RoboCup Rescue

EV3 Workshop

Part 1

LEGO MINDSTORMS EV3

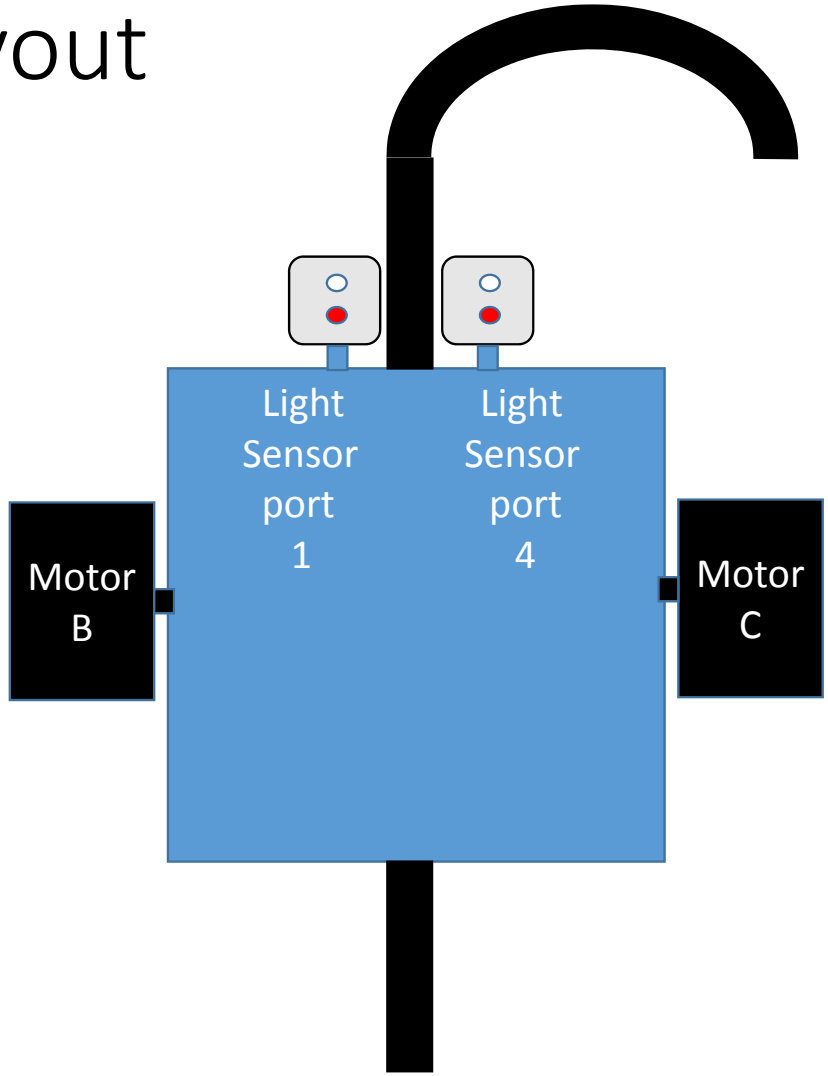
mta modern teaching aids

Energizer

Rescue Rules Changes for 2015

- Open Rescue may have a Silver and Black ‘Rescue Capsule’
- Green may not be Lego Green on intersection or Rescue Tile
- Challenge tiles, in all divisions. Designed to test teams programming skills
- Chemical Spill may be extended in Secondary and Open Rescue
- Doorways must be traversed successfully before a score is awarded
- Sensors must reacquire the line NOT the robot hull
- Reliability Score: Restarts or robot handling will incur a points penalty
- Recue Capsule cannot be replaced if moved by the robot
- Possibly more than five (5) preliminary rounds with a 3 team Trophy round to determine final placings.

Robot Layout

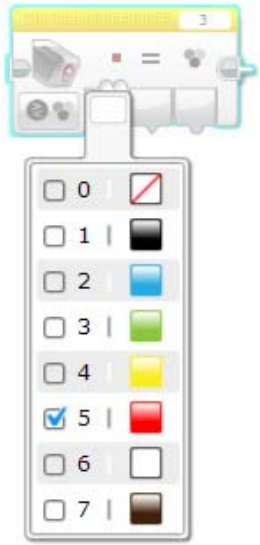


Line Following Logic

- Single Light Sensor
  - Edge Following
    - Left or Right hand side of the line
- Two Light Sensors
  - Still Edge Following but now we can turn both ways at intersections
- Three or more Light Sensors
  - Can be considered Line Following
  - Can navigate the course well but programing is more complex
- PID algorithms (Proportional, Integral, Derivative)

What does the Colour Sensor ‘See’?

- Robot Colour Sensors do NOT ‘see’ as we do.
- Humans can see approximately 10 million colours and shades
- The Lego Colour Sensor can sense 7 Lego colours
- It can only sense 100 shades of reflected light off a surface
- You will need to record what your robots Colour Sensors senses as White, Green, Black and Silver, as values between 1 -100
- Once you have these values, you are able to calculate mid-points so that you can program the robot to make correct decisions.

