CHAPTER 3

Hello World!

There is a tradition in the world of programming for the first program you write to display the words "Hello World!" on the screen. In keeping with tradition, I'm going to show you how to create a simple version of this program for SPOT. This will allow me to demonstrate one of the simplest blocks included with the NXT software—the DISPLAY block. Once we've taken care of tradition, I'll show you the rest of the DISPLAY block's features.

The DISPLAY Block

In Chapter 2, I explained to you the concept of pseudo-code. Let me now give some pseudo-code to SPOT: SPOT, I'd like you to display the words "Hello World!" on your LCD screen.

Pseudo-code doesn't get much simpler than this. All I want SPOT to do for now is put the words "Hello World!" on his LCD screen. To convert this pseudo-code to an NXT-G program, I'm going to use the DISPLAY block.

Let's start by opening up the NXT software and entering **HelloWorld** in the Start New Program text box (see Figure 3-1). Click the Go button, and the HelloWorld program is open and ready.

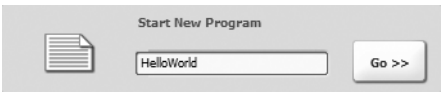

Figure 3-1. *Start a new program called HelloWorld*

Figure 3-2 shows the new program opened (see the tab called HelloWorld in the upper left corner?) and ready for you to start dropping NXT-G blocks. The word "Start" appears on the work space beam, telling you where your first programming block will be placed.

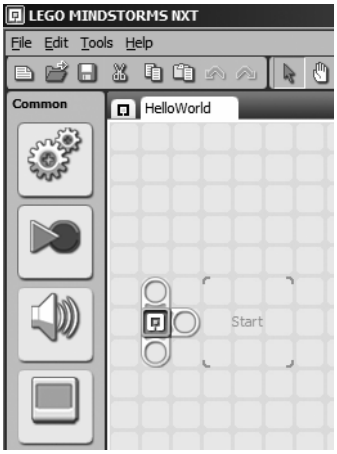

Figure 3-2. *The HelloWorld program is open and ready for the DISPLAY block.*

Are you ready for this? I want you to click the `DISPLAY` block on the Common Palette and hold down the mouse button. Drag and drop the block on the beam where it says "Start" (see Figure 3-3).

Figure 3-3. *The DISPLAY block*

Anytime you drop a block on the work space, the block's configuration panel will be displayed in the lower left corner of the screen. The configuration panel is where you will be doing most of the programming work for your bots. The configuration panel is similar to a car's dashboard. In a car, you can tune to a specific radio station, turn on the windshield wipers, and even find out the car's speed from the speedometer. The configuration panel allows you to turn on and off certain things as well as receive feedback. For example, you can use the `DISPLAY` block's configuration panel to choose what to put on the LCD screen, but the `DISPLAY` block can also receive input from items outside your control, just like your car can display a warning light on the dashboard when you need to check the oil or fill up on gasoline. Figure 3-4 shows the configuration panel for the `DISPLAY` block you just dropped on to the work space. To see the configuration panel for a block, simply click the block using the Pointer tool, and an aqua-colored band will appear around the block that is selected.

■ **Note** If you select multiple blocks, no configuration panel will be displayed.

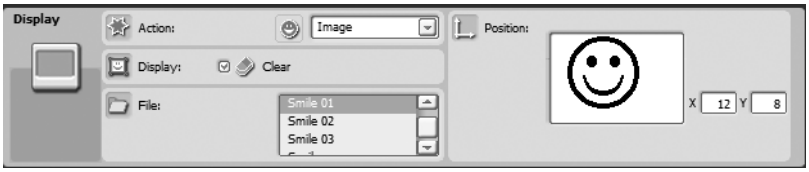

Figure 3-4. *Configuration panel for the DISPLAY block*

Now, to have SPOT's LCD screen display the words "Hello World!", make sure you've first selected the `DISPLAY` block (click it with the Pointer tool).

