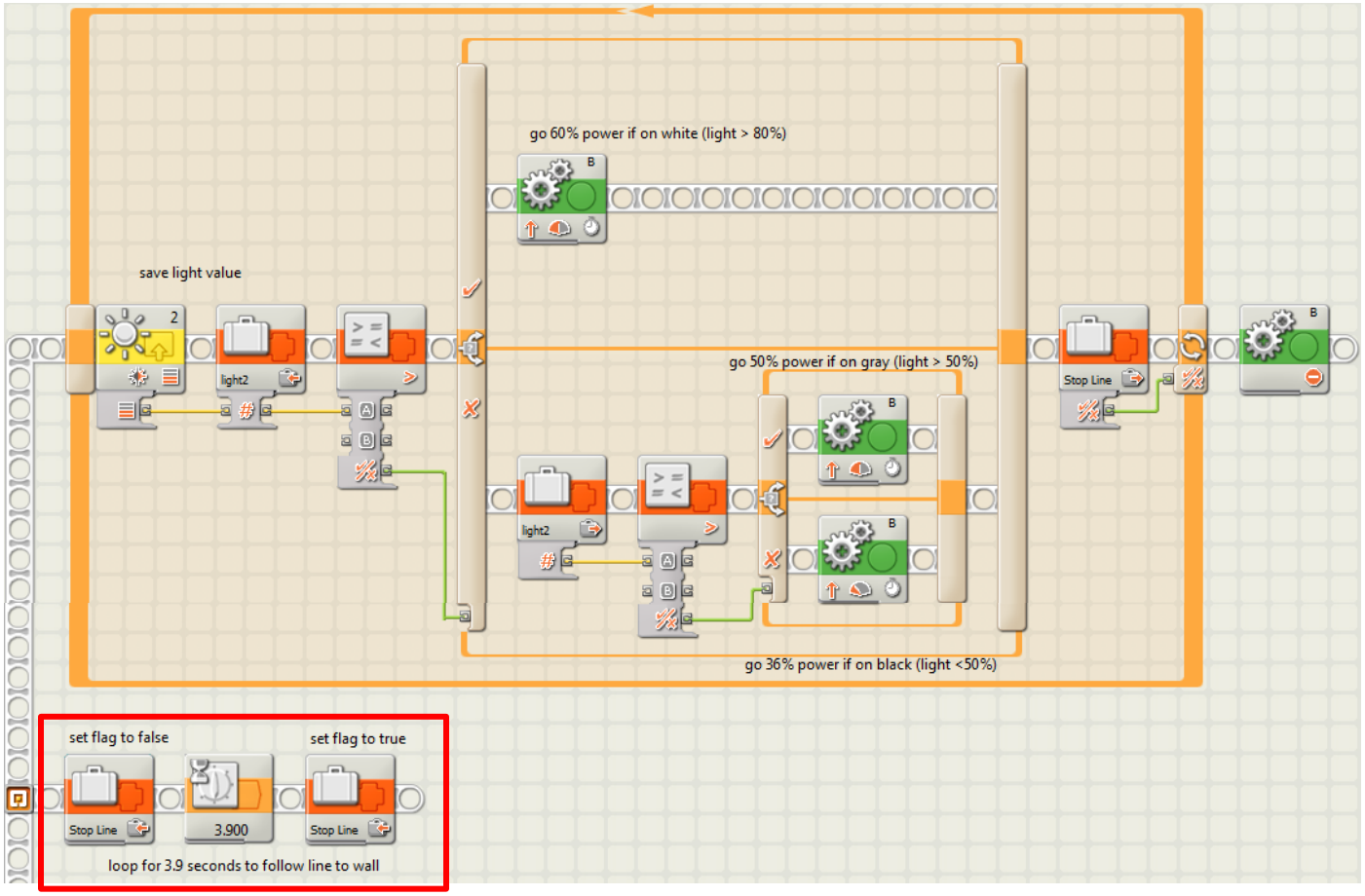
- KEY POINT: Pick two light sensors that have matching measurement values for black and white.
- Reason: The light sensor calibration program calibrates with one light sensor and applies the calibration to the other sensor.

TWO LIGHT SENSOR THRESHOLDS

- The left light sensor is tied to the left motor and the right light sensor is tied to the right motor.
  - 100 ≥ light > 80: motor at 60
  - 80 ≥ light > 50: motor at 50
  - 50 ≥ light ≥ 0: motor at 36
- The combination of these thresholds on the two sensor / motor pairs results in five separate zones of behavior.
- Adjust thresholds and relative motor power levels to optimize performance.

TWO SENSOR LINE FOLLOWING PROGRAM

- B motor control and stop trigger from *LINE-TIME.rbt*.



