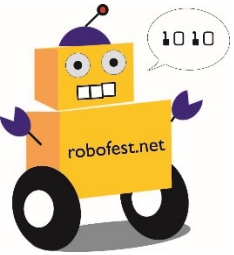


LAWRENCE TECHNOLOGICAL UNIVERSITY ROBOFEST®

BottleSumo EV3 Workshop

Using the EV3 Lab Software

Instructor: Dr. Fred Brauchler

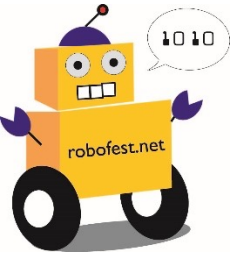
Assistant: Nirmit & Devson

Updates: Acadia Robotics April 2021

Thanks to our Sponsors!

Download EV3 Lab Software

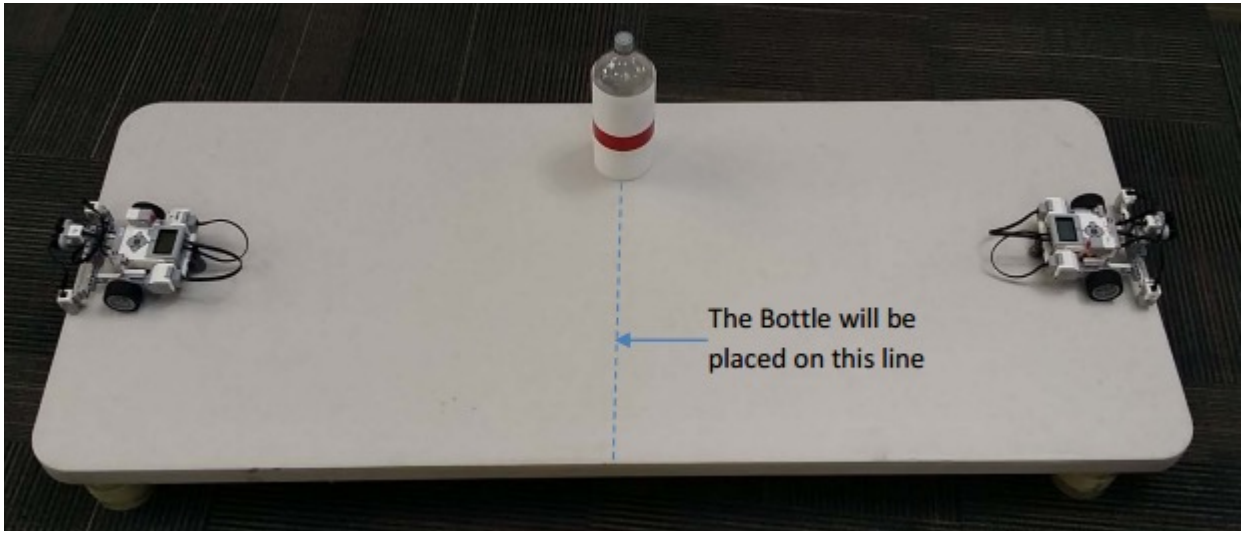
- This software is discontinued. It is still available for download, check your system requirements before download.
- A free EV3 download can be found here: https://education.lego.com/en-us/downloads/retiredproducts/mindstorms-ev3-lab/software
- Also note that EV3 software works with NXT robots and NXT sensors!



Course Overview

- 2016-17 RoboFest BottleSumo Rule Highlights
- EV3 Robot Configuration
- EV3 Software introduction
- Using the EV3 for BottleSumo
  - Sense the table edge
  - Finding the bottle/opponent
  - Push bottle/opponent off
- Physics and Strategy

Rules Overview

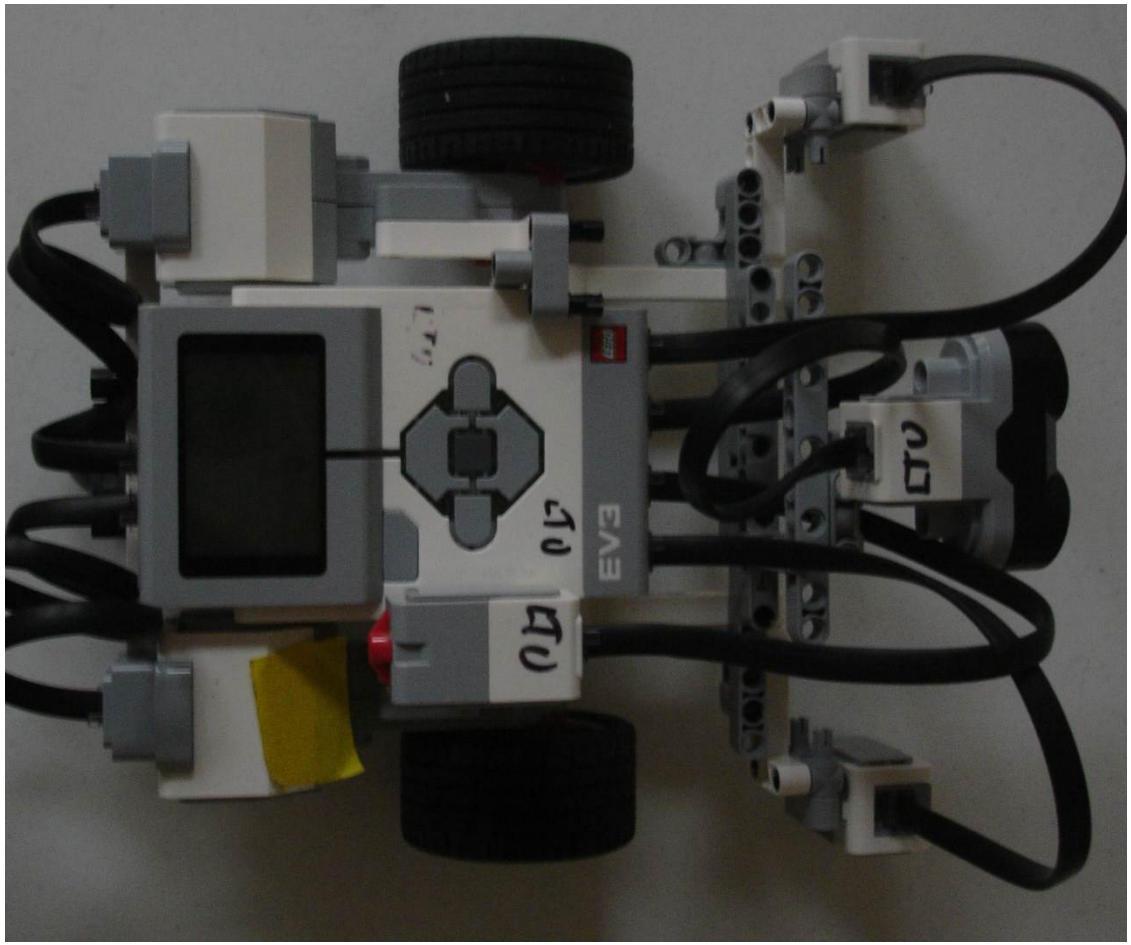


- Objective: Intentionally push the bottle or opponent off **AND** remain on the table for > 3 second after
- How to start the robot – unveiled 30 min before impound
- Starting location, orientation, and exact location of bottle zone unveiled after impounding
- Time trials – push 2 bottles off the table (3 bottle for Sr. teams)
- Please remember to read the rules!

Robot Configuration

Left Motor B

Left Light Sensor 4

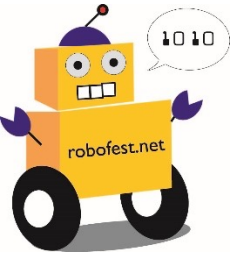


EV3 Brick

Ultrasonic Sensor 3

Right Motor C

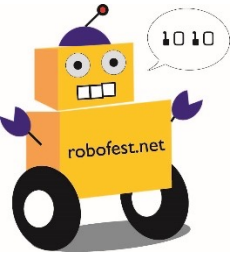
Right Light Sensor 1



Motor and Sensor Connections

- Left Motor connects to port **B**
- Right Motor connects to port **C**

- Left Light sensor connects to port **4**
- Right Light sensor connects to port **1**
- Ultrasonic sensor connects to port **3**

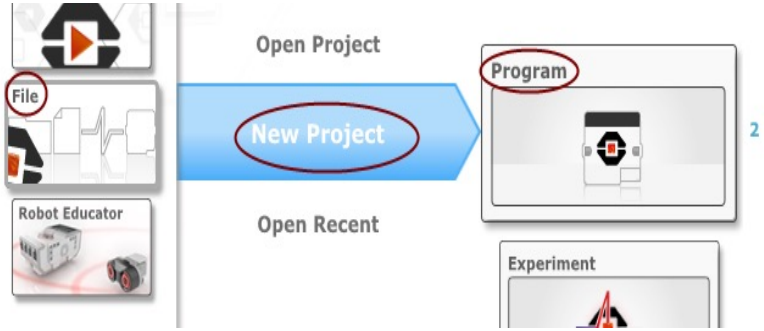


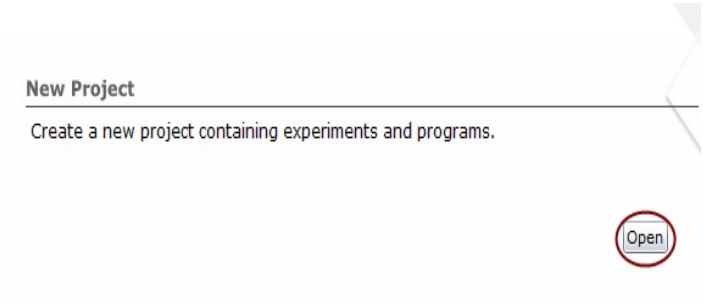
What does your robot need to do to win?

