# Forward Until Dark Light Sensor

*In this lesson, you will learn how the Light Sensor works, and how its feedback compares to the Touch and Ultrasonic Sonar sensors.*

***Touch Sensor***
The Touch Sensor detects physical contact with the orange trigger, and returns a SensorValue of 1 if it is pressed in, or 0 if it is not.

**Detects:** Physical contact

**Feedback:** 0 if unpressed, 1 if pressed

**Typical use:** Bumper

**Sample code:** `while (SensorValue(touchSensor) == 0)` will run the while() loop as long as the touch sensor is not pressed.

---

***Ultrasonic Sensor***
The Ultrasonic (sometimes called Sonar) Sensor sends out pulses of sound and measures the time it takes for the sound waves to bounce off an object and return. Since the speed of sound is known, the sensor calculates the distance based on the time, and reports the distance in centimeters as its SensorValue

**Detects:** Distance to object

**Feedback:** Range to object in centimeters (1-250)

**Typical use:** Obstacle detection and avoidance

**Sample code:** `while (SensorValue(sonarSensor) > 25)` will run the while() loop as long as there is no object detected within 25 cm.

# Forward until Dark Light Sensor (cont.)

And now, let's look at a new sensor.

**Detects:** Reflected + Ambient light

**Feedback:** Brightness (0-100)

**Typical use:** Line detection

**Sample code:** `while (SensorValue(lightSensor) > 40)` will run the while() loop as long as the light sensor value remains brighter than 40.

***Light Sensor***
The Light Sensor (in the normal Active mode) shines a light out in a cone in front of it, and measures how much light comes back to it, from either reflection or ambient sources. See additional explanation below.

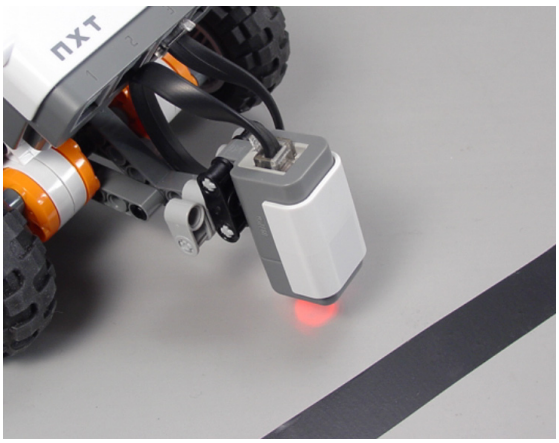
---

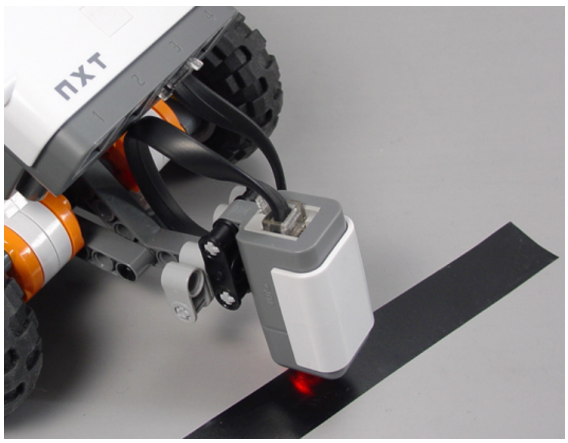
This is the Light Sensor. When turned on, it shines a cone of red light out from the red LED, and measures how much of it comes back into the light detector through the clear lens.

***Red lens***
A cone of red light shines out from the red LED.

***Clear lens***
A light detector measures how much light comes back.

# Forward until Dark Light Sensor (cont.)




The light sensor can detect the basic colors of objects and surfaces by aiming directly at them at close range. Light-colored surfaces, like this bright grey table, reflect a large amount of the light, and produce a high sensor reading. Dark-colored surfaces, like this strip of black electrical tape, reflect very little light, and produce a low sensor reading.

