

## Week 3

### Experiment # 8

If you focus a flashlight beam in front of the Boe-Bot, the circuit and programming techniques just discussed can be used to make the Boe-Bot turn so that it's pointing at the flashlight beam. Make sure the photo resistors are pointed so that they can make a light comparison. Aside from each being pointed 45° outward from the center-line of the Boe-Bot, they also should be oriented so they are pointing 45° downward from horizontal. In other words, point the faces of the photo resistors down toward the table top. Then, use a bright flashlight to make the Boe-Bot track the direction of the light.

**Objective:** To Program the Boe-Bot to Point at the Light

**Required parts:** Same as the previous Experiment.

**Circuit building:** Same as the previous Experiment.

**Programming the Boe-Bot to Point at the Light:** Getting the Boe-Bot to track a light source is a matter of programming it to compare the value measured at each photoresistor. Remember that as the light gets dimmer, the photoresistor's value increases. So, if the photoresistor value on the right is larger than that of the photoresistor on the left, it means it's brighter on the left. Given this situation, the Boe-Bot should turn left. On the other hand, if the `rcTime` of the photoresistor on the left is larger than that of the photoresistor on the right, the right side is brighter and the Boe-Bot should turn right. To keep the Boe-Bot from changing directions too often, a parameter for deadband is introduced. Deadband is a range of values wherein the system makes no attempt at correction. If the numbers go above or below the deadband, then the system corrects accordingly. The most convenient way to measure for deadband is to subtract the left `rcTime` from the right `rcTime`, or visa versa, then take the absolute value. If this absolute value is within the deadband limits, then do nothing; if otherwise, program an appropriate adjustment.

- Enter and run Program Listing 3.2.
- Shine a bright flashlight in front of the Boe-Bot. When you move the flashlight, the Boe-Bot should rotate so that it's pointing at the flashlight beam.
- Instead of using a flashlight, use your hand to cast a shadow over one of the photoresistors. The Boe-Bot should rotate away from the shadow.

### Program Listing 3.2

```
' Robotics! v1.5, Program Listing 3.2: Light Compass
' {$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 ' Declare a variable for counting.
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.
' Take the difference between right_photo and left_photo, then decide what
to do.
if abs(left_photo-right_photo) < 2 then main
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 Compass Works :** Program Listing 3.2 takes RC time measurements and first checks to see if the difference between the values returned by the **rctime** commands fall in the deadband using the command:

```
if abs(left_photo - right_photo) < 2 then main
```

If the difference between RC times is within the deadband, the program jumps to the **main:** label. If the measured difference in RC times is not within the deadband, two **if...then** statements decide which routine to call,

```
left_pulse or right_pulse.  
if left_photo > right_photo then right_pulse  
if left_photo < right_photo then left_pulse
```

The **left\_pulse** routine is shown below. Note that the **pulsout** commands and the **pause** command are not nested within a **for...next** loop. Instead, just one single pulse with a slightly smaller than usual **pause**

is delivered, then control is returned to the **main** routine. This allows the program to check and update the photoresistor values between each servo pulse. Note that the pause value is not 20 ms. This is because each **rctime** command takes a certain amount of time, which can be subtracted from the necessary **pause period**. The average of the combined pauses and RC times was 10 ms for the lighting conditions used in this example. Your lighting conditions are likely to be different.

```
left_pulse:  
pulsout 12, 500  
pulsout 13, 500  
pause 10  
goto main
```

## Tasks :

In a darker area, not only will the photoresistor values be larger, so will the difference between them. You may have to increase the deadband in low ambient light to detune the Boe-Bot to small and changing variations in light. The lower the light levels, the less you need the **pause** statements. If the Boe-Bot's performance starts to decrease, it's probably because the time between pulses has exceeded 40 ms. The first line of defense for this problem is to reduce the **pause period** in each subroutine to zero. The second line of defense is to check photoresistors during alternate pulses. That way, after the first pulse, the right photoresistor could be checked. Then, after the second pulse, the left photoresistor could be checked. You can try your hand at developing code that does this in this chapter's Projects section.

The deadband value is currently set to “2” in the expression:

```
if abs(left_photo-right_photo) < 2 then main
```

1. Experiment with different ambient light levels and their effect on deadband by trying this experiment in lighter and darker areas. In lighter areas, the deadband value can be made smaller, even zero. In darker areas, the deadband value should be increased.
2. Swap the conditions in the second and third **if...then** statement in Program Listing 3.2. Then re-run the program. Now your Boe-Bot points away from the light.