Staying Alive

Another short chapter? Of course! You can learn something and then get back to playing around with your NXT robots. But you won't get to play much if your robots keep falling asleep. This chapter will show you how to keep your robots working, even if the Sleep option on the Brick has been set to a short period of time like 2 minutes or 5 minutes.

The KEEP ALIVE Block

Your NXT Brick has a built-in feature that automatically turns off your robot after a certain amount of time has passed with no activity. This Sleep timer is configured on the Brick, and you can choose for the Brick to shut down after 2, 5, 10, 30, or 60 minutes of inactivity, or you can choose Never.

The Sleep timer is a useful feature, and you should definitely set it to a reasonable time. It can help save battery power if you accidentally leave the Brick turned on. If a program finishes executing, for example, and the robot stops moving while you're away for a little bit, the Sleep timer can make sure your robot turns itself off.

The KEEP ALIVE block and its configuration panel are shown in Figure 21-1.

Figure 21-1. *The KEEP ALIVE block and its configuration panel*

Notice that the configuration panel is blank! There are no settings to configure with the KEEP ALIVE block.

Here's how it works: During the execution of your program, if at any time a KEEP ALIVE block is executed, the Sleep timer resets to its initial starting time. So, if you configured your Brick for a 5-minute Sleep timer and 1 minute into the program a KEEP ALIVE block is encountered, the Sleep timer will reset to 5 minutes.

Now, to make this useful, you'll probably need to place the KEEP ALIVE block in a location where it will frequently be executed. The best location is a LOOP block that will occasionally (or continually) run the blocks inside it, including the KEEP ALIVE block. This is one method for continually resetting the Sleep timer.

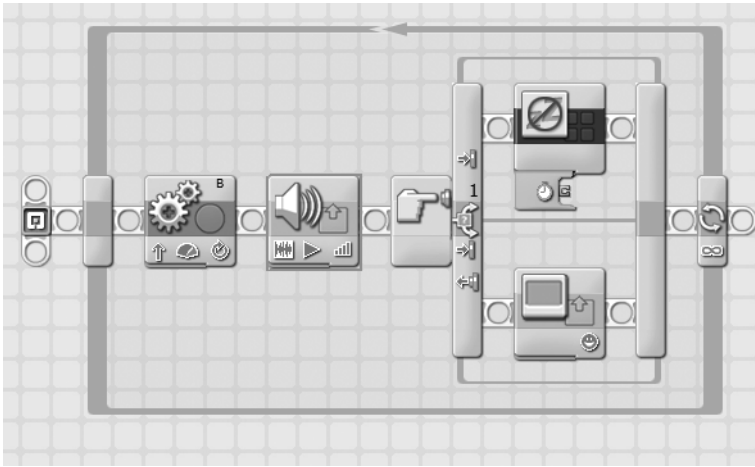An example is shown in Figure 21-2.

Figure 21-2. *Put a KEEP ALIVE inside a LOOP, and you can repeatedly reset the Sleep timer.*

In this simple program, the bot will spin around a few times, say "Hello", and then do it again and again and again . . . if the Touch sensor button is ever pressed, the SWITCH block will execute the KEEP ALIVE block, which resets the Sleep timer. If the button is never pressed, the bot will eventually turn off when the default Sleep timer value has been reached.

This is important: always check the Sleep timer setting on your Brick before running a program. Once the time has expired, the program will stop. This happens frequently when the Sleep timer is set to 2 minutes or 5 minutes, so double-check this before running a program, and be certain to set the Sleep timer to a setting that is longer than you expect the program to run.

To close out this chapter, the last item I want to mention about the KEEP ALIVE block is that it does have an output data plug in its data hub. This data plug will *only* supply a Number data type, and that number will always be the Sleep timer default value in milliseconds (1,000 milliseconds = 1 second). You might not find this data plug very useful, but it could be used as input to a COMPARE block that looks to see which is greater—a value from one of the Brick's three internal timers or the default Sleep timer value. Depending on the condition of the COMPARE block, your bot might perform some final action before letting the robot shut down using a STOP block *or*, as mentioned earlier, it might execute a KEEP ALIVE block to reset the Sleep timer value.

Well, that's it for the KEEP ALIVE block. Up next in Chapter 22 is the FILE ACCESS block. You'll learn how to program your bot to store data in files that can be accessed at any time, now or later.