

# Learning to Program

## Using LEGO Mindstorms NXT 2.0



by Matt DePorter  
Program Researcher  
School of Information:  
Technology, Science, and Arts  
University of Arizona



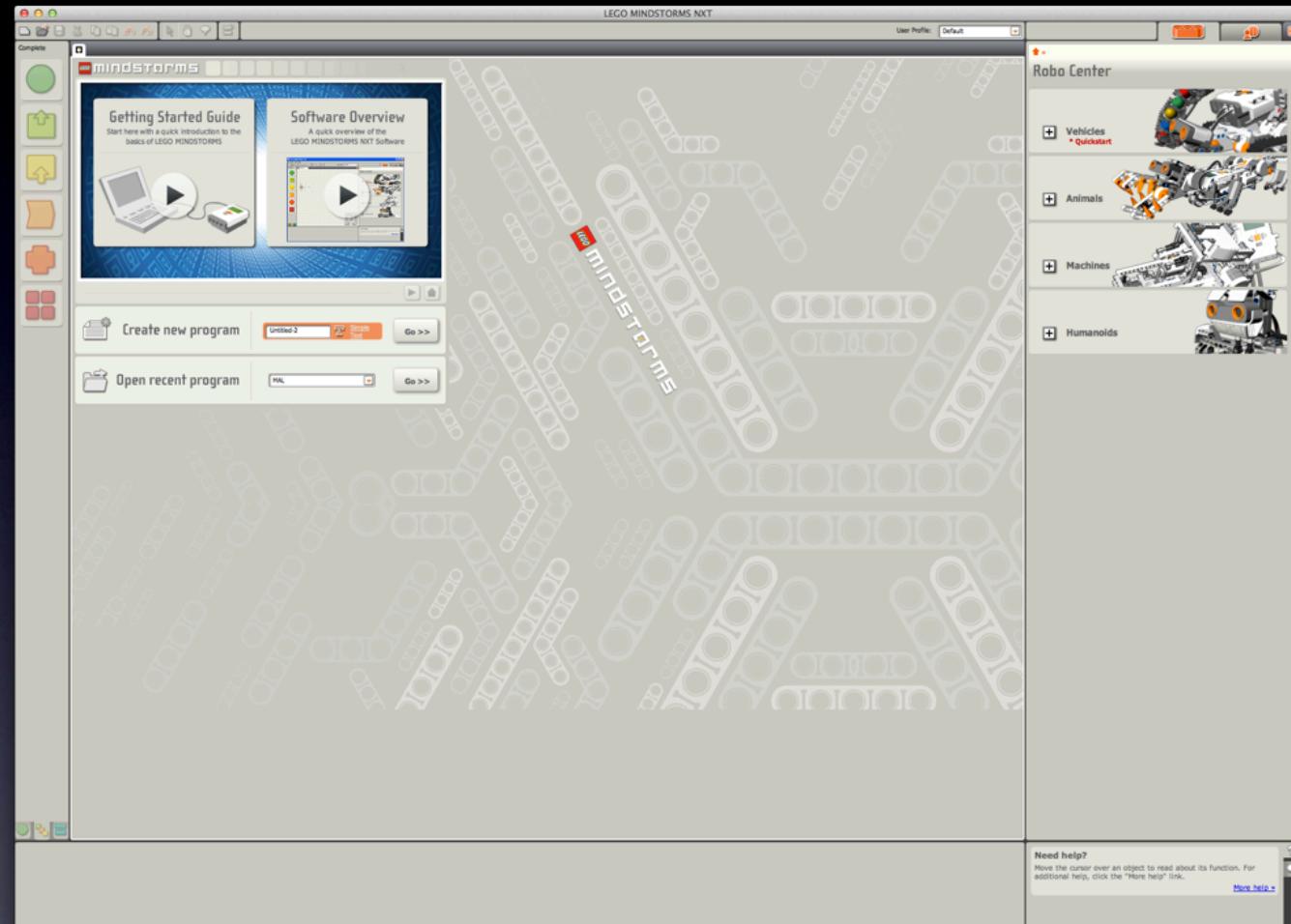
## NXT Mindstorm

- Three Servo Motors
- Ultrasonic Distance Sensor
- Color Sensor/RGB Light
- Two Switch/Bump Sensors
- Speaker
- Bluetooth
- Screen
- Buttons