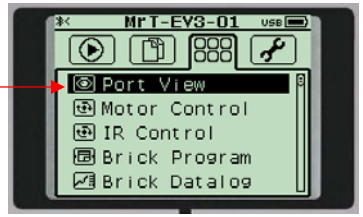


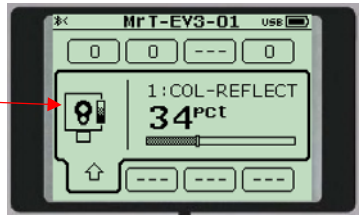
Colour Sensors – a Technical Perspective

- The Lego EV3 Colour Sensor is optimised to detect seven (7) Lego Colours, 0 – No colour, 1 – Black, 2 – Blue, 3 – Green, 4 – Yellow, 5 – Red, 6 – White and 7 – Brown.
- You will get consistent results IF the sensor is between 1 and 2 studs off the surface being read. (one stud is 4.8mm)
- HOWEVER, at 3 studs, Green will read as Black.
- This inconsistency alone places doubt on using a Colour Detection method in a RoboCup Rescue competition.
- When a robot navigates to the edge of one tile transitioning to an inclined tile, the robots sensors will be further or closer to the tile giving inconsistent readings.

EV3 View Mode

- The EV3 has a view mode that allows you to record what the Colour Sensors 'sees' when Comparing-Reflected-Light Intensity
- Place your robots Colour Sensors over the four different colours on the RoboCup Rescue mat
- On the EV3, right button across to the third tab
- Press the Centre button to select Port View
- Use the Left, Right, Up & Down buttons to select your sensors
- Read the values and record them on a table then calculate your midpoint/threshold values





Rescue Mat Colours	Light Intensity	Threshold values
Silver	92	86
White	80	57
Green	34	
Black	22	28

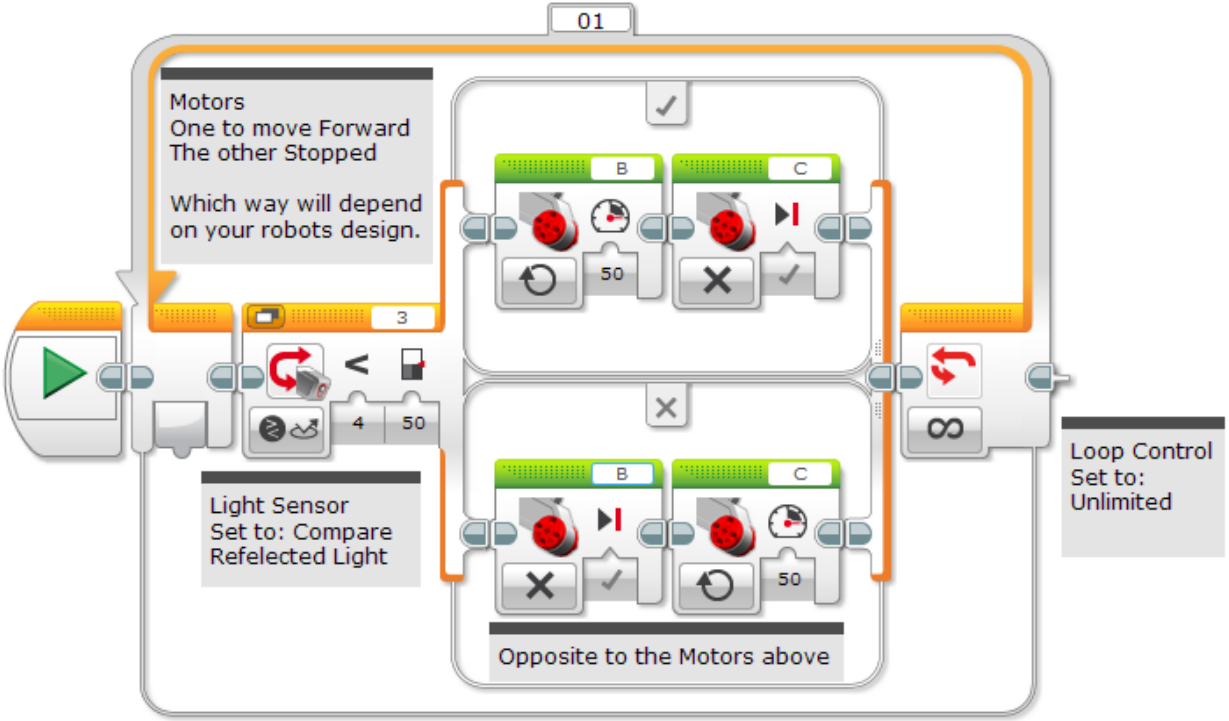
Add columns for each Sensor.
LS1, LS2 etc

Single Sensor Edge Following Logic

- Start the robot with the sensor to the Left of the Line
- Have motor B move forward. Motor C is stopped
  - The robot will turn to the right (towards the line)
- When the Light Sensor detects the Black Line. Stop Motor B
- Have motor C move forward. Motor B is still stopped
  - The robot will turn to the left (away from the line)
- When the Light Sensor detect white. Stop motor C
- Repeat the steps above, over and over to follow the left edge of the line.
- Calculate a threshold value as the mid-point between Black and White

