

BadgerBOTics Camp

Building on Talented Students

RoboLAB 2.9.3 Quick Sheet #4



Robot Functions - Basic

My robot is broken! It's not doing what I told it to do!

The biggest challenge for the novice robot programmer is being able to break a complex task into individual “action blocks”. As a human, you do 99% of your physical actions without conscious control.

A small child spies a window full of candy... in an instant their pupils go wide, they inhale sharply while at the same time turning toward the window. Their arms and legs coordinate to break into a run, they almost slam into the window but just before impact the arms come up to press against the glass, their legs brace to slow the body to rest, and the vocal cords and lungs work together to scream their only conscious thought, “CANDY!!”

Your robot can't do that... infact, it can really only do (or think) one thing at a time. So this means that you, as the programmer, have to break each mission down into individual tasks, and then every task into actual thoughts the robot will have as it completes the mission.

So is your robot really dumber than a 2 month old?? Not really, the NXT can understand hundreds of separate commands (icons in a RoboLab program) once you download its language (firmware) into its brain. Also, unlike a 2 month old, your NXT will always do EXACTLY what you told it to do. So when it doesn't work correctly... What did you tell it to do??

MOVE, WAIT, STOP:

There are only two actions the robot performs during MOST missions.

The robot, or a part of the robot, moves in one direction.

The robot moves in a curved path (turns).

Both actions reduce to the same sequence of events...

Motors are turned on...

They stay on for some amount of time, or until something happens...

The motors turn off...

BadgerBOTics Camps

Benjamin J. Senson, Camp Director

Memorial High School, 201 S. Gammon Road, Madison, WI 53717

MOVE IN ONE DIRECTION:

THE ENTIRE ROBOT MOVES IN A STRAIGHT LINE.

Because NXT motors mount in the same direction on the robot, rotating both motors in the same direction will lead to motion in one direction. Writing a RoboLab program to move in a straight line requires the following steps... in order.

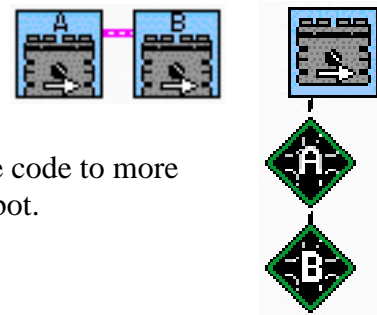
Begin the program, give it a name...

Remember that the NXT start icon is on the NXT submenu
Use the text tool to change the name from "rbl" to something with only 6 characters, and that has meaning during the competition, or at a later time.



Turn on both motors...

We want them to run in the same direction...at the same time.
Remember that computers think really fast, so two icons that are side-by-side run "at the same time" for all practical purposes.
Using one motor icon instead of two is a choice, but enhances the code to more closely match a verbal "command" that you would give to the robot.



Wait for...

In this case it is the passage of time.
Clocks are on the WAIT FOR submenu.
Add a text box from the MODIFIERS submenu to control how many thousandths of a second that the robot will continuously think about time. Use the text change tool to input the number. The robot will keep doing whatever it is already doing until it stops thinking about time. So the motors keep running for 500/1000 (or 1/2) second in this case.



Stop both of the motors...

Stop on a dime... slam on the brakes... STOP!

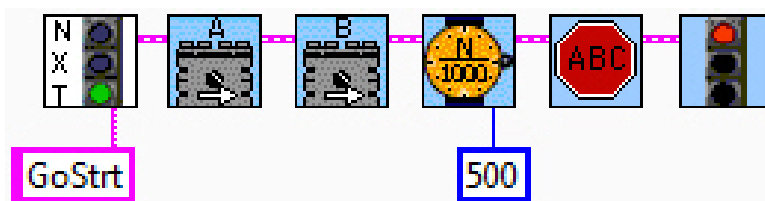


End the program...

The NXT's display will indicate that this icon was reached by changing to a "program end" screen on the NXT brick.



