

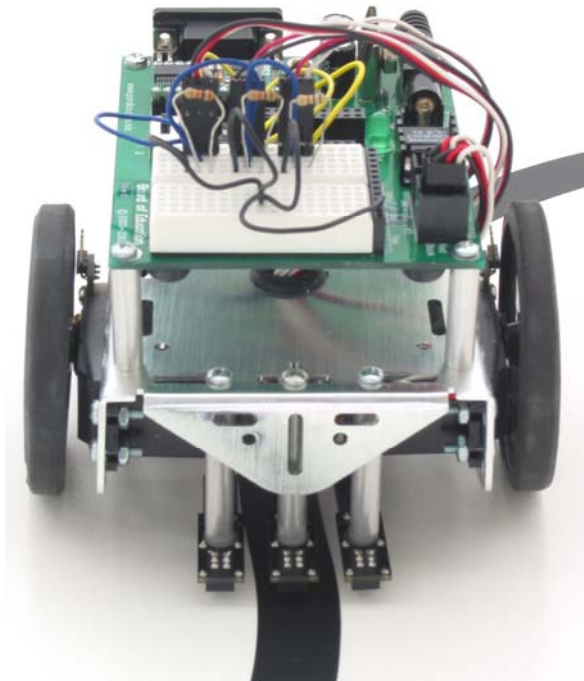
QTI Line Follower AppKit for the Boe-Bot (#28108)

Boe-Bot Line Following with QTIs

Just one of the many Boe-Bot[®] robot QTI sensor applications is line following. QTIs are inexpensive and great for line following because you can adjust their position, add or remove detectors, and use them in different ways depending on the course.

The QTIs were originally developed for the Parallax SumoBot[®] robot, where they are used to detect the white edge around the competition ring. The SumoBot Manual demonstrates how QTI modules can be used as analog sensors with the RCTIME command. By adding a resistor to the QTI circuit, you can make the sensor a purely digital device that returns a 1 when it detects a black line or a 0 if it detects a white background.

The QTI positions are adjustable for different sizes and types of lines. This activity demonstrates how the QTIs can be used for digital line following on a simple 3/4-inch wide electrical tape course with a white background. For a faster line-following application using all 4 QTI sensors, check out the Stamps in Class Mini Projects at <http://forums.parallax.com>.



Kit Contents

Part #	Description	Qty
805-00001	10-inch 3-lead extension cable	4
800-00016	3-inch jumper wires (bag of 10)	2
150-01030	Resistor - 10 kΩ	4
555-27401	QTI sensor	4
451-00303	3-pin male-male header	4
710-00007	7/8-inch screw, pan head, 4-40	4
700-00060	Standoff, round, 1-inch, 4-40	4
713-00007	Spacer, round, 1/2 inch	4
700-00002	3/8-inch screw, pan head, 4-40	4
700-00015	Washer, nylon, #4	4

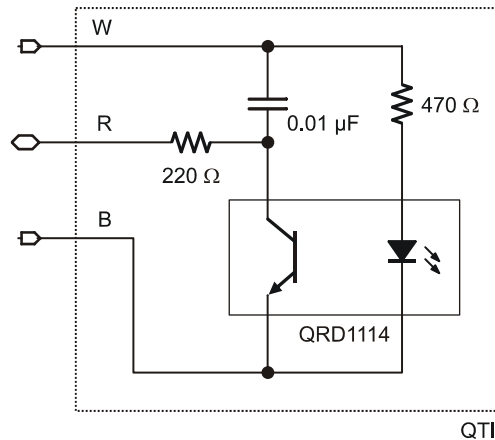


For this activity, you will need to supply your own:

- Built and tested Boe-Bot
- Black 3/4-inch electrical tape
- White poster board

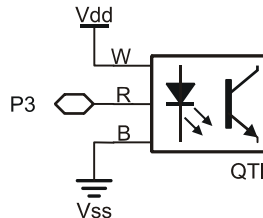
A Closer Look at the QTI

The QTI module is designed for close proximity infrared (IR) detection. Take a look at the small square black box just above the QTI label. It's nested below the capacitor and between the two resistors. That's a QRD1114 reflective object sensor. There's an infrared diode behind its clear window and an infrared transistor behind its black window. When the infrared emitted by the diode reflects off a surface and returns to the black window, it strikes the infrared transistor's base, causing it to conduct current. The more infrared incident on the transistor's base, the more current it conducts.