### Technical Specifications

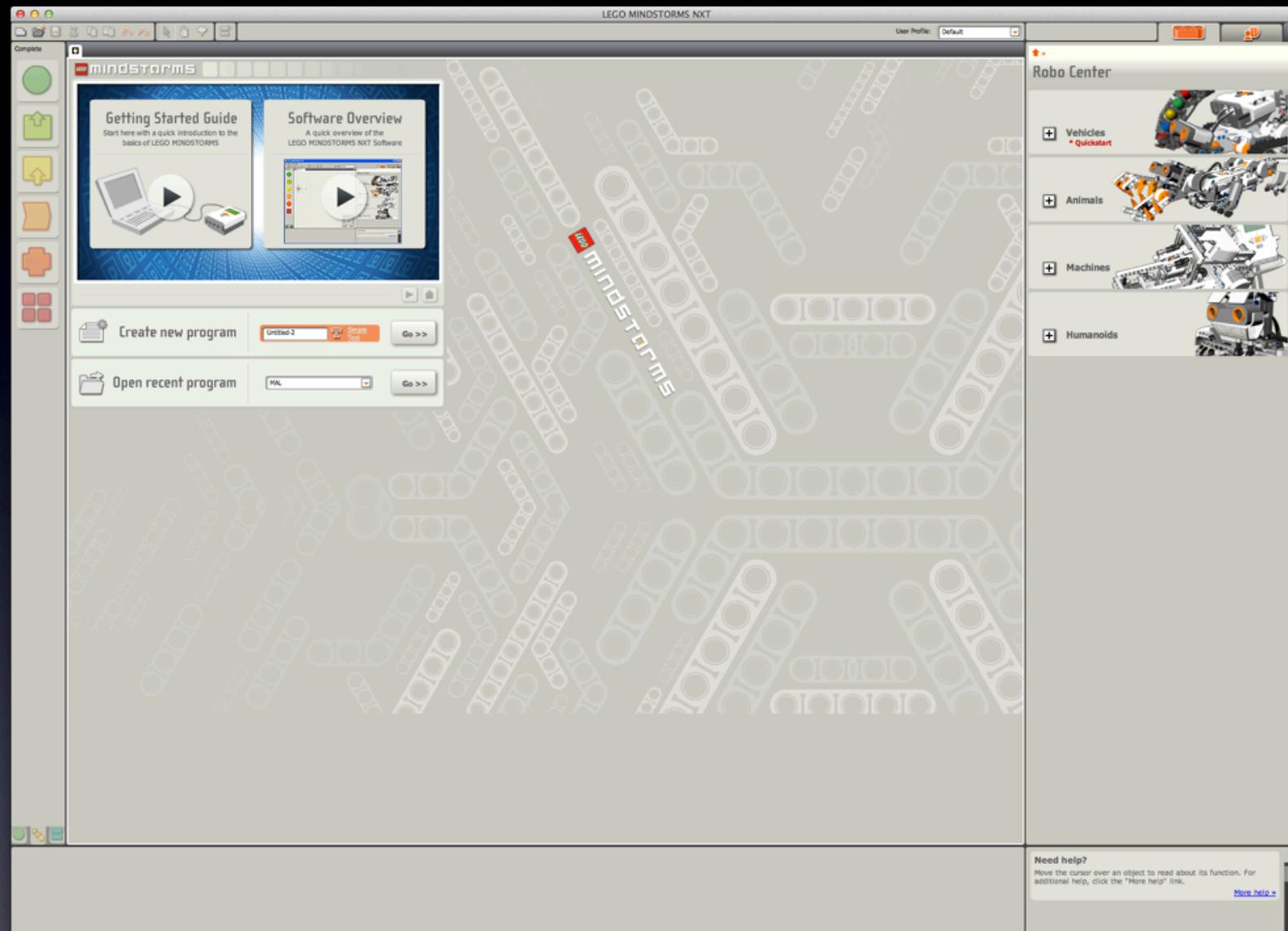
- Microcontroller: 32-Bit ARM7 8-bit AVR
- Flash Memory: 256KB (64KB RAM)
- 100x64 pixel LCD Display
- Bluetooth Class II
- 4 Input Ports
- 3 Output Ports
- Interface: USB

