



FLL Programming 101

EV3 on NXT



September 2013
Version 1.2a



Legal Stuff

© 2006 INSciTE in agreement with, and permission from FIRST and the LEGO Group. This document is developed by INSciTE and is not an official FLL document from FIRST and the LEGO Group. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc-sa/2.0/>

or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

LEGO®, ROBOLAB, and MINDSTORMS™ are trademarks of the LEGO Group used here with special permission. FIRST™ LEGO® League is a trademark owned by FIRST (For Inspiration and Recognition of Science and Technology) and the LEGO Group used here with special permission. INSciTE™ is a trademark of Innovations in Science and Technology Education.

INSciTE

PO Box 41221

Plymouth, MN 55441

www.hightechkids.org



Creative Commons License

High Tech Kids is committed to making the best possible training material. Since HTK has such a dynamic and talented global community, the best training material and processes, will naturally come from a team effort.

Professionally, the open source software movement has shown that far flung software developers can cooperate to create robust and widely used software. The open source process is a model High Tech Kids wants to emulate for much of the material we develop. The open source software license is a key enabler in this process. That is why we have chosen to make this work available via a Creative Commons license. Your usage rights are summarized below, but please check the complete license at: <http://creativecommons.org/licenses/by-nc-sa/2.0/>.



Credits

This presentation was developed by Doug Frevert. It is based on the work of Fred Rose. The accompanying labs were originally done in RCX Code by Joel Stone and converted to ROBOLAB by Doug Frevert. A portion of the material is taken from “*Building LEGO Robots for FIRST LEGO League*” by Dean Hystad. Amy Harris defined the 10 programming steps. Eric Engstrom, Jen Reichow, and Ted Cochran reviewed ongoing drafts. Eric taught the first class and helped modify the content accordingly.



Computer Programming 101

Objective

Learn to program robots.

FLL, NXT, EV3

Structure

Theory

Examples specific to EV3

Hands-on

This class **is**

An approach to programming

This class **is not**

Exhaustive reference on EV3



Class Agenda

Computer Basics

The Programming
Environment

Moving and Turning
Lab #1

MyBlocks and Loops
Lab #2

Sensors

Lab #3

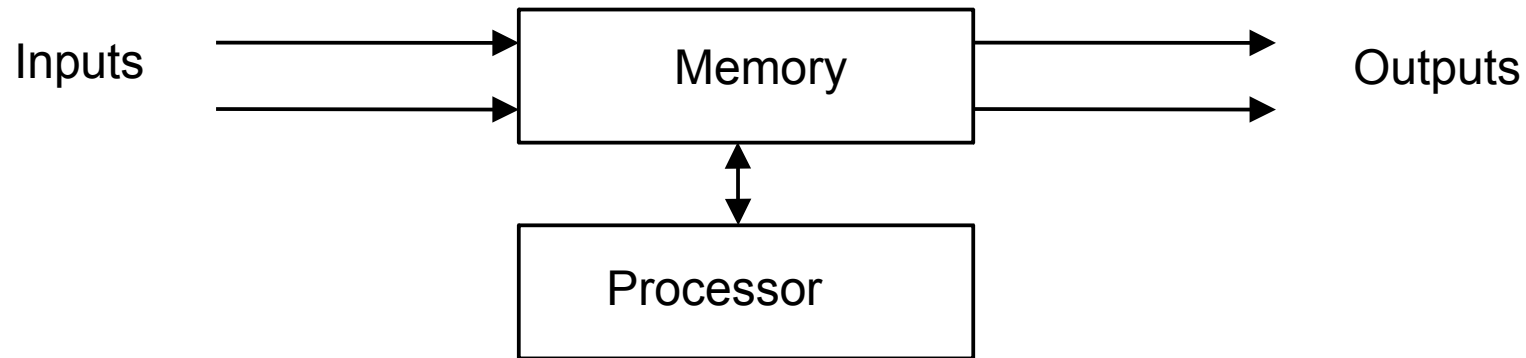
Advanced Topics
Problem Solving



Computer Basics



The Computer (Generic)



The processor executes commands.

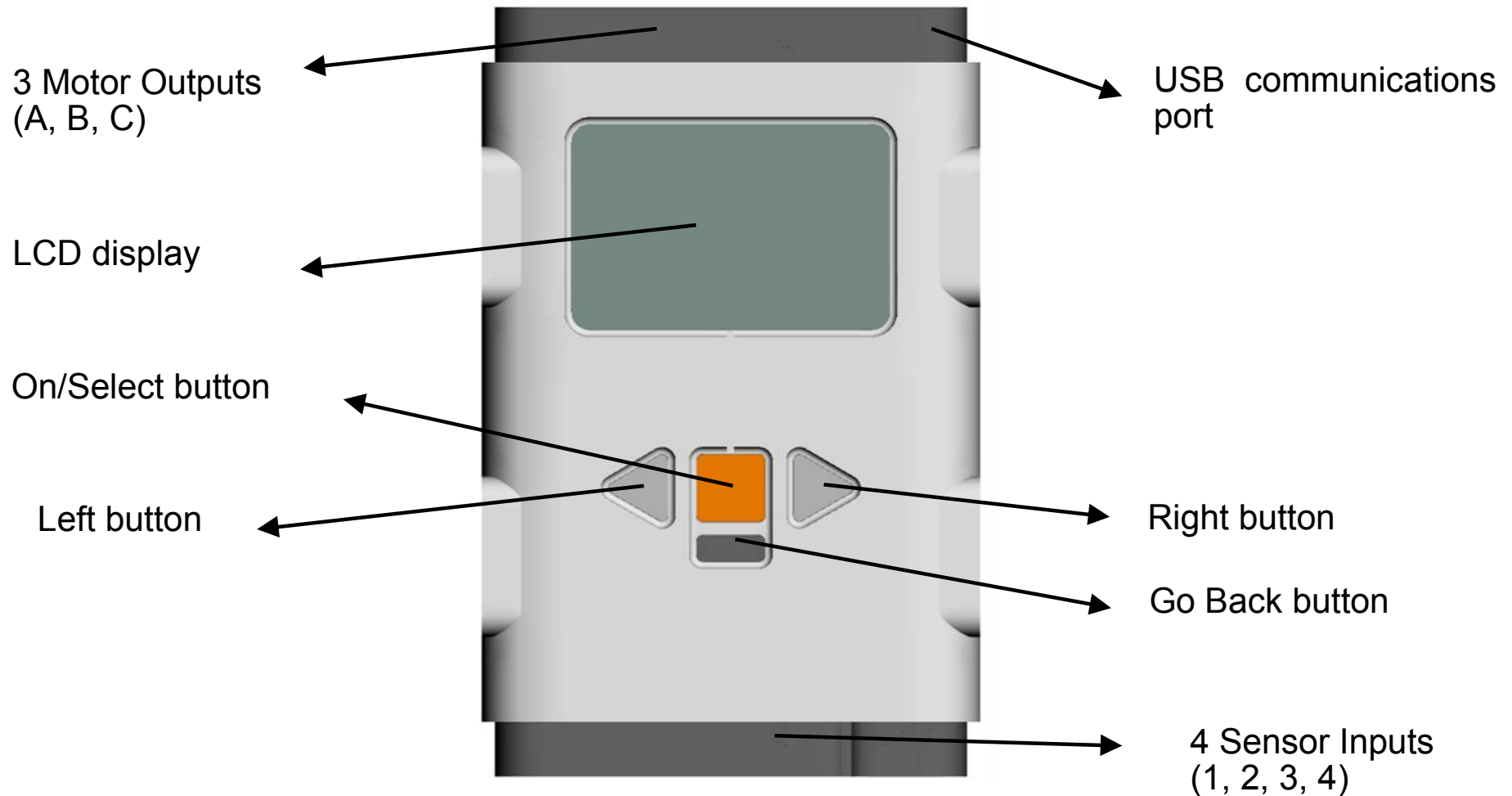
Memory stores program and data.

Inputs tell the computer about the world.

Outputs tell the world about the computer.



NXT



Processor: 32 bit ARM Atmel AT91SAM256 running at 50 Mhz

Memory: 64K Static RAM, 256K Flash



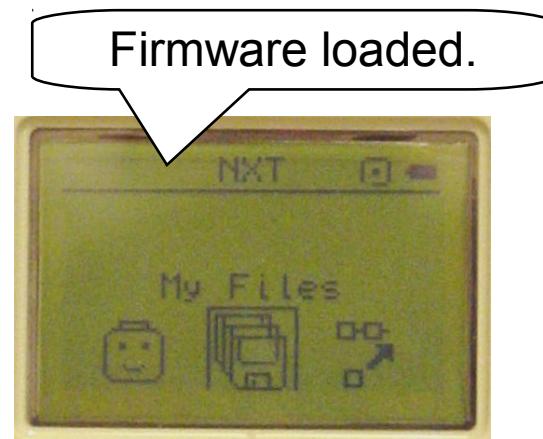
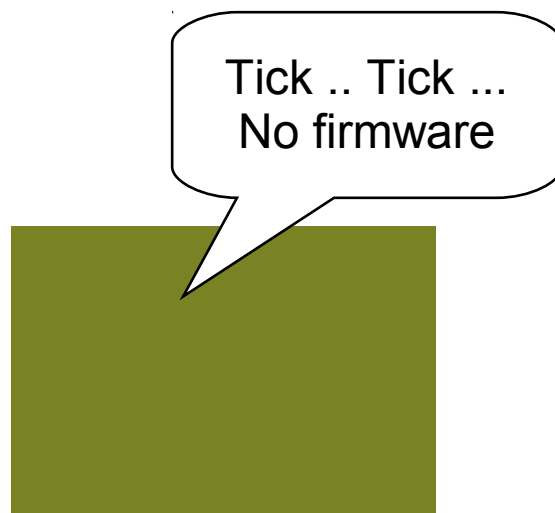
Firmware Loaded?

Firmware must be loaded onto an NXT so that the NXT can understand your programs.

Only required to be loaded

- To install a new firmware release

- To restore lost firmware (was a problem for RCX).





Writing a Computer Program

Specify the task

Inputs to be supplied

Outputs to be produced

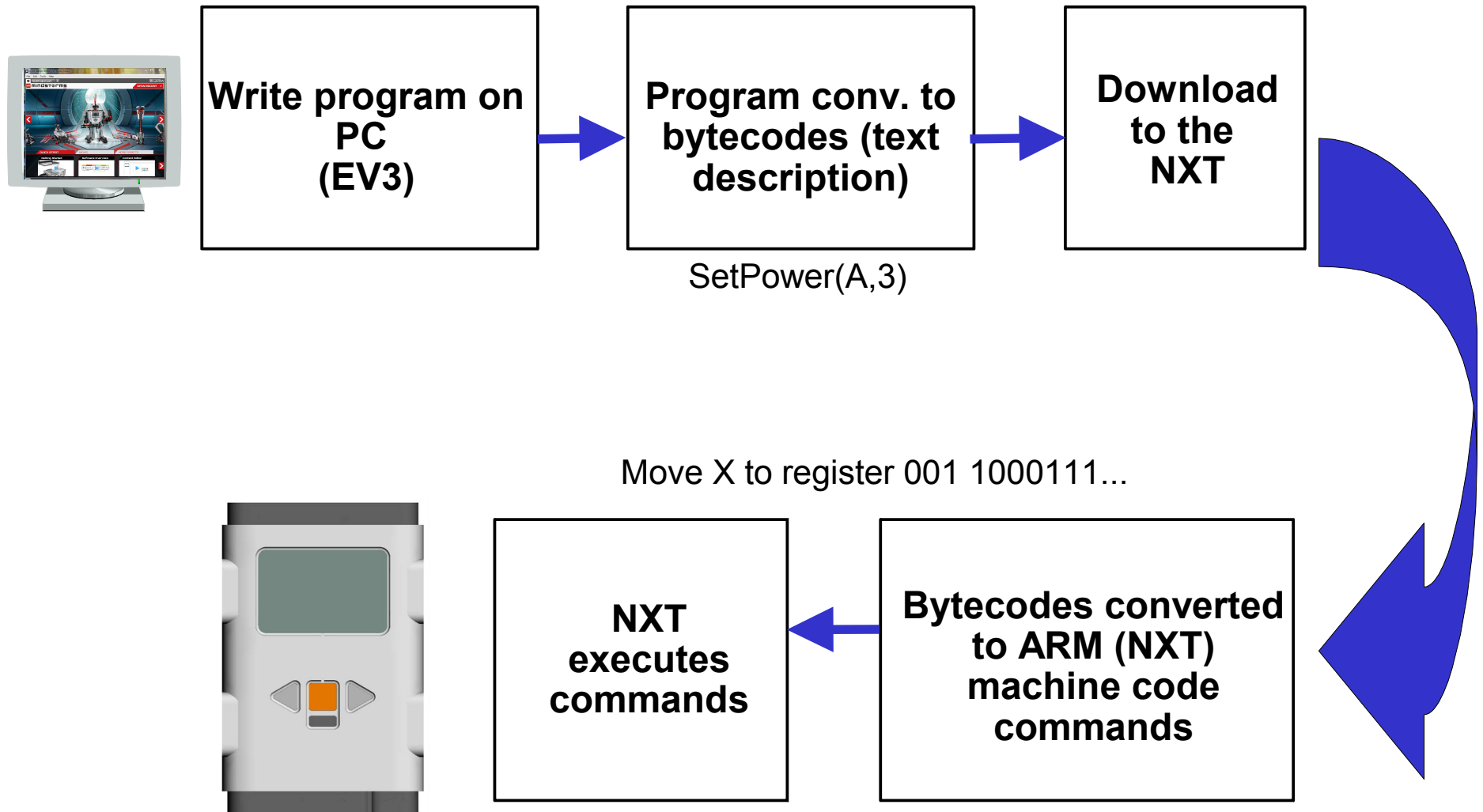
Devise an algorithm

Express that algorithm in a computer language

From: Introduction to Pascal, Welsh and Elder



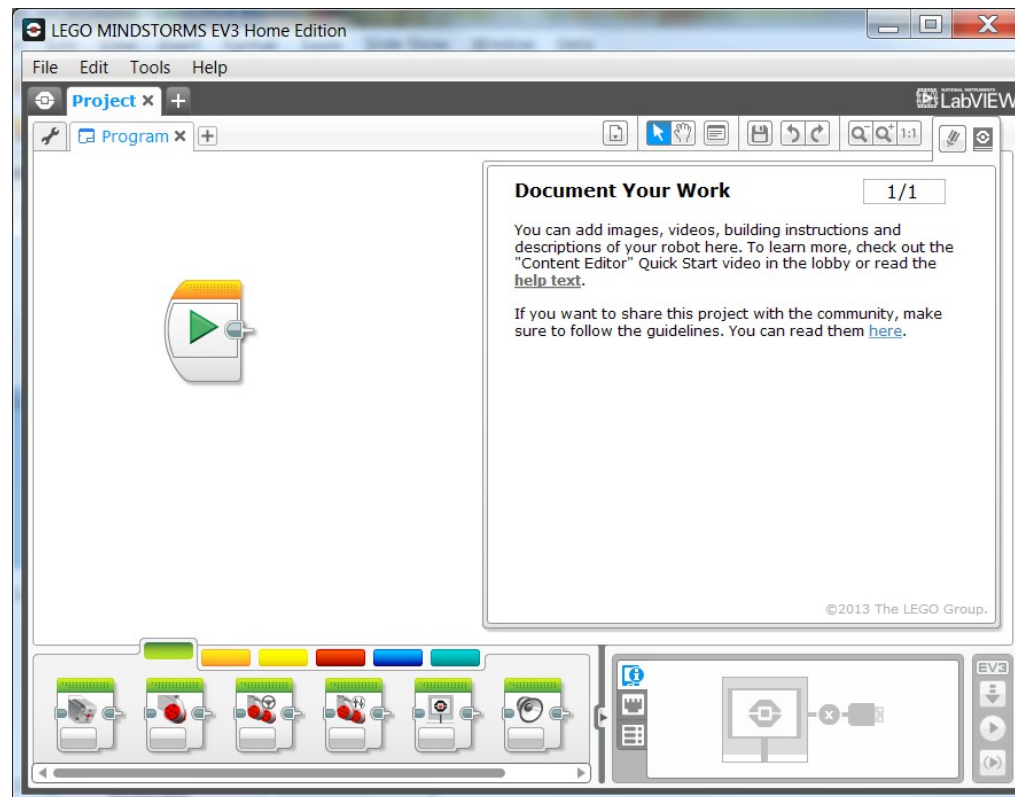
Running a computer program (NXT)