As you can see in Figure 3-4, by default, the `DISPLAY` block's Action section has a drop-down menu with the Image option selected (there are four options: Image, Text, Drawing, or Reset). Click the drop-down menu, and select Text from the options listed. You will now see a text box with the words "Mindstorms NXT" inside. Change the text to **Hello World!**, and you'll see the same text displayed in the Position section's preview box on the right side of the configuration panel (see Figure 3-5).

Figure 3-5. *The "Hello World!" text is displayed in the preview box.*

Now, using the File menu, select Save, and use the Browse button to choose a location to save the file on your computer. Click the Save button when you are finished. After saving, connect SPOT to your computer, and upload the `HelloWorld` program.

■ **Note** For the remaining chapters in the book, you'll need to remember to save your programs. I won't keep bugging you with instructions to save your programs and upload them to your robots. OK? Just get in a habit of saving often.

After the program is uploaded, select it from the File section, and press the orange button on the Brick to run the program.

Did you see it? The program probably ran so quickly that you didn't even see the text displayed! Why does this happen? Well, when the program runs it is supposed to write "Hello World!" to the LCD screen and then end. And that's exactly what happened—the text displays, and the program ends. This happens *so fast* that you don't even get to see the text displayed. The good news is that this is very easy to fix, so let me update my pseudo-code before I continue: SPOT, I'd like you to display the words "Hello World" on your LCD screen for 10 seconds.

There are numerous ways to keep the text on the screen until you have a chance to read it, but I'm only going to show you one method in this chapter. You'll discover other methods as you continue with the book.

To fix this problem, I'd like you to move the mouse pointer over the WAIT block icon on the Common Palette. When you do this, a collection of WAIT blocks will appear on a fly-out menu, as shown in Figure 3-6.

Figure 3-6. *Adding a WAIT block will allow you to view the "Hello World!" text.*

The WAIT block does exactly what it says—it waits. As you can see in Figure 3-6, there are many different types of WAIT blocks, but the one I'm interested in right now is the WAIT block that allows me to specify how many seconds to wait. That would be the TIME WAIT block (the block that is circled on the fly-out menu in Figure 3-6).

Select the icon for the TIME WAIT block, and place it immediately after the DISPLAY block. In the configuration panel for the TIME WAIT block, select a reasonable time for the text to be displayed—my pseudo-code asked SPOT to wait for 10 seconds, so that is what I will configure (see Figure 3-7).

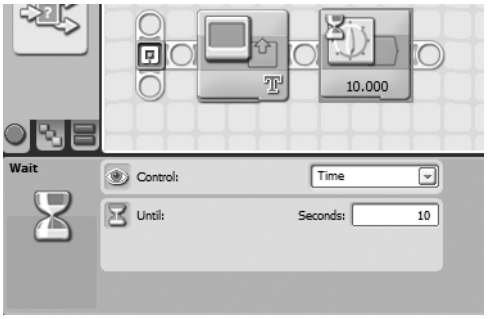

Figure 3-7. *Configure the TIME WAIT block for 10 seconds.*

Now run the program. You should see the text "Hello World!" display on the LCD screen for 10 seconds before the program ends.

You're probably thinking what I'm thinking, "That wasn't that exciting." But remember this: with programming, you have to start somewhere. And, in a few simple steps, you've now figured out how to add text to the LCD screen for any robot you build in the future. Now, let's look at some of the other things you can do with the DISPLAY block.

■ **Note** Every program block has its own unique settings, so each configuration panel is different. Sections for the DISPLAY block, for example, include Action, Display, File, Text, Type, and Position. Sometimes, a section will not be visible until other options are selected. Don't let this worry you; I'll be going over all the unique items for each block throughout the book.

Here are details for the DISPLAY block's sections:

The Display section only has one configurable item—a Clear checkbox. When this box is checked (and the block is executed in a program), the Brick's LCD screen will be cleared of any text or images that are currently on the screen. After the screen is cleared, the DISPLAY block will put what you configured on the screen.

If you leave the box unchecked, any text or graphics you configure the DISPLAY block to put on the LCD will display on the screen along with whatever is currently displayed, instead of replacing it.