When used as an analog sensor, the QTI can detect shades of gray on paper and distances over a short range if the light in the room remains constant. With this circuit, you can set P3 high and then test it with **RCTIME** to measure how long it takes the capacitor to discharge through the IR transistor. Since the IR transistor conducts more or less current depending on how much IR it receives, the **RCTIME** measurement can give you an indication of distance or shade of gray.

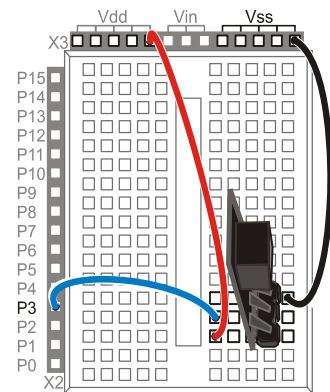
If all you want to know is whether a line is black or white, the QTI can be converted to a digital sensor by adding a 10 kΩ resistor across its W and R terminals. After doing so, the QTI behaves similarly to the circuit on the right below.



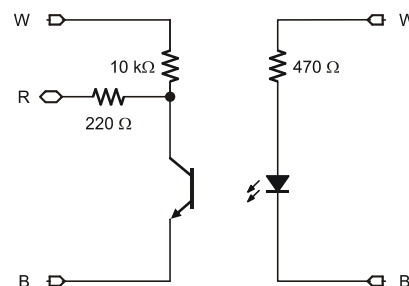
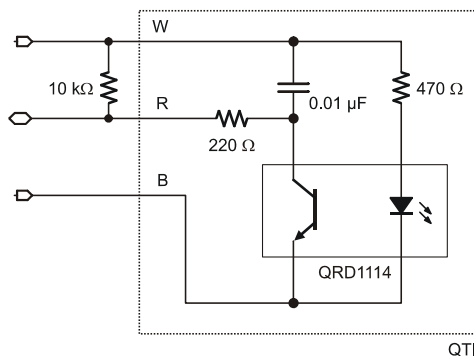
```

' AnalogQti.bs2
' {$STAMP BS2}
' {$PBASIC 2.5}
time VAR word

DO
  HIGH 3
  RCTIME 3, 1, time
  DEBUG CLS, ? time
  PAUSE 100
LOOP
  