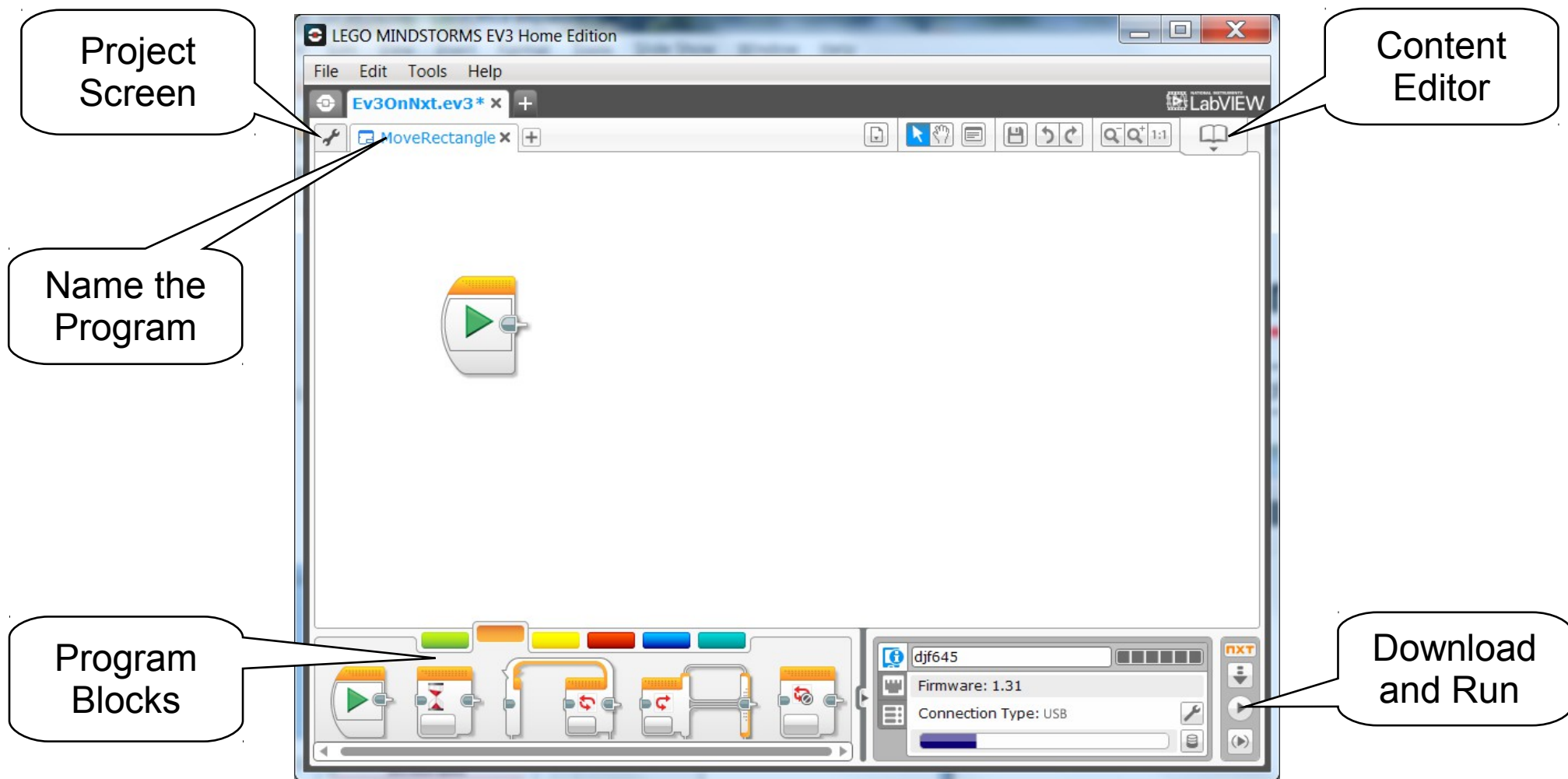
EV3

The Programming Environment



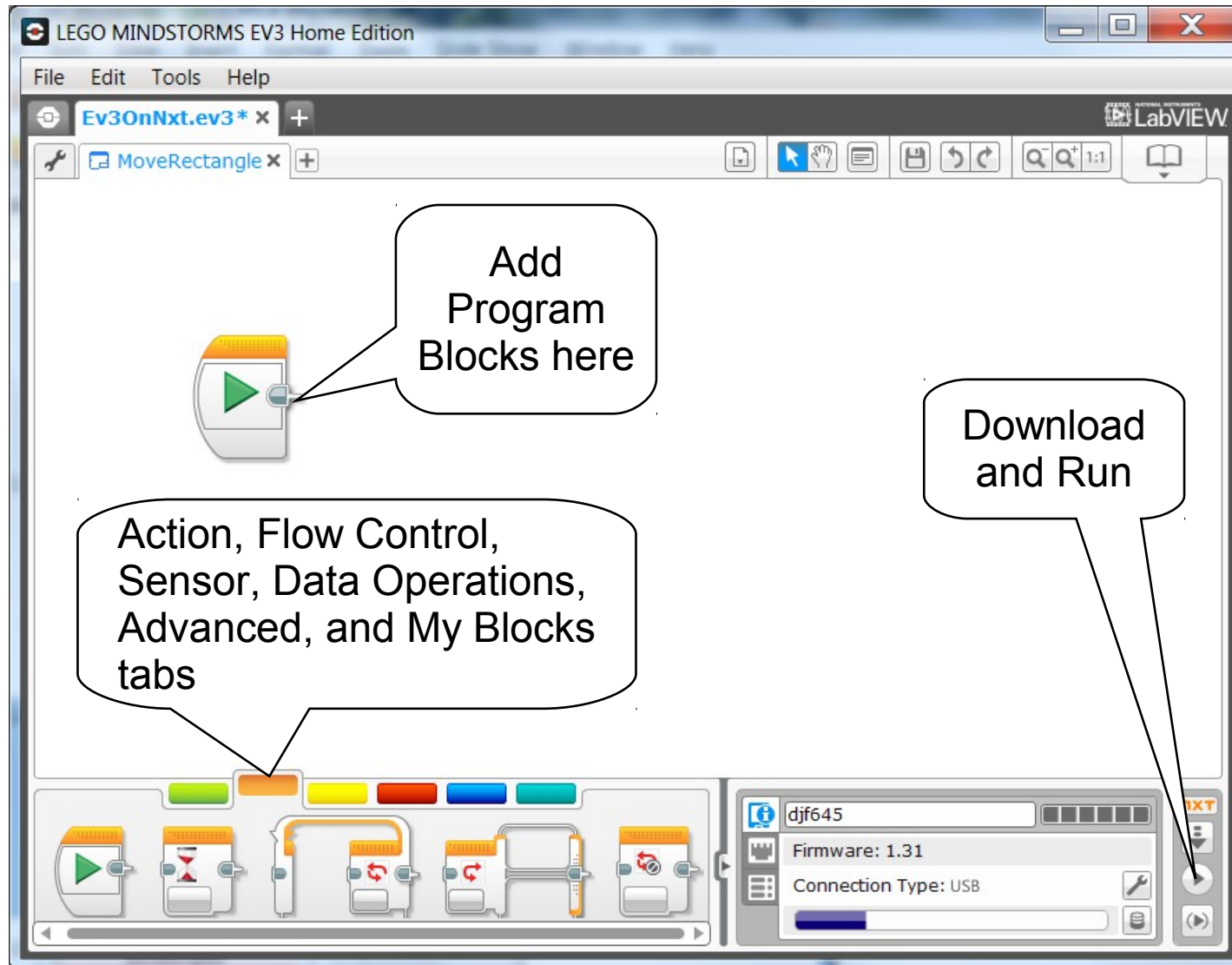


EV3 Opening Workspace





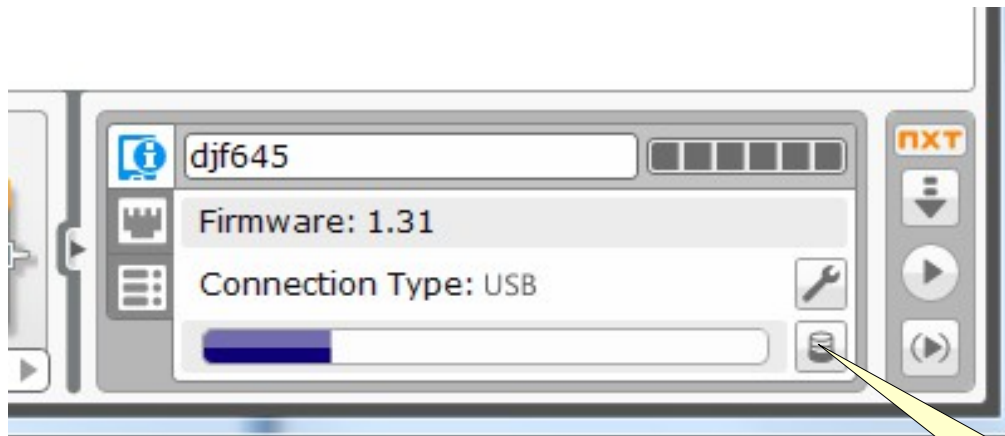
EV3 Work Space





EV3 to NXT

Communications



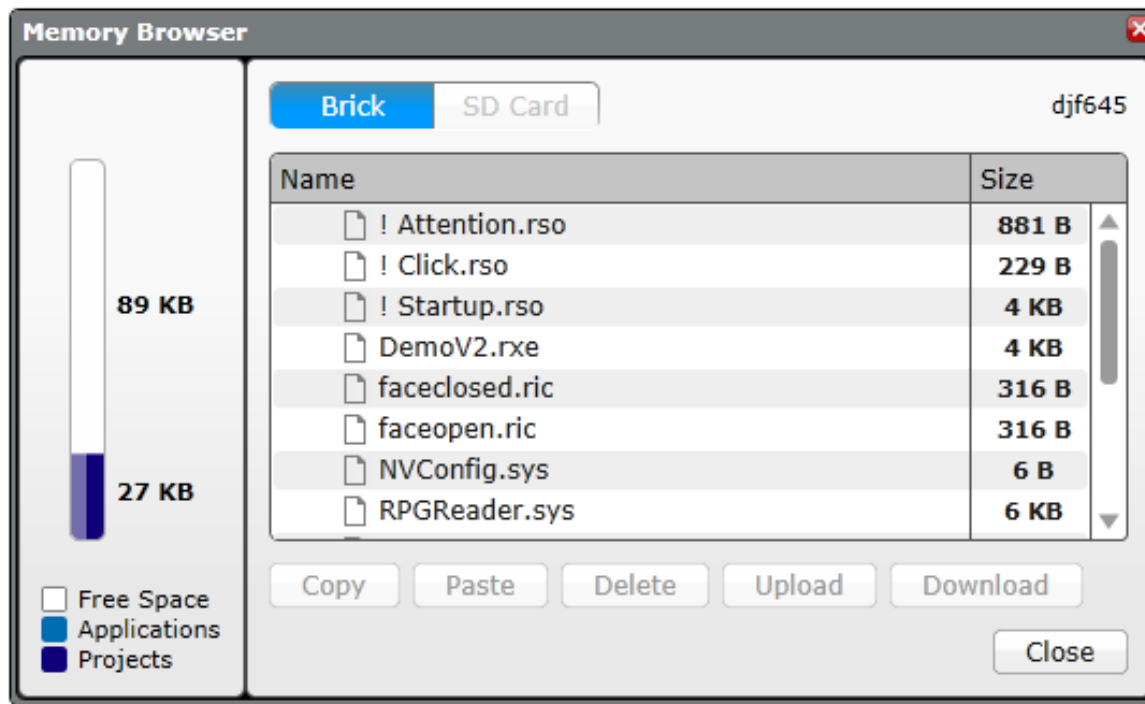
When connected by USB cable or Bluetooth
Give your NXT a new name
Check Battery voltage
View available memory (in KiloBytes)
Firmware version

View the NXT
memory
browser.
See the next
slide.



EV3 to NXT

Memory

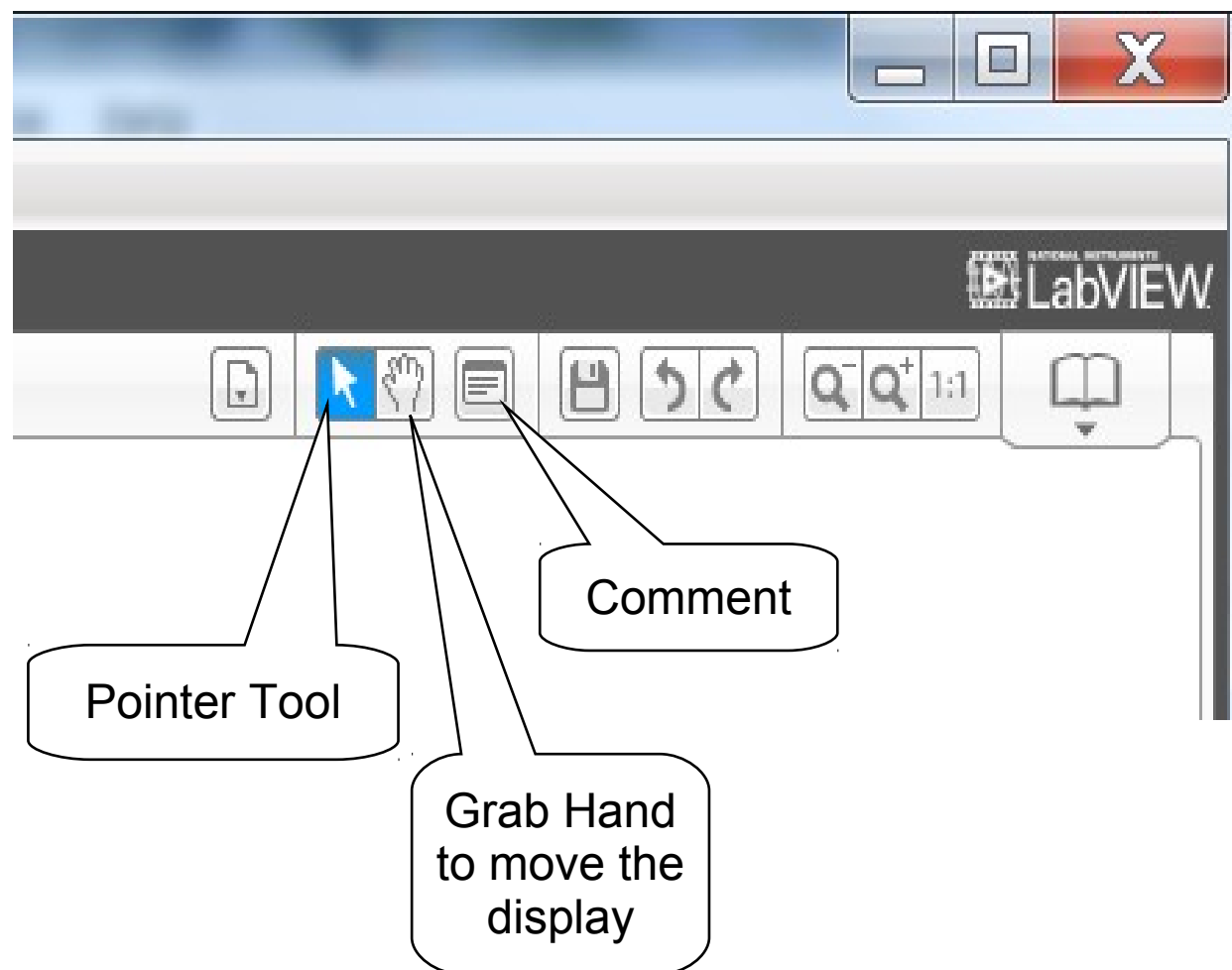


Select, then delete Programs, Sounds, Graphics, and Unused files.

Can free up to 118Kb of Free Storage on the NXT.



EV3 Work Space





Tips and Tricks (1)

The NXT has memory to store many programs
NXT automatically powers down.



Bluetooth Communications

NXT brick has bluetooth.

EV3 software to NXT via bluetooth is **not**
supported.

Disable bluetooth during competition.



Tips and Tricks (2)

Direction of connecting wires

NXT wire connectors only fit one way. Can not be rotated.

Batteries

AA

No worry about losing firmware.

NiMH rechargeable batteries work.

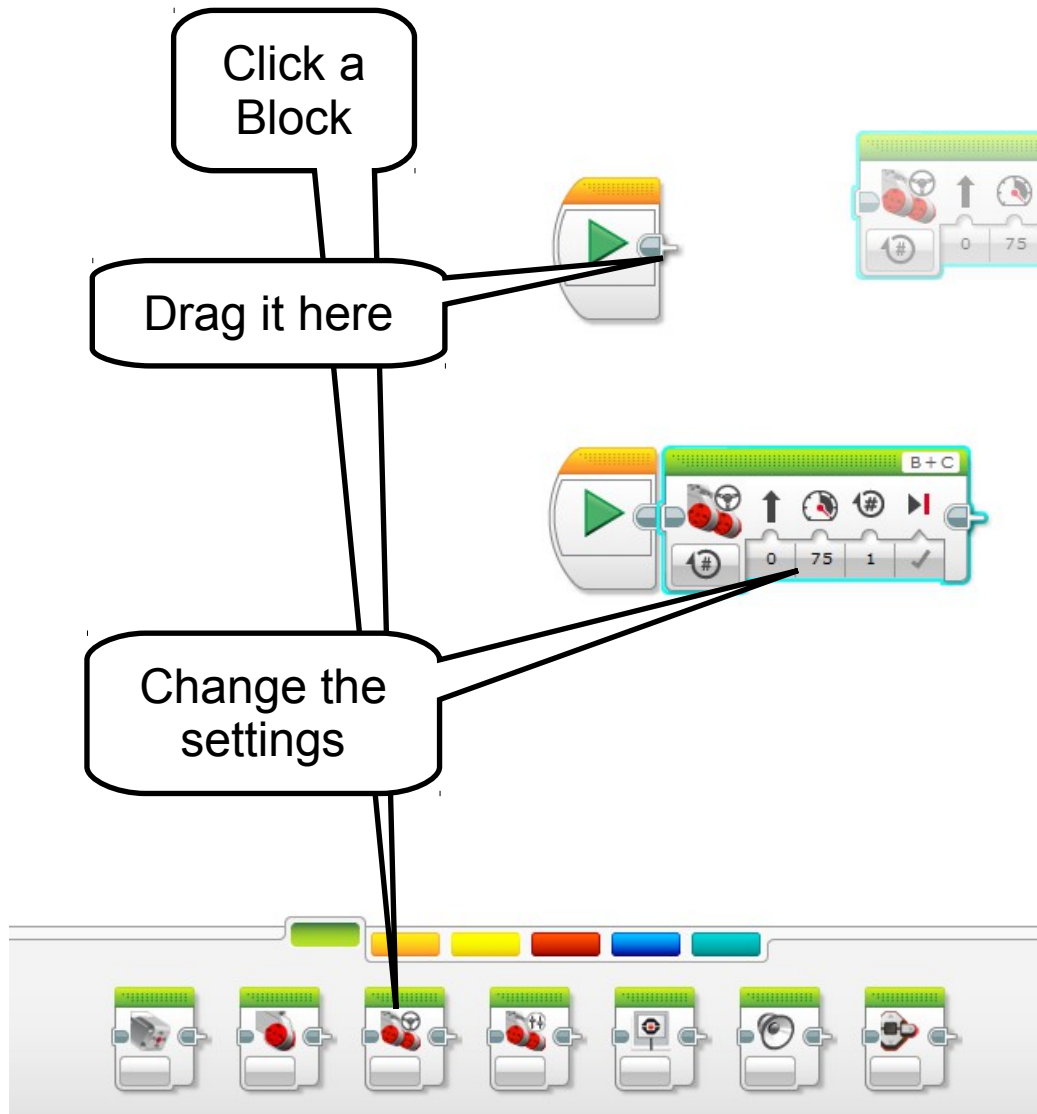
NiCads don't.

Lithium rechargeables come with the FLL Mindstorm kits.

Avoid stalling the motors, it drains batteries.



Adding a Block to a Program



Click on a Block

Move cursor onto Program and drop it into place. EV3 will make room.

Change settings



Action Blocks

Action Blocks

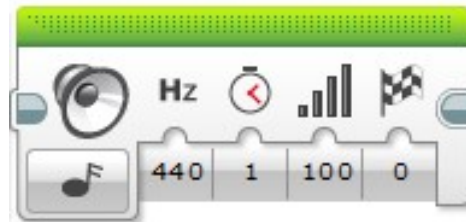
Move

Sound

Display



Move Steering



Sound



Display

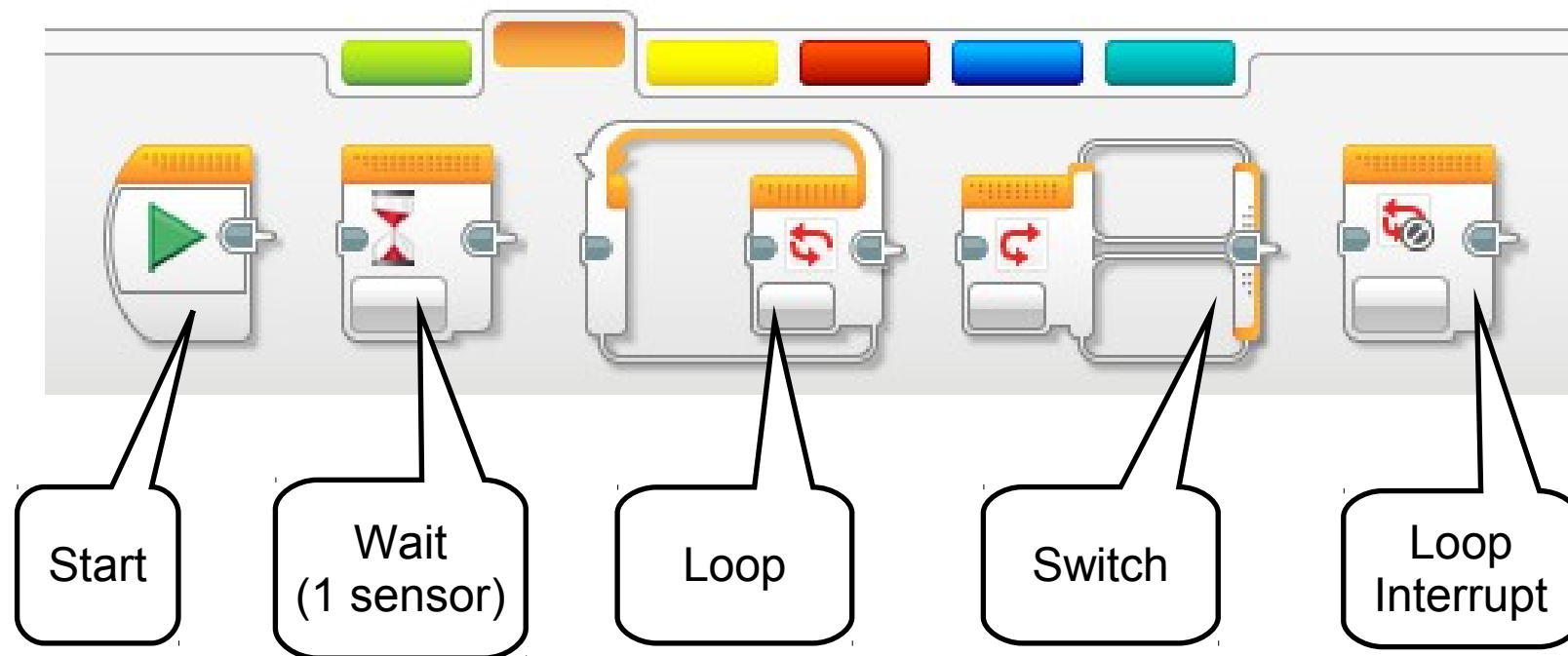
With many modifiers

Move direction, steering, distance, motors used . . .

Settings are visible without digging deeper.



Flow Controls

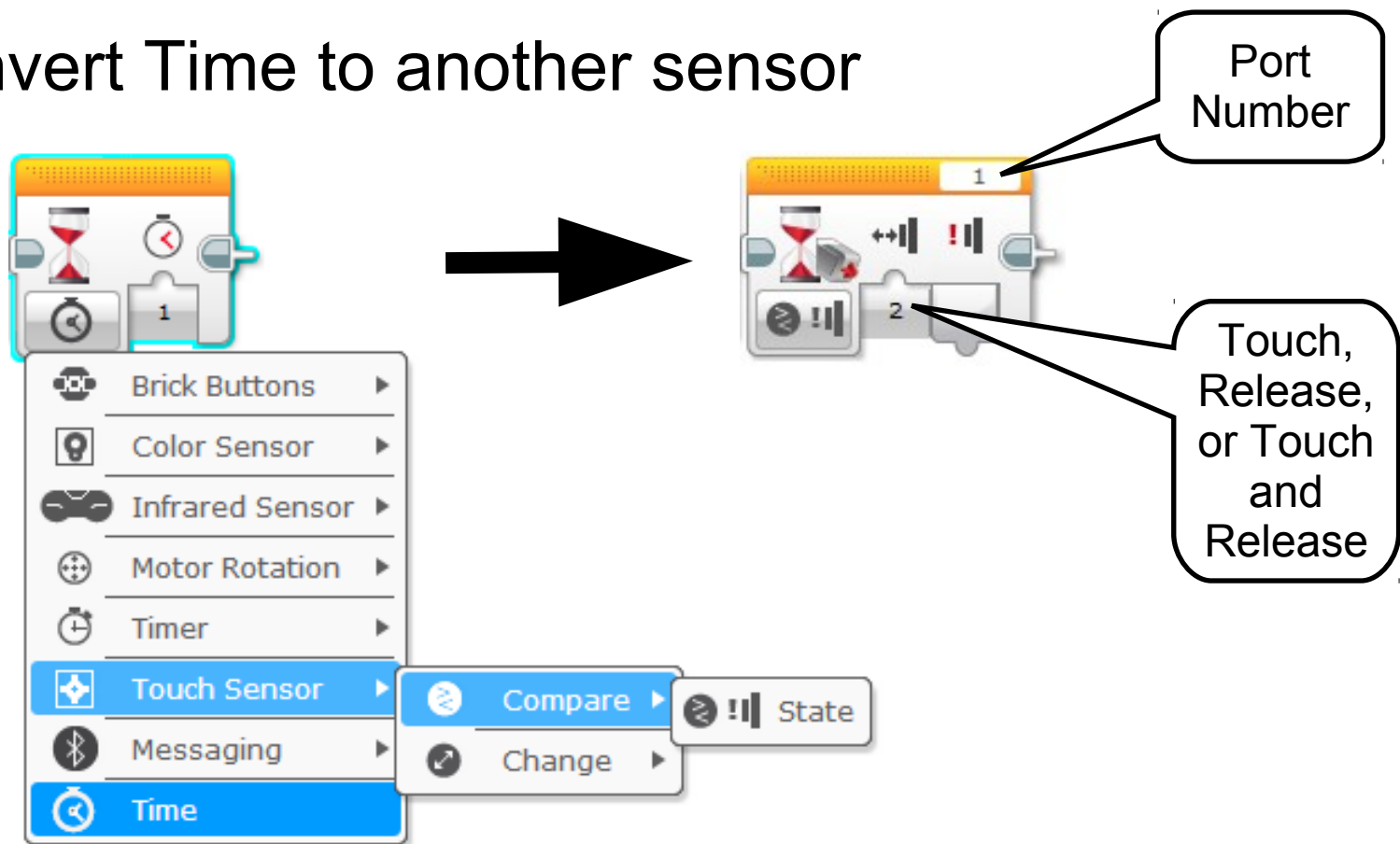




Wait Block

Wait block

Convert Time to another sensor





Motors

9 volt geared motor

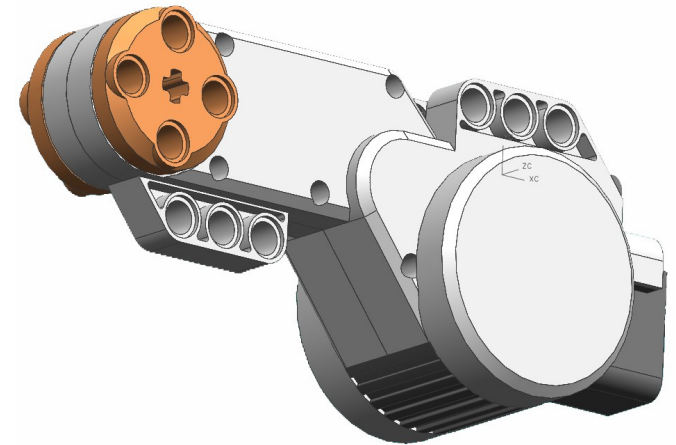
Making the motors turn is the **output** of your program.
It makes your creation a robot!

