Game Builders Academy™

Learn ★ Grow ★ Have Fun ★ Succeed!

Robotics

Level 1

Phil Lipsky
Marco DiVeronica
Austin Smith

TEAM GBA CORP. 35 Lace Lane, Westbury, New York 11590

Game Builders Academy™

Learn ★ Grow ★ Have Fun ★ Succeed!

Robotics Fundamentals, Level 1

written by Phil Lipsky,
Marco DiVeronica,
Austin Smith

Production Editor
Sally Brecher Rosenberg

Box/Binder Design
Michael Pugliese

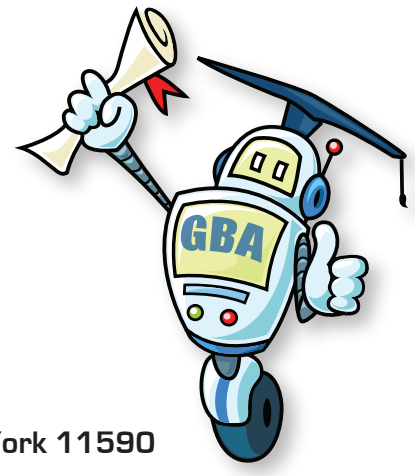
Marketing and Sales
Walter Ebe, John Taylor

Special Thanks
Melissa Ritz
Marin Kobin
Alexandra Silberglied
Stacey Rosenberg
Temma Clark-Braverman

For more information, contact Game Builders Academy, 35 Lace Lane, Westbury, New York, 11590.

Or find us on the World Wide Web at www.gbalearning.com.

ISBN-13:
978-0-9815502-4-4



TEAM GBA CORP. 35 Lace Lane, Westbury, New York 11590

Welcome to this Exciting Teaching Tool

Enjoy this wonderful journey with your students -
and let us know about your successes!

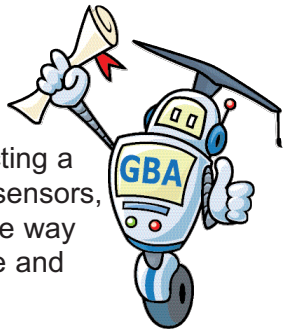
Course Overview

This course manual provides you and your students with a number of lesson plans, strategies, games, robotic builds, and programs to use in your classroom. The Robotics Level 1 program is designed to introduce students to various methods of robot constructions, as well as introduce students to fundamental Robotics programming for the LEGO® MINDSTORMS® robot. Along the way, there are many opportunities for students to apply, practice, and reinforce various academic subjects they've learned or are learning in school.

The NXT MINDSTORMS® robot, from LEGO®, is an instrument that holds enormous potential to teach children about programming fundamentals, building construction, and engineering basics. The NXT MINDSTORMS® robot can provide a fun and interactive way for students to immerse themselves in this technological age. Students will gain the skills and power to command their own personal robots.

Much of the information in this curriculum guide will provide an open book to questions, ideas, possibilities, and wonderment. Don't be surprised if your students ask questions that lead you away from the specific steps in the modules. Use this opportunity to explore a new topic with them and feel free to modify these modules to suit the students' needs. Most importantly — just have fun!

How to Use this Manual



This manual is designed to guide you and your students through the process of constructing a basic robot vehicle, programming basic movement and sounds, learning about different sensors, line detection programs, and using 3 motors and an auxiliary arm and catapult. Along the way there will be many opportunities for discussion about various academic subjects, creative and artistic ideas, and much more.

The manual is divided into 4 parts: Introduction to the Robot and Programming, Using the Sensors, Advanced Motion, and Optional Activities Each section has a number of different modules The modules are designed to be, as the name implies, modular. Many of them can be done in different sequence or on their own for variations on the given project. We highly recommend that you follow the sequence of modules as laid out in this manual. The order of modules has been carefully designed to build sequentially to the finished projects, and to build students' knowledge, skills, and confidence with each successive module. Most of the Optional Activities can be incorporated into your lessons for extra fun and learning.

Each module includes the detailed procedure(s), in a clear, numbered, step-by-step organization. Throughout the manual, there are also many sidebars on the right-hand side of the pages. These contain additional information, ideas for variations on the given module, ideas for discussion, academic connections, important information, and more.

Show samples of the Robot constructions and programs to your students during the first day of class so they can see the end goals, and so they will have a clear picture and overview of the projects with which they will be challenged.

There is a file that accompanies Modules 5-18 and shows how the programs should look after the completion of the module. These are fully working programs which can be downloaded to the student robots, should there be problems in execution within the class. These programs can be very helpful at avoiding unnecessary frustration. The robots can be "glitchy" at times and although it seems as if all the steps and procedures were executed correctly, the program still does not cooperate. How you utilize these files is up to you.

If a student misses one or more classes, when they return, they can use the Module files as their "starter" file on the day they return. Although this is, of course, not as good as having the student's complete the programs on their own, it at least allows the student to follow, complete, and learn the day's material, with a file that includes all the material that was covered on the day(s) missed.

The postmods also serve as reference for you and for your students as to exactly how the file is set up. So, as a teacher, you have each module on paper, with the steps - and you also have the specific postmod file that goes with that module to show exactly how the file looks with all the steps in the module completed.

Take the opportunities for academic discussions as far and as deep as you like, or not at all, as you see best for your class and the given day, students' responses, classroom situation, etc. Enjoy this wonderful journey with your students - and let us know about your teaching successes and students' accomplishments!

NXT Programs: Tips and Advice

If you are seeking additional information about Robotics programming or projects using the LEGO® MINDSTORMS® robots, feel free to explore the links and resources at our Web site:

www.gbalearning.com/robotics

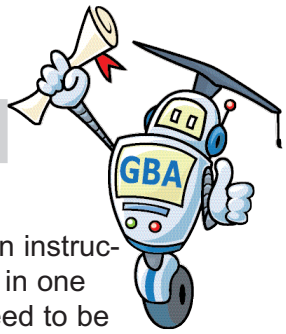
What's on the Included Disks

INSTRUCTOR FILES DISK

- **Link to Game Builders Academy Web site: http://www.gbalearning.com**
- **Post Modules Files:** There is a file that accompanies each programming Module and shows how the program should look up to, and after, the completion of each programming module. If a student misses one or more classes, when they return, they can use the Module file as their "starter" file on the day they return, or use the programming file to test their robot.
- **Student Tutorials:** PDF's of each Student Tutorial Module.
- **Student Worksheets:** These worksheets are designed to accompany specific modules in the book. Provided here in PDF form, they can be reprinted for classroom use.
- **Robot Movie Clips:** Short movie clips that can be played in class as needed. The clips are interchangeable and may be play individually or in one seating. It's a good rest time from programming and the short movie segments can give students a different perspective on these mechanical creatures.

continued on next page

Parts Management



Distribution of Materials:

LEGO® Pieces:

The management of the LEGO® pieces is one of the hardest logistics problems facing an instructor for this class. The first decision to make is storage; the LEGO® pieces can be stored in one central location or in each student's kit box. If the pieces are stored centrally, they will need to be separated into different containers, otherwise pieces will be too difficult to find when they are needed. The pieces can either be stored in a parts locker, purchased in a hardware store, or large freezer bags. This approach allows for management control with the LEGO® pieces, but it takes more time to distribute them to the students. Expect to spend time to organize them before the first class and after the last class. The other approach is to keep the pieces in the NXT robotics kit. This keeps the pieces handy, but they have a greater tendency to be lost since the students will be in charge of them during class. The students tend to become distracted and play with them. If the pieces are left in the kit, it's beneficial to remove pieces that won't be used during classes. Some of these pieces include the LEGO® men, rubber band, and translucent blocks.

Sensors:

The sensors can be stored either in the student's kit box or in a ziploc bag with their data wires. Note that it is important to keep control of the sensors since they can become expensive to replace. To maintain order with these, it is recommended that they be numbered, and kept with the robot they belong to.

Robots:

Make sure that there is enough storage space for the robots. Since they will be used every class, they should be stored some place where they will be retrieved easily. To accommodate the full robot build, including the catapult, there will need to be a cube of space around six inches by six inches by nine inches. Note that more space may be needed if the students are allowed to make their own custom builds.

Distribution depends on the lesson and the class. Normally, it is best to distribute the robots before class to save time, but sometimes there are exceptions. During the early classes, it is easier to distribute the robots when they will be needed because the robots will be new and distracting to the students. Also if the class is routinely distracted by the robots, it's best to wait until they need to build parts of the robot or download their programs. In cases like these, it's best to place the robots in an easily accessible area before class for quick distribution.

Because each robot will be assigned to a specific student or team, it will be important to have a way to distinguish the robots from each other. One easy way to do this is to place masking tape on the robots, and write the student's name on the tape. Or assign each robot a number, and write down which number belongs to which student. Names work best when each student has access to their own robot, and numbers work best when robots are shared among multiple students. If time permits students can also be instructed to create a flag for their robots (details for this are on page 123), which will help create a team dynamic, and foster a connection between the students to their robot.

USB cords:

USB cords can either be stored separately, in each robot kit or together. It is easier to keep track of the USB cords when they are together, but it is also possible for them to get tangled up. To prevent this from happening, when collecting the USB cables, lay them straight together. Then loosely wrap them to fit where they will be stored. The USB cables only need to be distributed whenever a program will be downloaded to the robot.

Computers:

If possible, each student should have access to a computer to gain experience writing programs. When students are sharing a robot in teams, each student's programs can be downloaded to the robot. During testing, each student can take turns testing their program regardless if it's the same program.

continued on next page

Parts Management

continued from previous page

Robot Maintenance:

Repair:

One benefit of the robot's design is that it is easy to repair. If the robot becomes disassembled for any reason, the best approach is to take a fully assembled robot, and use it as a model to repair the other. A good indication that the robot is properly repaired is that the wheels will spin without touching technic beams, and that the NXT brick is parallel to the ground.

Charging:

The robots will need power to operate. This will either come in the form of the included rechargeable battery pack, or AA batteries. To accommodate this, a number of power strips will be needed to charge the battery packs. While the battery pack is in the robot, connect the included AC adapter to the front of the robot and plug it in. If multiple classes are taught, the robots can be charged in shifts; otherwise the robots will have to be left charging after class. Even if the robots are charged, it's a good idea to keep emergency AA batteries to power the robots just in case. To replace the battery pack with AA batteries, the NXT brick will have to be removed from the robot, and the battery pack will have to be removed from the bottom. Then the batteries will have to be inserted, and the battery cover from the NXT kit will have to be snapped into place.

Play Area:

Robot Arena:

Testing the robots will require a lot of room for some activities, and watching the robots can excite the students. To handle this it helps to have an area designated for official testing; a "Robot Arena." Having a set area for students to test their robots provides structure to the testing, and allows the teacher to establish rules for the arena. These rules should be made clear to the students, and posted to avoid confusion. Here are a sample set of rules:

1. Only robots in the Robot Arena. (Students will want to hover over their robot and nudge their robots back on course. This should be discouraged to maintain order. For certain modules (tethered control) this rule will have to be relaxed.)

2. Teams will line up to be admitted into the Robot Arena. Only when a team is called into the Robot Arena will their robot be allowed in. (This will keep the robot arena orderly.)

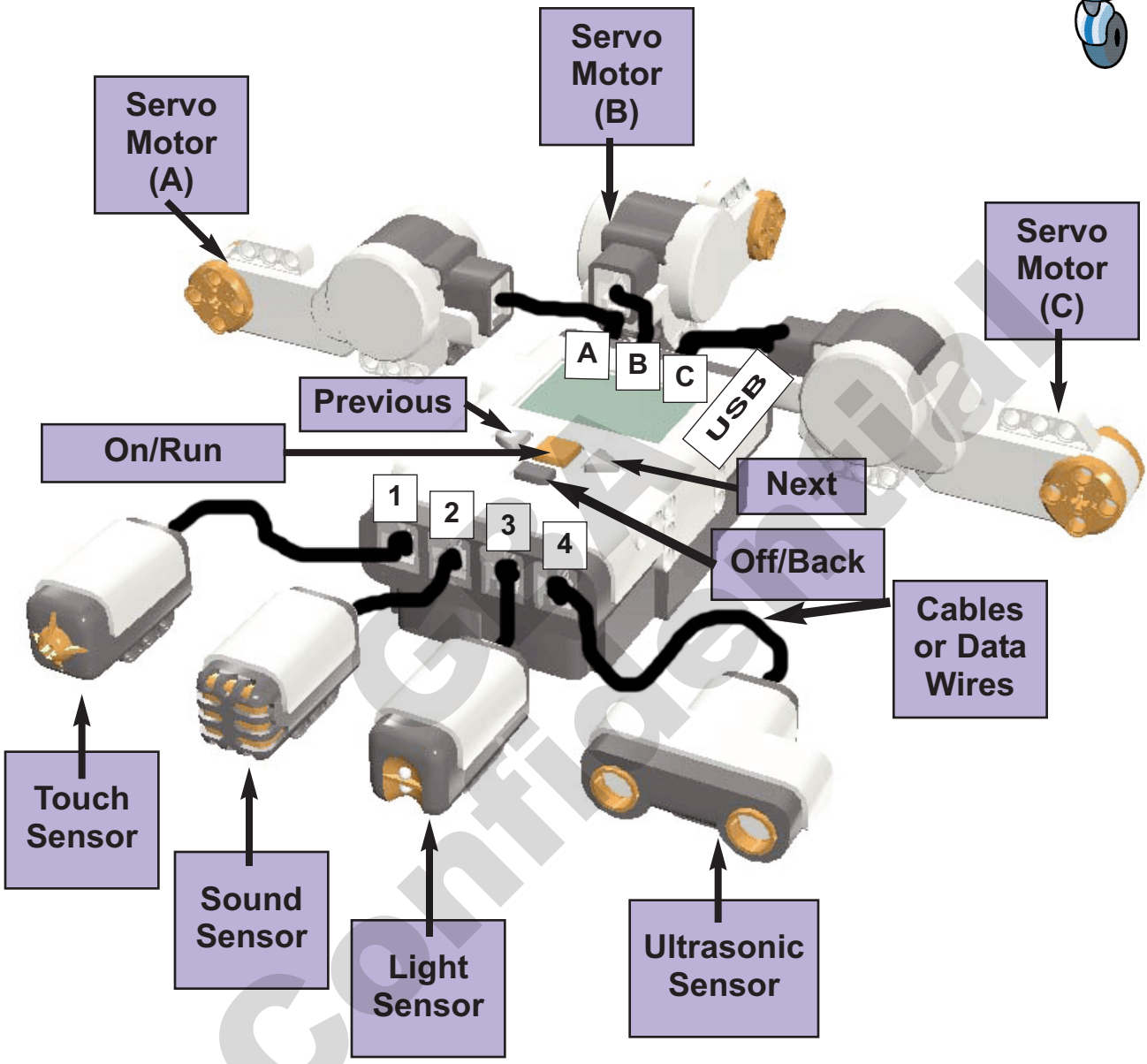
3. Only team members and instructors can touch their robots. (Students can sometimes become impatient waiting for other teams, and this rule will prevent any disagreements from starting.)

Side Areas:

With larger groups, there may be a need for smaller side areas outside of the robot arena for debugging and testing. These areas need to be kept orderly for the safety of both the students and robots, but the rules can be more lenient to allow for diagnosing why a program isn't working correctly. For example, to test an aspect of a program using the ultrasonic sensor, the student or instructor may have to manipulate the position of the robot manually. It is recommended, if there is enough staff, to have one person in charge of the robot arena, and other staff people distributed among the side testing area.

END 

Brick Overview and Connections Chart



Teacher's Manual Table of Contents

continued on next page

Teacher's Manual Table of Contents

Part I - Introduction to Robot and Programming

This section will introduce students to the robot and the NXT-G program.

Part I Modules

Module 1: Introduction to Brick and Sensors Using Try-Me Programs

OBJECTIVE:
There are a number of very simple test programs that are built directly into the NXT brick. They are all accessed through the "Try Me" menu. In this lesson, students will use these programs to become introduced to the brick and the sensors.

BUILD:

PARTS:
1 NXT brick
1 touch sensor
1 light sensor
1 ultrasonic sensor
1 servo motor

The Sensors Love Us:

Don't forget this mnemonic device to help remember which port each sensor plugs into. **T**he **S**ensors **L**ove **U**s: **T**ouch, **S**ound, **L**ight, **U**ltrasonic.

PROGRAM:
There is no program for this module.

Why isn't this working?

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The proper program is selected, and the brick is currently running it.

The Happy Button:

As a way to keep the kids focused and following instructions, we sometimes use names that are slightly comical. One example of this is calling the orange button on the brick the "Happy Button", since we use it to make selections on the brick. Another example of this is the "Happy Sound" which is the little jingle that plays when the brick is turned on.

A note on the "Try-Touch" program:

A good talking point during this program is binary input vs. input with wider range. The touch sensor can either be on or off, just like a light switch.

1. **Touch Sensor:** Use a data wire to **connect the touch sensor to port 1 on the brick**. To do this, push one end of the data wire into the data port in the touch sensor, and the other end into data port 1, at the bottom of the brick.

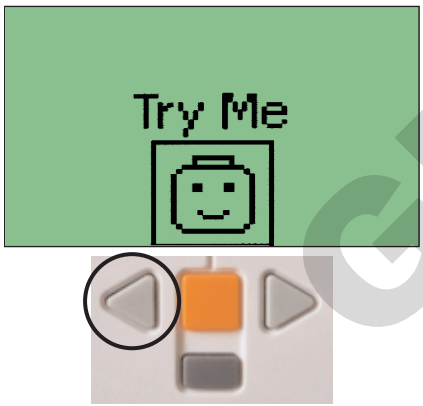
continued on next page

Module 1: Introduction to Brick and Sensors Using Try-Me Programs

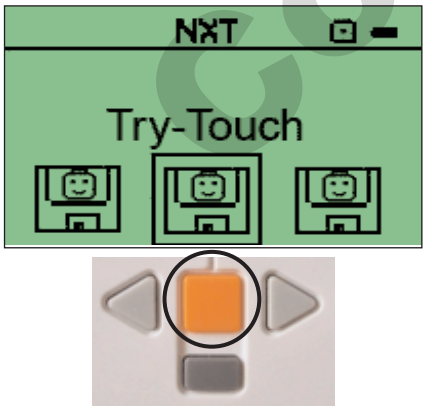
continued from previous page

2. **Press and hold the orange button** on the brick to turn it on. A sound plays and the LEGO® MINDSTORMS® logo is shown as the brick powers up. **My Files** will be highlighted.

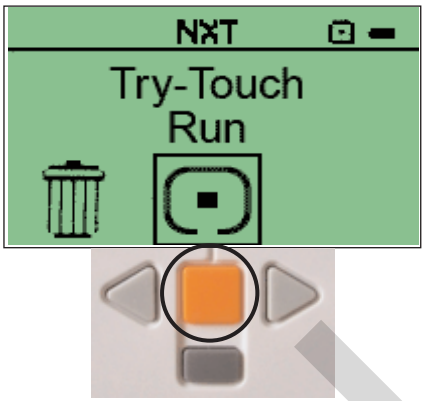
3. **Press the left arrow button once** to scroll to the **"Try Me"** menu.



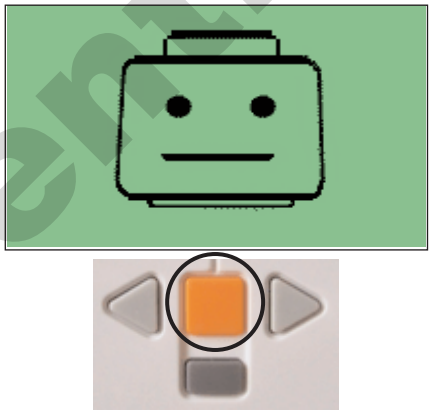
4. With the "Try Me" menu selected, **press the orange button once** to bring up the selection of "Try Me" programs. The **"Try-Touch"** program should now be highlighted.



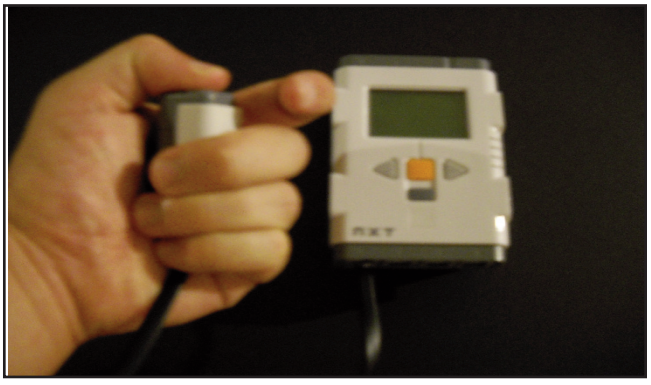
5. With the "Try-Touch" program selected, **press the orange button once** to bring up the **Run** screen for this program.



6. **Press the orange button once more to run the program**. A LEGO® man's face should appear on the display screen of the brick.



7. With the program running, **press the button on the touch sensor multiple times**. The face on the display changes on the smiley-face, and a sound should play whenever the button is depressed.



continued on next page

Module 1: Introduction to Brick and Sensors Using Try-Me Programs

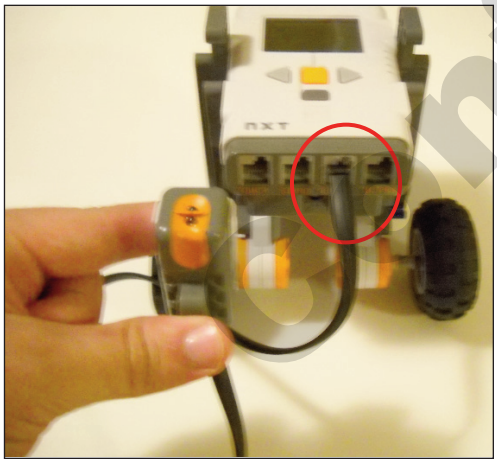
continued from previous page

8. When ready to continue to the next "Try Me" program, **press the gray button once**. "Try-Touch Aborted" will appear on the screen, and the screen will return to the run screen for the program. **Press the gray button once again to return to the "Try Me" menu**

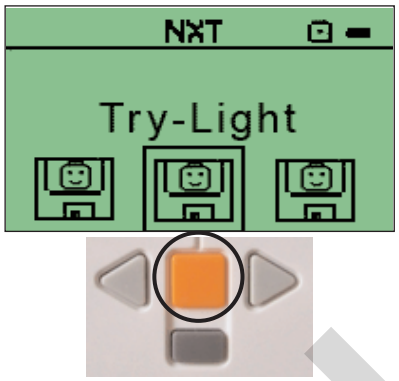
A FEW NOTES ON THE "TRY-LIGHT" PROGRAM

Since the "Try-Light" program is more variable than the "Try-Touch" program, kids enjoy experimenting with it. You can encourage this by not explaining how the pitch changes and asking the kids to explain to you what is happening.

9. **Light Sensor: Use a new data wire to connect the light sensor to port 3** on the brick. You can leave the previous data wire connected.

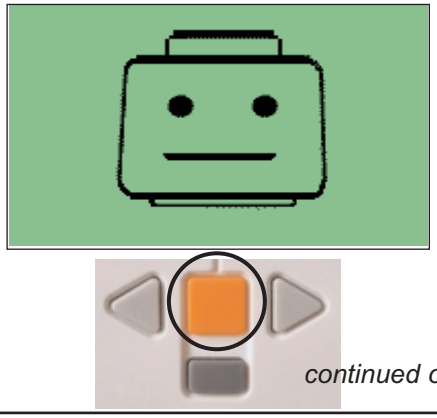


10. While in the "Try Me" menu **press the left button once.** The **"Try-Light"** program should be highlighted.



11. With the **"Try-Light"** program selected, **press the orange button once** to bring up the run screen for this program.

12. **Press the orange button once more to run** the program. A red light should be emitting from the light sensor, and a tone will be playing from the robot. As the red light reflects off of objects of darker hues, the pitch of the sound becomes lower, and as the light reflects off of lighter objects, the pitch raises.



continued on next page

Module 1: Introduction to Brick and Sensors Using Try-Me Programs

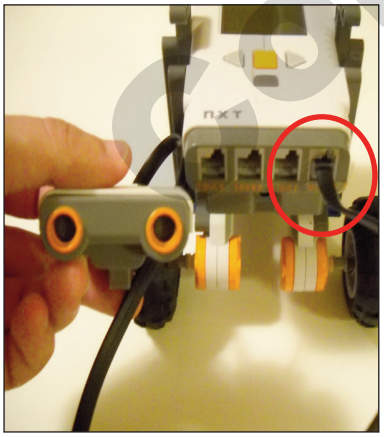
continued from previous page

13. When ready to continue to the next "Try Me" program, **press the gray button once**. "Try-Light Aborted" will appear on the screen, and the screen will return to the run screen for the program. **Press the gray button once again to return to the "Try Me" menu.**

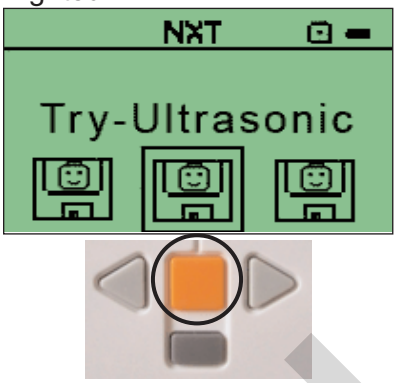
A NOTE ON THE "TRY-ULTRASONIC" PROGRAM:

The "Try-Ultrasonic" program is similar to the "Try-Light" program since they both play a sound that can be modified by the student's actions, but the fact that there's a point at which the ultrasonic sensor can no longer sense things is significant. This is a great time to teach the children about the limitations of the sensors (that past a certain distance, the ultrasonic sensor no longer reads distances), and thresholds (you can talk about the point where a light switch goes from on to off, and how it's similar to the farthest point the ultrasonic sensor can read).

14. **Ultrasonic Sensor: Use a new data wire to connect the ultrasonic sensor to port 4 on the brick.** You can leave the previous data wires connected.

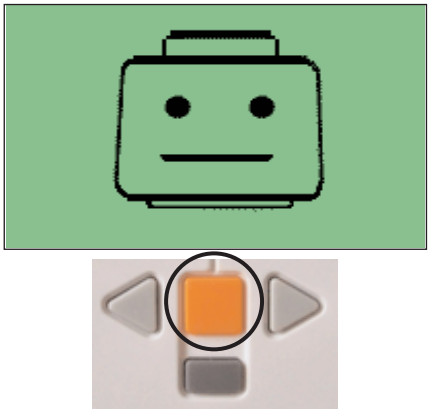


15. While in the "Try Me" menu **press the left button three times**. The "Try-Ultrasonic" program should be highlighted.



16. With the "Try-Ultrasonic" program selected, **press the orange button once** to bring up the **run** screen for this program.

17. **Press the orange button once more to run the program.** The brick will emit a tone. The closer the ultrasonic sensor is to an object, the lower the tone will be in pitch, and the farther away the sensor is, the higher the tone will be, until the point at which the ultrasonic sensor can no longer sense an object.



continued on next page

Module 1: Introduction to Brick and Sensors Using Try-Me Programs

continued from previous page

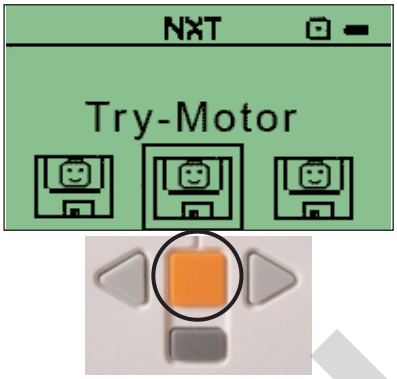
18. When ready to continue to the next "Try Me" program, **press the gray button once**. "Try-Ultrasonic Aborted" will appear on the screen, and the screen will return to the run screen for the program. **Press the gray button once again to return to the "Try Me" menu**.

A NOTE ON THE "TRY-MOTOR" PROGRAM

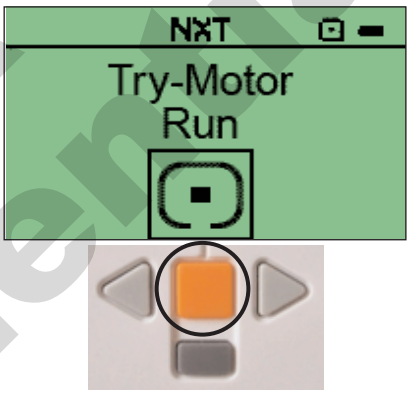
A talking point for this program can be the difference between input and output. After defining input and output, you could ask about how a motor or wheel is usually used (output), and how we can take in data from it (obtain data about how the motor is turning).

19. **Servo Motor: Use a new data wire to connect a motor to port A on the top of the brick.** You can leave the previous data wires connected.

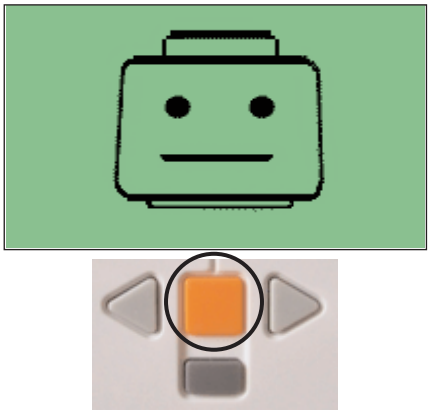
20. While in the "Try Me" menu press the left button twice. The **"Try-Motor"** program should be highlighted.



21. With the "Try-Motor" program selected, **press the orange button once** to bring up the **run** screen for this program.



22. **Press the orange button once more to run** the program. The brick will emit a tone. The tone will change in pitch as the students spin the orange wheel of the motor. The more the wheel has rotated away from its starting position the higher the tone will become.



continued on next page

Module 1: Introduction to Brick and Sensors Using Try-Me Programs

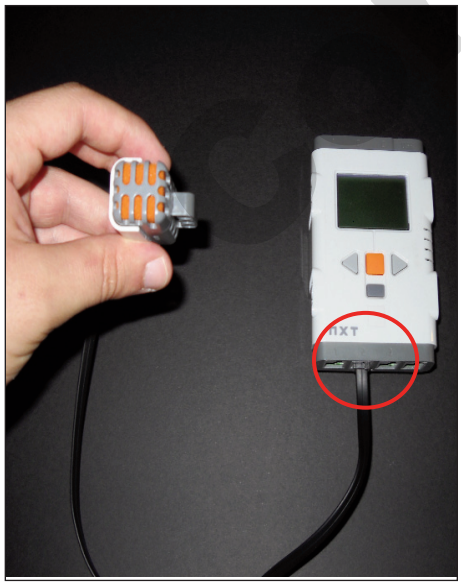
continued from previous page

23. When ready to continue to the next "Try Me" program, press the gray button once. "Try-Motor Aborted" will appear on the screen, and the screen will return to the run screen for the program. **Press the gray button once again to return to the "Try Me" menu.**

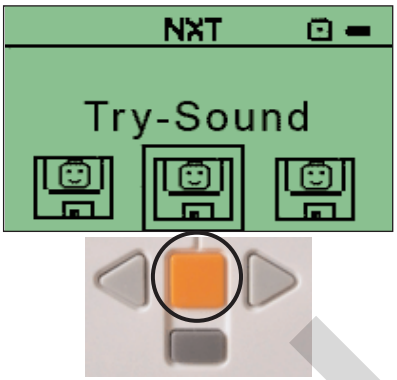
A NOTE ON THE "TRY-SOUND" PROGRAM.

This program shows a frustrating part of dealing with the sound sensor. It is almost impossible to get the motor to stop completely, and you can have the students speculate on why (since there are always small amounts of ambient noise). Because of this, the sound sensor is used sparingly in this book.

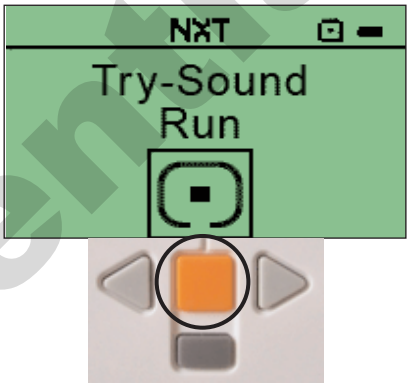
24. **Sound Sensor: Use a new data wire to connect the sound sensor to port 2 on the brick**. You can leave the previous data wires connected. **It is important that the motor is connected to the brick**, but you'll need to connect the **motor to port B instead of A**, otherwise the program won't run properly.



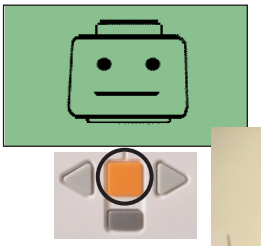
25. While in the "Try Me" menu **press the left button four times**. The **"Try-Sound"** program should be highlighted.



26. With the "Try-Sound" program selected, **press the orange button once** to bring up the **run** screen for this program.



27. **Press the orange button once more to run** the program. The wheel on the motor should start spinning. The more noise the sound sensor picks up, the faster the motor will spin.



