Speed Based on Volume Values & Assignment (Part 1)

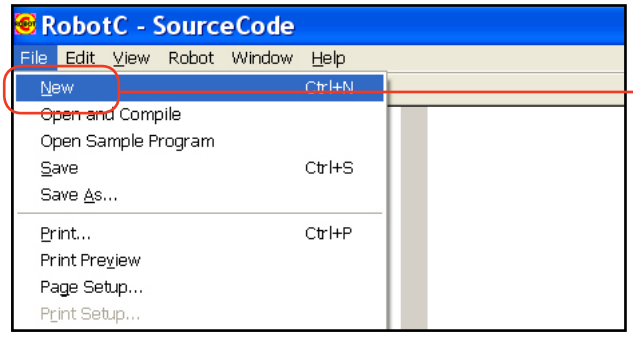
The Sound Sensor is the last of the standard NXT sensors. In essence it's a kind of microphone which senses amplitude (how loud or soft a sound is), but not anything else about it. The Sound Sensor, like the Light Sensor, reports values from 0-100 which do not correspond to any specific standard scale.

Sound Sensor
The Sound Sensor has an orange foam pad which resembles a microphone

Speed Based on Volume Values and Assignment (Part 1) (cont.)

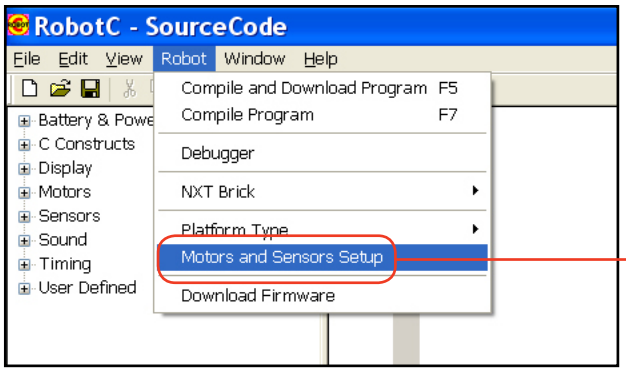
In this lesson you will learn how to use the Sound Sensor to manipulate your robot's motors

1. Start by opening a new program.



1. Create new program
Select File > New to create a blank new program.

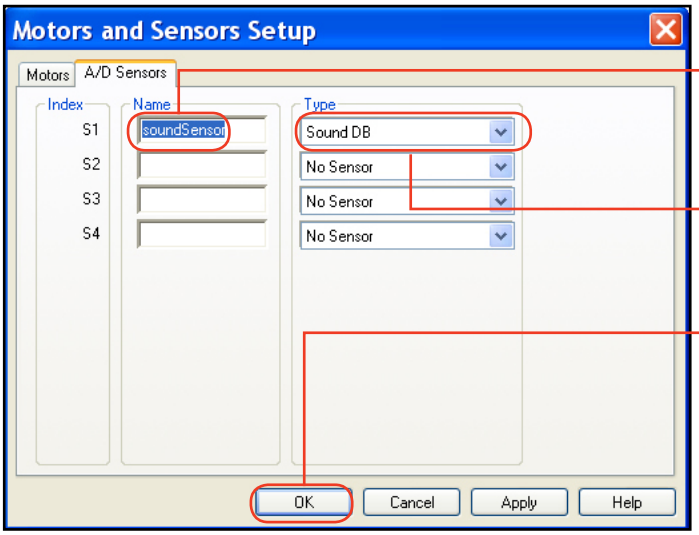
2. Open the Motors and Sensors Setup menu to configure the Sound Sensor.



2. Open "Motors and Sensors Setup"
Select Robot > Motors and Sensors Setup to open the Motors and Sensors Setup menu.

Speed Based on Volume Values and Assignment (Part 1) (cont.)

3. Configure the sensor on port 1 to be a "SoundDB" sensor named "soundSensor".

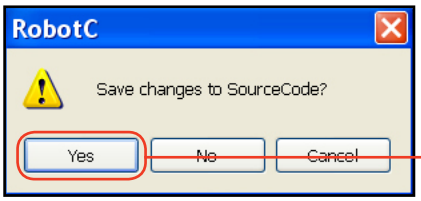


3a. Name the sensor
Name the Sound Sensor on port S1 "soundSensor".

3b. Set Sensor Type
Identify the Sensor Type as a "Sound DB" sensor.