Without load, motor shaft turns at about 150 rpm.

Servo sensitive to 1 degree.

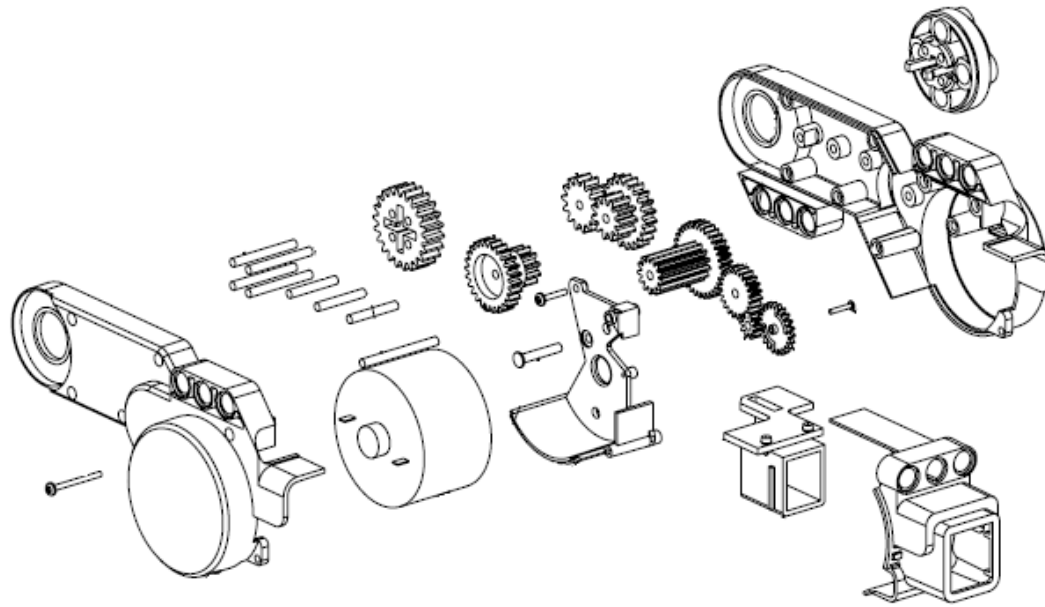
With a typical robot, 3-4 hours on a set of batteries.

FLL allows up to 4 motors.





Motor Details



Motor can be set to different power settings

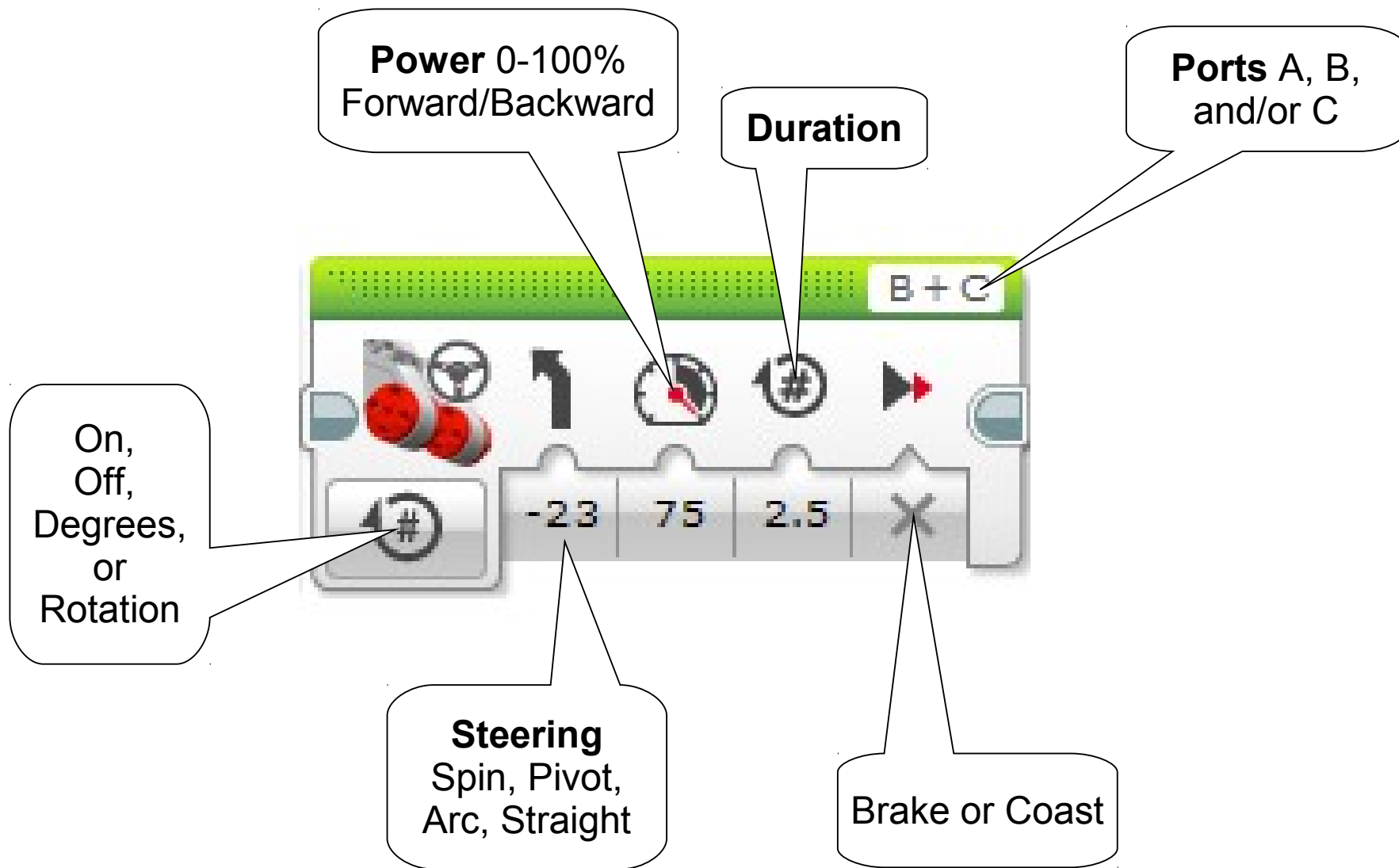
Power levels 0-100

Power is adjusted by Pulse Width Modulation

Turning the power setting up higher
essentially makes the shaft turn faster.



Using the Steering Block





Lab One



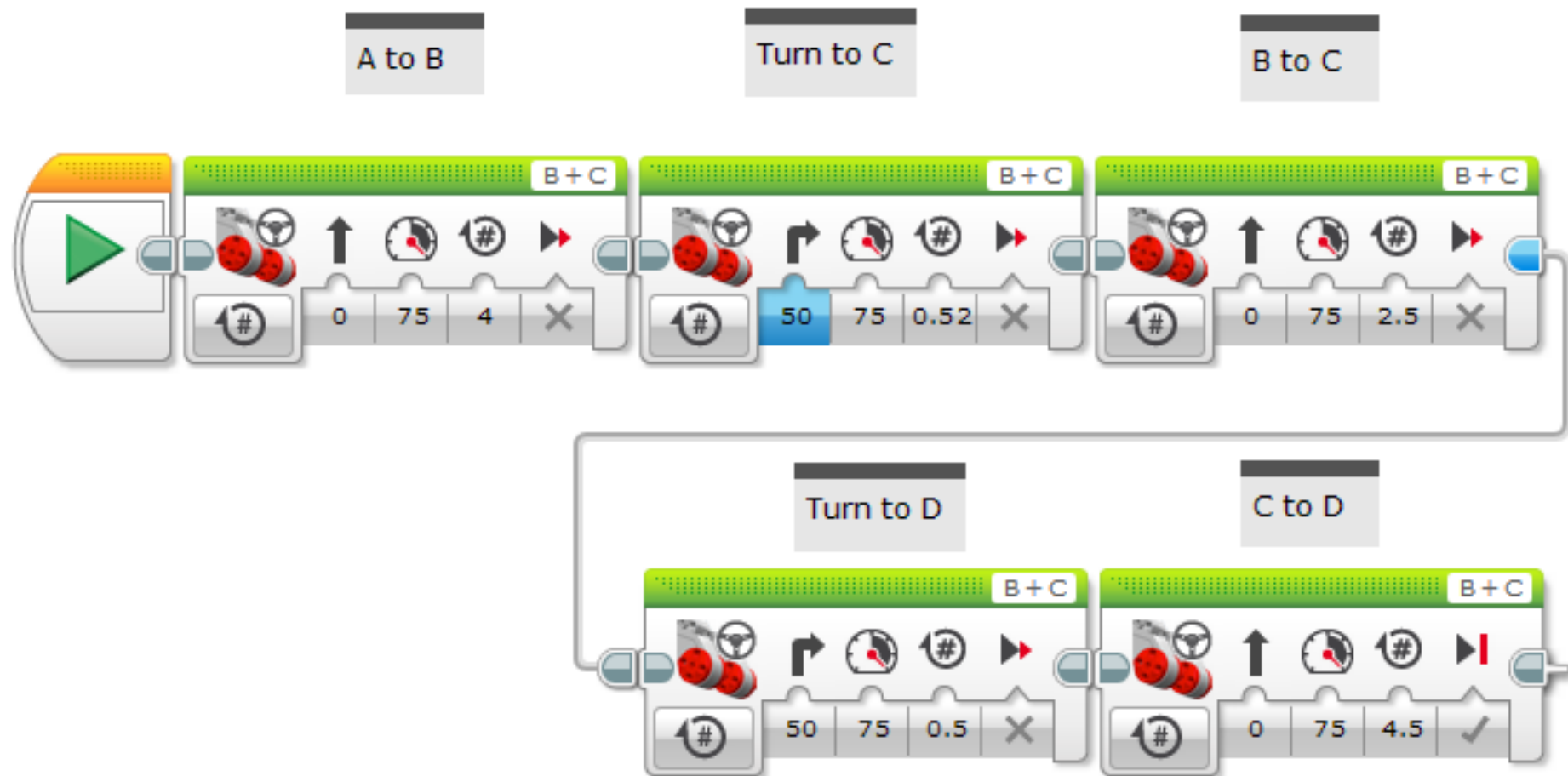
Task:

**Given rectangle ABCD.
Move from A to B to C to D.
Extra Credit: A to C to D to A.**





Lab 1 An Answer





Problem Solving



Generic Problem Solving Process

Define the problem

Brainstorm solutions

Evaluate solutions Pick one

Try (implement) it

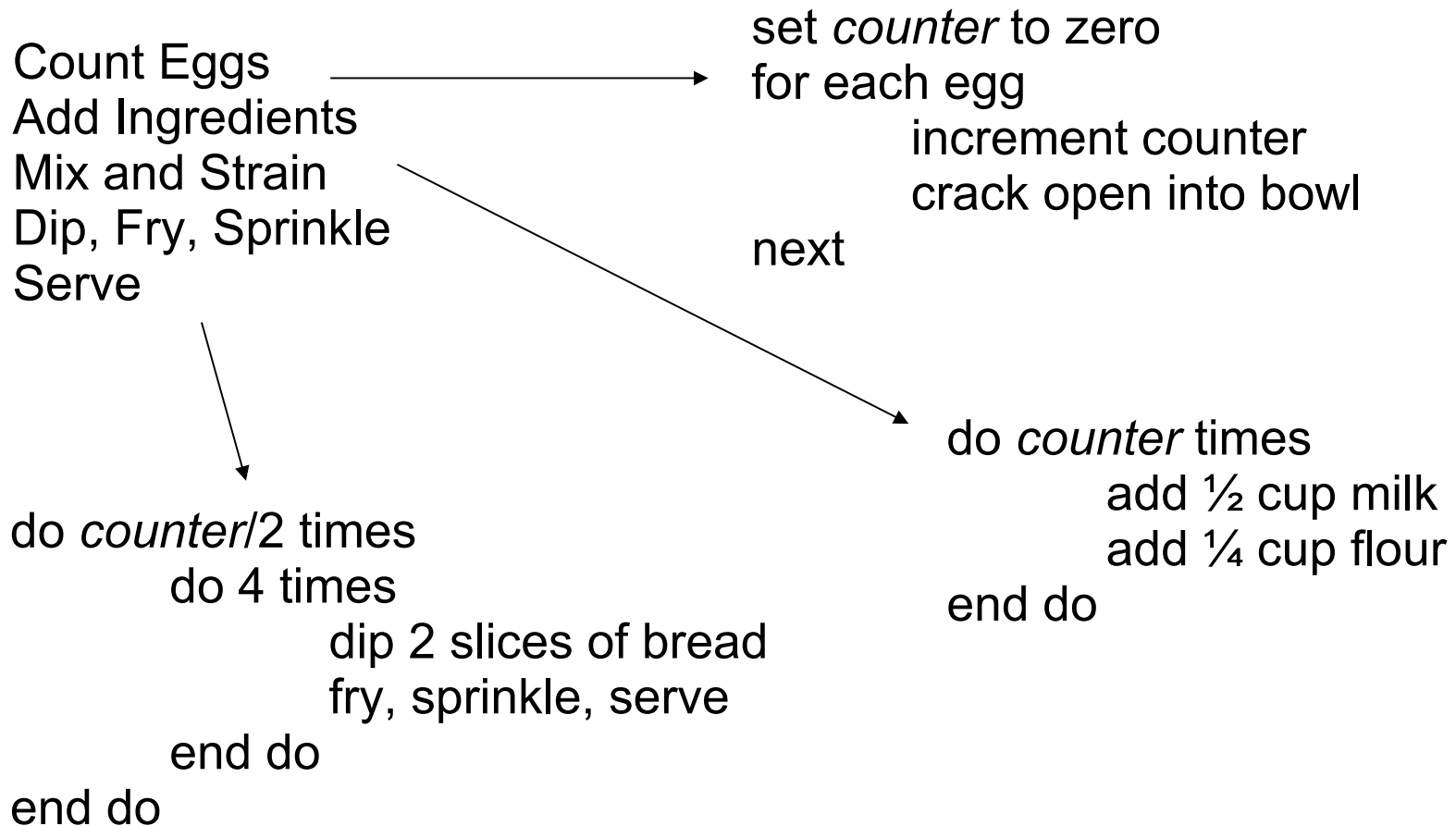
Evaluate results

Express the solution as an algorithm, then
convert it into a computer program.



Pseudocode

Pseudocode: Not software code, not English, but somewhere in between.

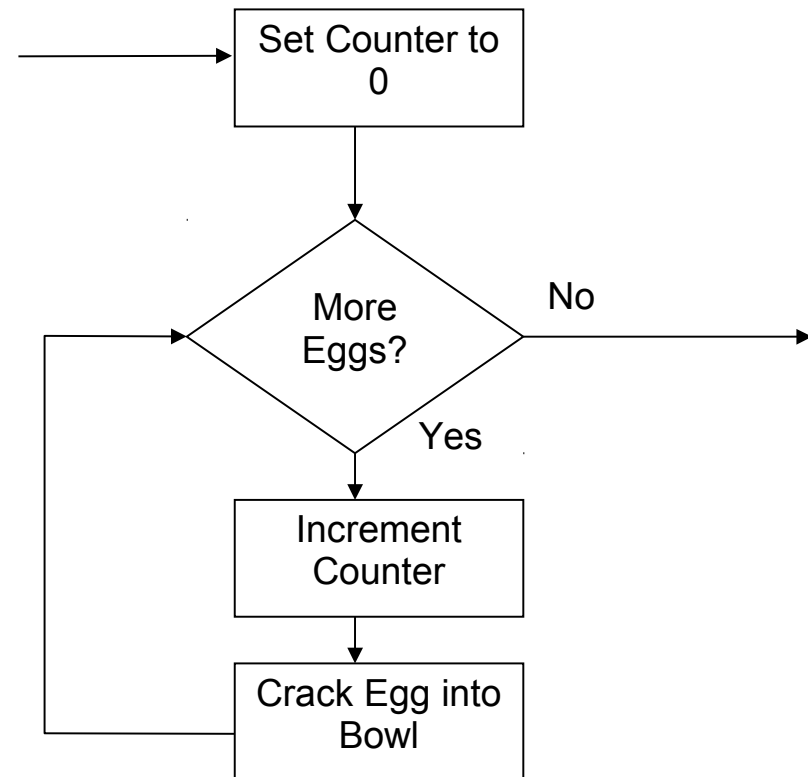




Flowcharts

Flowcharts: A graphic representation of logic.
Convert from pseudocode.
One step closer to software.

set counter to zero
for each egg
 increment counter
 crack into bowl
next





Debugging and Analysis

Split into pieces. Test each piece.

Do little pieces at a time

For example, get the robot to where it needs to be first, then work on getting it to do something

Reuse pieces that work

For example, you know how to turn 90°

Brainstorm a new solution

Look outside your box.

Ask for help.



Keep It Simple Strategies

KISS #1: Subroutines
#2: Comments
#3: Loops



KISS #1: Subroutines

Wrap a complicated process into a neat and tidy package.

Once wrapped, just worry about the package.

In EV3, Subroutines are MyBlocks
Select from the MyBlocks tab.

It starts out empty.





Subroutines: When to Use

To do the same thing from different places.

Reuse

To divide a task into pieces.

Modules

To hide complex details.



MyBlock Names

Useful and informative

ClearSoccerField *not* Csf_amy_3a

12 characters visible on a MyBlock

15 characters visible on the NXT



Suggest using “action + to + target”:

Fwd2Wall or ForwardToWall or Forward_To_Wall

FwdDist

TurnRight

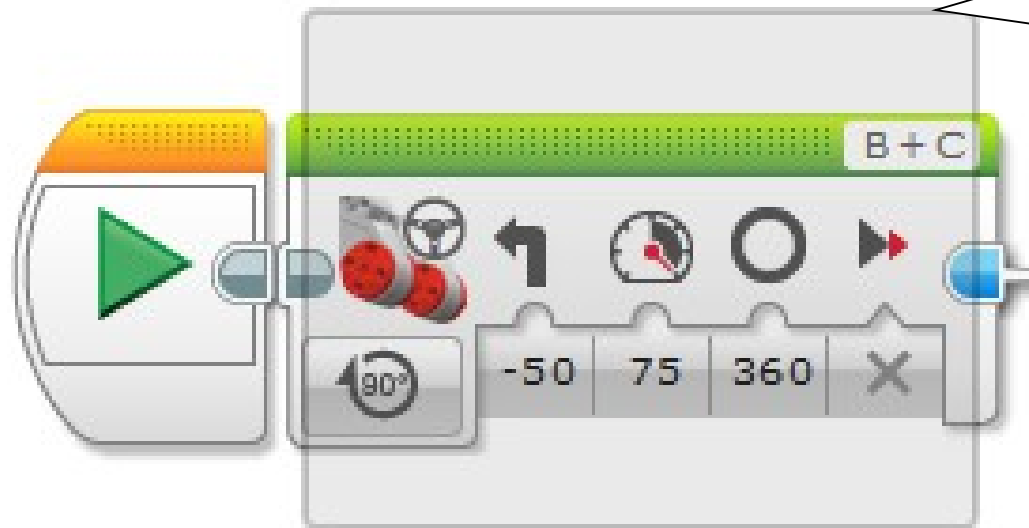
Name the task accomplished, not how it was done.

FollowLine *not* FollowLine1LightSensor



MyBlock Creation 1

Carefully select a section of working code.

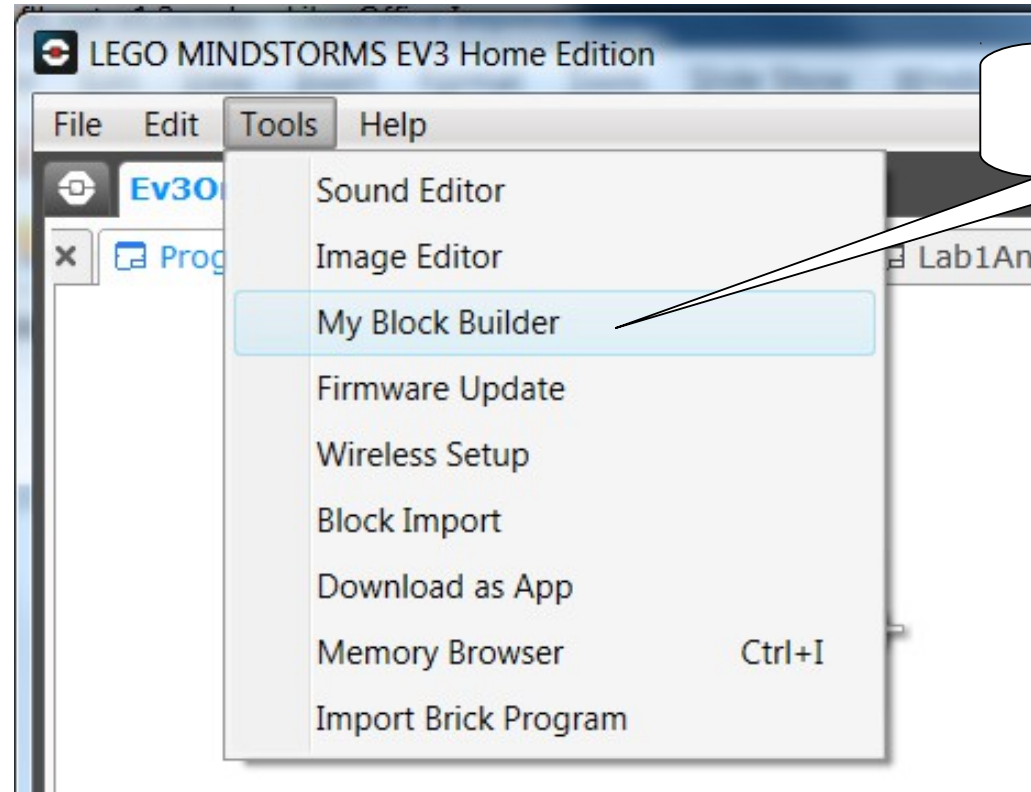


Click and drag a selection box around a set of blocks.



