

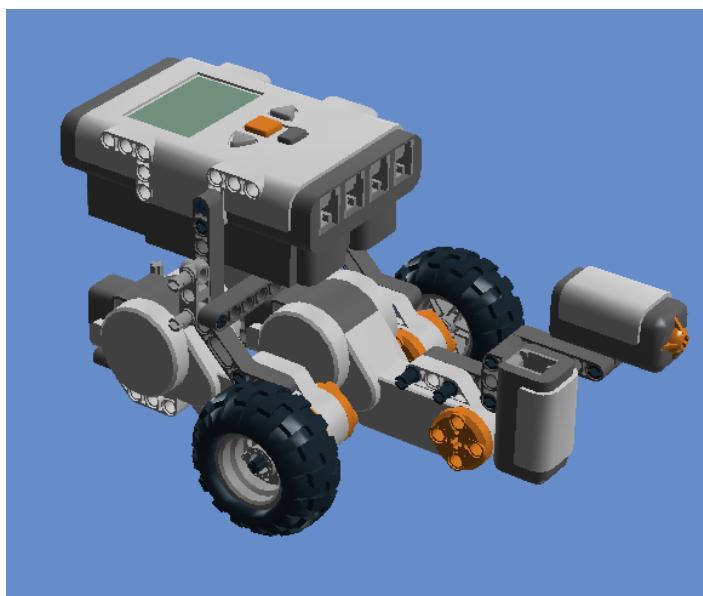
What is a Robot?

A Robot is a machine that can be programmed to do a task without being directly controlled by a human being.

There are two parts to a robotic system.

Hardware: This is the mechanical robot. The motors, gears, wheels, parts, computers that make up the machine. This is the part you build.

Software: This is the instructions for the tasks and actions the robot will do. This is the part you write. MINDSTORMS NXT-G is used to create the software that controls the actions of the robot hardware. NXT MINDSTORMS software is a visual/icon system of assembling instructions. The flow of direction usually moves from left to right. This software is an example of a compiled program. Programs you write in NXT must be compiled and downloaded to the NXT Brick before the Robot will perform the program. (Scratch and Python are "Interpreted Languages.")

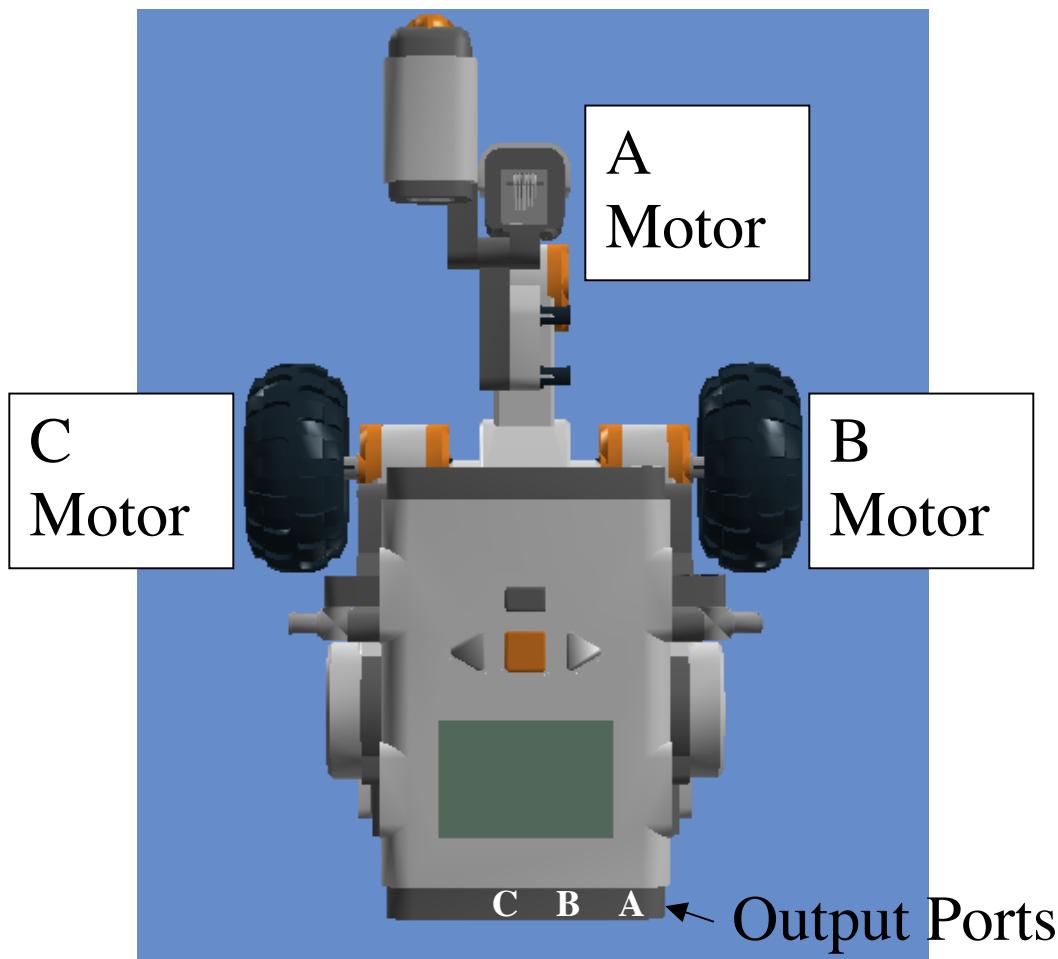


Robot Base: Outputs

(Building Directions:

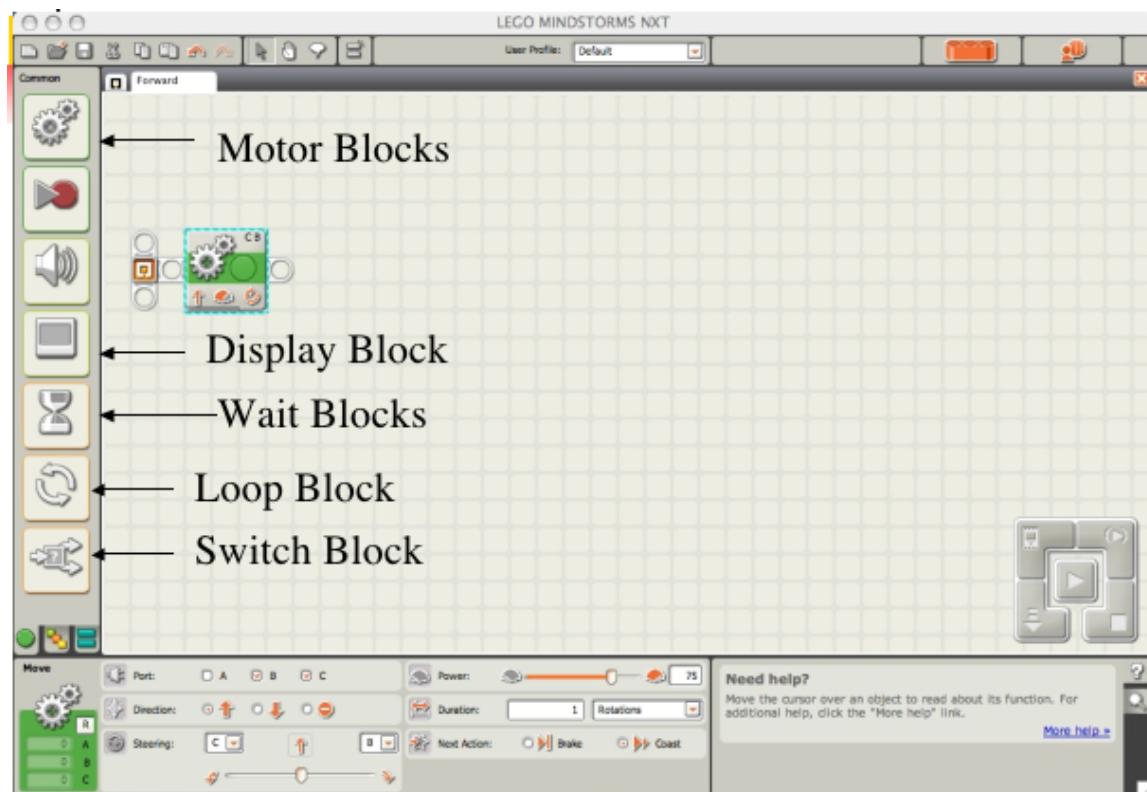
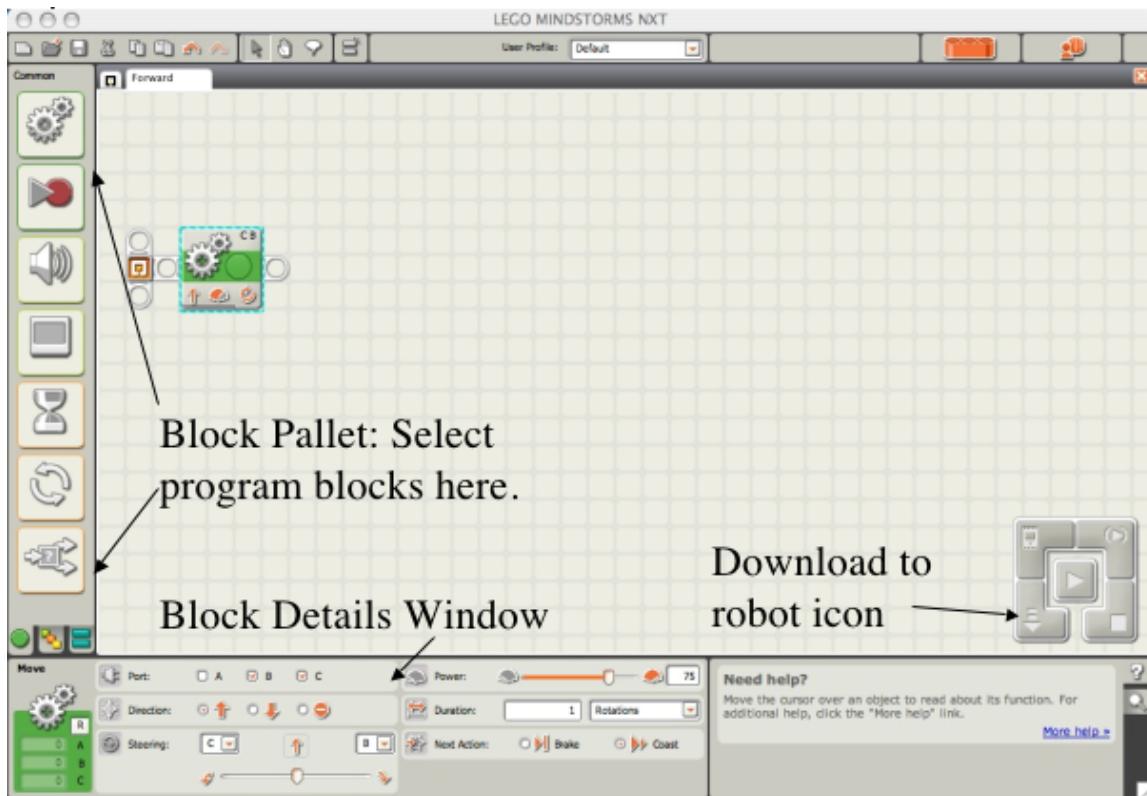
http://www.nebomusic.net/michaud_TriBot_build_instructions.html)