THE WHOLE PROGRAM: Use the wiring tool to connect the pieces!



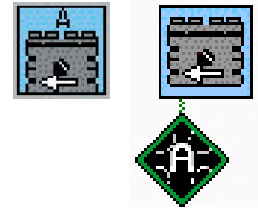
MOVE IN ONE DIRECTION:

THE ROBOT MOVES IN A CURVED PATH (TURNS), MOVE ONE PART.

To run only one motor at a time, we only have to change the motor icons in the RoboLab program to match the following.

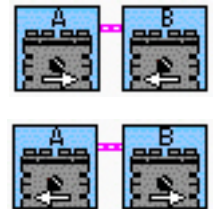
Turn on a motor...

Using one motor icon to start the motor.



To make a sharper turn...

We often want the robot to make the turn without actually changing position. The robot will try to do this when we run both motors in opposite directions.



CONTROL THE STOP:

Stop the motors...

Stop on a dime... slam on the brakes... STOP!

The red stop signs turn off the power to the motor AND apply the brakes.

Using these icons will bring the robot, or the part, to a very rapid stop.



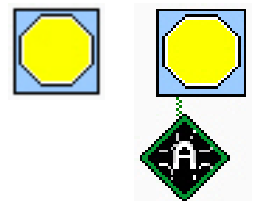
Stop only one of the motors...

Remember that your robot will do exactly what you tell it to do. If both A and B motors are running, and you only stop motor A, then motor B will keep running!



Coast the motors...

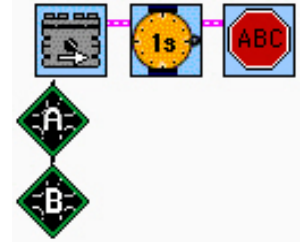
Sometimes we don't want the motion to actually stop, but we do need to turn off the power to a motor. The yellow stop sign is the "Coast" command. Use it with a port modifier to coast only one motor, otherwise they will all glide to a stop.



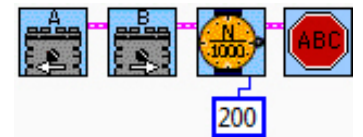
CREATING A COMPLEX PROGRAM: PUTTING ALL THE PIECES TOGETHER...

Once you understand how to write a simple program, all you need to do is put in more “blocks” of code. Each block will take care of ONE basic motion. Assemble the blocks in order and the robot will execute them one at a time until the entire mission is accomplished.

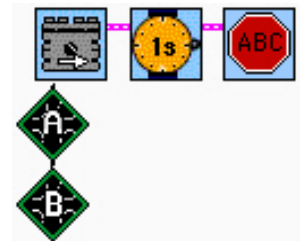
BLOCK: Go Straight



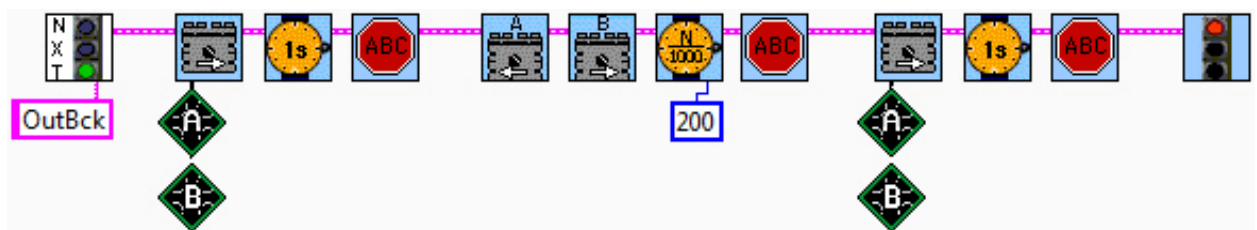
BLOCK: Turn Around



BLOCK: Go Straight (Return to Base?)



THE WHOLE PROGRAM: Wire to connect the pieces!



Click with the typing tool to enter “Comments”... debugging is easier!