END

Module 2: Experiment and Play with the View Menus

OVERVIEW:

The view tools, which are built into the NXT brick, allow students to see numerical readings of what the various sensors are detecting. These view tools are useful to familiarize students with the brick and sensors. The view tools will also be useful later on as a diagnostic tool, letting students and teachers see what the sensors are reading.

BUILD:

- Brick
- 5x data wires
- 1x touch sensor
- 1x light sensor
- 1x sound sensor
- 1x ultrasonic sensor
- 1x servo motor

PROGRAM:

There is no program for this module.

The Sensors Love Us:

Don't forget this mnemonic device to help remember which port each sensor plugs into. **T**he **S**ensors **L**ove **U**s: **T**ouch, **S**ound, **L**ight, **U**ltrasonic.

Why isn't this working?

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The proper program is selected, and the brick is currently running it.

The default ports:

Technically speaking, you could plug in the sensors to any port 1-4, and the motors into any port A,B or C, but the default ports used during programming are 1:Touch, 2:Sound, 3:Light, 4:Ultrasonic, A: Auxiliary motor, B and C: Movement motors. Because sticking to these defaults makes programming easier, we stick with them.

TEST:

1. Use a data wire to connect the touch sensor to port 1 on the brick. To do this push one end of the data wire into the data port in the touch sensor, and the other end into data port 1 at the bottom of the brick.

Navigating the menus:

The directions on navigating through the menus assumes that you have just exited the previous program. If that's not the case, ignore the number of arrow button presses and just browse the menu for the indicated tool.

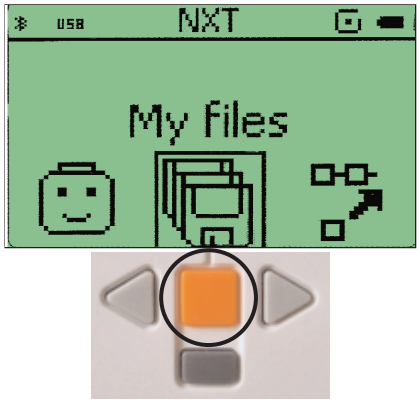
How are the view tools useful later on:

The view tools are useful for teachers after this lesson to test and calibrate the sensors. For example, if you want the robot to be able to tell the difference between two different colored objects, you can use the "Reflected Light" view tool to measure the reading of both objects before class. Also, if a student's robot is not performing correctly, you can run the appropriate view tool to make sure that the sensor is operating correctly.

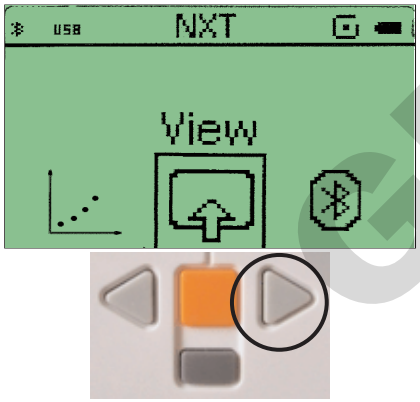
continued on next page

Module 2: Experiment and Play with the View Menus *continued from previous page*

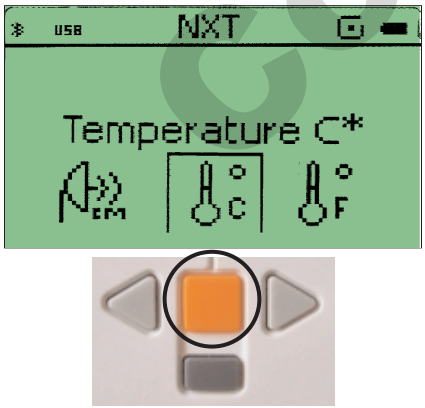
2. Press and hold the orange button on the brick to turn it on. A sound plays and the LEGO® MINDSTORMS® logo is shown as the brick powers up. "My Files" will be highlighted.



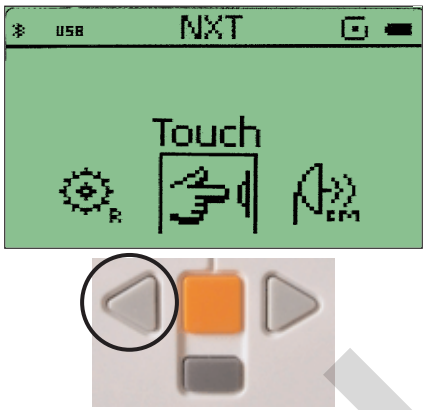
3. Press the right arrow button three times to scroll to the View menu.



4. With the View menu selected, press the orange button once to bring up the selection of View tools. The "Temperature °C" tool should now be highlighted.



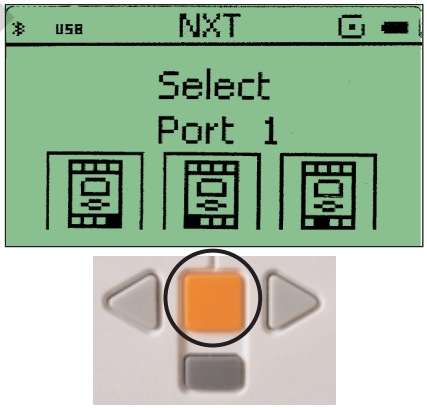
5. Press the left arrow button three times to navigate to the "Touch" view tool.



A NOTE ON THE TOUCH VIEW TOOL:

A good talking point during this program is binary input vs. input with wider range. The touch sensor can either be on or off, just like a light switch. This can be seen easily and connected to binary mathematics since the output is only 1 or 0.

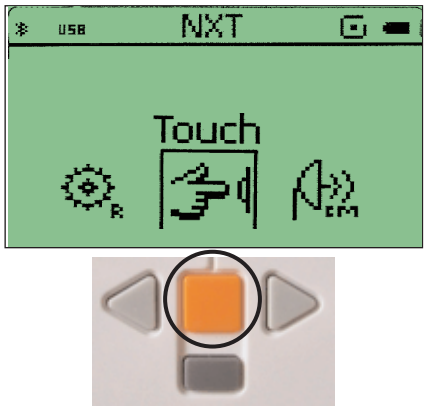
6. With the "Touch" view tool selected, press the orange button once to bring up the port selection screen for this tool.



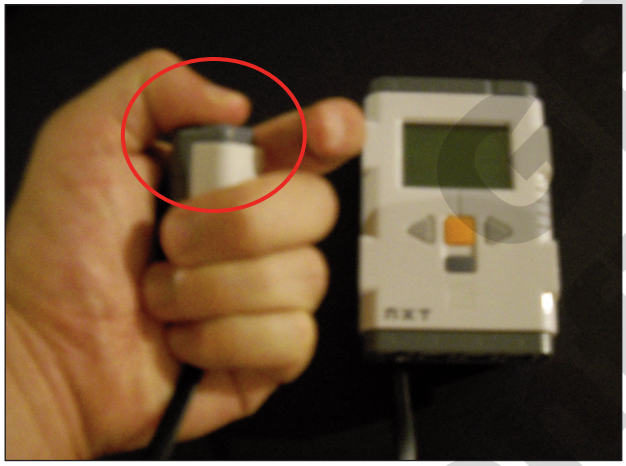
continued on next page

Module 2: Experiment and Play with the View Menus *continued from previous page*

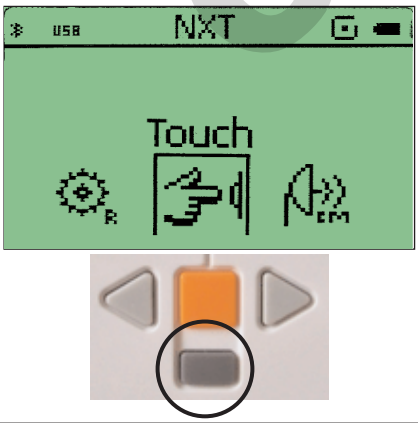
7. Press the orange button once more to select port one. The "Touch" view tool should now be activated, and the number 0 should be displayed to the screen.



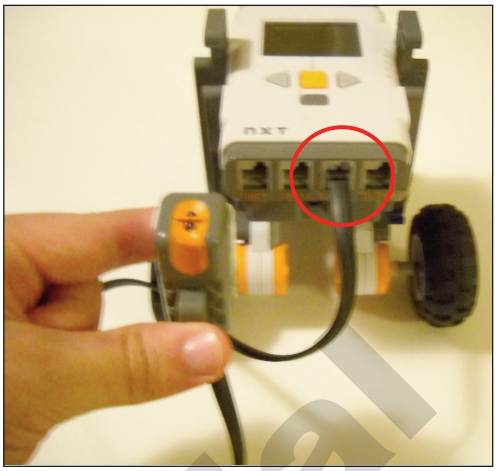
8. With the program running, press the button on the touch sensor multiple times. The number on the screen turns to 1 whenever the touch sensor is pressed in and 0 if it is not.



9. When ready to continue to the next view tool, press the gray button once. The "Touch" view tool will once again be highlighted.



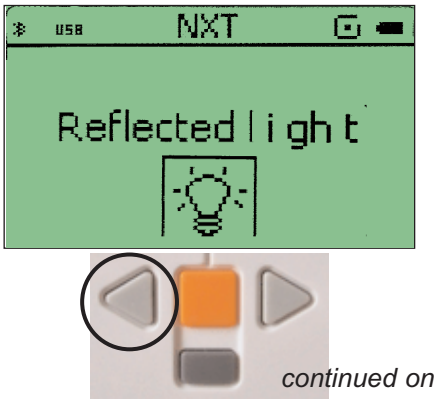
10. Use a new data wire to connect the light sensor to port 3 on the brick. You can leave the previous data wire connected.



NOTES ON THE REFLECTED LIGHT VIEW TOOL:

Since this tool shows its output as a percentage instead of a binary output, the kids can experiment with it much more. To encourage this, you can set up challenges, such as asking the kids to go around the classroom to find the highest and lowest levels they can, trying to see if they can get 100% or 0%. 100% can be achieved through using a mirror, or by shining one robots light into another robots sensor, and 0% can be achieved by totally blocking the sensor portion from all light. This is also a good spot to introduce the mathematics of percentages, and what they mean, as well as discuss how to use the light sensor effectively, by holding it close enough to the object to be able to see it, but not so close as to block the sensor.

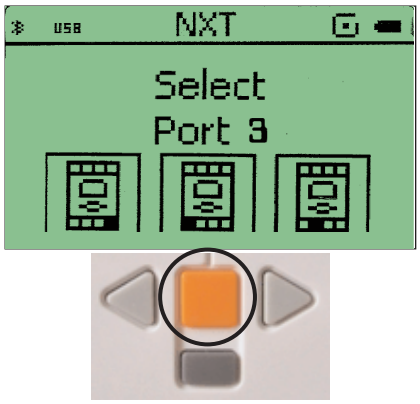
11. While in the view menu press the left button four times. The "Reflected Light" view tool should be highlighted.



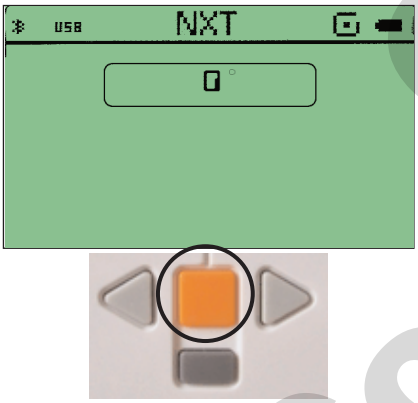
continued on next page

Module 2: Experiment and Play with the View Menus *continued from previous page*

12. With the "Reflected Light" view tool selected, press the orange button once to bring up the port selection screen for this tool. Press the right arrow twice to select port 3.



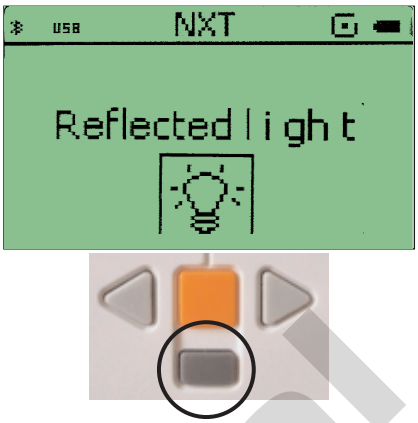
13. Press the orange button once more to run the tool. A red light should be emitting from the light sensor, and a percentage should be shown on the screen. As the red light reflects off of objects of darker hues, the percentage lower, and as the light reflects off of lighter objects, the percentage raises. The percentage shown is how much of the red light is taken back into the light sensor.



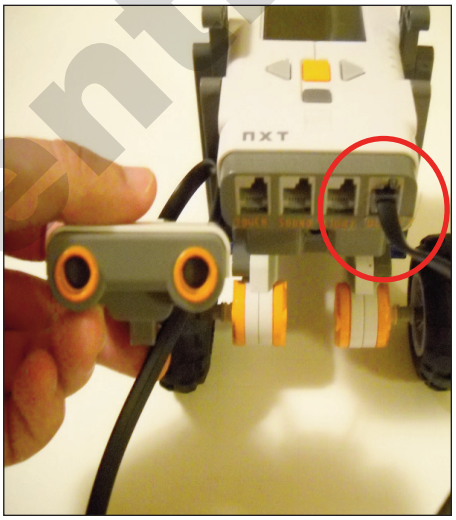
A NOTE ON THE ULTRASONIC INCH VIEW TOOL:

The ultrasonic inch view tool illustrates the idea of thresholds even better than it's "Try Me" counterpart, since it uses question marks to indicate that there is nothing below it's threshold value. The ultrasonic inch and ultrasonic centimeters are similar enough that either could be used for this demonstration, however children may be more familiar with inches. In later programs centimeters will be used however, since they give greater control over the robot, since they are smaller.

14. When ready to continue to the next view tool, press the gray button once. The "Reflected Light" view tool will once again be highlighted.



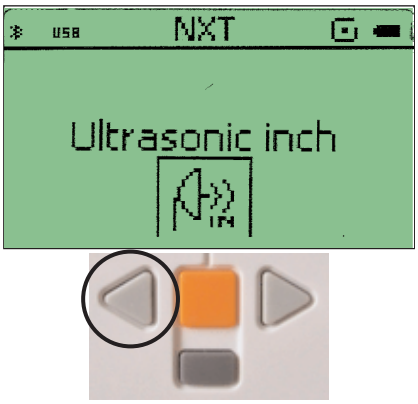
15. Use a new data wire to connect the ultrasonic sensor to port 4 on the brick. You can leave the previous data wires connected.



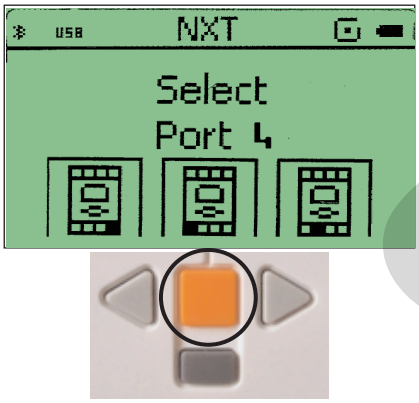
continued on next page

Module 2: Experiment and Play with the View Menus *continued from previous page*

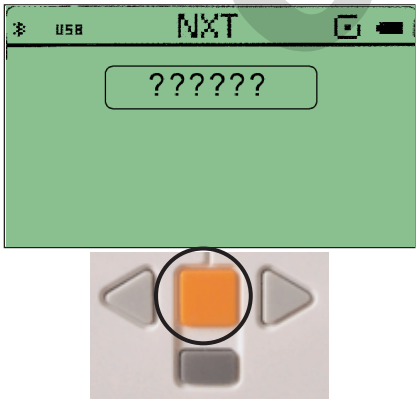
16. While in the view menu press the left button six times. The "Ultrasonic Inch" view tool should be highlighted.



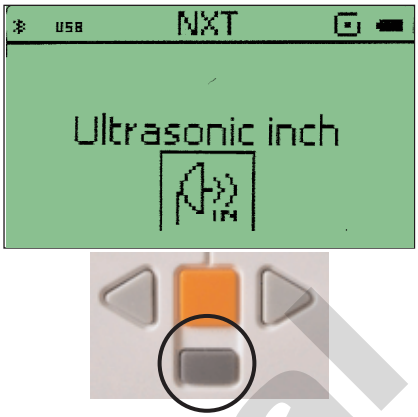
17. With the "Ultrasonic Inch" view tool selected, press the orange button once to bring up the port selection screen for this program. Press left once to select port 4.



18. Press the orange button once more to run the tool. The brick will either show a number followed by "In" or a series of question marks. The question marks will show that the ultrasonic sensor is not currently reading an object; the number will show that it's reading an object that many inches away.



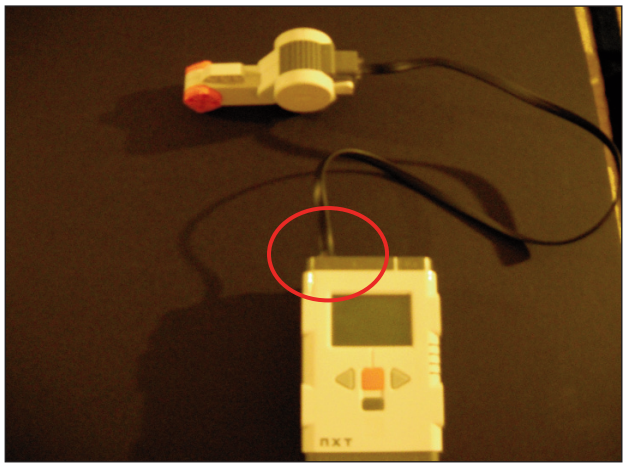
19. When ready to continue to the next view tool, press the gray button once. The "Ultrasonic Inch" view tool will once again be highlighted.



A NOTE ON THE MOTOR DEGREES VIEW TOOL:

Like the ultrasonic sensor, the motor has two different view tools, rotations and degrees. Degrees are far smaller than wheel rotations, so they are used here so that students can more easily see that they are affecting output. One good talking point during this step is the difference between robot movement and wheel movement. If you want the robot to rotate 90°, that doesn't match up to 90° of a wheel rotation. This is important to remember later on when programming movement.

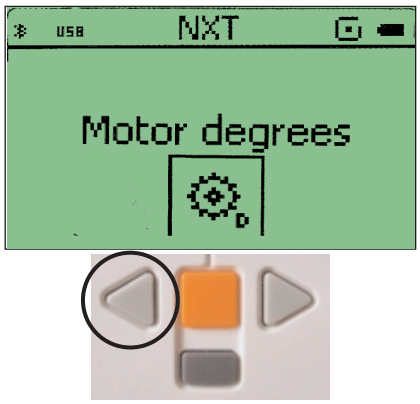
20. Use a new data wire to connect a motor to port A on the top of the brick. You can leave the previous data wires connected.



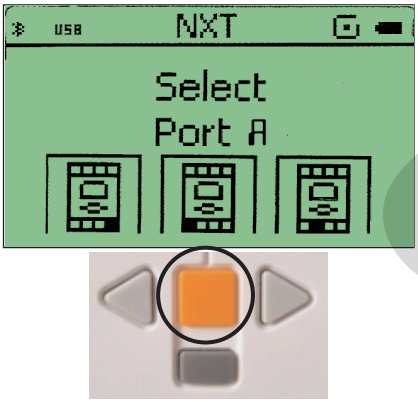
continued on next page

Module 2: Experiment and Play with the View Menus *continued from previous page*

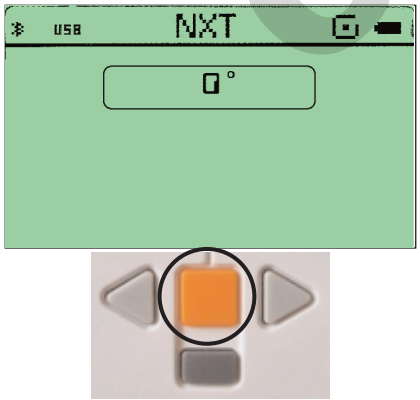
21. While in the view menu press the left button two times. The "Motor degrees" view tool should be hi1ghlighted.



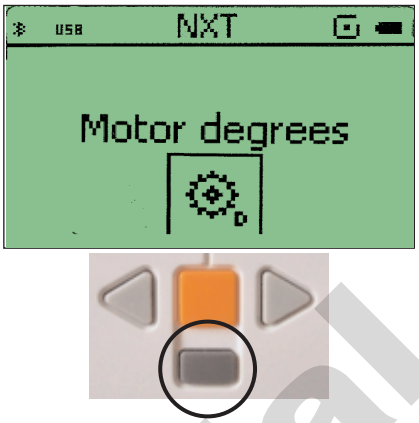
22. With the "Motor Degrees" view tool selected, press the orange button once to bring up the port selection screen for this program. Port A should be selected automatically.



23. Press the orange button once more to run the tool. The brick will show a number followed by the degrees symbol. Turning the orange wheel clockwise will raise the number, while rotating it counter clockwise will lower the number.



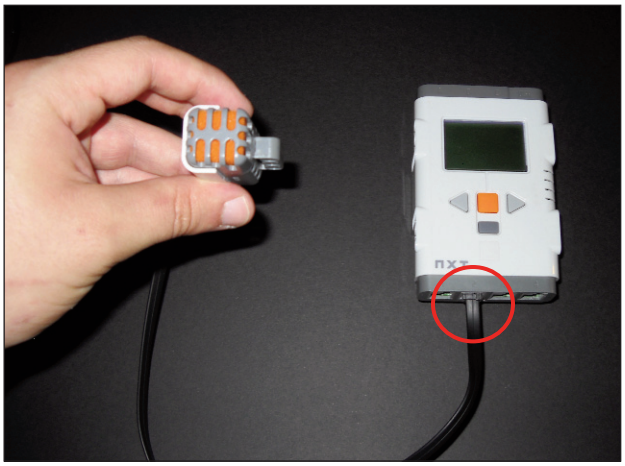
24. When ready to continue to the next view tool, press the gray button once. The "Motor degrees" view tool will once again be highlighted.



A NOTE ON THE SOUND DB VIEW TOOL:

This program shows a frustrating part of dealing with the sound sensor. It is almost impossible to get the sound sensor to register 0% noise, and you can have the students speculate on why (since there is always a small amounts of ambient noise). Because of this, the sound sensor is used sparingly in this book. The difference between the Sound dB view tool and the Sound dBA view tool is that the dBA view tool is limited to the range of sound a person can hear, while the other is not. One good talking point for this section is the limitations of the sound sensor. Although we can make noises louder than those that register 100% on the robot, it can't differentiate them. Therefore it is better to work within the volumes the robot can register.

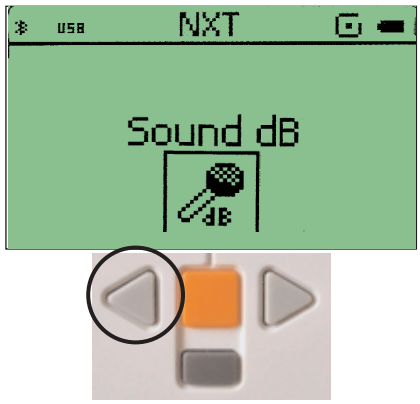
25. Use a new data wire to connect the sound sensor to port 2 on the brick. You can leave the previous data wires connected.



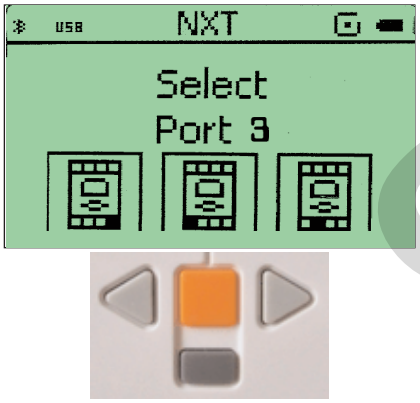
continued on next page

Module 2: Experiment and Play with the View Menus *continued from previous page*

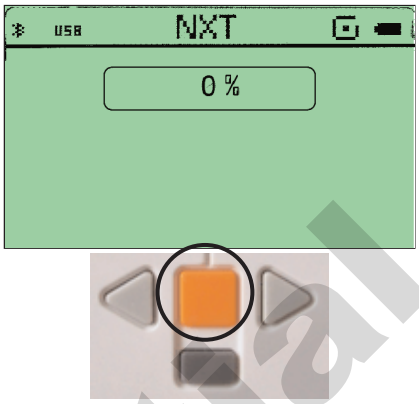
26. While in the view menu press the left button five times. The "Sound dB" view tool should be highlighted



27. With the "Sound dB" view tool selected, press the orange button once to bring up the port selection screen for this program. Press the right button twice to select port 3.



28. Press the orange button once more to run the tool. The brick will show a percentage which indicates the level of sound in the room. Zero percent represents absolute silence, and 100% indicates that the sound sensor reached the maximum volume it can detect.



END

continued on next page

Module 3: Building the Base Robot

OVERVIEW

In the lesson, students will build the base robot. This specific build of the robot allows for great flexibility in adding additional motors, parts, etc. It is an efficient design that uses the minimum number of parts required. This build is the basis for most of the following lessons.

TEACHER'S TIP

To help out the Students; you may want to pre-bag these parts in separate bags to make it easier for them to access. Be aware of the number of parts available in order to cut down on part management issues. Also be sure to disconnect any existing data wires currently connecting the sensors or motors to the NXT Brick.

BUILD LIST:

1x NXT brick
2x NXT Motors
2x Tires
2x Rims
2x Thin Wheels
6x Bushings
1x half-bushing
1x gray axle (length 3)
3x black axles (length 6)
1x H-piece
3x technic beams (length 9)
2x technic beams (length 11)
6x technic beams (length 3)
2x L-beams (length 3x5)
14x Short Friction Pins
8x Long Friction Pins
2x axle pins with friction (blue)
1x black 90 degree joint (length 3)
1x gray 90 degree joint (length 2)

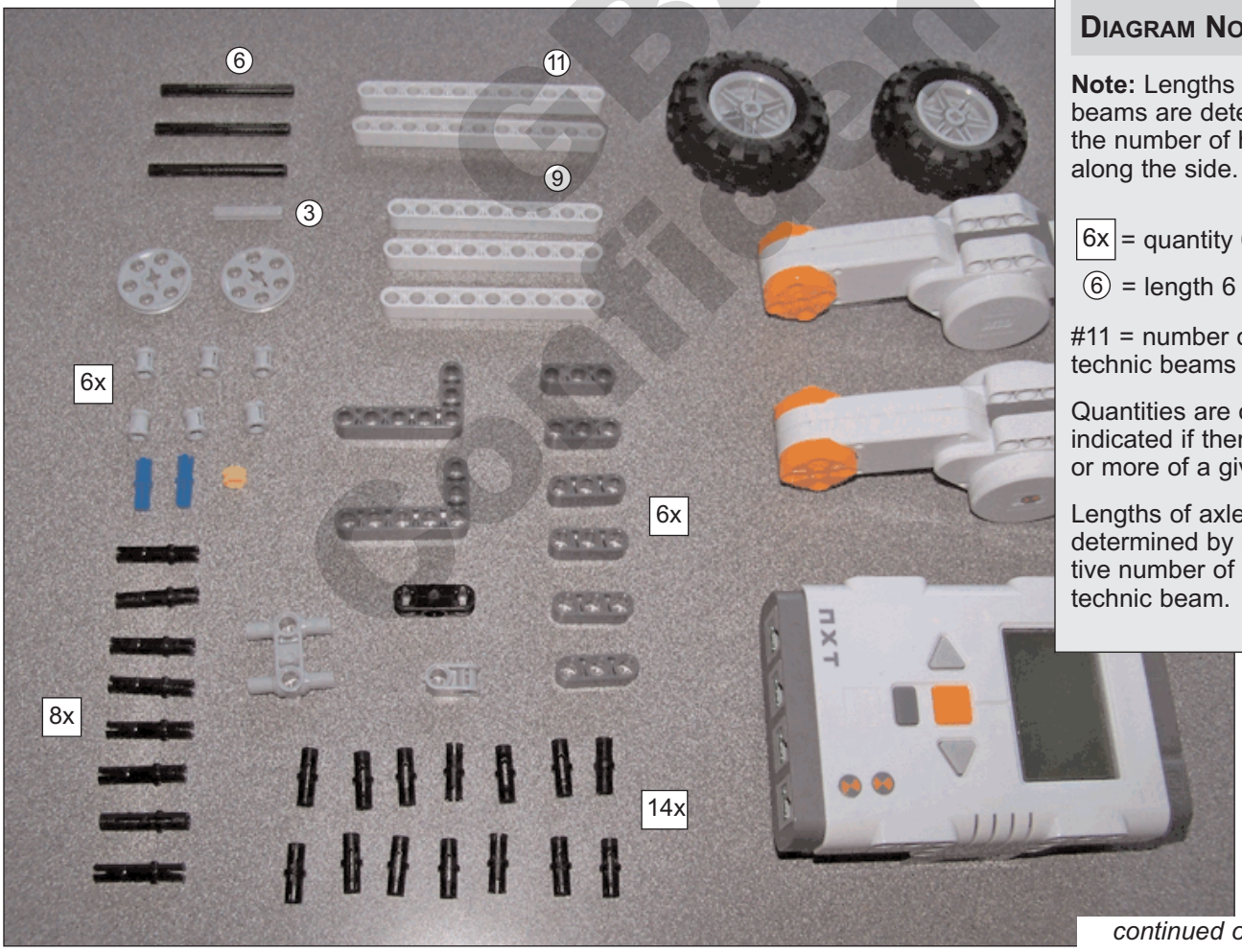


DIAGRAM NOTES

Note: Lengths on technic beams are determined by the number of holes along the side.

6x = quantity 6

⑥ = length 6

#11 = number of holes in technic beams

Quantities are only indicated if there are 6 or more of a given part.

Lengths of axles are determined by comparative number of holes on a technic beam.