# Mindstorm NXT 2.0 Software



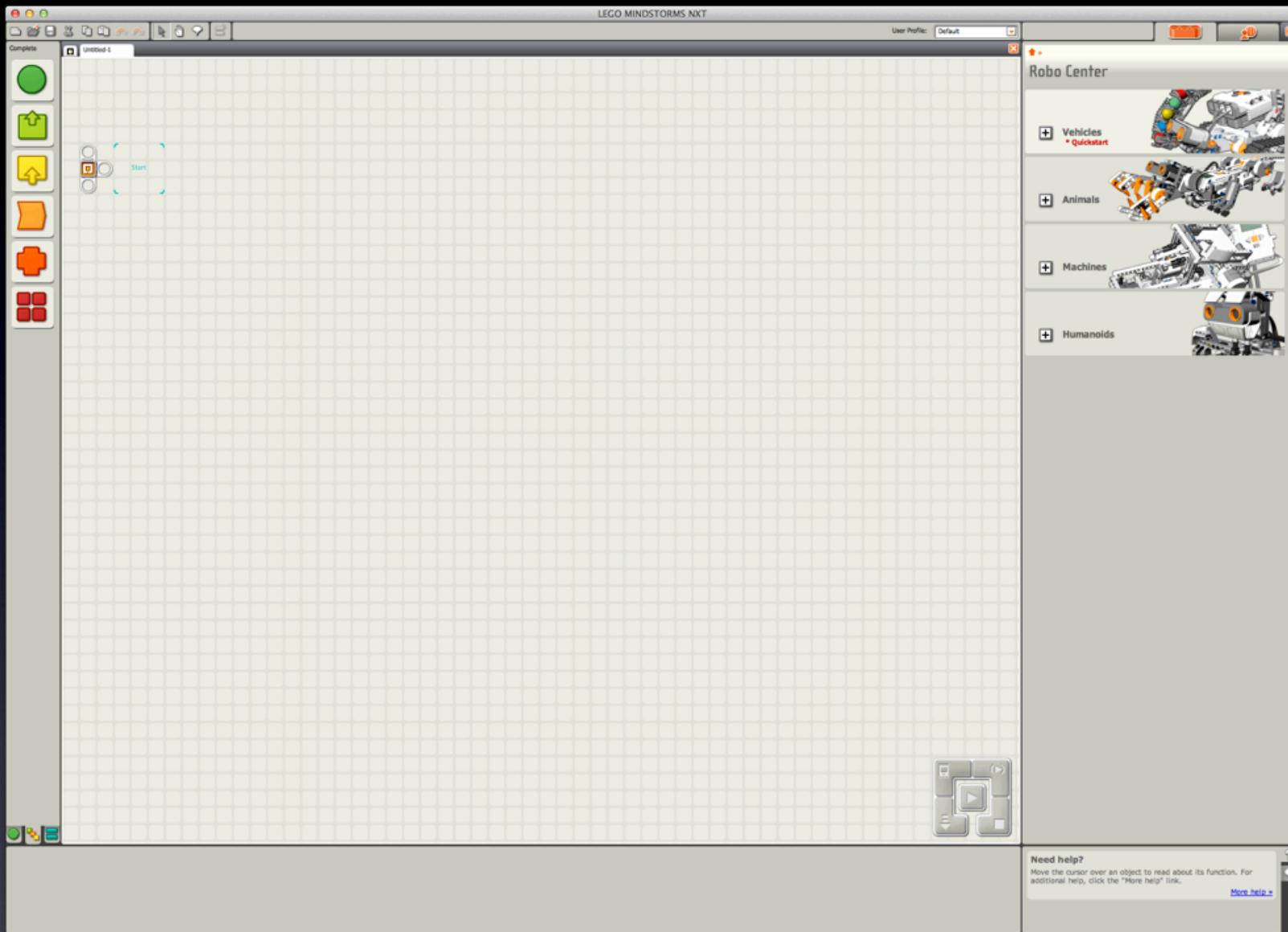
- Great way to learn how to program
- Intuitive Programming Environment
- Powered by National Instruments LabVIEW
- Windows and Mac Compatible

# Getting Started



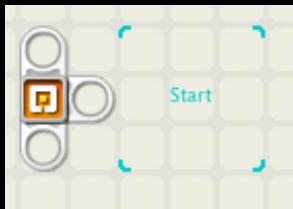
To begin, click Go under the Create new program option

# Getting Started



This is what an empty program looks like, you will add code here.