Motors: **C = Left**
 B = Right
 A = Center



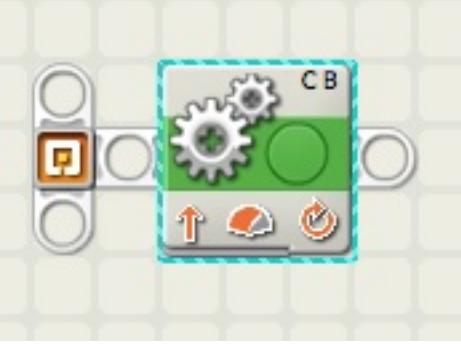
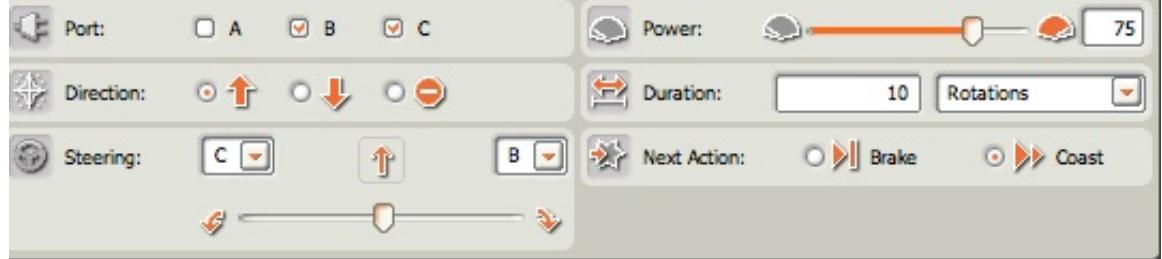
Inputs: **1 = Touch Sensor**
 2 = Sound Sensor
 3 = Light Sensor
 4 = Ultrasonic Sensor

NXT – G Programming Interface:

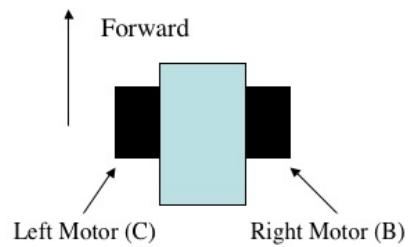


Using Motor Blocks:

Movement Blocks direct motors A, B, or C to rotate clockwise or counterclockwise. The motors have sensors inside them that can measure how much they rotate. Motors C and B usually are used for the drive wheels on a vehicle Robot. You may change the following variables on a motor on the details window.

 <p>This icon creates a new movement block.</p>	 <p>This icon represents the movement block inside the program.</p>
 <p>This is the detail window for the selected block. You may change the following variables:</p> <ul style="list-style-type: none">• Which motors are controlled by the block (A, B, or C) (B and C are usually the drive motors in vehicle robots.)• Direction• Steering (B and C working together)• Power (Speed)• Duration (How Long - How Far)• Next Action (Brake or Coast)	

Swing and Point Turns: Basic Two Drive Wheel Design



A. Swing Turn -> One wheel turns and robot pivots on opposite wheel

Right Swing Turn => Left Motor (C) Rotates Forward

Left Motor (C)

Right Swing Turn => Left Motor (C) Rotates Forward

Port: C
Power: 75
Direction: Up
Steering: C
Duration: 1 Rotations
Next Action: Brake

Left Swing Turn => Right Motor (B) Rotates Forward

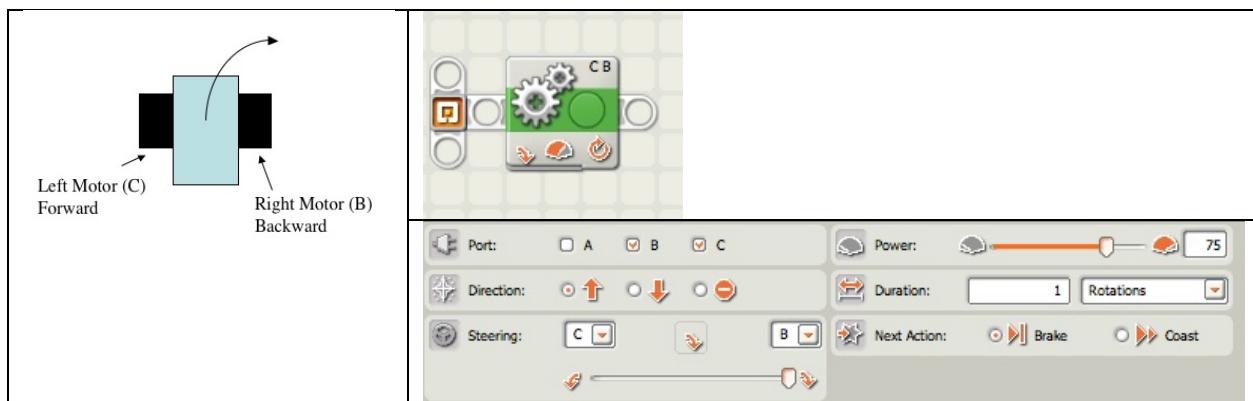
Right Motor (B)

Left Swing Turn => Right Motor (B) Rotates Forward

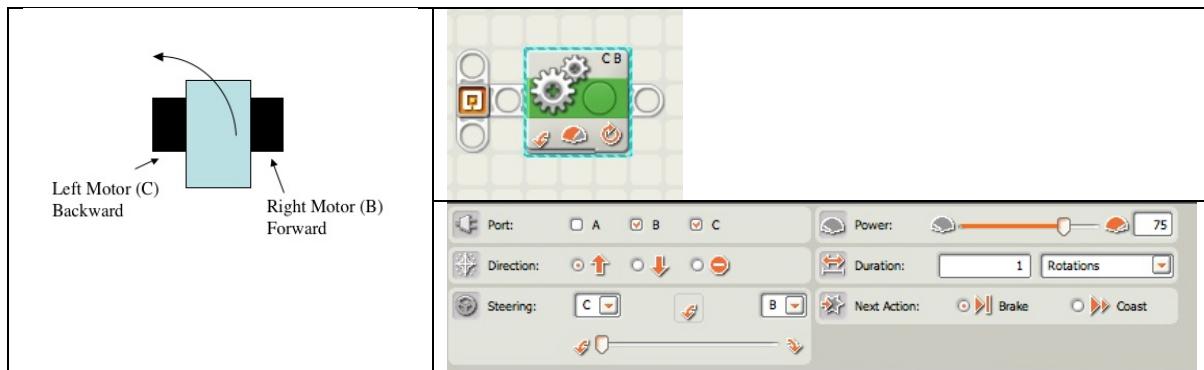
Port: B
Power: 75
Direction: Up
Steering: B
Duration: 1 Rotations
Next Action: Brake

B. Point Turn => Both wheels rotate in opposite directions and robot pivots over center point. (Use 1 Motor C/B block with directional arrow pushed all the way to left or right)

Right Point Turn => Left Motor (C) Rotates Forward while Right Motor (B) Rotates Backward (Motor C/B block with directional arrow pushed all the way to the right.)



Left Point Turn => Left Motor (C) Rotates Backward while Right Motor (B) Rotates Forward (Motor C/B block with directional arrow pushed all the way to the left.)



Robot Project 1: Drive in a Square

Objectives:

Select Movement Block parameters for motors selection, direction, distance, power, and turning.

Use Swing turns to change direction.

Use loop blocks to repeat commands.

Project:

Write an NXT-G program that will drive your robot in a square.

Use rotations to measure distance.

Use a loop block to keep the program short.