High readings vs. low readings can therefore be used to distinguish light surfaces from dark ones. To make this work for the while() loop, we'll need to use the same technique we used with the Ultrasonic Sensor: set a threshold value to create a "cutoff" point between light and dark.

The sensor gives a light intensity reading of 0-100. But unlike the Ultrasonic Sensor, where the number was in centimeters, the Light Sensor's values are relative only, and do not correspond to any set system of units. In fact, any light source – lamps, sunlight, shadows – and even the height of the light sensor off the table can affect how much light the Light Sensor sees for the same surface. So how can you set a fair cutoff (threshold) between light and dark?

In the next section, you will use the NXT's View Mode to see for yourself what sorts of numbers you get for different surfaces. You will use these real-world readings as reference values for light and for dark. Your readings will give you measured "anchors," that take into account the colors of surfaces, and lighting conditions, and will allow you to make a proper choice of threshold.

# Forward until Dark Thresholds 201

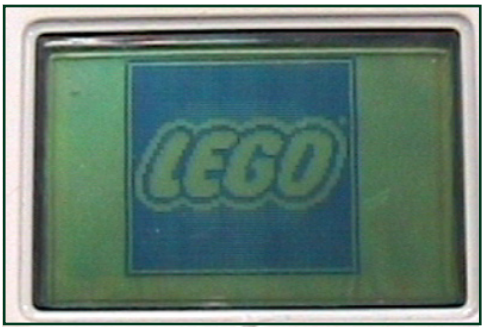
**Reminder! Light sensor readings and other numbers used in this printed guide may not be right for your environment. Your room's lighting and the position of the sun and shadows will cause light sensor readings to vary. Measure often!**

So higher is brighter, and lower is darker, but if you remember from the last time we worked with a large range of values, we set a threshold to separate the two values we care about. Before we can set a threshold for the Light Sensor, we need to know what values mean 'Light" and what values mean "Dark." Let's take some readings to find out.

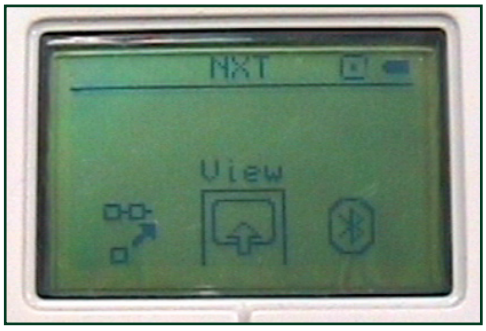
---

*In this lesson, you will learn how the Light Sensor works, and how its feedback compares to the Touch and Ultrasonic Sonar sensors.*

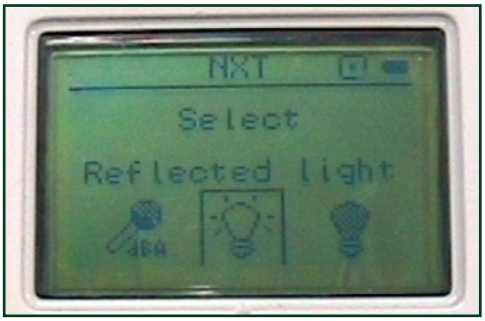
**1.** View the Reflected Light values in View Mode.




***1a. Turn on NXT***
Turn on your NXT if it is not already on.




***1b. Navigate to View Mode***
Use the left and right arrow buttons to find the View option, and press the Orange button to select it.



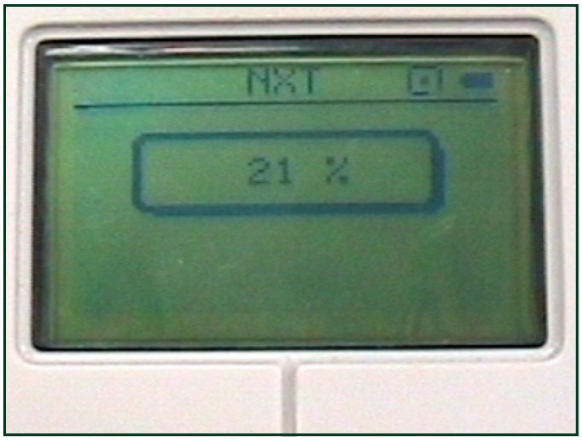

***1c. Select Reflected Light***
Use the left and right arrow buttons to find the Reflected Light option, and press the Orange button to select it.