# Crash course in NXT



This is where your program starts



These are tools to control your robot



These are tools to read sensor data



These are tools to control your controls and sensors.  
This will add functionality to your robot, and give it life.

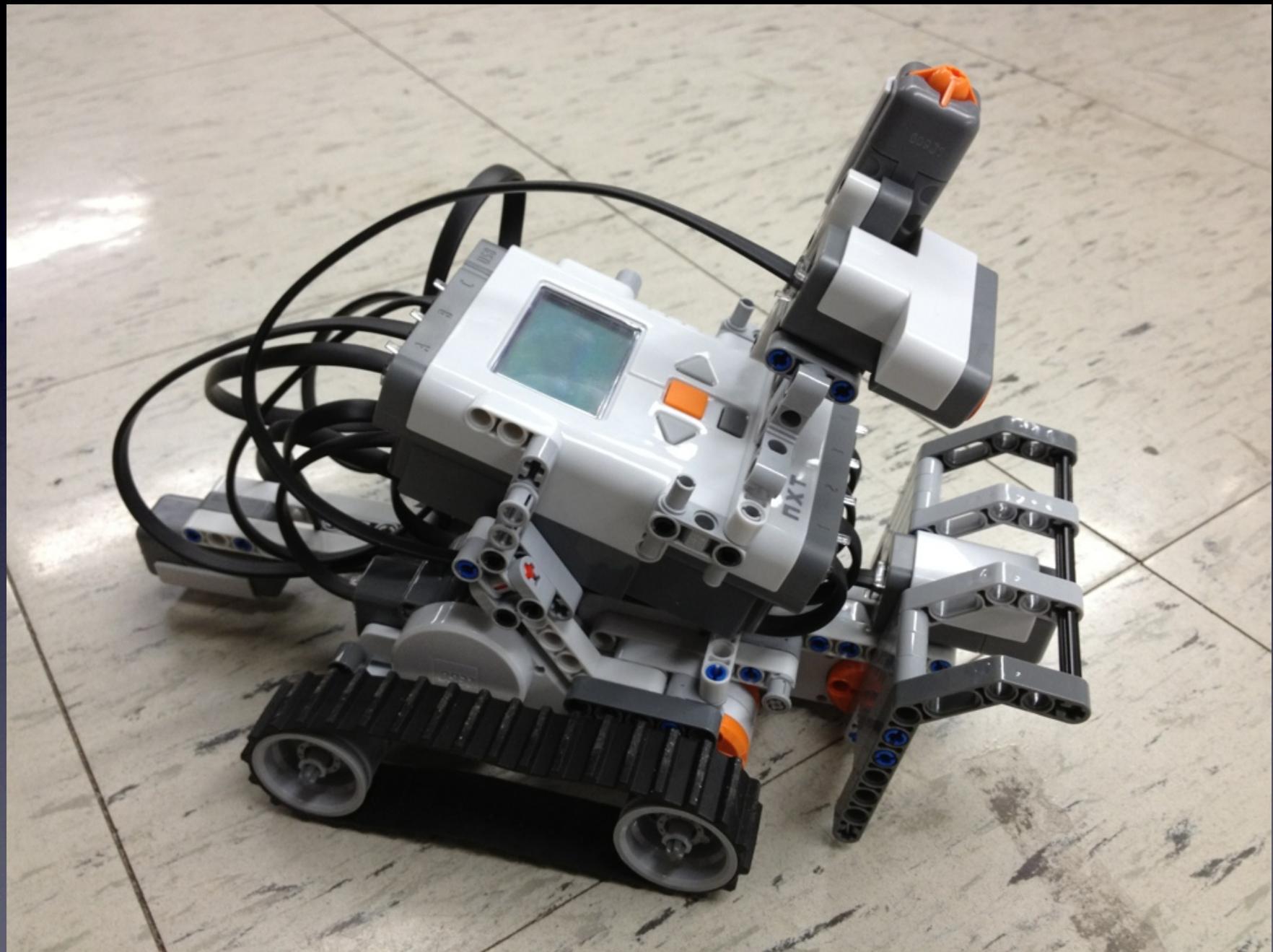


These buttons will compile, upload and run your code on  
the NXT Brick, or the brain of your NXT robot.