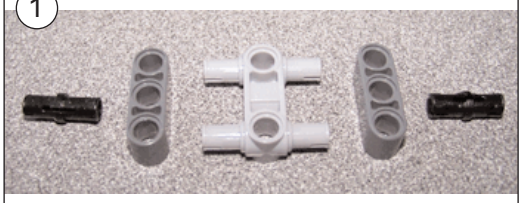
continued on next page

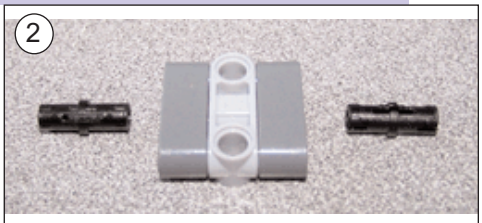
Module 3: Building the Base Robot *continued*

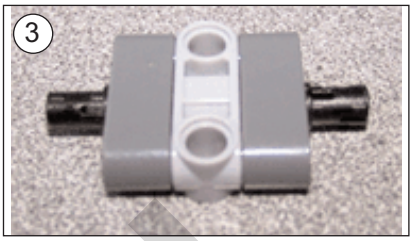
1. Assemble small center piece:

Parts Needed:

2x technic beams (#3)
1x H-piece
2x Short Friction pins

1


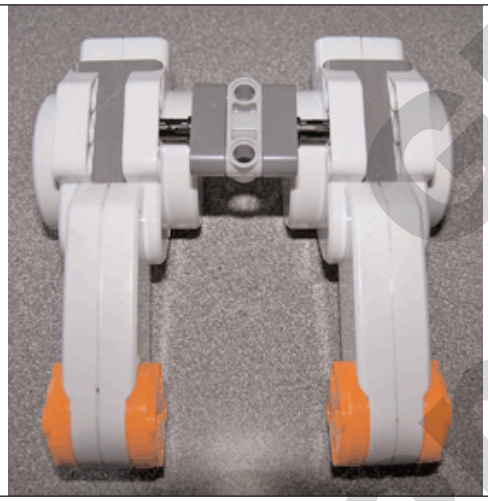
2


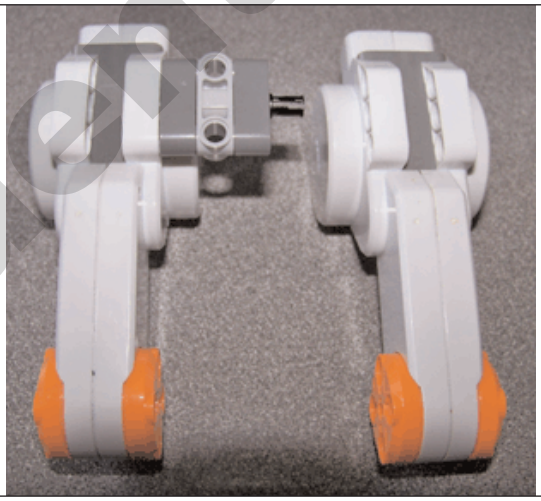
3


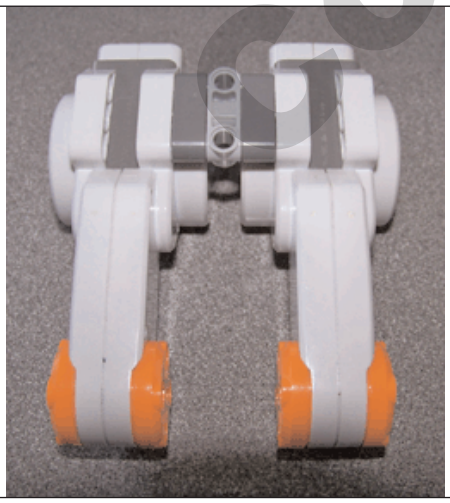
2. Connect center piece in between two motors

Parts Needed:

Small Center piece
2x NXT Motors

1


2


3


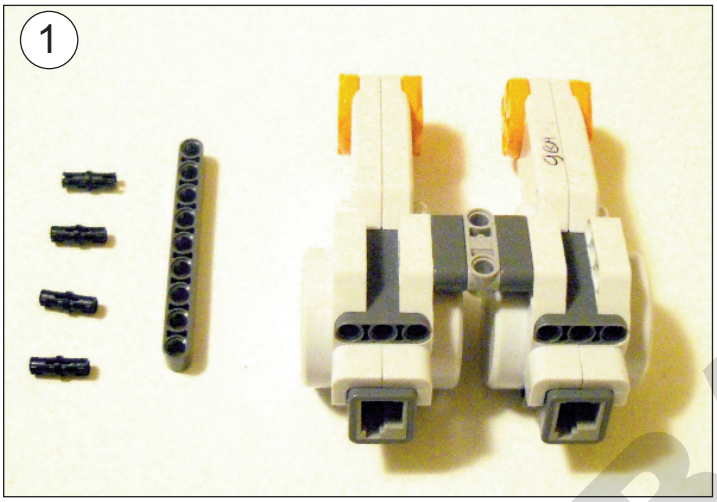
continued on next page

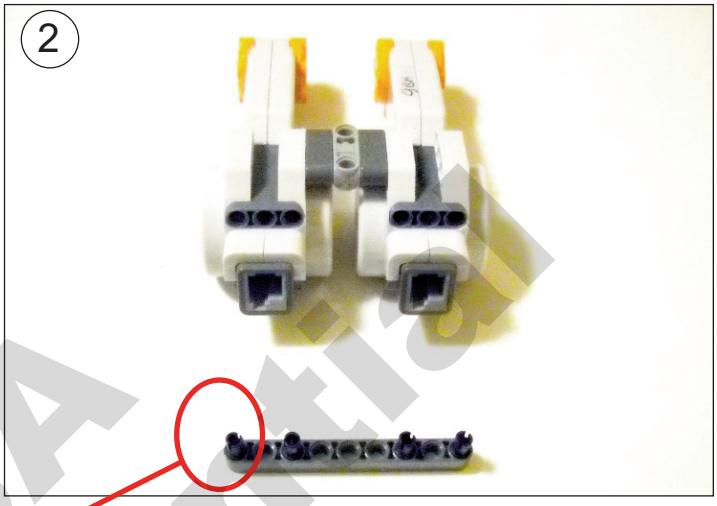
Module 3: Building the Base Robot *continued*

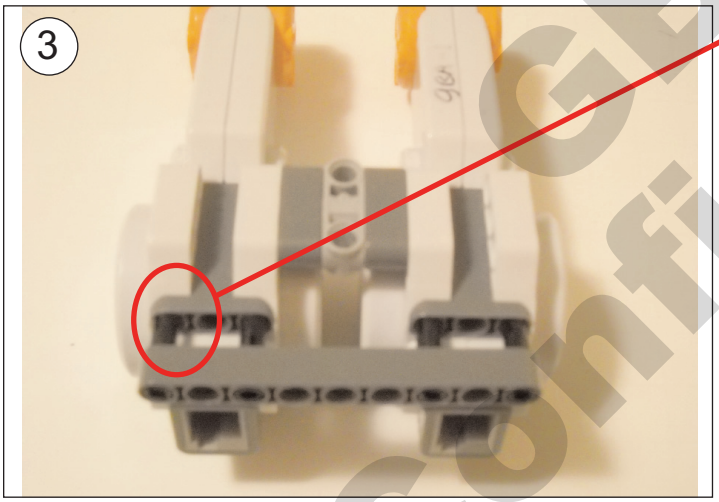
3. Turn base around (orange circle facing away from you) and add technic beams to both motors

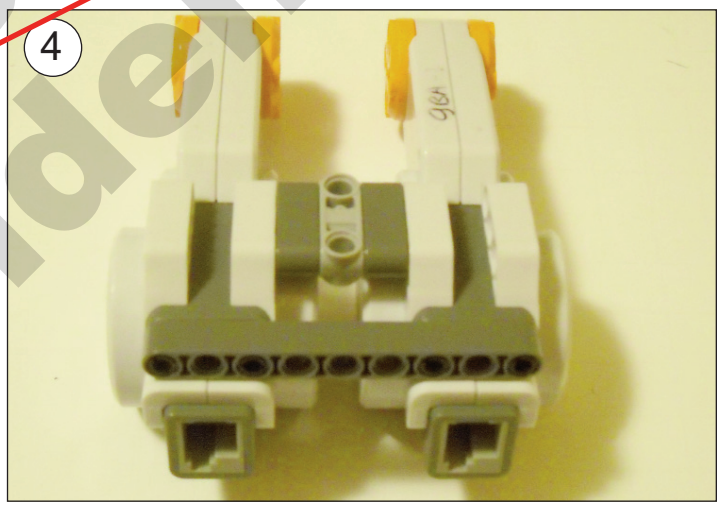
Parts Needed:

Base
4x Short Friction Pins
1x technic beam (#9)





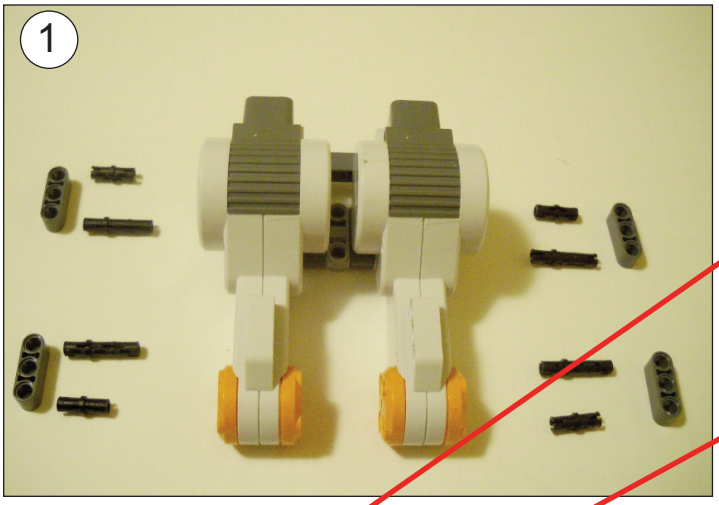


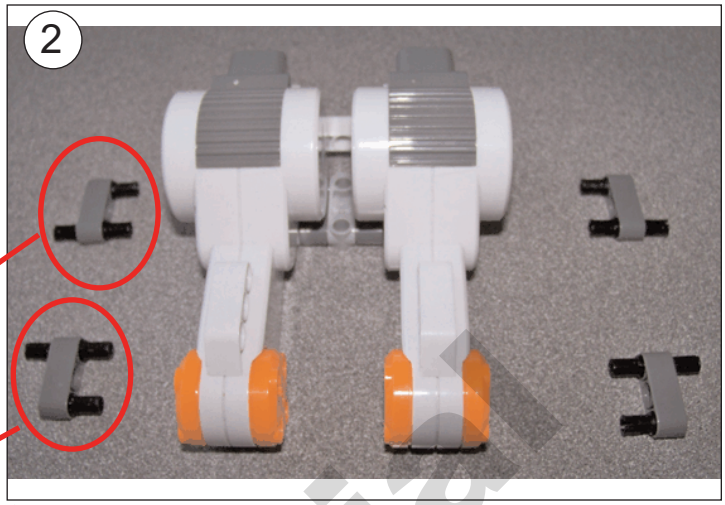


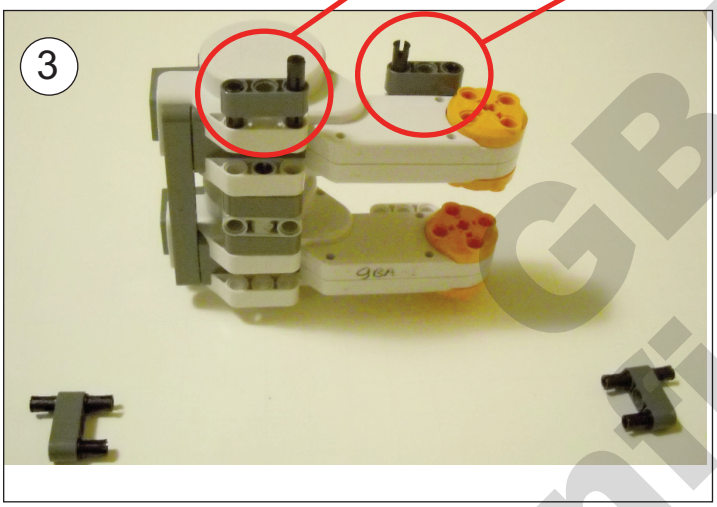
continued on next page

Module 3: Building the Base Robot *continued*

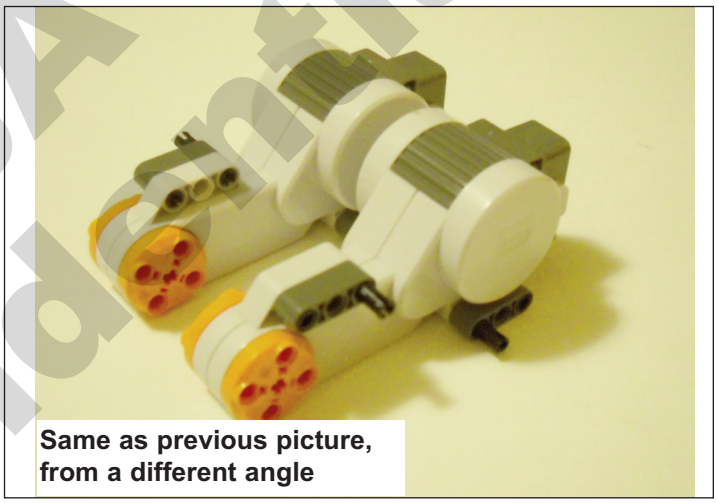
4. Turn upside down; add four small (#3) technic beams

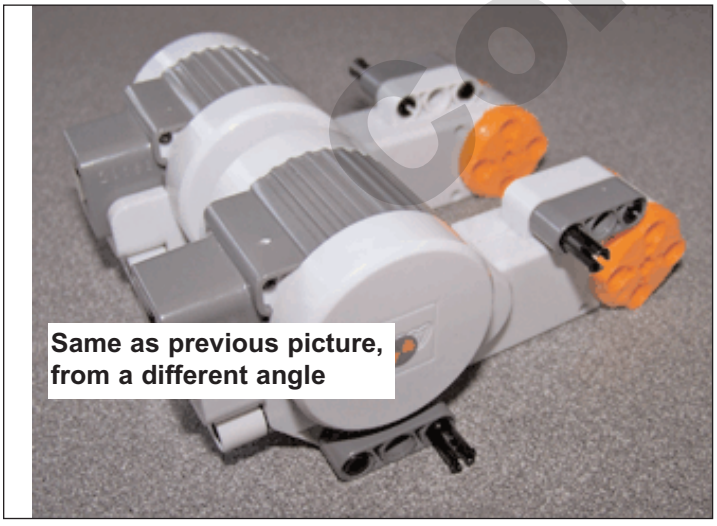


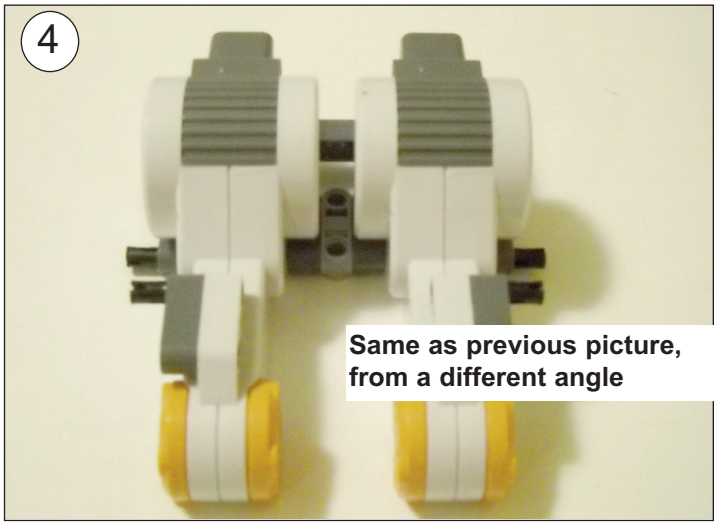




Put remaining tech beams on opposite side of base



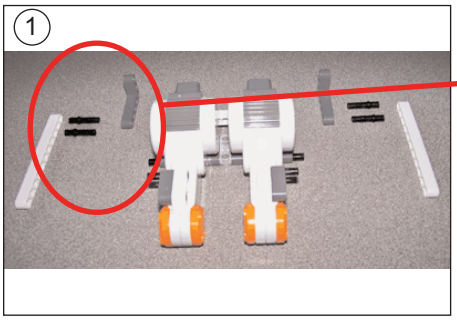


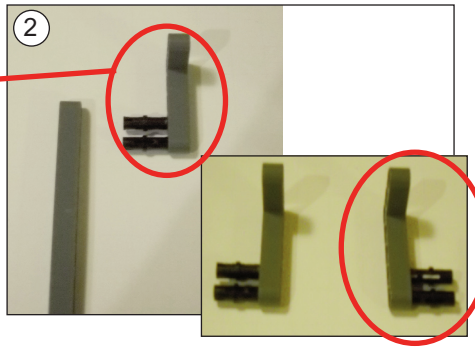


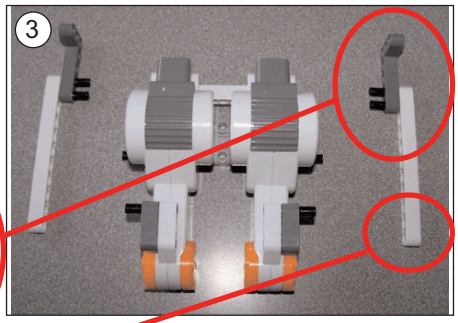
continued on next page

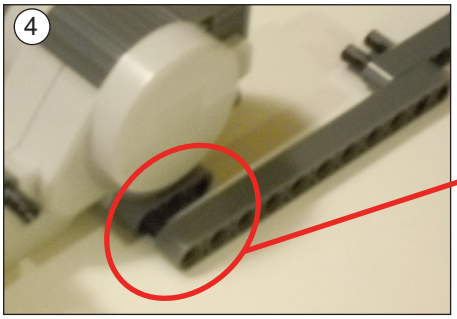
Module 3: Building the Base Robot *continued*

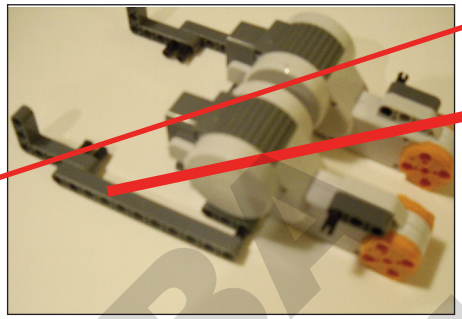
5. Add support structures to sides of base:



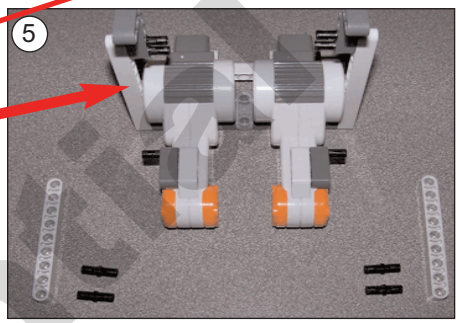






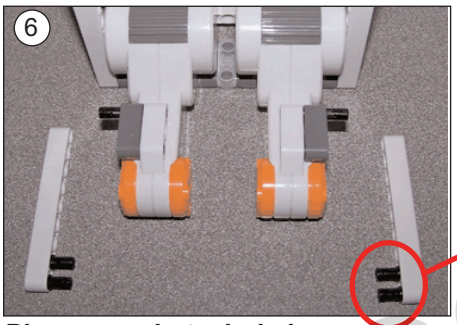


Same as previous picture, from a different angle

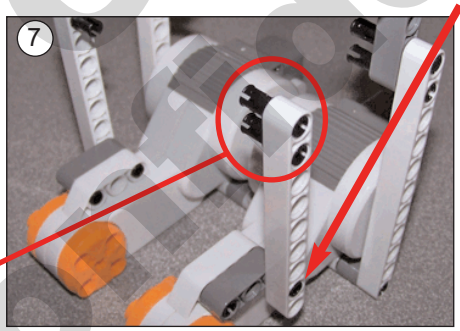


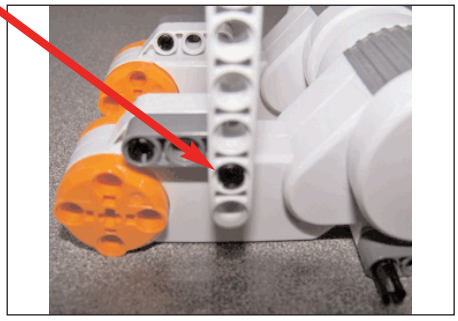
Swivel support structure so it sits vertically (90 degrees)

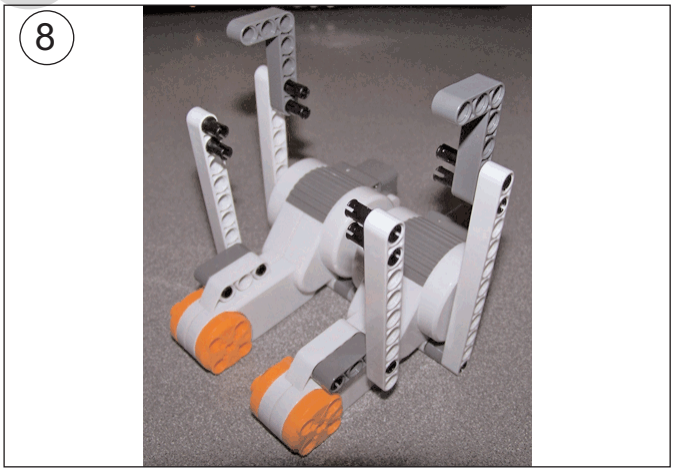
Connect to **second hole from bottom** on technic beam



Place pegs in technic beams, then hold technic beam vertically and attach



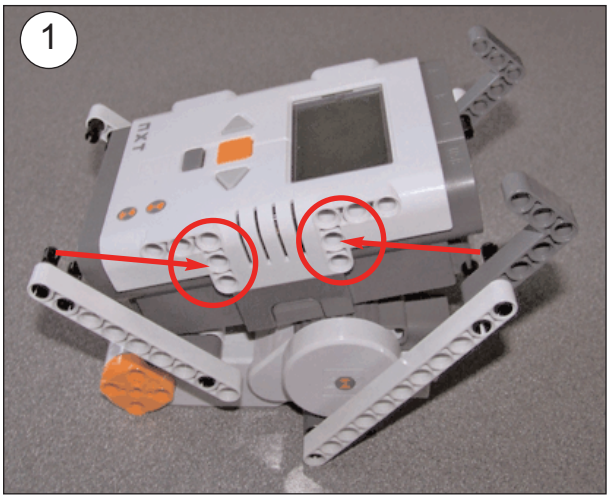


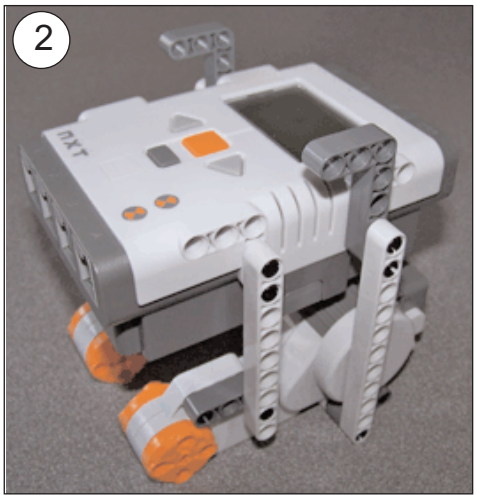


continued on next page

Module 3: Building the Base Robot *continued*

6. Spread supports, hold brick in position and connect supports to brick on both sides:





How to Remove Connecting Pins

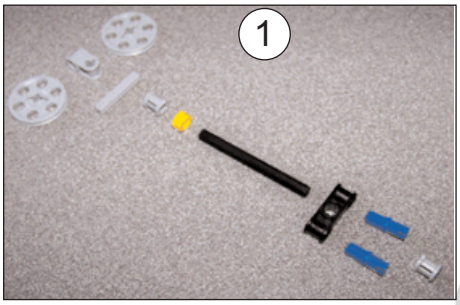
It can sometimes be difficult to remove connecting pins. Either use another pin to push through, or use pliers (gently).

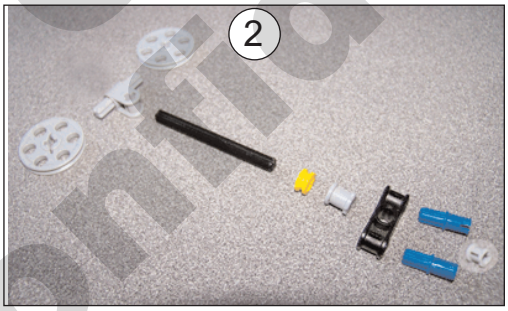
Note: Be sure to connect two pins, sticking out from the tech beams, into the bottom holes located on the side of the brick.

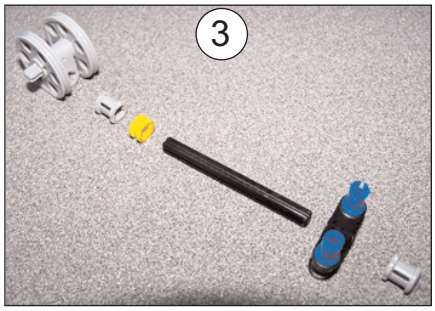
Note: Make sure the NXT brick port Number #1-4 is facing towards the orange section of the motors. Position brick so LCD screen is furthest away from orange end of the motors.

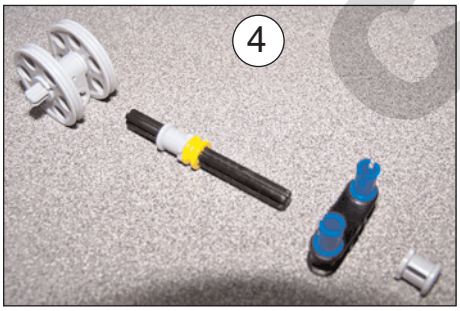
7. Build the back wheel

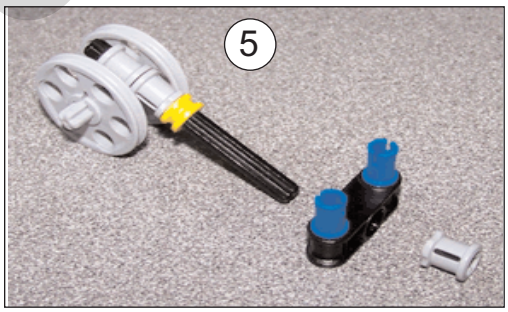
Parts Needed:	1x black 90 degree joint (length 3)	1x black axles (length 6)
	1x gray 90 degree joint (length 2)	2x Bushings
2x thin wheels	2x axle pins w/friction (blue)	
1x half-bushing	1x gray axle (length 3)	

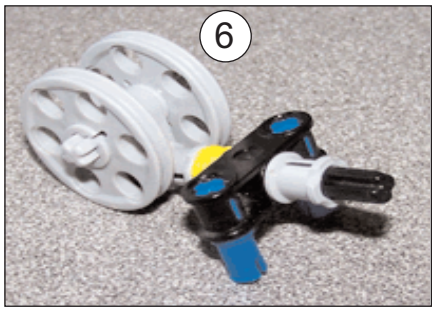






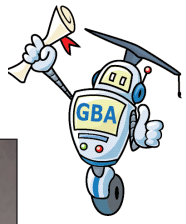






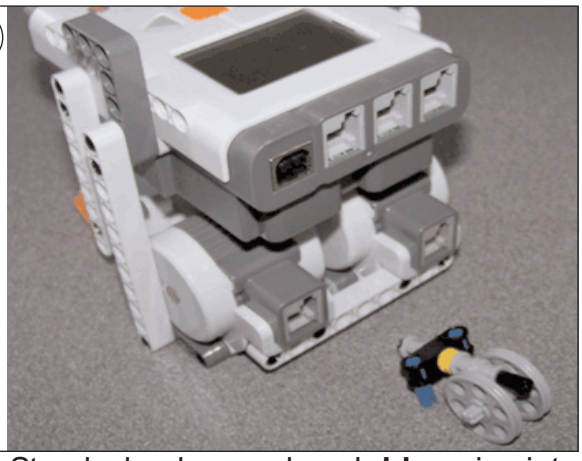
continued on next page

Module 3: Building the Base Robot ***continued***



8. Connect the back wheel to the technic beam that joins the two motors:

1



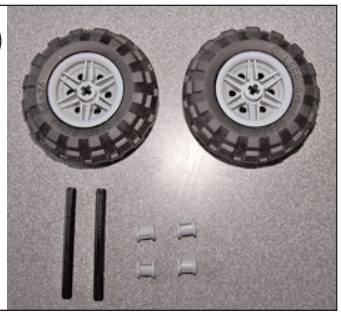
2

Stand wheel up and push **blue** pins into technic beam between two motors

9. Build the front wheels

Parts needed:
2x Rims
2x Black axles (length 6)
2x Tires
4x Bushings

1



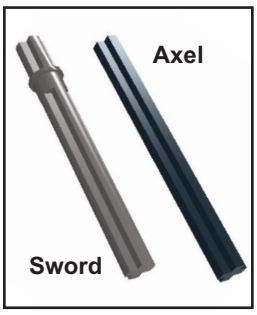
Place Rims inside of Tires.

2

Place Bushings two-thirds of the way on Black axles

3

Push the shorter visible end of axles into Rims



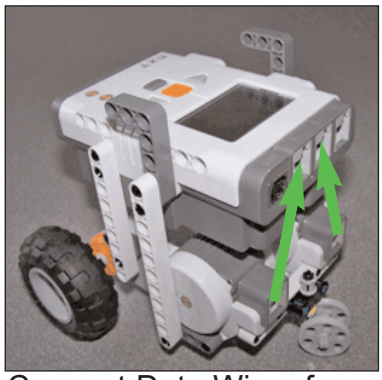
Note: The Difference between axles and swords is related to the "hilt" or piece that prevents the sword from sliding all the way through. The sword provides an accurate spacing between the wheel and motor. However if you're stuck using an axle just be sure the tire does not grind against the motor or technic beam. If it does grind against the robot, flip the tire outward or inward.

continued on next page

Module 3: Building the Base Robot *continued*

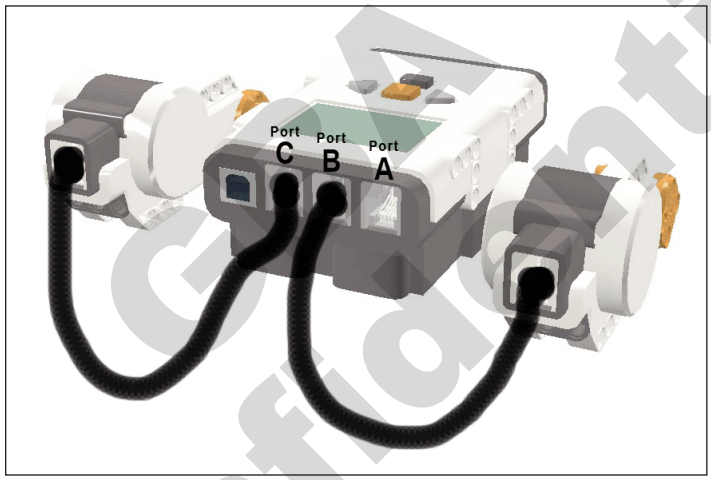
10. Connect the front wheels:

Push tire axles into motors



Connect Data Wires from Ports B and C to Motors

Note: After building the robot, be sure to connect the Motors to the correct ports. Port B will connect to the left hand side Motor. Port C will connect to the right hand side Motor.



END

Module 4: Introduction to On-Brick Programming

OVERVIEW

The robot can be given a few simple commands by programming directly on the NXT brick. After this lesson, all programming will be done using the NXT-G software on the computer, since this form of programming is too limited for anything beyond the most basic movement. However the simple on-brick programming is a great way to get the students to have their robots start performing basic actions.

ABOUT ON-BRICK PROGRAMMING:

On-brick programming is a quick method to allow students to test out the robot without the use of the NXT-G software. Though it is fast and easy to use; it's also limits the actions you can perform with your robot. As an instructor, note that you can only program two move actions, and two sensor actions (although we use the "Empty" option in this module, as the sensors are not yet connected) before looping the program or making it stop.

HOW TO FIX AN ON-BRICK PROGRAM:

If a mistake is made during programming, you can return to the previous step by pressing the gray button underneath the orange button.

BUILD:

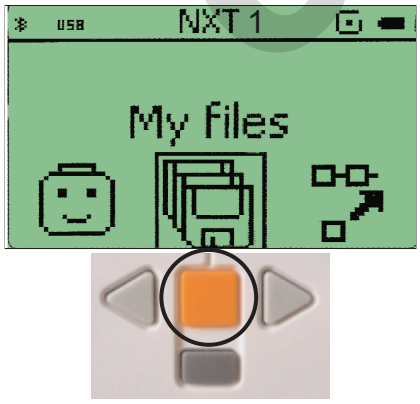
Check that cables have been attached from motors to their appropriate B and C ports on

PROGRAM:

There is no program for this module.

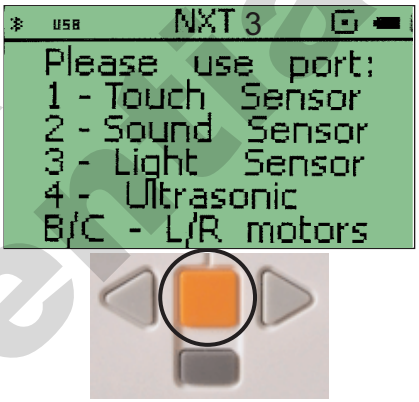
TEST

1. **Press the orange button** to turn the brick on.

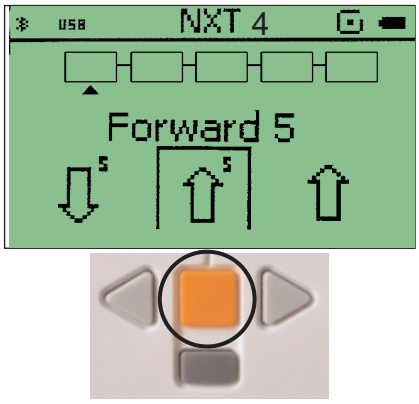


2. Press the **gray right button once** to scroll to the **NXT Program** menu.

3. Press the **orange button** to bring up a **list of the ports** used in this mode.



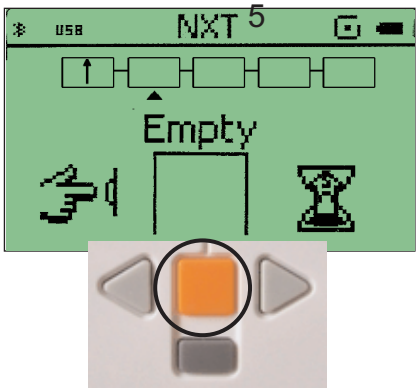
4. Press the **orange button** again to bring up the programming screen. **"Forward 5"** will be selected.



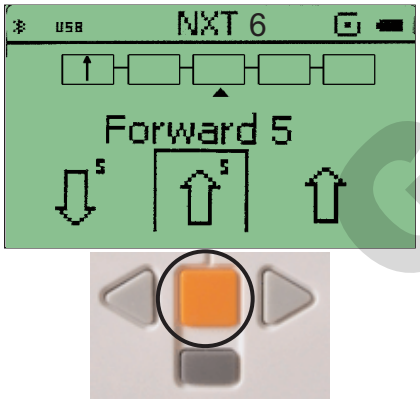
continued on next page

Module 4: Introduction to On-Brick Programming *continued from previous page*

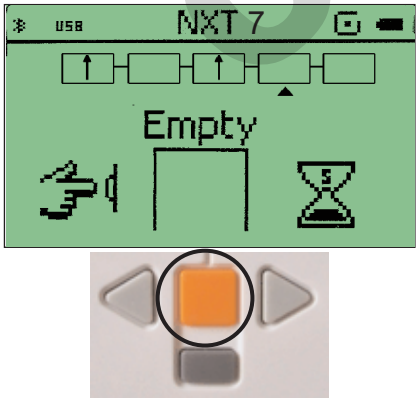
5. Press the **right arrow once** to select "Forward", then press the **orange button once** to set it as the first programming block. The next block will automatically be highlighted and **"Empty"** will automatically be selected.