MyBlock Creation 2

Click the
My Block
Builder



My Block
Builder



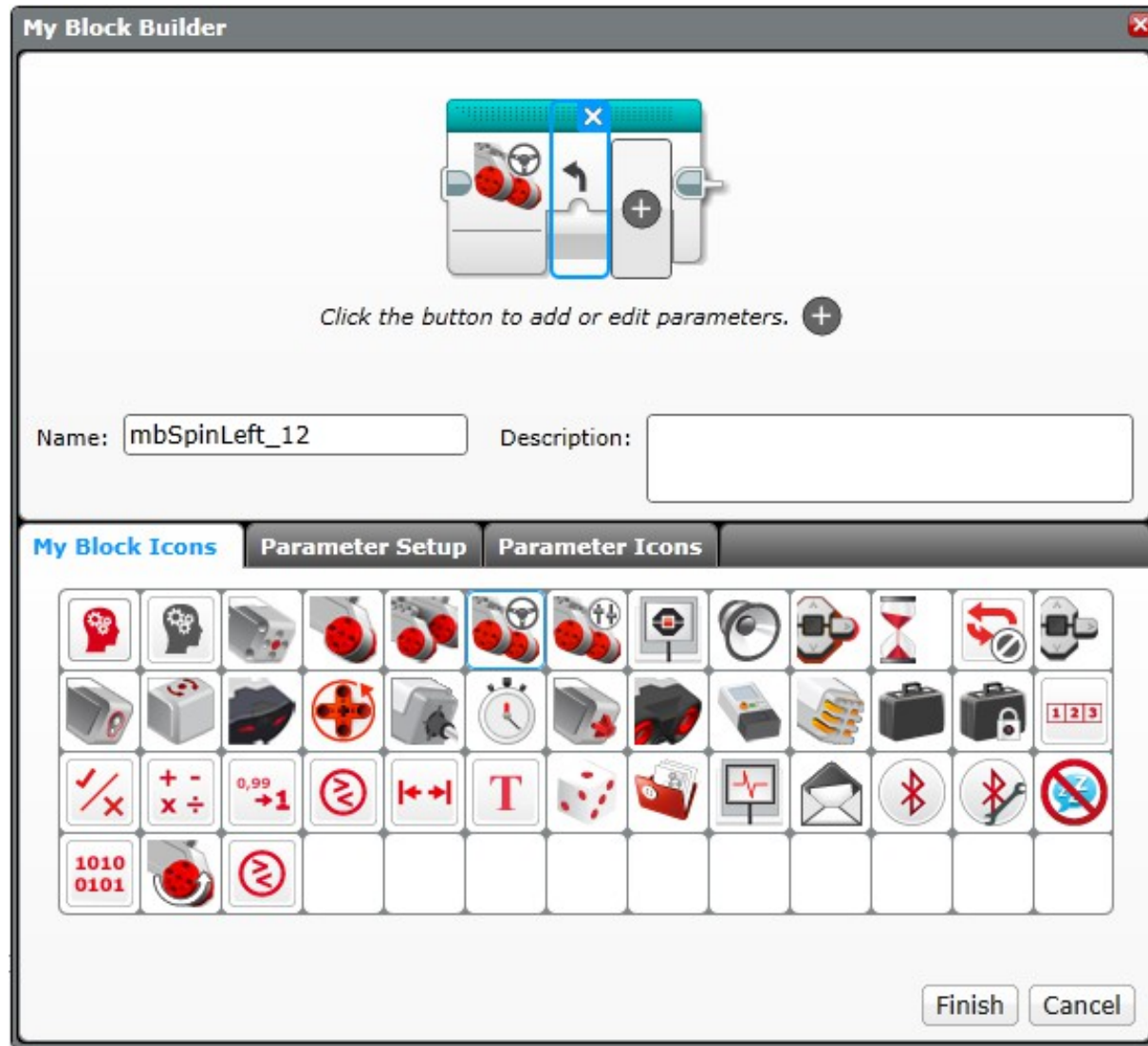
MyBlock Creation 3

Name it.

Describe it.

Pick an icon.

Select a
Parameter
and
“Parameter
Setup”





MyBlock Creation 4

For each
parameter
Name
Input/Output
Data Type
Default Value
Finish

Click the button to add or edit parameters.

Name: Description:

My Block Icons **Parameter Setup** **Parameter Icons**

Name:

Parameter Type: ☒ Input ☐ Output

Data Type:

Default Value:

Parameter Style:

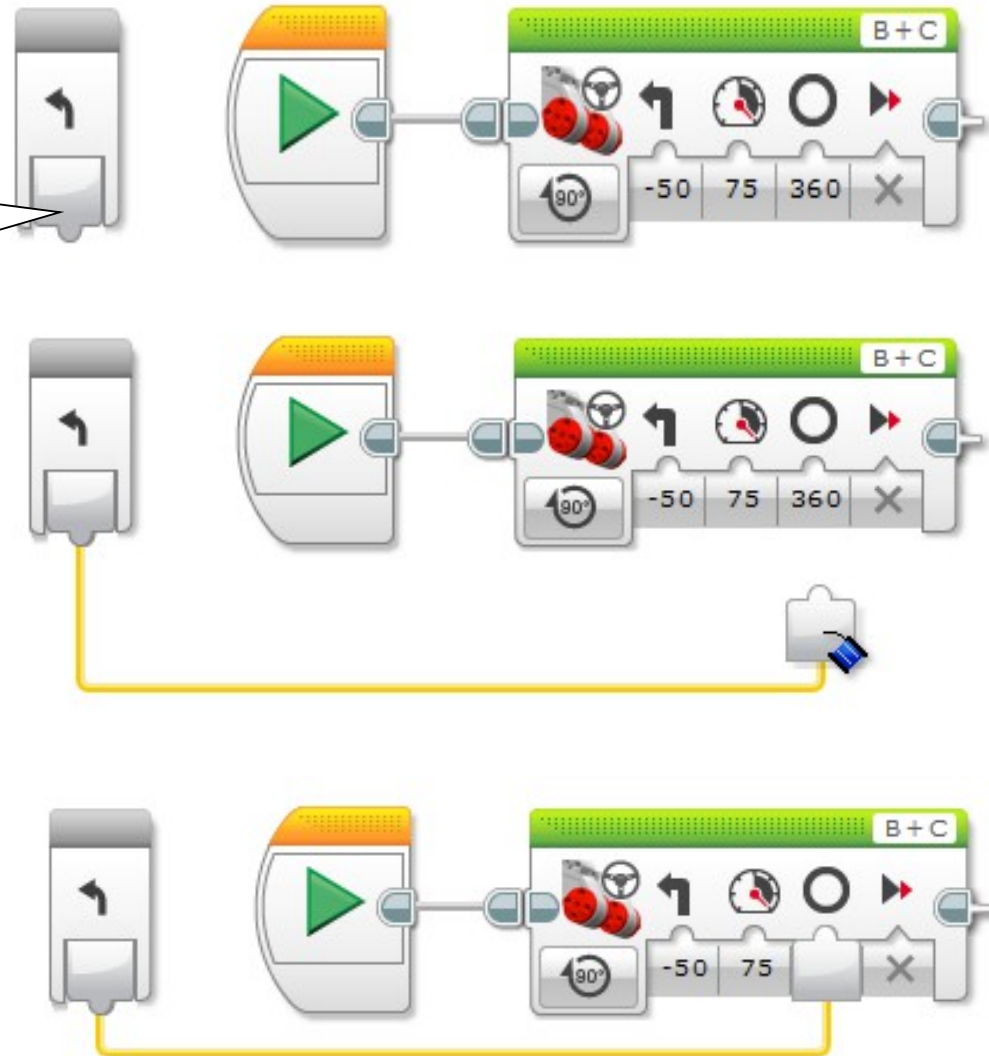
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>



MyBlock Creation 5

Rewiring may be required

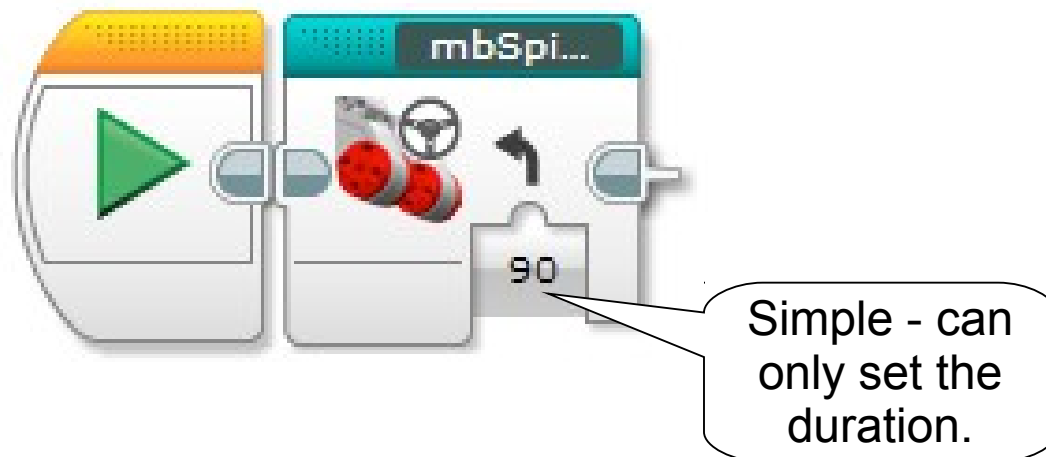
Input parameter needs wiring



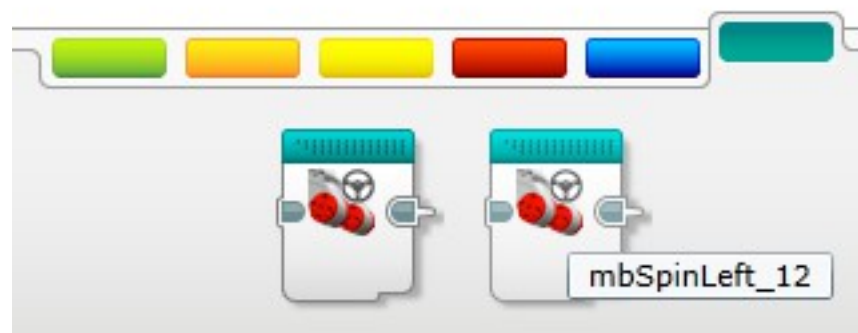


MyBlock Creation 6

Only the spin duration can be set.

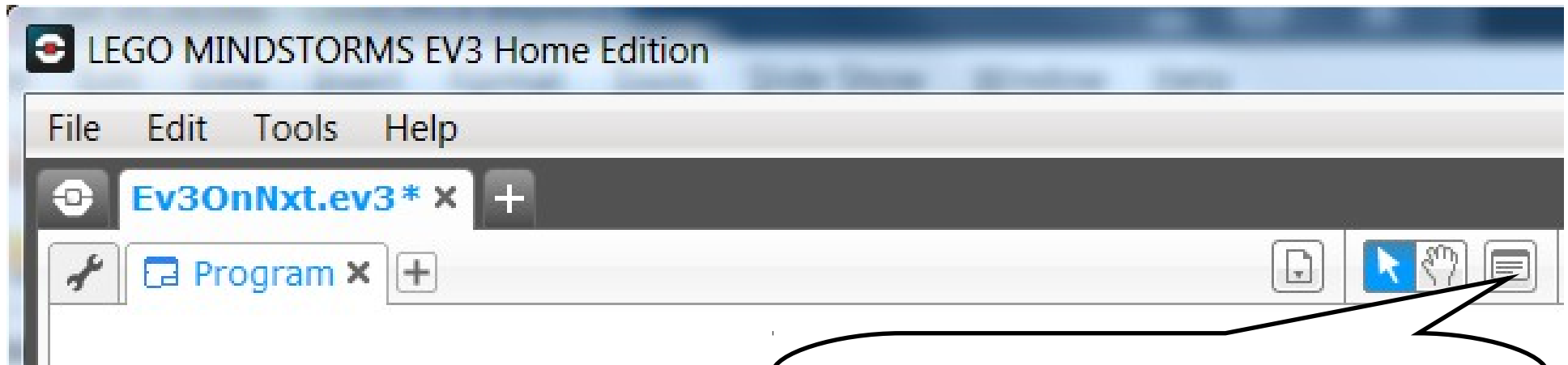


To add the new MyBlock to a program, select it from the MyBlock Tab.





KISS #2: Comments



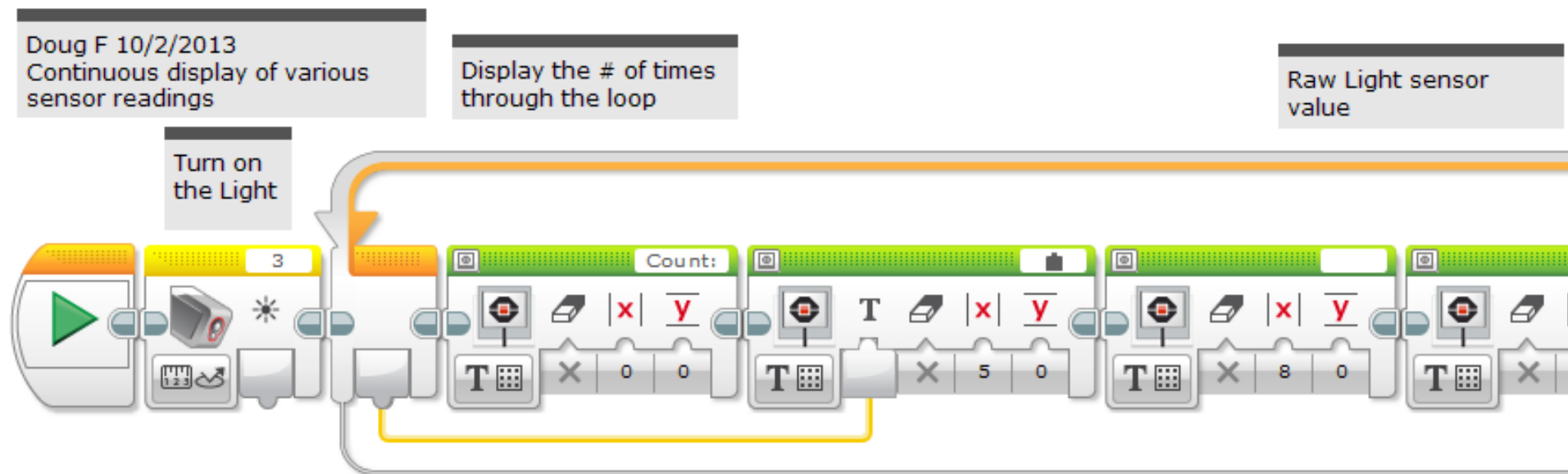
Explain the program.
Programmers forget.

Teams compete in FLL. More than one
person will be working on the program.

Click in the program, then click here and
a comment box will appear. Click the
comment box and start typing.
Drag borders to re-size the box.



Comment Use

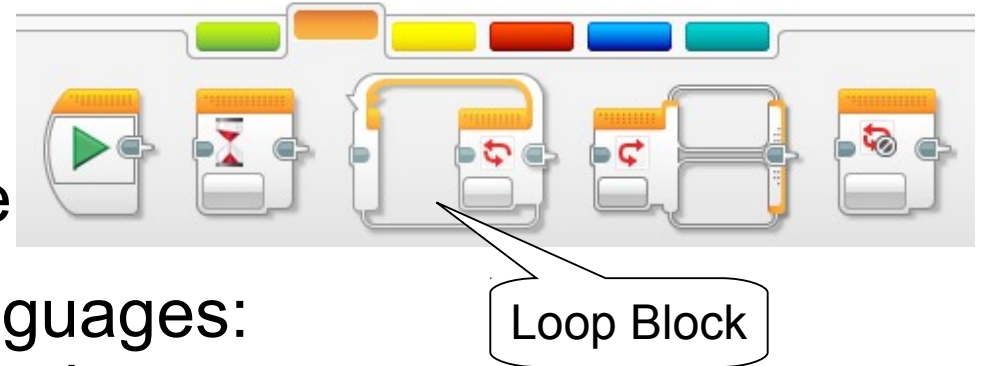


Add who, when, how to use, assumptions taken, and expected results.



KISS #3: Loops

Loops are a control structure



In other programming languages:

For ... Next Do loop n times

Do ... Until Do it. Unless some test, do it again.

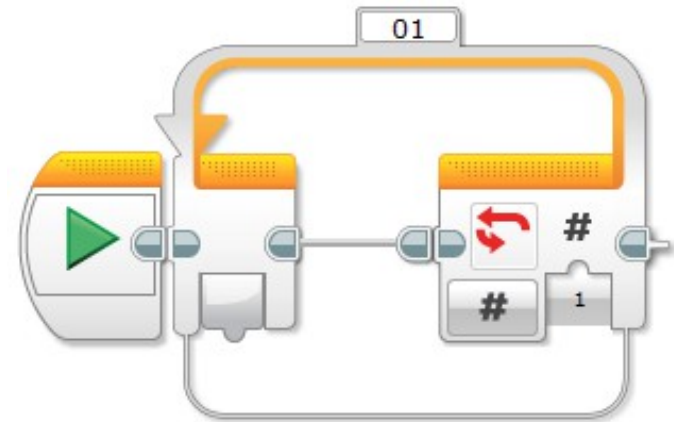
There are loops for

Forever

Every sensor (including time)

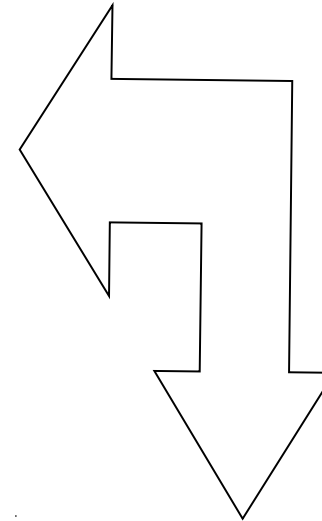
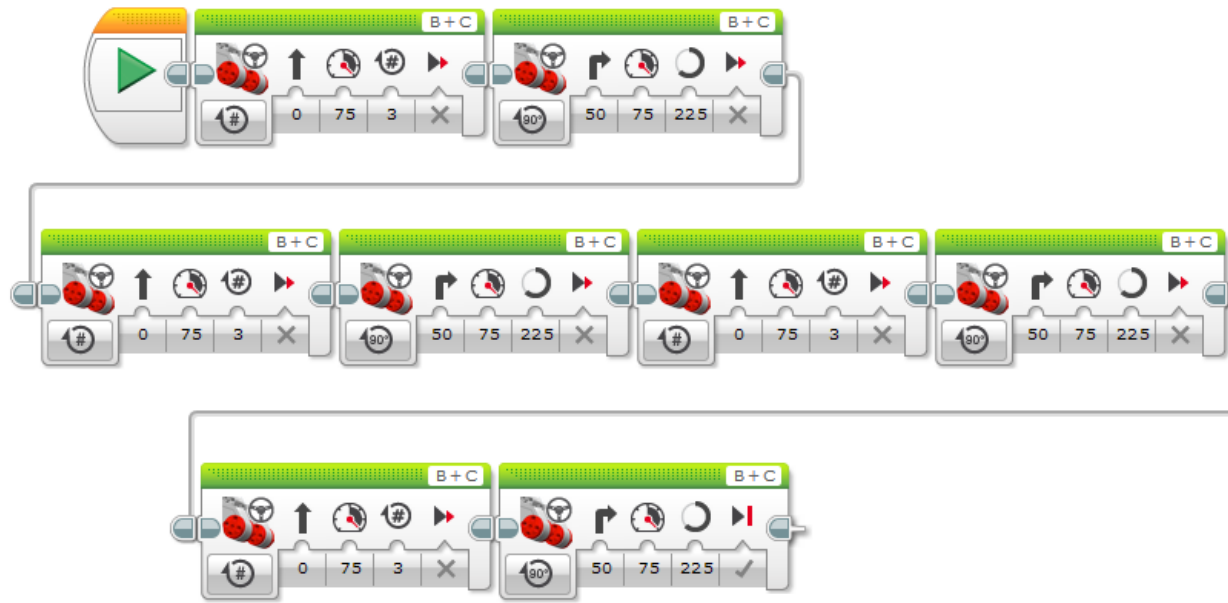
Logic

Count

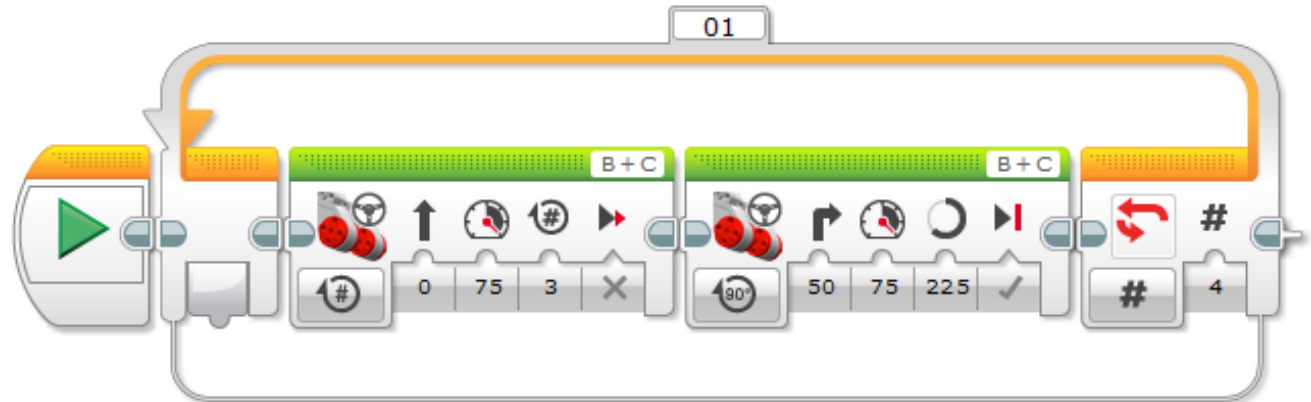




Simple Loop



Convert this
to something
simpler using
a loop





Lab Two

Task:

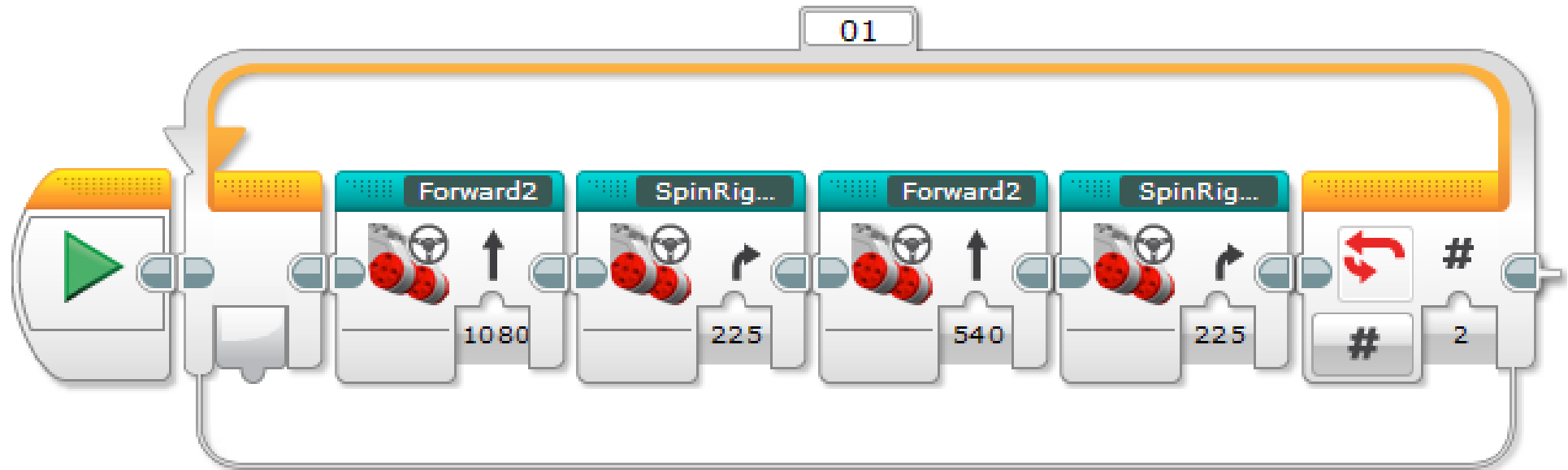
Make Forward and Right Turn MyBlocks.