The Arrow is used to move, edit and change tools.  
The Hand is used to move the workspace around.  
The Speech bubble is for commenting on your code.

# What you will be programming!



# What you have at your fingertips:

These are the sensors on your robot, and which port they are connected to:

- 1 - Ultrasonic Sensor
- 2 - Rear Bump Sensor
- 3 - Top Emergency Stop Button
- 4 - Color Sensor
- A - Claw Motor
- B - Right Drive Motor
- C - Left Drive Motor



# General rules for programming

- Think about what you want to do before you start programming
- Write down your tasks for your robot before programming
- Put those tasks in order based on priority
- Take a look around the software, and see what you have at your disposal
- Find tools that will do what you are proposing in the NXT development environment

# General rules for programming your NXT

- Always make sure that your tool is connected to the main rail. If there it is not connected, it will not be used by your robot
- When moving widgets onto workspace, give the computer time to make space around that object, otherwise bad things can happen
- When using Flow tools, always have the widget you want to run inside the scope of that “circle”
- Always double check which port your widget is configured to use. If you choose the wrong port, you will not be able to communicate with the device
- If you are using control wires, to remove them click the right terminal on the next widget to remove that wire
- When using motor, alway use Brake for "Next Action"
- ALWAYS save your work. Please do it frequently too!

# Tools of Life!

## Switch

A switch is a tool that lets you choose how the tool will be used. This is a control statement. If your condition is true, it will run the blocks in the top loop, otherwise it will run the blocks in the lower portion.  
Note: If you select Flat View (Right), it will just stack the conditions on top of one another. You will either have to choose the Check or the X to see the blocks inside.



# Tools of Life!

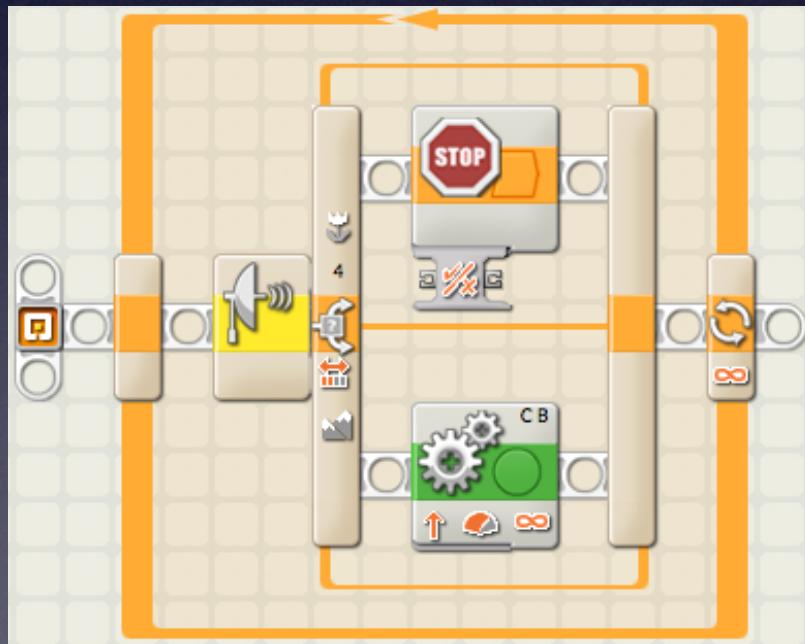


## Stop

This block will halt the execution of all code.

For example, on our robot, when the top button is pressed, it will kill all commands issued to the robot. Choose carefully where you place this block, since you cannot run any code after it.

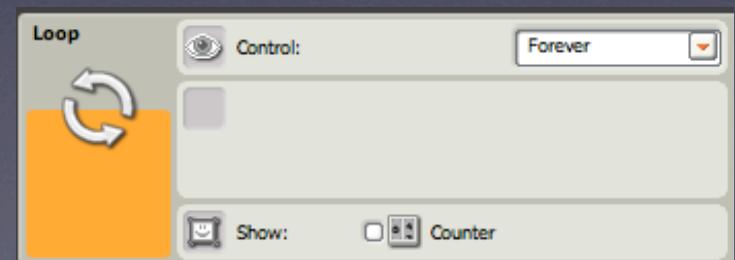
## Example code



## Loop

This block will loop over everything that appears inside its brackets. It will expand to fit any block you put inside of it.