***Caution! Do not choose "Light Sensor*"!***
Light Sensor* (and all sensors with a * at the end of their names) refers to the old RCX-generation Light Sensor, and will not provide the correct readings for the NXT Light Sensor.


***1d. Select Port 1***
Make sure your Light Sensor is plugged into Port 1 on the NXT. Select Port 1 on screen.

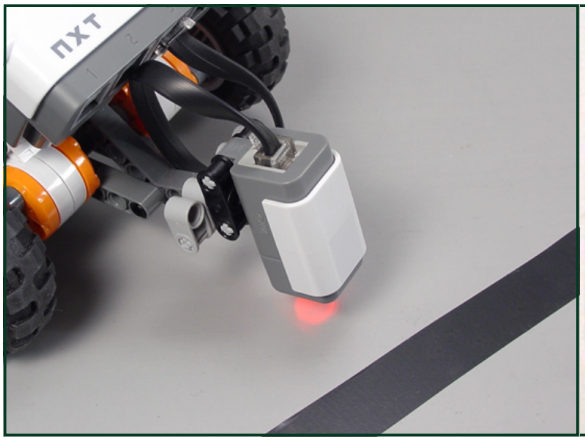
# Forward until Dark Thresholds 201 (cont.)

**Checkpoint**
You are now seeing the sensor's value live, in real time.

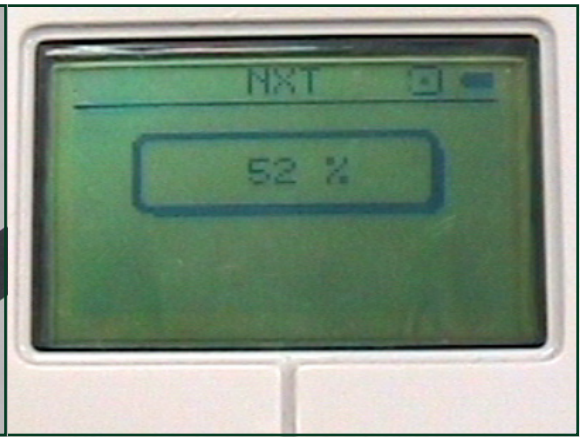



---

**2.** Place the robot so the light sensor is over the light surface, move your hand away (it can cast a shadow and mess up your readings), and record the reading on the screen.



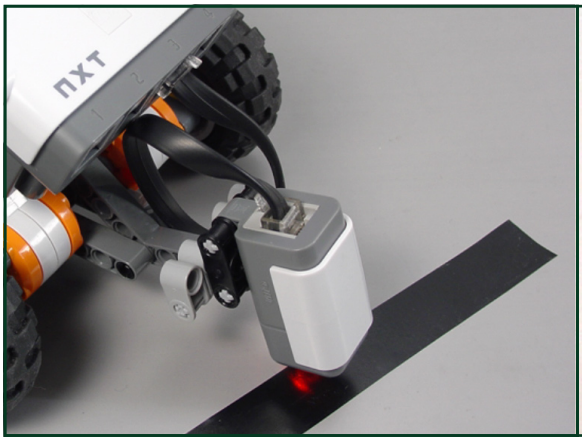

***2a. Place robot over light surface***
Position the robot so that the light sensor shines on a light-colored surface.

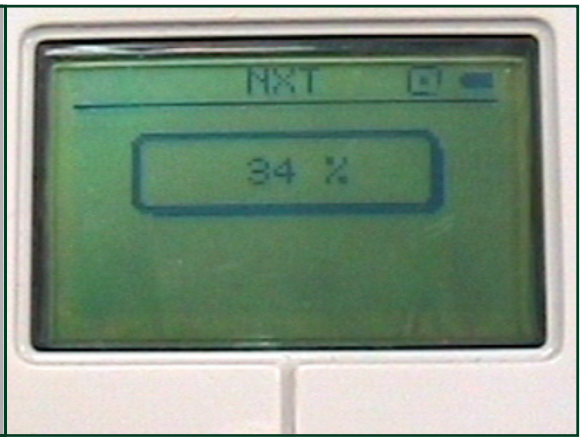



***2b. Record "light" sensor value***
On a separate sheet of paper, write down the Light Sensor value for a "light" surface.