TWO SENSOR LINE FOLLOWING PROGRAM (continued)

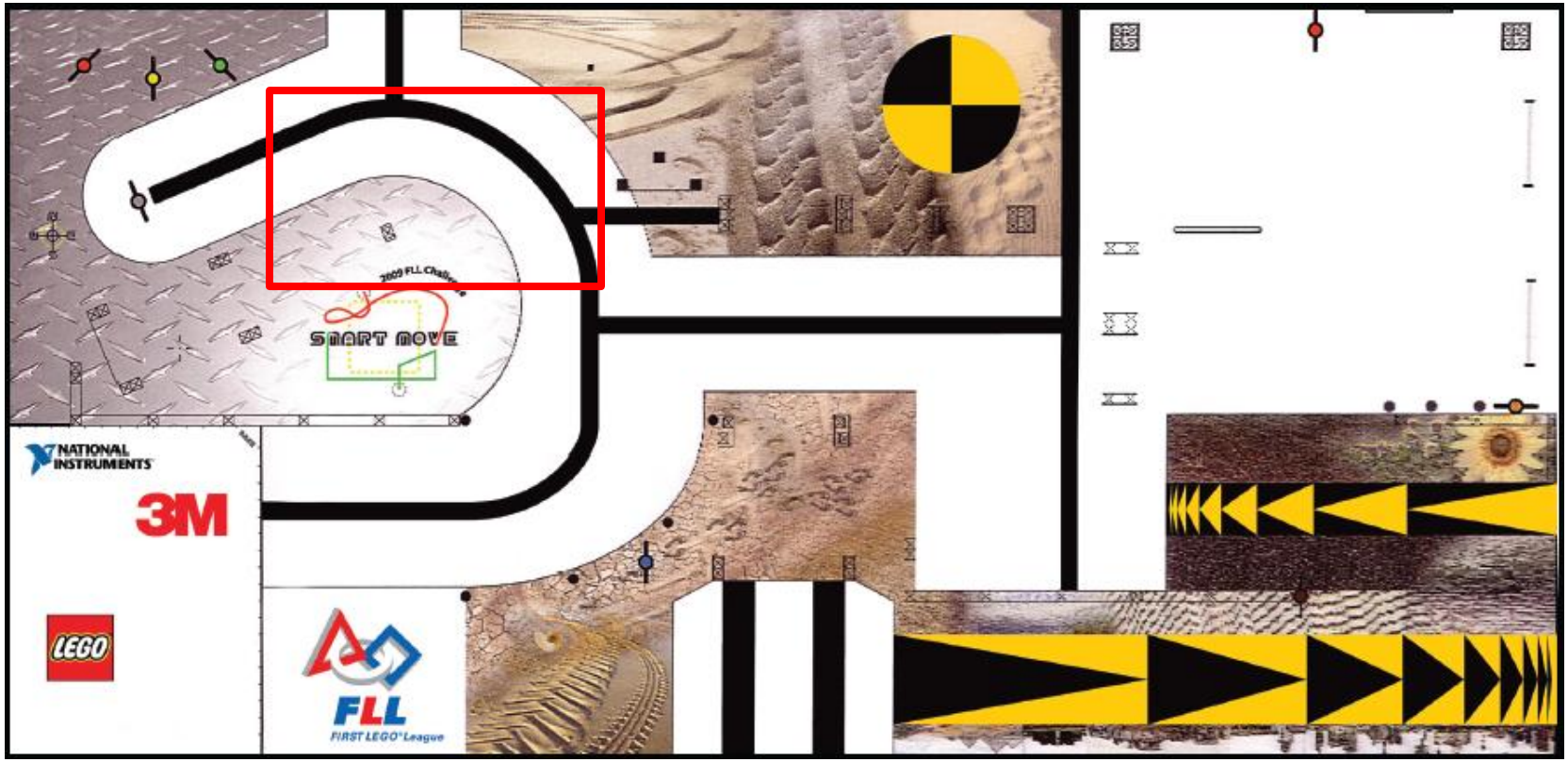
- Please note the logic in the red rectangle in the previous slide.
- By using the Stop Line variable, one part of the program can detect when 3.9 seconds has elapsed (more than enough time to reach the east wall) and signal to the line-following portions of the program to stop.
- We will modify this logic later to incorporate other line stopping triggers.

CURVE FOLLOWING

- To create a line following program that can track a curve, start by creating a program that runs the left and right motors in parallel without light sensor feedback.
- If the curve goes to the left, set the left motor's power level significantly below the right motor's. Lower the left motor's power level until the robot follows the curve. This becomes the baseline left motor power level.
- For the 2009 Food Factor challenge, there was a large left curve as seen on the next slide. Setting the right motor to 60% and the left motor to 35% matched the curve.
- With a constant white area to the left of the curve, use the left light sensor to track the left edge. Best results were found by keeping the right motor at a constant 60%, and setting the left motor to either 30% or 45% based on the left light sensor reading.