- Not fall off the table
- Find objects: bottle or opponent
- Push objects off the table

EV3 Software

- Click the LEGO MINDSTORMS icon on
- Click File > New Project > Program > Open

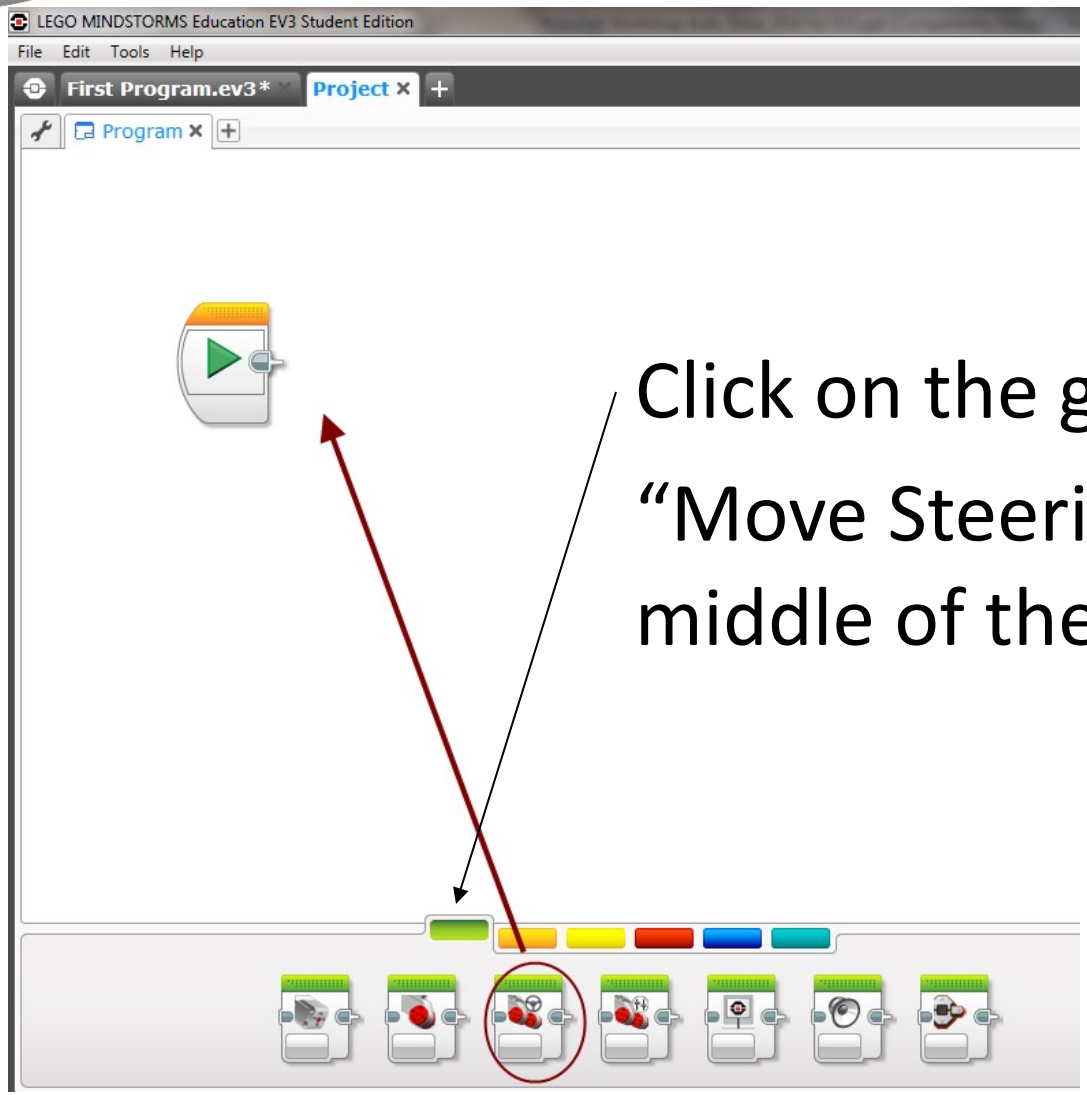




For Home Edition optional

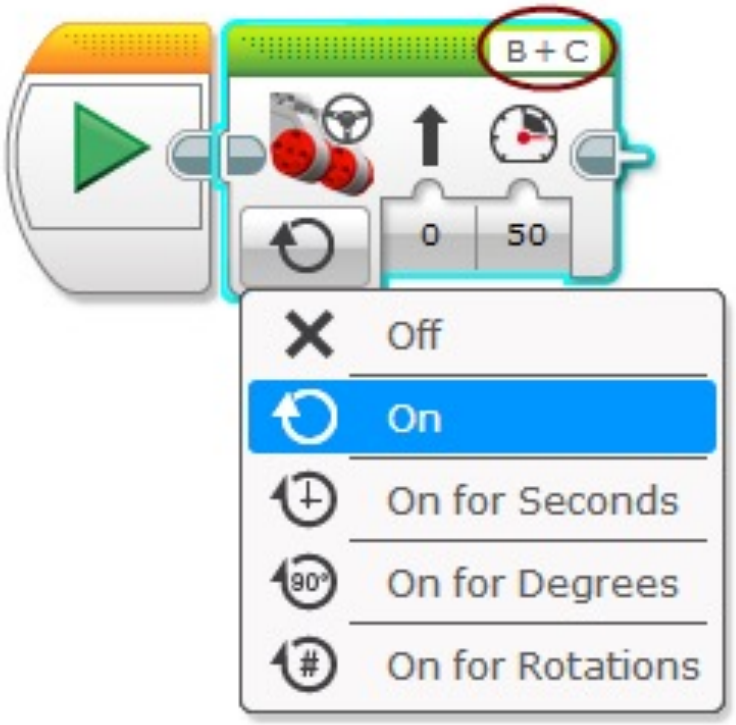
- Select the “+”

Creating a Program



Click on the green tab, drag a "Move Steering" block to the middle of the screen

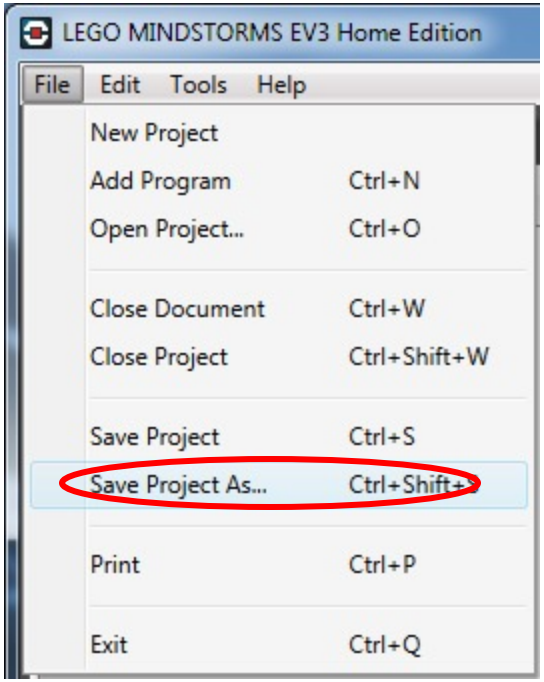
Move “Forever”



Set the motors to “On”

Make sure you’re using the right motors!

Save Your Program



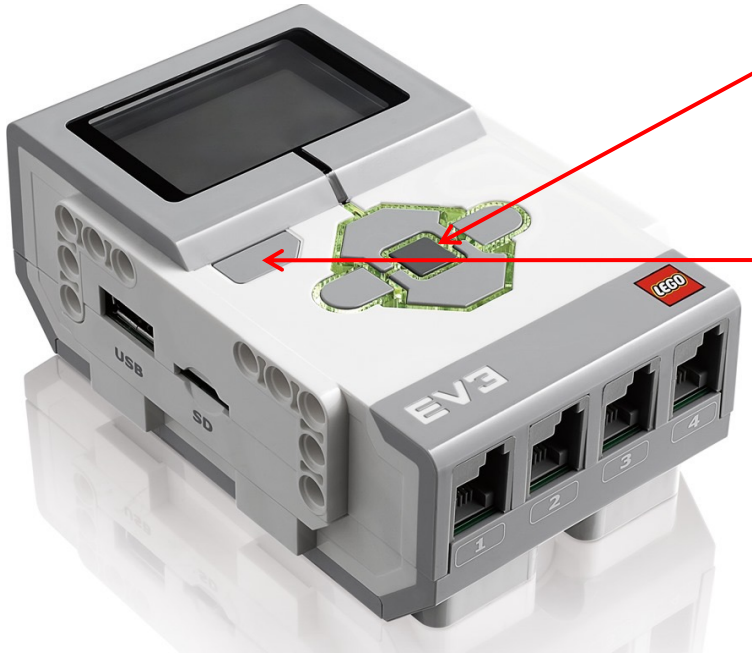
1. Click “File” then “Save As”
2. Choose a memorable name
3. Click Save