# Forward until Dark Thresholds 201 (cont.)

**3.** Now, place the light sensor over a part of the dark line, and record that reading.




***2a. Place robot over dark surface***
Position the robot so that the light sensor shines on a dark-colored line.

***2b. Record "dark" sensor value***
On a separate sheet of paper, write down the Light Sensor value for a "dark" surface.

---

**4.** A fair place to set the cutoff line is right in the middle between these two values. That would be the average of these two values.

$$\frac{value\ 1 + value\ 2}{2} = Threshold\ Value$$

***4a. Add "light" and "dark" values***
The first step in finding an average is to find the sum of the two values.

***4b. Divide sum by 2***
Since there were two values (light and dark), divide the sum by 2 to find the average.

***4c. Write down Threshold value***
This average is fairly situated, exactly between the other two values. Record this

---

**End of Section**

With the threshold set at the point indicated by the red line, the world of light sensor readings can now be divided into two categories: "light" values above the threshold, and "dark" values below the threshold. This distinction will allow your robot to find the line, by looking for the "dark" surface on the ground.

The threshold you have calculated marks the cutoff line for your lighting conditions. Any sensor values above the threshold value will now be considered light, and any below it will be considered dark.

# Forward until Dark Wait for Dark

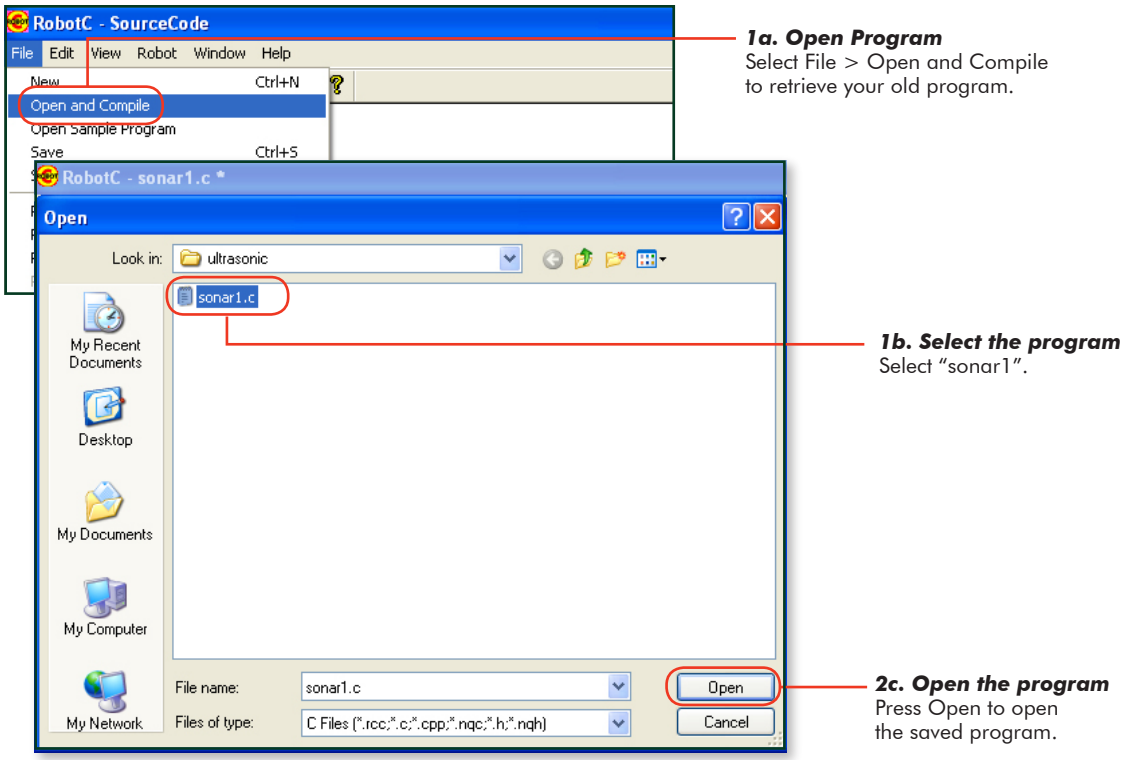
**Reminder! Light sensor readings and other numbers used in this printed guide may not be right for your environment. Your room's lighting and the position of the sun and shadows will cause light sensor readings to vary. Measure often!**