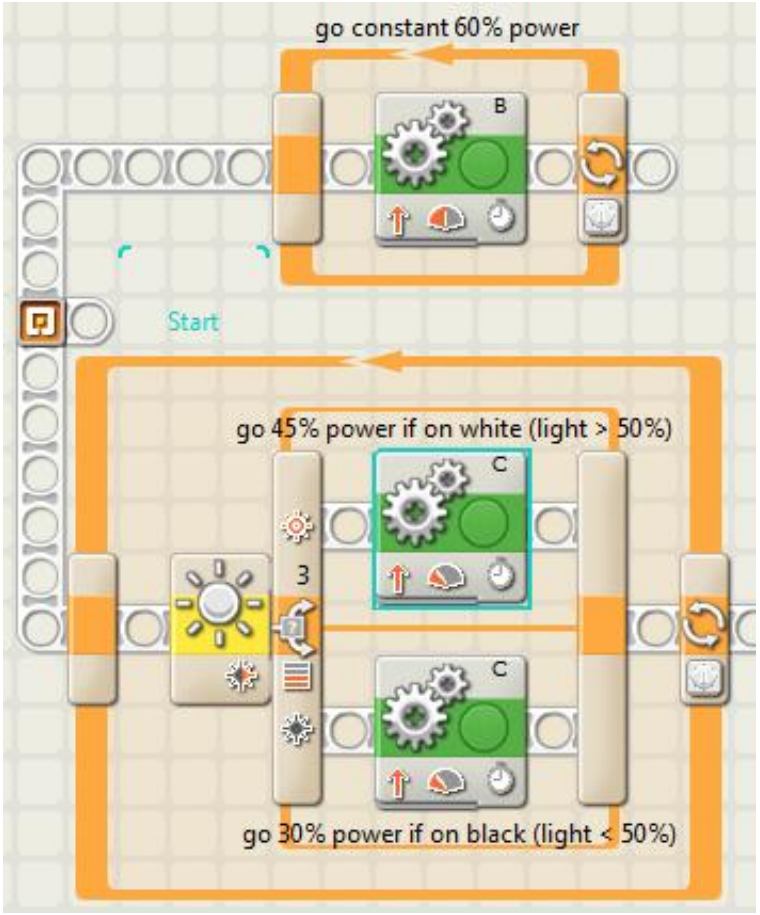
CURVE FOLLOWING (continued)

- 2009 Food Factor challenge mat.



CURVE FOLLOWING (continued)

- Program *LINE-CURVE.rbt* tracks the curve on the 2009 Food Factor mat.



POSITIONING LIGHT SENSORS

- Align middle of light sensors with middle of robot.
- Point sensors straight down, preferably 5 mm above mat.
- Configure sensors back-to-back with 1M gap, resulting in fields of view separated by 33 mm, compared to a line width of 26 mm.
- Place sensors forward of the motor axes. The farther forward the sensors are relative to the motor axes, the tighter the robot movements will be to keep sensors over the line.
- Adding a shade around the light sensors can partially shield ambient light.

CALIBRATION PROGRAMS

- A great source for calibration programs is http://www.nxtprograms.com/line_follower/steps.html.
- Download calibration programs, and load *Calibrate 3.rbt* and *View Light 3.rbt* onto the robot.
- To calibrate, run *Calibrate 3*. Follow instructions on NXT display. Run *View Light 3* to see calibrated light values at various locations on mat. Pick black and white line locations on table where light sensors will be used.
- *Calibrate 3* is quick to run. Run it before every round, if possible, when team comes to the table.
- You can delete the calibration by running *Del Calibrate.rbt*, although I don't know why you ever would.