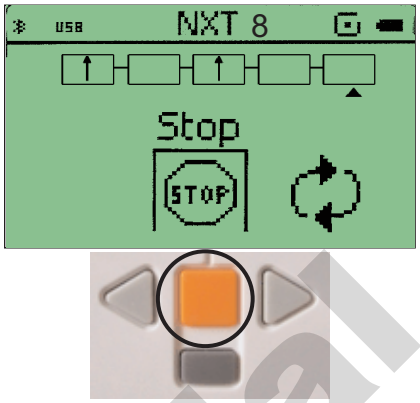
6. With "Empty" selected, press the **orange button once** again to set it as the second programming block. The next block will automatically be highlighted and **"Forward 5"** will automatically be selected.



7. Press the **right arrow once** to select "Forward", then **press the orange button once** to set it as the third programming block. The next block will automatically be highlighted and **"Empty"** will automatically be selected.



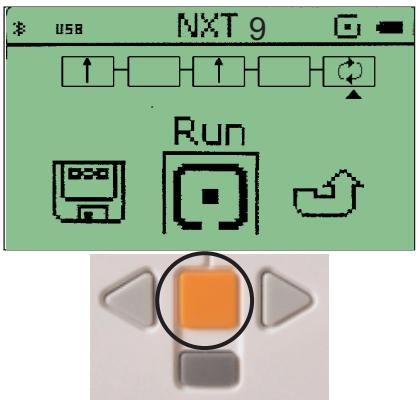
8. With "Empty" selected, press the **orange button once again** to set it as the fourth programming block. The next block will automatically be highlighted and **"Stop"** will automatically be selected.



WHAT IS A PROGRAMMING LOOP?

A programming loop allows the program to continue after it has completed the actions programmed on it. Programming loops have your robot repeat it's programming indefinitely, until the user stops it. For future modules, programming loops will become standard, as they allow for more advanced behaviors.

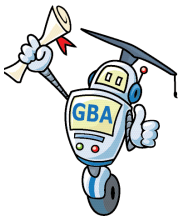
9. Press the **gray right button** once to select **"Loop"**, then press the **orange button once** to set is as the last programming block. The programs run screen will now be shown, and "Run" will be automatically selected.



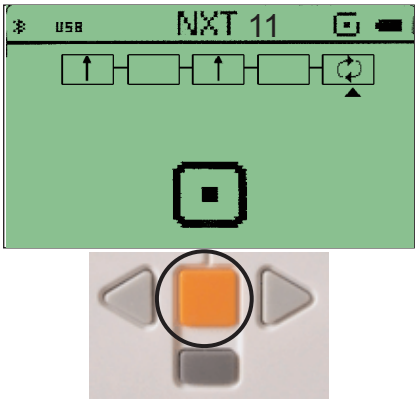
10. Place the robot on a flat surface, with room to move forward.

continued on next page

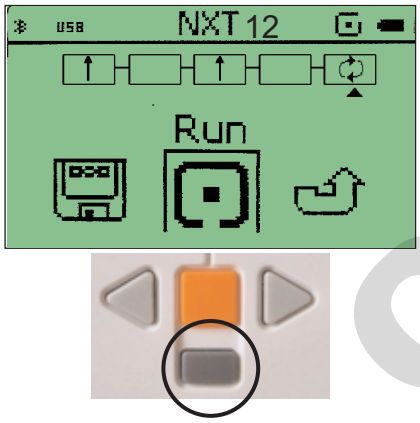
Module 4: Introduction to On-Brick Programming *continued from previous page*



11. With "Run" selected, press the orange button once to run the program. The robot should move forward continuously.



12. Press the gray button once to stop the robot.



END 

Correct connection for the motors:

Make sure that the data wires are connected from port B to the motor on the left side, and from port C to the motor on the right side if the robot is facing you. Make sure that the data wires are not crossed.

Why isn't this working?

If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The proper program is selected, and the brick is currently running it.

Module 5: Introduction to NXT-G Programming

OVERVIEW:

In this module the students will learn the basics of using the NXT-G software to program the robot. Though this software allows for more advanced behavior than the on-brick programming, the first program will be simple and have the robot just move forward continuously.

HOW THE PROGRAM WORKS:

The robot will move forward continuously in a stuttering motion, until the program is stopped by the gray button.

PROGRAM:

1. Launch the NXT-G software.
2. A loading screen will show then the main menu will show. If the robo center is shown, hit the small orange 'x' to close it.
3. You are ready to start a new program; click on the "Go" button on the "Start New Program" tab. The screen will change to the programming grid.
4. Working in the Common palette, click and drag a loop block onto the sequence beam. The beam will extend to include the block.
5. Click and drag a move block from the Common palette onto the sequence beam, inside the loop. The loop will expand to include the move block inside of it.
6. Go to "File" then "Save as"; the "Save as" screen will show. Click on browse to select a destination and name for the file; click save when done. Do not exit out of the software, or close the file.

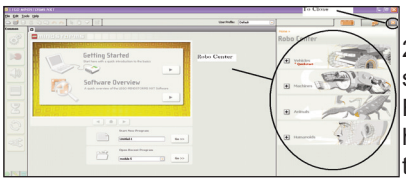
continued on next page

WHY ISN'T THIS WORKING?

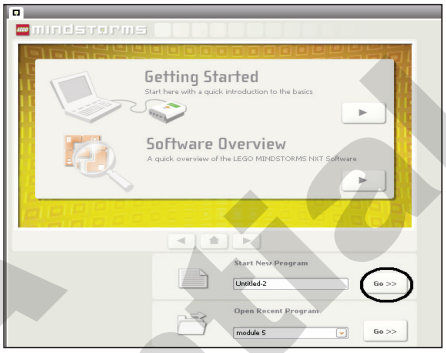
If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The proper program is selected, and the brick is currently running it.
5. Make sure that the commands are on the sequence beam, and not grayed out.
6. Make sure that no options were changed on the move block or on the loop.
7. Make sure that the move block is inside of the loop block.
8. Re-download the program to make sure that the newest version of the program is running.
9. If all else fails, remake the program to make sure that all the steps are correct.

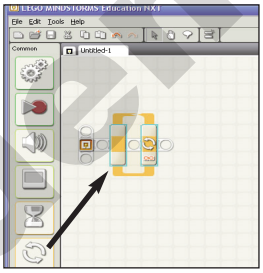
1. Launch the NXT-G Software



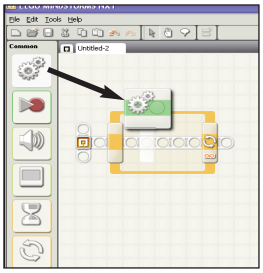
2. Wait for Loading screen to begin. If Robo Center is open, hit the small orange x to close the window.



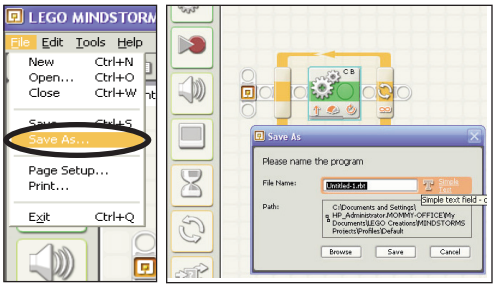
3. Click on the GO button on the Start New Program tab.



3,4. The main programming grid appears. Drag a loop block onto grid.



5. Click and drag a move block inside the loop.



6. File, Save As... to save the student file.

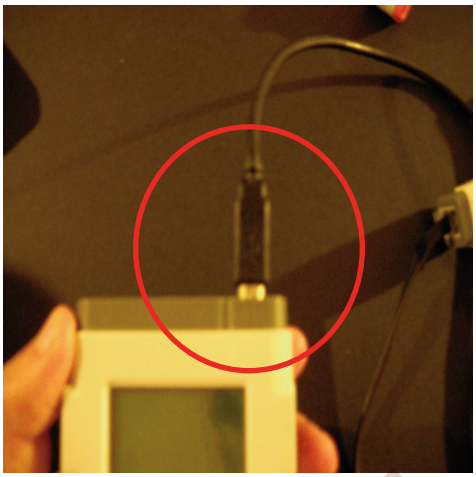
continued on next page

Module 5: Introduction to NXT-G Programming *continued from previous page*

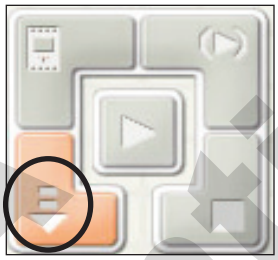
TEST:

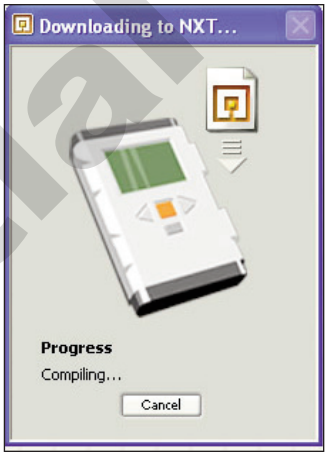
1. Connect the robot to the computer using a USB cable.
2. Press the orange button to turn the robot on.
3. With the program still open in the NXT-G software, click on the download button. A status indicator will appear on the computer screen letting you know how the download is progressing. When the download is finished the indicator will disappear and the robot will beep. Once this happens remove the USB cable from the robot.
4. "My files" should now be selected on the robot. Press the orange button to go into this menu; "Software files" should automatically be selected.
5. With "Software files" selected press the orange button again to browse your files. The newest file downloaded will automatically be selected. If the appropriate file is not selected, use the left and right arrows to select it.

continued on next page

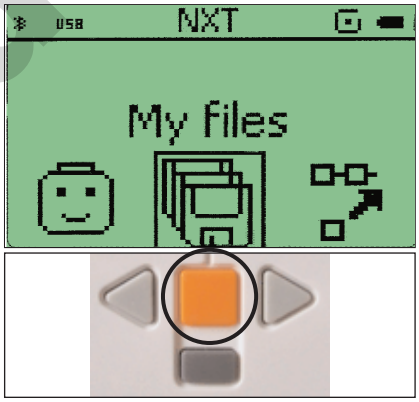


1,2. Connect USB cable, press orange button.

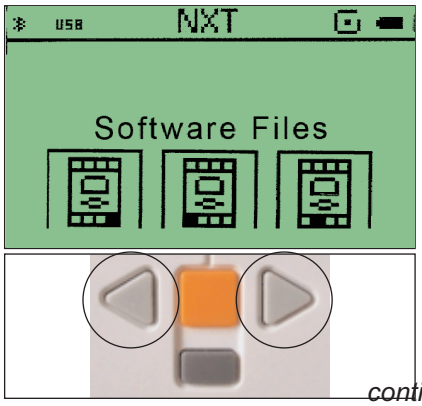




3. Click on the download button. Look for confirmation on the computer screen to see if program has downloaded.



4. My Files selected on the screen.



5. Press the orange button to get to the Software files menu, then use right or left arrows to navigate to your specific file.

continued on next page

A NOTE ON USB CABLES:

Depending on their age, students should be familiar with USB cables, and will easily be able to connect them to the robots. If there are any difficulties, remember that the both sides of the cable are only designed to fit in one direction.

WHAT IS A SEQUENCE BEAM?

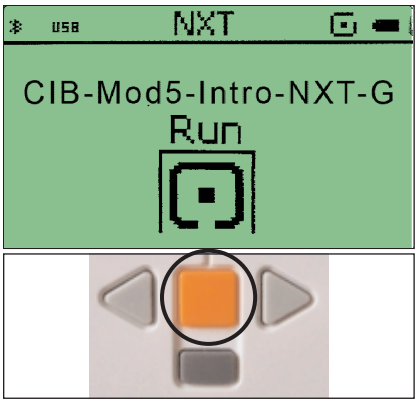
The sequence beam is a visual tool to help the user of the NXT-G software visualizes the robot's programming. The robot will start at the "Start block" on the left side and follow the sequence beam in the order it comes across them. If a block is not on the sequence beam, it will not be performed.

A NOTE ABOUT LOOP BLOCKS:

Give the program a chance to recognize that the user wants to place a block inside of it before the block is released. Emphasize this fact to the students, as most of the time a "click and drag" command can be completed as soon as the object is in the appropriate place. In this case the student should wait for the loop to expand.

Module 5: Introduction to NXT-G Programming *continued from previous page*

6. With the correct file selected, press the orange button to go to the run screen for this program. Place the robot on a flat surface (with area in front of it to move), and press the orange button to run the program. The robot should move forward in a straight line.
7. Press the gray button on the robot to stop the program when done.



END

AN ALTERNATIVE ACTIVITY:
To extend this activity and create a goal for the students, you could have a "Robot Race". Once all of the robots are programmed place two parallel lines of tape a few meters away from each other. Have half the students line up with their robots on one line facing the other, with the robot on the "Run" screen for their program. The other half should line up on the other line ready to catch the robots and turn the program off. Once all of the robots are lined up, have the students start the robots at the same time. Once the race is over, you can have the students switch roles and race again.

What are the options at the bottom for?

Whenever a programming block is selected, the options for that block appear at the bottom. In this program we use the default options for all the blocks, but in later modules we will use these to manipulate these to alter the robot's behavior.

A note about saving:

It is worth repeating that the file name should have information on both the author and what the program does to make it easy to identify. (Details are located in the Programming Overview, page 128.)

Programming Overview:

There is an in-depth overview of this program on page 128. It contains common pitfalls and troubleshooting, as well as detailed descriptions of the programming blocks used in these Modules.

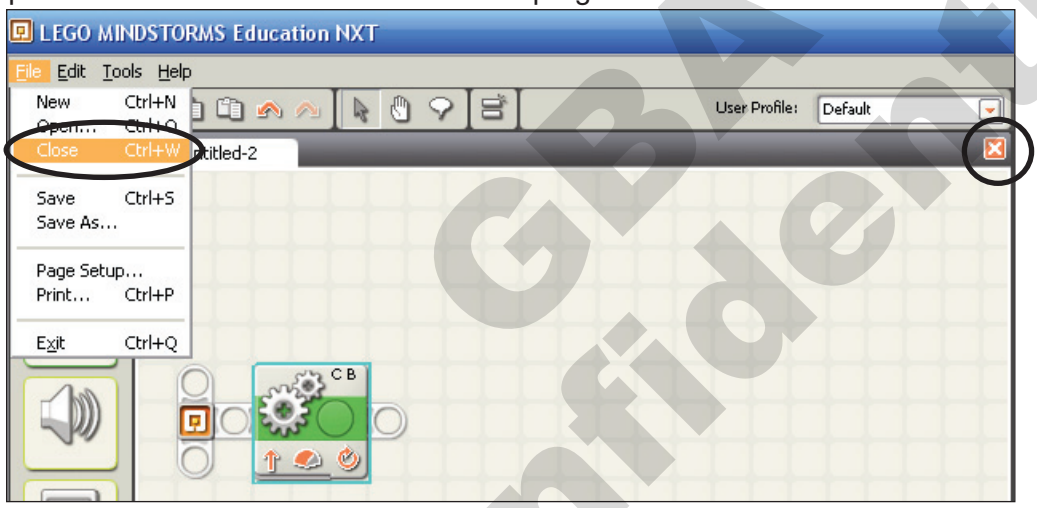
Module 6: Program Forward and Backward Movement

OVERVIEW:
In this module the students will expand upon the knowledge they gained in the previous module by learning how to change the options for a programming block. The objective is to have the robot move forward a set distance, then return to the starting point.

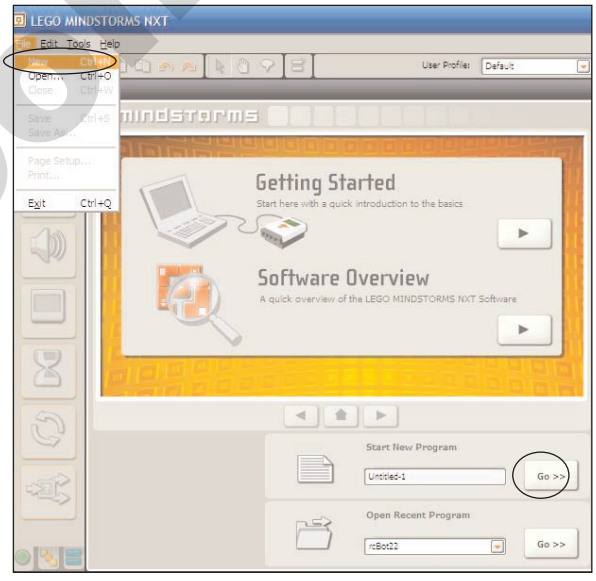
HOW THE PROGRAM WORKS:
The robot will move forward for five wheel rotations, then immediately move backwards for five wheel rotations. The robot will then stop.

BUILD:
None

PROGRAMMING STEPS:
1. If the NXT-G software is not already open, launch it and exit the robo center. If the program is running, close out of any open files. Use File>Close, or press on the Red X to close the current program.



2. You are now ready to start a new program. Click on the "Go" button on the "Start New Program" tab or use File>New. The screen will change to the programming grid.



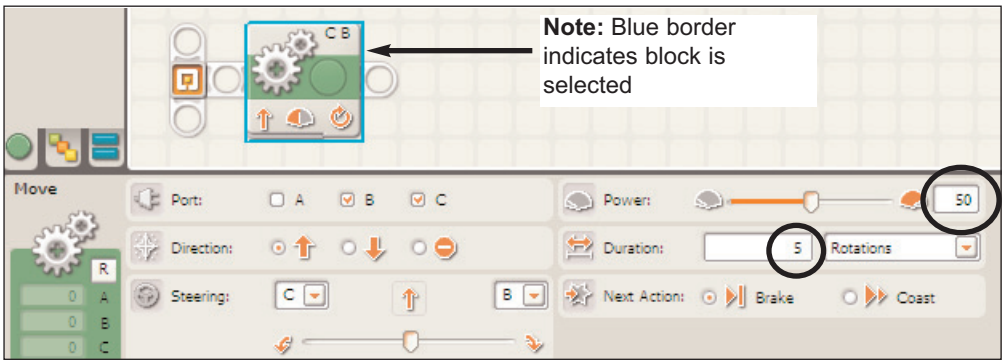
PROGRAMMING BLOCKS AND OPTIONS

When you change the options for a block, you may notice icons on the power block change. These icons show the selected options for that block and are discussed in more detail in the program overview section. These changes will not always show on the block and may show up after deselecting. To be certain double check the options located at the bottom.

continued on next page

Module 6: Program Forward and Backward Movement *continued from previous page*

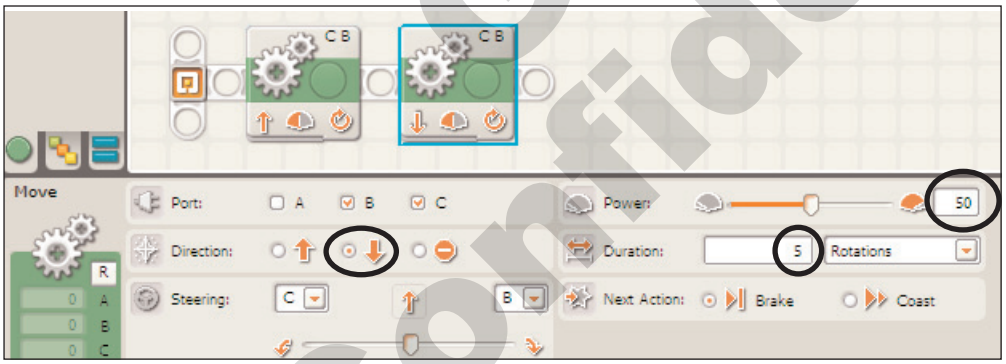
3. Working with the Common Palette, click and drag a move block onto the sequence beam. The beam will extend to include the block; click on the move block to select it. (A blue border will be around it) The move block's option palette will be shown on the bottom of the screen. Set the option for power to 50 and the duration to 5 rotations. This will make the robot go for a set number of wheel rotations slightly slower than normal, then stop.



Port: B C
Direction: Forward
Steering: Straight

Power: 50
Duration: 5 Rotations
Next Action: Brake

4. Drag a second move block onto the sequence beam after the first one. Make sure that the second block is selected, and then set the power to 50, the duration to 5 rotations, and the direction to backwards. This will make the robot go backwards a set number of wheel rotations slightly slower than normal, then stop. 5. Save the program using "save as" just like in the previous module. Do not exit out of the software, or close the file.



Port: B C
Direction: Backward
Steering: Straight

Power: 50
Duration: 5 Rotations
Next Action: Brake

TEST:

1. Connect the robot to the computer using a USB cable, and make sure that the robot is turned on. (picture of the robot with a USB cable connected to the USB port)
2. With the program still open in the NXT-G software, click on the download button. A status indicator will appear on the computer screen letting you know how the download is progressing. When the download is finished the indicator will disappear and the robot will beep. Once this happens remove the USB cable from the robot.

Note on the Duration Option:

When you change the units in this option, the program will automatically convert the value for you, meaning if you change 2 rotations to degrees, the program will change the value to 720 degrees. Because of this we recommend changing the units and then the value.

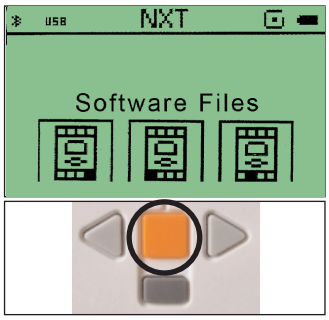
Note on the Power Option:

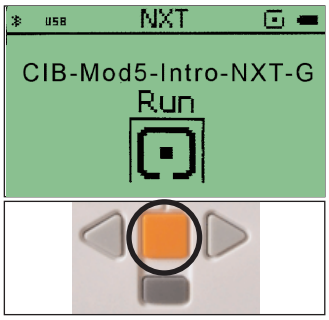
You can manipulate the power of the motors in two ways: You can modify the bar on the option, or directly change the number next to it. We usually recommend changing the number, since it is more accurate.

continued on next page

Module 6: Program Forward and Backward Movement *continued from previous page*

3. Navigate to the software files menu on the robot, and find the new program. Press the orange button to enter the run menu for the program.





4. Place the robot on a flat surface with area in front of it to move. Press the orange button to run the program. The robot should move forward in a straight line then move backwards in a straight line.

5. Press the gray button on the robot to stop the program when done.

END

AN ALTERNATIVE ACTIVITY:
To extend this activity you could make an accuracy challenge. Place two parallel lines of tape a few meters away from each other. Have the students use trial and error to get their robot to move to the line, stop on it, and then back up. Emphasize that this is a competition for accuracy and not speed, and that the robot will be more accurate at lower powers. (Remember "Power" is located as one of the options of the "Move" block.)

APPROPRIATE WORKSHEETS:
Worksheet four allows students to experiment with the options menu for the move blocks, and teaches data taking and estimation. Rulers will need to be provided.

WHY ISN'T THIS WORKING?

If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The proper program is selected, and the brick is currently running it.
5. Make sure that the commands are on the sequence beam, and not grayed out.
6. Make sure that the direction, power and duration settings are correct.
7. Re-download the program to make sure that the newest version of the program is running.
8. If all else fails, remake the program to make sure that all the steps are correct.

HOW TO NAVIGATE THROUGH THE ROBOT'S MENUS

To back up out of a menu, press the gray button beneath the orange button on the robot. This is useful if the robot has recently run a program, and you want to switch to a different program.

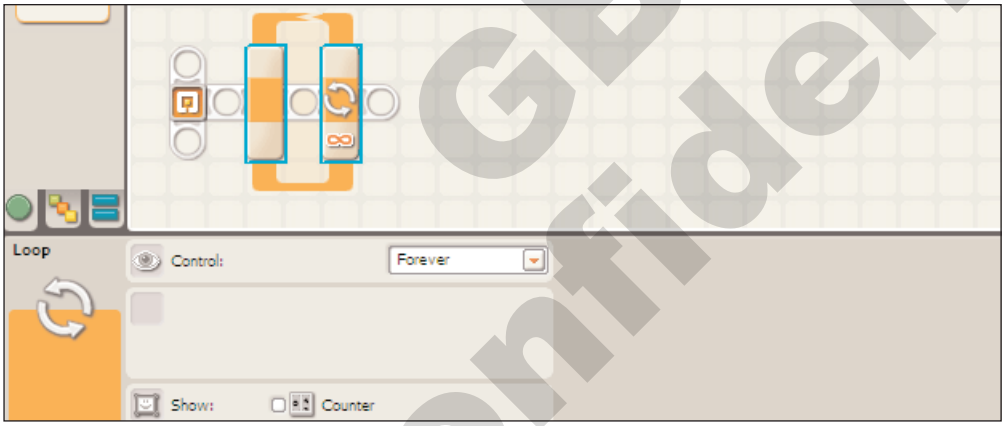
Module 7: Program Movement with Turns

OVERVIEW:
In this module the students will program their robot to repeatedly move forward a set distance, and then make a 180 degree turn. The student will gain experience by making this slightly more advanced program. This will introduce the student to how the robot makes turns.

HOW THE PROGRAM WORKS:
The robot will move forward five wheel rotations, make a clockwise 180 degrees turn, and then continuously repeat the program.

BUILD:
None

PROGRAMMING STEPS:
1. If the NXT-G software is not already open, launch it and exit the robo center. If the program is running, close out of any open files.
2. You are now ready to start a new program. Click on the "Go" button on the "Start New Program" tab. The screen will change to the programming grid.
3. Click and drag a loop block from the common palette onto the sequence beam. The beam will extend to include the block.



Control: Forever

A NOTE ON TURNING:

The robot's turning radius can be affected by many different variables, including the charge of the battery, the surface the robot is on, and numerous other things. Because of this, it may be difficult to turn accurately, and the values given here may need to be modified by trial and error to suit your classroom.

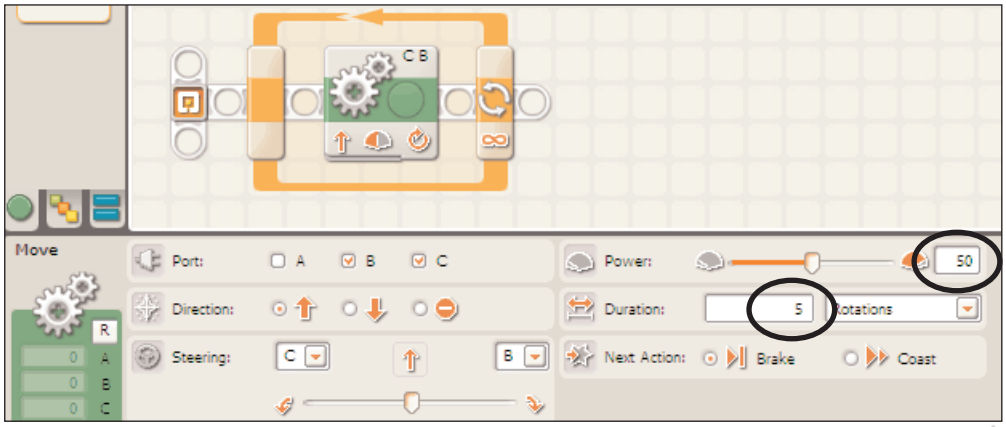
HOW FAR IS A WHEEL ROTATION?

When moving forward with both wheels, one wheel rotation results in approximately 7.5 inches of movement. Since the distance covered is determined by the number of rotations, the speed set by the power isn't relevant. When making turns with two wheels, the degree of turning is based on the number of wheel rotations, as well as many other factors which can make it fairly inaccurate. When only one wheel is turned, one rotation results in a ninety degree turn, and is usually more accurate.

continued on next page

Module 7: Program Movement with Turns *continued from previous page*

4. Click and drag a move block from the common palette onto the sequence beam inside the loop. The loop will expand to include the move block inside of it. With the move block selected, set the power to 50, and the duration to 5 rotations. This will cause the robot to move for 5 rotations at a manageable speed.



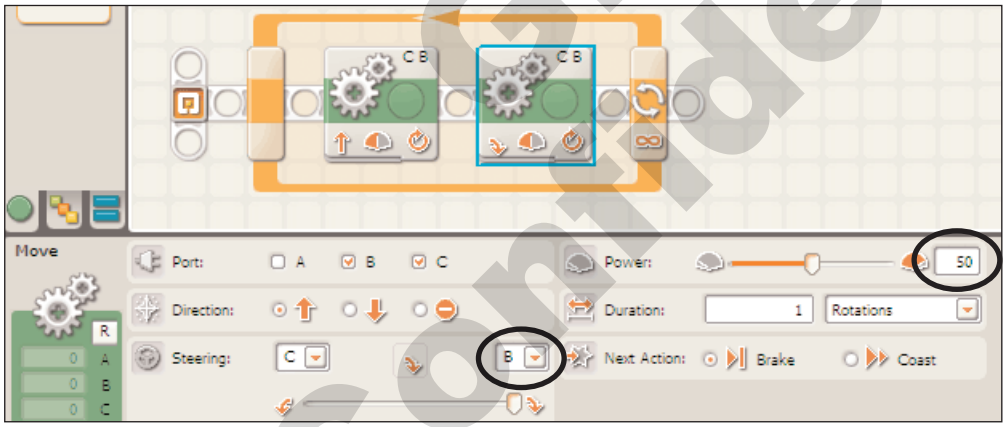
Port: B C
Direction: Forward
Steering: Straight

Power: 50
Duration: 5 Rotations
Next Action: Brake

Different ways to program turns

By changing the steering, direction and ports we can choose to have the robot turn in many different ways. In this module the wheels turn opposite each other. Although controlling each wheel separately, this is the easiest way to turn.

5. Drag a second move block onto the sequence beam after the first one. Make sure that the second block is selected, set the power to 50, and move the mark on the steering bar all the way to the right. This will cause the robot to turn around.



Port: B C
Direction: Forward
Steering: Full Right

Power: 50
Duration: 1 Rotation
Next Action: Brake

6. Save the program using "save as" just like in the previous module. Do not exit out of the software, or close the file.

TEST:

1. Connect the robot to the computer using a USB cable, and make sure that the robot is turned on.

continued on next page

Module 7: Program Movement with Turns *continued from previous page*

2. With the program still open in the NXT-G software, click on the download button. A status indicator will appear on the computer screen letting you know how the download is progressing. When the download is finished the indicator will disappear and the robot will beep. Once this happens remove the USB cable from the robot.
3. Navigate to the software files menu on the robot, and find the new program. Press the orange button to enter the run menu for the program.
4. Place the robot on a flat surface with area in front of it to move, and press the orange button to run the program. The robot should move forward in a straight line, then turn around and return.
5. Press the gray button on the robot to stop the program when done.

END

AN ALTERNATIVE ACTIVITY:
To extend this activity you could challenge the students to change the program to move in a rectangle 5 rotations in length. To achieve this student's will need to find a way to modify the turn from 180 degrees to 90 degrees. One way to do this is to change the duration to 0.5 rotations.

WHY ISN'T THIS WORKING?

If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The proper program is selected, and the brick is currently running it.
5. Make sure that the commands are on the sequence beam, and not grayed out.
6. Make sure that the direction, power and duration settings are correct.
7. Make sure that the move block is inside of the loop block.
8. Re-download the program to make sure that the newest version of the program is running.
9. If all else fails, remake the program to make sure that all the steps are correct.

Module 8: Program Robot to Play Sounds

OVERVIEW:

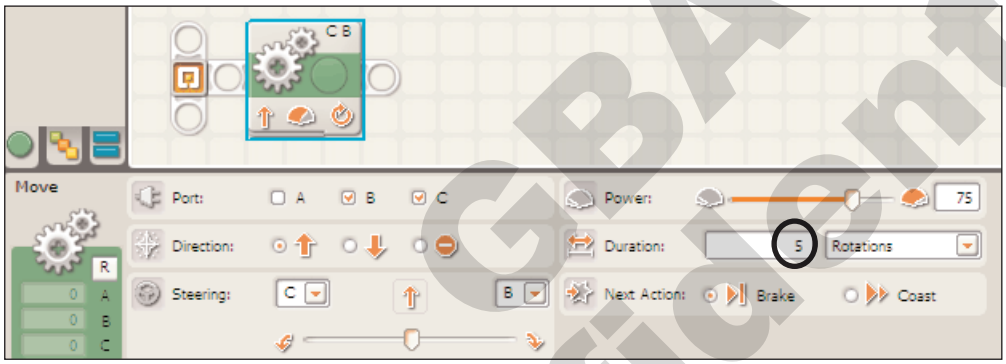
In this module the students will program their robot to play sounds. The robot will move forward, play a sound, spin and then play another sound. Sounds are a fun addition to any program and simple enough that students will be able to add sounds to future programs on their own.

HOW THE PROGRAM WORKS:

The robot will move forward five rotations, play a sound the student selected, spin clockwise, and then play a second student selected sound.

