



# Feedback

**Y**our LEGO Mindstorms NXT kit comes with a collection of motors and sensors. And the NXT Brick has three built-in timers plus the buttons on the Brick. Would it surprise you to learn that all of these items are able to provide some sort of feedback to the NXT Brick? The sensors are a little obvious; a sensor is designed to respond to external conditions such as light, touch, or sound and report this information to the Brick. But what about motors? And how can a timer be used as input or feedback to the Brick? All these questions have answers, and this chapter provides them.

## What's Your Condition?

Let's start with the word *condition*. A traffic light has three conditions: red, yellow, and green. A light switch has two conditions: on and off. Just using these two simple examples, I can give you some pseudo-code for SPOT:

**Me:** SPOT, move forward until the traffic light turns red.

**Me:** SPOT, display the words "Light On" on your LCD screen until the light switch is off.

In both these examples, I'm assuming that SPOT has eyes and can see the traffic light or light switch. If SPOT doesn't have eyes, then I would need a way for the traffic light or light switch to provide their conditions to SPOT. The traffic light and light switch could then provide feedback, or input, to SPOT.

Asking a traffic light to provide input isn't realistic, but your NXT robots do have the ability to receive feedback from items such as the sensors. So I can change the pseudo-code for SPOT:

**Me:** SPOT, move forward until the Light sensor reports a value of 20.

**Me:** SPOT, display the word "Hello" until the Touch sensor reports that it has been pressed and released.

**Me:** SPOT, play a C note when the Ultrasonic sensor detects an object 6 inches in front of you.

The motors, too, can provide feedback:

**Me:** SPOT, spin motor A until motor B reports that it has spun 10 degrees.

**Me:** SPOT, display "5 rotations" when motor C has rotated five times.

I mentioned that the Brick has three built-in timers plus three buttons that can be used, so I could also write the following pseudo-code:

**Me:** SPOT, when 20 seconds have elapsed, turn 90 degrees.

**Me:** SPOT, play a B note if I press the left button and a C note if I press the right button.

Okay, so you can see that the sensors, motors, buttons, and timers can provide input to the Brick to control other actions (MOVE, SOUND, and other blocks). You program your robot to perform specific actions based on the conditions of these items. Just like you know a light switch has two conditions, on and off, you need to know the various conditions that the sensors, buttons, timers, and motors possess and can report to the Brick.

So, for the rest of this chapter, I'm going to explain the settings for these items, so you'll know how to configure them properly. Future blocks that we'll cover, including the WAIT, LOOP, and SWITCH blocks, will depend on your understanding of how to properly configure the conditions.

The method I'm going to use to do this is fairly simple: I'll describe each item (sensor, motor, button, and so on) and provide a description of the settings that can be modified within the item's configuration panel. Please note that I am also providing information on the so-called *legacy* items: the previous version of Mindstorms (Robotics Invention System, or RIS) included motors, sensors, and a lamp. These RIS items can be used with NXT robots but require special converter cables to connect to the NXT Brick. I call them "legacy" because they come from the older Mindstorms system, but they are still very useful and fully compatible with the NXT system.

---

**Note** Right now, *only* the Education version of the NXT kit comes with software support for the legacy/RCX items. If you are running the Retail version, you may download NXT programming blocks for the RCX motors and sensors by visiting <http://mindstorms.lego.com/Support/Updates/> and downloading the files from the Legacy Block Library section.

---

## Configuring the Sensors

Let's start with the sensors—NXT Touch, RIS Touch, NXT Sound, NXT Light, RIS Light, NXT Ultrasonic, NXT Rotation, RIS Rotation, and RIS Temperature—which have some interesting rules to abide by.

First, sensors detect a change in a condition. The condition could be a change in light level, a change in volume (sound), or maybe a change in position (movement). The sensor is simply "watching" the changes in its condition. The Light sensor, for example, can detect changes in the level of lighting in the room.

Second, sensor programming blocks can *only* respond to one condition at a time. When you drop a sensor block into an NXT-G program, it has to be configured to test one condition. A single Light sensor, for example, cannot be configured to respond to whether the light in a room is below 80 *and* greater than 50. In order to test both conditions, you would need to use two Light sensor blocks in your program.

Finally, sensors can provide only a Logic type response: True or False. Think of our traffic light for a moment. Let's assume that every car has special traffic light sensors mounted on top. These light sensors are programmed to examine the conditions of the traffic light. Cars will move while the light is green and stop when the light is red. If the light is yellow, the car will slow down.