```



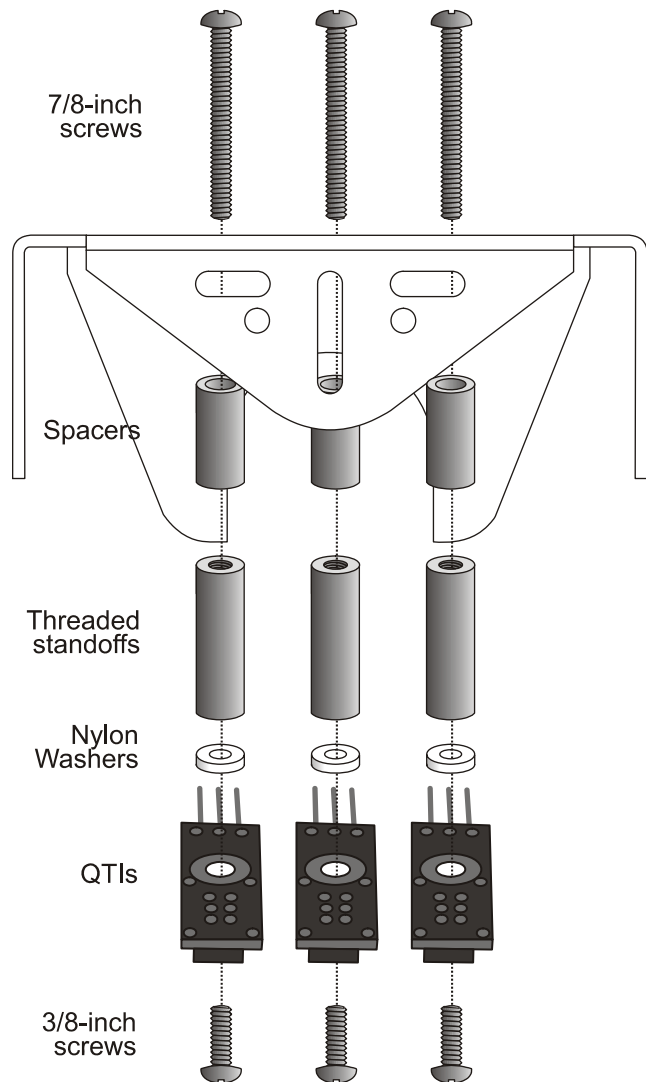
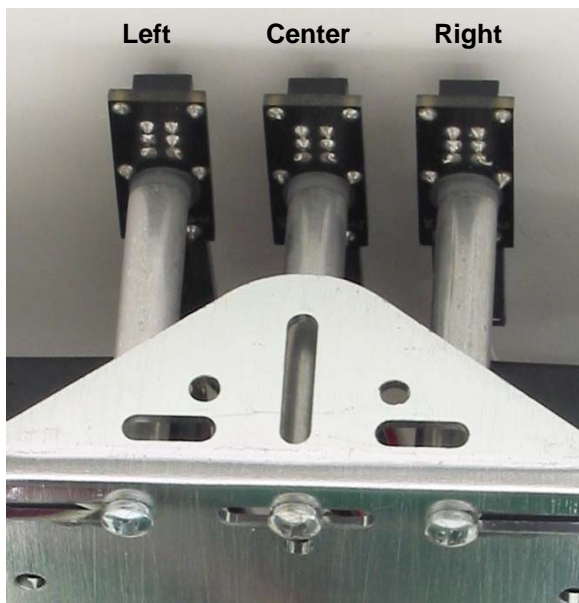
When W is connected to Vdd and B is connected to Vss, the R terminal's voltage will drop below 1.4 V when the IR transistor sees infrared reflected from the IR LED. When the IR LED's signal is mostly absorbed by a black surface, the voltage at R goes above 1.4 V. Since the BASIC Stamp interprets any voltage above 1.4 V as 1 and any voltage below 1.4 V as 0, this circuit gives us a quick and easy way to detect a black line on a white background.



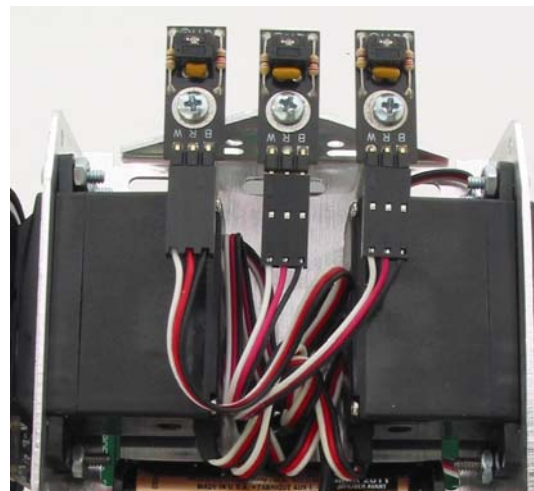
Mounting the QTIs

- ✓ Use the 7/8-inch screws to attach the 1/2-inch spacers and 1-inch standoffs to the underside of the chassis.
- ✓ Use the 3/8-inch screws to attach the QTIs and nylon washers to the other ends of the standoffs.

As with the rest of the Boe-Bot material, you can also imagine yourself sitting in the Boe-Bot's driver seat. When you look down over the front of the breadboard, you will see the left, center, and right QTIs.