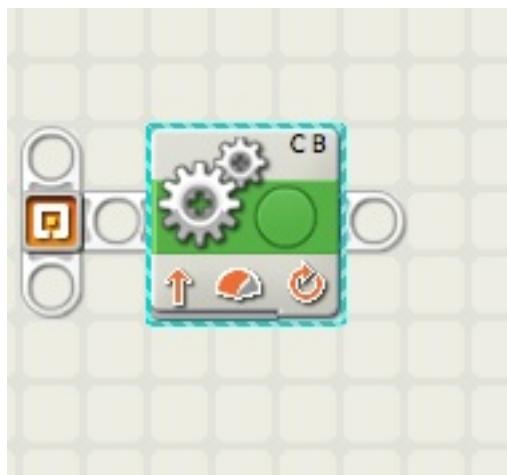
Download and run your program to the robot.

Process:

1. Open the Mindstorms NXT-G software. (Double click the orange square icon.)



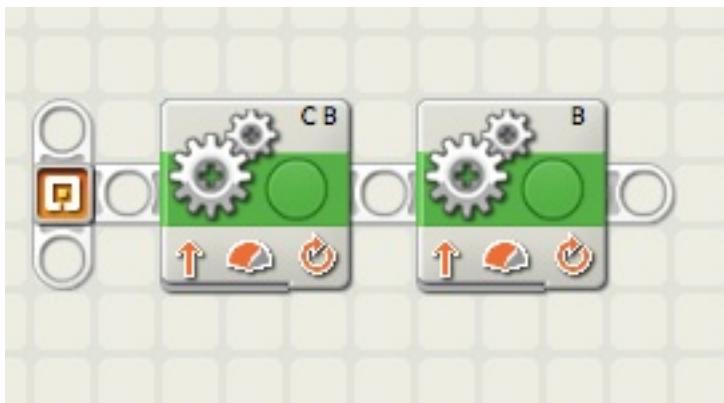
2. Drag a Move block and connect it to the first part of the program



3. Change the parameters for the block: (Located on bottom of window)
- Motor check B and C
 - Direction to Forward
 - Power to 75
 - Duration to "4 Rotations."



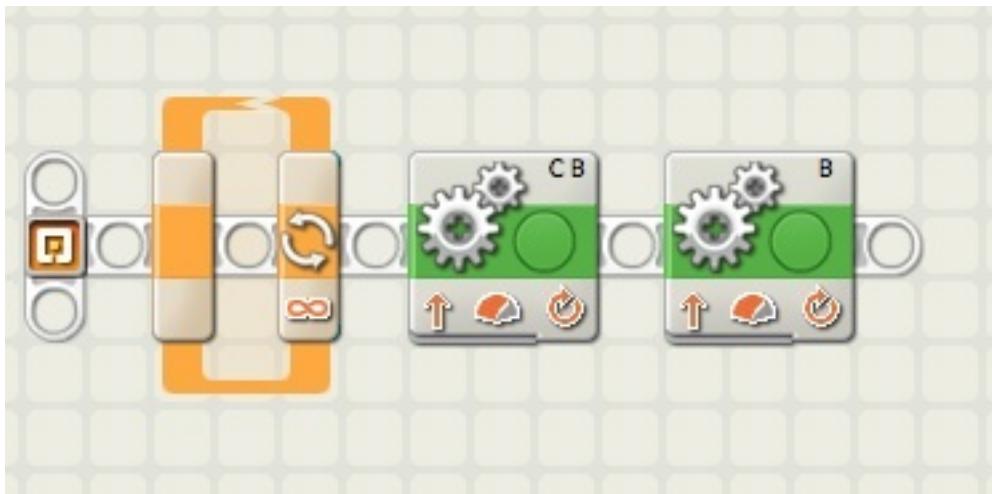
4. Drag a new Move block and set it next to the first block.



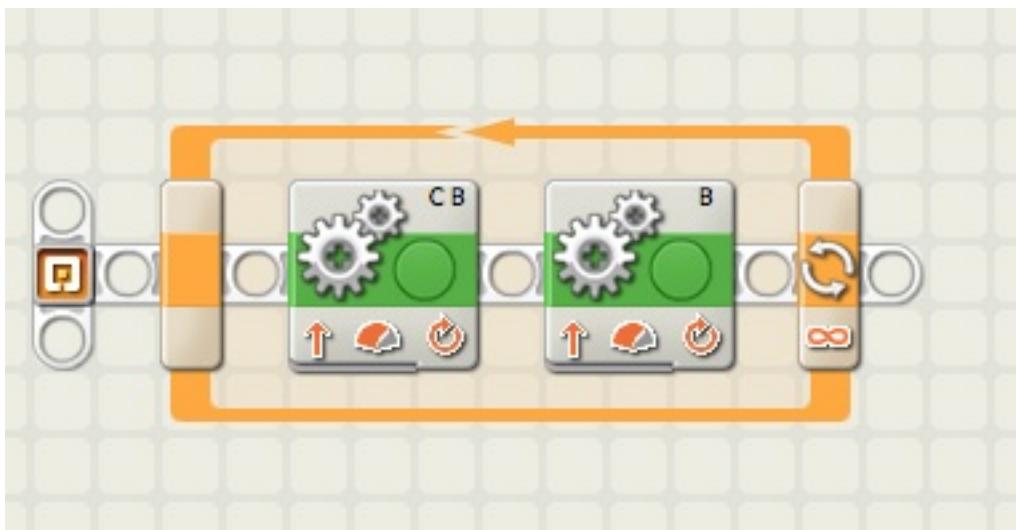
5. Change the parameters for this new block:
- Motor check to B only
 - Direction to Forward
 - Power to 75
 - Duration to "1 Rotations"



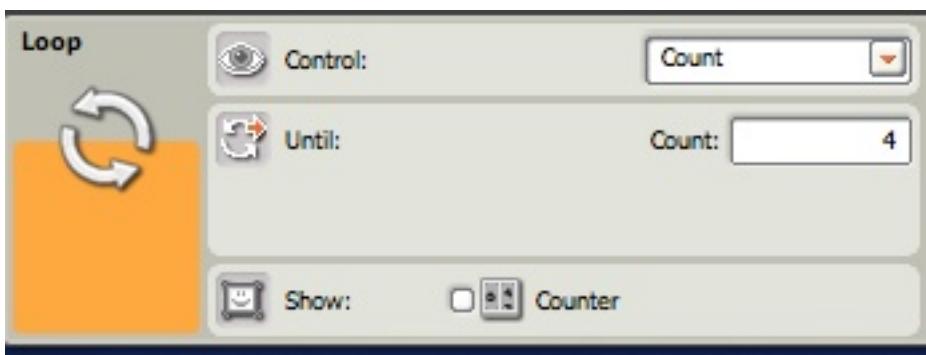
6. Drag a Loop Block and set it before the two movement blocks:



7. Put the Move Blocks inside the Loop.



8. Set the Loop count to 4. (Why do we set the Loop count to 4?)



9. Save your program as "LastnameSquare."
10. Carefully plug the USB cord into the top of the NXT-G Brick on the robot.
11. Plug the other end of the USB cord into the computer.
12. Turn on the Robot. (Press the Orange button once.)
13. Click the Lower Left Download Icon. (Looks like a pointing down arrow)



14. The robot will beep. After the beep and the downloading window closes, unplug the robot.
15. Take the robot to the table.
16. Select "Software" on the Robot.
17. Scroll until you see your program "LastnameSquare."
18. Put the robot on the table and press the Orange button to start and run your program. Press the Grey rectangle to stop the robot.
19. Make adjustments to your program as needed to make a square.

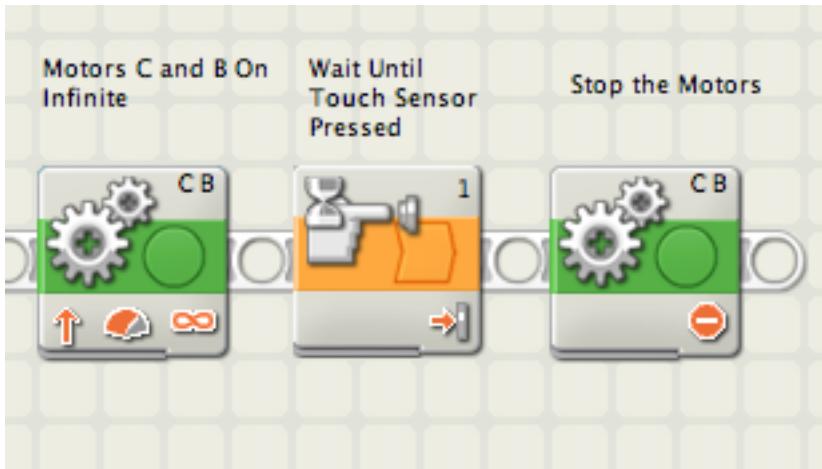
Using Wait Blocks for “Stop Programs”

One of the basic uses of sensors is to have the robot:

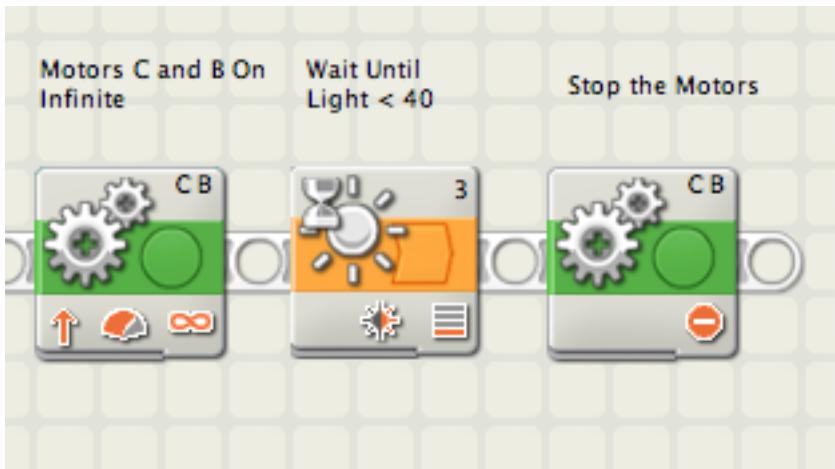
1. Start an action (running motors)
2. Wait until a Sensor Condition is met
3. Stop the Motors

We will call these “Stop Programs” and have 5 examples:

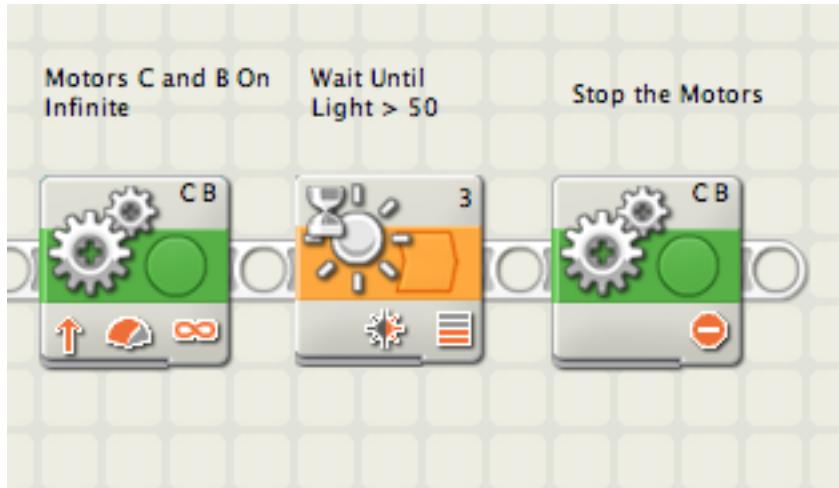
Example 1: Touch Stop



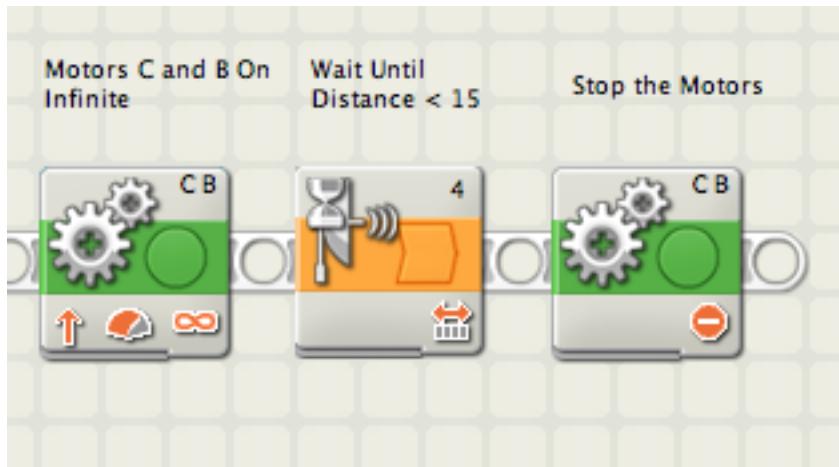
Example 2: Dark Stop



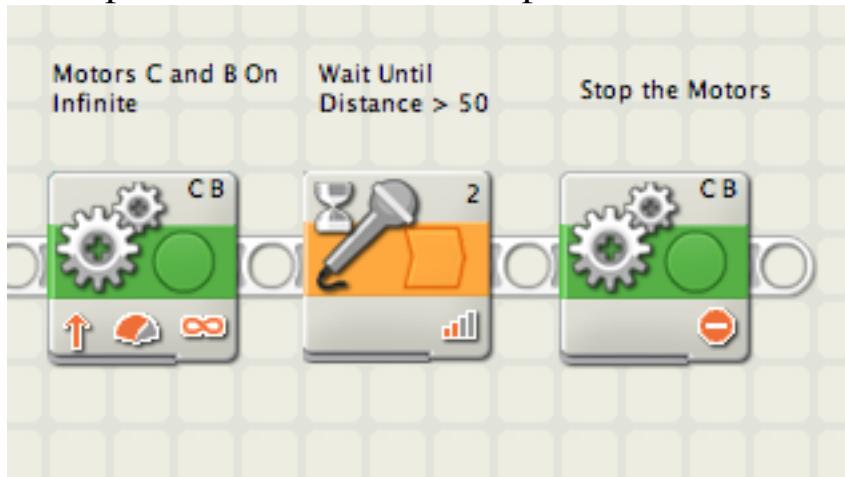
Example 3: Light Stop



Example 4: UltraSonic Stop



Example 5: Sound Sensor Stop



Basic Line Follow Instructions

Objectives:

Use Conditional Switch Block to evaluate Input Values of Light Sensor

Use Loop Block to Repeat Switch Block

Project:

Write an NXT-G program that will have your robot follow a line using a simple Algorithm.

Key Blocks:

Switch Block:

Similar to an “if else” statement. Block checks a condition and directs the program to a sequence of actions depending on the condition. (Light Vs. Dark, Loud Vs. Quiet, Near Vs. Far, True Vs. False)

Example:

If Light_Sensor < 50

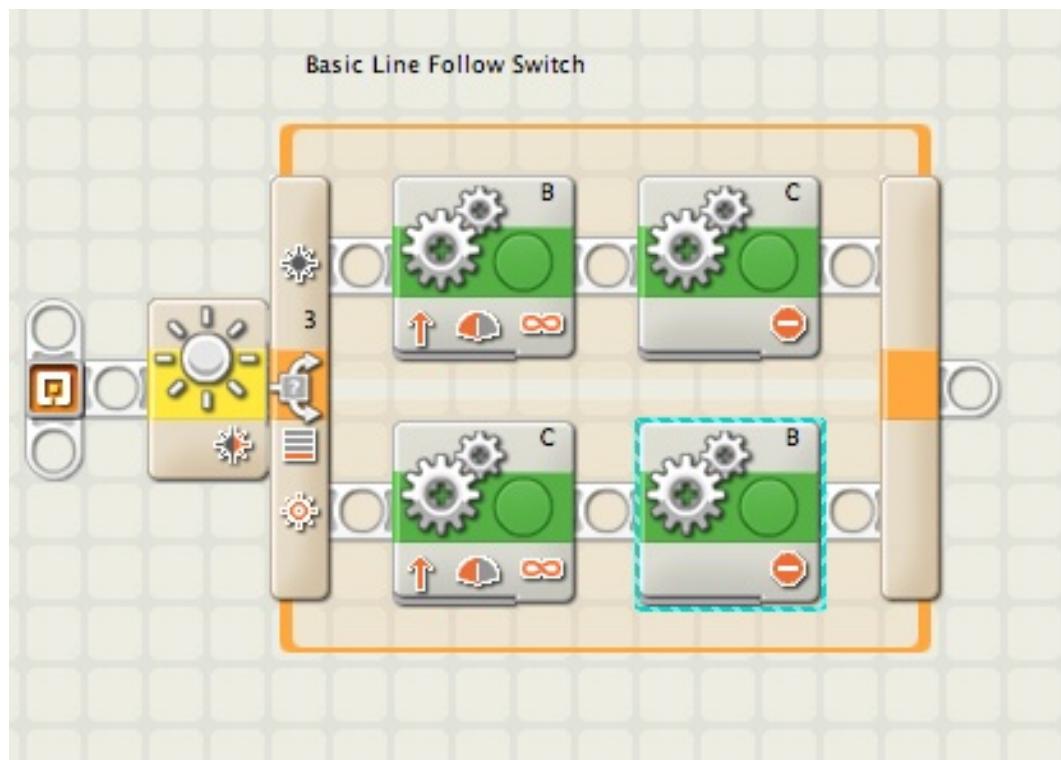
 Motor B On 50

 Motor C Off

Else if Light Sensor > 50

 Motor C On 50

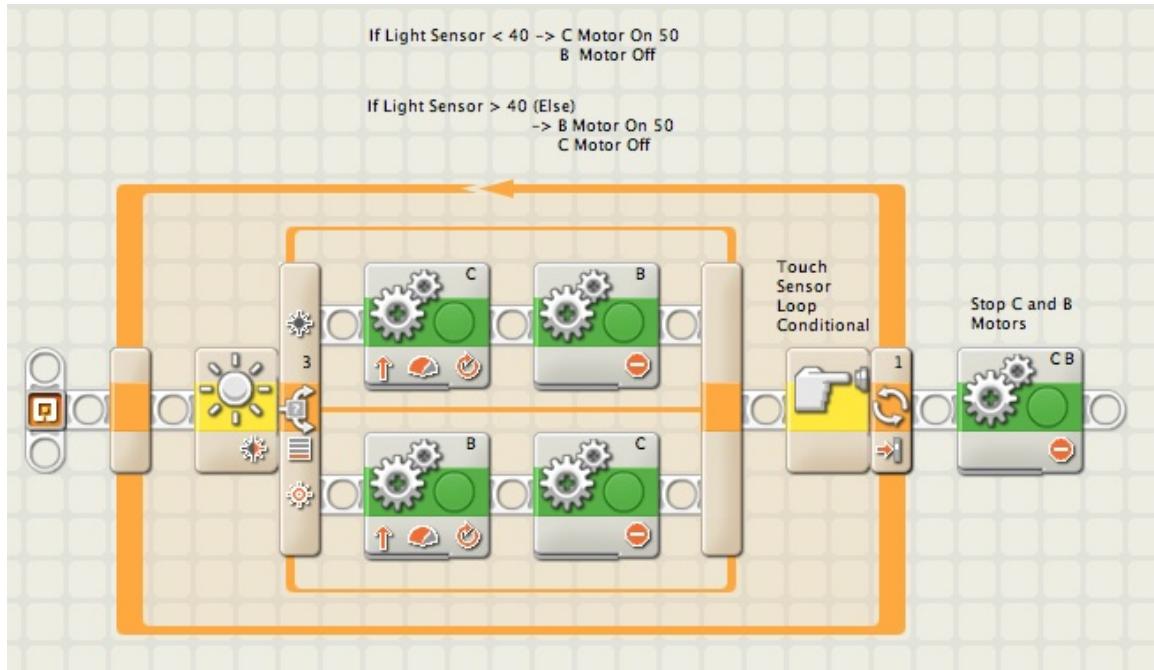
 Motor B Off



Loop Block:

Repeat a sequence of steps until a condition is met. (Sensor, Count, Value, Forever)

Example:



Process for Creating Simple Line Follow:

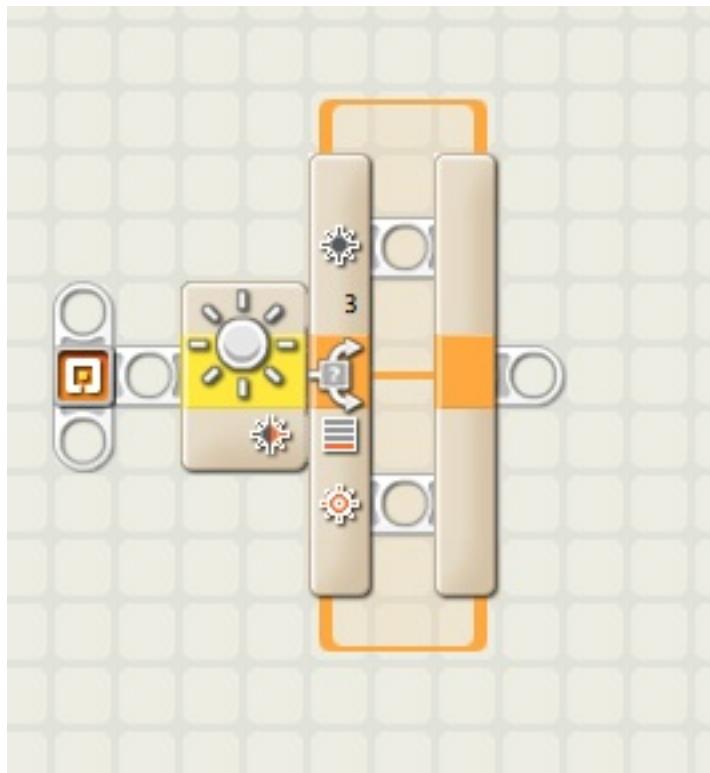
1. Open the Mindstorms NXT-G software. (Double click the orange square icon.)



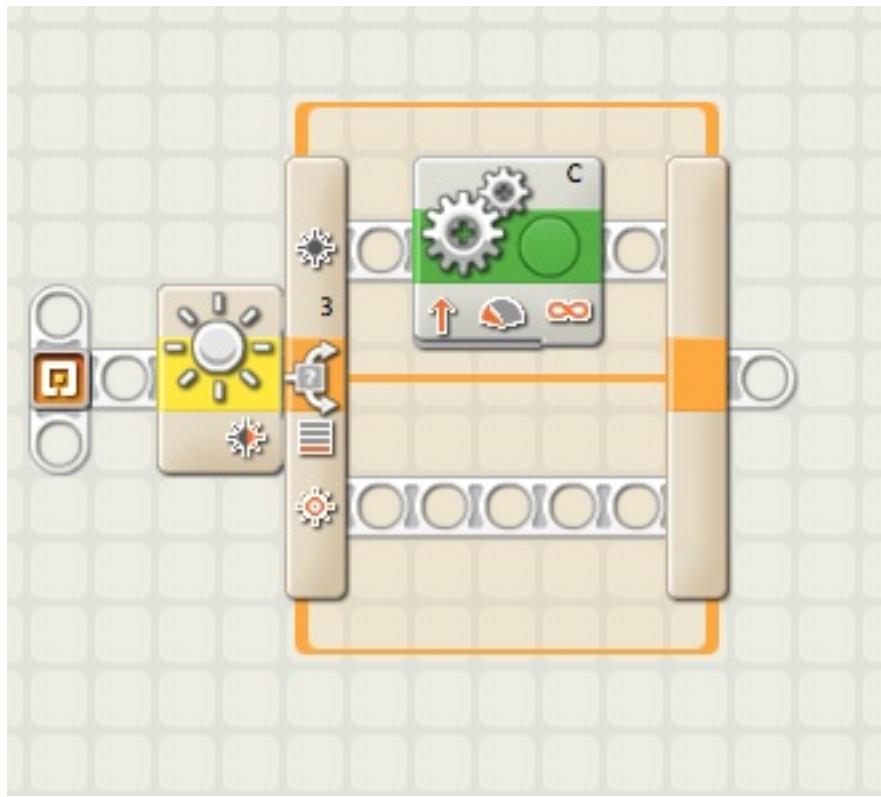
2. Name your program "LastnameBasicLineFollow" and Click "GO."

3. Drag a Switch Block to the Programming Area.

- a. Set the Sensor to "Light Sensor"
- b. Set the Compare to < 40.