Downloading a Program

- Verify the robot is turned on
- Connect the USB cable

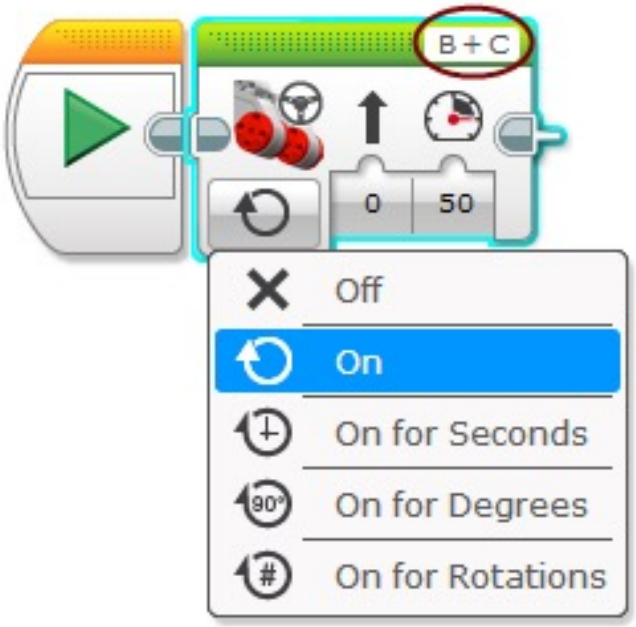
- Click refresh
- Click the box under USB
- Click the blue arrow to download

Running a Program



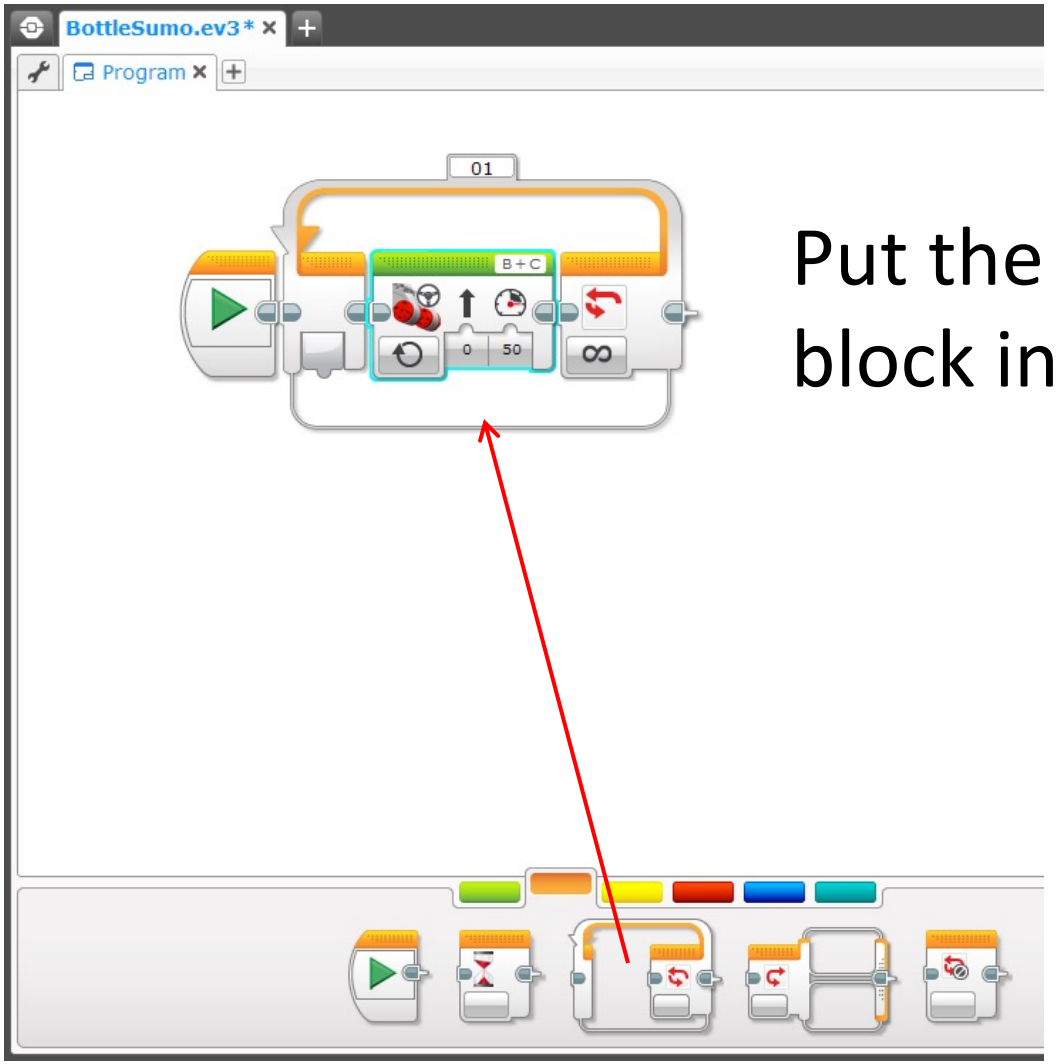
- Press the middle button to run your program
- Press the top left button to stop your program
- Use the surrounding buttons to switch menus and view programs

Motors “On”



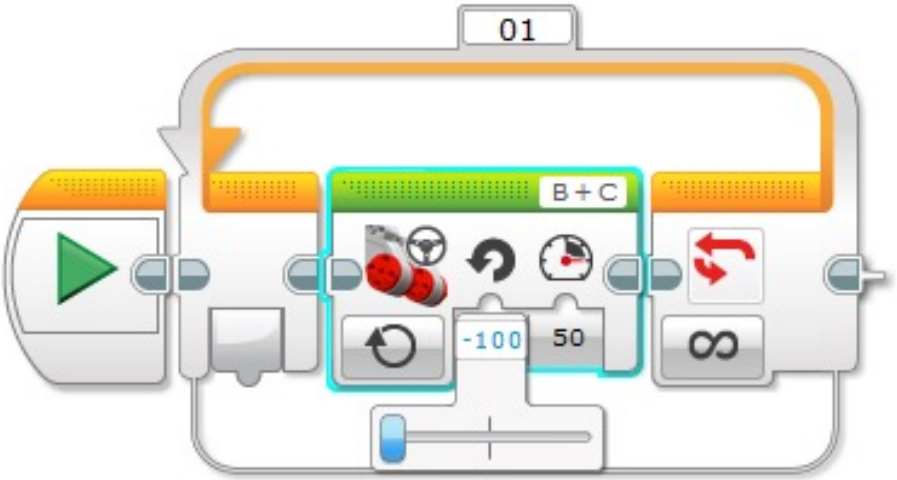
- What happened when run?
  – the program finished and turned motors off before they moved.
- Did the motor turn on?
  – electrically yes, but maybe not long enough to see them move

Loops



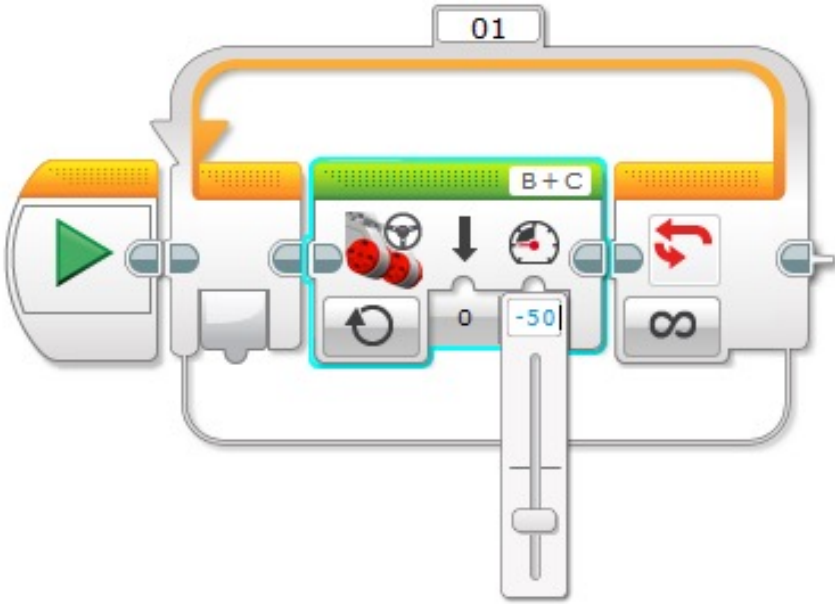
Put the **Move Steering** block in an infinite loop.

Spin



- We can make the robot spin by changing the “Steering”
- ±100 for a point turn (one motor forward, other backwards)

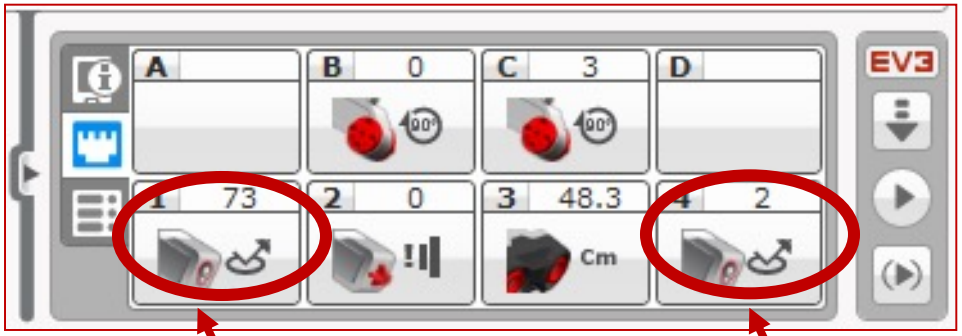
Moving Backwards



- We can still use the Move block.
- Change the speed to a negative number

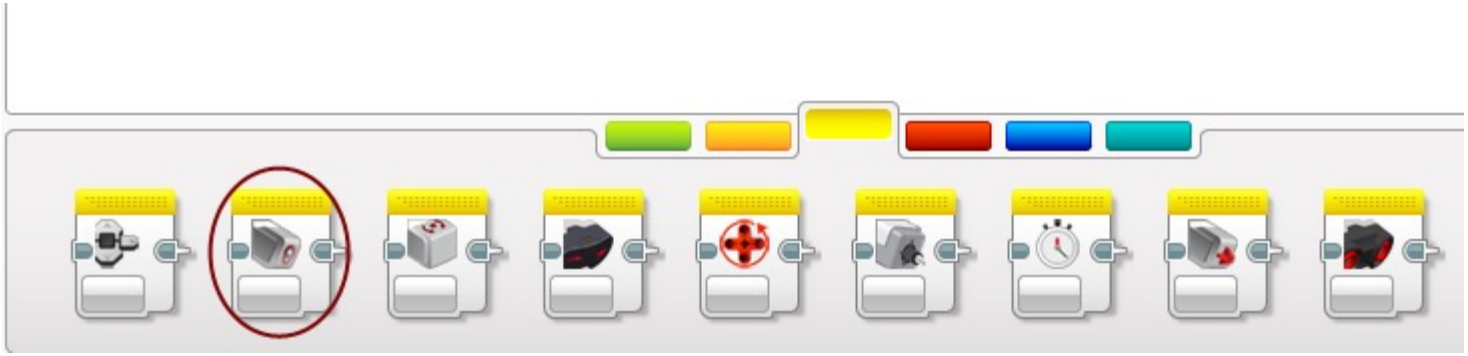
Light (Color) Sensor Readings

Put one **Light Sensor** on the table and the other **Light Sensor** off the edge of the table