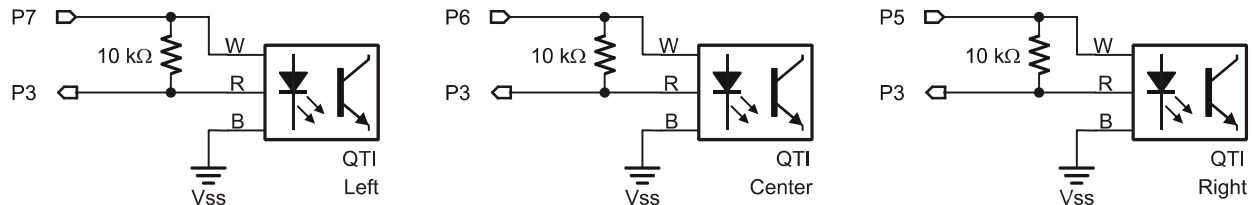


- ✓ Connect each servo extension cable to a QTI sensor, making sure the black wire lines up with the "B", the red wire lines up with the "R", and the white wire lines up with the "W" on the QTI.
- ✓ Pull the cables through the large hole in the Boe-Bot chassis one at a time. It's between the servos and should be lined with a rubber grommet.
- ✓ Label the cables, L for left, C for Center, and R for right.

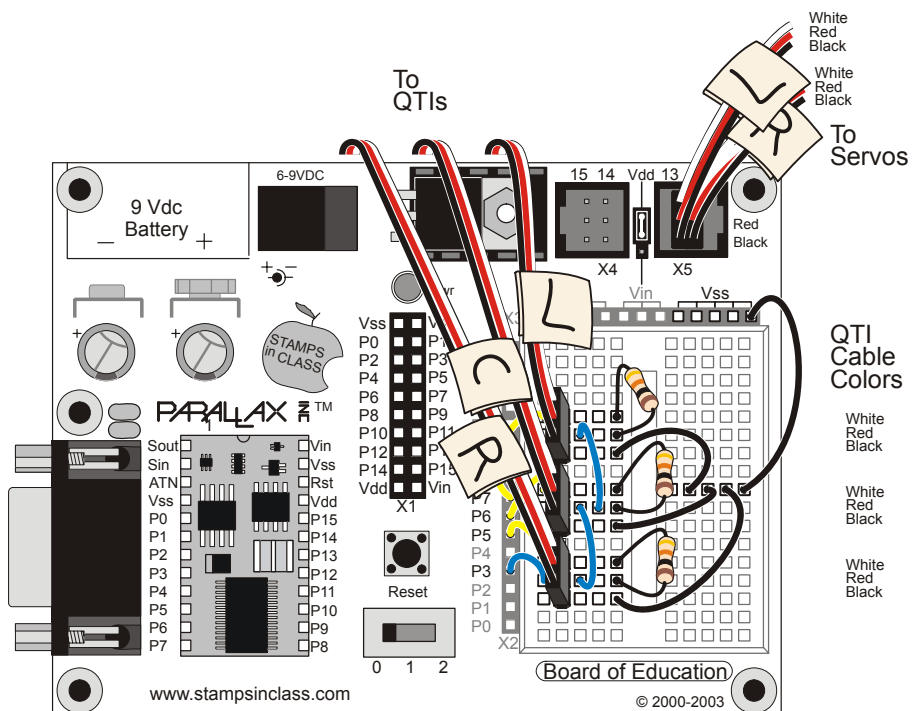
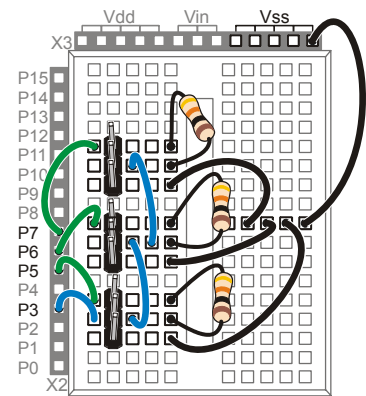