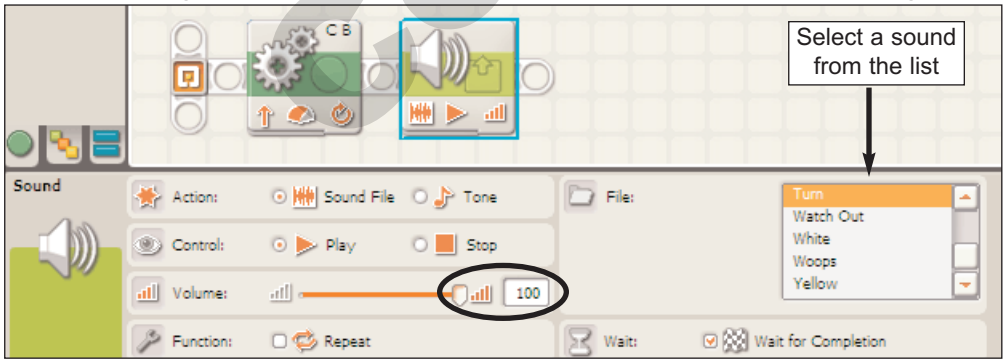
PROGRAMMING STEPS:

1. If the NXT-G software is not already open, launch it and exit the robo center. If the program is running, close out of any open files.
2. You are now ready to start a new program. Click on the "Go" button on the "Start New Program" tab. The screen will change to the programming grid.
3. Click and drag a move block from the common palette onto the sequence beam. The beam will extend to include the block. With the move block selected, set the duration to 5 rotations.



Port: B C
Direction: Forward
Steering: Straight
Power: 75
Duration: 5 Rotations
Next Action: Brake

4. Click and drag a sound block from the common palette onto the sequence beam. Make sure it's to the right of the first move block. With the sound block selected, change the volume to 100, and pick a file from the list on the right.



Action: Sound File
Control: Play
Volume: 100
Function: Unchecked (no repeat)
File: Student to choose any sound
Wait: Wait for Completion

A NOTE ABOUT THE ROBOT'S SPEAKERS:

The speakers on the robots are not loud, and in a room full of kids, it could be almost impossible to hear your robot. Raising the volume to 100 will at least help with this.

A NOTE ON SOUNDS ON THE COMPUTER:

When you are browsing the sound files, the computer will play the sound as a preview. If this isn't happening, make sure that the sound isn't muted on the computer. Note that the sound will only play if a new sound file is selected, not if you click on an already selected file. Some sounds are obnoxious and can create loud distractions; make sure your computer speakers are not turned all the way up.

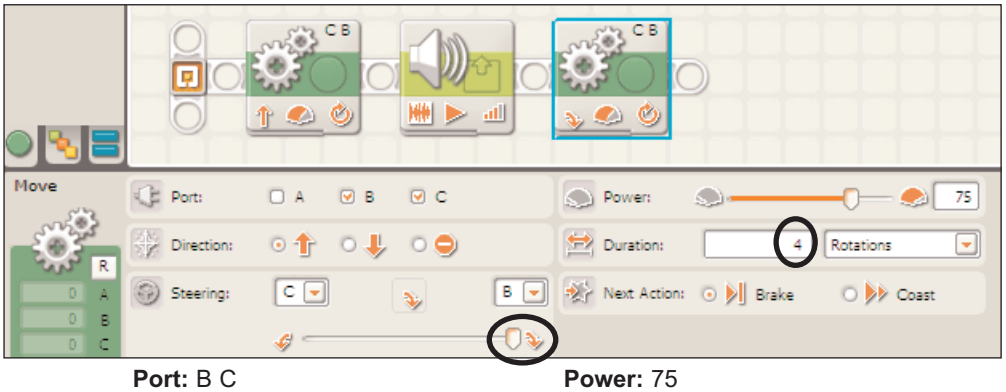
PLAYING A TONE:

If you change the action option from "sound file" to "tone" you can have the robot play a tone like a piano. If you have the robot do several of these in a row, it can play a song

continued on next page

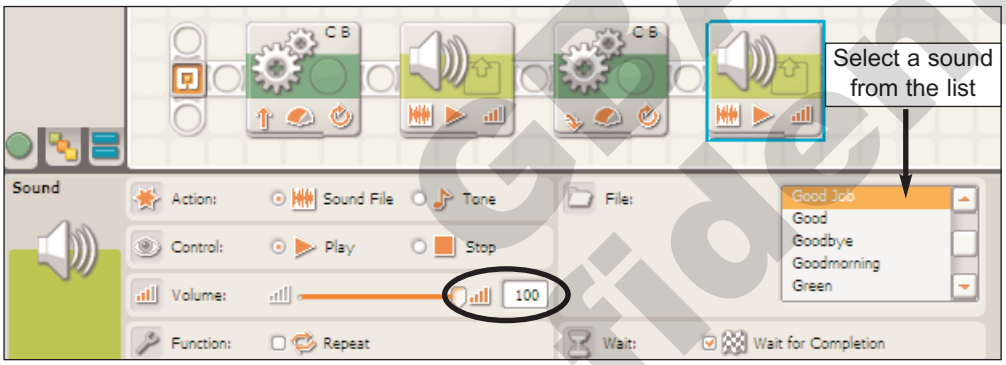
Module 8: Program Robot to Play Sounds *continued from previous page*

5. Click and drag a move block from the common palette onto the sequence beam after the sound block. With the new move block selected, change the duration to 4 rotations and move the steering mark all the way to the right. This will cause the robot to spin in a circle, and imitate a little dance.



Port: B C
Direction: Forward
Steering: Full Right
Power: 75
Duration: 4 Rotations
Next Action: Brake

6. Click and drag a sound block from the common palette onto the sequence beam in front of the second move block. With the sound block selected, change the volume to 100, and pick a file from the list on the right.



Action: Sound File
Control: Play
Volume: 100
Function: Unchecked (no repeat)
File: Student to choose any sound
Wait: Wait for Completion

7. Save the program using "save as" just like in the previous modules. Do not exit out of the software, or close the file.

continued on next page

Module 8: Program Robot to Play Sounds *continued from previous page*

TEST:

1. Connect the robot to the computer using a USB cable, and make sure that the robot is turned on.
2. With the program still open in the NXT-G software, click on the download button. A status indicator will appear on the computer screen letting you know how the download is progressing. When the download is finished the indicator will disappear and the robot will beep. Once this happens remove the USB cable from the robot.
3. Navigate to the software files menu on the robot, and find the new program. Press the orange button to enter the run menu for the program.
4. Place the robot on a flat surface with area in front of it to move. Press the orange button to run the program. The robot should move forward in a straight line, play a sound, spin around, and play another sound.
5. Press the gray button on the robot to stop the program when done.

END 

AN ALTERNATIVE ACTIVITY:

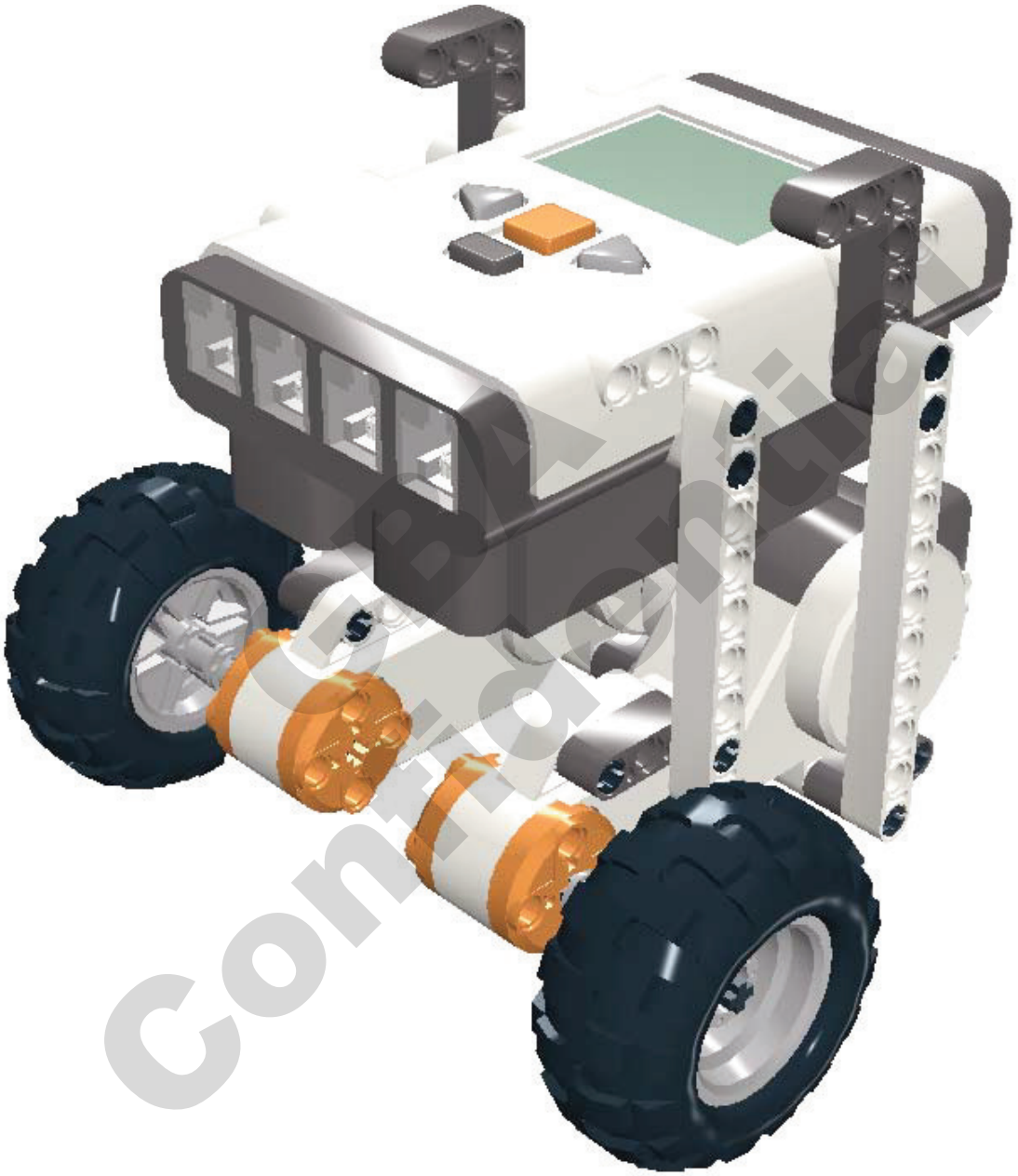
To extend this activity you could have students create any number of victory dances, or use sheet music to try to create a song with the tone tool.

WHY ISN'T THIS WORKING?

If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The proper program is selected, and the brick is currently running it.
5. Make sure that the commands are on the sequence beam, and not grayed out.
6. Make sure that the power, direction, duration on the move blocks are correct.
7. Make sure the only option changed on the sound block is the file that will play.
8. Re-download the program to make sure that the newest version of the program is running.
9. Make sure that the volume on the robot is not muted (under the settings menu of the robot)
10. If all else fails, remake the program to make sure that all the steps are correct.

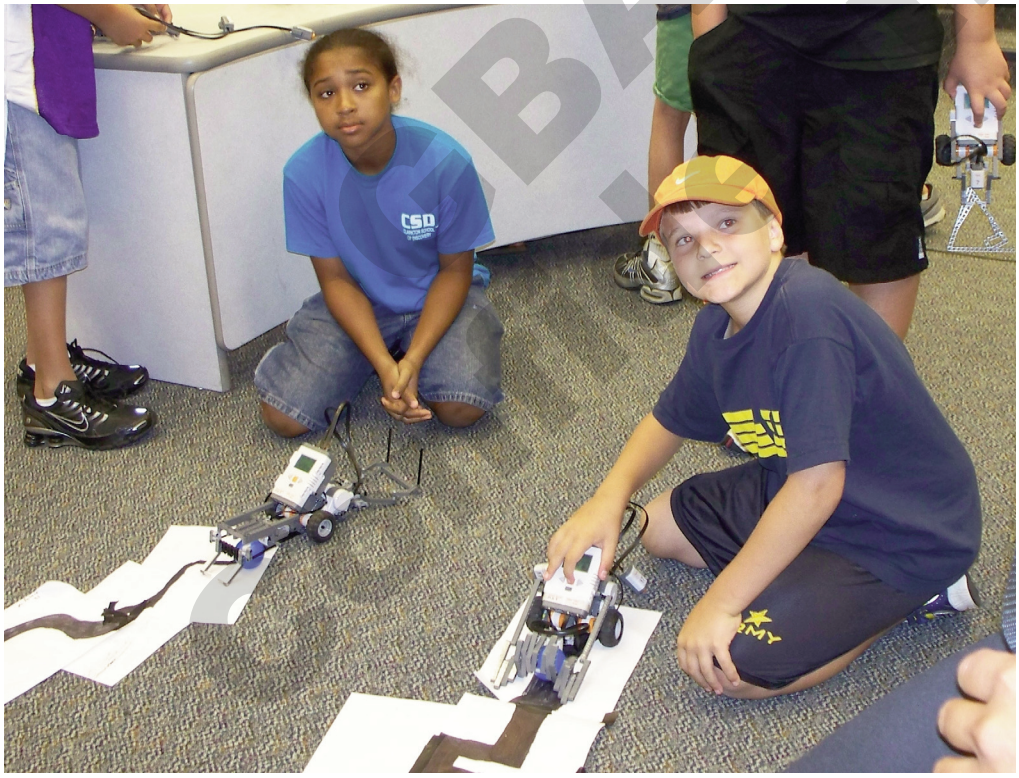
Module 13: Navigating a Maze with the Light Sensor *continued from previous page*



Part II - Using the Sensors

In this section, students program the robot to use the touch, ultrasonic, and reflected light sensors to navigate the robot's environment.

Part 2 Modules



Module 9: Obstacle Avoidance Using the Touch Sensor

OVERVIEW:
There are many uses for the touch sensor. One basic use is to detect when the robot is about to bump into a wall or other object. In this module students will attach the touch sensor to the front of the robot. When the robot hits an object, and the touch sensor is pressed in, the robot will detect the object and act accordingly.

HOW THE PROGRAM WORKS:
The robot moves forward continuously until the touch sensor is pressed; indicating that the robot has touched an object. The robot will then back up, turn, and move forward again until it bumps into another wall or object. This happens repeatedly (loops).

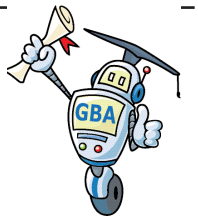
BUILD:
Playfield Materials: Walls, furniture, or objects with some weight (such as hardcover books).

> **Note**: The object that the robot approaches must have sufficient weight or resistance so the touch sensor button is pressed, and the object itself is not moved. Walls, bookshelves, and stacked hardcover books make excellent obstacles.

PLAYFIELD SETUP:
Walls and other objects should be arranged to block the robot at various points at right angles to each other. Objects must be heavier than robot or have sufficient resistance so that touch sensor is pressed. Hardcover books laid flat and stacked or standing (open book) will work well. Boxes of various kinds may also work; including the bins that the robot came in. (Sample arrangements below are not to scale. Spread objects further apart than appears in diagrams.)

continued on next page

Module 9: Obstacle Avoidance Using the Touch Sensor *continued from previous page*



BUILD THE TOUCH SENSOR ATTACHMENT- PART 1

Build the touch sensor attachment and connect it to the front of the basic robot. Follow the diagrams below.

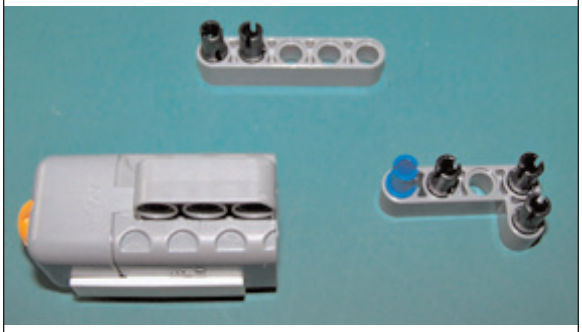
Notes on Part 1:

The Touch Sensor build below is a simple and easy build that can be completed quickly. After connecting the simple attachment, you can go directly to page 47 and program the robot. However, due to the small point of contact for the touch sensor, if the robot touches an object or obstacle on an angle, the touch sensor will not be pressed. To fix this problem and make the resulting build much more effective; you need to add a larger attachment that sits in front of the touch sensor. This will cover a large area, so if the robot approaches an obstacle from an angle, the larger attachment will be uniformly pushed into the touch sensor, ensuring that the touch sensor is pressed and the obstacle is detected. The build for this additional attachment starts in page 43.

Parts List	
1 touch sensor	1 L-beam (length 4 x 2)
1 technic beam (length 5)	1 friction/axle pin (blue)
	5 connector pins

1

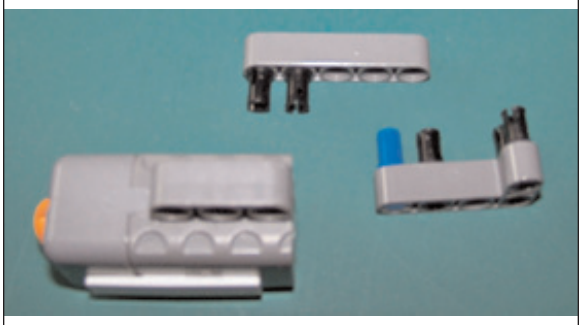
2



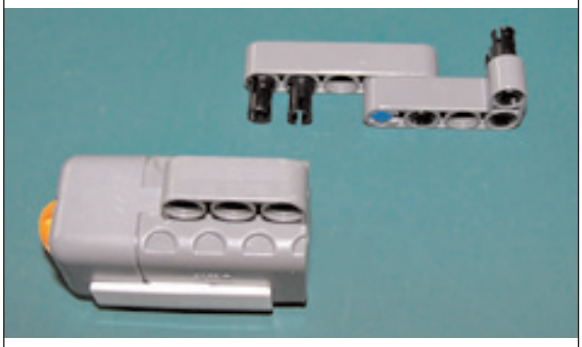
continued on next page

Module 9: Obstacle Avoidance Using the Touch Sensor *continued from previous page*

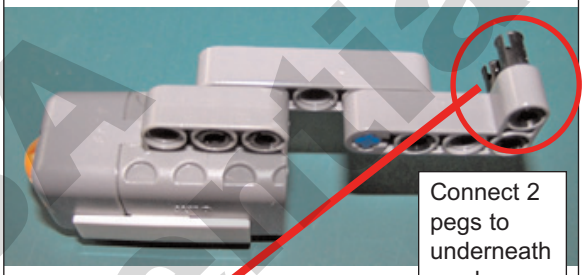
3



4

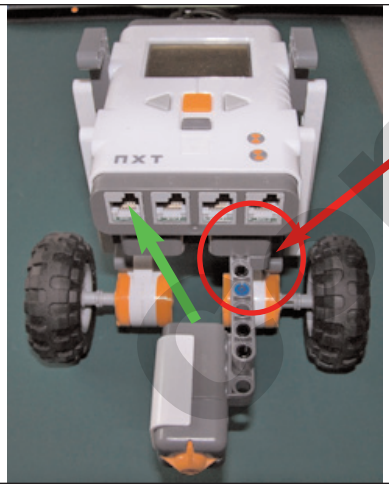


5



Connect 2 pegs to underneath as shown

6

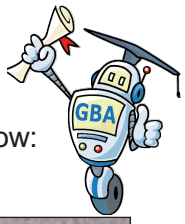


Note: Don't forget to connect the data wire from the touch sensor to Port 1.

Note: This completes the first part of the touch sensor attachment; you can use this basic connection to get started. If you're satisfied with the build then jump straight into programming steps. Otherwise you can use the attachment on the following pages for a more effective and reliable touch sensor.

continued on next page

Module 9: Obstacle Avoidance Using the Touch Sensor *continued from previous page*

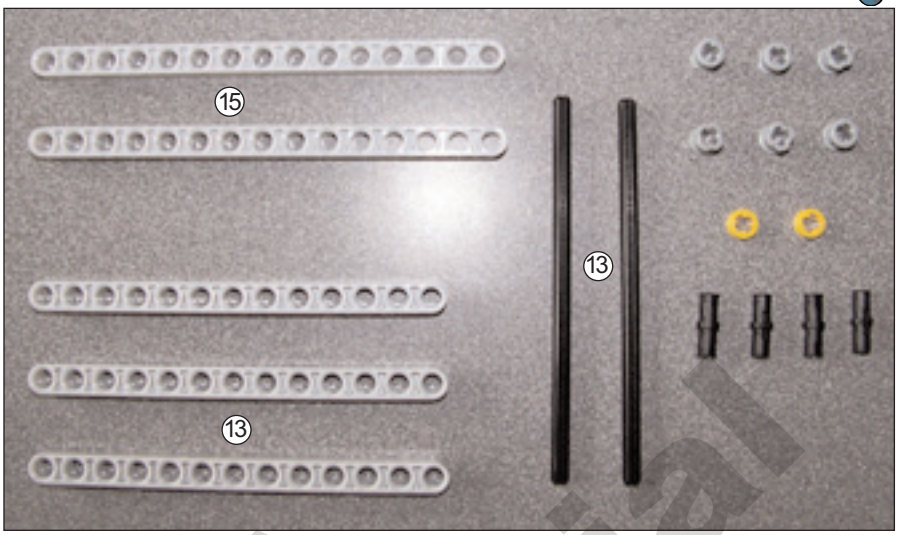


BUILD THE TOUCH SENSOR ATTACHMENT- PART 2

Build the touch sensor attachment and connect it to the front of the robot. Follow the diagrams below:

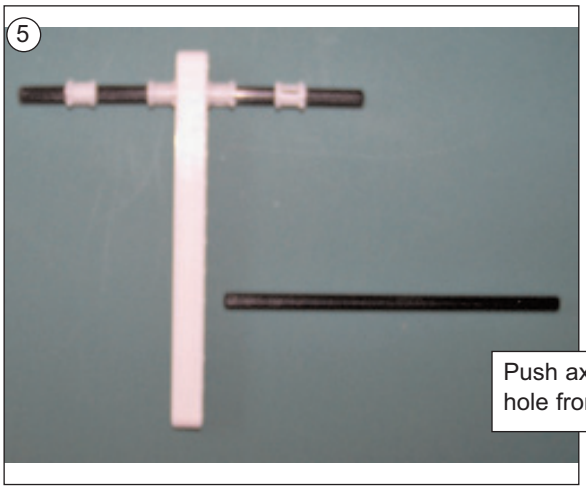
Parts List

- **2** technic beams (length 15)
- **3** technic beams (length 13)
- **2** axles (length 13)
- **6** full bushings
- **2** half bushings
- **4** connector pins

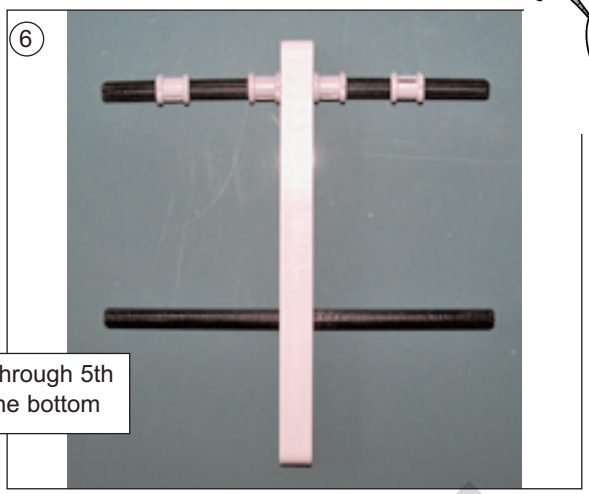


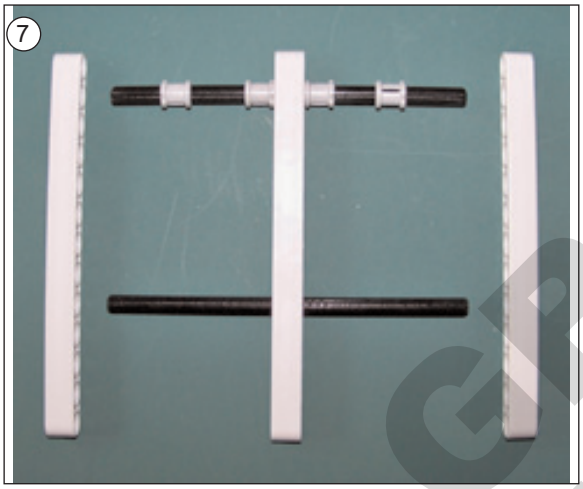
continued on next page

Module 9: Obstacle Avoidance Using the Touch Sensor *continued from previous page*



Push axel through 5th hole from the bottom





continued on next page

Module 9: Obstacle Avoidance Using the Touch Sensor *continued from previous page*

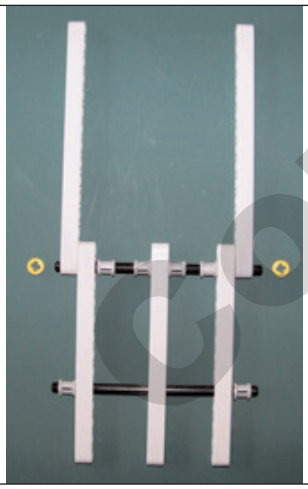
9

10

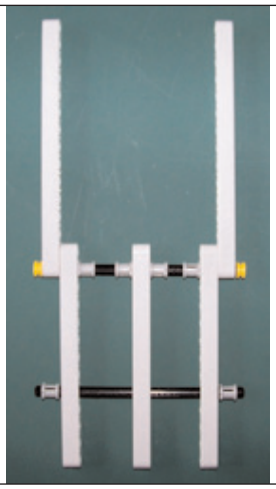
11

12

13

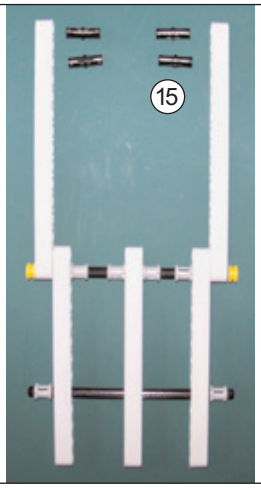


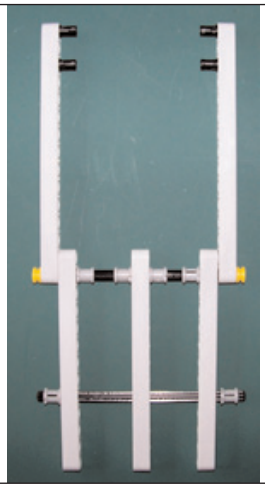
14

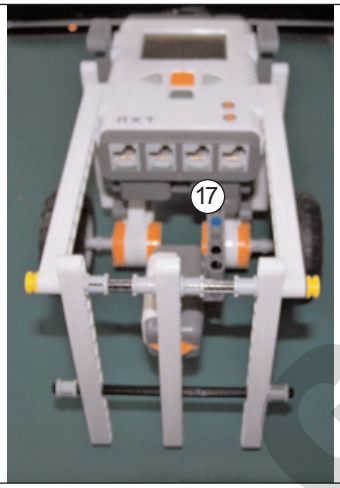


continued on next page

Module 9: Obstacle Avoidance Using the Touch Sensor *continued from previous page*







Robot seen from different angle

continued on next page

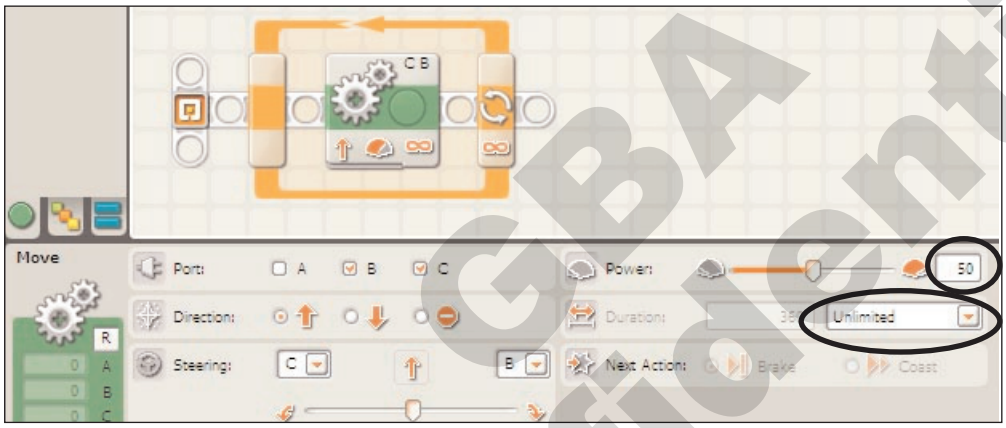
Module 9: Obstacle Avoidance Using the Touch Sensor *continued from previous page*

PROGRAMMING STEPS

1. Add a Loop block.

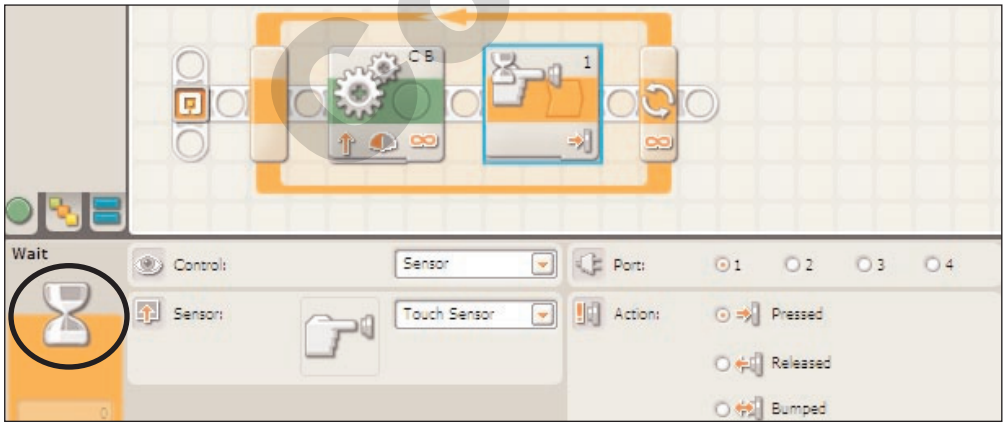
Control: Forever

2. Add a Move block inside of the Loop block, and then change the power option to 50. Make sure the duration is set to Unlimited.



Port: B C
Direction: Forward
Steering: Centered
Power: 50
Duration: Unlimited
Next Action: option not active

3. Use the mouse to Hover over the "Wait Blocks" in the Common Palette, and then select the touch sensor block. Move it inside of the loop.



Control: Sensor
Sensor: Touch Sensor
Port: 1
Action: Pressed

Note on Step 2

In step 2, the power can be set to whatever students prefer or whatever works best for the task. When sensors are involved, slower movement usually works better. Slower movement gives the sensor(s) more time to detect objects and obstacles.

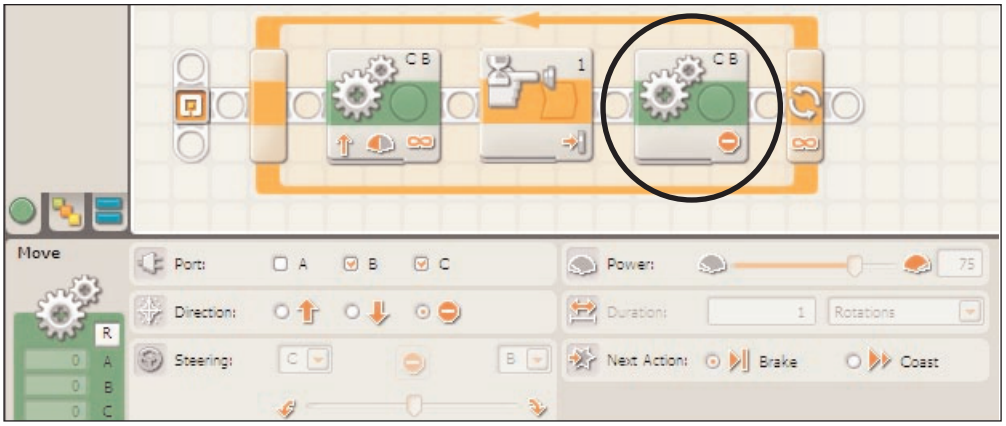
Touch Sensor Actions:

Though these actions may seem self-explanatory it's important to note the difference between the executions of these actions. If you choose to "press" the touch sensor in; the following action will occur as soon as the button is down. If you choose to "release" or "bump" the touch sensor then the action will not occur until the button is released. Try to remember that as you're using your program; you might find your touch sensor wedged between the object and itself, so choose your action accordingly.

continued on next page

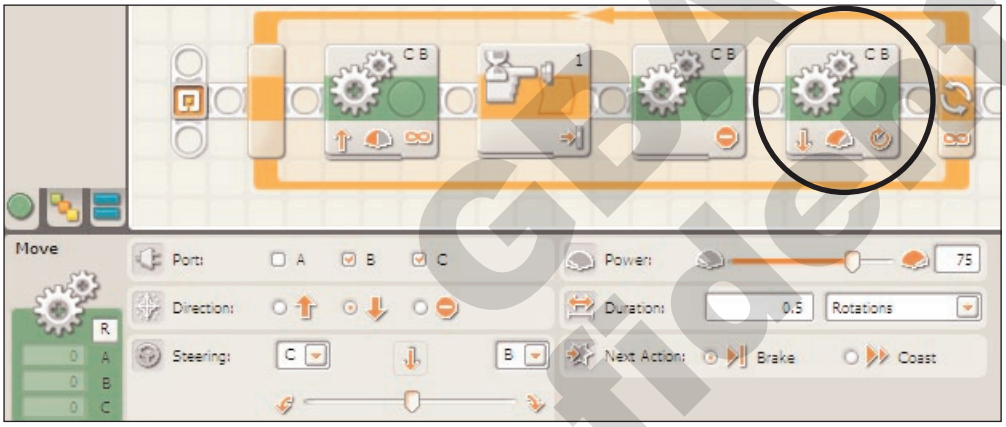
Module 9: Obstacle Avoidance Using the Touch Sensor *continued from previous page*

4: Add a move block inside of the loop, and then change the direction to stop. This will gray out most of the other options.



