

Week 3

Experiment # 9

Section 1.

Simply by adding some forward motion to your Boe-Bot, you can turn it into a light-seeking robot, a photophile. An interesting experiment to try is to program the Boe-Bot to move forward and seek out light. Then, take it into a dark room with the door open to a brighter room. Assuming there are no obstacles in its way, the Boe-Bot will make its way to the door and exit the dark room.

Objective: To Program the Boe-Bot for Light Following

Required parts: Same as the previous Experiment.

Circuit building: Same as the previous Experiment.

Programming for Light Following: Programming the Boe-Bot to follow light requires that only a few modifications to Program Listing in the previous experiment be made. The main change is that measurements within the dead band resulted in no motion in previous experiment. In Program Listing 3.3 when the difference between RC times falls within the dead band, it results in forward motion. Let's see how it works.

Program Listing 3.3

```
' Robotics! v1.5, Program Listing 3.3: Light Follower
' {$Stamp bs2} ' Stamp Directive.
'----- Declarations -----
left_photo var word ' For storing measured RC times of
right_photo var word ' the left & right photoresistors.
'----- Initialization -----
output 2 ' Set P2 to output.
freqout 2, 2000, 3000 ' Program start/restart signal.
low 12 ' Set P12 and 13 to output-low.
low 13
'----- Main Routine -----
main:
' Measure RC time for right photoresistor.
high 3 ' Set P3 to output-high.
pause 3 ' Pause for 3 ms.
rctime 3,1,right_photo ' Measure RC time on P3.
' Measure RC time for left photoresistor.
high 5 ' Set P5 to output-high.
pause 3 ' Pause for 3 ms.
rctime 5,1,left_photo ' Measure RC time on P5.
' Check if difference between RC times is within the deadband, 2 in this
case.
' If yes, then forward. If no then skip to check_dir subroutine.
if abs(left_photo-right_photo) > 2 then check_dir
forward_pulse:
pulsout 12, 500
pulsout 13, 1000
pause 20
goto main
' Jump to either right_turn or left_turn depending on which RC time is
larger.
check_dir:
if left_photo > right_photo then right_pulse
if left_photo < right_photo then left_pulse
'----- Navigation Routines -----
left_pulse: ' Apply one pulse to left then
pulsout 12, 500
pulsout 13, 500
pause 20
goto main ' go back to main routine.
right_pulse: ' Apply one pulse to right then
pulsout 12, 1000
pulsout 13, 1000
pause 20
goto main ' go back to main routine.
```

How the Light Follower Program Works: As in the previous program, the first **if...then** statement tests for a difference in RC time measurements within the dead band. This statement has been modified so that it skips the **forward_pulse** routine if the difference between RC times falls outside the dead band. On the other hand, if the difference in RC times is within the dead band, the forward pulse is executed. After the forward pulse, the program is directed back to **main** and the RC times are checked again.

```
if abs(left_photo-right_photo) > 2 then check_dir
forward_pulse:
pulsout 12, 500
pulsout 13, 1000
pause 20
goto main
```

If the difference between RC times is not within the dead band, the program skips to the **check_dir** label. The **if...then** statements following the **check_dir** label are used to decide whether to apply a pulse to the left or a pulse to the right depending on the inequality of the **right_photo** and **left_photo** values. In this way, the program either applies a single forward pulse or a single turn pulse each time the photoresistors are checked.

```
check_dir:
if left_photo > right_photo then right_pulse
if left_photo < right_photo then left_pulse
```

Tasks:

1. Repeat the previous Your Turn exercise. You can now lead your Boe-Bot around with a flashlight.
2. Instead of pointing the photoresistors at the surface directly in front of the Boe-Bot, point them upward and outward as shown in Figure 3.2. With the photoresistors adjusted this way, the Boe-Bot will roam on the floor and try to always find the brightest place.

Depending on the luminance gradient, you may have to increase the deadband to smooth out the Boe-Bot's light roaming. Alternatively, the deadband may need to be decreased to make it more responsive to seeking out the brighter areas.

Section 2.

If the Boe-Bot can be programmed to follow a flashlight beam focused in front of it, why can't it follow a white stripe on a black background? The answer is, there's no good reason. The Boe-Bot can follow a white stripe on a black background, and it's a project in this chapter's Projects section. By the same token, the Boe-Bot should be able to follow a black stripe on a white background. Regardless of the color of the stripe, this activity is generically referred to as "line following."