Move around the rectangle twice, ending at A.





Lab 2 Answer



One possible answer.



Data Input

Sensors



Sensors

Allow your robot to detect the real world.

Touch

Has your robot made contact with something?

Light

Is the surface light or dark?

Sound (Microphone)

Ultrasound (Distance)

Rotation

Embedded in the motors

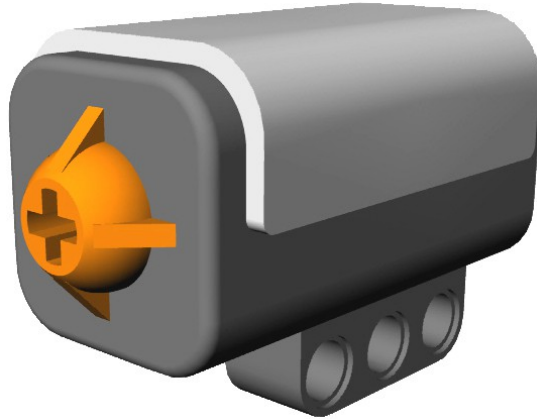
Time

Internal sensor, keeps track of time

Battery Voltage



Sensor #1: Touch



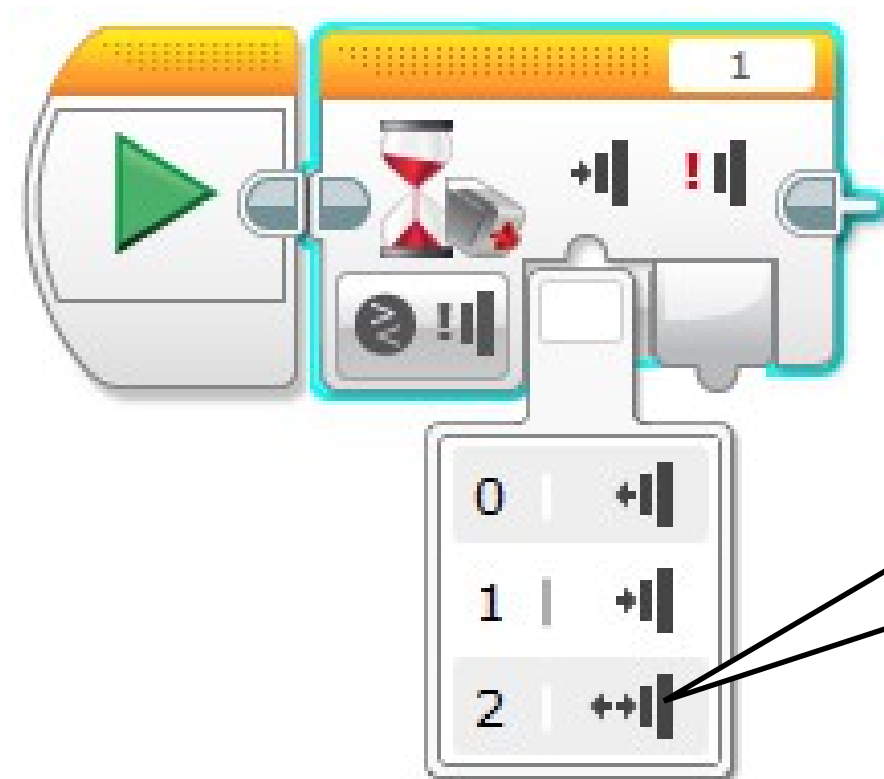
To detect touching or bumping into something.

Good for detecting robot arm movements.

The sensor activates when the arm moves far enough to push in the touch sensor.



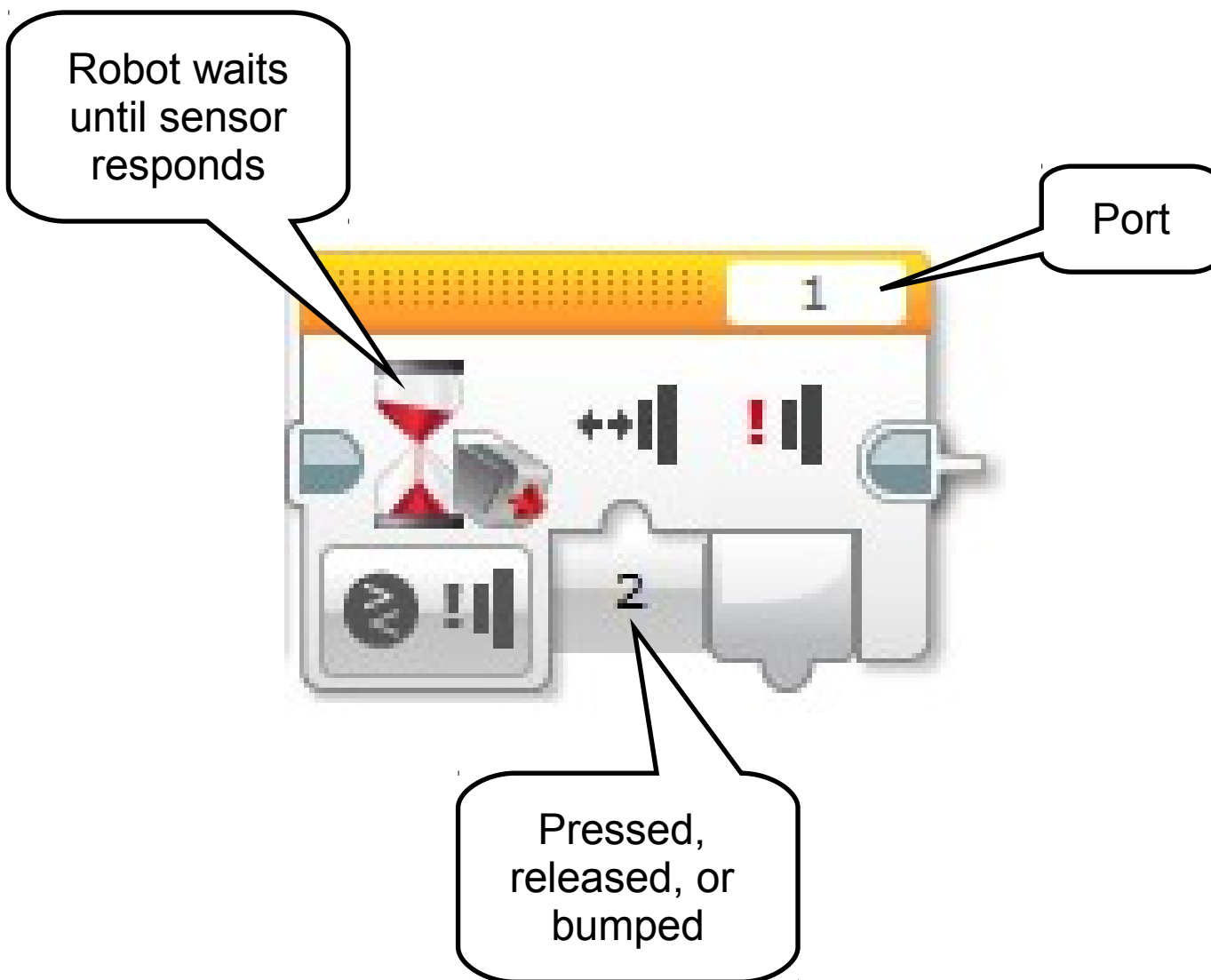
Pressed, Released, Bumped



Bumped: pressed and released in any order.



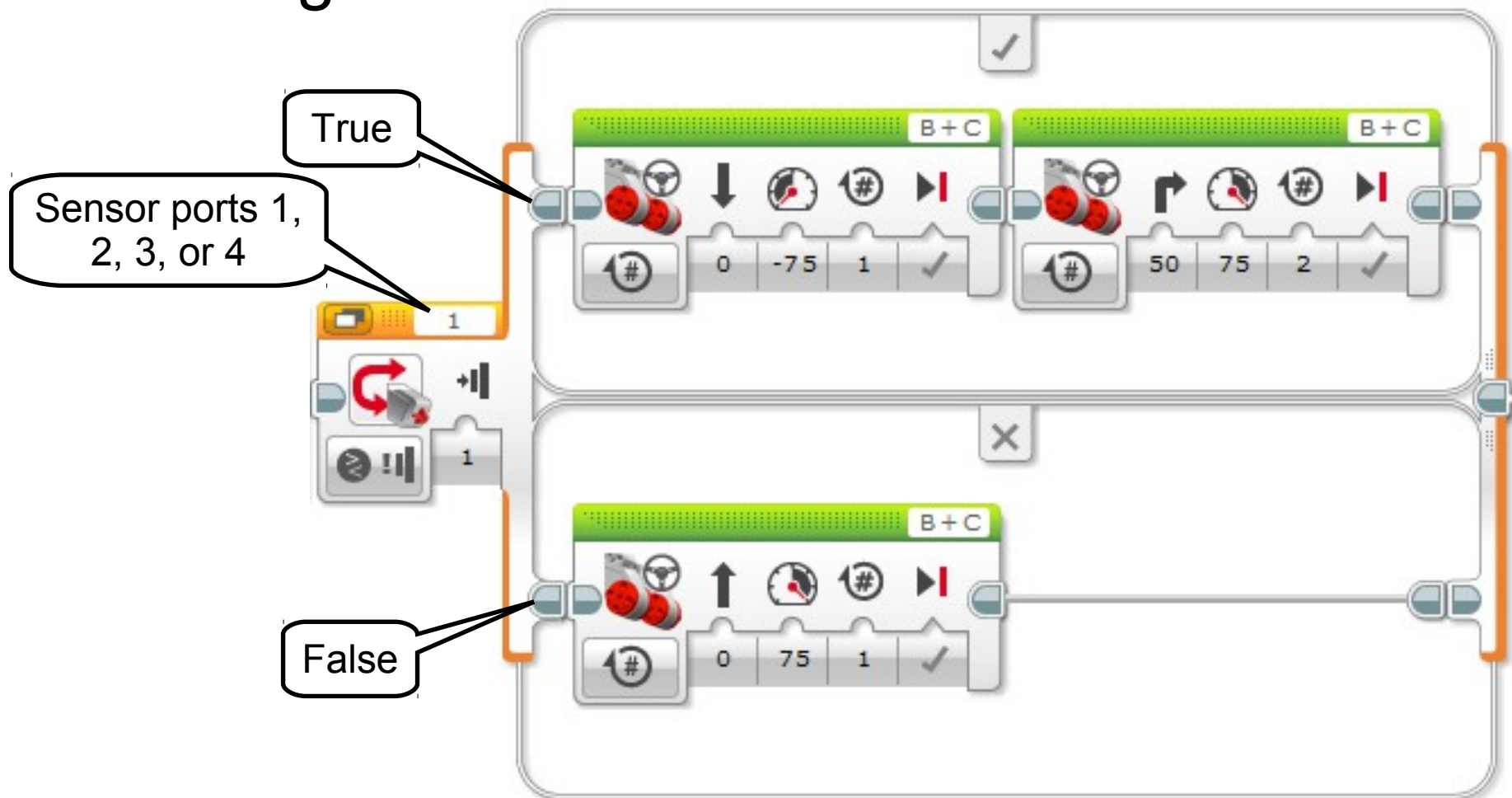
Touch Sensor WaitFor Block





Touch Sensor Switch Block

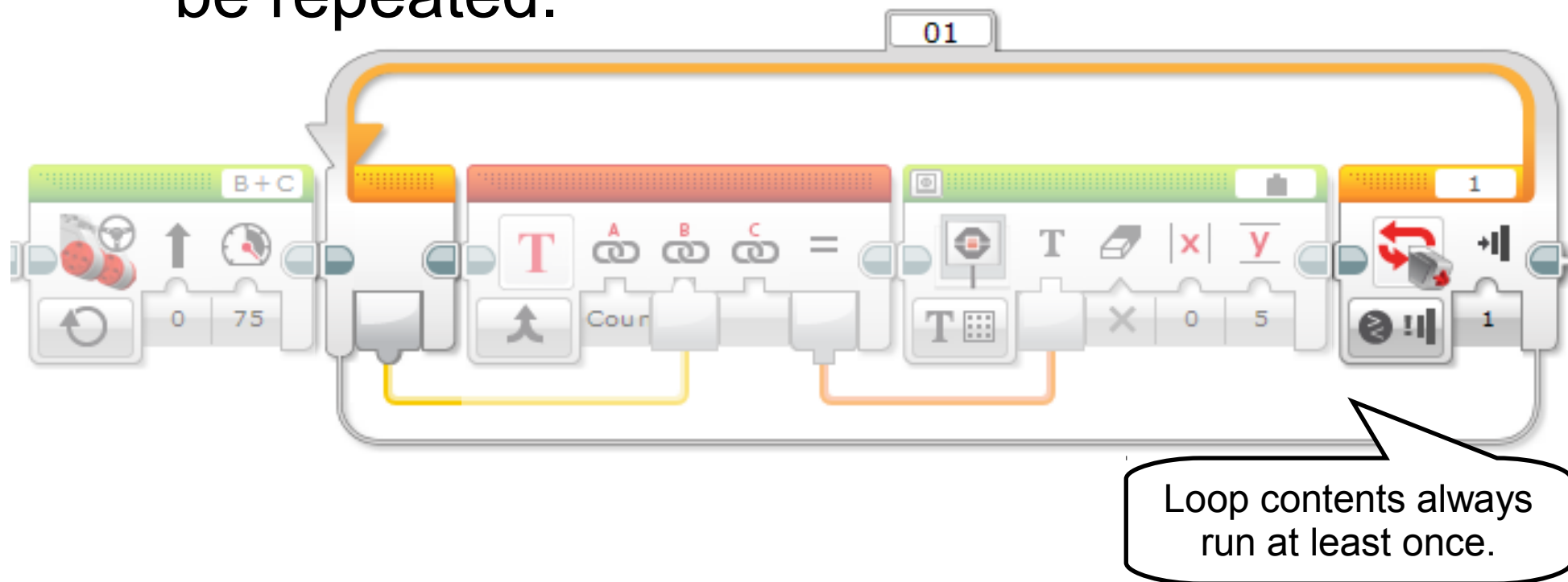
Waiting for a touch sensor can be useful, but many times you want to do different things based on the current value.





Touch Sensor Loop Commands

Loop until a touch sensor is pressed. Useful if the loop contains commands that may be repeated.



For instance, a routine that beeps until a bumper hits something.



Sensor #2: Light

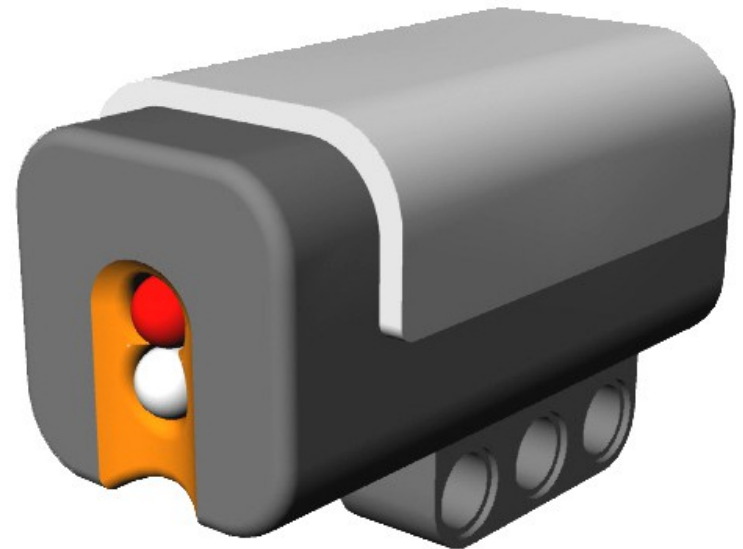
Operates in "percent" mode
0 to 100

Higher number = more light. A
lighter surface reflects more
light.

Calibrate the sensor.

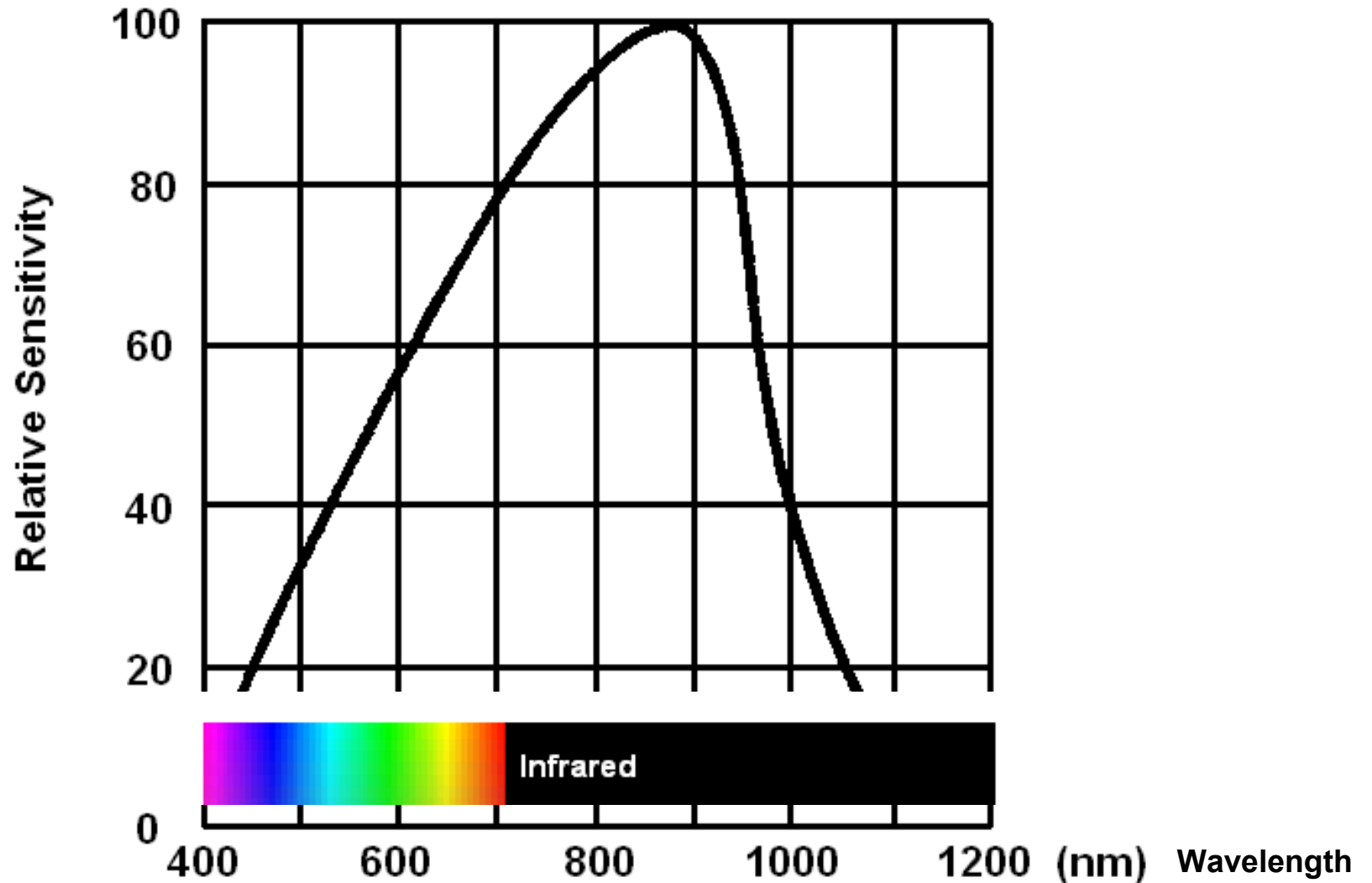
Light can be turned off.