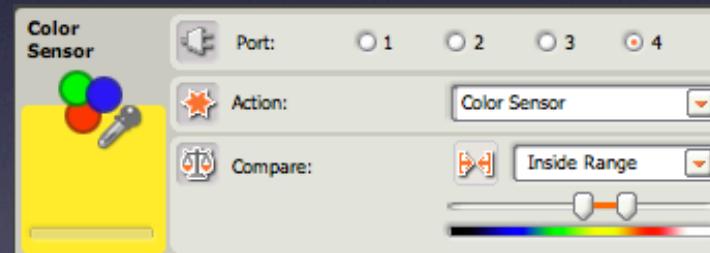
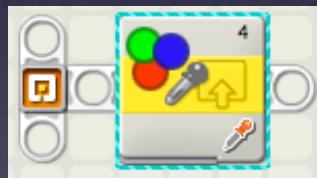
1. Select how your loop will operate. Select from: Forever, Sensor, Time, Count, or Logic. These will determine your termination condition for your loop. You can use other blocks to change the way your loop operates. To connect those blocks to this loop you use what LabVIEW or NXT calls wires.
2. Then choose whether you want the loop to show you how many times it has run.



# Tool Overview

## Color Sensor

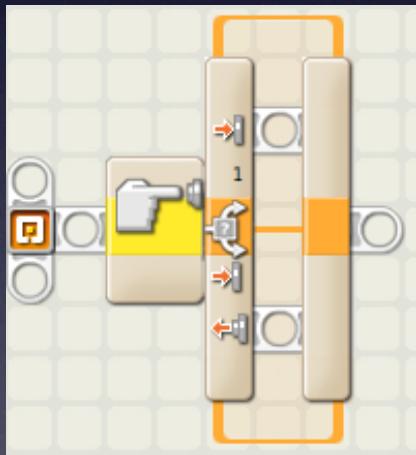
1. Specify a port,
2. Choose a color range for the sensor to look for.



# Tool Overview

## Touch Sensor

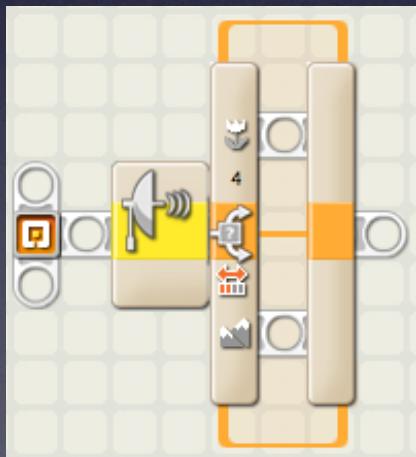
1. Specify a port.
2. Select if you want to read a press, a release or just a bump



# Tool Overview

## Ultrasonic Range Finder

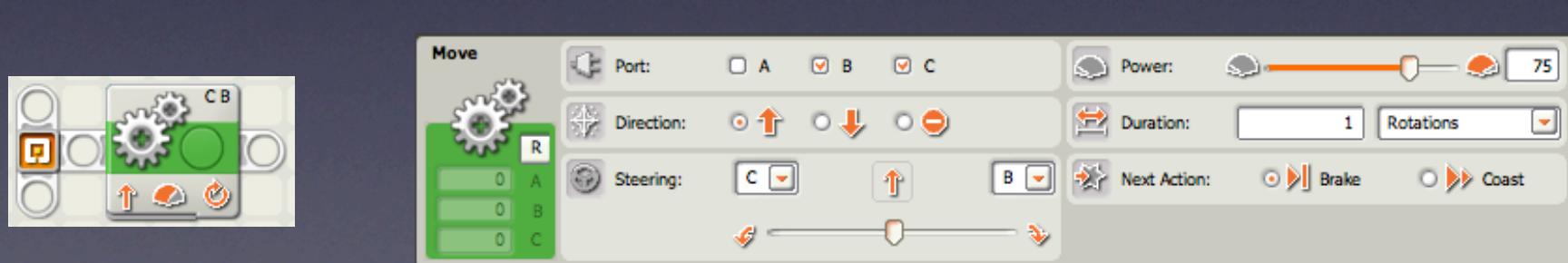
1. Specify a port.
2. Choose a distance threshold.
3. Select whether you want the distance to be less than or greater than your specified distance.
4. Choose a unit of measure for sensor to measure with.