So higher is brighter, and lower is darker, but if you remember from the last time we worked with a large range of values, we set a threshold to separate the two values we care about. Before we can set a threshold for the Light Sensor, we need to know what values mean 'Light" and what values mean "Dark." Let's take some readings to find out.

---

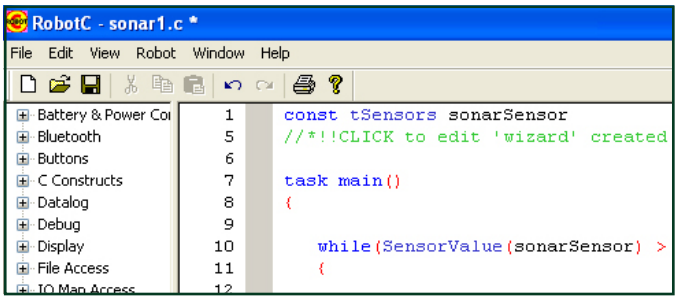
*In this lesson, you will use the Light Sensor and the Threshold you calculated in the previous section to adapt your Ultrasonic Wall Detector program to detect a dark line instead.*

**1.** Open "sonar1", the Ultrasonic Sensor program from the Wall Detection (Ultrasonic) lesson.



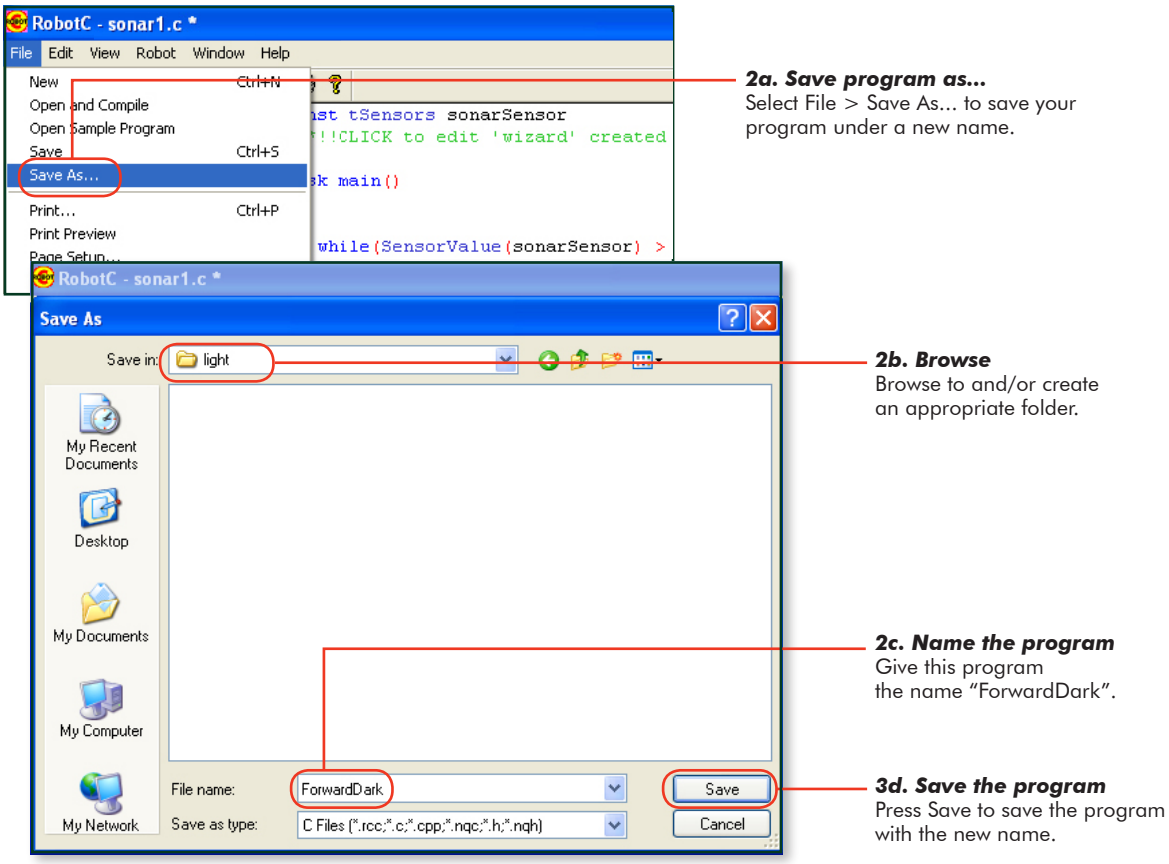
# Forward until Dark Wait for Dark (cont.)