MidPoint = (BlackValue + WhiteValue) / 2

EV3 Edge Following Code

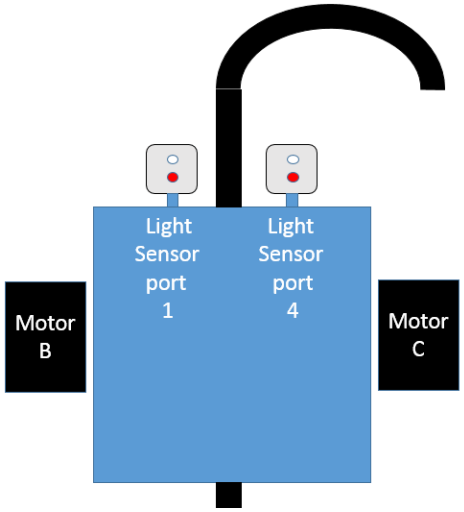


Once you have a Left Edge Line Following robot, create a Right Edge Line Follower

2 Sensor Line Follower

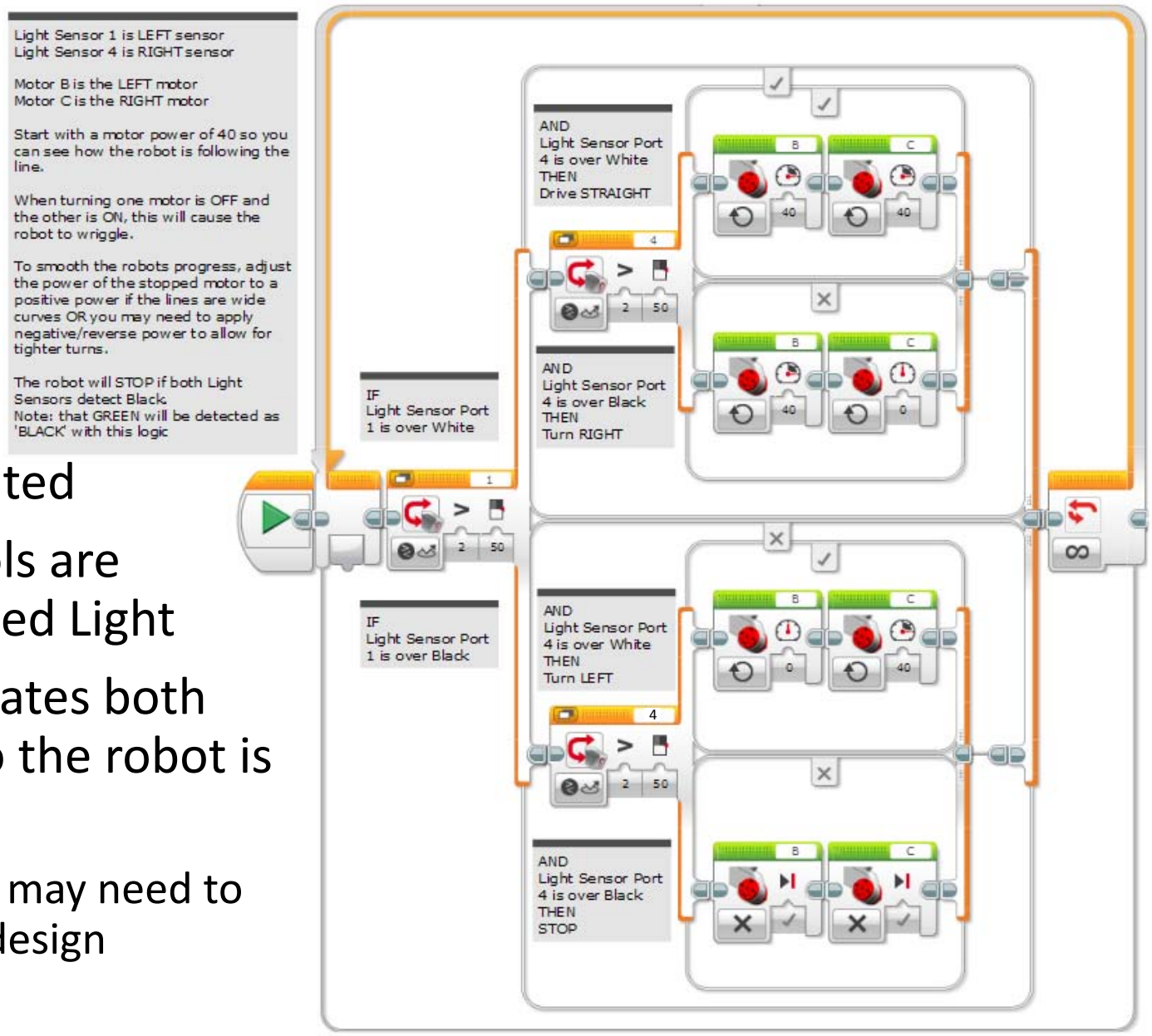
- **4 step logic**

1. IF both Light Sensors detect White then?
   Move Forward
2. IF LS1 detects White and LS4 detects Black?
   Turn Right
3. IF LS1 detects Black and LS4 detects White?
   Turn Left
4. IF LS1 and LS2 both detect Black?
   Stop