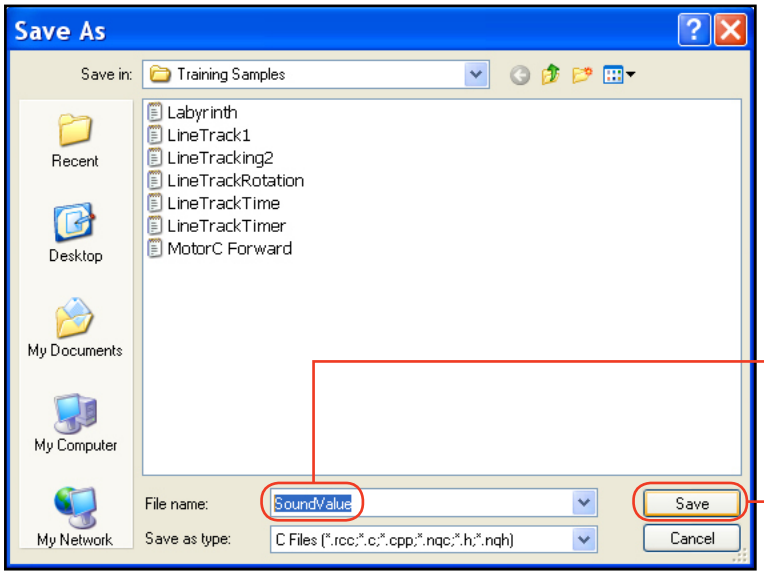
3c. Click OK
Click the "OK" button to save your changes.

4. You will be prompted to save the changes you have just made. Press Yes to save.



4. Select "Yes"
Save your program when prompted.

5. Save this program as "SoundValue".



5a. Name the program
Give this program the name "SoundValue".

5b. Save the program
Press Save to save the program with the new name.

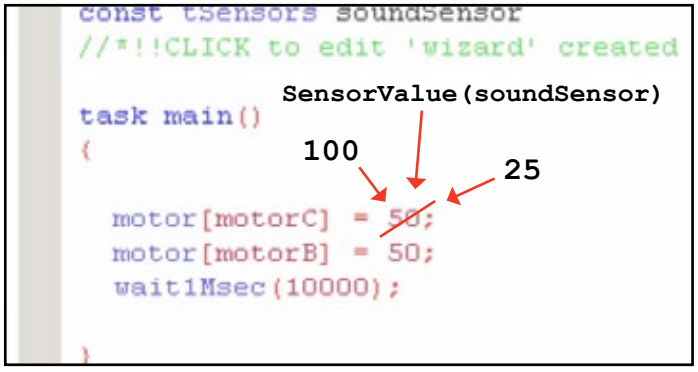
Speed Based on Volume Values and Assignment (Part 1) (cont.)

The Sound Sensor is now configured. Now, start the program by creating a task main() structure, then add a forward movement command for 10 seconds with both motors at 50% power.

```
1  task main()
2  {
3     motor[motorC] = 50;
4     motor[motorB] = 50;
5     wait1Msec(10000);
6  }
```

Checkpoint
Let's analyze what we're telling the robot to do. The basic motor command sets a given motor's power level. In this case, you're setting Motor C and B's power level to 50. 50 is just a number. If you wanted to set the power to 25, you would put 25 here. 100 works too. Really, any number value will do....

The Sound Sensor reading is also a number value. If the Sound Sensor is reading a sound level of 40, SensorValue(soundSensor) is the number value 40! We could simply put **`SensorValue(soundSensor)`** in place of the number we've been using, and the motor power would be set to the Sound Sensor's value! Let's try it.



Speed Based on Volume Values and Assignment (Part 1) (cont.)

6. Motor powers are number values. You can replace any number value with another, like changing a 50 to 75 or 100. SensorValue(soundSensor) is also a number. Replace 50 with the sensor value.

```
1  task main()
2  {
3     motor[motorC] = SensorValue(soundSensor);
4     motor[motorB] = SensorValue(soundSensor);
5     wait1Msec(10000);
6  }
```

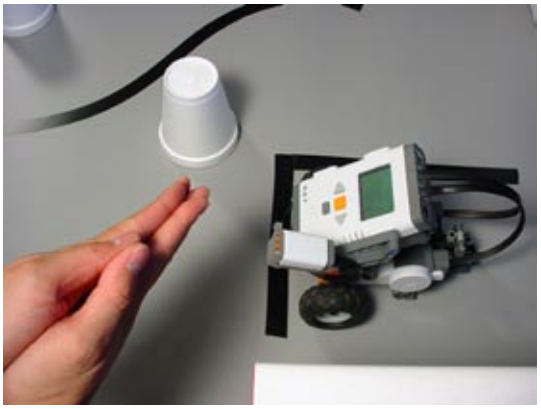
6. Modify the code
Replace the number values of 50,to the value of the Sound Sensor, S1.

Checkpoint. In theory, our program should now work like this:

- The Sound Sensor reads the amount of sound in the environment
- The Sound Sensor sets the motor power to be equal to the sensor's numeric value
- The robot should run at a speed determined by the Sound Sensor reading – fast for loud, and slow for quiet

Speed Based on Volume Values and Assignment (Part 1) (cont.)