Port: B C
Direction: Stop
Steering: option not active
Power: 75
Duration: option not active
Next Action: Brake

5: Add another move block, and then set the direction to backwards. Make sure the duration is set to 0.5 rotations.

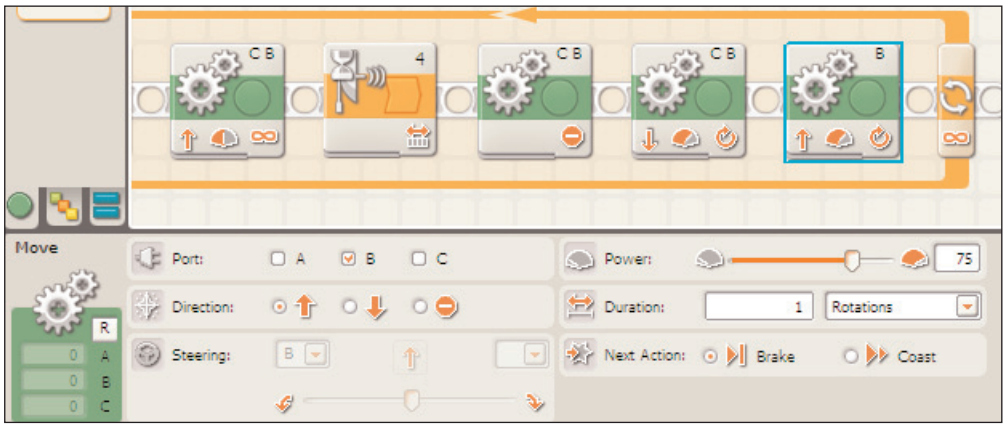


Port: B C
Direction: Backward
Steering: Centered
Power: 75
Duration: 0.5 Rotations
Next Action: Brake

continued on next page

Module 9: Obstacle Avoidance Using the Touch Sensor *continued from previous page*

6. Add another move block, and deselect motor C in the port option. This will cause only motor B to spin its wheel, and allow the robot to turn to the right. Note that this is a more accurate method of turning than using steering as we have previously done.



Port: B
Direction: Forward
Steering: option not active

Power: 75
Duration: 1 Rotations
Next Action: Brake

TEST THE PROGRAM

1. Download the program to the robot.
2. Position the robot to face a wall or other (suitably heavy) object, and run the program. The robot should approach the obstacle, and then as it comes close enough, the touch sensor should be pressed in by the object approached. The robot should then stop, back up, turn, and start the loop of all steps again by moving towards the next obstacle.

   Optional: If some of your students are finished with making, test, and debugging your program then have them add sounds to their robot! As a suggestion, have students add a sound block after hitting an object and then an applauding sound when the robot moved away from it.

   Note: Wires may get stuck on walls, or tangled in wheels. Avoid this problem by attaching an appropriate sized data wire or wrapping the extra wire with a rubber band.

END

WHY ISN'T THIS WORKING?

If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports
4. The Touch Sensor is connected to the correct port
5. The proper program is selected, and the brick is currently running it.
6. The program is within a loop
7. The "Move" blocks are set to the proper rotations, power, port, or direction
8. The Touch block is set to "press" and not "Released"

Module 10: Obstacle Avoidance Using the Ultrasonic Sensor

OVERVIEW

In the previous module, students used the touch sensor to detect obstacles. One of the limitations of this method is that the robot must come close to the wall and cannot detect objects at any distance. The Ultrasonic sensor will overcome these challenges and enable the robot to detect objects at various distances.

There are many uses for the ultrasonic sensor. One basic use is to detect when the robot is about to bump into a wall or other object. In this module, students will attach the ultrasonic sensor to the front of the robot, program the robot to stop when a wall or obstacle is detected, back up, turn, and continue until it detects another wall or object.

ABOUT THE ULTRASONIC SENSOR

To detect distance, the ultrasonic sensor sends out a sound wave. That sound wave then bounces off an object and returns to the ultrasonic sensor. The time it takes for the sound wave to travel back and forth is used to calculate the distance between the ultrasonic sensor and the object. This same technique is used by certain mammals, including bats, whales, dolphins and others. It is known as "echo location."

DIFFERENT WAYS TO MAKE OBSTACLES:

Any object could be stacked high enough or used to make the robot turn. You can try being creative in your approach by including art designs on these objects to make them stand out. Fold a piece of cardboard, or foam board (filled with artist designs from your class) and stack it high enough for the robot to "see."

HOW THE PROGRAM WORKS

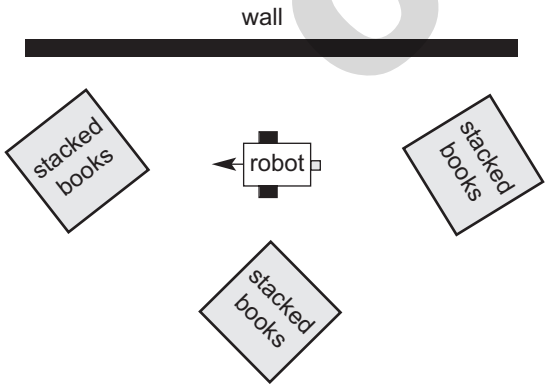
The robot moves forward until the ultrasonic sensor detects an obstacle or object at a distance of 6 inches or less (students will adjust this distance to suit different playfield setups). When an obstacle is detected, the robot stops, backs up, turns, then continues forward. This program loops.

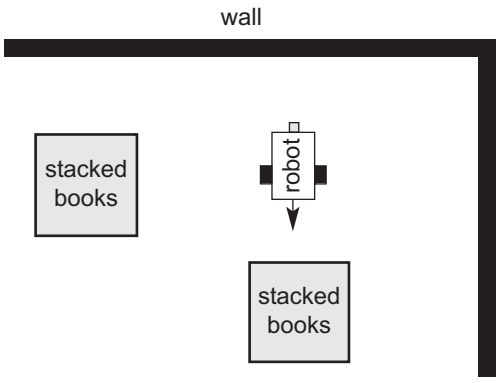
PLAYFIELD MATERIALS: Objects tall enough to be sensed by the robot.

PLAYFIELD SETUP:

Walls and other objects, arranged to block the robot at various points. Objects of any weight can be used - they do not have to be as heavy as in the previous lessons. Boxes of various sizes will work, including the bins that the robots came in. Items do not have to be at right angles to each other (Sample arrangements below are not to scale. Spread objects further apart than they appear in diagrams.)

Height is an issue: The Ultrasonic sensor can detect various distances of an approaching object only when it can "see" the object. Any object placed around the robot must be stacked high enough so that the ultrasonic sensor can "see" it approaching. Any object obstructing its view or within its peripheral vision will offset the sensor. You can prevent this by keeping all data wires from hanging overhead the sensors "eyes". Also keep in mind that the robot will not be able to see an object from an angle if it's not facing it.





continued on next page

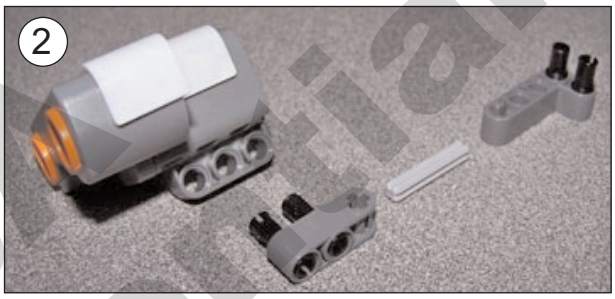
Module 10: Obstacle Avoidance Using the Ultrasonic Sensor *continued from previous page*

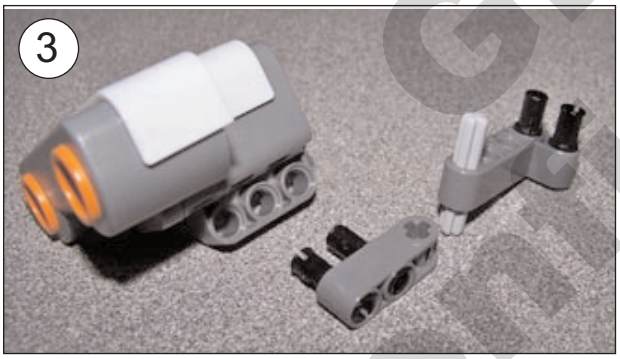
BUILD THE ULTRASONIC SENSOR ATTACHMENT

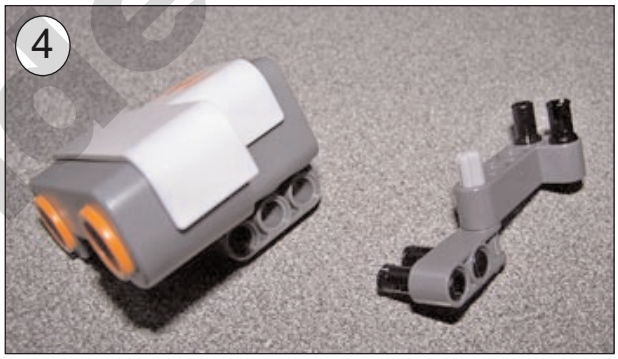
Build the attachment for the ultrasonic sensor, remove the touch sensor attachment from the previous lesson, then attach the ultrasonic sensor to the front of the robot. Follow the diagrams below:

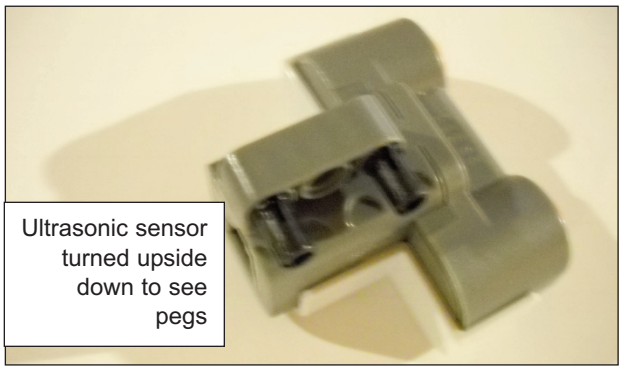
Parts List

1 ultrasonic sensor
1 turney thing (2 x1)
1 axle (length 3)
1 L-beam (length 4 x1)
4 connector pins



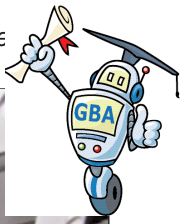


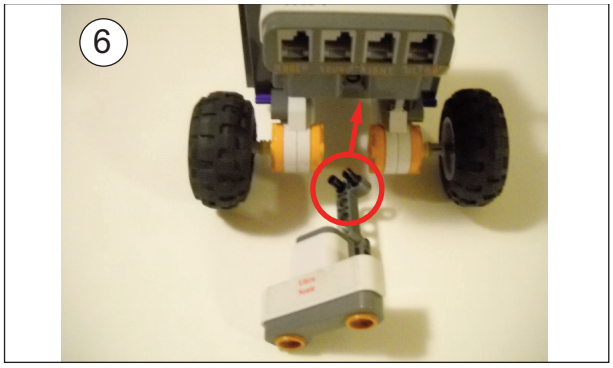


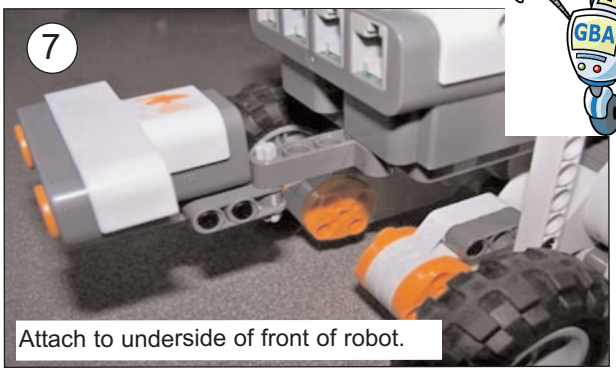


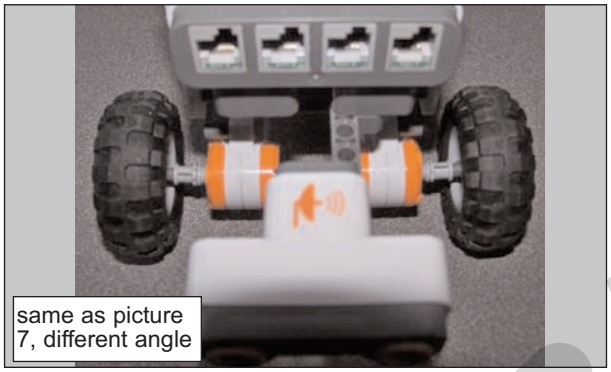
continued on next page

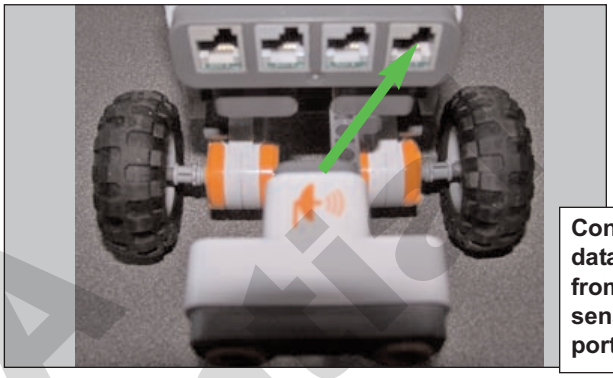
Module 10: Obstacle Avoidance Using the Ultrasonic Sensor *continued from previous page*











Connect a data wire from the sensor to port 4

Why isn't this working?

If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The sensor is connected to the correct port
5. The proper program is selected, and the brick is currently running it.
6. The program is within a loop
7. The "Move" blocks are set to the proper rotations, power, port, or direction
8. The Ultrasonic Block is set to the appropriate distance

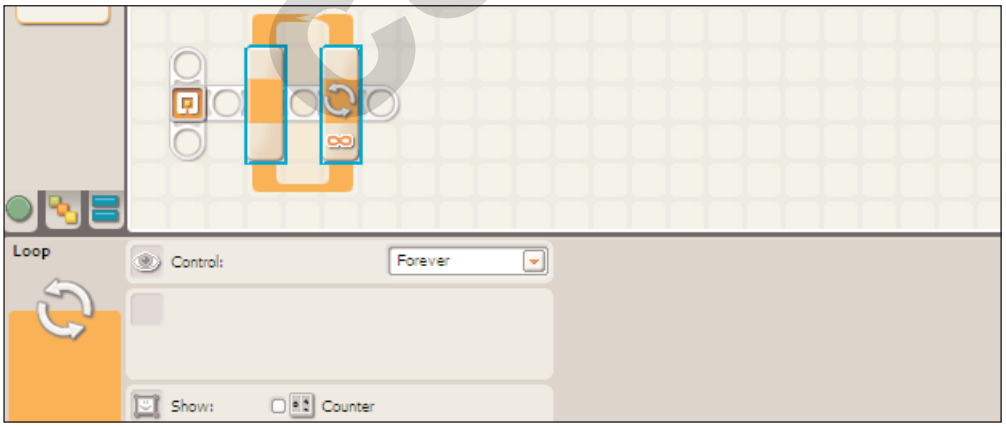
Making A Flowchart

A flowchart is a very useful visual tool used by programmers to plan their programs. It is a set of boxes containing instructions written in regular English linked by arrows in the order in which they are performed. Introducing flowcharts is a great way to help students plan and understand their programs. While discussing this and future programs, it is a good idea to create a flowchart to plan the program.

Note: For a loop, use an arrow from the last box to the first.

PROGRAMMING STEPS:

1. Add a Loop block.

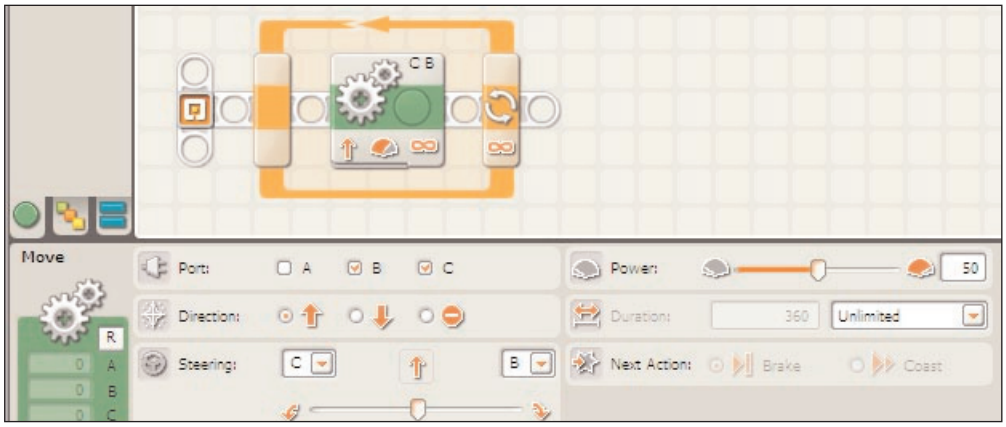


Control: Forever

continued on next page

Module 10: Obstacle Avoidance Using the Ultrasonic Sensor *continued from previous page*

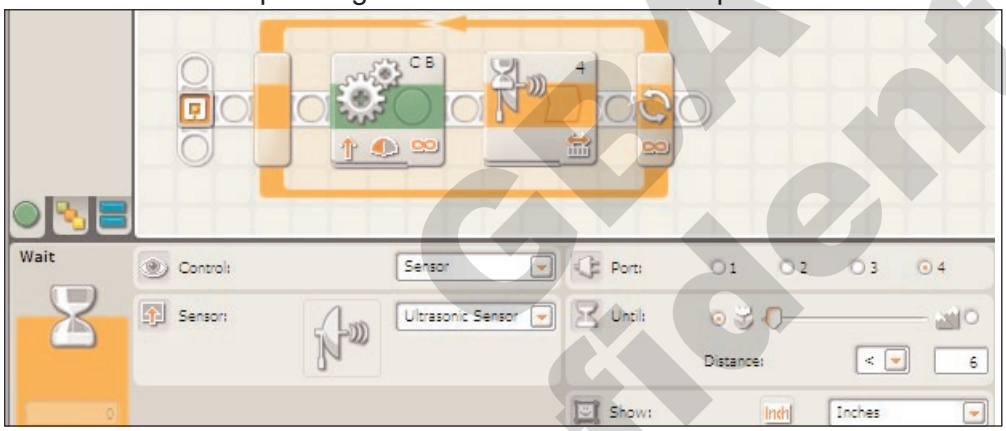
2. Add a move block inside of the loop block, and then change the power to 50. Switch the duration to unlimited.



Port: B C
Direction: Forward
Steering: Centered

Power: 50
Duration: Unlimited
Next Action: option not active

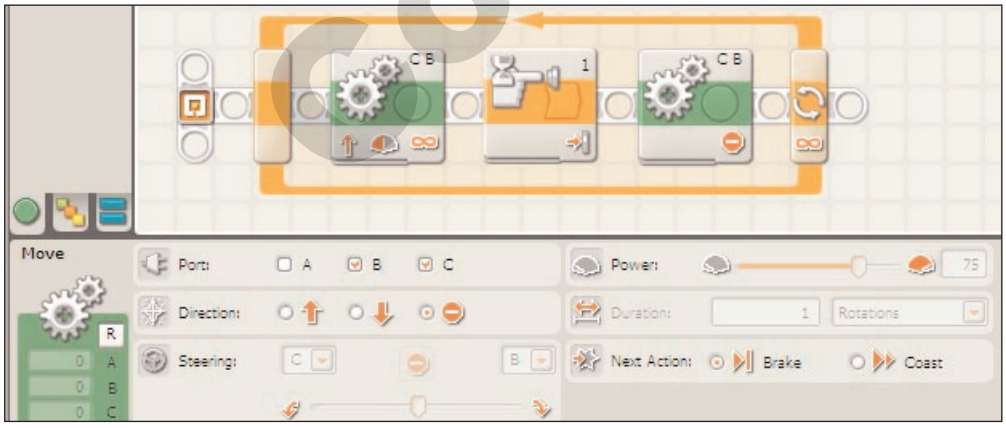
3. Hover the mouse over the "Wait Blocks" in the Common Palette; add an Ultrasonic Sensor block to the loop. Change the number in the distance option to 6 inches.



Control: Sensor
Sensor: Ultrasonic Sensor

Port: 4
Until: < 6
Show: Inches

4. Add a move block inside of the loop, and then change the direction to Stop. This will gray out most of the other options.



Port: B C
Direction: Stop
Steering: option not active

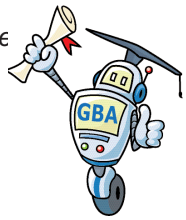
Power: option not active
Duration: option not active
Next Action: Brake

Inches vs. Centimeters:

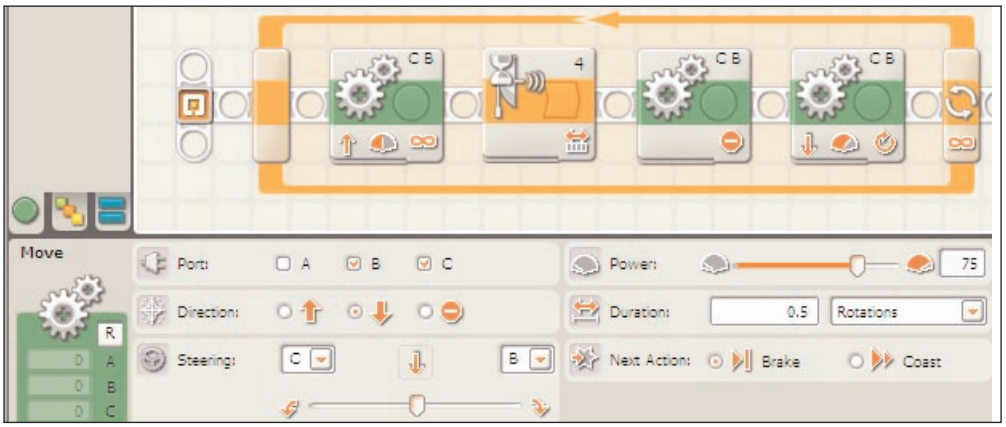
Inches are able to provide a board range of distance for your robot to read and act accordingly. However if you're looking for the best results, use centimeters instead of inches. Centimeters will provide the closest measurement that you program into distance since they are a smaller unit. As an added bonus the NXT program will convert Inches to Centimeters for you! Just be sure to input the number of inches you want then switch the tab (located on the "Show" strip) from inches to centimeters.

continued on next page

Module 10: Obstacle Avoidance Using the Ultrasonic Sensor *continued from previous page*



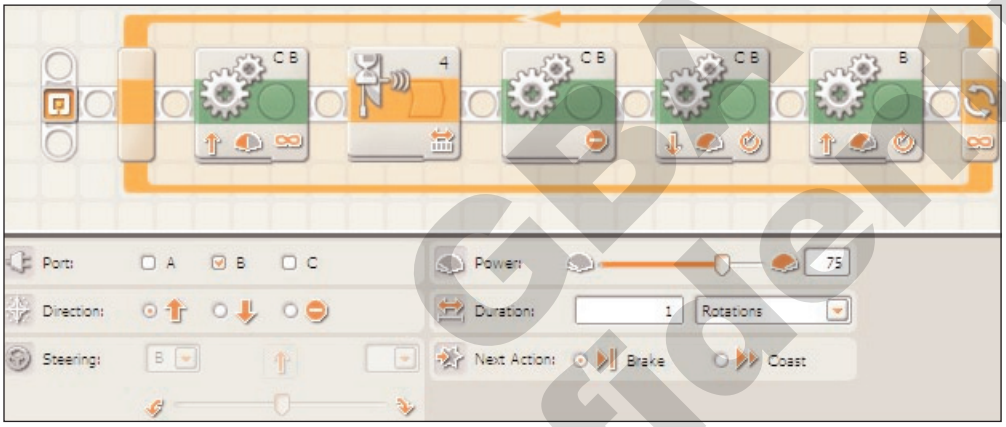
5. Add another move block, and then set the direction to backwards. Make sure the duration is set to 0.5 rotations.



Port: B C
Direction: Backward
Steering: Centered

Power: 75
Duration: 0.5 Rotations
Next Action: Brake

6. Add another move block, and deselect motor C in the port option.



Port: B
Direction: Forward
Steering: option not active

Power: 75
Duration: 1 Rotations
Next Action: Brake

TEST THE PROGRAM

1. Download the program to the robot.
2. Position the robot to face a wall or other object, and run the program. The robot should approach the obstacle, stop, back up, turn, and start the loop of all steps again by moving towards another obstacle.

GAME/CHALLENGES:

Once you're finished attaching and programming the ultrasonic sensor to the robot you could build a maze for it to navigate through!

END

NOTE:

Remember that turning with one motor is far more accurate than turning using both motors.

OPTIONAL ADDITIONS:

Changed the block located in "Step 6" from Port B to Port BC. Make sure you change the steering to the appropriate direction and record the difference in movement.

WIRES:

Wires may get stuck on walls, or tangled in wheels. Avoid this problem by attaching an appropriate sized data wire or wrapping the extra wire with a rubber band.

Module 11: Line Following Using the Light Sensor

OVERVIEW:
The light sensor can take readings in two different ways; either based on ambient light or reflected light. The ambient light setting uses the general light level in the room, while the reflected light setting activates the built in red LED (light emitting diode) to read the amount of light reflected back into the sensor. The modules in this book will use the reflected light setting for this sensor. When reading reflected light, the sensor's measurements are based on the grayscale value of the object that the sensor is reading. Pure white gives the highest reading (100%), and pure black gives the lowest reading (0%). However these readings are under theoretically ideal conditions with theoretically ideal colors; this is never the case in actual use. The best way to tell what the grayscale values of the materials are is to use the view menu to take a reading of them.

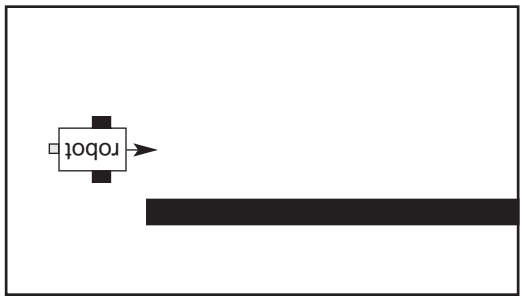
In this module, students will program their robots to follow a black line using the light sensor. Complex line following programs allows the robot to follow a line directly without swerving back and forth on the line. Simple ones swerve back and forth on the boundary of the line. The program in this module will be of the simpler boundary style.

HOW THE PROGRAM WORKS
When the robot detects reflected light, above the threshold set, motor B will spin and motor C will stop. When the robot detects reflected light, below the threshold set, motor C will spin and motor B will stop. With this functionality, the robot will continue to swerving back and forth on the boundary of the line, constantly moving forward. This program is effective as long as the line does not make sharp turns. It's recommended that the course use gradual curves instead of right or acute angles.

PLAYFIELD MATERIALS: White Poster board and black electrician's tape.

PLAYFIELD SETUP:
Use the electrician's tape to mark out a path for the robot to follow on the poster board. Remember not to include any sharp turns, and leave enough room to compensate for the robot's turning radius. You can either set up a closed course or an open course, as shown below. You could set up the playfield yourself, or you could have the students cooperate and design their own paths together by connecting various poster boards together.

Closed Course



Open Course

What is Grayscale:

Grayscale is a monochromatic color system that only includes black, white and shades of gray. This is how the light sensor senses "color". The light sensor can sense the intensity of a color, but it does not truly sense colors.

A Note on the Playfield Setup:

When creating the playfield it is important to make sure that the black line doesn't approach the edge of the playfield to make it as easy as possible for the light sensor to see the difference between the black line and white Poster board, and not wander off on the floor of the classroom. If some of the robot's wheels are on the playfield and some are off, the robot could also move the playfield if it is not secured to the floor.

Optional materials: You can use dark tape, dark construction paper with glue or tape, markers or any other reasonably dark material that will stick to the Poster board, or any other light colored material that will remain on the floor.

How to make a bigger playfield: You can use duct tape on the bottom of the posterboard pieces to extend the playfield past one posterboard.

continued on next page

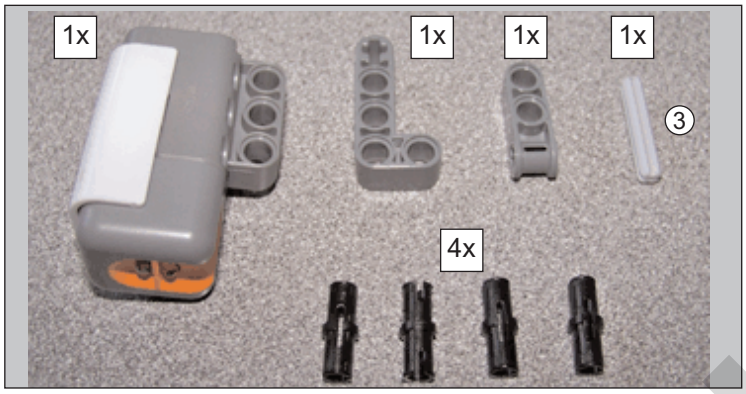
Module 11: Line Following Using the Light Sensor *continued from previous page*

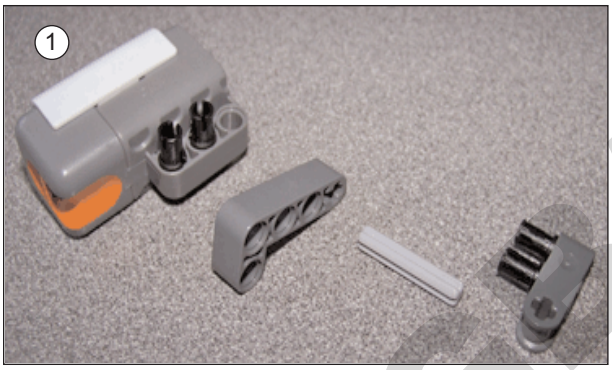
BUILD THE LIGHT SENSOR ATTACHMENT

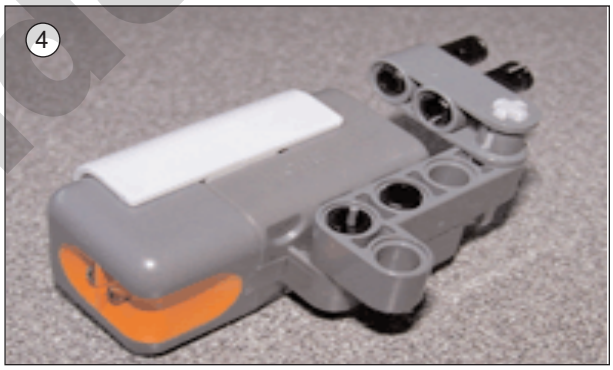
Build the attachment for the light sensor, remove the ultrasonic sensor attachment from the previous lesson, then attach the light sensor to the front of the robot. Follow the diagrams below:

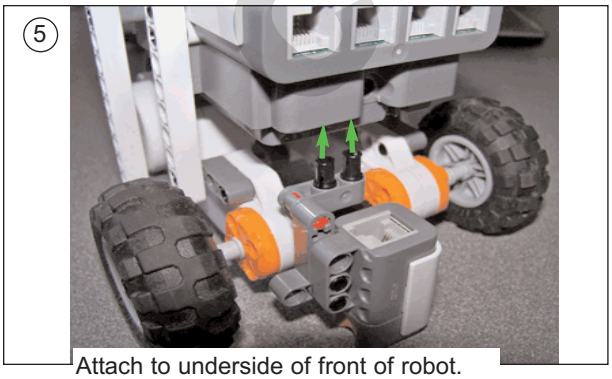
Parts List

1 light sensor
1 L-beam (length 4 x1)
1 gray right angle (2x1)
1 axle (length 3)
or (1 red/blk pin)
4 connector pins

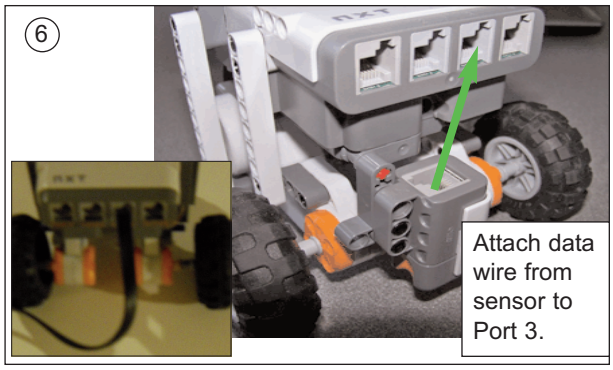








Attach to underside of front of robot.