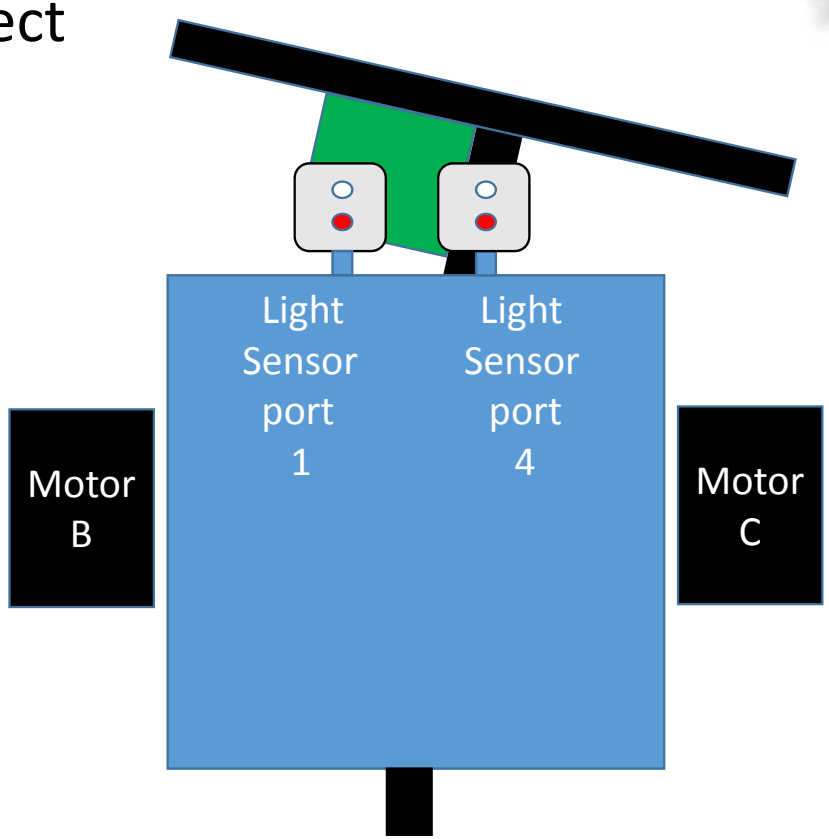
EV3 Two Sensor Line Following Code

- You need One
  Loop Block set to Unlimited
- The three Switch Controls are set to Compare - Reflected Light
- The bottom switch indicates both sensors detect ‘black’ so the robot is programed to Stop.
- Note: your motor directions may need to change depending on your design

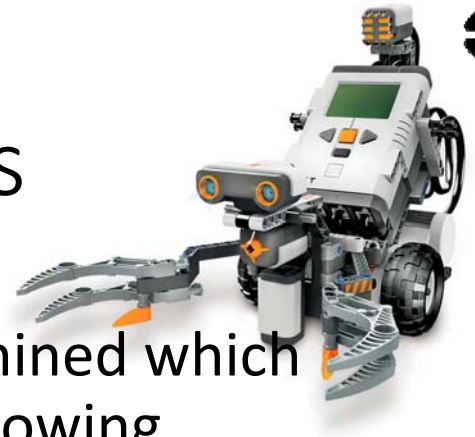


Detecting the Intersections – Method 1

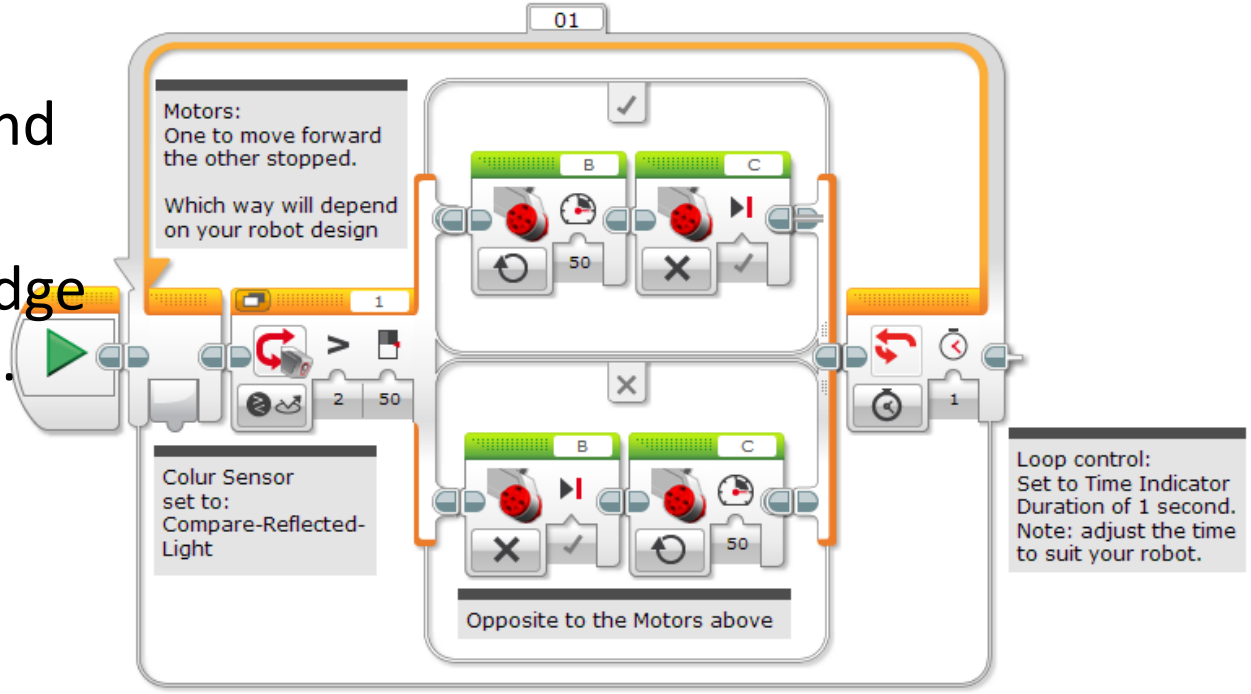
- As an exercise, we will entertain the idea that the Colour sensor can detect the Intersections as Lego Green.
- Change the Colour Sensors mode to Measure – Colour
- IF LS1 detects Green, Turn Left
  IF LS2 detects Green, Turn Right
- Note: this method will only work on fields where Intersection Green is equal to Lego Green, otherwise use method 2 (*go to slide 17*)