Building the Sensing Circuits

If you apply 5 V to a QTI's W pin, its R pin will rise above 1.4 V if it detects a black surface, or fall below 1.4 V if it detects a white surface. In other words, the QTI sends a binary-1 if it does not see its IR reflection or a binary-0 if it does. Only one QTI should be turned on at any given time to make sure that one QTI doesn't see the reflection of another QTI's IR signal. With this rule in mind, P5, P6, and P7 each connect to a QTI's W pin. P5 connects to the right QTI, P6 to the center QTI, and P7 to the left QTI. All the B pins are tied to Vss. All the R pins are tied together and connected to P3. We'll turn each QTI on, one at a time, read P3, and then turn that QTI off again. If the QTI that receives 5 V sees its reflection, it will drive the voltage at P3 low; otherwise, it will be pulled high by the 10 kΩ resistor.



- ✓ Insert the 3-pin headers into the breadboard.
- ✓ Add the resistors and jumper wires as shown.
- ✓ Plug the cables for the right, center and left QTIs into their corresponding 3-pin headers on the breadboard: Right to P5, Center to P6, Left to P7.
- ✓ Check to make sure you have your cables connected so that the black cable wires are connected to Vss, the red cable wires are bussed to P3, and each cable's white wire is connected to P5, P6, or P7 respectively.
- ✓ Double-check to make sure each cable is properly connected to its QTI (B = black wire, R = red, and W = white).



Testing for Line Detection

It's a good idea to test all the sensors with the Debug Terminal before taking the Boe-Bot for a spin on the line following course.

- √ Affix a few inches of 3/4-inch wide electrical tape to a white sheet of paper, or use the test line to the right of the program listing.
- √ Enter and run `QtiBwDetect.bs2`
- √ Place the left QTI directly over the electrical tape (and the other QTIs over white background).
- √ The Debug Terminal should read 100.
- √ Place the center QTI over the electrical tape; the Debug Terminal should display 010.
- √ Place the right QTI over the electrical tape; the Debug Terminal should display 001.
- √ Place the Boe-Bot so that the electrical tape is between the right and center QTIs; the Debug Terminal should display 011.
- √ Place the Boe-Bot so that the electrical tape or stripe is between the left and center QTIs; the Debug Terminal should display 110.

If you had problems with two QTIs sensing 1 when only one of them should have, try adjusting the standoffs so that the QTIs are further apart. Not so far that you can get 000 when the line is between two QTIs though! On the other hand, if only one QTI sensed 1 when the stripe was between two of them, they may need to be positioned closer together. Otherwise, your Boe-Bot is ready for line following.

```
' QtiBwDetect.bs2
' Boe-Bot detects electrical tape with 3 QTI modules.

'{$STAMP BS2}
'{$PBASIC 2.5}

qtiLeft      VAR      Bit
qtiCenter    VAR      Bit
qtiRight     VAR      Bit

DO

  HIGH 5: PAUSE 1: qtiRight  = IN3: INPUT 5
  HIGH 6: PAUSE 1: qtiCenter = IN3: INPUT 6
  HIGH 7: PAUSE 1: qtiLeft   = IN3: INPUT 7

  DEBUG HOME, BIN1 qtiLeft, BIN1 qtiCenter, BIN1 qtiRight

  PAUSE 100

LOOP
```



Always keep in mind that each W (on/off) line is tied to an individual I/O pin (P5, P6, P7), but all the R (sense) lines are tied to P3. The program turns on power to the right QTI with the command `HIGH 5`. `PAUSE 1` allows time for the voltage at the QTI's R pin to settle to its final value. Then, the command `qtiRight = IN3` stores the output sensed by `IN3` in the `qtiRight` bit variable. `INPUT 5` turns the right QTI off. The process repeats twice more, storing the output of the center QTI in the `qtiCenter` variable, and the output of the left QTI in the `qtiLeft` variable.