LINE FOLLOWING START TO FINISH

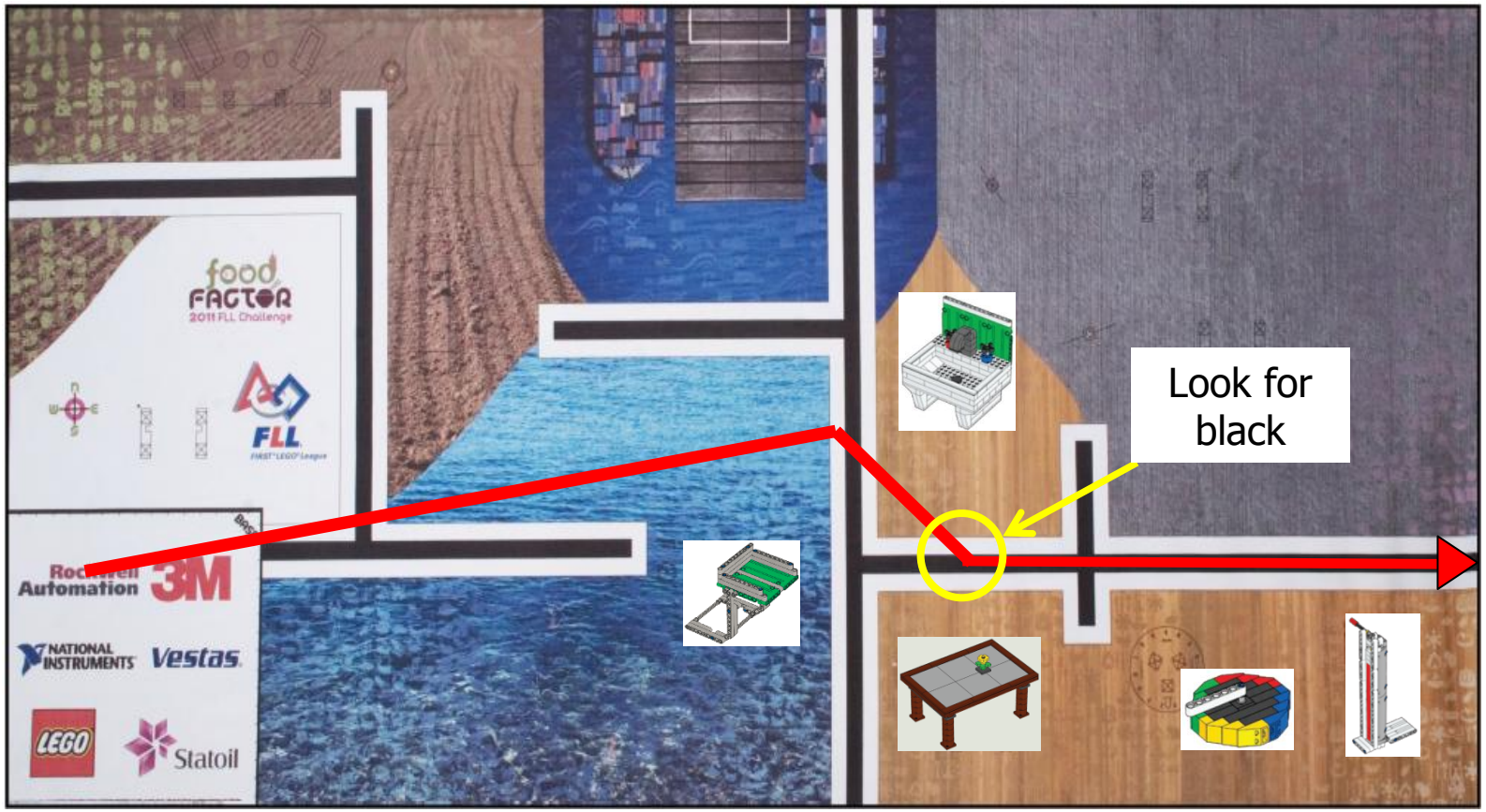
- Although we've covered the programming logic for following a line, it is only one component of a complete solution.
- There are actually five components that must be done well:
  1. Finding the line
  2. Aligning to the line
  3. Following the line
  4. Stopping along the line
  5. Orienting the robot after stopping

FINDING THE LINE

- KEY POINT: Finding the line is the hardest aspect in line following and has the greatest risk for failure.
- Unless the line goes directly out of the base, the robot must always find the line.
- Procedure:
  1. Turn the robot so that it will intercept close to the line's beginning. This will give the robot greater length to straighten as it follows the line. However, make sure that there is enough margin not to miss the line's beginning.
  2. Move the robot to stop one or two inches before the line. Going directly is faster than searching from the beginning. It also minimizes the area to search for the line, where the robot can prematurely trigger in finding the line.
  3. If the mat region is light in color near the line, have the robot search for black.
  4. If the mat region is dark in color near the line, have the robot first search for white and then search for black.