This is useful when you want text to appear on multiple lines; you can use multiple DISPLAY blocks to keep adding text to make sentences and even paragraphs. And, without clearing the screen, you can create your own simple images using the Drawing option in the Action section, which I'll explain next.

With the Action section, you have four options in the drop-down menu: Image, Text, Drawing, and Reset. By default, the drop-down menu is set to Image for a new DISPLAY block placed on the work space.

When you select Image in the drop-down menu, the File section is displayed; this section gives you access to a collection of small built-in pictures that can be displayed on the LCD screen (see Figure 3-8).

Figure 3-8. *Choose an image from the File section to place on the LCD.*

By clicking and holding the image in the preview pane on the right side of the configuration panel, you can drag the image around the small pane and place it wherever you wish. You can also use the X and Y coordinates to type in numbers that will place the image at a location of your choosing (see this book's appendix for a brief explanation of the X/Y coordinate system if you're unfamiliar with it). You can use this ability to move the image around the preview pane to place multiple images (which require using additional DISPLAY blocks) on the LCD screen.

The next option in the Action drop-down menu is Text. You've already used this in the previous "Hello World!" example, but I'd like to add that you also have the ability to drag the text around the preview pane and place it in a particular location. The LCD screen is broken into eight horizontal lines; you can use the small drop-down box next to the Preview pane to choose a number between 1 and 8 to define the line where text is placed.

The third option in the Action drop-down menu is Drawing (see Figure 3-9). You can choose to draw a line or a circle or to place a single point on the LCD screen, so your artistic talents will be somewhat limited. To create a detailed drawing, you would have to place dozens or more DISPLAY blocks one after the other, and the combination of lines, circles, and points would create the image. But the Drawing options can be useful to draw boxes around other text on the screen, so keep that in mind.

Figure 3-9. *The Drawing option can be used to place points, lines, and circles.*

To use the Drawing tool, select Point, Line, or Circle from the Type section (this section only appears if you have selected Drawing in the Action drop-down menu). For the point, you can drag it around the Preview pane and place it anywhere. You can also use the X and Y coordinates to place the point more accurately.

If you choose the Line, the end point of the Line is at position 10,10 (in the lower left corner). Click anywhere in the Preview pane to draw a line from that point to the place where you clicked. You can change the end point (10,10) by entering new coordinates in the X and Y boxes. You can also type in X and Y coordinates for the other end of the line for more accurate control over it.

Finally, for the circle, you have the option of changing the radius of the circle by typing the number in the Radius text box. Drag the circle around the Preview pane to place it properly.

The final option in the Action drop-down menu, Reset, is useful when you would like to clear the LCD screen of any items. The default NXT screen (which shows the name of the program currently running) will appear on the LCD screen.

Before closing out this chapter on the DISPLAY block, I want to cover one additional item briefly: data hubs (this topic will be covered in more detail in Chapter 7).

Data Hubs

Most programming blocks come with what's called a "data hub." Take a look at Figure 3-10.

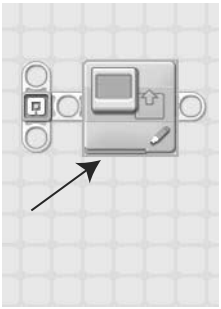

Figure 3-10. *Click on the DISPLAY block here, and the data hub will drop down.*

If you click the bottom-left edge of a block, this section will drop down and reveal the data hub (see Figure 3-11). Click the section again, and the data hub will close. It might take some practice to find the correct place to click, so try it a few times until you get used to opening and closing the data hub.

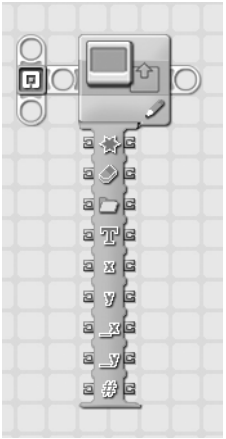

Figure 3-11. *The DISPLAY block's data hub can be used for more advanced programming.*

What is this data hub? The data hub allows you to draw *data wires* from one block to another using *data plugs*. Data wires and plugs will be covered in much more detail in Chapter 7, but for now, all you really need to know is that wires can connect blocks to share data. Data plugs are places on the block where you will connect wires. So there will be a data plug on one block with a wire going out and another data plug on a different block with a wire coming in. Data wires can carry information such as text, numbers, and other values. Remember all those items you could configure in the `DISPLAY` block? Well, items such as the text displayed or the radius of a circle can all be configured without using the configuration panel. Instead, you can draw data wires from one block's plugs into plugs on another block. I'll cover this topic in more detail later in the book (in Chapter 7), but right now, I just want you to take a look at Figure 3-12, so I can give you a preview of what's to come.