Modify the Edge Following programs

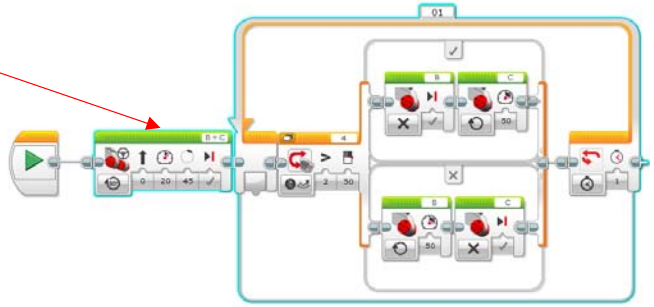


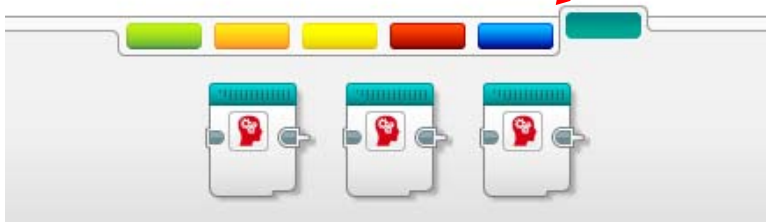
- Once the robot has detect the Intersection and determined which direction to turn, we can use the Left & Right Edge Following programs to move past the Intersection.
- Change the Loop Control to Time Indicator, set to 1 second Adjust this time if needed.
- This should let your robot ‘Edge Follow’ past the intersection.
- After 1 second the main program will revert back to 2 sensor Line Following

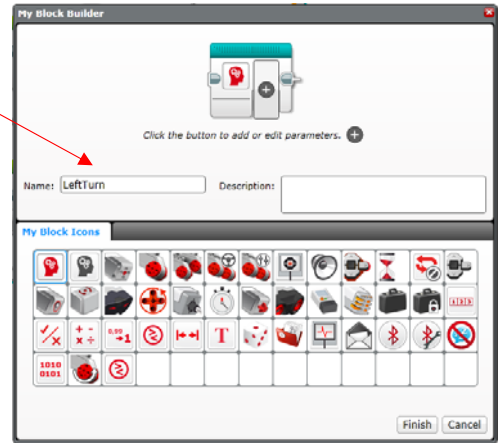


My Block the Edge Followers

- Make the Left and Right Edge Followers into MyBlocks
- Select all of the Left Edge blocks except the start
- Select Tools, then the My Block Builder
- Name it LeftTurn (no spaces) Select Finish
- Your MyBlocks are now stored under the Aqua Palette
- Repeat for the RightTurn