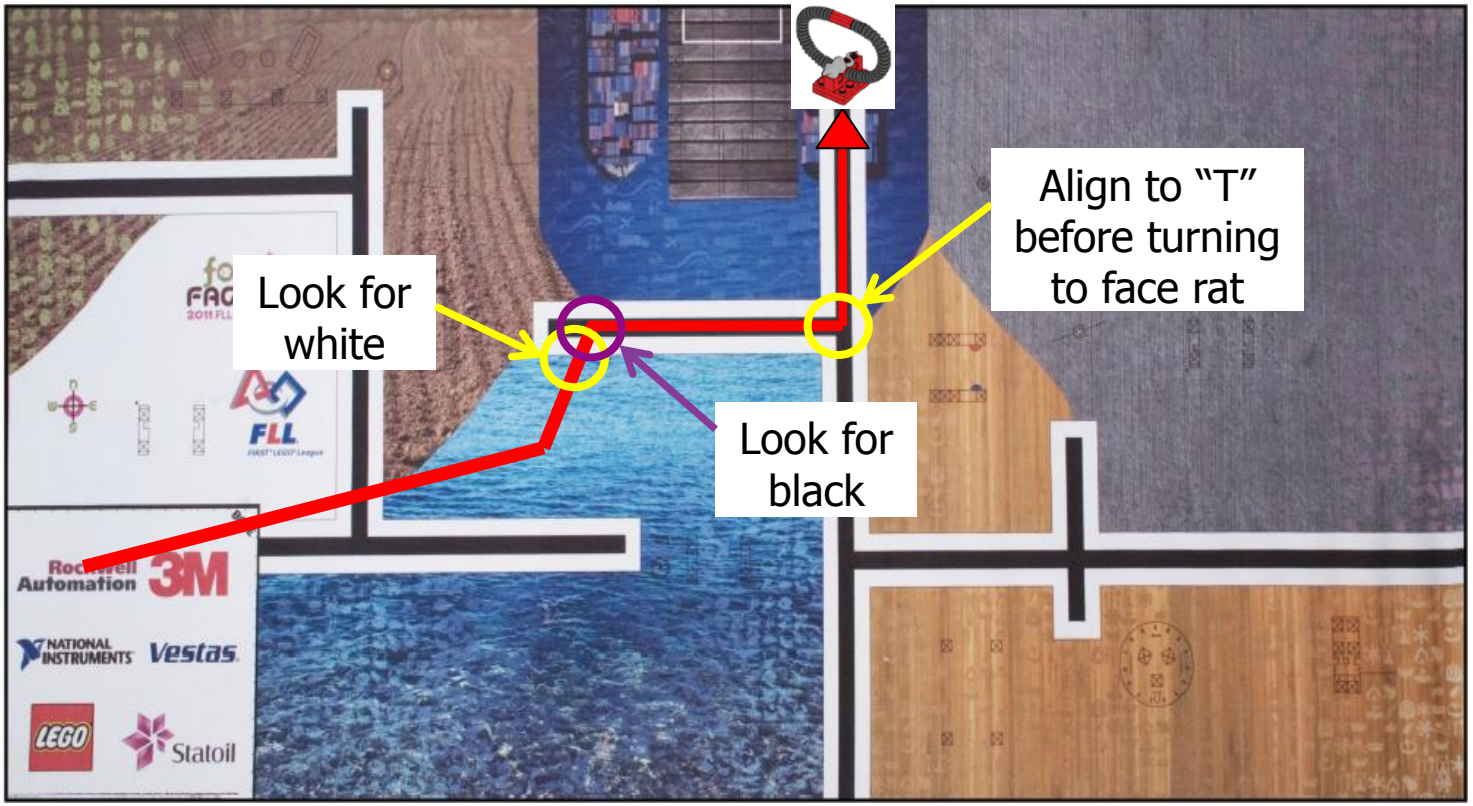
FINDING THE LINE (LIGHT REGION)

- In 2011 Food Factor, the thermometer and timer were located at the end of far line. The field near the line was light in color.



FINDING THE LINE (DARK REGION)

- In 2011 Food Factor, the second rat was located at the end of the top line. The field near the middle line was dark in color.



Note: GET TO *T.rbt* is a program that takes robot from base to T alignment.

LIGHT VS. DARK AREA

- How do you know if the area nearby the line is light or dark? Run *View Light 3.rbt* with calibrated robot. Search the entire area that robot can be on searching up to white line. Record minimum and maximum values.
  - If the maximum value if less than 50, then it is a dark area.
  - If the minimum value is greater than 50, then it is a light area.
- If neither is true, consider approaching the line further down where the color is more uniform.

WHITE AND BLACK LINE THRESHOLDS

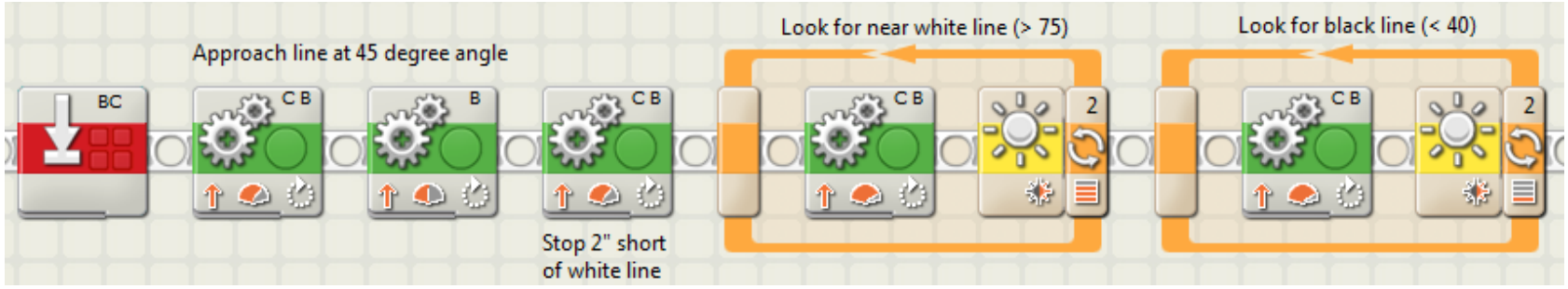
- Good values are 20 - 25 for the black line threshold and 75 - 80 for the white line threshold.
- KEY POINT: Don't pick black / white line thresholds that are too tight.
- In the 2011 Food Factor VA/DC FLL Championship, the Sea Monsters used 10 for the black line threshold, heading towards the thermometer and timer. Even though the robot had been calibrated on each table, this threshold was too tight, resulting in the robot missing the line and costing 43+ points and Sea Monsters' sixth consecutive robot performance trophy.
- It should be noted that the Sea Monsters ran this program at least 100 times in practice without ever encountering this issue. This demonstrates the need to take into account light variability at different tables and not rely just on calibration.