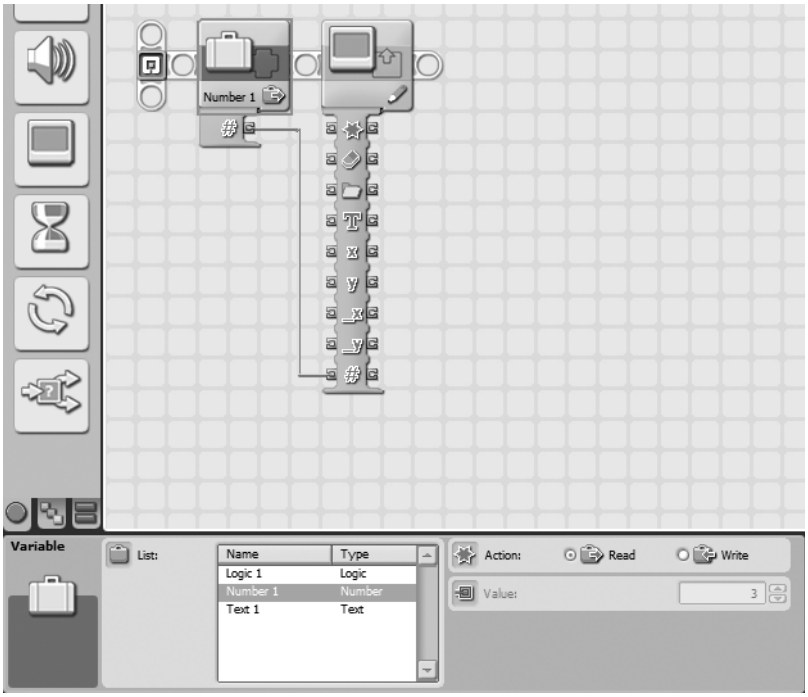

Figure 3-12. *An example of one DISPLAY block plug providing data to another block.*

In Figure 3-12, I've placed a `VARIABLE` block in front of the `DISPLAY` block. I cover the `VARIABLE` block in Chapter 18, but for now, all you really need to know is that this type of block can hold text, a number, or what's called a logic value (either True or False). In this example, I've configured the `VARIABLE` block to hold a number: 3.

Now, here's where it gets fun. Remember that when you draw a circle you can specify the radius of the circle in the configuration panel? Well, this `VARIABLE` block has only one plug in its data hub. For this block, it's holding the value of 3. I click that plug and draw a line into the last plug of the `DISPLAY` block's data hub. That last plug corresponds to the radius of a drawn circle (hover your mouse pointer over a plug, and it will tell you what it is). When I drag the wire *out* of the `VARIABLE` block plug and *into* the `DISPLAY` block plug, the line becomes solid yellow, and I know I've correctly configured the `DISPLAY` block.

■ **Note** Plugs on the left side of a block's data hub are known as *input plugs*. Plugs on the right side of the block's data hub are called *output plugs*.

If the line is dotted, it tells me I've incorrectly connected two plugs. This can happen for many reasons. For example, if I had put text into the VARIABLE block and dragged the wire into the DISPLAY block's Radius plug, I would get a dotted line. This happens because the Radius plug expects a number value to be coming out of the VARIABLE block, not text.

As I mentioned earlier, I'll cover data plugs in more detail in Chapter 7. Before you begin using these data wires for more advanced programming, however, you need to understand the basics of the programming blocks.

Let's continue learning about blocks in Chapter 4 with the MOVE block.