**Checkpoint.** The program should look like the one below.



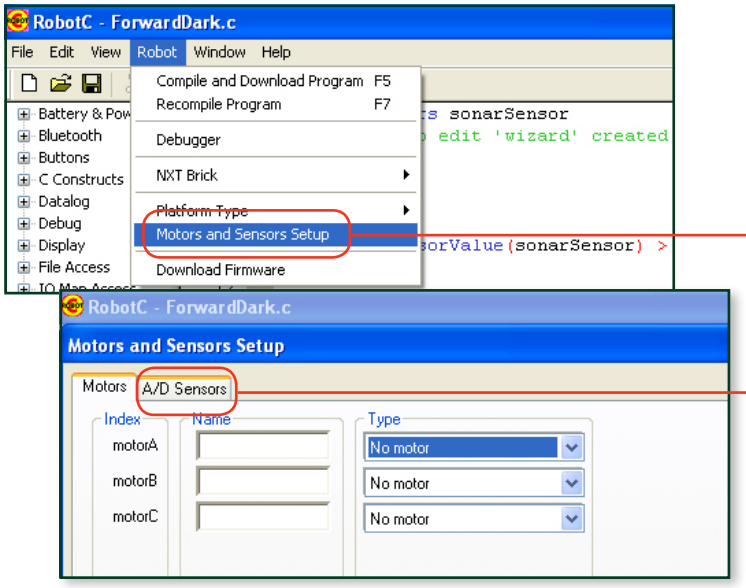

---

**2.** Because we're going to be changing the program, save it under the new name "ForwardDark".



# Forward until Dark Wait for Dark (cont.)

**3.** Open the Motors and Sensors Setup menu, and go to the Sensors tab.

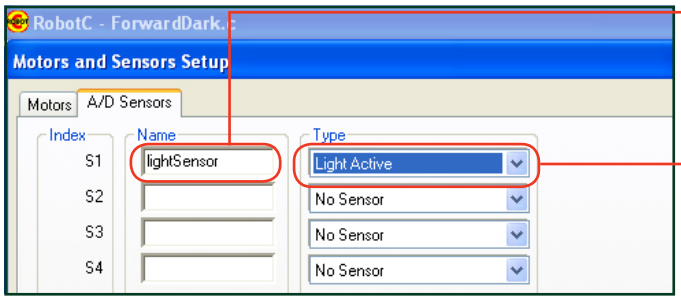



***3a. Open "Motors and Sensors Setup"***
Select Robot > Motors and Sensors Setup to open the Motors and Sensors Setup menu.