OTHER CONSIDERATIONS FOR APPROACHING LINE

- Approach the line at a sharp angle (at least 45°). Two reasons:
  - Reduces the size of the area that robot must search through to find the line.
  - Reduces the line's cross-section where the robot will be starting.
- Pick a path that stays clear of mission objects.

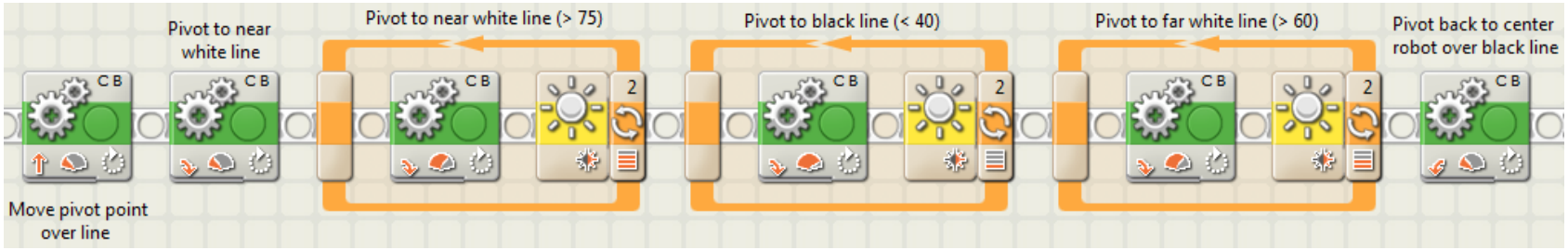
PROGRAM FOR FINDING THE LINE (DARK REGION)

- Below is the beginning portion of the "T" block (*GET TO T.rbt*). This block gets the robot to the T intersection below the rat as seen on page 21.
- There are two loops: the first looks for the white line and the second looks for the black line. They both use sensor #2, which is on the right hand side. We could have used either light sensor, but sensor #2 would be more likely to cross the line.
- In the loop, the B-C motors advance 3 degrees per iteration or about 2 mm per iteration.
- If we were searching in a light area, the first loop could be removed, but a lower threshold would then be used in the second loop.



ALIGNING TO THE LINE

- Since the robot is now just over the line and at an angle, we want to move the robot so that its pivot point (the middle point between the two wheels) is over the line and the two light sensors are bracketing the line.
- Procedure:
  - Move robot forward by the distance of the light sensor to the motor axis, which should put the pivot point over the black line.
  - Rotate the robot so that the leading light sensor is not yet or just over nearer white line.
  - Continue rotating robot so that it first senses white line, then center black line and finally second white line.
- Below is the next portion of *GET TO T.rbt*.

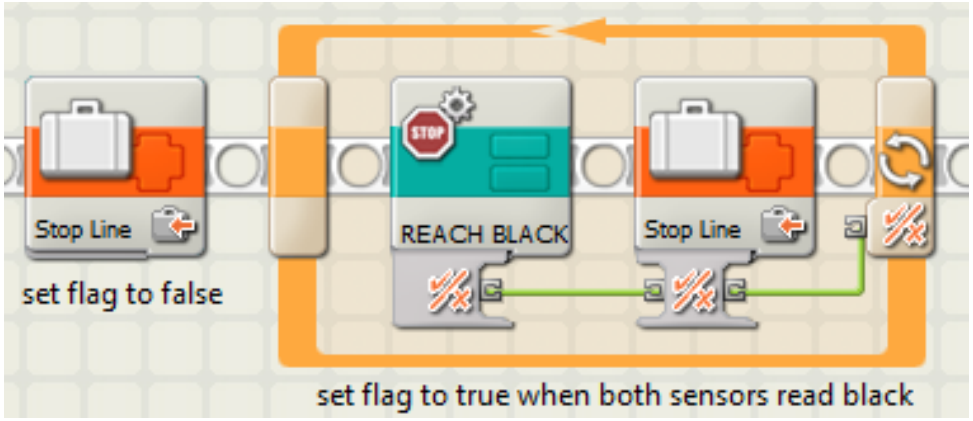


STOPPING ALONG THE LINE

- There are at least four triggers for stopping the robot as it is following a line:
  1. Time (use this when colliding into a wall or object)
  2. Crossing another line (T intersection)
  3. Sensor reading (such as a touch or ultrasound sensor to determine that a wall or object has been reached)
  4. Distance traversed (only use when the robot knows precisely from where it is starting its line following)
- As noted on page 12, *LINE-TIME.rbt* uses a timer to set the Stop Line variable when a stop should be triggered. Replace the timer with logic to customize for the other three triggers.

