

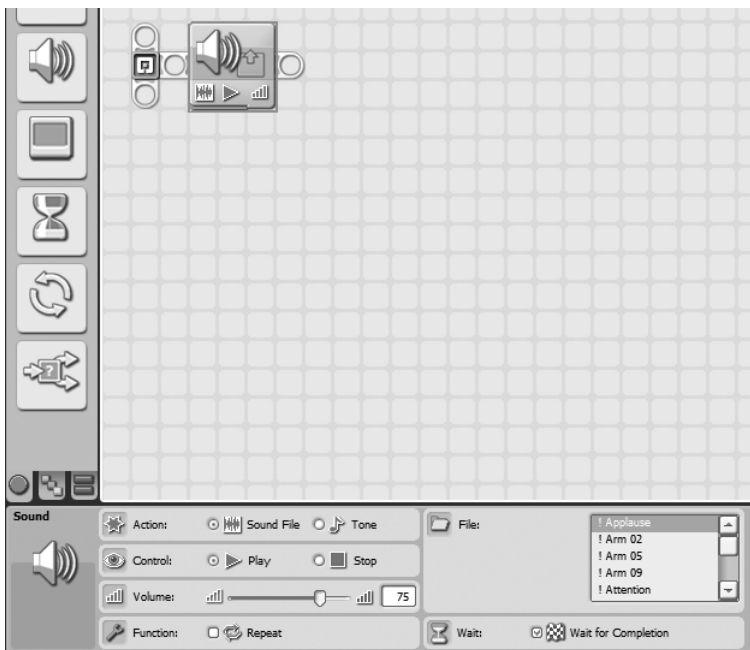


# Make Some Noise!

**S**ound can be used to give a bot more personality. Think about how boring R2-D2 would be without all the chirps and whistles. Well, you can give your bot plenty to say by using the **SOUND** block. Sound isn't limited to just giving your bots character, though. Sounds can be useful as a way to judge your bot's progress through a maze, for example, with the bot programmed to issue specific sounds when it makes a left or encounters a dead end. Read on for all the details on the **SOUND** block.

## The **SOUND** Block

When you drop in the **SOUND** block on the beam, you are given access to the configuration panel shown in Figure 6-1.



**Figure 6-1.** *The **SOUND** block's configuration panel*

## Action Settings

The first section I want to cover is the Action section. You have two options: Sound File or Tone.

### Sound File

Select the Sound File option, and take a look at Figure 6-2. See the section called File? Clicking the Sound File option opens the File section; the File section contains a large collection of prerecorded sounds that your bot can play through the Brick's speaker. Click one of the sound files, and if your computer has speakers, you will hear the sound file play.



**Figure 6-2.** The Sound File option allows you to select a sound from the File section.

### Tone

Now, select the other option, Tone. Notice that the File section changes to a section called Note (see Figure 6-3).



**Figure 6-3.** The Tone option allows you to specify tones from the Note section.

## Note Settings

The Note section provides you with a few options. The easiest option is to simply click one of the piano keys. You will hear the tone played if your computer has speakers attached. Notice that when you click on a key, the note you click is displayed in the text box above the keys in the form of a letter: A, B, C, D, E, F, or G (with sharps for the black keys).

The other option available in the Note section is the ability to specify how long the note will play. Type a number in the text box for the number of seconds to play the note.

That covers the Action section; the remaining sections for the SOUND block (Control, Volume, Function, and Wait) are the same whether you choose the Sound File or Tone option. Now, let me explain each of these remaining sections.

## Control Settings

The second section on the SOUND block is Control. This section has two options: Play and Stop (see Figure 6-4).



**Figure 6-4.** *The Control section of the SOUND block*

The Play option is simple. Select it, and any sound file or tone you selected in the Action section will play. Not too difficult, right?

The Stop option requires a little more explanation. To do this, I need to jump ahead to the Function section (see Figure 6-5).



**Figure 6-5.** *The Function section of the SOUND block*

## Function Settings

The Function section only has one option: Repeat. If the box is checked, the Sound File or Tone will continue to play until your program ends *or* until another SOUND block is reached with the Stop option selected in the Control section. If the box is unchecked, the sound file or tone will only play *one* time.

So, you can see that the Stop option is useful *only* when you have another SOUND block that is continually playing.

## Volume Settings

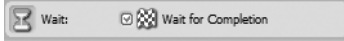
OK, now for the Volume section. You can see in Figure 6-6 that the Volume control can be changed either by using the slider or by typing a value (0 to 100) in the text box. You will have to experiment with the Volume control to determine what works best for your robots, but keep in mind that loud sounds will use up more battery power than sounds played at a lower volume.