4. Drag a Motor Block to top inside of the Switch Block

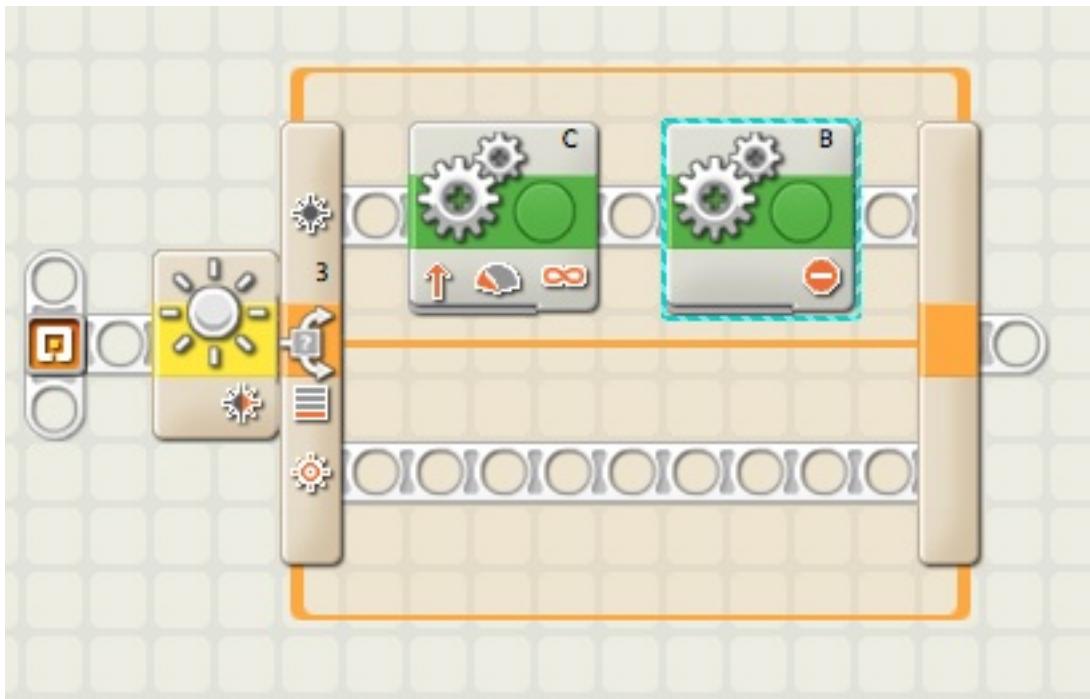


5. Set the Motor Block Parameters to:

- Motor Selection = C
- Power = 40
- Duration = Unlimited



6. Drag another Motor Block and put it on the top inside of the Switch Block.

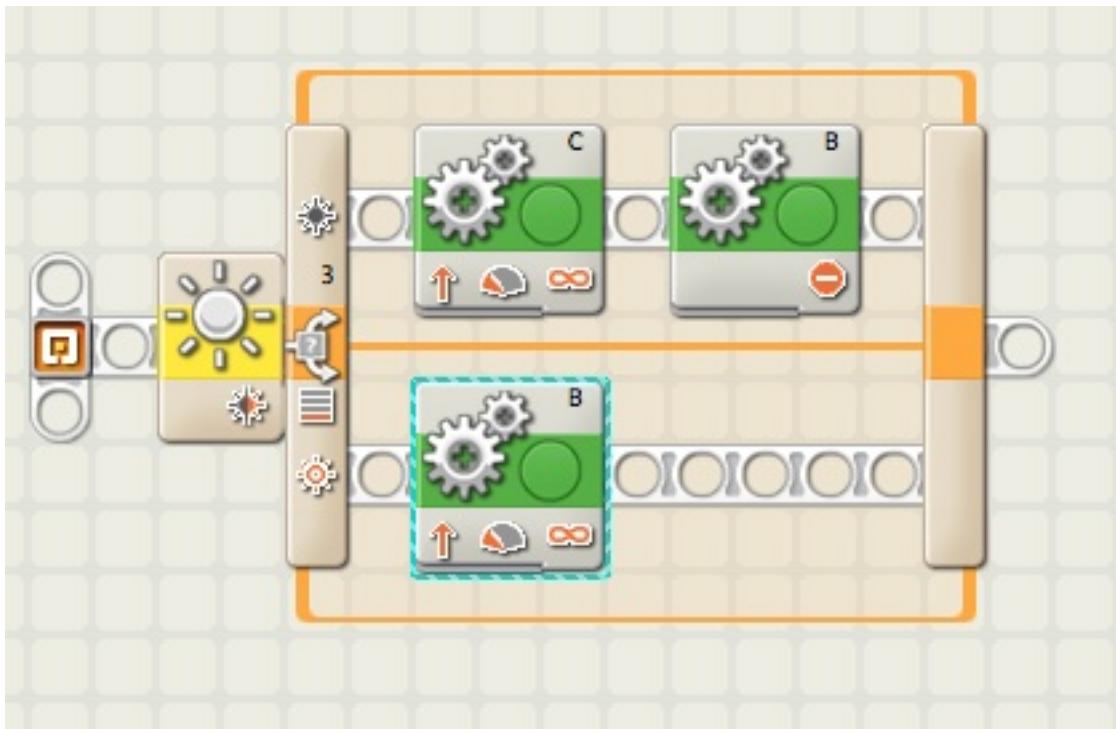


7. Set the Motor Block Parameters to:

- Motor Selection = B
- Direction = Stop



8. Drag another Motor Block and put it on the bottom inside of the Switch Block.

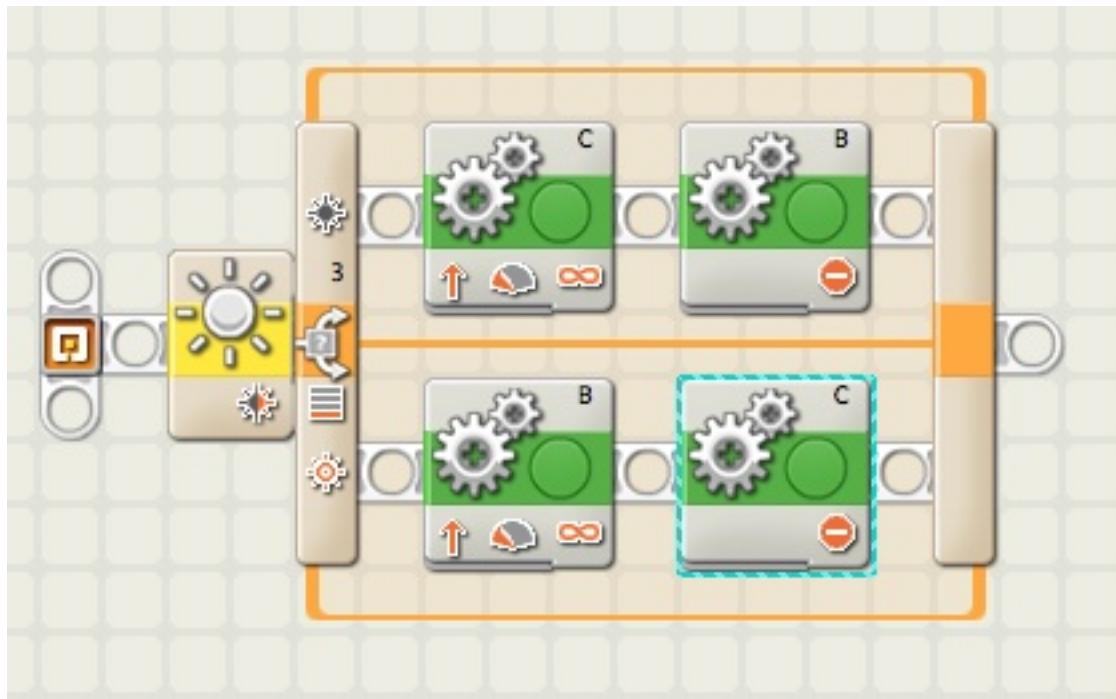


9. Set the Motor Block Parameters to:

- Motor Selection = B
- Power = 40
- Duration = Unlimited



10. Drag another Motor Block and put it on the bottom inside of the Switch Block.

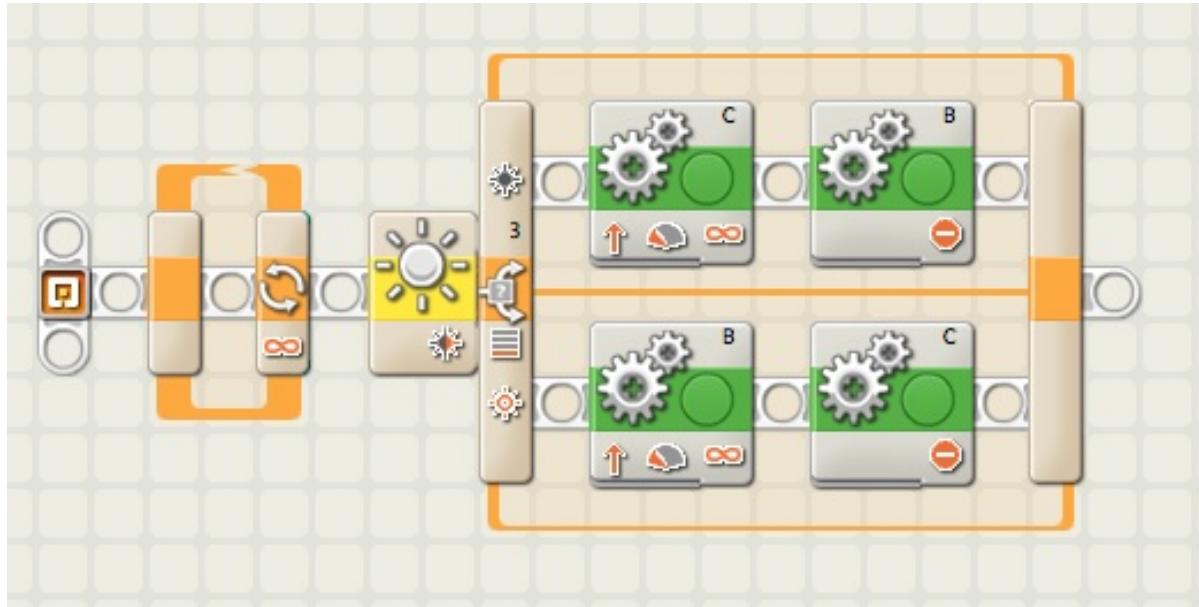


11. Set the Motor Block Parameters to:

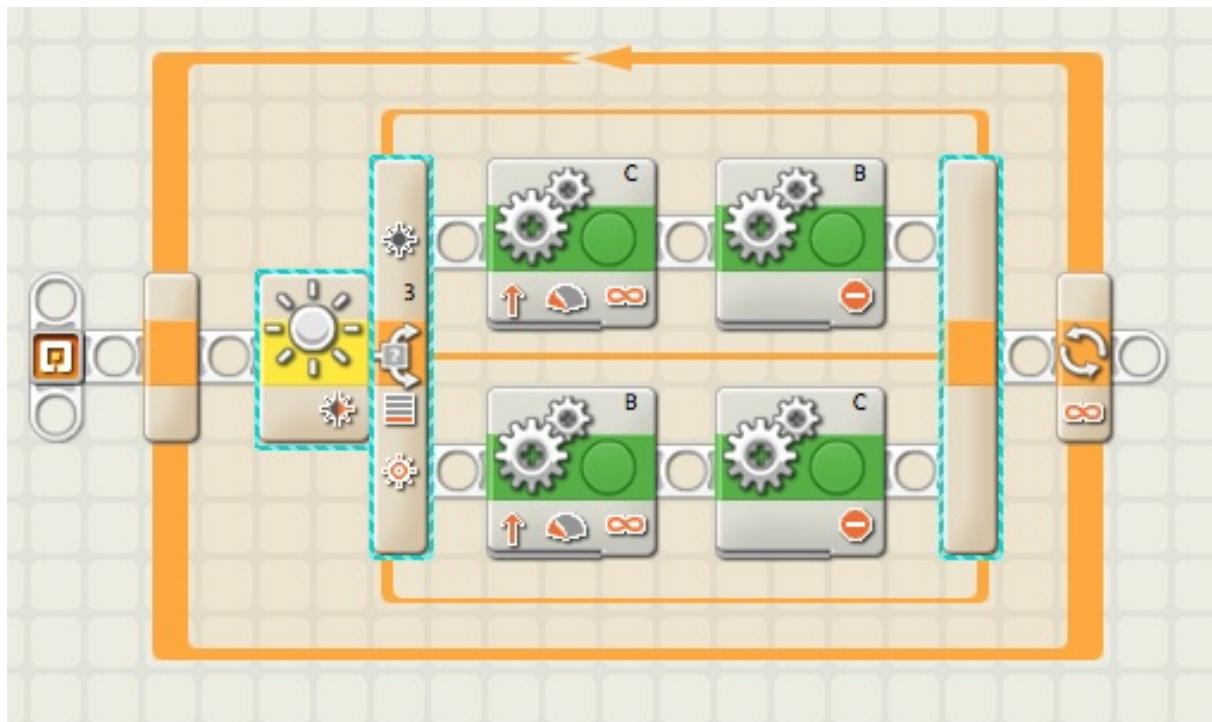
- Motor Selection = C
- Direction = Stop



12. Drag a Loop Block into the programming area place it to the left of the Switch Block.



13. Drag the Switch Block inside the Loop Block.



14. Save Program. (File -> Save)

15. Download and test with the Robot.

Notes:

- A. You will need to work with the Light Value Numbers and Motor Speeds to "tune" the line follow to track the line depending on light conditions and path of line. A lower motor speed will allow the robot to track a "more curvy" line.
- B. This program will track the Right Hand side of the line. How can you change the program to track the Left Hand side of the line?
- C. The robot makes a wiggly track to follow the line. How can you modify the program to allow for more efficient motion with the Robot.