The recommended width for the black stripe is about 5 cm. either construction paper or electrical tape works fine. With some calibration along with controlled lighting conditions, the Boe-Bot is a very faithful stripe follower.

- Shadows and bright lights can be misleading, so try to keep the lighting as uniform as possible. For example, overhead fluorescent lights with no light from windows will work well.
- Also, make sure to bend the photo resistors as far over the front of the Boe-Bot as possible. In other words, readjust the photo resistors from flashlight beam following.

Objective: To Program the Boe-Bot for Line Following

Required parts: Same as the previous Experiment.

Circuit building: Same as the previous Experiment.

Programming for Line Following: By changing one value and three parameters from the previous example program, the Boe-Bot can now follow bold, black stripes on a white background. Program Listing 34 demonstrates this. In the comments at the beginning of the program, data from tests performed using Program Listing 3.1 are shown. The white **rctime** readings were taken while the photo resistors were looking at a white surface in front of the Boe-Bot. The black **rctime** readings were taken while the photoresistor were looking at a black surface. The average difference between the black and white readings is 77 in this example. The average can go as low as 45, and the example should still work without a hitch. When the difference is smaller, the dead band value will have to be decreased. When the difference is larger, the dead band will have to be increased for better performance.

- Test the photoresistors using Program Listing 3.1. Use the information gathered to guess at a deadband value for stripe following.
- Adjust the dead band value to your predictions, and then run your modified version of Program Listing 3.4. Try different dead band values until you find one that makes your Boe-Bot perform well when it follows the stripe.

Program Listing 3.4

```
' Robotics! v1.5, Program Listing 3.4: Black Stripe Follower
' {$Stamp bs2} ' Stamp Directive.
' Program Listing 4.1 readings with Boe-Bot looking at white/black surfaces.
' color left right
' white 58 67
' black 127 152
'----- Declarations -----
left_photo var word ' For storing measured RC times of
right_photo var word ' the left & right photoresistors.
'----- Initialization -----
output 2 ' Set P2 to output.
freqout 2, 2000, 3000 ' Program start/restart signal.
low 12 ' Set P12 and 13 to output-low.
low 13
'----- Main Routine -----
main:
' Measure RC time for right photoresistor.
high 3 ' Set P3 to output-high.
pause 3 ' Pause for 3 ms.
rctime 3,1,right_photo ' Measure RC time on P3.
' Measure RC time for left photoresistor.
high 5 ' Set P5 to output-high.
pause 3 ' Pause for 3 ms.
rctime 5,1,left_photo ' Measure RC time on P5.
' Check if difference between RC times is within the deadband, 7 in this
case.
' If yes, then forward. If no then skip to check_dir subroutine.
' IMPORTANT: The deadband value (currently 7) should be made smaller in
' brighter rooms and larger in darker rooms. Otherwise, the Boe-Bot will not
' detect the black stripe. It will take some fine tuning to optimize the
' Boe-Bot's ability to follow stripes.
if abs(left_photo-right_photo) > 7 then check_dir
forward_pulse:
pulsout 12, 500
pulsout 13, 1000
pause 20
goto main
' Jump to either right_turn or left_turn depending on which RC time is
larger.
check_dir:
if left_photo < right_photo then right_pulse
if left_photo > right_photo then left_pulse
'----- Navigation Routines -----
left_pulse: ' Apply one pulse to left then
pulsout 12, 500
pulsout 13, 500
pause 20
goto main ' go back to main routine.
right_pulse: ' Apply one pulse to right then
pulsout 12, 1000
pulsout 13, 1000
```

```
pause 20  
goto main ' go back to main routine.
```

How the Black Stripe Follower Program Works: The three lines below are the only ones changed between Program Listing 3.3 and Program Listing 3.4. The dead band was increased from “2” to “7,” and the inequality signs (< and >) of the lower two if...then statements were swapped. As noted earlier, that’s all it takes to change a robotic light seeker (photophile) into a light avoider (photophobe). Light avoidance is key to this program.

```
if abs(left_photo-right_photo) > 7 then check_dir  
.  
.  
.  
if left_photo < right_photo then right_turn  
if left_photo > right_photo then left_turn
```

Tasks :

1. Try a black stripe with a 45° turn in the middle of it.
2. Try a black strip with a 90° turn in it, and see if you can pick a deadband that will navigate it.
Remember, you may need to adjust your deadband to succeed in these maneuvers.
3. For either or both of the maneuvers above, find the upper and lower limits of deadband values with which the Boe-Bot still can successfully navigate.