7. Save, download, and run your program. Clap your hands to change the sound sensor value.



7a. Make some noise!
Run the program then clap your hands to change the sound sensor value.

7b. Observe the (lack of) reaction
The robot doesn't seem to do anything different...

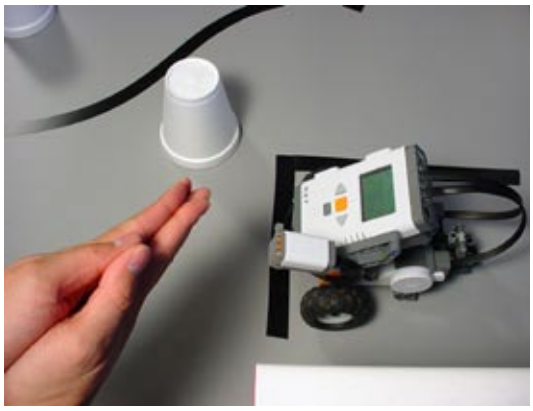
End of Section

The robot's reaction to the level of sound in the environment was pretty disappointing – nothing happened. In the next section, we'll take a look at what's going on, where our understanding went wrong, and how the problem can be fixed.

Speed Based on Volume Values & Assignment (Part 2)

In this lesson you will make the robot's motors use the Sound Sensor's values in real-time.

Try running the robot again, but make the sound **just as** you press the Start button.



Clap and Run
This time, clap (or talk into the Sound Sensor) just as you press the Start buton.

Observe the behavior
The robot moves much faster.

The robot is clearly responding to sound levels, but not at the right time. Remember the line tracking behavior? The wait1Msec command tells the robot to go to sleep for a period of time. Going to sleep means the robot isn't watching the sound sensor or updating motor values! If we want to keep the motor's power level up to date with the sensor, we will need to make sure that the power level command gets run over and over. We'll need to use a while loop and a Timer.

Speed Based on Volume Values and Assignment (Part 2) (cont.)

1. Delete the wait statement, and add a while() loop around the motor behaviors.

```
1  task main()
2  {
3      while()
4      {
5          motor[motorC] = SensorValue(soundSensor);
6          motor[motorB] = SensorValue(soundSensor);
7          wait1Msec(10000);
8      }
9  }
```

1a. Add this code
Place the while loop so that the motor commands go inside its curly braces.

The (condition) is not yet specified.

1b. Delete this line
We don't want the robot "sleeping" when it needs to update motor powers.

---

2. Timers must first be initialized, so add a ClearTimer(T1) just before the loop. Check the timer in our while loop condition, we use timer1[T1] less than 10,000 milliseconds, or 10 seconds.

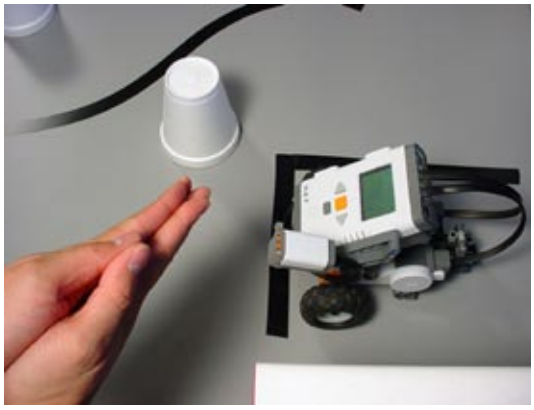
```
1  task main()
2  {
3      ClearTimer(T1);
4      while(time1[T1] < 10000)
5      {
6          motor[motorC] = SensorValue(soundSensor);
7          motor[motorB] = SensorValue(soundSensor);
8      }
9  }
```

2a. Add this code
Timers must be reset before use.

2b. Add this code
The (condition) will now check whether the timer, T1, is less than 10000ms (10 seconds). The loop's {body} will run while this is true, i.e. less than 10 seconds have passed since the reset.

Speed Based on Volume Values and Assignment (Part 2) (cont.)

3. Save, download, and run your program.



Run the program
Run the program and clap your hands repeatedly.

Observe the behavior
The robot moves depending on how much noise it detects!

End of Section

The robot is now checking the sensor repeatedly, and updating the motor power with the new sensor values as quickly as it can, over and over again. As a result, the robot is now responsive to new sound levels in the environment. Rather than just on or off, loud or soft, we've programmed the robot to change the motor power level in direct proportion to the sound level. This is a powerful way to use sensor values. It takes advantage of their numeric nature to link a sensor value with another numeric value, motor power output.

In the next Unit's challenges, you'll have additional opportunities to look even more deeply into the nature of numbers and other data types in ROBOTC. For the immediate future, we think you'll find this Volume Based on Speed behavior helpful on the Obstacle Course. See you on the field!