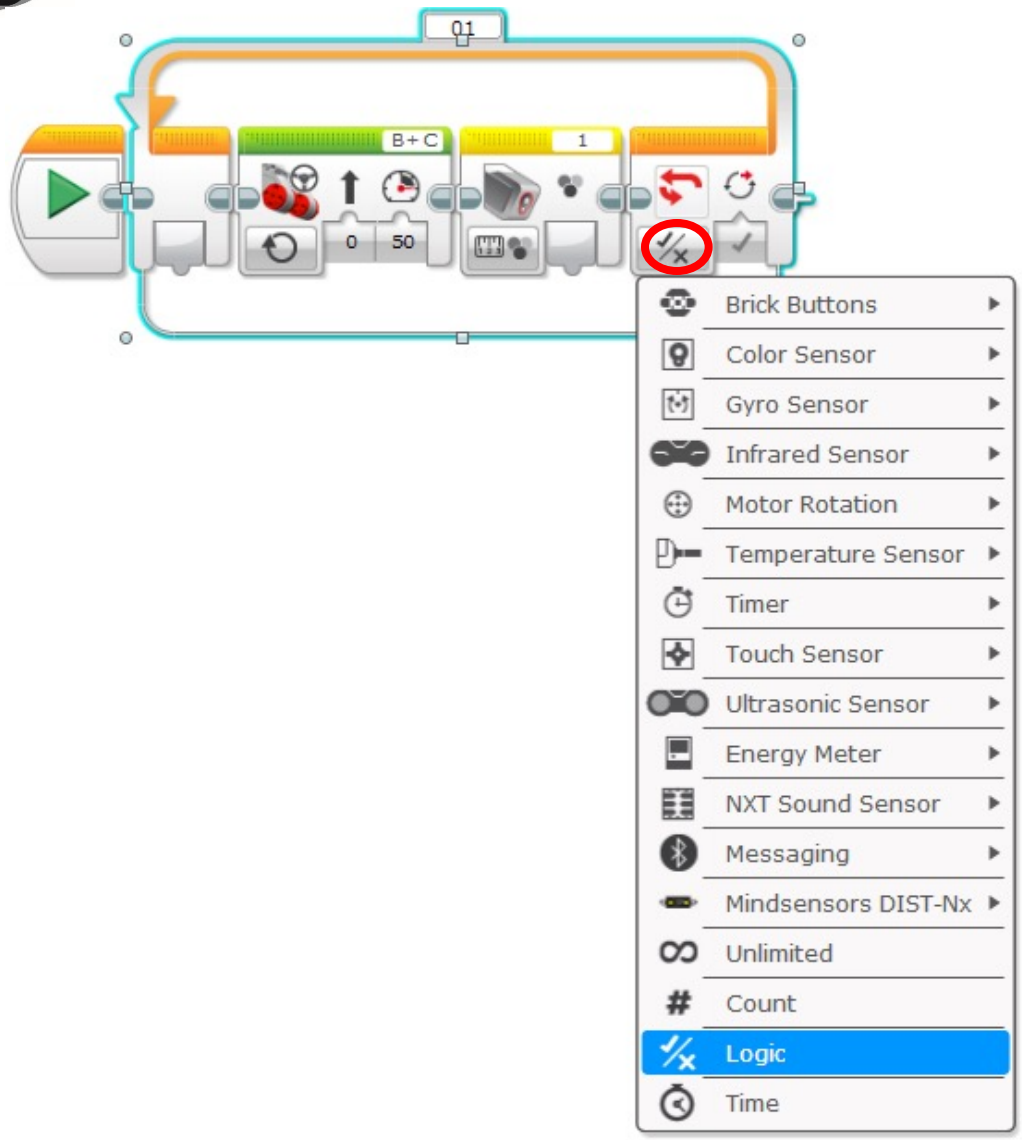


Color Sensors

Click on the yellow tab and drag a color **Color Sensor** block up

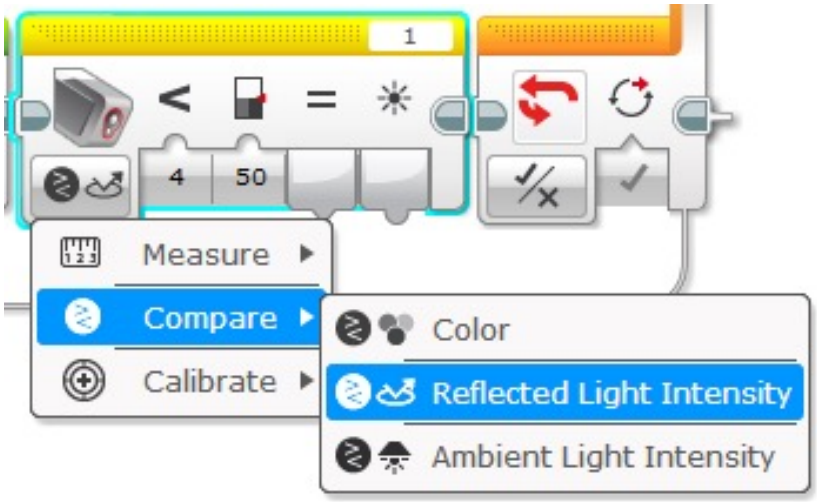


Finding the Edge



- Change the loop to Logic
- Now it will "loop until true"

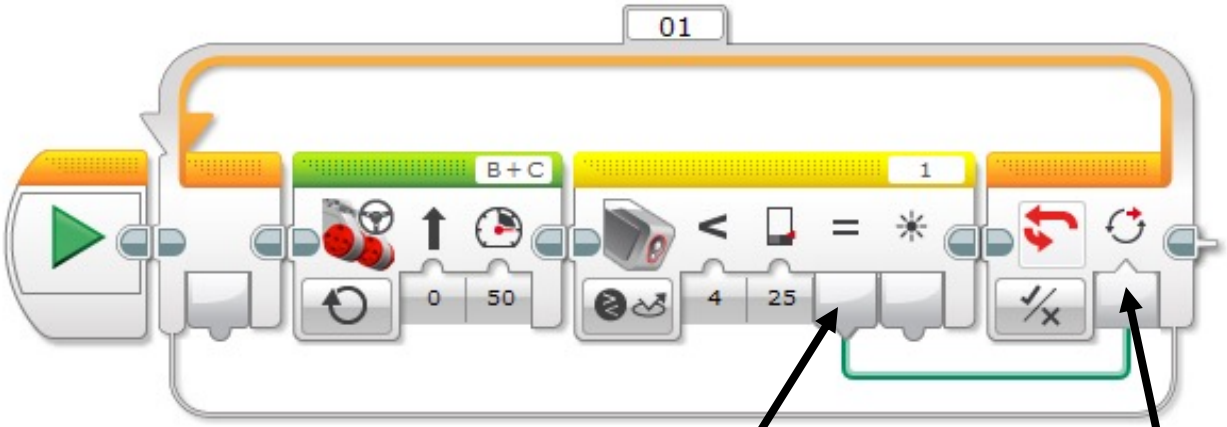
Finding the Edge



- Change the light sensor to **Compare > Reflected Light Intensity**

- Light sensor threshold: **Average** of the "on the table" and "off the table" values. Put your value here.