# Tool Overview

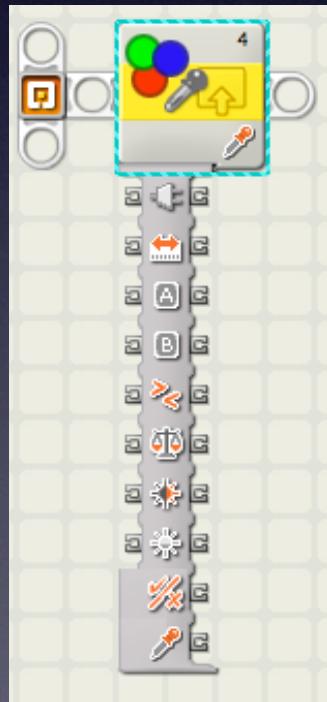
## Servo Motor

1. Choose motor(s) to use by selecting ports.  
(Ports B and C are selected in this case)
2. Specify a direction for the motor(s) to go, either forward, backwards, or just to stop an already moving motor.
3. If you are using two motors, you can choose to make them drive straight, turn left or right.
4. Select a power level. This option is to adjust the power the servo will have.
5. Select a duration. You can pick from: Unlimited, Degrees, Rotation, Seconds. Depending on what you choose, will determine what you enter for a duration. If you chose degrees, you would specify how far that servo will rotate. If you chose unlimited, the option will be greyed out, and the motor will run until the NXT software issues a new command to the motor.
6. Next Action. This lets you choose if the motor will stop after this block, or coast.



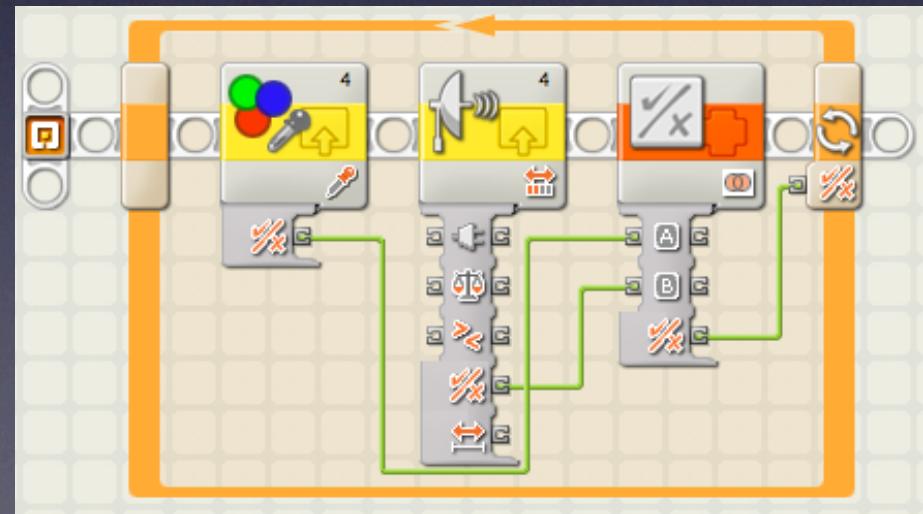
# Block Expansion Options

Most blocks have the ability to display all input and output options for that block. A regular block will appear like the block on the left.