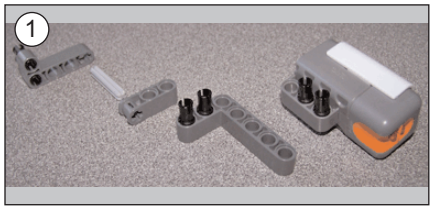


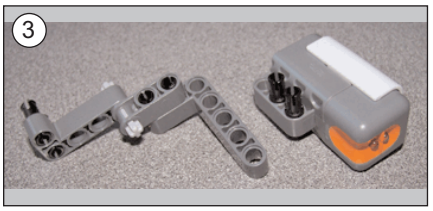
continued on next page

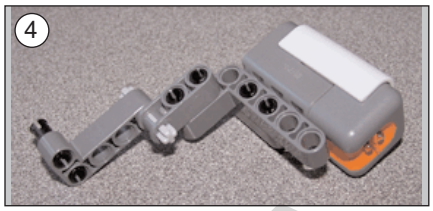
Module 11: Line Following Using the Light Sensor *continued from previous page*

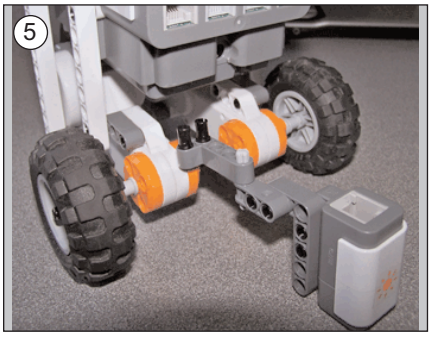
Below is an alternate build for the light sensor attachment.

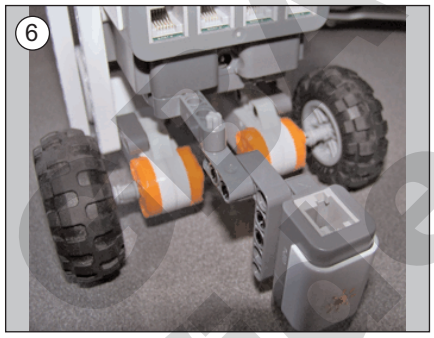
This version positions the sensor further out in front of the robot. It also positions the sensor in the precise horizontal center of the robot. Parts are the same as above, with addition of 1 I-beam (5x3) and 2 additional connector pins.











Attach to underside of front of robot.

How to Take Measurements With the View Menu:

To effectively program the robot with the light sensor it's important to know the reflected light values for the various objects plan to use. The most accurate way to find this out is to use the reflected light view tool built into the robot.

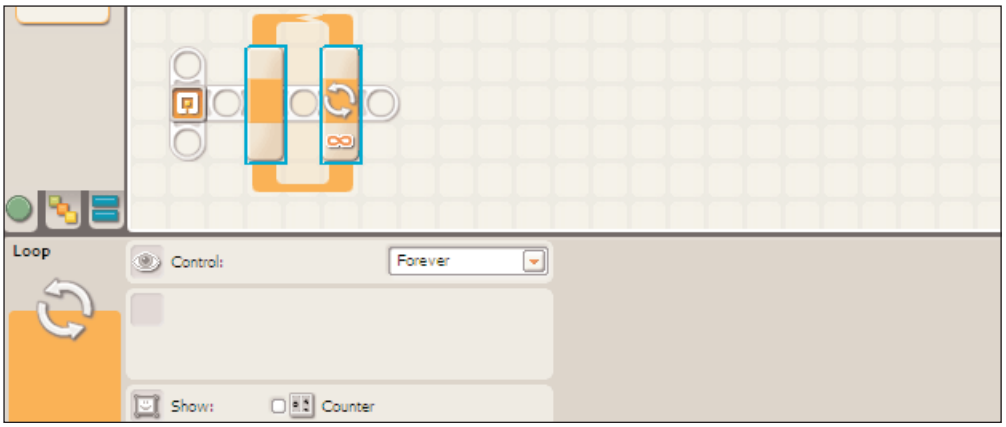
1. Press the orange button to tun the brick on.
2. Press the right arrow twice to highlight the view menu.
3. With the view menu highlighted, press the orange button to access the view menu.
4. Press the right arrow four times to highlight the reflected light view tool.
5. Press the orange button to bring up the port selection screen.
6. Press the right arrow twice to highlight port 3.
7. Press the orange button to select port 3 and activate the tool.
8. Hold the light sensor a small distance away from the object you want a reading about. The reflected light value will appear on the brick.

continued on next page

Module 11: Line Following Using the Light Sensor *continued from previous page*

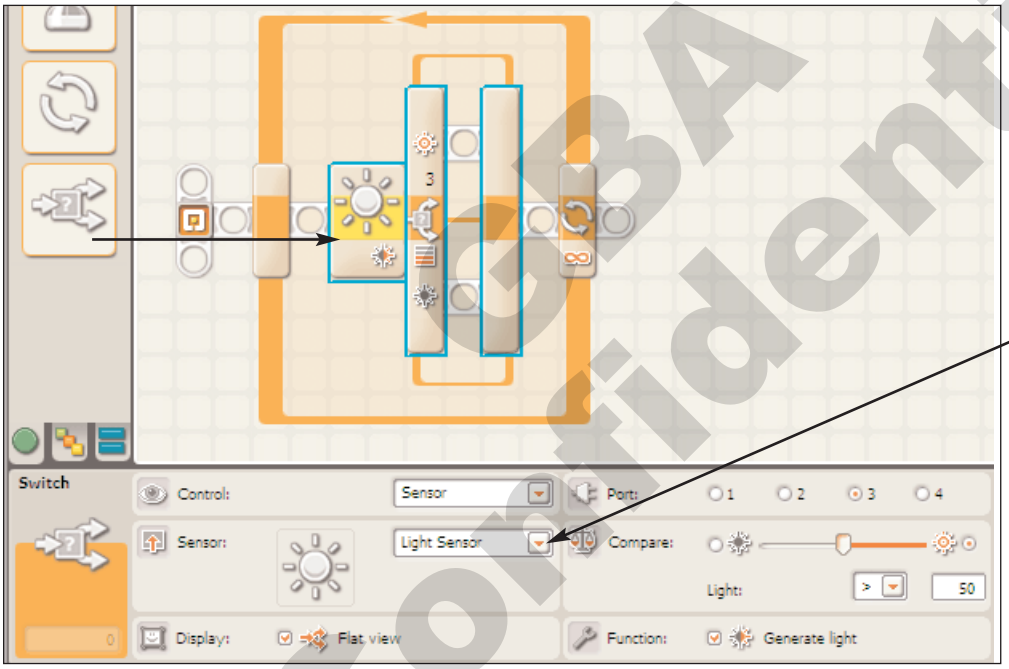
PROGRAMMING STEPS:

1. Add a loop block.



Control: Forever

2. Add a switch block inside of the loop. The program may pause for a second as this happens.



Control: Sensor
Sensor: Light Sensor
Display: Flat view

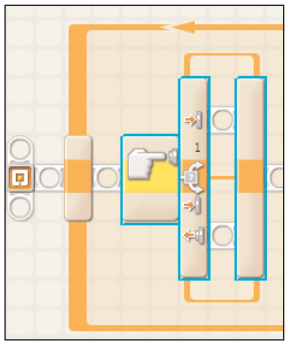
Port: 3
Compare: > 50
Function: Generate light

3. Use the sensor option to select the light sensor, which will change the available options on the right side of the options palette.

What is Grayscale:

Grayscale is a monochromatic color system that only includes black, white and shades of gray. This is how the light sensor senses "color". The light sensor can sense the intensity of a color, but it does not truly sense colors.

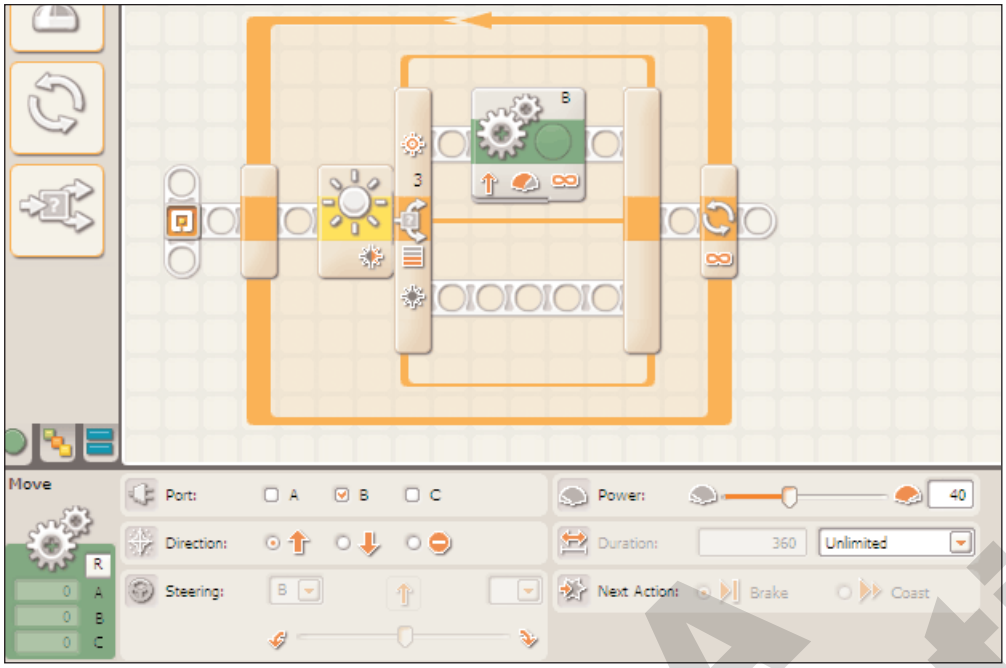
Note: When you add a Switch block, the default setting will be a Touch sensor switch. You can change this in the settings, using the Sensor option.



continued on next page

Module 11: Line Following Using the Light Sensor *continued from previous page*

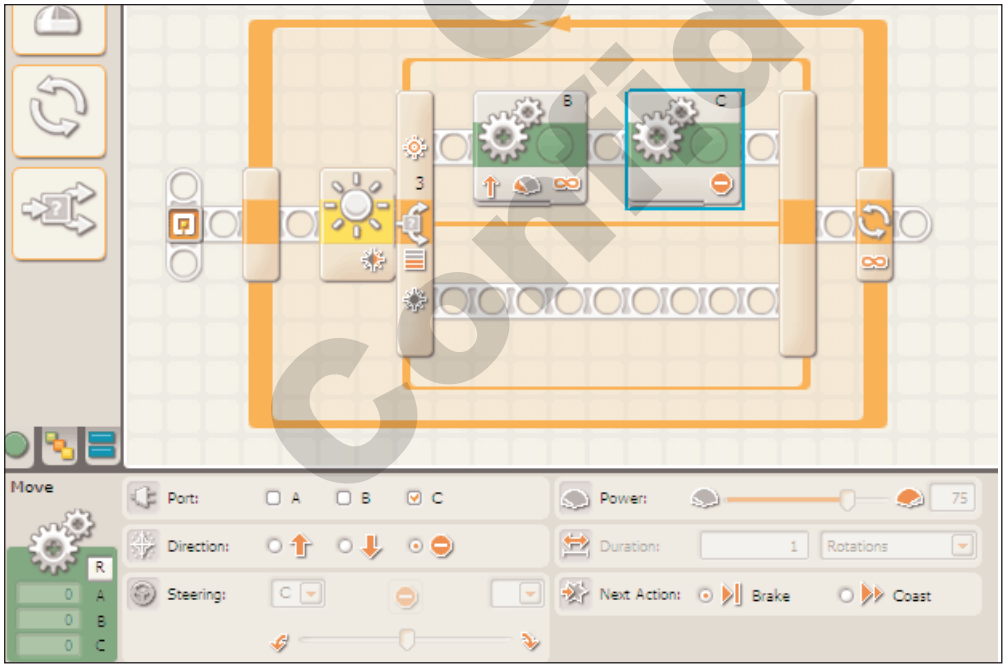
4. Add a move block to the top half of the switch block. Deselect motor C under the port option, change the duration to unlimited, and the power to 40.



Port: B
Direction: Forward
Steering: option not active

Power: 40
Duration: Unlimited
Next Action: option not active

5. Add another move block to the top half of the switch. Deselect motor B under the port option, and set direction to Stop.



Port: C
Direction: Stop
Steering: option not active

Power: option not active
Duration: option not active
Next Action: Brake

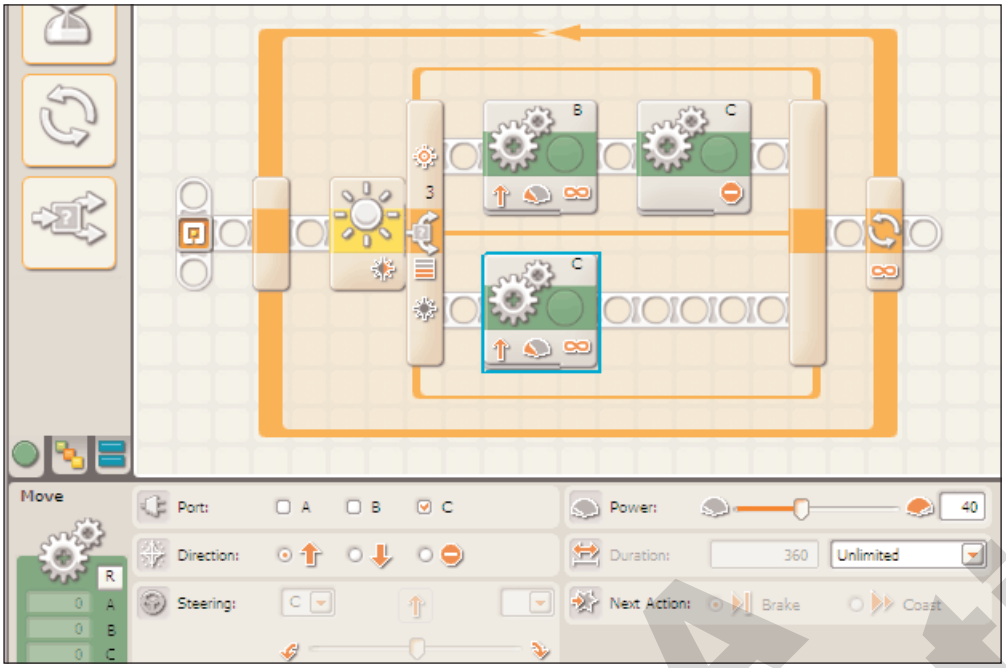
How A Switch Block Works

A switch block is a two-choice decision mechanism. The switch block has two sections, each with their own set of blocks or subprogram. If a certain condition exists, one subprogram runs, while the other subprogram runs if the opposite condition is true. For example, the condition in this module is that the level or reflected light is above 50. If the light level is greater then 50 the subprogram to turn right runs, however if the light level is less than 50 the subprogram that turns to the left runs. Note that the switch statement can only select between two options in the MINDSTORMS® software, but in most programming languages a switch statement can choose between any number of choices.

continued on next page

Module 11: Line Following Using the Light Sensor *continued from previous page*

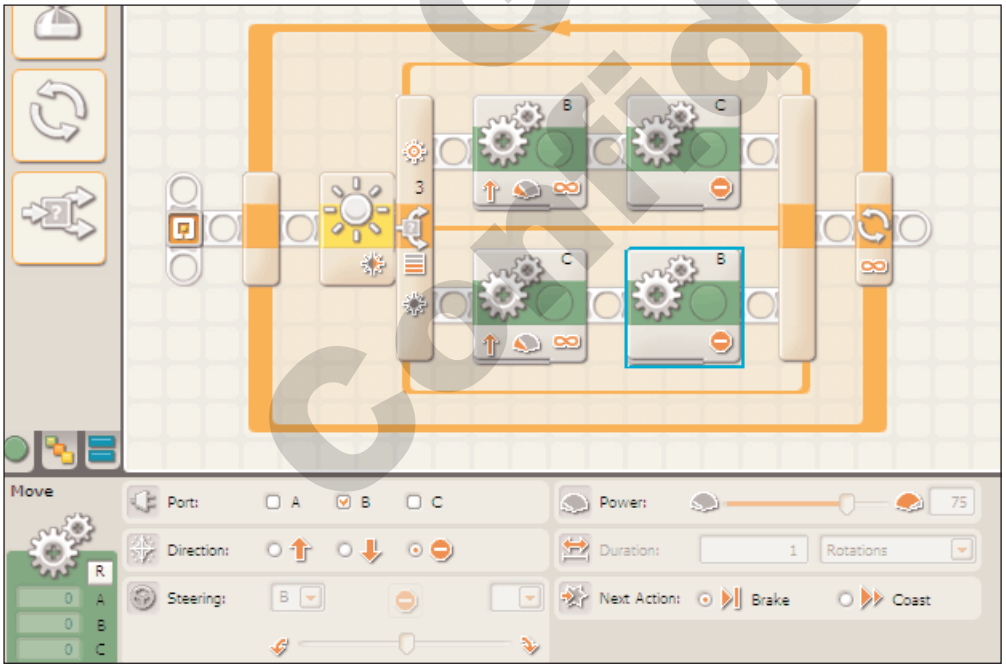
6. Add a move block to the bottom half of the switch. Deselect motor B under the port option, change the duration to unlimited, and the power to 40.



Port: C
Direction: Forward
Steering: option not active

Power: 40
Duration: Unlimited
Next Action: option not active

7. Add another move block to the bottom half of the switch. Deselect motor C under the port option, and set direction to Stop.



Port: B
Direction: Stop
Steering: option not active

Power: option not active
Duration: option not active
Next Action: Brake

8. Save the program using "Save as".

continued on next page

Module 11: Line Following Using the Light Sensor *continued from previous page*

TEST THE PROGRAM

1. Download the program to the robot, and then navigate to it on the robot.
2. Position the robot with the light sensor slightly to the left of the black line; turn the program on. The robot should sweep back and forth, with the light sensor finding the line then turning away, moving forward constantly.

GAME/CHALLENGES:

To extend this module have the students create their own paths. One fun way to do this is to make a "Super Course." To do this, distribute poster board, tape, or black markers to the students. Instruct them to make a line on the poster board with the rule that the line must start on one side of the paper and end on the other. Make sure the line is thick so that the robot's sensor can find it. Once all of the students are done, connect the pieces of paper with the lines connected to create one giant course.

Optional Additions: One challenge to extend to students who complete this program too quickly is to change the program to follow the right side of the line instead of the left. (You would need to swap the top and bottom sections of the switch)

WHY ISN'T THIS WORKING?

If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The sensor is connected to the correct port
5. The proper program is selected, and the brick is currently running it.
6. The program is within a loop
7. The "Move" blocks are set to the proper rotations, power, port, or direction
8. Make sure the switch statement has the right options selected.
9. Make sure that the lines on the playfield are thick enough and far enough apart for the light sensor to properly read them.

END

RUNNING MORE THAN ONE ROBOT AT A TIME ON A COURSE:

You may want to run more than one robot at a time on the same course, for either time reasons, or to keep all the students occupied. If so, make sure that the tracks are big enough to handle more than one robot wobbling on them, that all the robots are running at the same speed so that they don't crash into each other, and that their wires don't catch each other.

Module 12: Stay within Boundaries – Line Detection Using the Light Sensor

OVERVIEW

The previous module used the light sensor to enable the robot to follow a path. In this module, students will program their robots to move around within a bordered area (playfield), while not going out of the playfield area. We'll be using the light sensor outside of a switch statement.

HOW THE PROGRAM WORKS

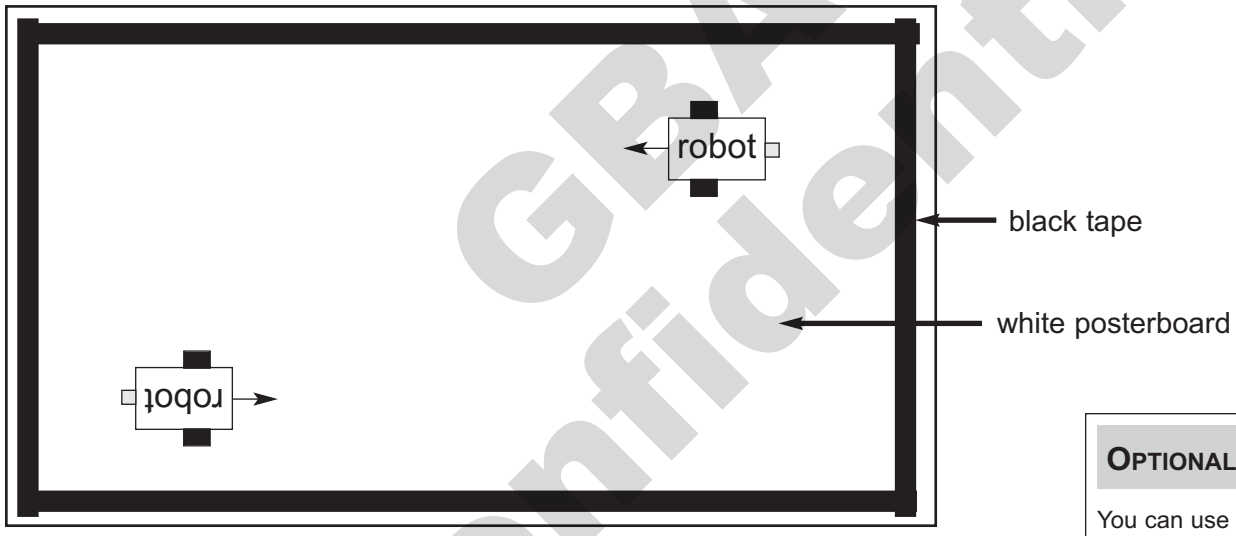
The robot moves forward until the light sensor detects a black line. At this point the robot stops, backs up slightly, turns around, and continues on its way. The program loops in this manner so that the robot continues moving while staying in the playfield area.

PLAYFIELD MATERIALS:

White poster board, duct tape, and black electrician's tape.

PLAYFIELD SETUP:

White posterboard with black electrician's tape is the primary playfield.To make the playfield, use duct tape to put together multiple pieces of white poster board. On the other side of the poster board, put black electrician's tape around the perimeter of the poster board to create the border of the playfield.



OPTIONAL MATERIALS:

You can use dark tape, dark construction paper with glue or tape, markers, or other reasonably dark material that will stick to the poster board. You could use light colored material that will remain on the floor.

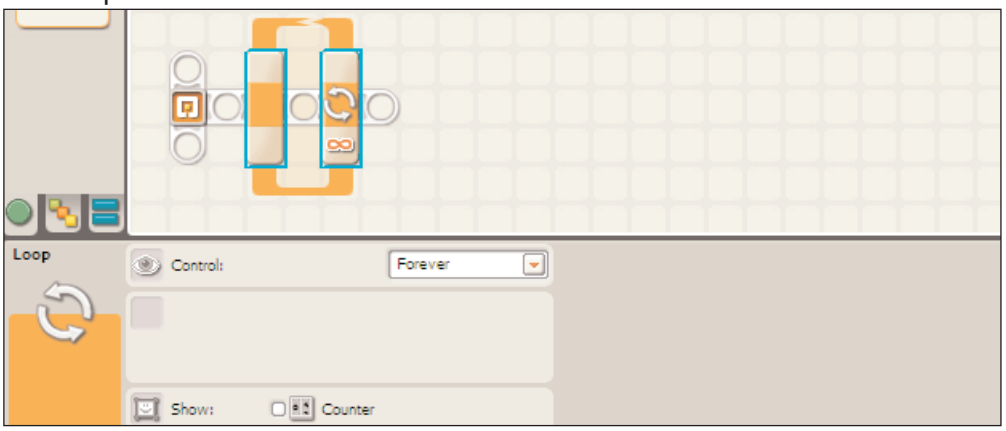
continued on next page

Module 12: Stay within Boundaries – Line Detection Using the Light Sensor

continued from previous page

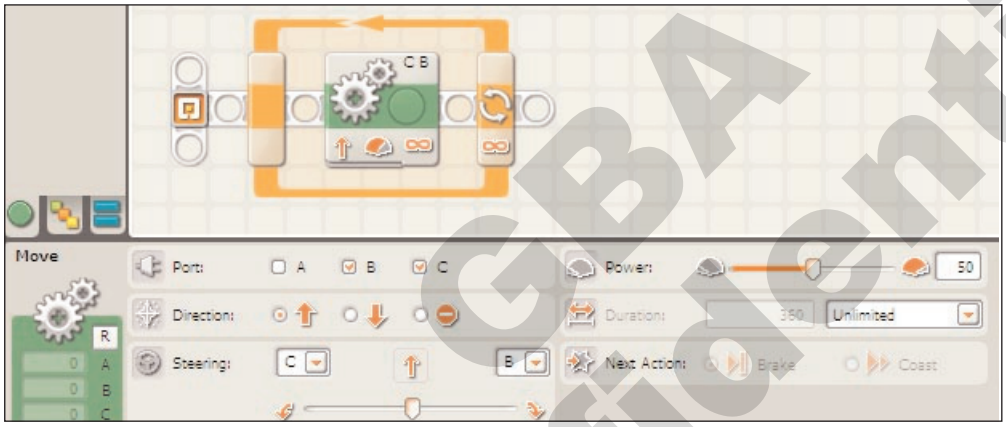
PROGRAMMING STEPS:

1. Add a loop block.



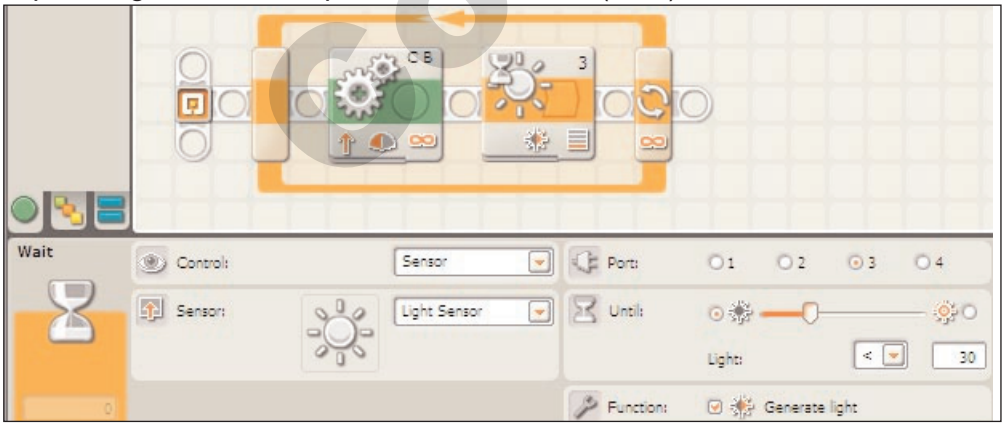
Control: Forever
Show: not checked

2. Add a move block inside of the loop block. Set the power to 50, and the duration to be unlimited.



Port: B C
Direction: Forward
Steering: Centered
Power: 50
Duration: Unlimited
Next Action: option not active

3. Add a light sensor block from the wait section of the common palette to the loop; change the "Until" option to less than 30 (< 30).



Control: Sensor
Sensor: Light Sensor
Port: 3
Until: < 30
Function: Generate Light (checked)

Note on the Threshold:

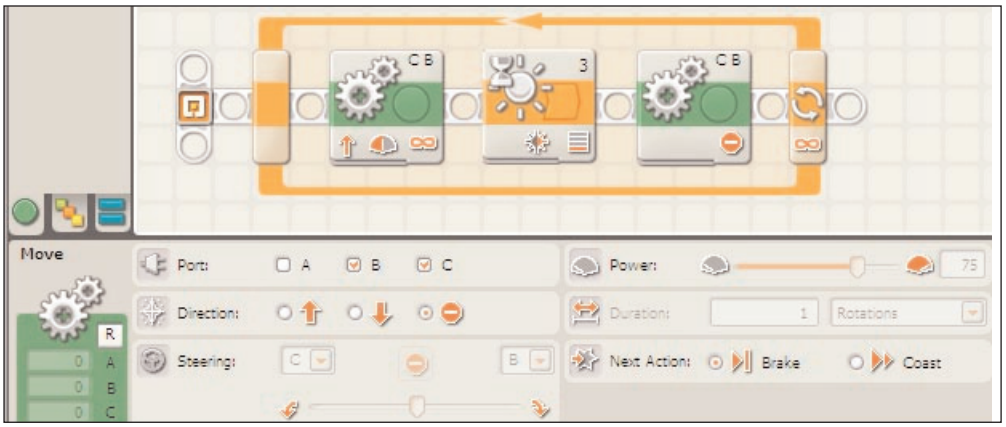
Most sensors work on a threshold, or a value above which they are activated. For example, if the amount of light sensed by the light sensor is above a certain value, the robot will perform the actions after the light sensor. However until that point the robot will continuously move forward.

continued on next page

Module 12: Stay within Boundaries – Line Detection Using the Light Sensor

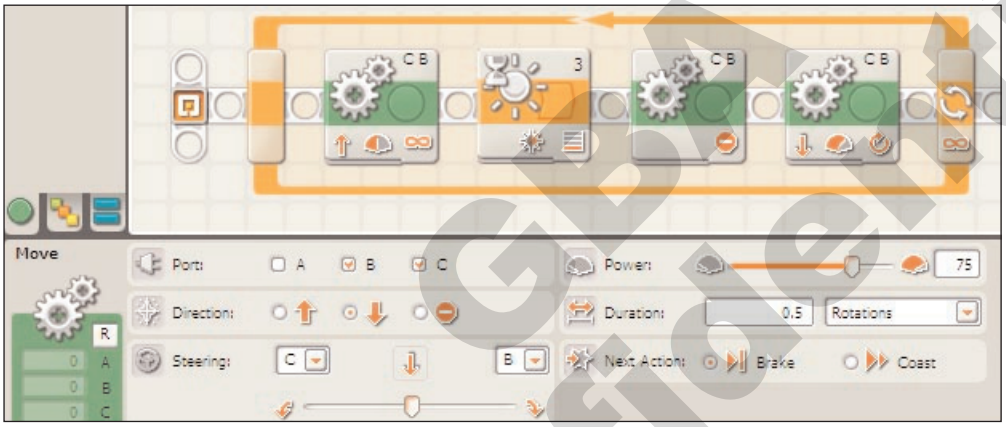
continued from previous page

4. Add another move block after the light sensor block, and set the direction to Stop.



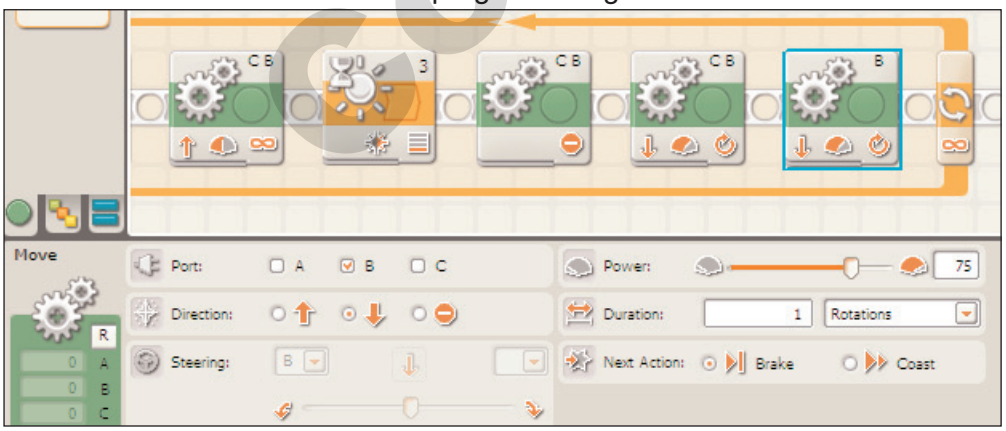
Port: B C
Direction: Stop
Steering: option not active
Power: option not active
Duration: option not active
Next Action: Brake

5. Add another move block, set the direction to backwards, and the duration to 0.5 rotations.



Port: B C
Direction: Forward
Steering: Centered
Power: 50
Duration: Unlimited
Next Action: option not active

6. Add another move block, deselect motor C in the port options, and set the direction to backwards. Save the program using "Save as."



Port: B
Direction: Backward
Steering: option not active
Power: 75
Duration: 1 Rotations
Next Action: Brake

continued on next page

Module 12: Stay within Boundaries – Line Detection Using the Light Sensor

continued from previous page

TEST THE PROGRAM

1. Download the program to the robot, and then navigate to it on the robot.
2. Position the robot anywhere inside of the playfield, and run the program. The robot should move forward until it reaches the edge of the playfield, at which point the robot should back up, turn and then continue on its way.

GAME/CHALLENGES:

Sumo bots: In this challenge, a number of robots are let loose inside the playfield and the winner is the last robot on all wheels and in the arena. It is very important to enforce that robots cannot leave the arena, and students cannot enter the arena. To expand this idea even further, allow students to modify the program designed in this module or make their own program. Attach any of the sensors they have already learned, or build any attachments to improve the robot's performance.