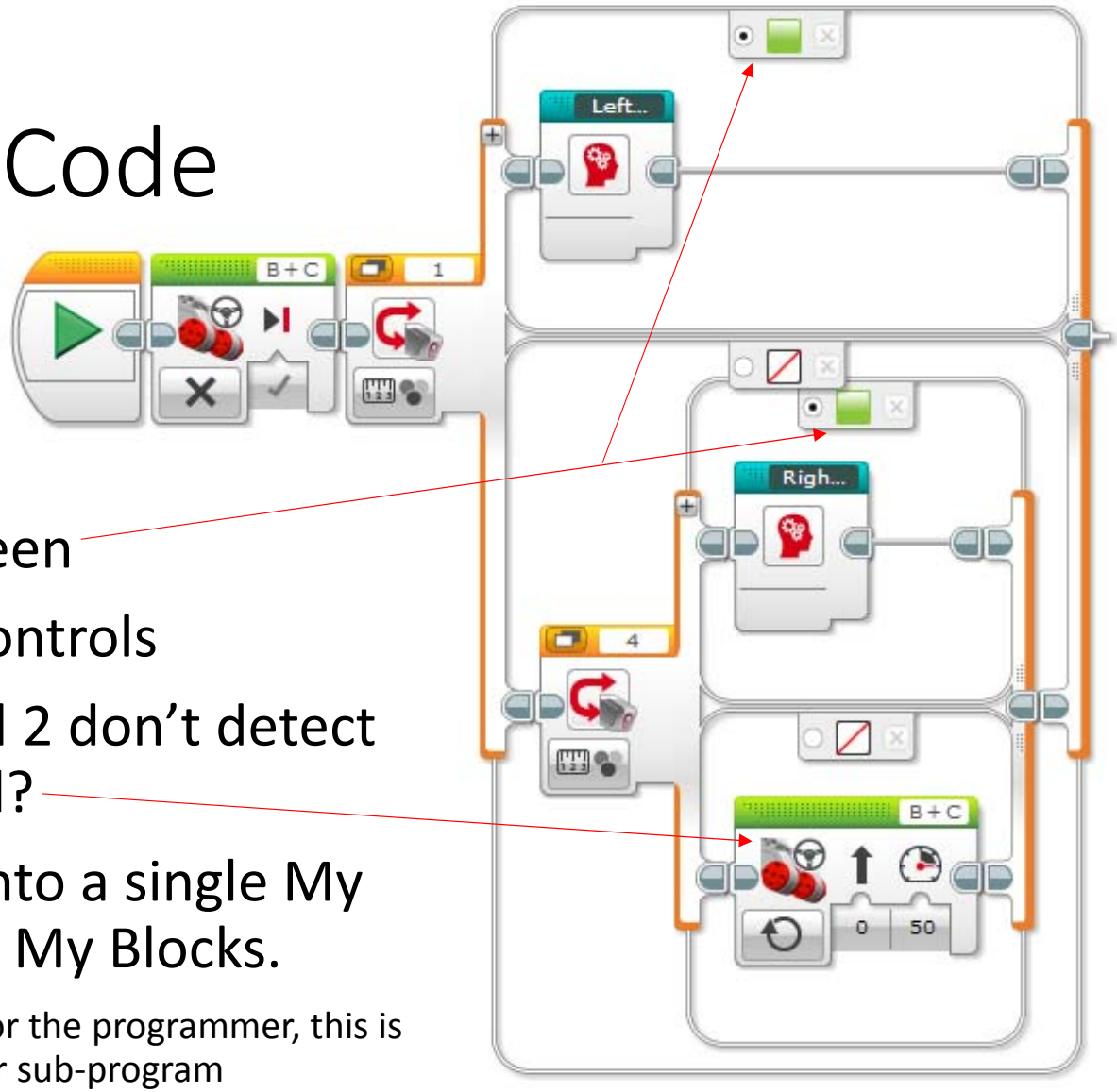






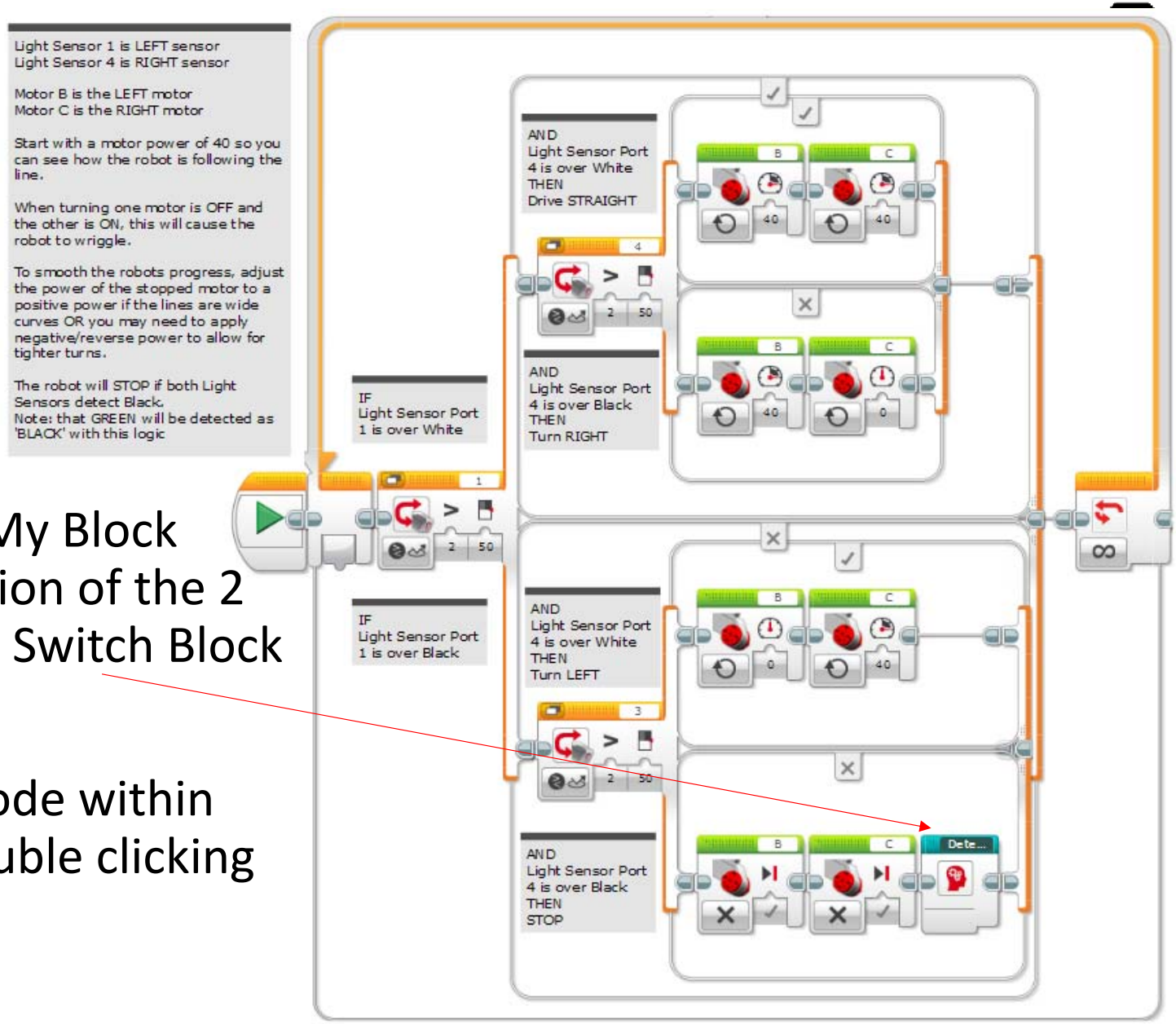
Detect Intersection Code

- Set the two Switch Controls to Measure – Colour
- Change the Colour Mode to Green
- Note the ports on the Switch Controls
- Note: at the moment IF LS1 and 2 don't detect Green the robot moves forward?
- Turn Detect Intersection code into a single My Block including the Edge Follow My Blocks.
- Yes: you can have My Blocks within My Blocks. For the programmer, this is simply calling a sub-program from within another sub-program



Line Following & Intersection Detection

- Include the new Detect Intersection My Block into the bottom section of the 2 Sensor Line Follower Switch Block