These traffic light sensors, then, can detect one of three possible conditions: green, yellow, or red. An NXT sensor can only check *one* condition, so if this were an NXT traffic light sensor, each car would require three traffic light sensors:

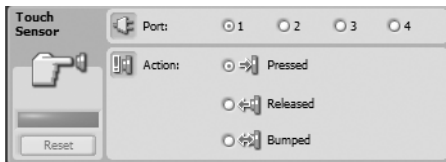
- If the first traffic light sensor detects green, the car should keep moving.
- If the second traffic light sensor detects yellow, the car should start slowing down.
- If the third traffic light sensor detects red, the car should stop.

Notice that *only* one condition can be true at any given time; the light cannot be green and red at the same time. So when the sensor detecting green is triggered, yellow and red cannot be triggered. Therefore, we can examine a sensor's Logic data type for a True/False response to the condition the sensor is configured to monitor.

Let's take a look now at the sensors.

## NXT Touch Sensor

The NXT Touch sensor's configuration panel is shown in Figure 9-1.



**Figure 9-1.** *The NXT Touch sensor's configuration panel*

In the Port section, select the port where the NXT Touch sensor has been connected. The default port for the Touch sensor is Port 1.

In the Action section, you can select Pressed, Released, or Bumped. Remember that Bumped means quickly pressing and releasing the button—less than .5 seconds for the entire press-and-release action.

The NXT Touch sensor's data hub is shown in Figure 9-2. Remember that you can hover your mouse pointer over a data plug to obtain the plug's name.



**Figure 9-2.** *The NXT Touch sensor's data hub*

The port number can be configured using the Port data plug and a Number data type wire with an input value of 1, 2, 3, or 4. This value can also be used as output by dragging a data wire out of the output Port data plug.

The Action data plug can use a Number data type with an input value of 0 for Pressed, 1 for Released, or 2 for Bumped.

The Yes/No data plug can provide an output value of True or False.

The Raw Value data plug can provide an output Number value with a range of 0–1024.

For the remaining blocks in this chapter, I will not be describing the blocks' data hubs—only the blocks' configuration panels (I simply wanted to show you that the sensor blocks can use their data hubs for even more options).

Click the Help menu, and select the “Contents and Index” option to view the Help files. Select a programming block from the left side of the screen (see Figure 9-3). Details for the block's data hub can be found at the bottom of that block's Help documentation (also shown in Figure 9-3). You can always consult this Help documentation to determine the data types for the various data plugs (Number, Logic, or Text).

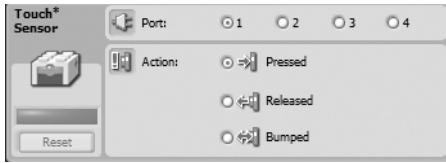
This chart shows the different characteristics of the plugs on the Touch Sensor block's data hub:

Plug	Data Type	Possible Range	What the Values Mean	This Plug is Ignored When...
	Number	1 - 4	1 = Port 1, 2 = Port 2, 3 = Port 3, 4 = Port 4	
	Number	0 - 2	0 = Pressed 1 = Released 2 = Bumped	
	Logic	True/False	Result of comparison	
	Number	0 - 1024	Raw (unscaled) value read from sensor	

**Figure 9-3.** *Touch Sensor Block is selected on the left and the Data Hub details are shown on the right.*

## RIS Touch Sensor

The legacy RIS Touch sensor's configuration panel is shown in Figure 9-4.



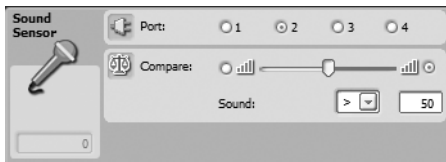
**Figure 9-4.** *The RIS Touch sensor's configuration panel*

In the Port section, select the port where the RIS Touch sensor has been connected. The default port for the Touch sensor is Port 1.

In the Action section, you can select Pressed, Released, or Bumped.

## NXT Sound Sensor

The NXT Sound sensor's configuration panel is shown in Figure 9-5.



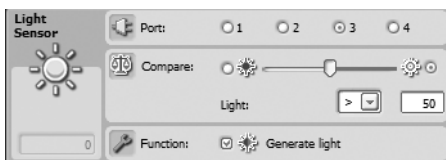
**Figure 9-5.** *The NXT Sound sensor's configuration panel*

In the Port section, select the port where the NXT Sound sensor has been connected. The default port for the Sound sensor is Port 2.

In the Compare section, you must configure the sound value (0–100) as well as whether the sensor will monitor for sounds less than or greater than the configured value. You can use the drag bar to select the value or enter a numeric value in the range of 0–100 in the text box. Select the less than option (<) or the greater than option (>) from the drop-down menu *or* click the left or right radio button to configure this option.