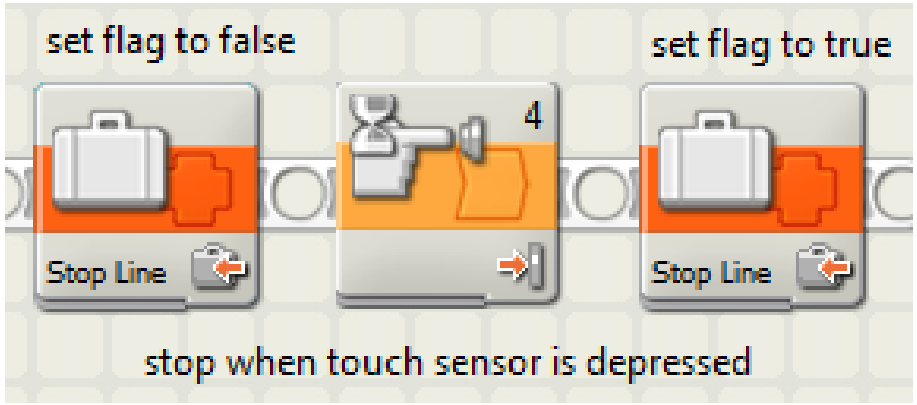
STOPPING AT T INTERSECTION

- *LINE-BLACK.rbt* uses the *REACH BLACK* block to detect when both light sensors are over black. This causes the loop to terminate and the Stop Line variable to be set.



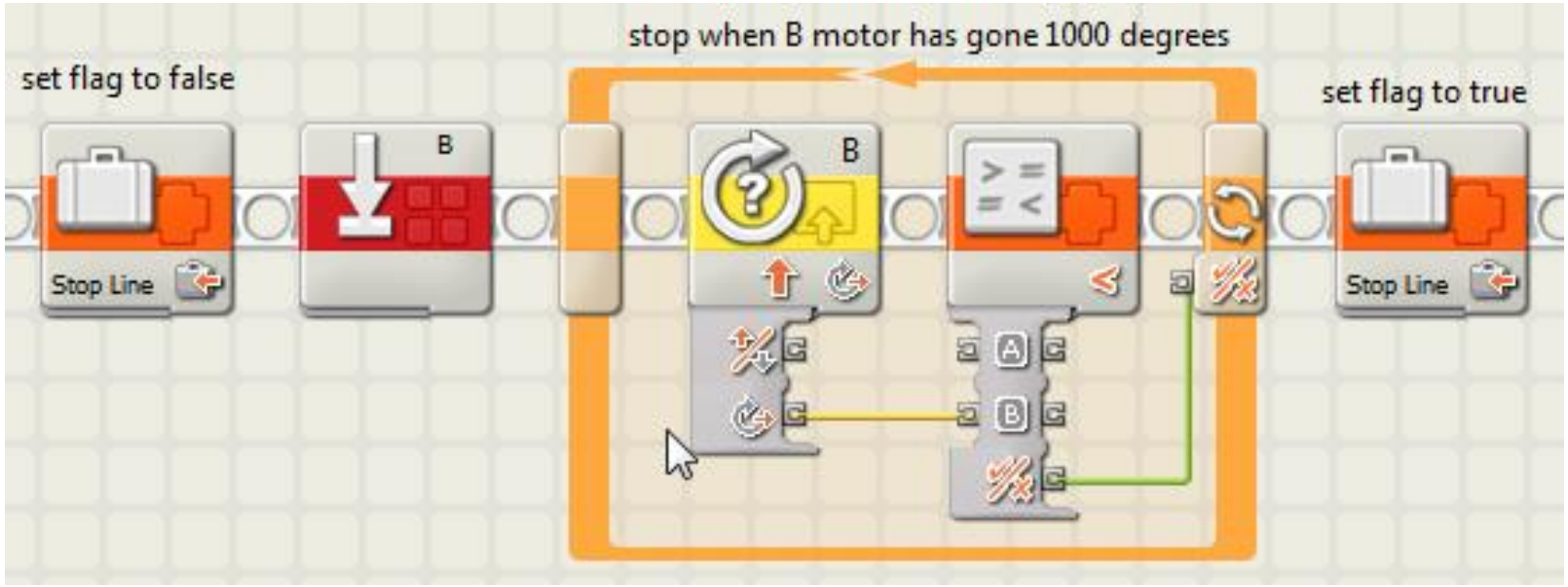
STOPPING ON SENSOR TRIGGER

- *LINE-SENSOR.rbt* stops when a touch sensor mounted on the front of the robot touches an object near the line.