Cup Soccer: Place numerous cups open side down inside of the playfield, and separate students into small teams of around three robots per team. Place two teams of robots on the field and run them at the same time. Points are scored by pushing cups outside of the playfield, and the winner is the team with the most points. You can modify this challenge to fit different class sizes and time requirements by changing the size of the teams or using a tournament structure to this challenge.

WHY ISN'T THIS WORKING?

If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The sensor is connected to the correct port
5. The proper program is selected, and the brick is currently running it.
6. The program is within a loop
7. The "Move" blocks are set to the proper rotations, power, port, or direction
8. Make sure the threshold selected for the sensor block is correct, and that the comparative is set to less than.

CLASS TESTING:

To have different teams test the program during development, it's a good idea to make a few smaller practice fields (single piece of white posterboard with black outlines).

END

Module 13: Navigating a Maze with the Light Sensor

OVERVIEW

In this module, students will program the robot to maneuver through an obstacle course that is made up of black lines on white posterboard.

So far, the student has used the light sensor to program simple repeating behaviors on the robot. However, programs can also be created for the robot to follow a series of steps to complete specific challenges. In this challenge, the students will be shown a maze and will need to design a program that uses the light sensor to navigate it.

HOW THE PROGRAM WORKS

To navigate the sample course, the robot moves forward until it detects a black line. It then turns to the right 90 degrees and moves forward again until it detects another black line. The robot then turns left 90 degrees and moves forward until it detects a black line. The robot then turns to the left 90 degrees and moves forward a set number of rotations (since there is no line to detect before this turn), turns 90 degrees to the right, and moves forward until it detects a black line, and stops at the goal.

> **Note:** The following program will navigate the example course below, but different courses will require different programs.

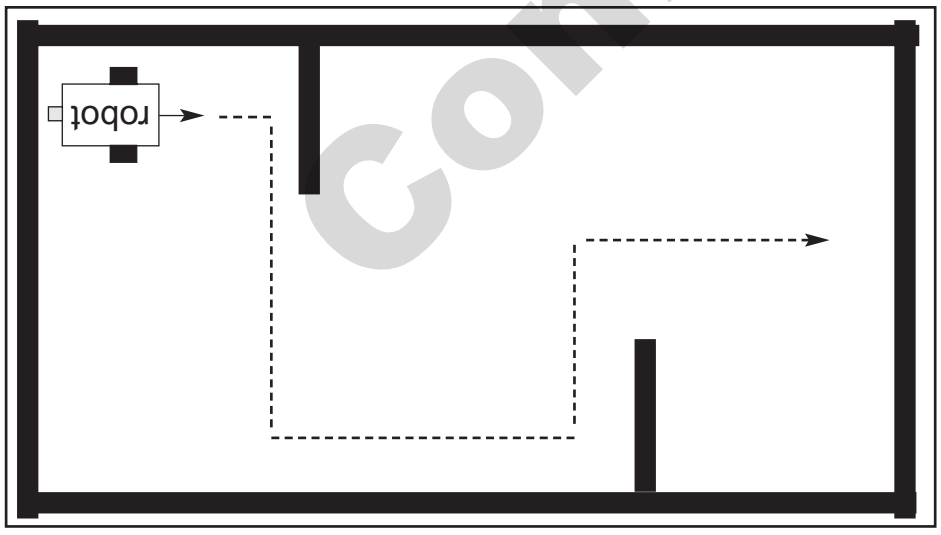
PLAYFIELD MATERIALS:

White posterboard, duct tape and black electrician's tape.

PLAYFIELD SETUP:

White posterboard with black electrician's tape is the primary playfield. The first playfield is a large piece of white posterboard, with black tape around the edges to create a border. A standard piece of posterboard at least 28" x 22" is suitable. To make the playfield, use duct tape to put together multiple pieces of white posterboard, then on the other side of the posterboard put black electrician's tape on the border of the playfield, and at various places, use cut strips of black duct tape perpendicular to the border to make a course. Make the sample maze in this module as well as at least one other maze for the students to navigate.

Variation, (sample) light sensor obstacle course

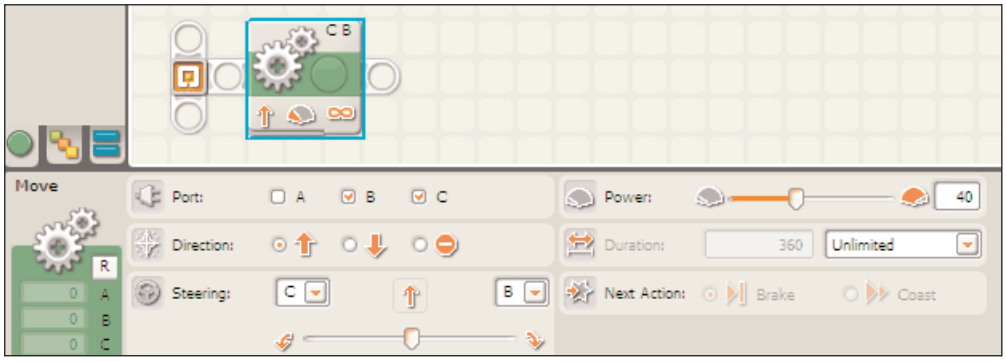


continued on next page

Module 13: Navigating a Maze with the Light Sensor *continued from previous page*

PROGRAMMING STEPS:

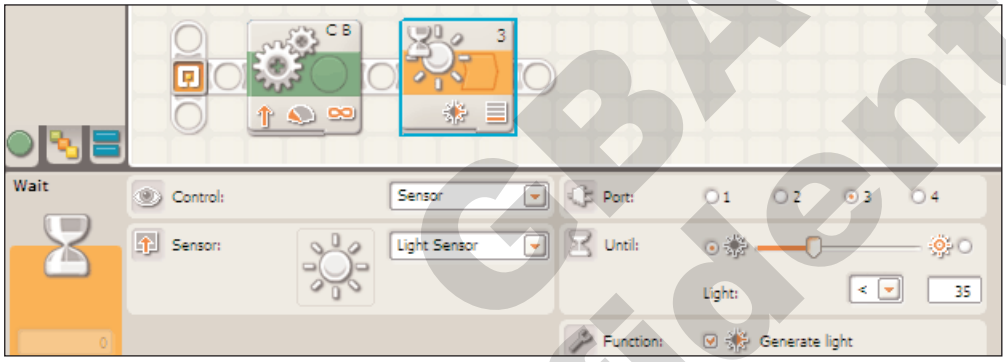
1. Add a move block to the sequence beam, change the power setting to 40, and the duration to unlimited.



Port: B C
Direction: Forward
Steering: Centered

Power: 40
Duration: Unlimited
Next Action: option not active

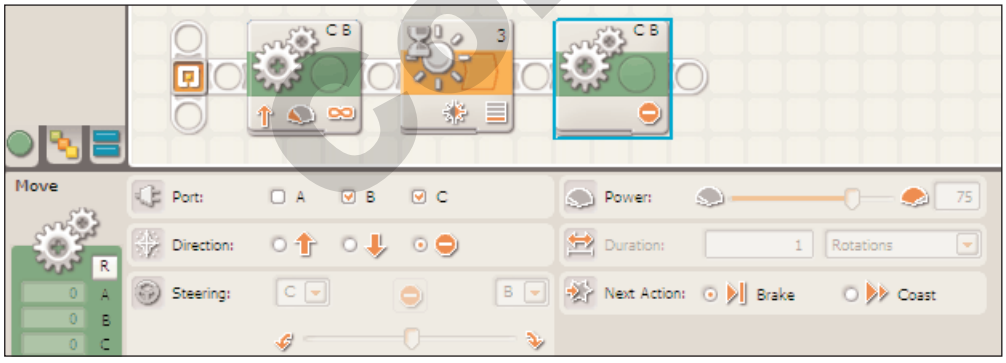
2. Add a light sensor block, and change the until option to less than 35 (<35).



Control: Sensor
Sensor: Light Sensor

Port: 3
Until: < 35
Function: Generate light

3. Add a move block and set direction to Stop.



Port: B C
Direction: Stop
Steering: option not active

Power: option not active
Duration: option not active
Next Action: Brake

Cumulative Error:

Each turn that the robot makes may not be 100% precise. Each minor imprecision adds to any that follow. For example, if each turn is 3 degrees off, the robot will be 9 degrees off course after three turns.

When a number of minor imperfections combine to result in a noticeable imperfection, the result is considered "cumulative error." Students may need to make minor adjustments to their programs to compensate for cumulative error, and even then, students and instructors cannot expect the robot to perform perfectly.

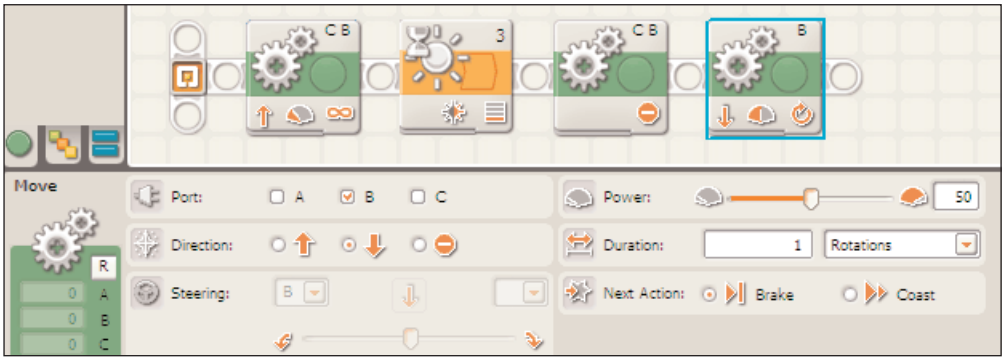
Different Ways To Turn the Robot

The robot can be made to turn by various methods. Both wheels can turn in opposite directions. Or, one wheel can rotate either forward or backward. Both of these methods cause the robot to turn, but turning by only rotating one wheel usually results in more precise turns.

continued on next page

Module 13: Navigating a Maze with the Light Sensor *continued from previous page*

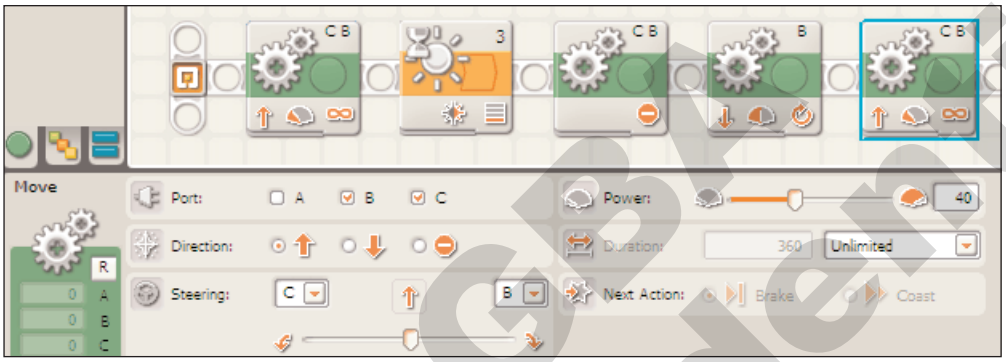
4. Add another move block, deselect motor C, set direction to backward, and the power to 50.



Port: B C
Direction: Backward
Steering: option not active

Power: 50
Duration: 1 Rotations
Next Action: Brake

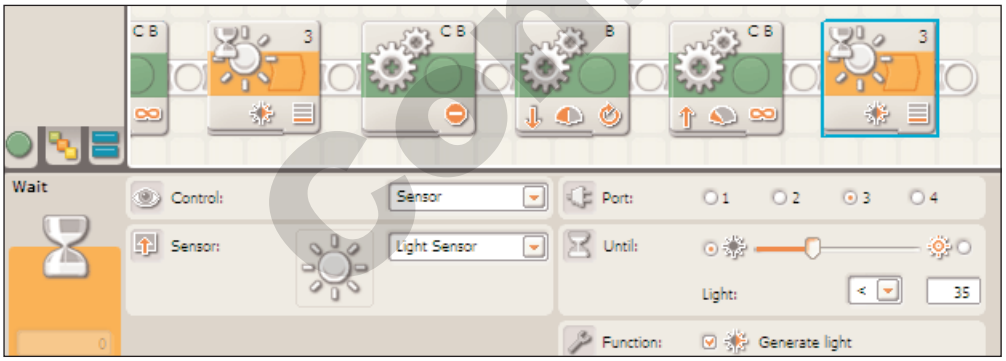
5. Add another move block, set the power to 40, and the duration to unlimited.



Port: B C
Direction: Forward
Steering: option not active

Power: 40
Duration: Unlimited
Next Action: option not active

6. Add a light sensor block, and change the until option to less than 35 (<35).



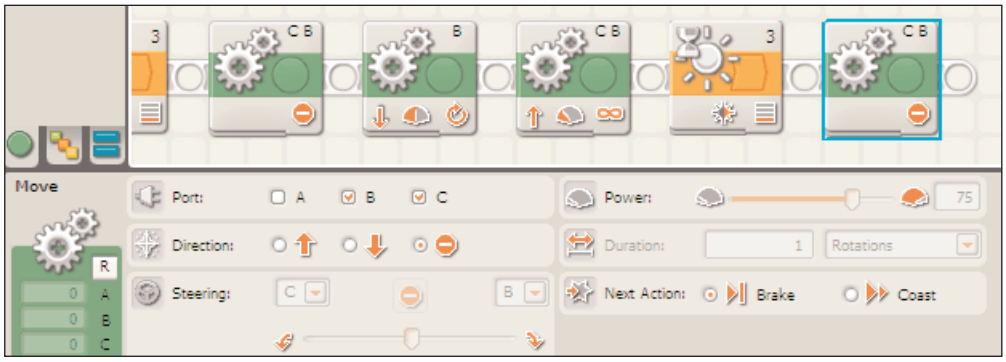
Control: Sensor
Sensor: Light Sensor

Port: 3
Until: < 35
Function: Generate light

continued on next page

Module 13: Navigating a Maze with the Light Sensor *continued from previous page*

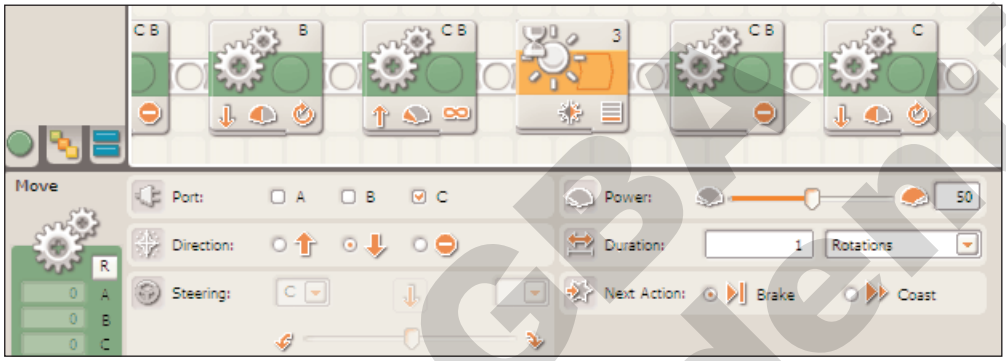
7. Add a move block and set direction to Stop.



Port: B C
Direction: Stop
Steering: option not active

Power: option not active
Duration: option not active
Next Action: Brake

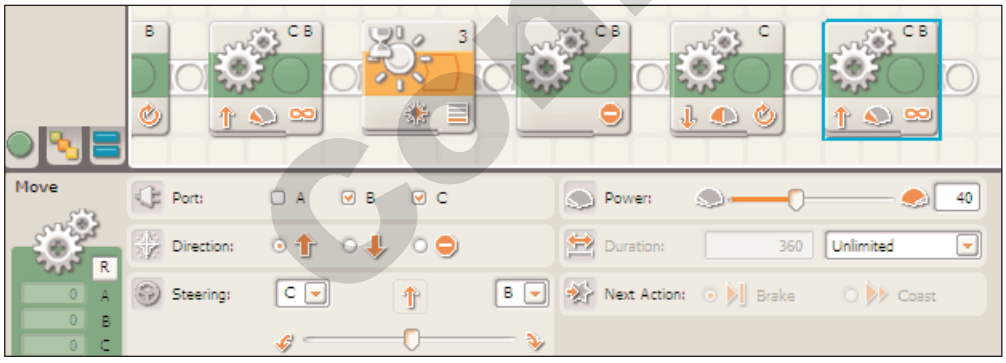
8. Add another move block, deselect motor B, set direction to backwards, and the power to 50.



Port: C
Direction: Backward
Steering: option not active

Power: 50
Duration: 1 Rotations
Next Action: Brake

9. Add another move block, set the power to 40, and the duration to unlimited.



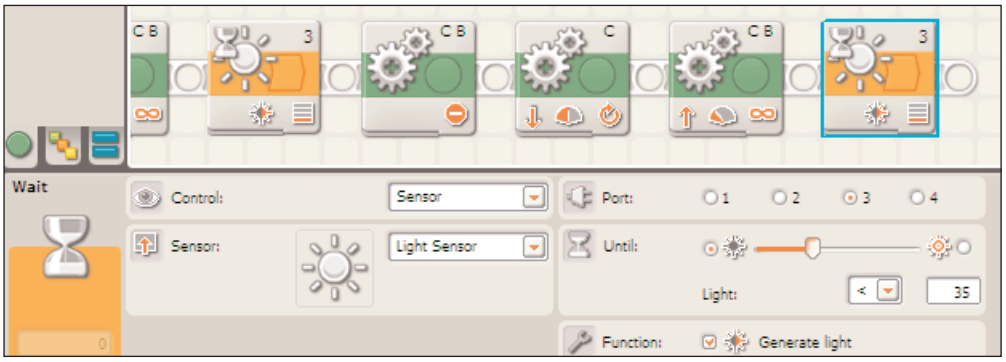
Port: B C
Direction: Forward
Steering: Centered

Power: 40
Duration: Unlimited
Next Action: option not active

continued on next page

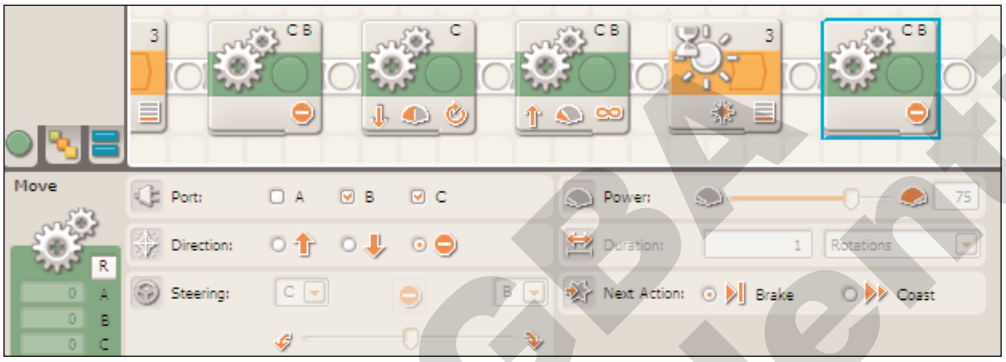
Module 13: Navigating a Maze with the Light Sensor *continued from previous page*

10. Add a light sensor block, and change the until option to less than 35 (<35).



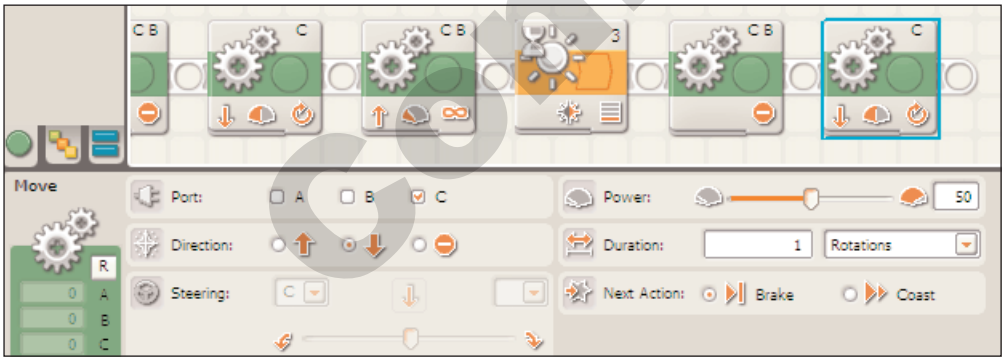
Control: Sensor
Sensor: Light Sensor
Port: 3
Until: < 35
Function: Generate light

11. Add a move block and set direction to Stop.



Port: B C
Direction: Stop
Steering: option not active
Power: option not active
Duration: option not active
Next Action: Brake

12. Add another move block, deselect motor B, set direction to backwards, and the power to 50.

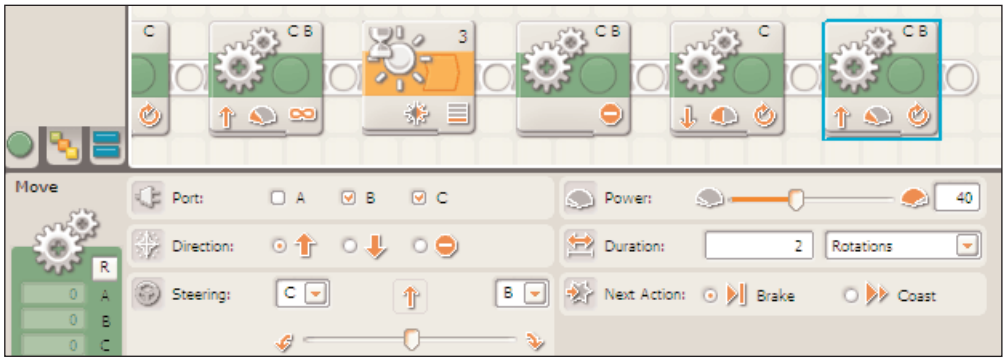


Port: C
Direction: Backward
Steering: option not active
Power: 50
Duration: 1 Rotations
Next Action: Brake

continued on next page

Module 13: Navigating a Maze with the Light Sensor *continued from previous page*

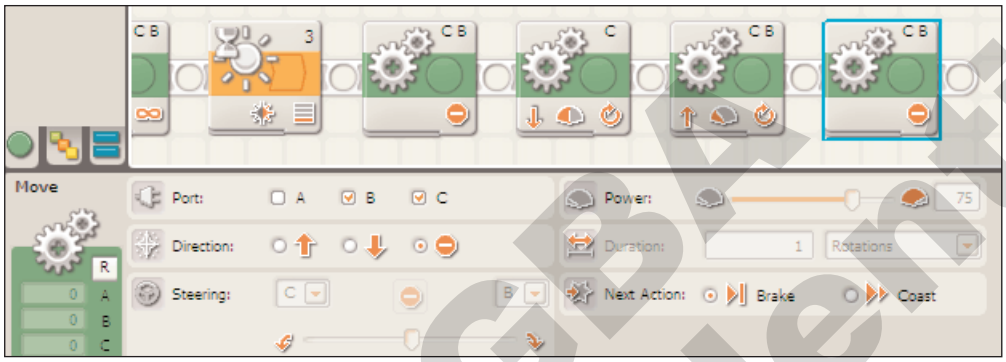
13. Add another move block, set the power to 40, and the duration to 2 rotations.



Port: B C
Direction: Forward
Steering: Centered

Power: 40
Duration: 2 Rotations
Next Action: Brake

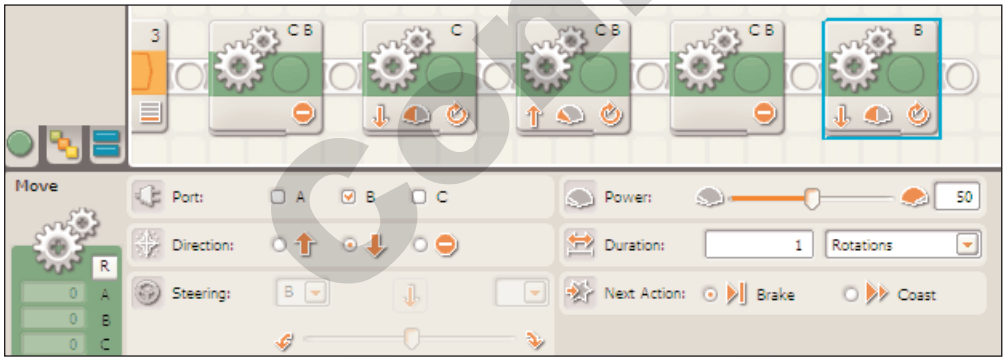
14. Add a move block and set direction to Stop.



Port: B C
Direction: Stop
Steering: option not active

Power: option not active
Duration: option not active
Next Action: Brake

15. Add another move block, deselect motor C, set direction to backwards, and the power to 50.



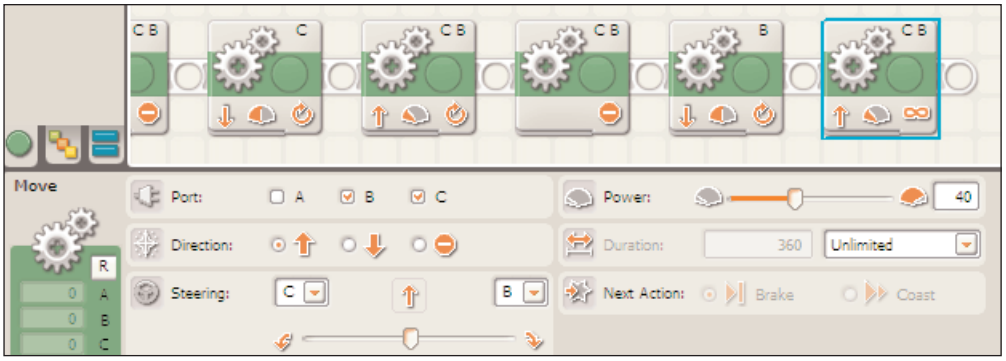
Port: B
Direction: Backward
Steering: Centered

Power: 50
Duration: 1 Rotations
Next Action: Brake

continued on next page

Module 13: Navigating a Maze with the Light Sensor *continued from previous page*

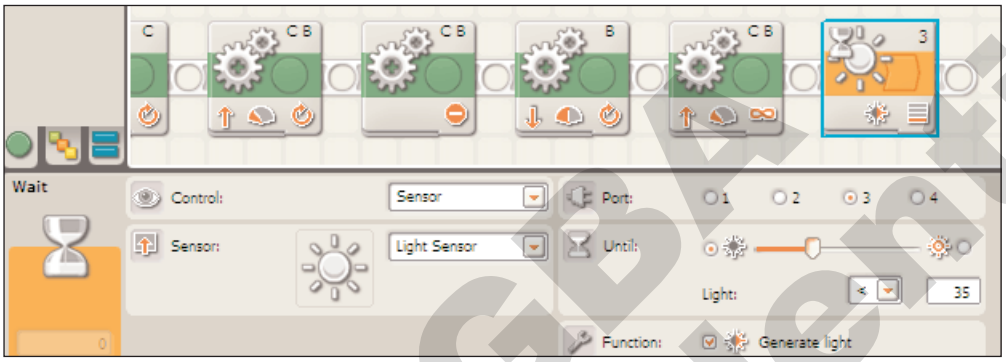
16. Add another move block, set the power to 40, and the duration to unlimited.



Port: B C
Direction: Forward
Steering: Centered

Power: 40
Duration: Unlimited
Next Action: Brake

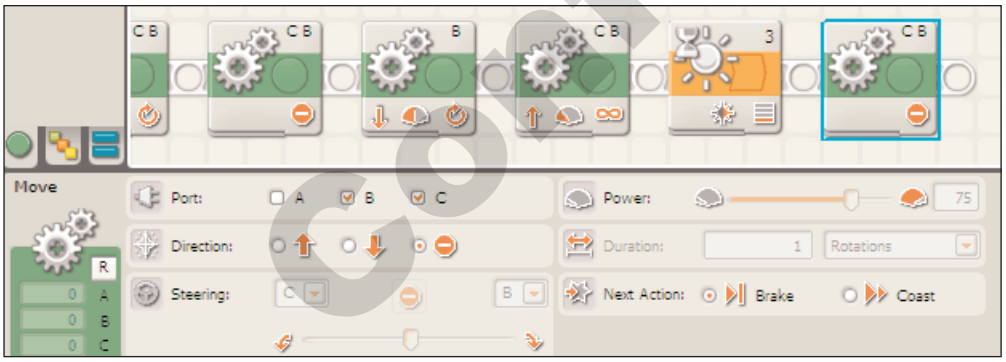
17. Add a light sensor block, and change the until option to less than 35 (<35).



Control: Sensor
Sensor: Light Sensor

Port: 3
Until: < 35
Function: Generate light

18. Add a move block and set direction to Stop.



Port: B C
Direction: Stop
Steering: option not active

Power: option not active
Duration: option not active
Next Action: Brake

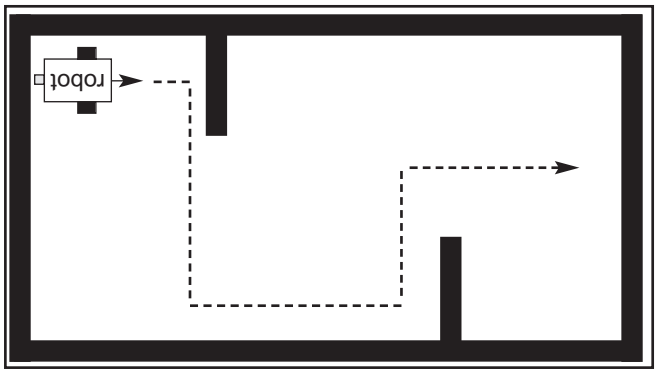
continued on next page

Module 13: Navigating a Maze with the Light Sensor *continued from previous page*

TEST THE PROGRAM

1.Download the program to the robot, then navigate to it on the robot.

2.Position the robot to the starting position of the maze, then start the program. The robot should successfully navigate the maze on the poster board without crossing any of the black lines.



GAME/CHALLENGES:

Once the robot is successfully able to navigate the sample maze, students should be shown a different maze and be tasked with designing and running a program that will successfully navigate the new maze.

Or students can design a maze of their own to challenge others in the class.

WHY ISN'T THIS WORKING?

If things are not working properly, check to make sure that:

1. The brick is powered on.
2. All wire connections are secure
3. The wires are connected to the correct ports.
4. The sensor is connected to the correct port
5. The proper program is selected, and the brick is currently running it.
6. The program is within a loop
7. The "Move" blocks are set to the proper rotations, power, port, or direction
8. Make sure the threshold selected for the sensor block is correct, and that the comparative is set to less than.

END

Appendix II: Student Worksheets

The following pages contain worksheets to enhance the lessons in some of the modules. Each worksheet is meant to provide a fun challenge to the students and reinforce various lessons present in the modules. You may choose to use these worksheets at your discretion when they apply to a specific module.

Worksheets

continued on next page

Game Builders Academy - Robotics Level 1 **Your Name:**

Worksheet #1 Senses and Sensors - Objectives:
To understand how robots take in information about their surroundings. To compare and contrast this with how humans take in information about their surroundings; To communicate the results of our comparisons.

Complete the table below. In the blank boxes, write how a human or robot takes in each type of sensory input (if a robot or human does not have one of the senses listed, leave the box blank):

Sense	Human	Robot
Sight		
Sound		
Touch		
Smell		
Taste		

Answer the questions below:

1. How many different senses does your robot have? _____
2. How many different senses does a human have? _____
3. What senses to they both have? _____
4. Draw a Venn diagram to illustrate your results.

   Use the open space to draw your diagram ↗

Write a short summary of your results and conclusions (use the back of this paper if you need more space):

Game Builders Academy - Robotics Level 1 ANSWER SHEET

Worksheet #1 Senses and Sensors - Objectives:
To understand how robots take in information about their surroundings. To compare and contrast this with how humans take in information about their surroundings; To communicate the results of our comparisons.

Complete the table below. In the blank boxes, write how a human or robot takes in each type of sensory input (if a robot or human does not have one of the senses listed, leave the box blank):

Sense	Human	Robot
Sight	Eyes	Light Sensor
Sound	Ears	Sound Sensor, Ultrasonic Sensor (The ultrasonic sensor uses ultrasonic sound waves to measure distance)
Touch	Skin, Fingers, Etc.	Touch sensor
Smell	Nose	None
Taste	Tongue	None

Answer the questions below:

1. How many different senses does your robot have? __4___
2. How many different senses does a human have? __5____
3. What senses to they both have? _ Sight, Sound, Touch _____
4. Draw a Venn diagram to illustrate your results.

   Use the open space to draw your diagram ↗

Write a short summary of your results and conclusions (use the back of this paper if you need more space):
The Venn diagram should have distance on the robot only side, sight sound and touch in the center, and smell and taste on the human only side.

Game Builders Academy - Robotics Level 1 **Your Name:**

Worksheet #2 Basics of Using the Mindstorms Brick and Sensors - Objectives:
To learn how to connect the various sensors to the Mindstorms brick; To learn how to navigate and use the brick's on-screen menus; To learn about the sensors by playing with the Try-Me programs on the brick.

Use the words listed below to label the ports of the brick (you may draw pictures instead of using words):

Light

Touch

Ultra-Sonic

Sound

Motor

To Computer