Shines a red light.





Light Sensor Spectrum



Most sensitive to red/IR light.



Light Sensor Readings

Lowest likely reading 5%

Highest likely reading 100% (pointing at a light)

Readings also depend on the color of the surface

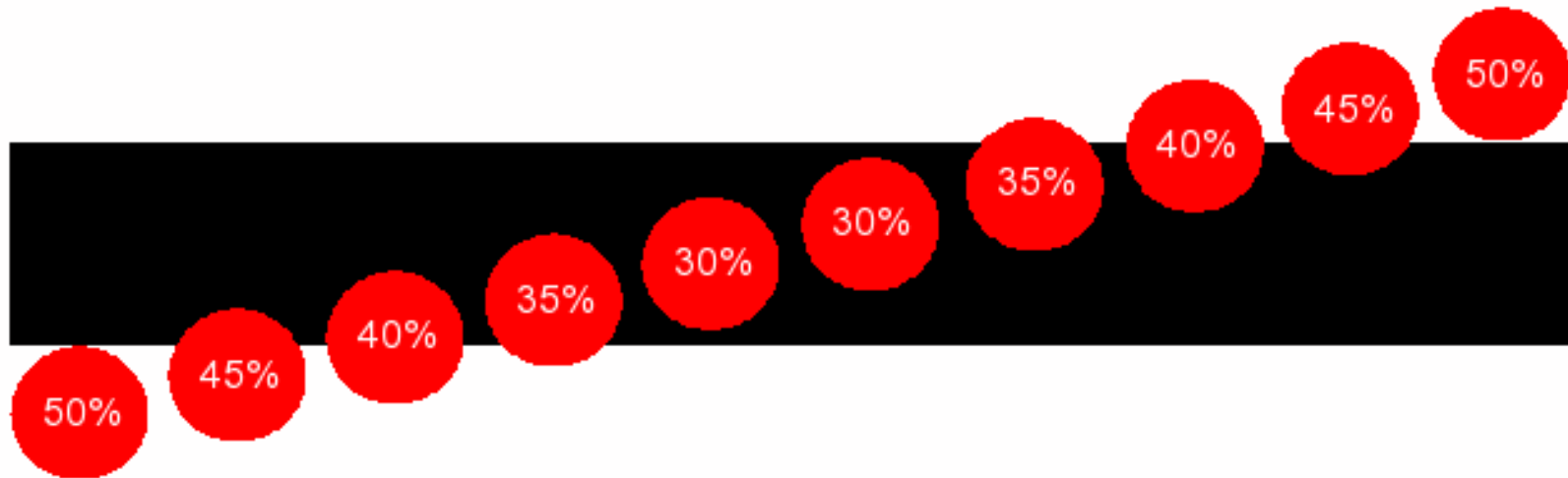
See “Building LEGO Robots for FIRST LEGO League” by Dean Hystad.

Sensitive to the distance between the sensor and the reflecting surface. Variations can make the readings unusable. Keep the sensor close to the surface, but not too close.

Shield the sensor from other light sources.



Light Sensor Readings



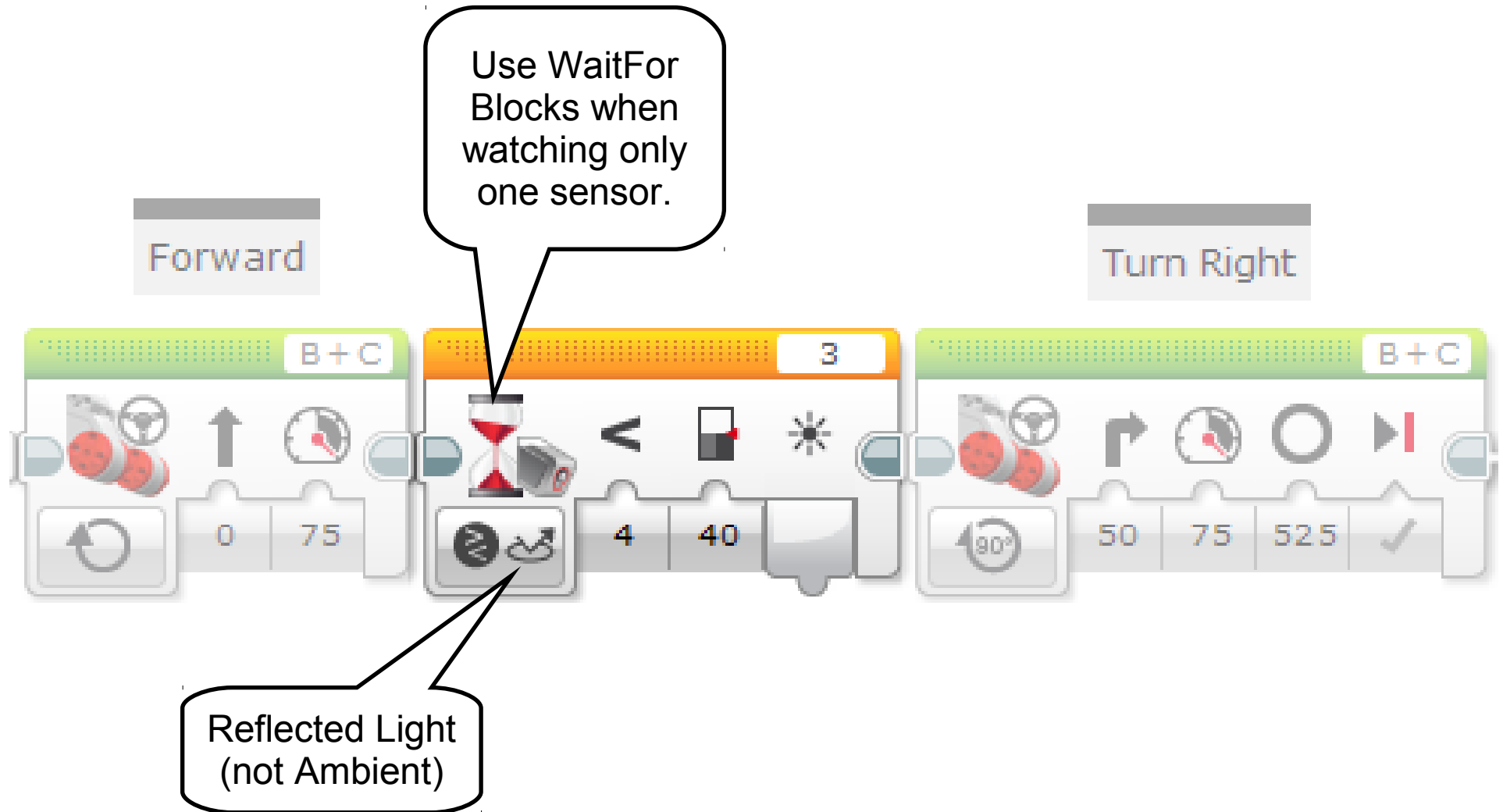
The light sensor averages its readings over roughly a circular area.

Cross a line too fast and you may miss the line.

Test and recalibrate on competition day.

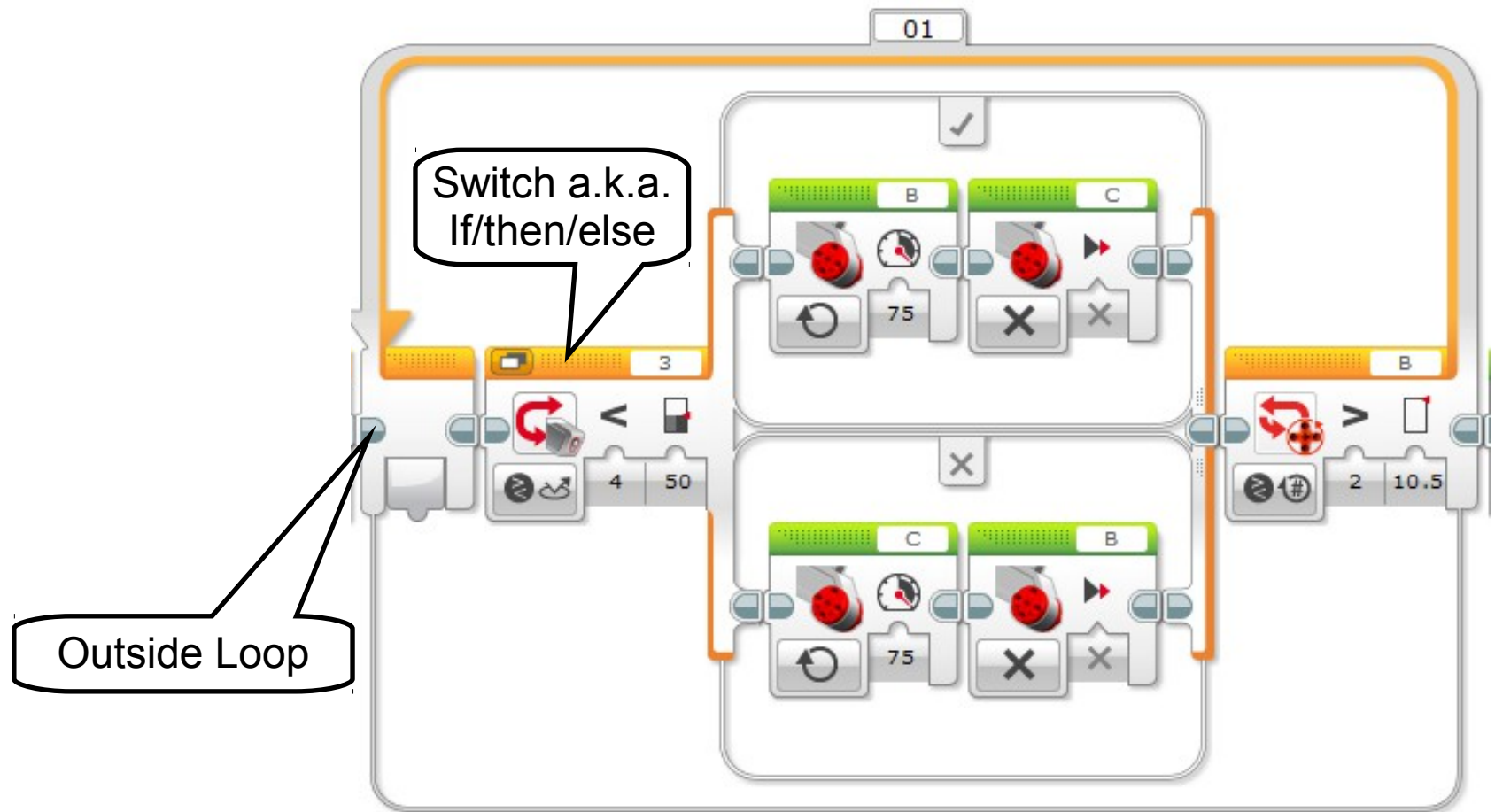


Light Sensor WaitFor Block





Light Sensor Switch Block

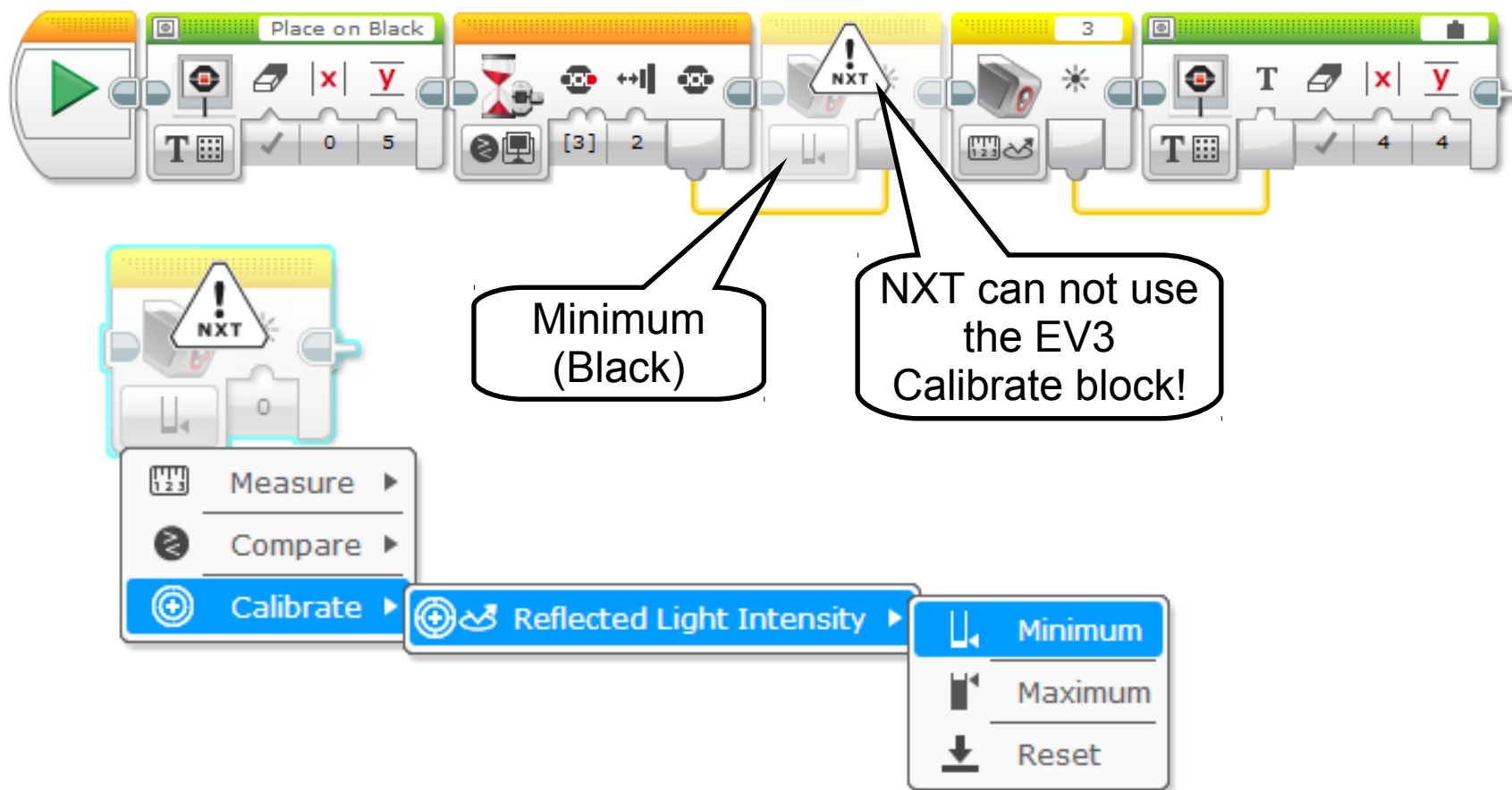


If darker than 50, turn motor B on. Stop C.
Otherwise, turn motor C on. Stop B.

What is this?



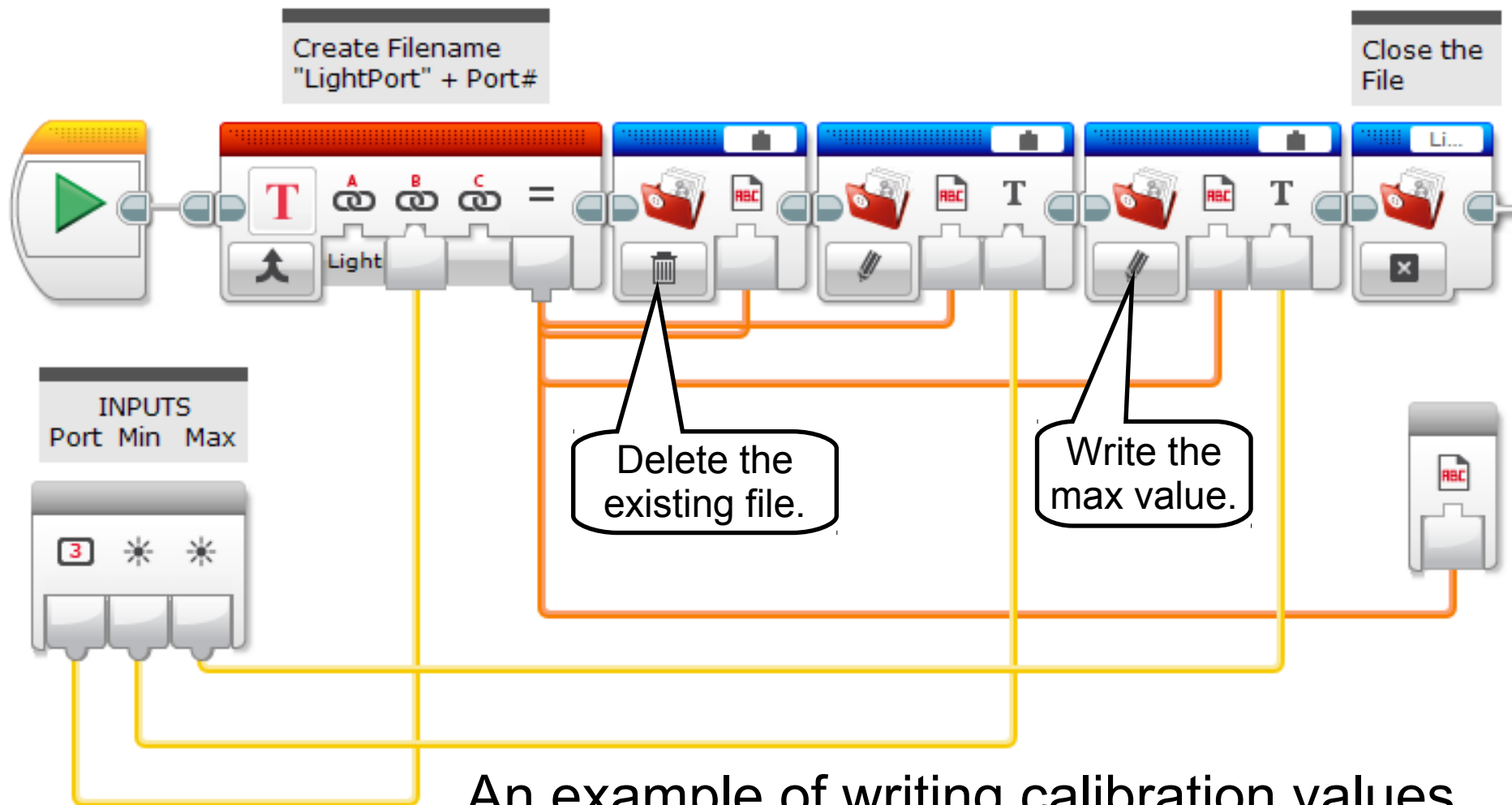
Calibrate Light Sensor



Move your robot over light and dark areas.
Resets light sensor percentages.



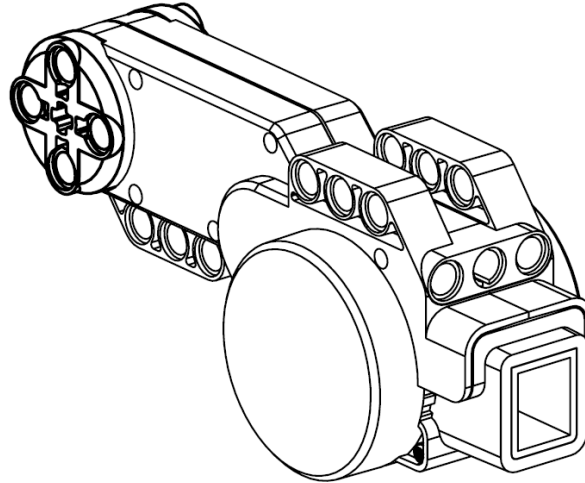
Calibrate Light Sensor +



An example of writing calibration values to a file on the NXT.



Sensor #3: Rotation



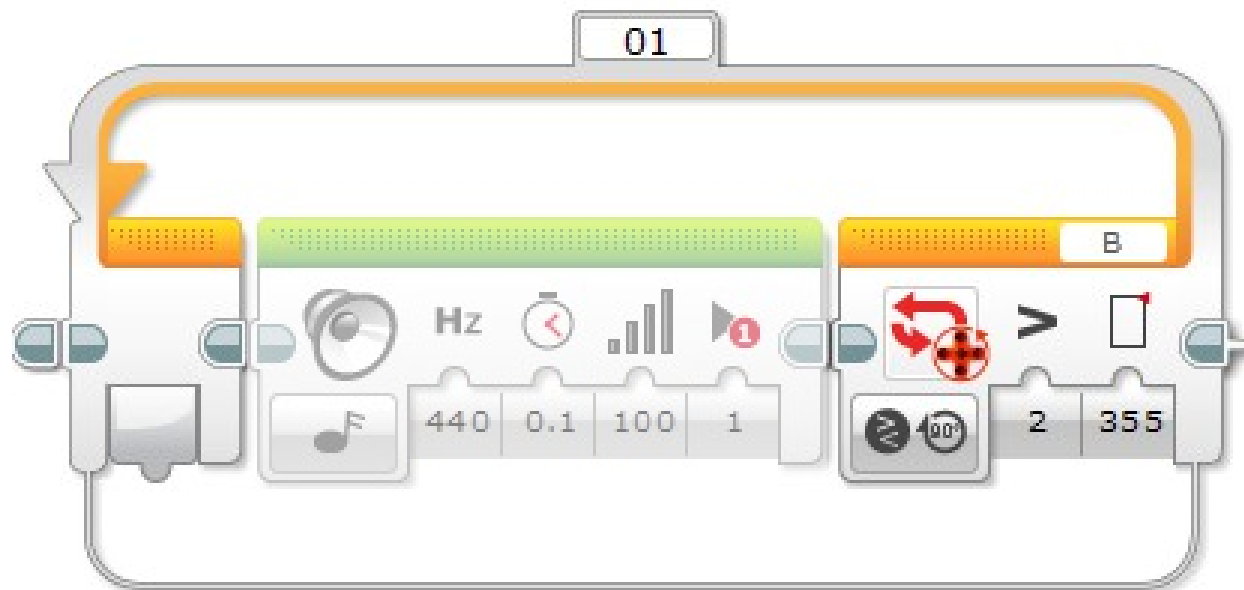
Measures how far a rotating axle has turned. As the axle turns, a counter in the NXT is incremented or decremented.