Finding the Edge

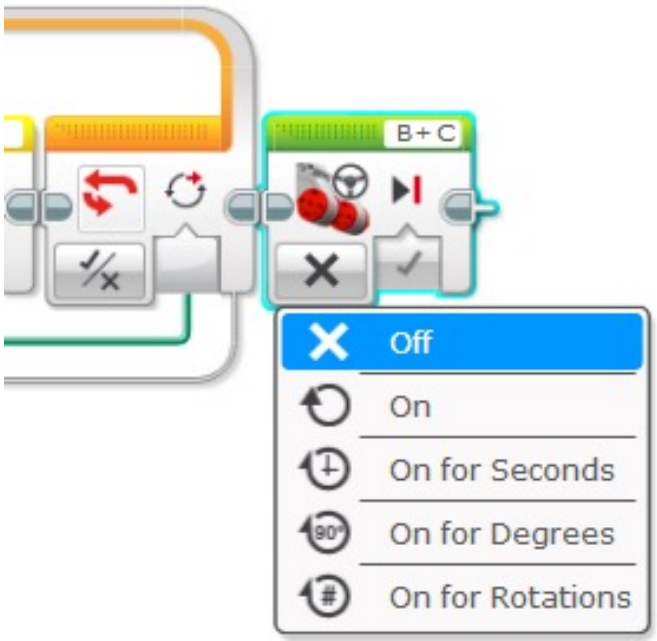


- Use your mouse to draw a green "logic" data wire
- A yellow data wire is for a "number"

Start Here and Click the Mouse

Drag to Here and Release

Stopping at the Edge

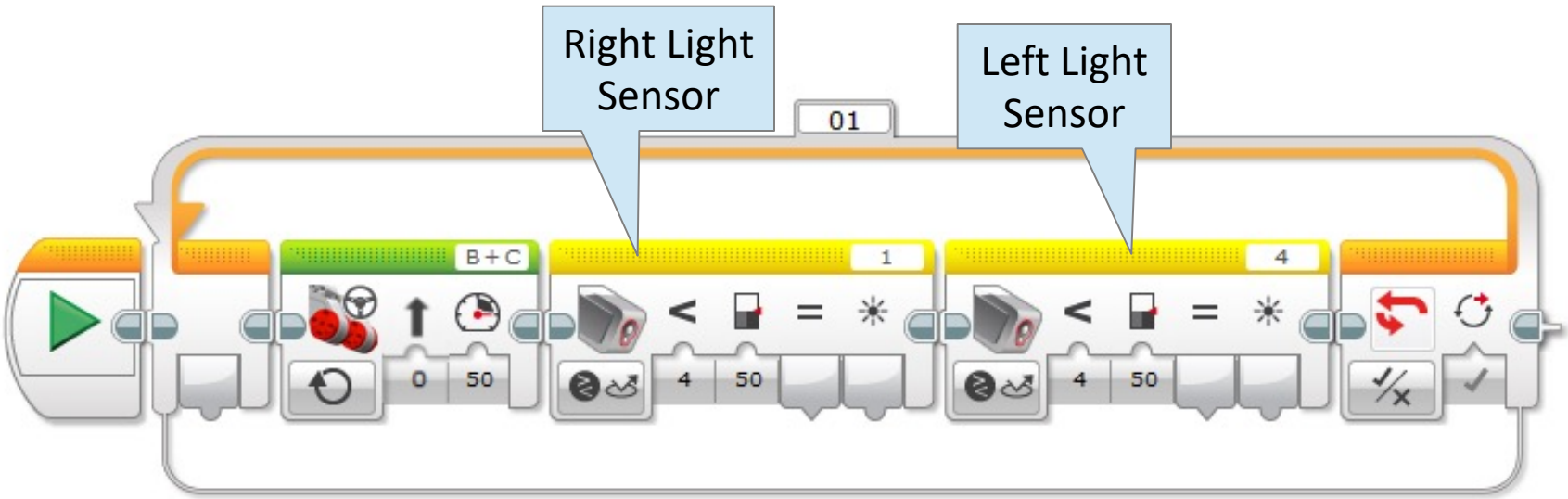


Make sure to turn off the motors once your robot finds the edge!

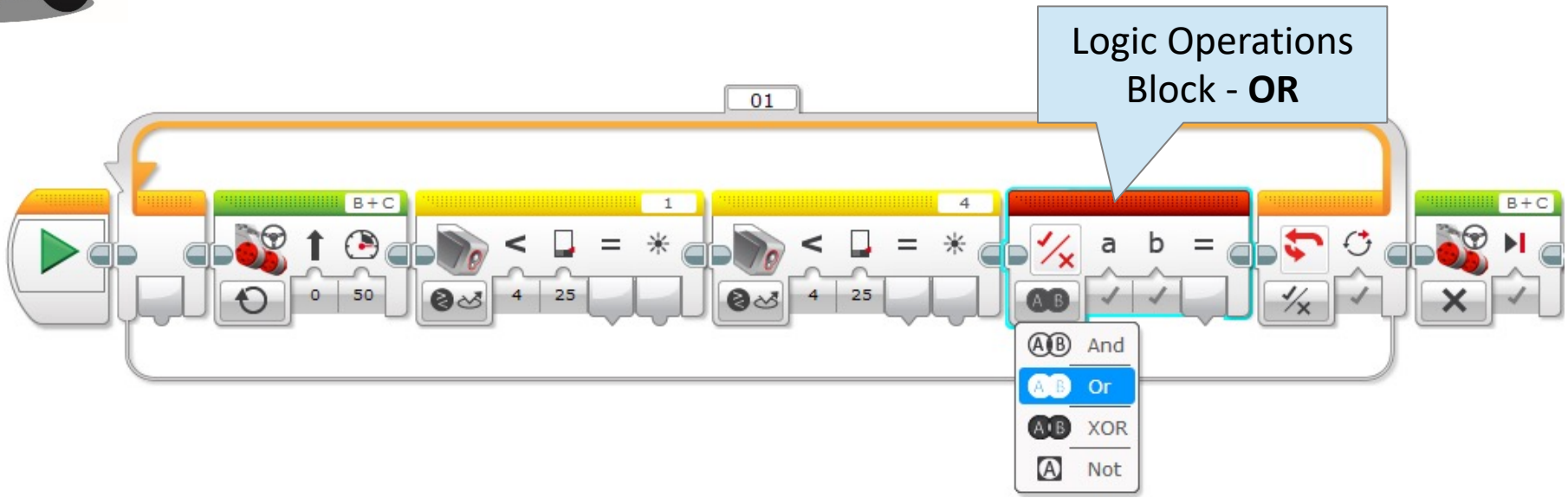
Edge Problems

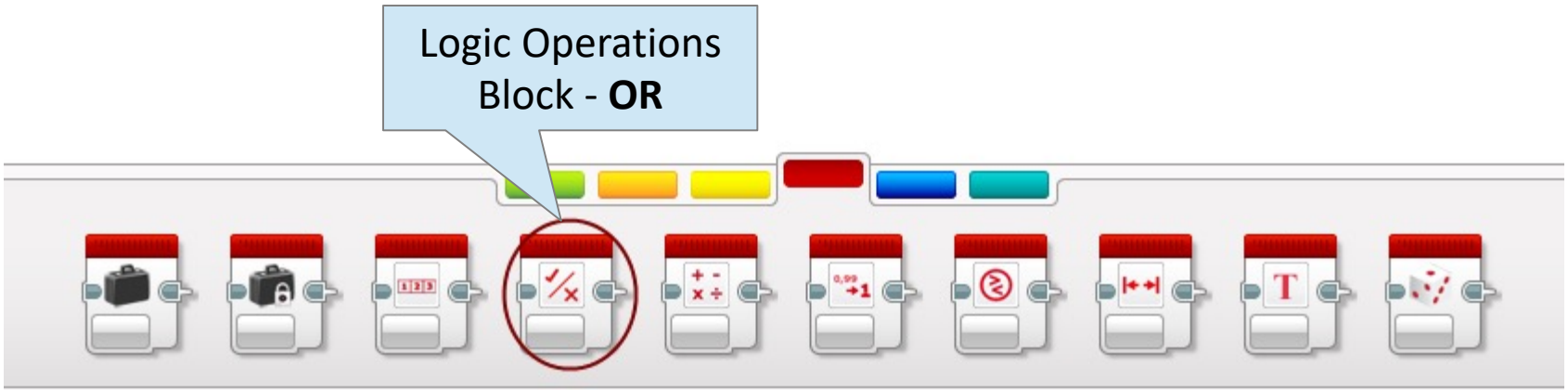
- If your robot stops too late or not at all:
  - Check that the move block is “On” not “On for Seconds/degrees/rotations”
  - Check color sensor is shining a red light
  - Check color sensor values on/off the table, the threshold average, and the “less than” inequality
  - Check light sensor port number (1 for left sensor)
- We only are looking at one light sensor – what if the other one goes off the table first?

Finding Edge with Both Sensors



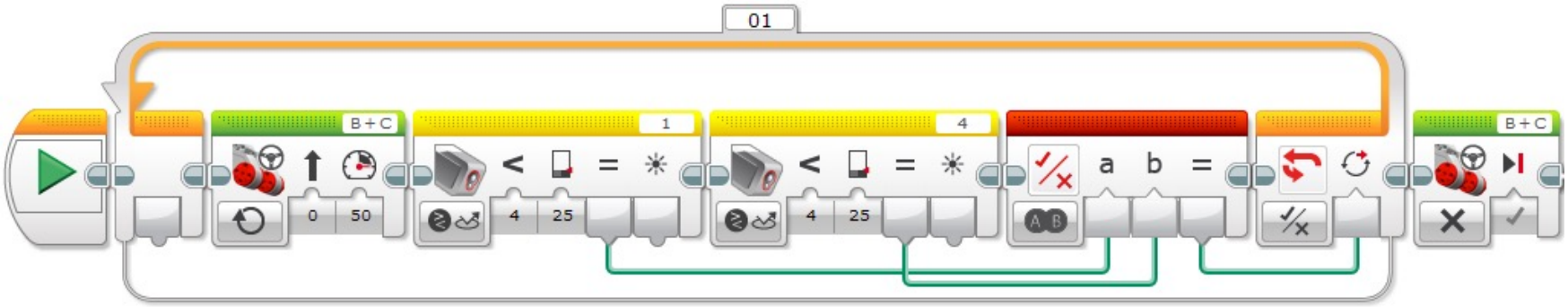
Finding Edge with Both Sensors



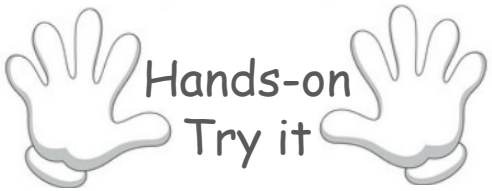


Finding Edge with Both Sensors

Connect to the right wire hub.
→ The wire should be green for logic.