The `DEBUG` command displays the binary QTI readings. The leftmost digit indicates the state of the left QTI, the center digit indicates the center QTI, and the right digit indicates the right QTI. `PAUSE 100` is there to prevent serial buffer overload on slower computers.

Simple Line-Following

QtiLineFollow.bs2 is designed to start following a line as soon as you place the center QTI over the electrical tape. It will stop line following as soon as it runs out of electrical tape. Start with an easy course, like a large S shape as shown below. If the QTIs passed the "Testing for Line Detection" tests, it should navigate the course with ease.

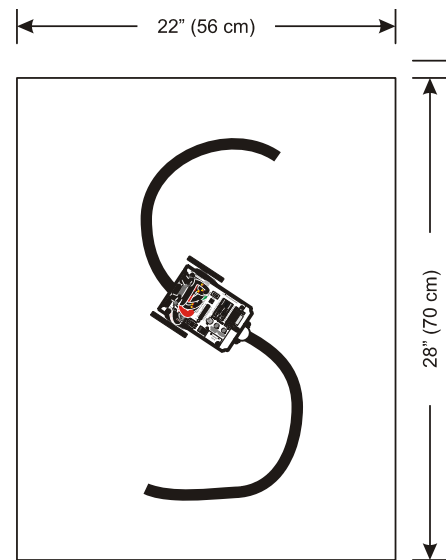
- ✓ Enter and run QtiLineFollow.bs2.
- ✓ Place the Boe-Bot on the course so that it straddles the line with the center QTI over the electrical tape.

If your left and right QTI's cables are reversed, the Boe-Bot will appear to try to jump off the line at the first opportunity. Otherwise, it should faithfully follow the line until it reaches the end of the tape.

```
' QtiLineFollow.bs2
' Boe-Bot follows electrical tape with 3 QTI modules.
' {$STAMP BS2}
' {$PBASIC 2.5}

qti          VAR      Nib
DO
  Right:    HIGH 5: PAUSE 1: qti.BIT0 = IN3: INPUT 5
  Center:   HIGH 6: PAUSE 1: qti.BIT1 = IN3: INPUT 6
  Left:     HIGH 7: PAUSE 1: qti.BIT2 = IN3: INPUT 7

  SELECT qti
    CASE %010 ' Forward
      PULSOUT 13, 850
      PULSOUT 12, 650
    CASE %011 ' Pivot right
      PULSOUT 13, 850
      PULSOUT 12, 750
    CASE %001 ' Rotate right
      PULSOUT 13, 850
      PULSOUT 12, 850
    CASE %110 ' Pivot Left
      PULSOUT 13, 750
      PULSOUT 12, 650
    CASE %100 ' Rotate Left
      PULSOUT 13, 650
      PULSOUT 12, 650
  ENDSELECT
  PAUSE 20
LOOP
```



Instead of three bit variables, one nibble variable stores the three bit values. One bit is unused, but it will come in handy for working with four QTIs. The right QTI's output is stored in `qti.BIT0`, the center in `qti.BIT1`, and the left in `qti.BIT2`. A `SELECT...CASE` statement examines the pattern of 1s and 0s in the `qti` variable, and then delivers servo pulses accordingly. Only five of the eight possible cases are evaluated, one for each individual QTI, one for left + center, and one for right + center.

- ✓ Try other courses to test the limits of the program.
- ✓ Try modifying the program to solve courses that the unmodified program could not solve.
- ✓ Modify the program so that it smoothes out the Boe-Bot's responses to changes in the line's direction. Your code should take steps toward a maximum speed each time it detects that a given pattern is detected.
- ✓ Detect intersections and make random turns.
- ✓ Add a fourth QTI and modify the `SELECT...CASE` so that the middle two QTIs follow the line, and the outer ones are only used for detecting sharper turns.
- ✓ Try line following with just two QTIs. How about with just one? Can you do it?
- ✓ Challenge a friend, or set up a Boe-Bot line following competition.

For more Boe-Bot and QTI applications, check the Stamps in Class forum at <http://forums.parallax.com>.