360 counts per rotation.

Each motor has an embedded rotation sensor.



Rotation Sensor Loop Block

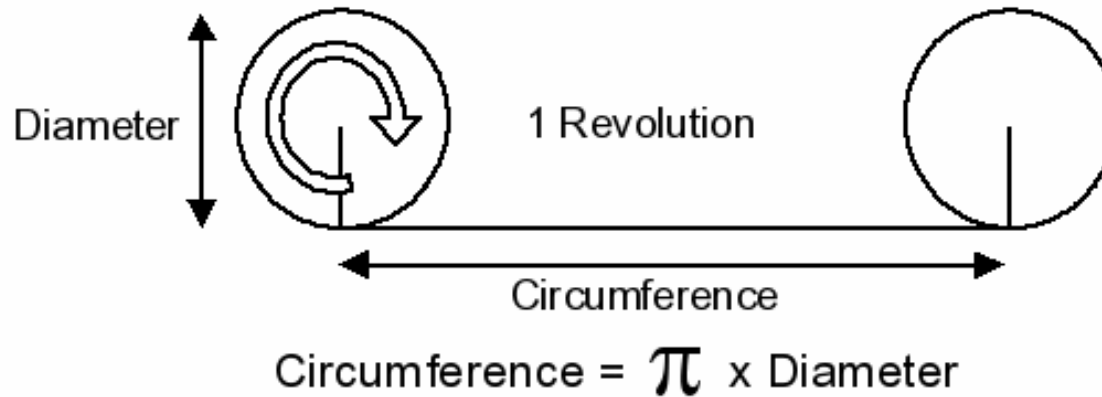


Possible to make a rotation loop.
Select any loop. Pick “Sensor” type and then
“Rotations.”



Calculating Distance

The rotation sensor also brings in the possibility of doing some real math!



We'll leave that as an exercise for the reader!

Of course, trial and error also works.

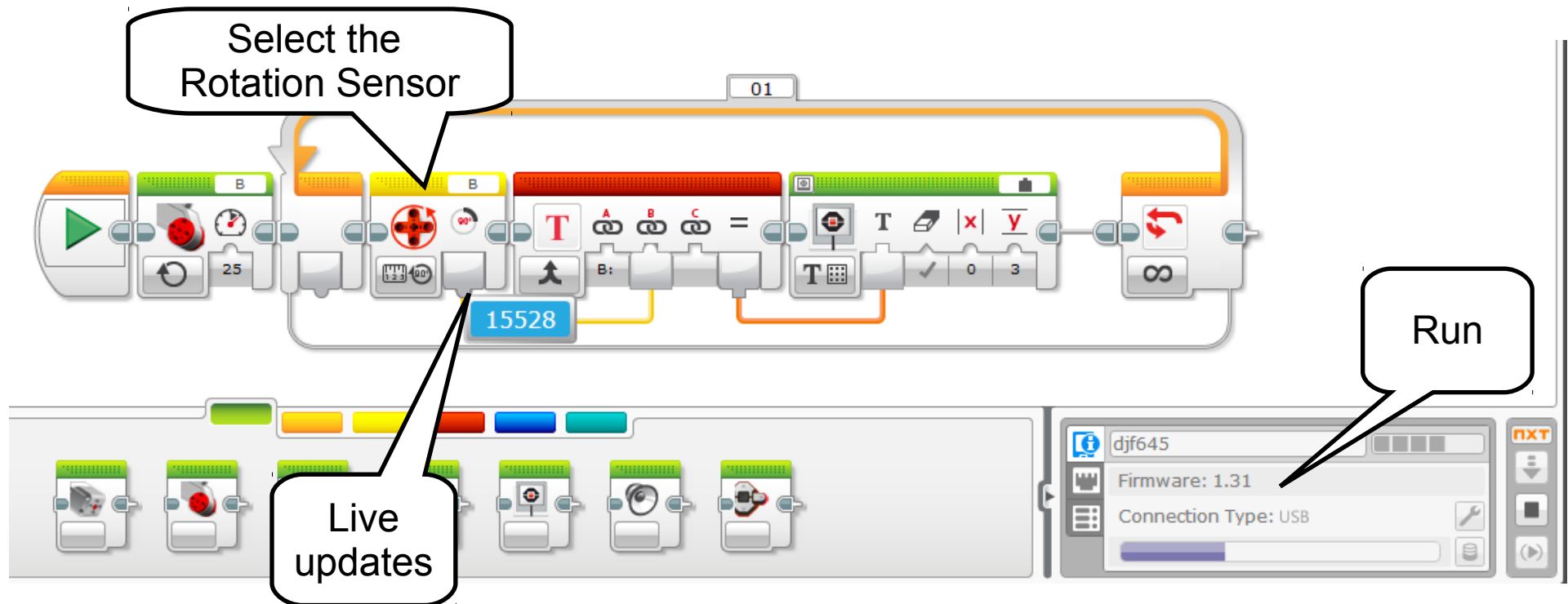
Sources of error in calculation - dirt on surface, using a skid rather than a wheel, backlash (poor fitting gears).



More on Rotation Sensor

Rotation sensor **counts forward and backwards.**

Live updates (Bluetooth or USB connected).





Debugging and Analysis

Common problems

Programming: **reset the sensor to zero** before use.

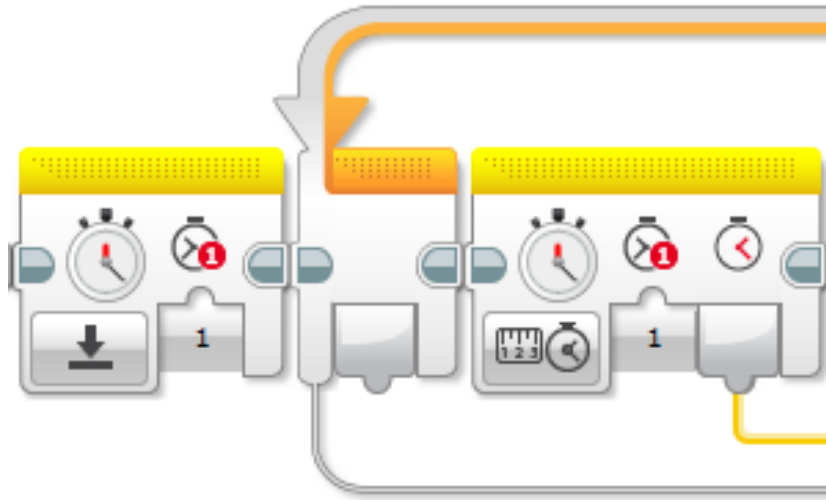
Design: inadequate sensor resolution (trying to measure something very accurately, when the sensor is not that accurate).

Control: starting, stopping, turning **too fast**.

Variations in the initial conditions: not putting everything in the same place before pushing the run button.



Sensor #4: Timer



8 Timers

Time in seconds

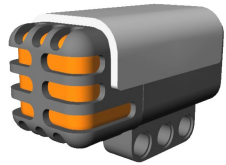
Greater or Less than
tests

Reset

Trigger Point



Other Sensors



Sound (microphone)

Is the FLL competition too loud?



Ultrasonic (distance)

Interference with other NXTs?

NXT Buttons

One touch running of the next program.

Received Messages



Keep it Simple Strategies

KISS #4: Variables

#5: Parallel Sequence



What's a Variable?

A value that you can change during your program.

This value is “variable”, hence the name.

For example, your program may store a light sensor reading in a variable called *LightBright*. Use that value later.

Use a meaningful name.

Useful to pass values to MyBlocks



KISS #4: Variables

LEGO MINDSTORMS EV3 Home Edition

File Edit Tools Help

Ev3OnNxt_v2.ev3 x +

LineFollow x LightSensor x LightSensorViewer x Wait4Dark x

Project Title: Ev3OnNxt_v2 Share Project

1 PROJECT PICTURE 2 PROJECT DESCRIPTION

☐ Daisy-Chain Mode

Programs Images Sounds My Blocks Variables Exportable Items

Type	Name
Numeric	LightMin
Numeric	LightPort
Numeric	LightMax

Copy Paste Delete Add

Add or Delete variables for the whole project on the "Variables" tab.

Variable data type
Number
Text
Logic



Using a Variable

Pick the variable Block

Pick Write, Numeric

Set the value

LightMin

Read

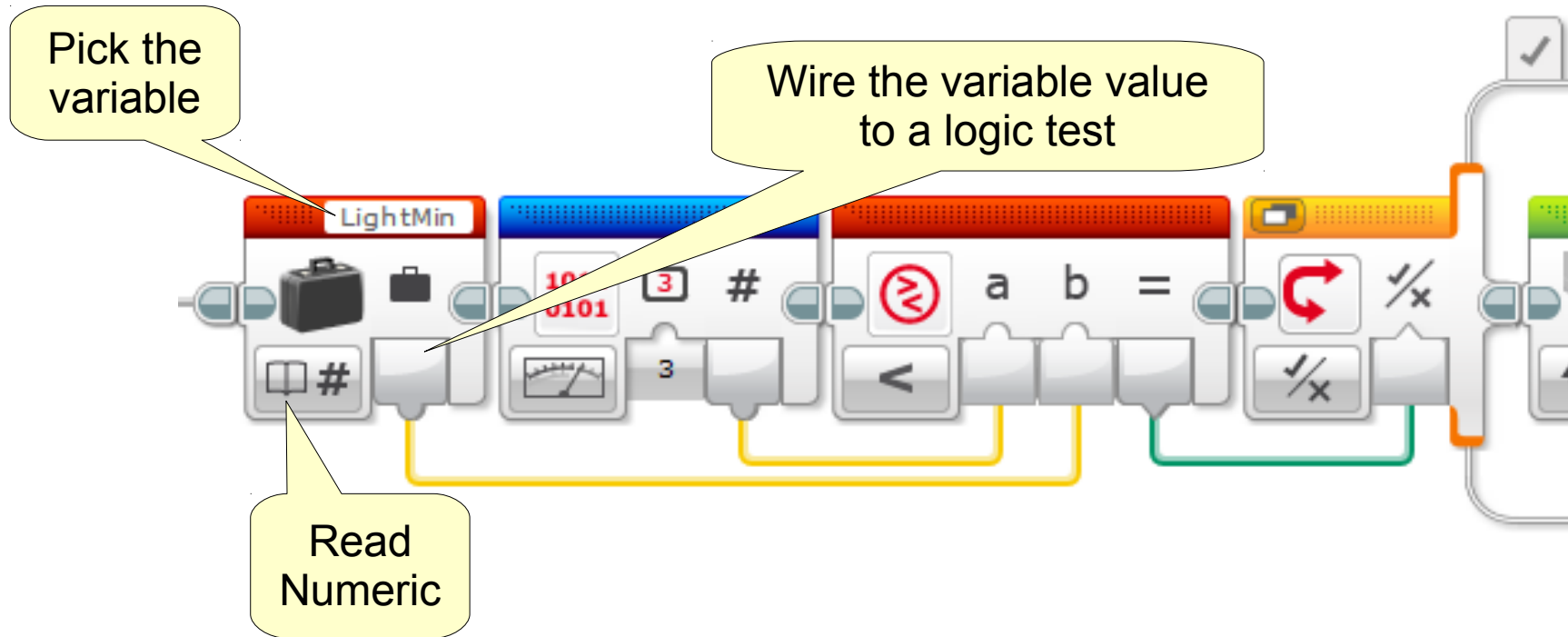
Write

- Text
- Numeric**
- Logic
- Numeric Array
- Logic Array

Variable blocks: [Variable Icon], [Variable Icon], [123], [✓/✗], [+/-], [0.99 +1]

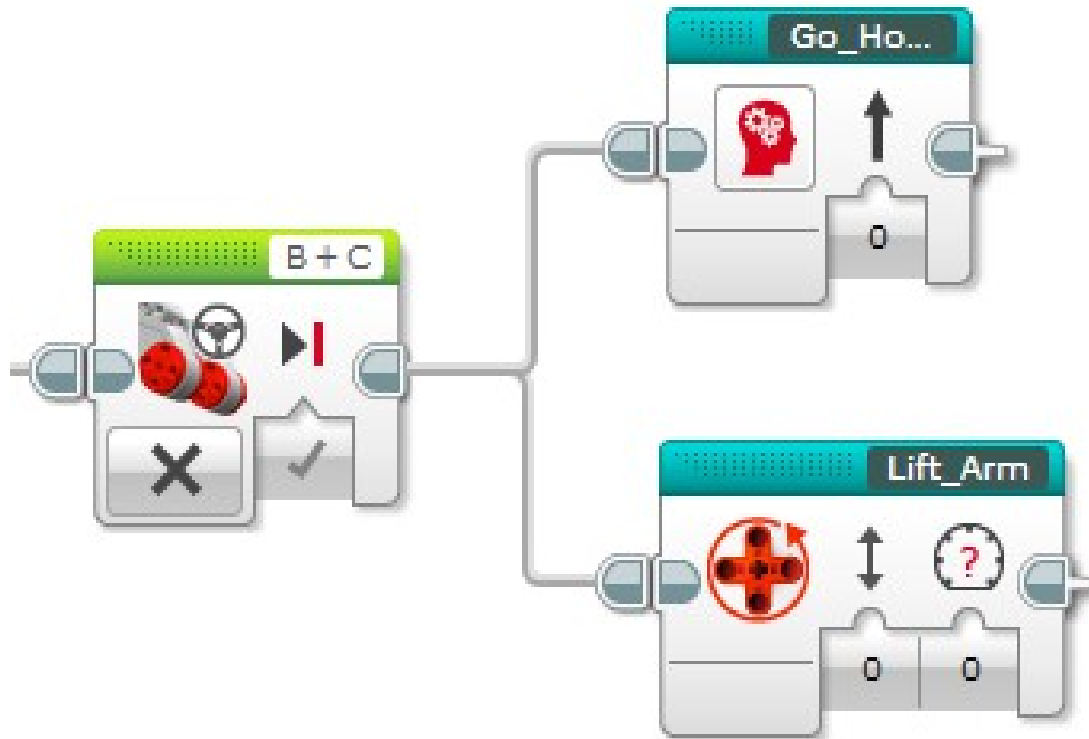


Using a Variable





KISS #5: Parallel Sequences



Make two tasks that run independently.
(Can you walk and chew gum?)

One task lifts the arm.
The other task heads for home.



Lab Three

Task:

Use a light sensor to detect the lines.

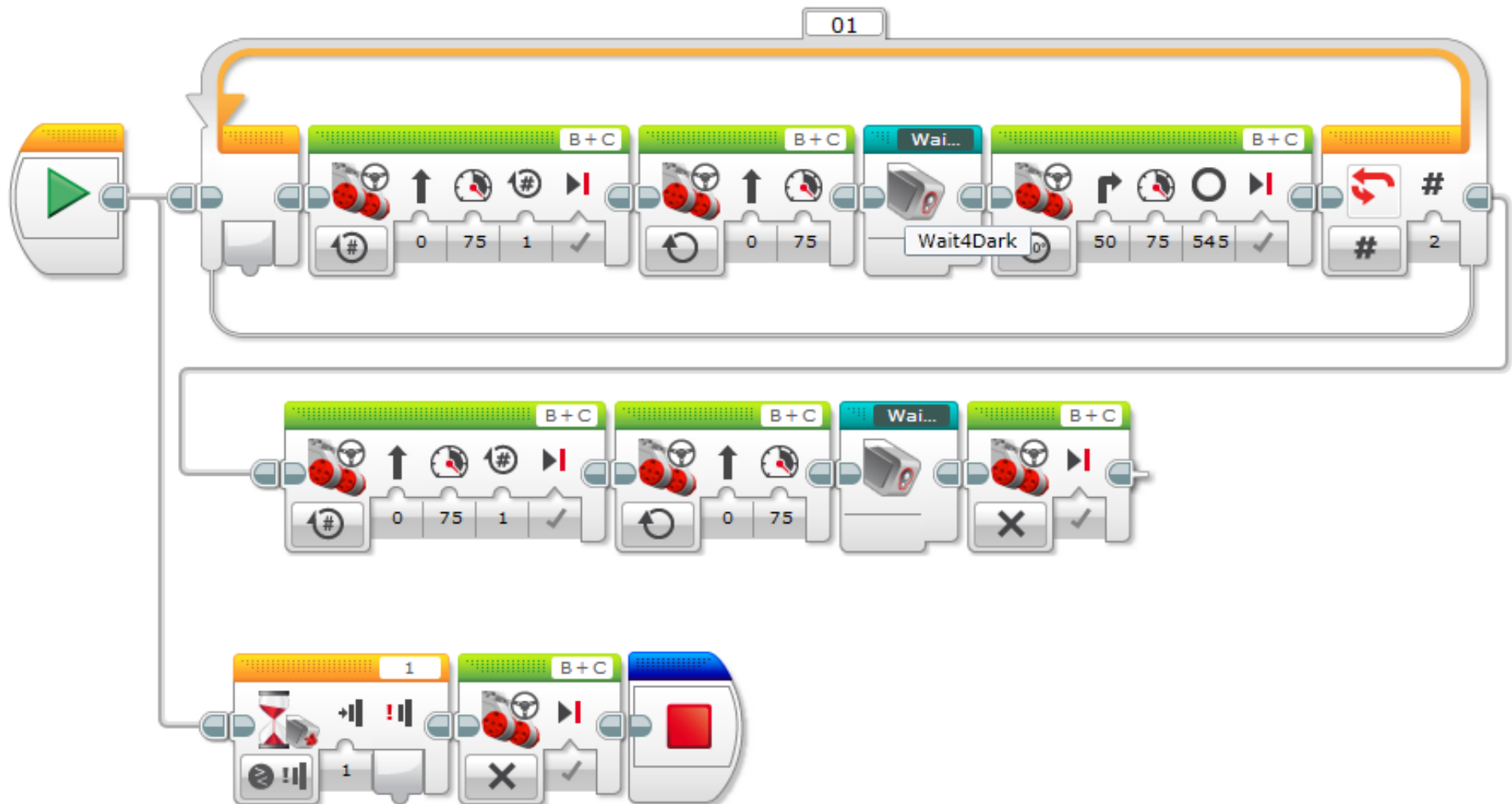
Move from A to B to C to D.

Extra Credit: While moving, touch the touch sensor to stop.





Lab 3 An Answer





Advanced Topics

Debugging Tools

Switch between Cases

File Operations

Bluetooth Communication Setup

Additional Resources



One or two quick notes, a good ear can hear the difference.



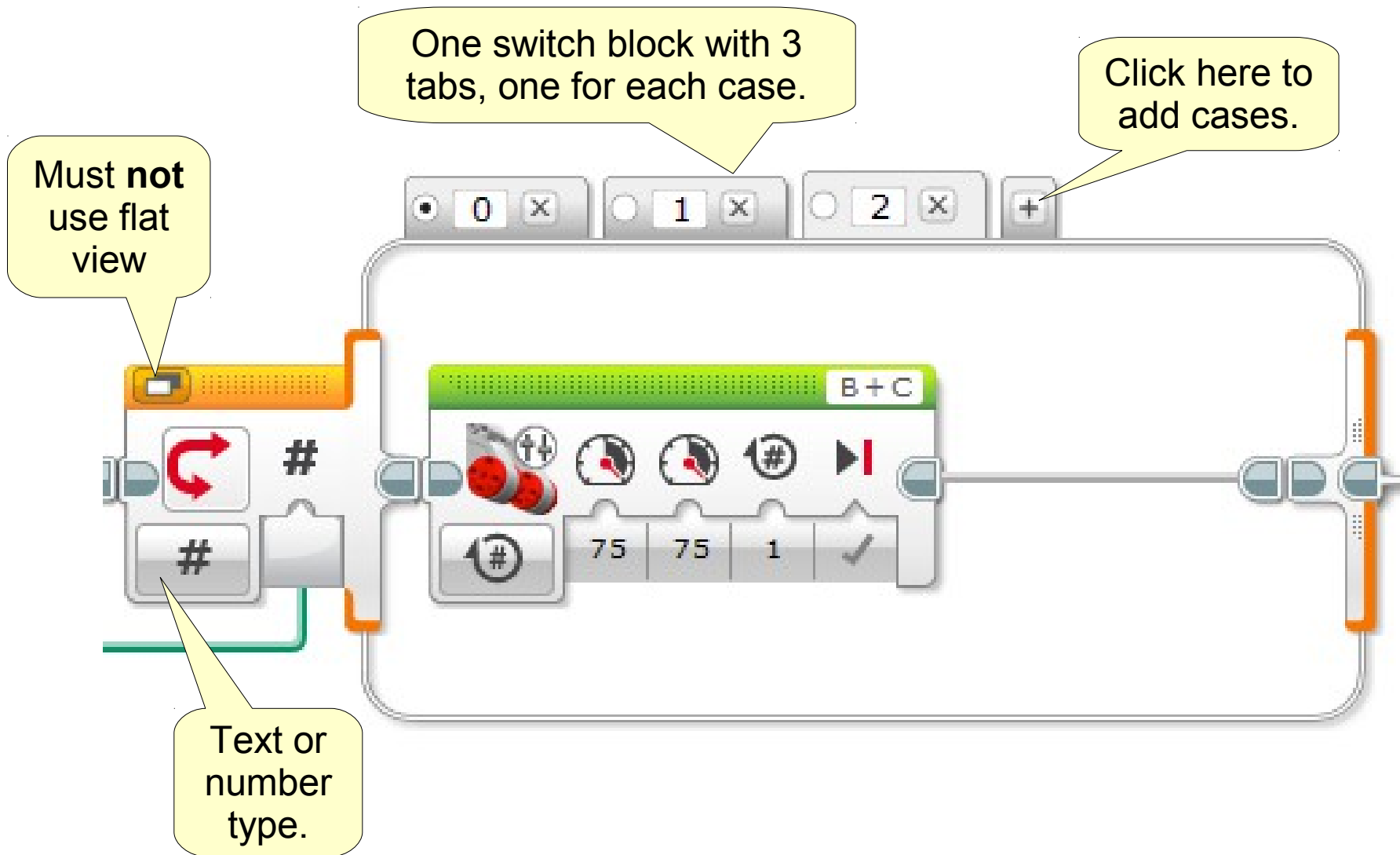
Write text, even graphics, to the NXT LCD panel.