- Test your code
- You can adjust the code within the My Blocks by double clicking them



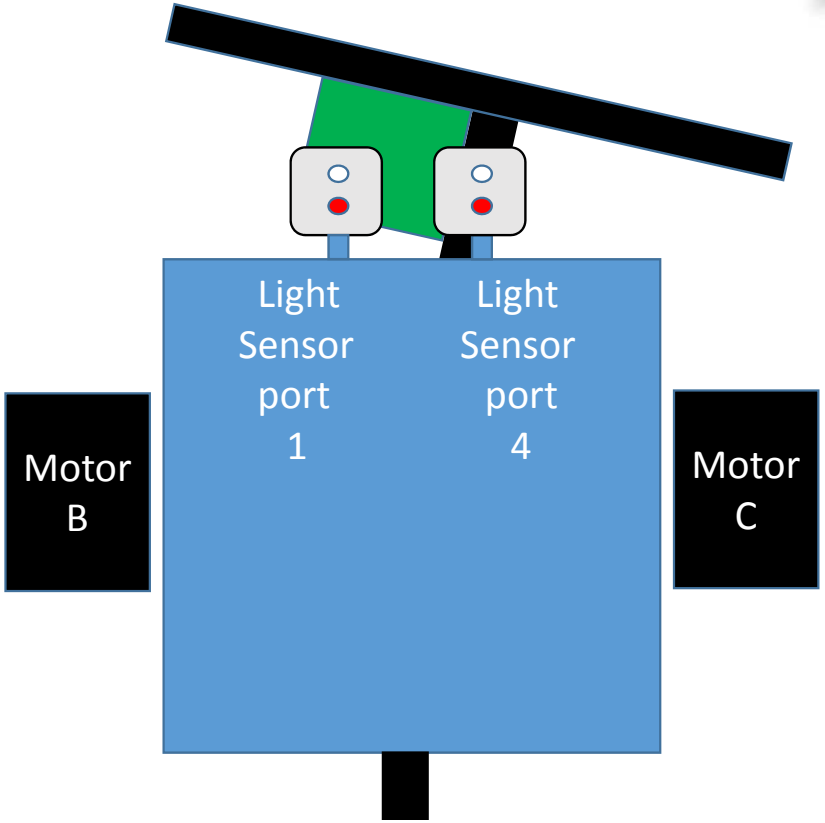
Alternative to Colour Detection

- Lego Green is not supported in RoboCup Junior Rescue which means that teams may need to use alternative methods.
- Select Light Sensors that can detect Black, Green, White and Silver with as big a difference as possible. Have a wide dynamic range.
- You will need to treat the Green Intersections as a shade of Grey and have a multi level algorithm for detecting all the different shades from Black, Green, White to Silver on the Rescue Course.
- The Colour Sensor will need to be set into Compare-Reflected-Light Intensity (Ambient may also work?)