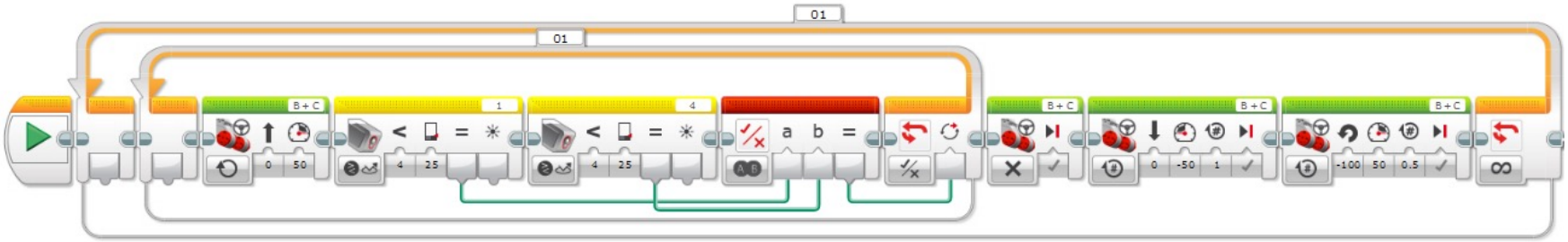


find-edge.ev3p

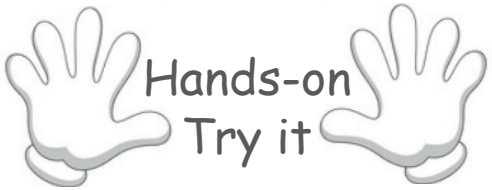


Finding and Avoiding Table Edges

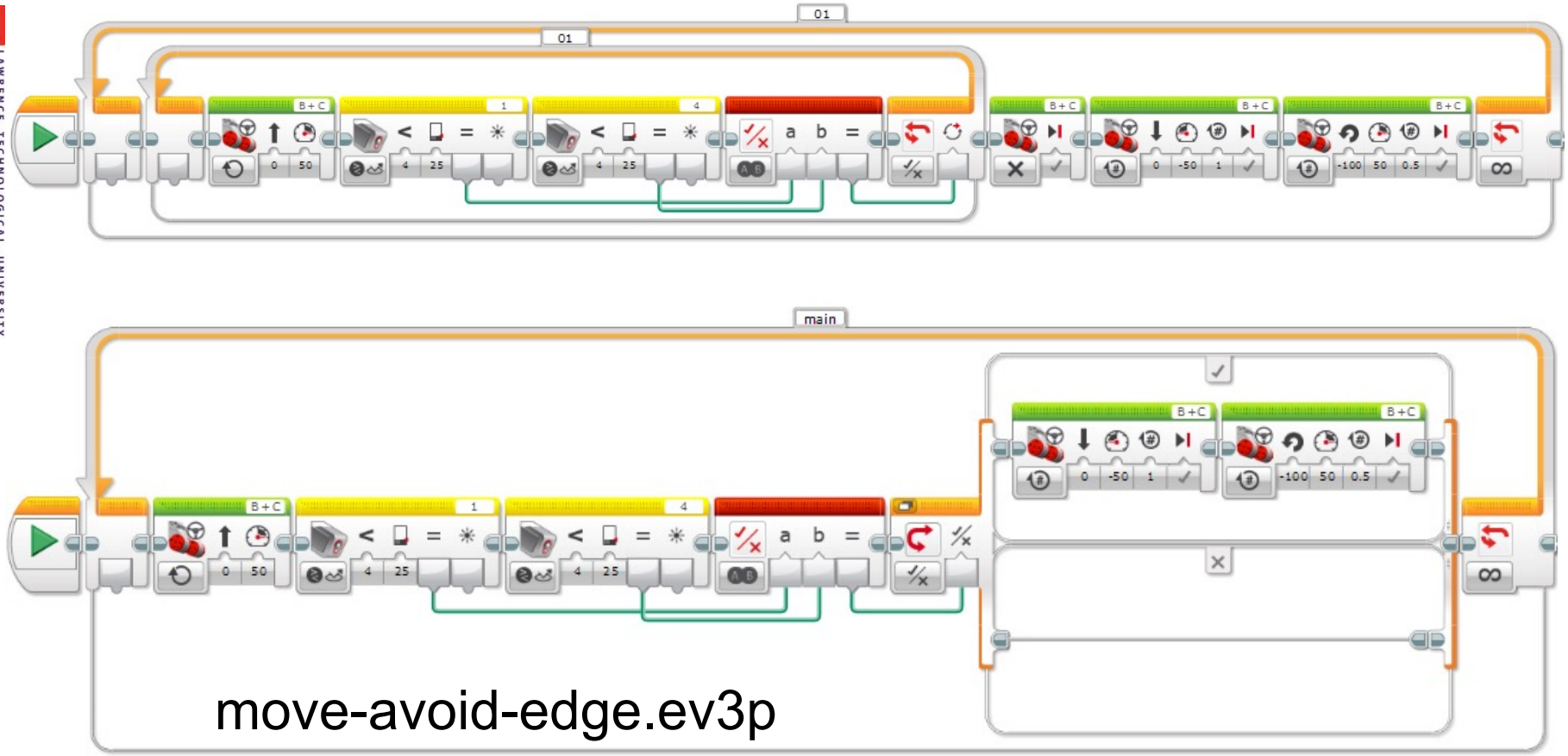
What should you do when you get to an edge.
→ Back up and turn.



avoid-table-edge.ev3p



Alternate Program



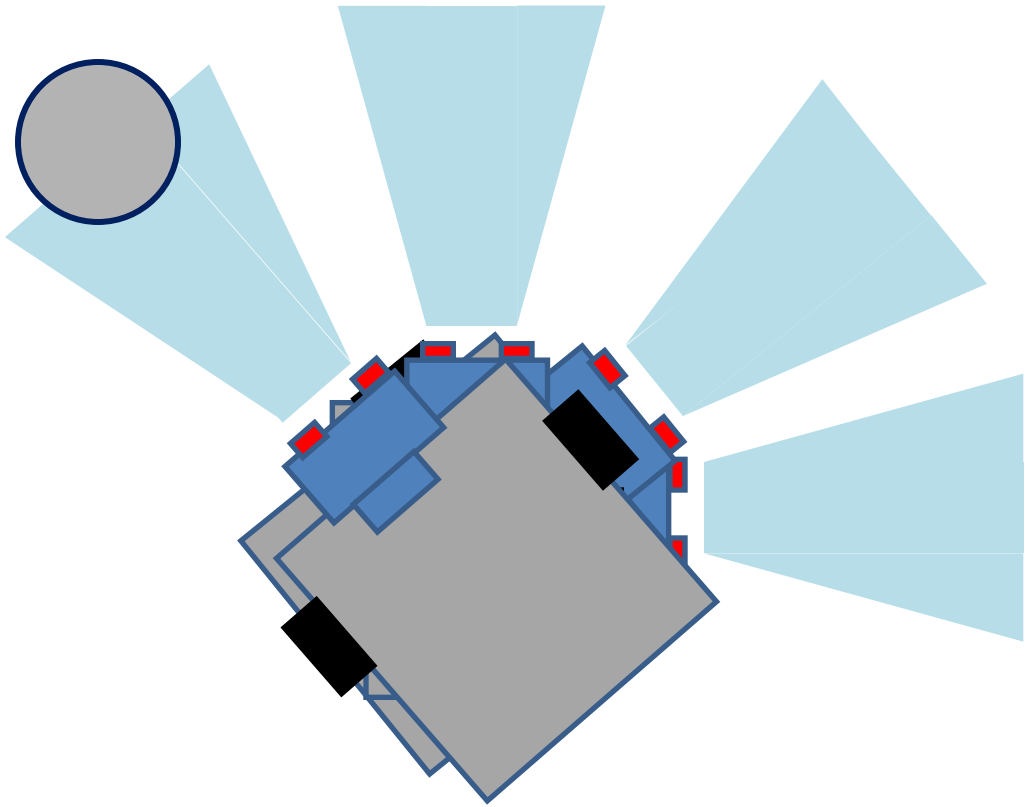
This program uses a switch block instead of a loop to avoid tying up the program.