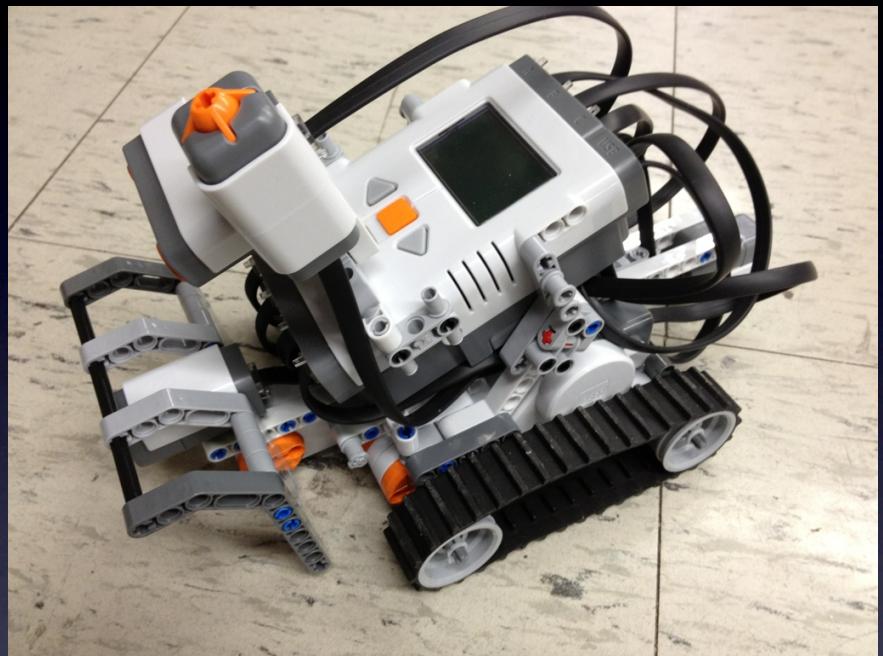
If you hover your mouse over the seam between the block and the option, a different cursor will appear. If you click it, it will expand like the block in the middle. This allows you to provide input to that block, or get the output of that block. An example of this appears on the right.

This is an example of some blocks connected with wires. We are taking the output of the color sensor, which gives either True or False depending on if the color is in your specified range. Same for the Ultrasonic sensor. If it is less than or greater than your specified range. We take those outputs and pipe them into a Logic Block. This will either AND, OR, XOR, or NOT your logic coming from other sensor/ blocks.  
The output will stop the execution of the loop if either sensor reads its True case.

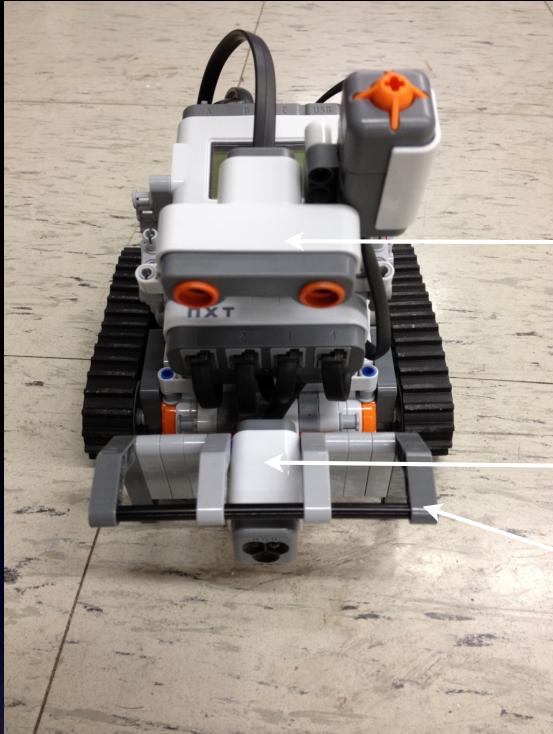


# Now, what you get to do!

- If top Emergency Stop button is hit
  - Stop all programming code
- Otherwise
  - If Color Sensor is detecting Blue
    - Stop Drive Motors
    - Drive forward for a quarter second
    - Rotate Claw Motor Forward 72 degrees
    - Drive backwards for one rotation
    - Rotate Claw Motor Backwards 72 degrees
    - Drive Backwards while turning left for one rotation
    - Stop Drive Motors
  - Otherwise
    - If Rear Bump Sensor is hit
      - Drive forward one rotation
      - Drive Forward while turning right
    - Otherwise
      - Drive forward with Drive Motors (Unlimited)
      - If Ultrasonic Sensor detects something less than 5 inches away from the sensor
        - Stop Drive Motors
        - Drive Forwards while turning right for two rotations
      - Otherwise
        - Drive forward with Drive Motors (Unlimited)



Pseudocode!



Ultrasonic Sensor

Color Sensor

Claw



Bump Sensor

NXT Brick

Emergency Stop