NXT Live updates

From your PC (or MAC), use EV3 to view the values of variables and sensors.



Switch Between Cases





Rotation Sensor Reset Block

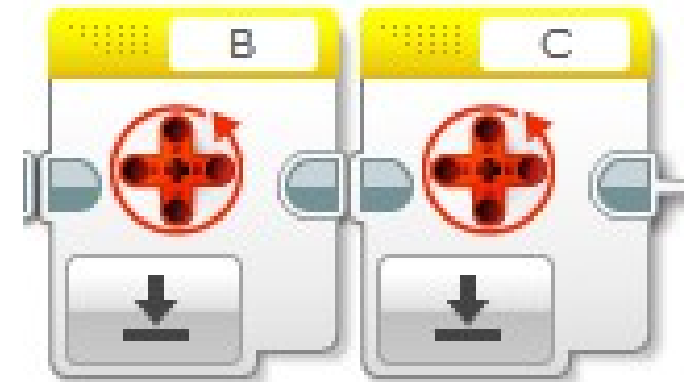
Motor and Move Blocks are servo controls.

Corrects momentum over runs

For several blocks in a series, the correction is based on a total rotation target.

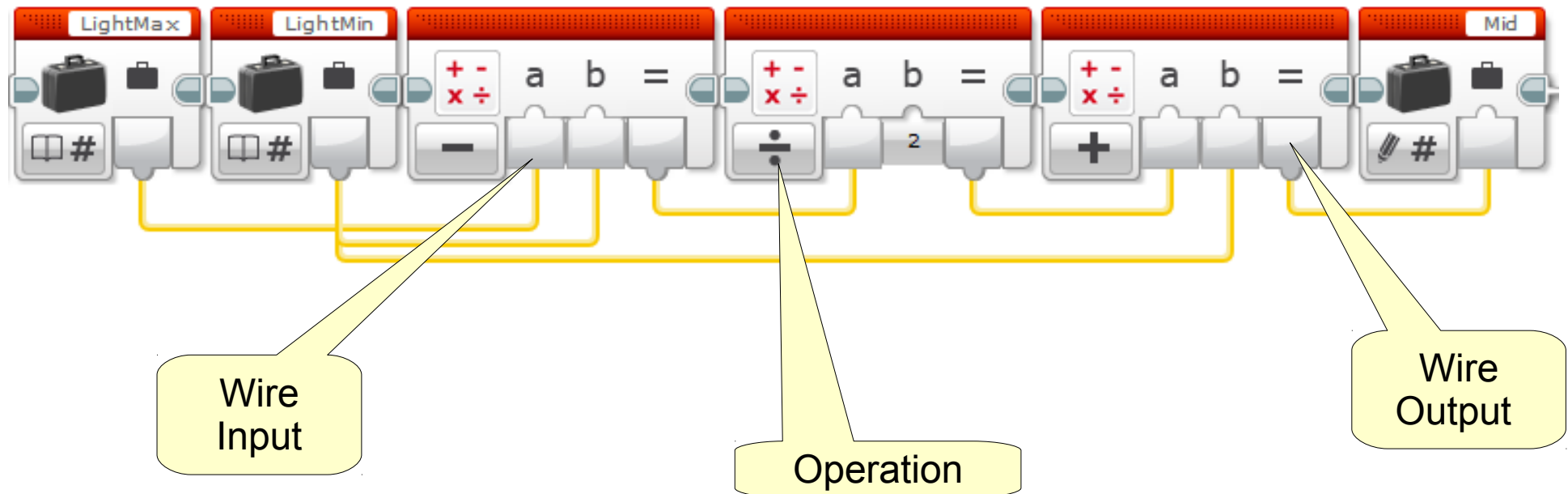
Even applies across programs

Use the Rotation Sensor Block to reset each motor.



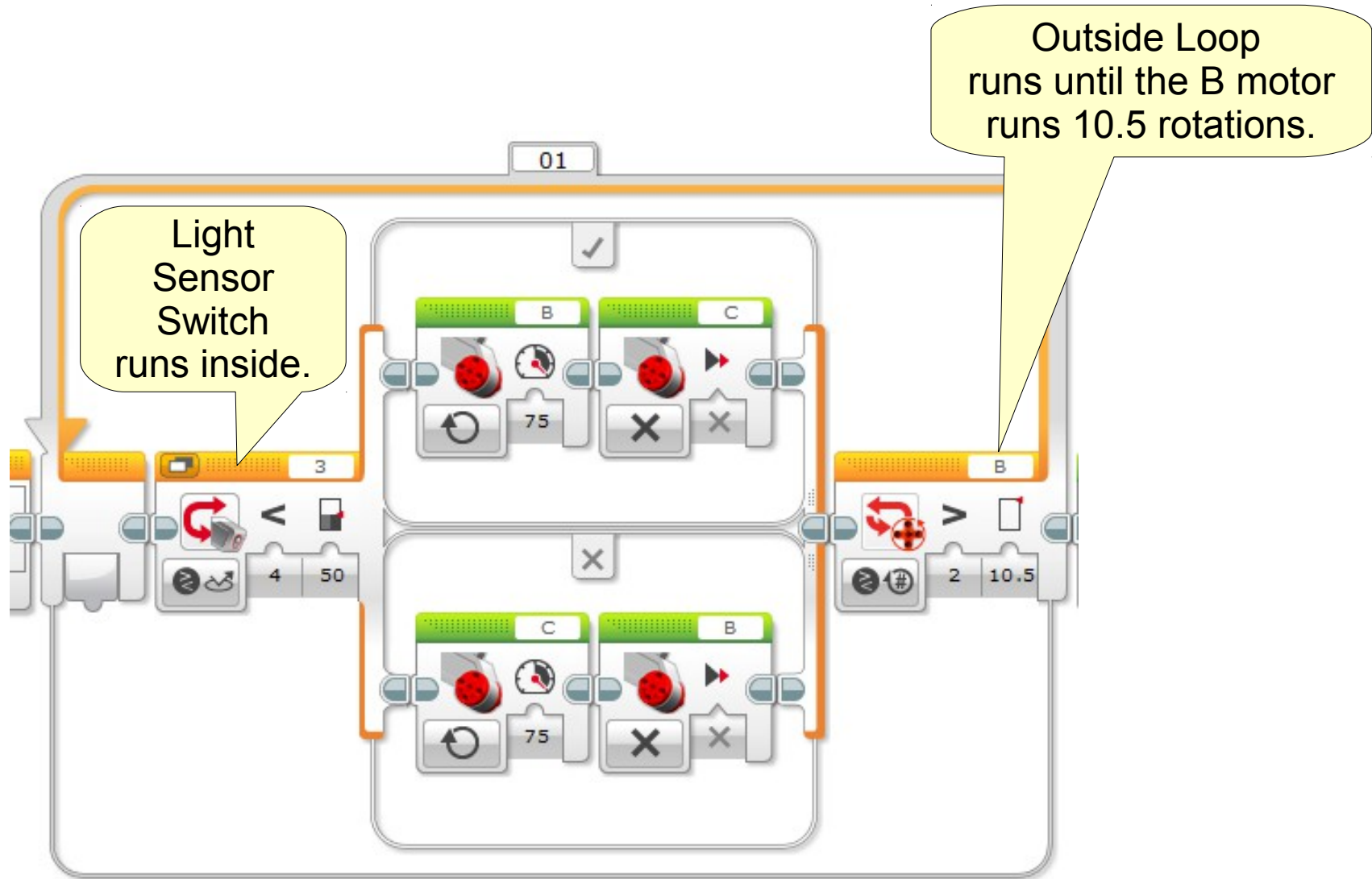


Arithmetic





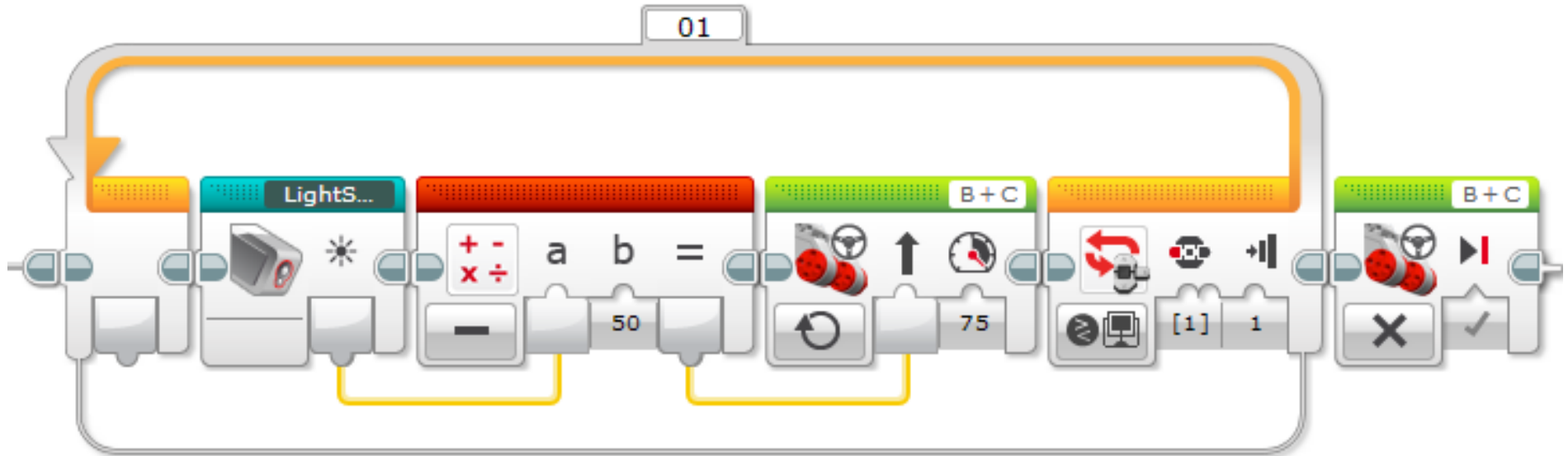
Combining Structures





Comparing Algorithms

Compare that line follower to this one:



Better

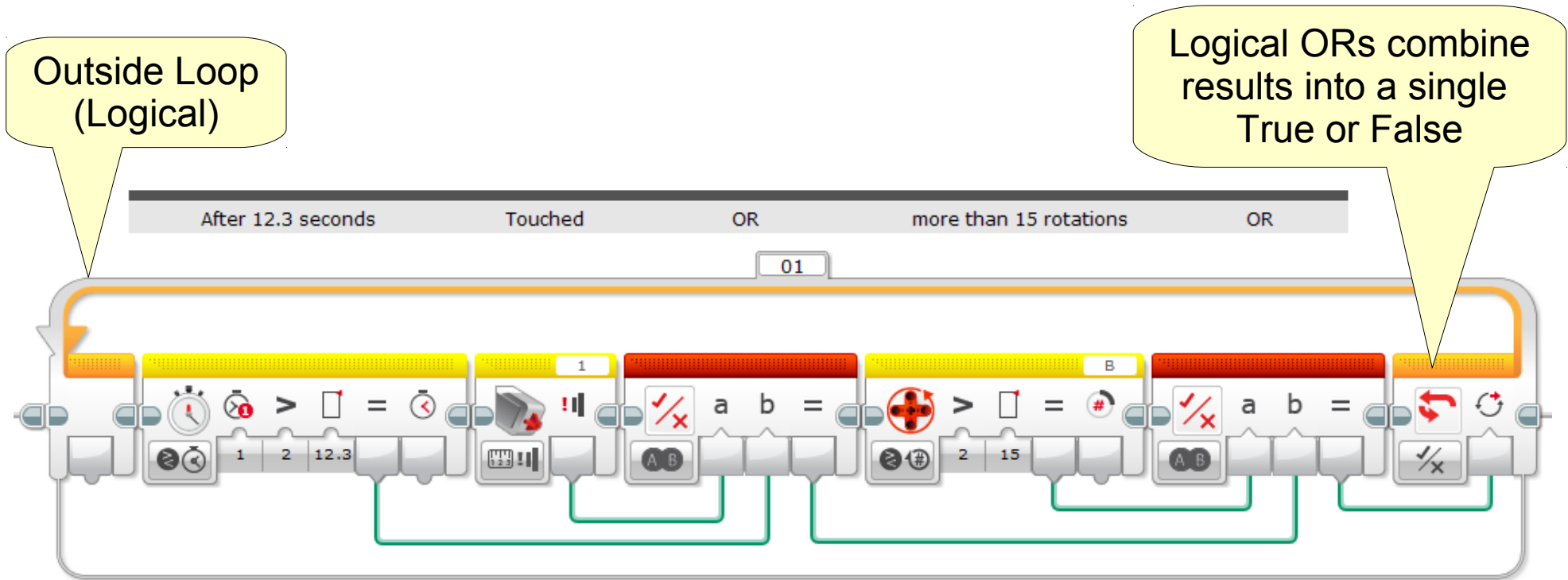
Simpler, proportional steering

Worse

More sensitive to proper calibration



Watching 3 Sensors

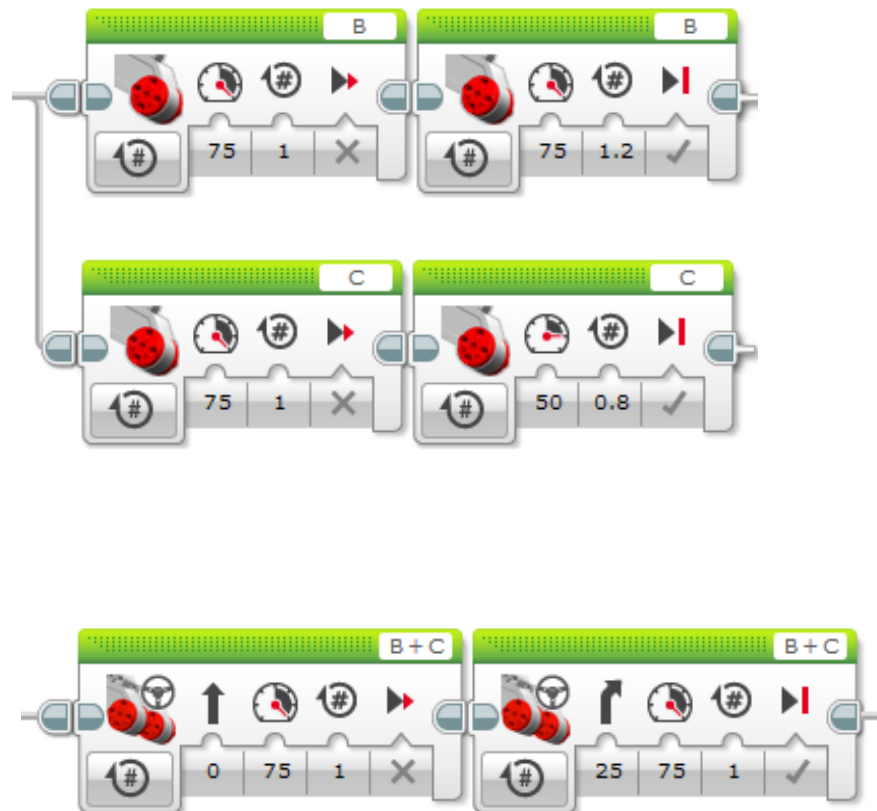


Wait for any 1 of 3 sensors



Optimizing Code

Which is faster?
more reliable?
best?



Use the one that makes sense to you, the programmer.



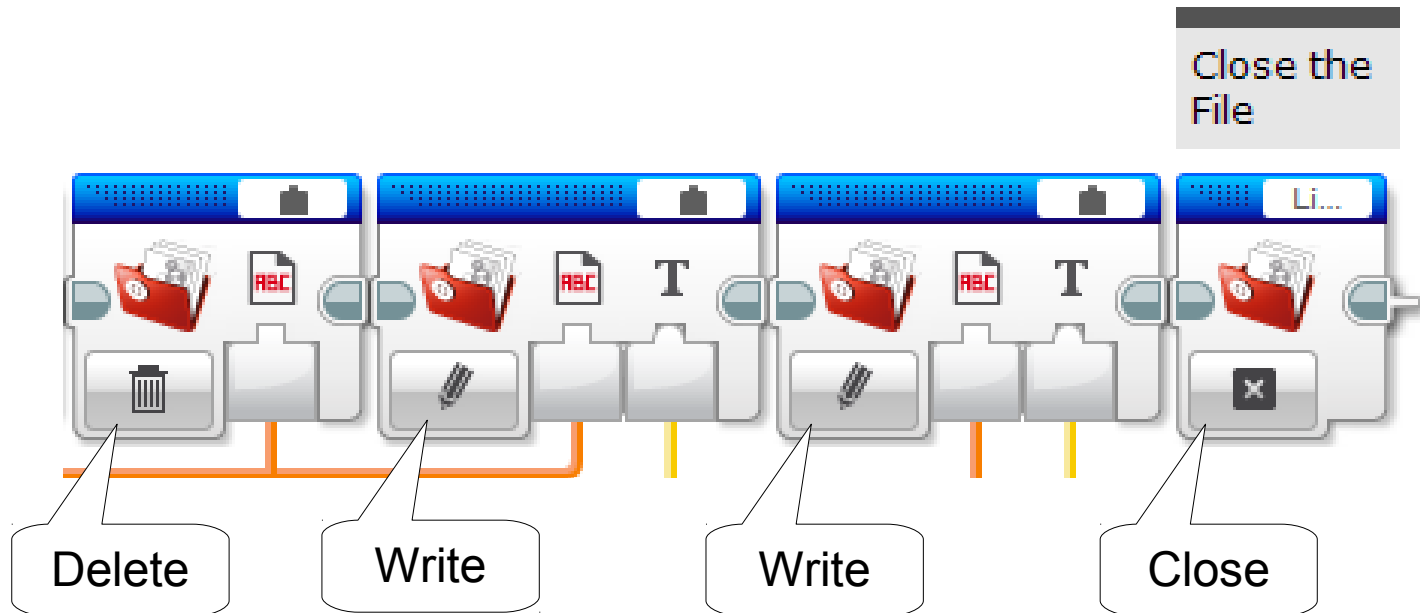
File Operations

Read

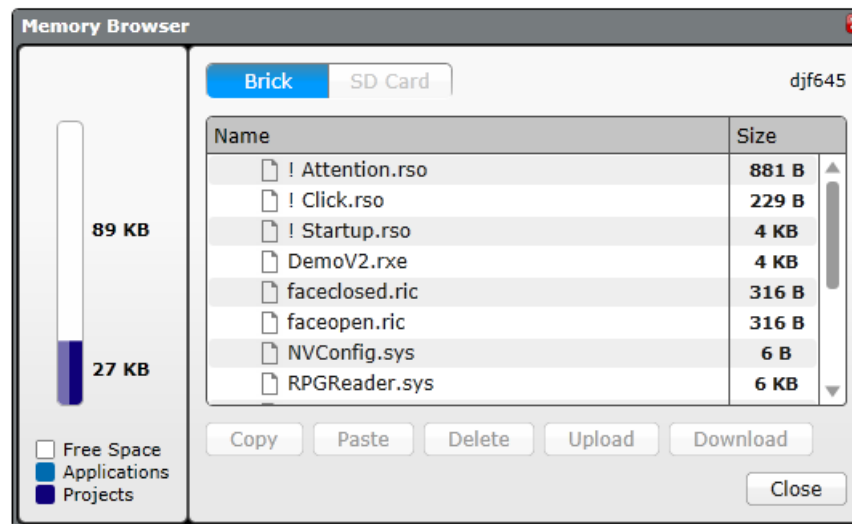
Write

Close

Delete



Upload/Download





Bluetooth Communication Setup

On the NXT, select

Bluetooth -> On/off -> ON

Bluetooth -> Visibility -> Visible

In EV3, select a NXT and

click "Connect"

Enter a passkey like "mysocket"

and click "OK."

On the NXT a screen will appear. Re-enter the same passkey. (Be quick, you've got 30 sec.)

After EV3 shows "connected", on the NXT,

Bluetooth -> Visibility -> Invisible

**Sorry, but EV3
software to the NXT
brick does not
support Bluetooth.**



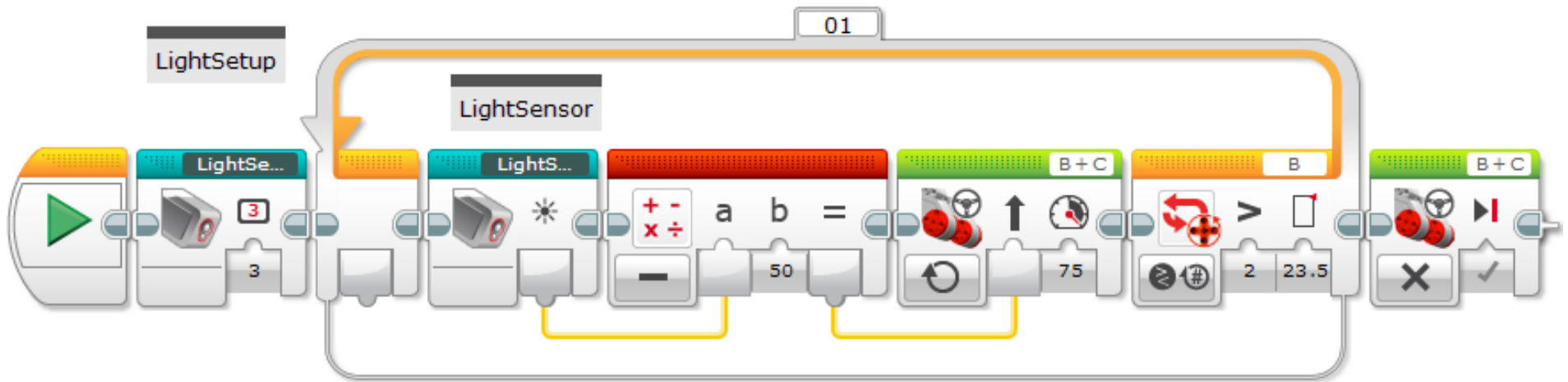
Lab Four

Task:

**Move exactly one lap around an oval.
(Black 2cm line on white paper)**



Lab 4 An Answer



Did you start by copying? Why not?
Is building a robot like taking a test?