***3b. Select the A/D Sensors tab***
Click the "A/D Sensors tab" on the Motors and Sensors Setup menu.

---

**4.** Use the Motors and Sensors Setup interface to name the S1 sensor "sonarSensor", then select "SONAR 9V" as its type.




***4a. Name sensor "lightSensor"***
Enter the name "lightSensor" in the S1 name box.

***4b. Make type "Light Active"***
Use the dropdown box to make "Light Active" the sensor type.

# Forward until Dark Wait for Dark (cont.)

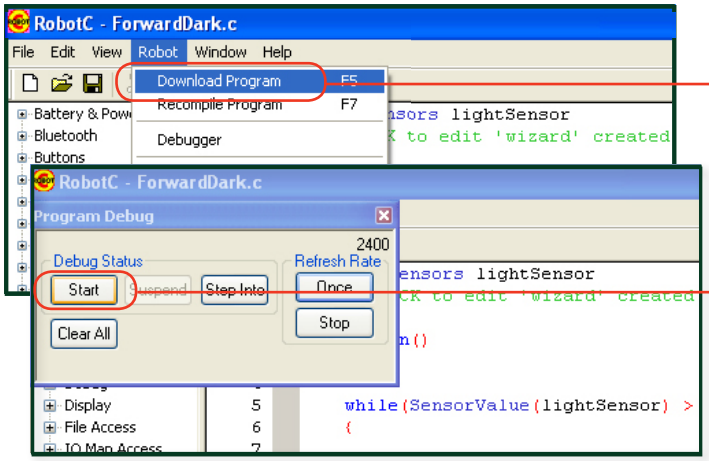
**5.** Modify the (condition) in the while() loop to watch for the lightSensor value to be greater than (brighter than) the threshold.

```
Auto const tSensors lightSensor = (tSensors) S1;
Auto //*!!CLICK to edit 'wizard' created sensor
 1
 2 task main()
 3 {
 4
 5    while(SensorValue(lightSensor) > 40)
 6    {
 7
 8       motor[motorC] = 50;
 9       motor[motorB] = 50;
10
11    }
12
13    motor[motorC] = 0;
14    motor[motorB] = 0;
15    wait1Msec(2000);
16
17 }
```

***5a. Modify this code***
Change the while() loop condition's value so that it will check whether the **Light Sensor**'s value is greater than the **threshold value** you calculated in the last lesson.

***5b. Modify this code***
Change the speed of Motors C and B to 0 so that the robot stops when it reaches a black line, rather than reversing at 50% power.

**6.** Download and Run the program.

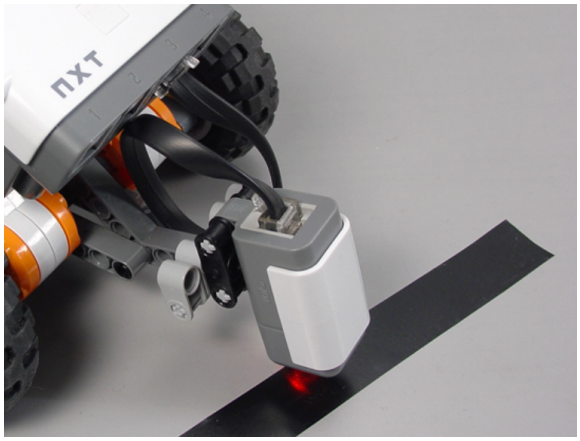



***6b. Download the program***
Click Robot > Download Program.

***6c. Run the program***
Click "Start" on the onscreen Program Debug window.

**Tip:** If your robot stops immediately or runs past the line without stopping, check your light sensor values using the View mode. Lighting conditions (position of the sun, room lighting) may have changed, and your threshold may need to be adjusted.

# Forward until Dark Wait for Dark (cont.)




**End of Section**
When the robot sees "dark" (a value below the threshold), the (condition) is no longer satisfied, and the program moves on to the stop commands, causing the robot to stop at the dark line.

---

As a final exercise, consider what would happen if you were to turn the lights off (or on) in the room where the robot is running. Make your prediction, and test it!