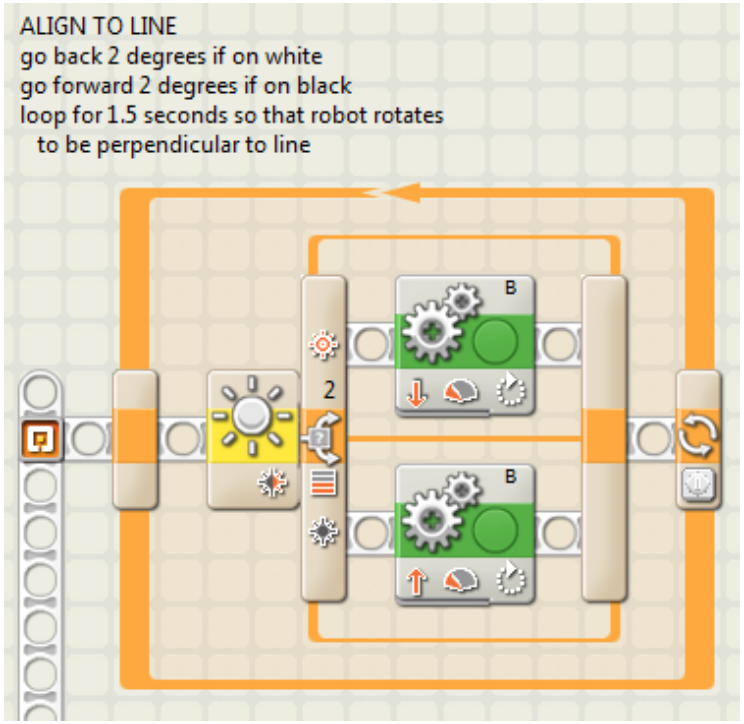
STOPPING ON DISTANCE TRAVERSED

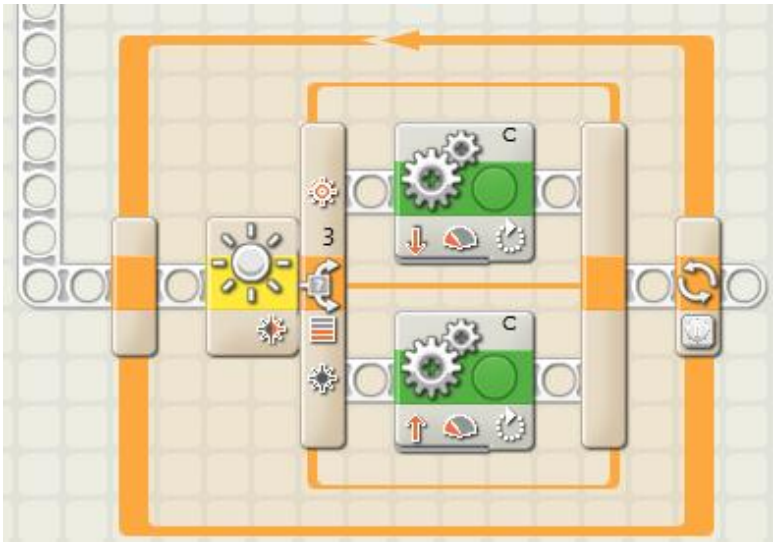
- *LINE-LENGTH.rbt* stops when the B motor has moved 1000 degrees.



ORIENTING THE ROBOT AFTER STOPPING

- KEY POINT: When the robot stops at a T intersection, use the block REWIND, which will put both of the robot's light sensors directly over the forward edge of the T. The robot will then know both its position and orientation accurately.





STRESS TESTING DESIGN

- KEY POINT: Stress test all of your light sensor missions.
- Stress test #1: run the missions in a darkened room.
- Stress test #2: run the missions with a bright flashlight illuminating the same mat spots that are illuminated by the LEDs.
- Adjust thresholds or add shading to make more consistent.
- The above stress tests should have uncovered the line search threshold problem that the Sea Monsters experienced at the VA/DC FLL Championship described earlier.

KEY POINT SUMMARY

- Best performance is when the light source is on and the light sensor is from 5 to 8 mm above mat (page 3).
- For most consistent performance, shading the light sensors will minimize the effects of ambient light on the light sensor readings. However, judicious picking of sensor thresholds and testing with a variety of light levels should minimize the need for shading. (Page 3)
- Calibrate the light sensors to minimize variability between tables (page 4).
- The relative difference between the left and right motor power levels have significant effects on performance (page 6).
- Use two light sensors to bracket line (page 8).

KEY POINT SUMMARY (continued)

- Pick two light sensors that have matching sensor measurement values for black and white (page 9).
- Finding the line is the hardest aspect in line following and has the greatest risk for failure (page 19).
- Don't pick black / white line thresholds that are too tight (page 23).
- When the robot stops at a T intersection, use the block REWIND, which will put both of the robot's light sensors directly over the forward edge of the T. The robot will then know both its position and orientation accurately. (Page 31)
- Stress test all of your light sensor missions (page 32).