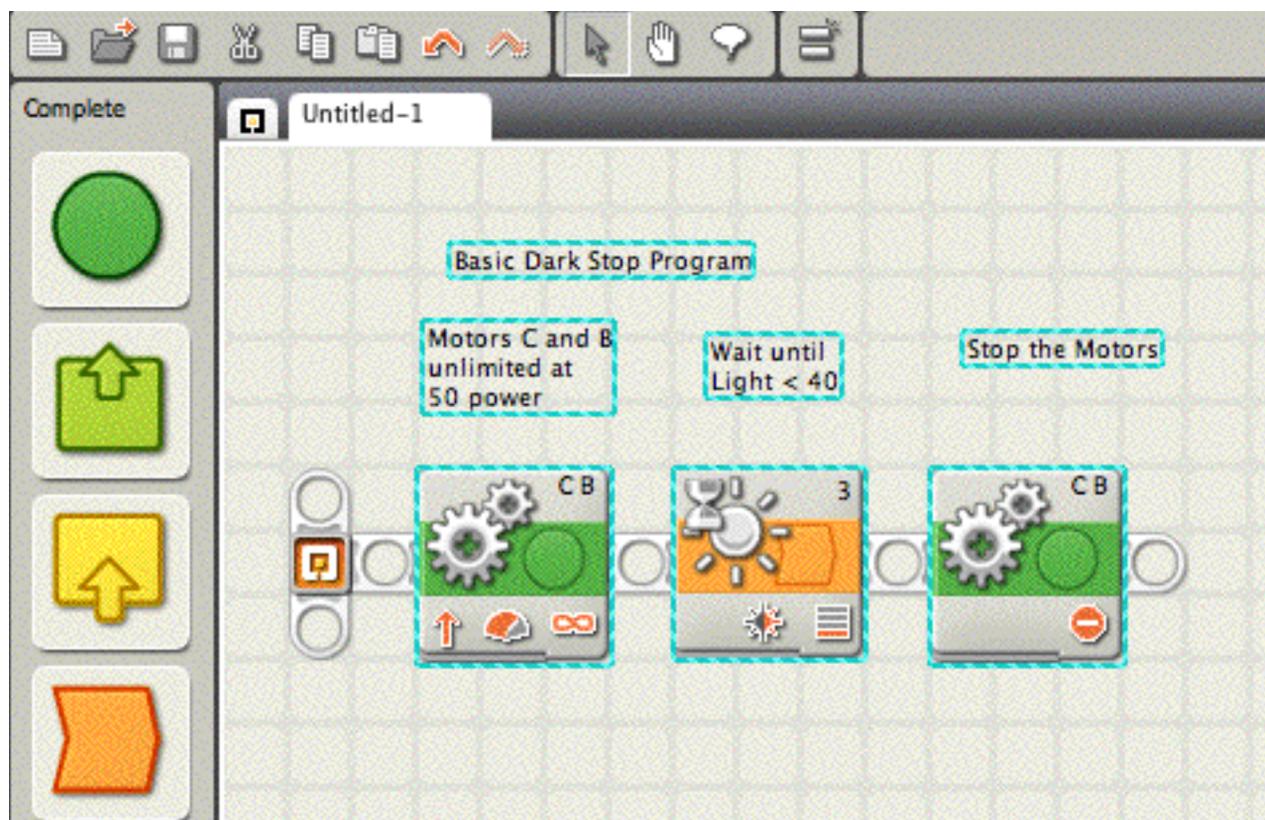
NXT-G and MyBlocks

MyBlocks: A MyBlock is a user defined segment of code that resides within one programming block. The user can take multiple blocks and "group" them together into a block that can be reused in different programs. This is similar to defining methods or functions in other programming languages.

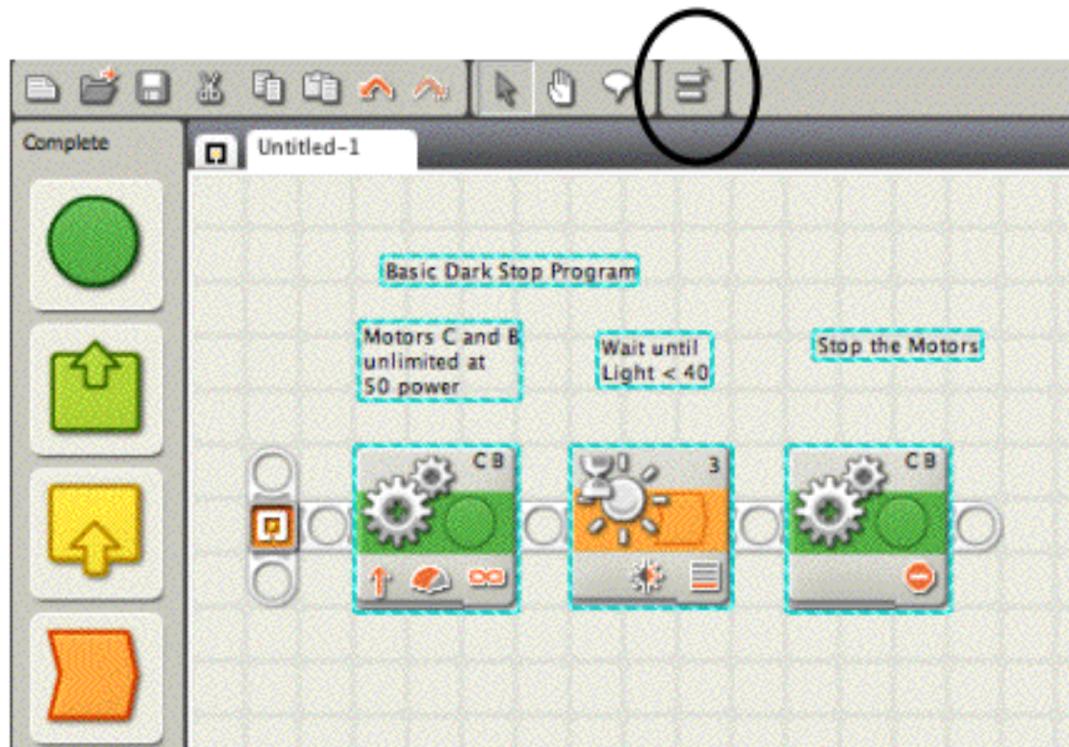
Generally, if you are using a sequence of blocks more than twice in a program - you should turn it into a MyBlock. (Examples include Dark Stops and Line Following).

Directions for making a MyBlock.

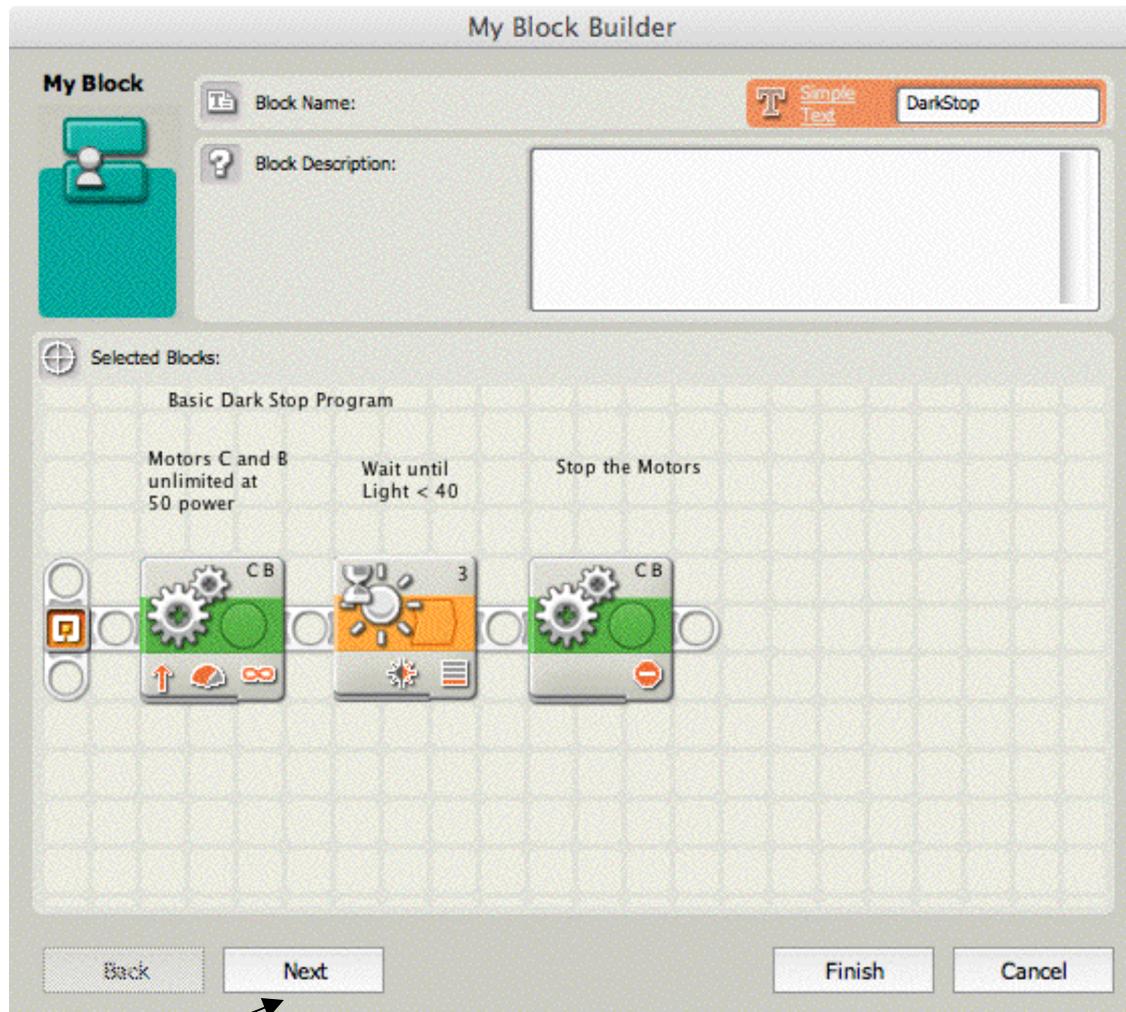
1. Select the Blocks you wish to group together. (In this example, we are going to group a "Dark Stop" routine.