## NXT Light Sensor

The NXT Light sensor's configuration panel is shown in Figure 9-6.



**Figure 9-6.** *The NXT Light sensor's configuration panel*

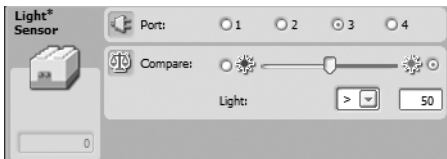
In the Port section, select the port where the NXT Light sensor has been connected. The default port for the Light sensor is Port 3.

In the Compare section, you must configure the light value (0–100) as well as whether the sensor will monitor for a light value less than or greater than the configured value. You can use the drag bar to select the value or enter a numeric value in the range of 0–100 in the text box. Select the less than option (<) or the greater than option (>) from the drop-down menu, *or* click on the left or right radio button to configure this option.

If you leave the “Generate light” box checked, the Light sensor will turn on its built-in LED to provide an artificial light source for assisting with determining light levels. If you uncheck the box, the Light sensor will only detect *ambient light levels*—that is, normal light conditions.

## RIS Light Sensor

The RIS Light sensor’s configuration panel is shown in Figure 9-7.



**Figure 9-7.** *The RIS Light sensor’s configuration panel*

In the Port section, select the port where the RIS Light sensor has been connected. The default port for the Light sensor is Port 3.

In the Compare section, you must configure the light value (0–100) as well as whether the sensor will monitor for a light value less than or greater than the configured value. You can use the drag bar to select the value or enter a numeric value in the range of 0–100 in the text box. Select the less than option (<) or the greater than option (>) from the drop-down menu, *or* click the left or right radio button to configure this option.

## NXT Ultrasonic Sensor

The NXT Ultrasonic sensor’s configuration panel is shown in Figure 9-8.



**Figure 9-8.** *The Ultrasonic sensor’s configuration panel*

In the Port section, select the port where the NXT Ultrasonic sensor has been connected. The default port for the Ultrasonic sensor is Port 4.

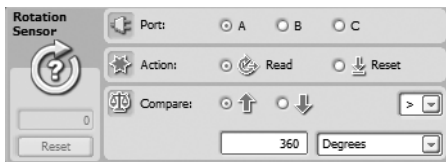
In the Compare section, you must configure the Distance value as well as whether the sensor will monitor for a distance value less than or greater than the configured value. You can use the drag bar to select the value in the range of 0–100, or you may type a value in the text box with a lower limit of 0 but an upper value greater than 100. Using the drag bar limits you to a

range of 0-100, but you are able to type in a value greater than 100. Select the less than option (<) or the greater than option (>) from the drop-down menu, *or* click on the left or right radio button to configure this option.

In the Show section, select either Centimeters or Inches from the drop-down menu.

## NXT Rotation Sensor

The NXT Rotation sensor's configuration panel is shown in Figure 9-9. Remember that the Rotation sensor is built into the NXT motors, so don't go looking for an actual NXT Rotation sensor by itself.



**Figure 9-9.** *The NXT Rotation sensor's configuration panel*

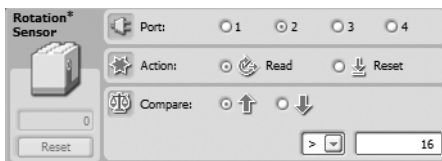
In the Port section, select the motor port where the NXT motor (Rotation sensor) has been connected. The default port for the Rotation sensor is Port A.

In the Action section, you should select the Read option if you want the built-in rotation sensor to monitor the value (count) returned by the sensor. Select the Reset option to set the sensor count back to zero.

In the Compare section, select the motor's spin direction (Forward or Reverse) to monitor. Select Degrees or Rotations from the bottom drop-down menu, and enter a numeric value in the text box. You must also decide whether the sensor will monitor for a value less than or greater than the configured value by selecting the option in the other drop-down menu (using the Yes/No data plug, you can test for a True or False logic response based on the settings you have configured in the panel).

## RIS Rotation Sensor

The RIS Rotation sensor's configuration panel is shown in Figure 9-10.



**Figure 9-10.** *The RIS Rotation sensor's configuration panel*

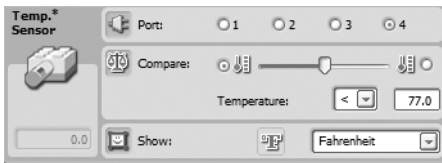
In the Port section, select the port where the RIS Rotation sensor has been connected. The default port for the Rotation sensor is Port 2.