**Figure 6-6.** *The Volume section of the SOUND block*

## Wait Settings

The last section in the SOUND block is the Wait section (see Figure 6-7). When you have selected a sound file or tone to play and the Repeat box (in the Function section) is not checked, the Wait for Completion checkbox is available.



**Figure 6-7.** *The Wait section of the SOUND block*

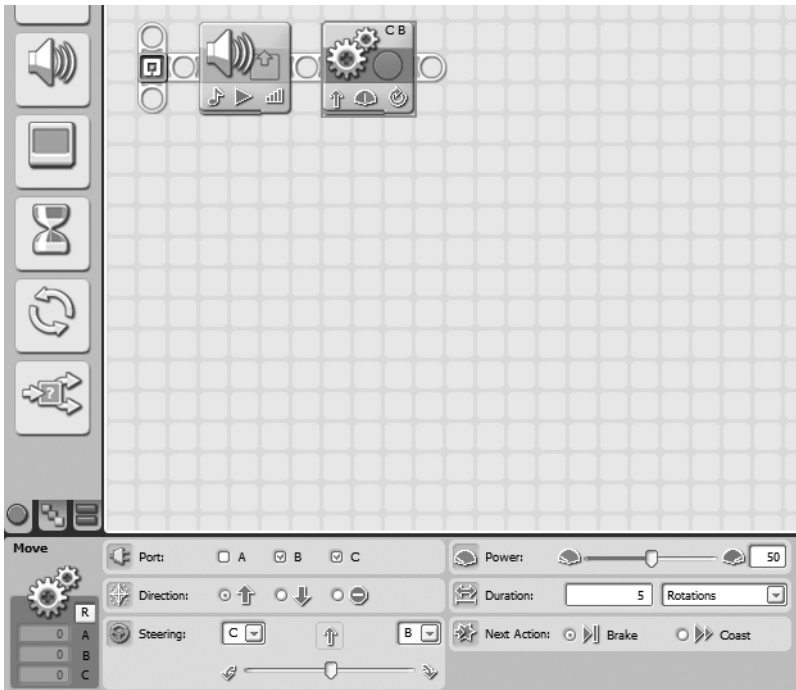
If you place a check in the Wait for Completion box, the Sound File or Tone you choose will play completely before any further programming blocks are executed. Let me give you an example using pseudo-code: SPOT, play me a C note for 10 seconds and then move forward 5 rotations.

Now, here's how I will convert the pseudo-code into a NXT-G program. First, I drop in a SOUND block and configure it to play a C note for 10 seconds (see Figure 6-8). I'm going to leave the Wait for Completion box unchecked and set the Volume to 75.



**Figure 6-8.** *SPOT will first play a C note for 10 seconds.*

Next, I'm going to drop in a MOVE block (see Figure 6-9). I'll configure this MOVE block to spin motor B and motor C FORWARD for 5 rotations and then Brake. I'm also going to set Power at 50.



**Figure 6-9.** *SPOT will then MOVE forward 5 rotations.*

Next, I save the program, upload it to SPOT, and run it. Go ahead and create and run this same program on your bot. What happens?

Well, I pressed the Run button and the C note started to play. But before the C note stopped, motors B and C started spinning. What happened?

Go back to Figure 6-8, and notice that the Wait For Completion box is unchecked. This means that when the SOUND block starts playing, the program will continue to the MOVE block that spins the motors. But that's not what I wanted SPOT to do. Take a look again at the pseudo-code: SPOT, play a C note for 10 seconds and *then* move forward 5 rotations.

I wanted SPOT to play the C note for 10 seconds before moving forward. To do this, I simply need to go back to my SOUND block and check the Wait for Completion box. This will tell SPOT to wait until the SOUND block is finished (10 seconds) before continuing with the program. So, I make this change (see Figure 6-10) and run the program again.



**Figure 6-10.** Making one change to the SOUND block will fix the problem.

This time, SPOT does exactly what I wanted him to do. He plays the C note for 10 seconds, and when the SOUND block is finished, motors B and C spin, and SPOT moves forward 5 rotations. Perfect!

Now you can add sound files and tones to your robots to give them more personality. But before we finish this chapter, let me also tell you that sounds can be used when testing your robots. For example, in a complex program you can drop in a SOUND block to let you know when the robot has reached a certain portion of the program. Let's say you want to know when your robot has reached the part of a program where it has to decide between turning left and turning right. You could place a SOUND block directly in front of the MOVE blocks and program a sound to play when turning left and a different sound for turning right. After you've tested and verified the program is working properly, you can remove the SOUND blocks and run the program normally. In this example, SOUND blocks give you an idea of where a robot's program is currently executing by giving you an audible alert, which is very useful for troubleshooting and testing.

Now, before we investigate any new blocks, I want to go over the different ways your robots receive input—this includes motors, sensors, timers, and the buttons on the Brick. Chapter 7 covers all of this, so read on!