2. Click the MyBlock icon.

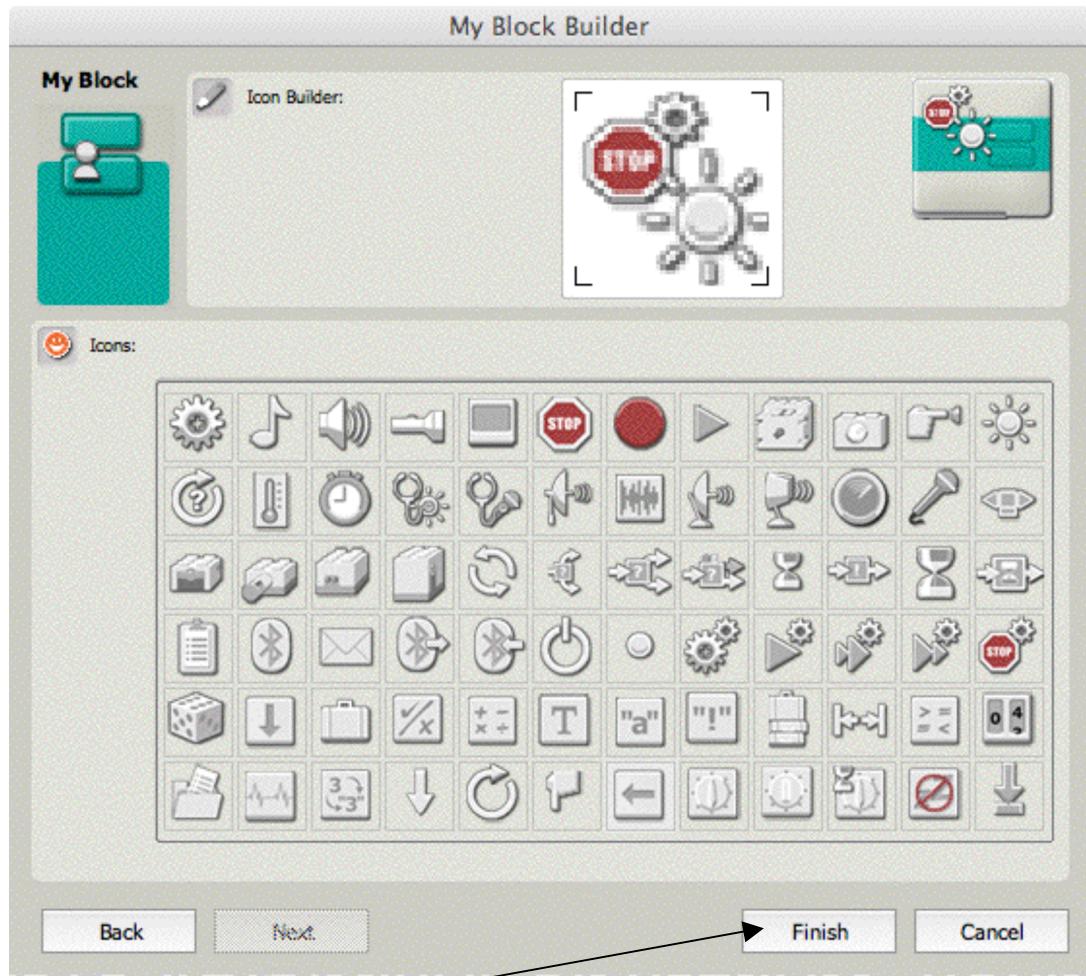


3. Name your MyBlock - use a name that describes the MyBlock action. Example: "DarkStop"



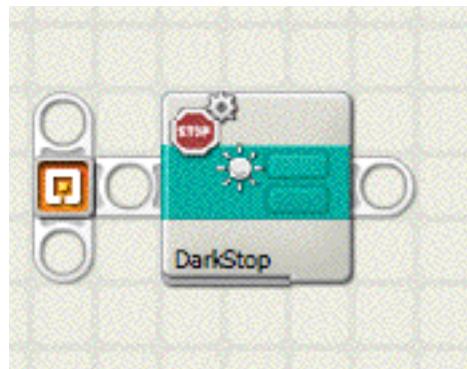
4. Click Next.

5. Drag icons to "decorate" your block. This will give you a visual cue as to the purpose of the block. In this example, we will use a Stop Sign and a Light for "Dark Stop".

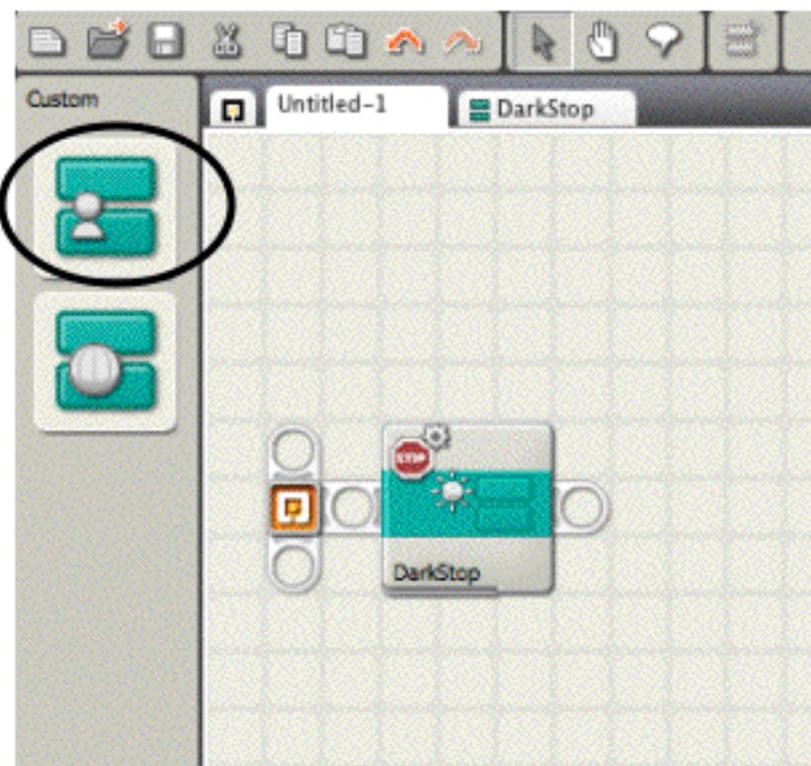


6. Click "Finish"

7. The computer saves the MyBlock and puts it back into the original program.



8. You can access the MyBlock from the MyBlocks menu.



Programming Challenges for Driving:

Drive a Rectangle Activity:

<http://www.nebomusic.net/NXT-G-DriveARectangle.html>

Maze #2

Maze #1

Distance and Turns Worksheets

Programming Challenges for Sensors and Loops:

Wait Block Programs:

Touch Stop

Dark Stop

Ultrasonic Stop

Line Following:

<http://www.nebomusic.net/NXT-G-BasicLineFollow.html>

Mine Sweeping (Robot Course 2011)

<http://www.nebomusic.net/RobotTrainingCourse2011.html>

Green City

Rescue Soldier

Advanced Missions:

Avoider Robot: Use Ultrasonic Sensor or Touch to have robot move about the room and avoid walls or objects.

Intermediate Line Follow

PenBot: Draw a Square, Face

MusicBot: Program machine to play “Hot Cross Buns”

Turns Worksheet (See next page)

Create “My-Blocks”

Use NXT-G to program a PID line following program.

Turn Angle Chart

For your robot, figure out the Motor parameters for a 90 degree and 45 degree swing and point turns for right and left.

	Motor(s)	Direction	Degree Rotation
Swing Left 90			
Swing Right 90			
Point Left 90			
Point Right 90			
Swing Left 45			
Swing Right 45			
Point Left 90			
Point Right 90			