In the Action section, you should select the Read option if you want the built-in rotation sensor to monitor the value (count) returned by the sensor. Select the Reset option to set the sensor count back to zero.

In the Compare section, select the motor's spin direction (Forward or Reverse) to monitor. Enter a numeric value in the text box for the number of ticks you wish to monitor; there are 16 ticks in 1 rotation. You must also decide whether the sensor will monitor for a value less than or greater than the configured value by selecting the option in the other drop-down menu.

## RIS Temperature Sensor

The RIS Temperature sensor's configuration panel is shown in Figure 9-11.



**Figure 9-11.** *The RIS Temperature sensor's configuration panel*

In the Port section, select the port where the RIS Temperature sensor has been connected. The default port for the Temperature sensor is Port 4.

In the Compare section, you must enter a numeric Temperature value in the text box or use the drag bar to set a value. The value for Celsius must be in the range of  $-20$  (negative 20) to 70. The value for Fahrenheit must be in the range of  $-4$  (negative 4) to 158. Select the less than option ( $<$ ) or the greater than option ( $>$ ) from the drop-down menu, *or* click on the left or right radio button to configure this option.

In the Show section, select either Fahrenheit or Celsius from the drop-down menu. The value ranges the sensor can monitor are determined by this selection.

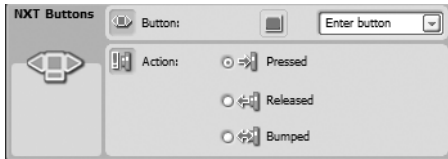
## Other Input Types

There are three other methods that can be used to provide feedback to your NXT robots. These include the NXT buttons, the NXT timers, and a message received using Bluetooth technology by the NXT Brick using a RECEIVE MESSAGE programming block. I'll cover the RECEIVE MESSAGE block (and its partner, the SEND MESSAGE block) in Chapter 25, so let me close this chapter with details on the NXT buttons and timers.

Your NXT Brick has four buttons on its front: Left, Right, Enter (orange), and Cancel. The Cancel button cannot be used as input; its function is simply to cancel a running program, or if you are navigating around the on-screen LCD tools, it can be used to move back to a previous screen. That leaves the Left, Right, and Enter buttons—any program you create can use these three buttons to provide input.

Figure 9-12 shows the configuration panel for the NXT BUTTONS block.



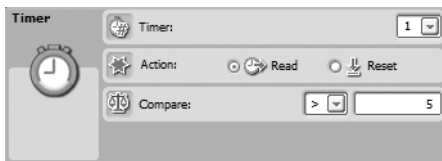


**Figure 9-12.** *The NXT buttons are configured here.*

In the Button section, select the button (Enter, Left, or Right) from the drop-down menu. The default selection is the Enter button.

In the Action section, select Pressed, Released, or Bumped. These work just like the Touch sensor, by the way.

The NXT Brick also has three built-in Timers. These timers begin counting the moment you press the Enter button to begin running a program. Figure 9-13 shows the TIMER block configuration panel.



**Figure 9-13.** *The NXT Timer configuration panel*

In the Timer section, select 1, 2, or 3 from the drop-down menu to select the timer you wish to use. The default selection is Timer 1.

In the Action section, select the Read option to obtain the current value of the timer. If you select Reset, the timer will be reset to zero when this block is executed.

The Compare section allows you to specify a value (in seconds) as a trigger that can be tested. Select either the greater than option or the less than option from the drop-down menu, and the TIMER block can now be used to provide a True/False Logic data type response. For example, if you configure the TIMER block for greater than 10 seconds for Timer 1, the Logic data plug will provide a False response until Timer 1 exceeds 10 seconds. After that, the data plug will provide a True response.

## Using the Blocks

This chapter only showed you how to configure these blocks using the configuration panel. The real power comes with using these in conjunction with other blocks such as the LOOP, WAIT, and SWITCH blocks.

Chapter 11 will show you how to use the LOOP block to repeat certain tasks. When using a LOOP block, you can have the block *loop* forever, that is, performing the tasks over and over again until you cancel the program. Your other option is to configure the LOOP block to stop looping when a certain condition is met, for example, if the Light sensor detects a light level below 20, the Sound sensor detects a noise over level 80, or maybe a motor spins more than 20 degrees (Rotation sensor).

It's the same with the WAIT and SWITCH blocks. Both of these blocks will allow you to use what you've learned in this chapter to give your robots decision-making power.

So, read on—the WAIT block is up next in Chapter 10.