Ultrasonic Sensor

- Measures the distance to the closest object
- Uses sonar waves like a bat
- Can be used to see if an object is within a certain range of the robot

Finding the Bottle/Opponent

- Point Turn (Spin) robot
- Stop when object found

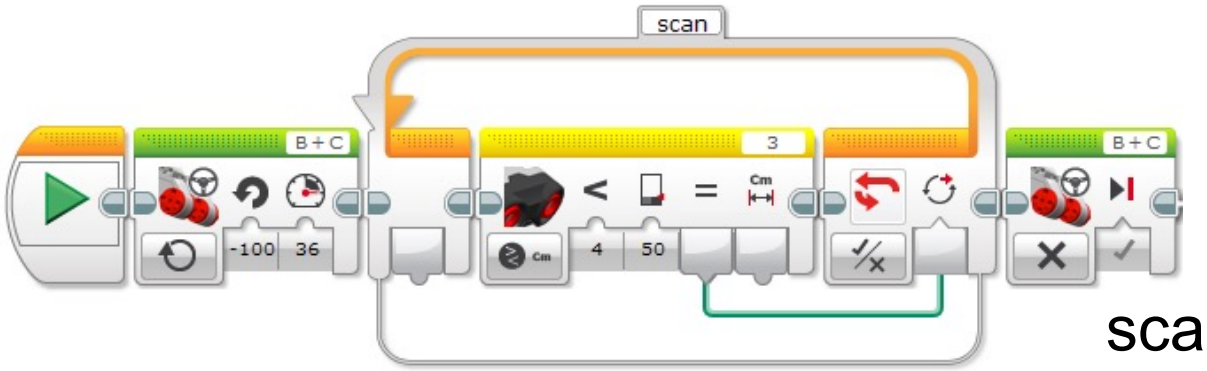


Finding the Bottle/Opponent

Scan_wait.ev3p

These two programs do the same thing:

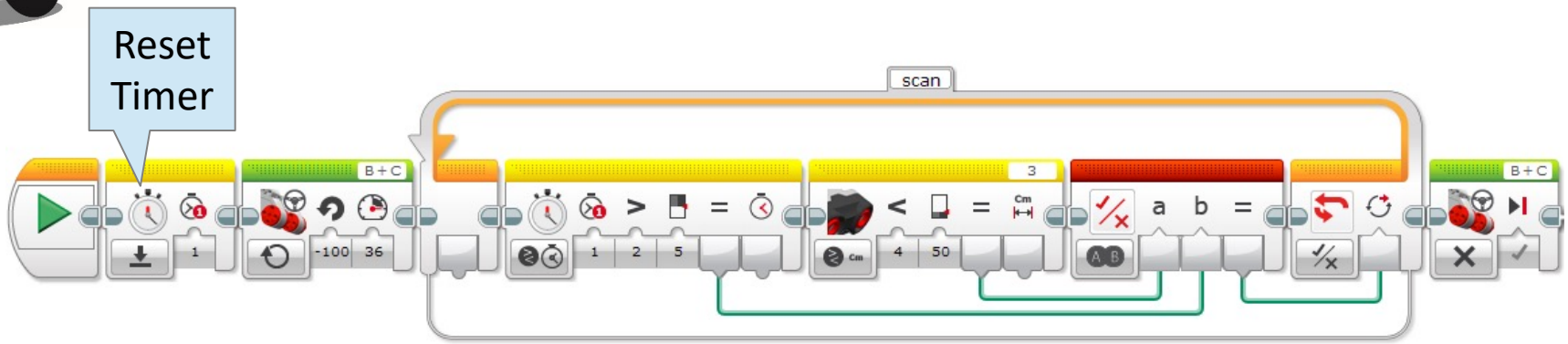
- Point turn
- Wait until sensing an object < 50cm
- Then stops motors



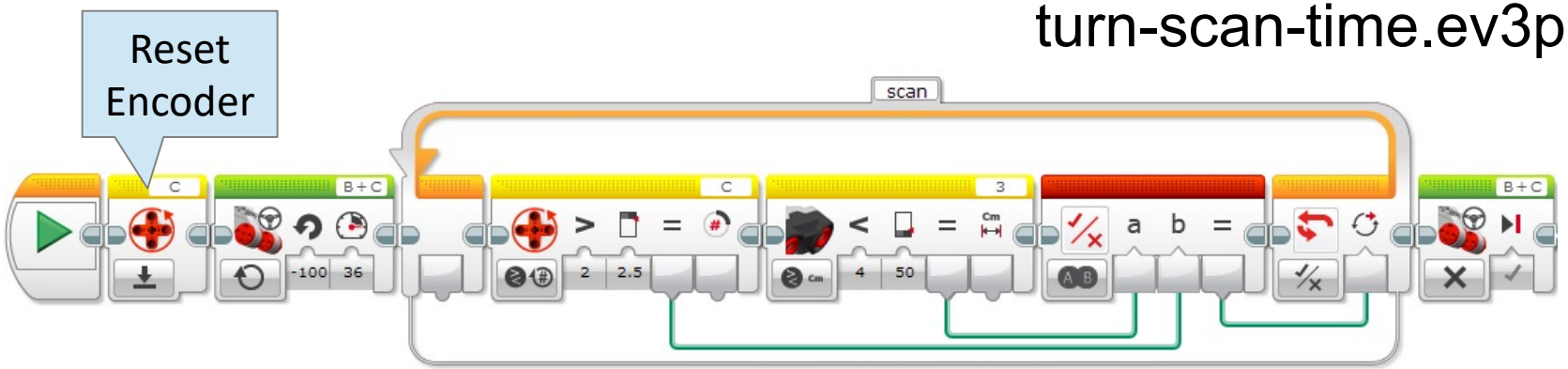
scan_loop.ev3p

Lower program allows more options – example:
What if nothing within 50cm? – see next page

Finding the Bottle/Opponent

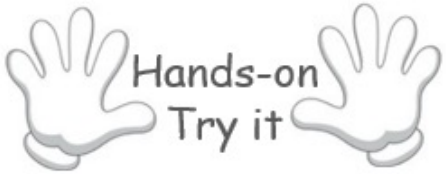


turn-scan-time.ev3p



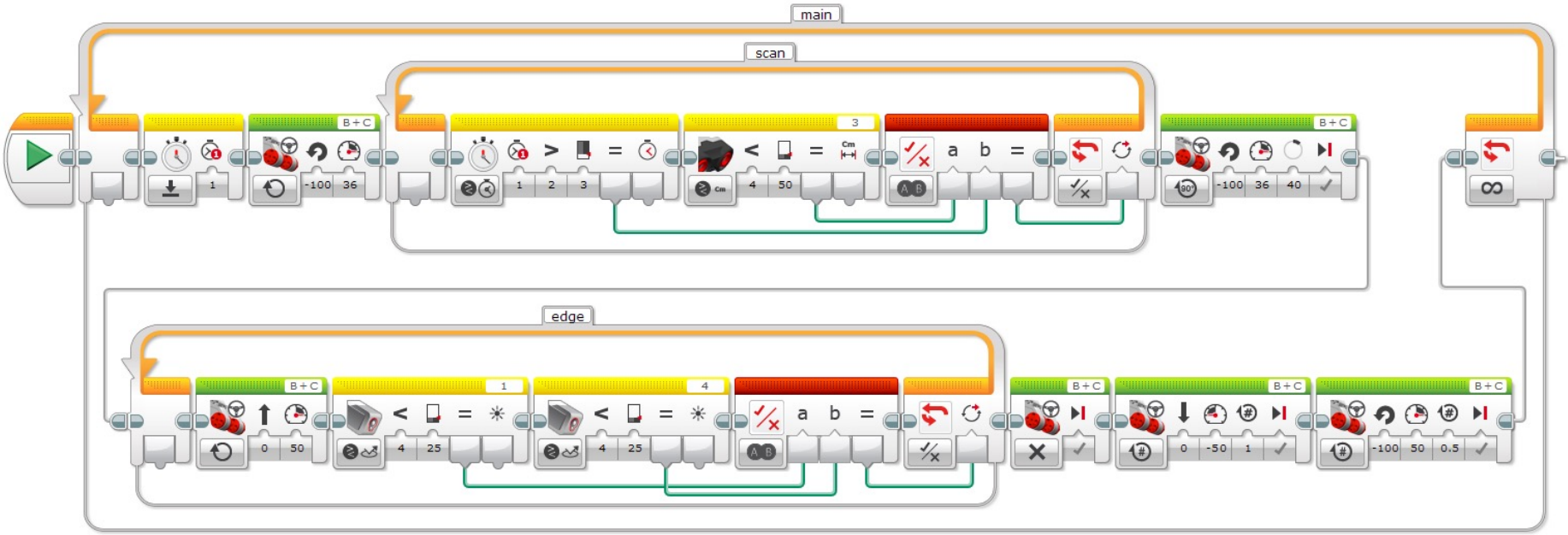
turn-scan-encoder.ev3p

The added condition will limit how long (time) or how far (rotations) to turn and search

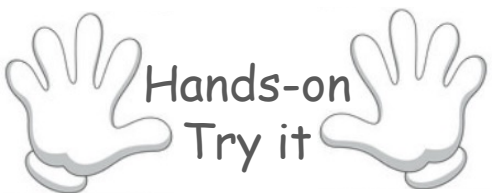


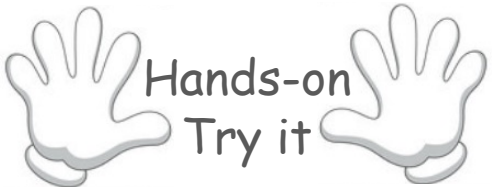
Simple Sumo Program

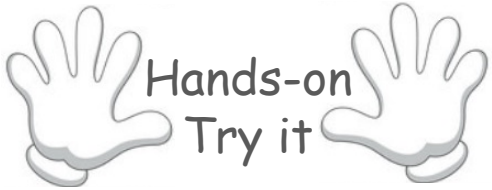
Putting it together:



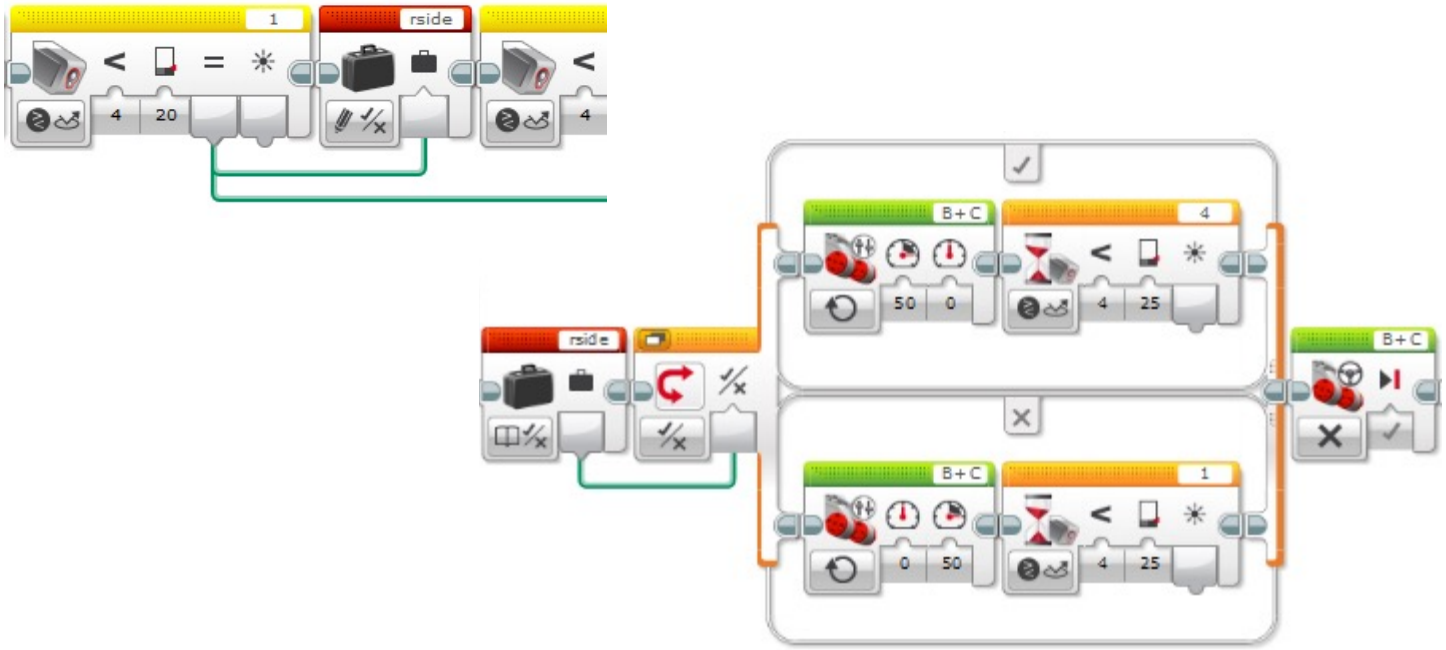
basic-sumo.ev3p







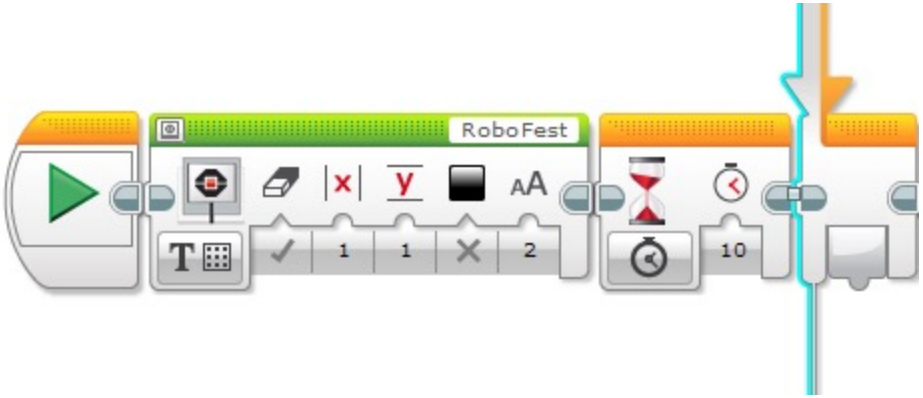
Push it all the way off the edge



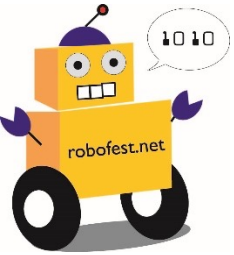
- Set variable “rside” when right light sensor is off edge
- Value of “rside” determines which way to move
- Push bottle/opponent all the way off table
- See program “better-sumo.ev3p”

Unknown Factor examples

- Display “RoboFest” while waiting 10 seconds to move after starting program



- Wait for a black card to be removed from a sensor



Physic of Sumo

Your robot will be more effective at pushing other robots off the table if you consider

- Mass
- Velocity
- Force
- Power

Mass

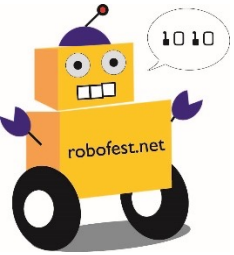
- Quantity of matter [kilograms]
- m=mass, ν=velocity, p=(linear) momentum

  $p=m*\nu$
- F=force, a=acceleration

  $F=m*a$

  •With a larger the mass, more Force is required for same acceleration

  •Where to add mass? Consider the center of gravity (should be low and inside wheel base)

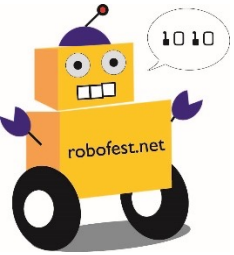


Velocity

- Velocity = Speed and Direction [meter/second]
- m=mass, ν=velocity, p=(linear) momentum

  $p=m*\nu$
- If we increase the velocity, we increase the momentum
  - The larger the momentum, the greater the impact force will be applied to the opponent
  - How do we increase the velocity of the robot?

    Higher motor speed setting, larger wheels, gear-up



Force

- Pushing (Pulling) or Twisting
- Linear Force [Newton] or Torque [N-m]
- T=torque, r=wheel radius
- Force=Torque/radius
- How do we increase the Force applied by the wheels?
  - Increase the torque (lower gear ratio)
  - Decrease the radius (smaller wheels)

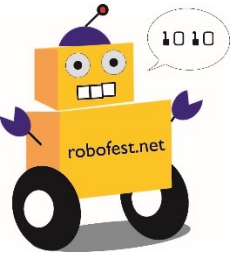
Power

- Power is Force *Distance / time [W=N*m/s]
- P=Power, F =Force, T=Torque, ν=velocity, ω=angular velocity

  $P=F*\nu$

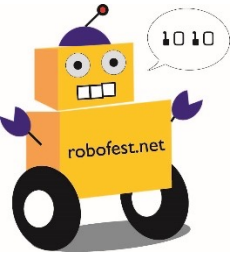
  $P=T*\omega$

- How do we increase the Force and/or velocity?
  - Increase the power (Motor power)
  - Make sure batteries are fully charged!



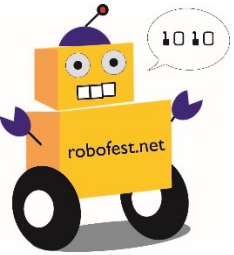
Build a better Robot (later)

- Sturdy construction
  - At least 2 attachment points for each part so robot stays together
  - Compact design
  - Triple pegs on multiple beams
- Wheel base
  - Wide base for slow turns
  - Narrow base for fast turns
  - Which is best for stability of robot?



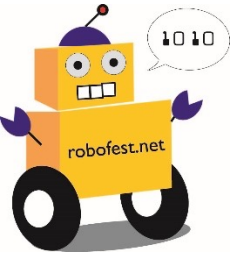
Robot Design cont.

- Wheels
  - How many tires? 2 or more?
  - Tires- rubber or plastic? (for traction?)
  - Size – large or small?
  - Placement of wheels – front or back?



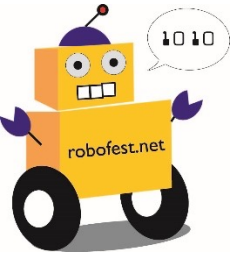
Robot Design cont.

- Sensors
  - Placement of light sensors (height above table? How far in front or to side of wheels?)
  - Placement of sonar sensor (low or high?)
  - Use additional sensor(s)?
    - Touch sensor(s) to detect if an opponent is pushing your robot?
    - Touch sensor(s) to detect if you are pushing an opponent or the bottle?



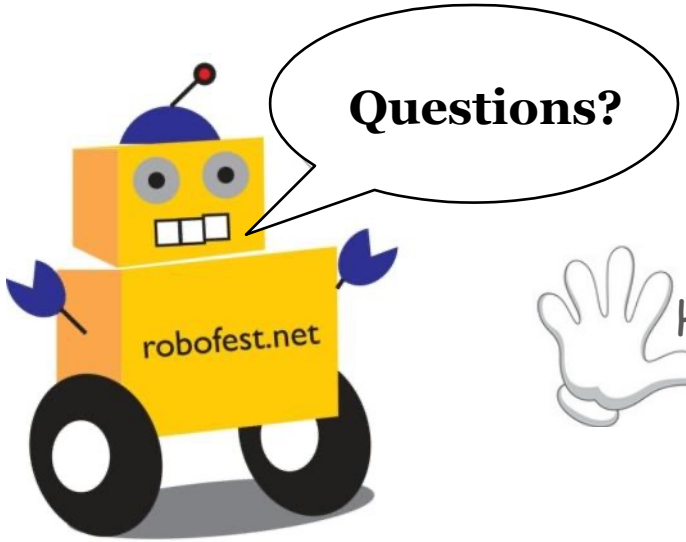
Strategy (for the future)

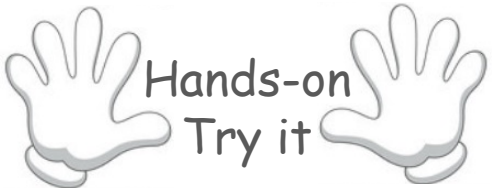
- Try to lift an opponent off the table?
- Try to hide from an opponent (cloaking)?
- Bounce off an opponent if attacked?
- Add a motor to power a mechanical device to attack your opponent (try to flip your opponent over)?
- Better search for opponent? Find again if opponent moves before you get to him.



LAWRENCE TECHNOLOGICAL UNIVERSITY
ROBOFEST

Little Robots, Big Missions





chung@LTU.edu