Detecting the Intersections – Method 2

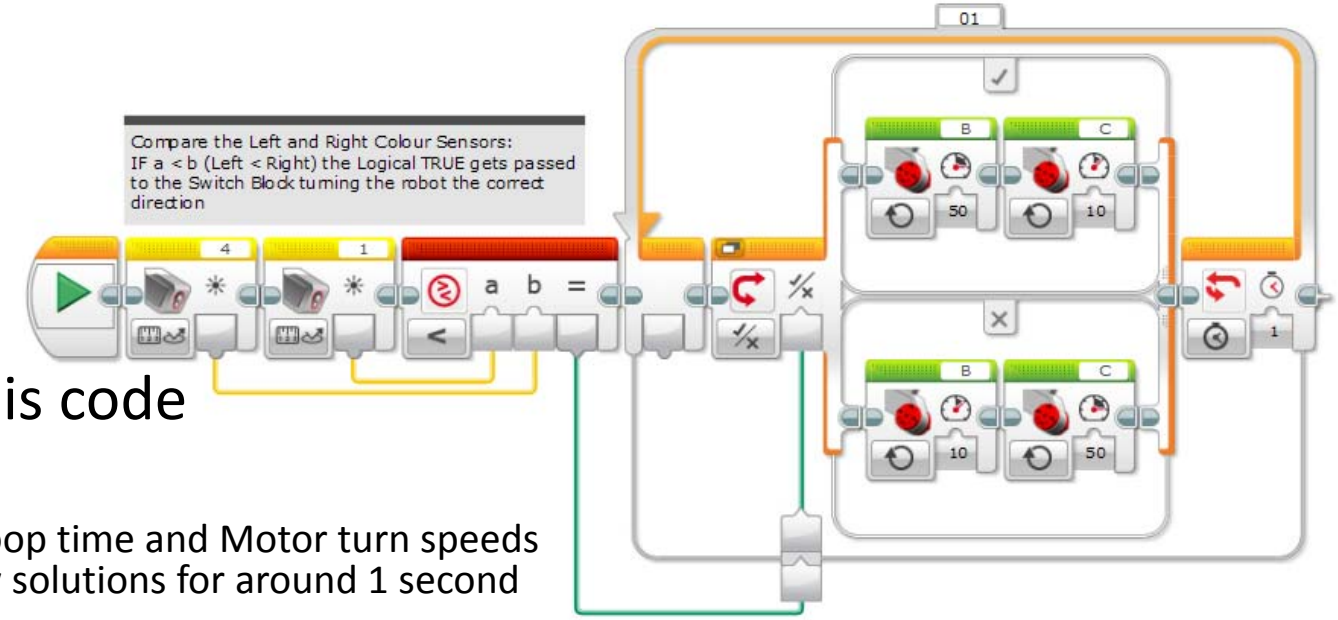
- As the robot moves from side to side along the line it will stop at an Intersection using the 2 sensor code.
- Compare the values of each sensor
- IF LS1 < LS2, **Turn Left**
  left is less than right sensor

  IF LS2 < LS1, **Turn Right**
  right is less than left sensor



Compare the difference

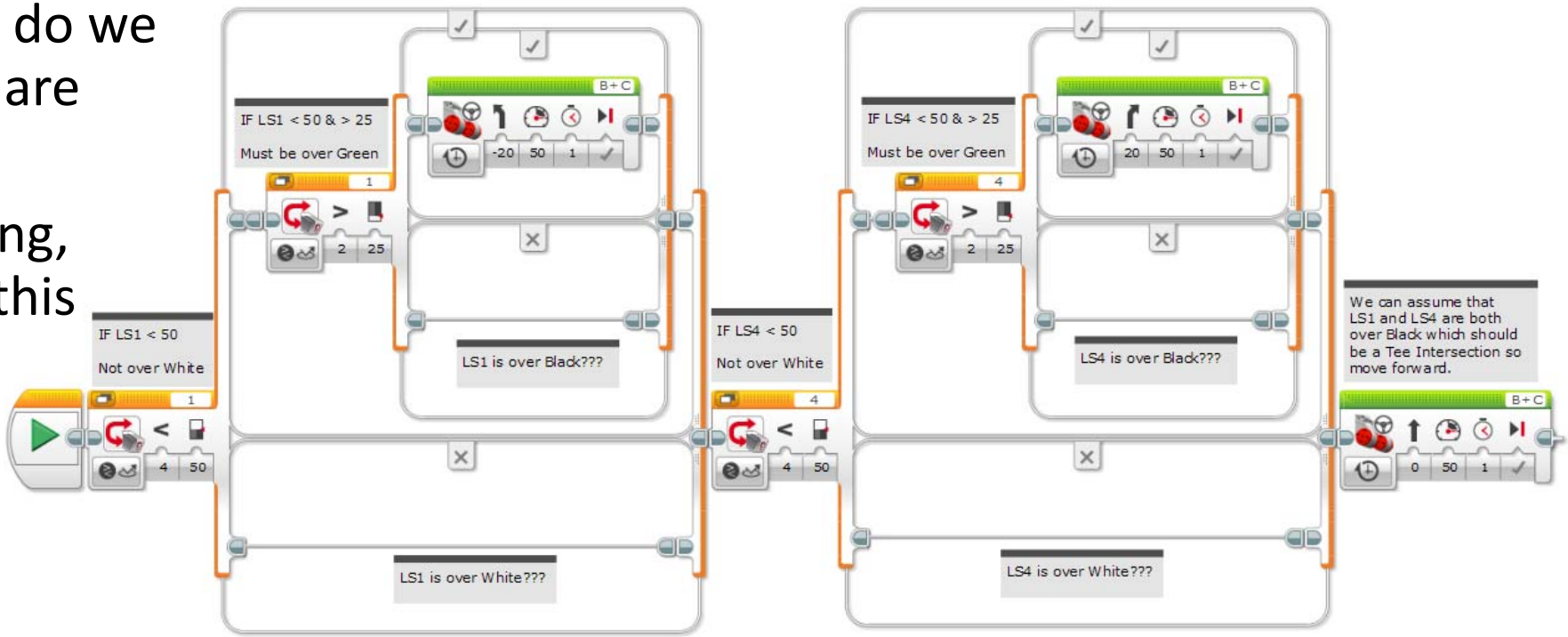
- Having stopped at an Intersection the robot will now compare the amount of reflected light of each Colour Sensor
- A Comparison Block compares the values of the two Light Sensors returning a Logical True or False from is a < b
- This Logical value is then passed to a Switch Block that determines which direction to turn the robot on the course
- After testing My Block this code



- Note: You will need to play with the Loop time and Motor turn speeds OR, use the Left and Right Edge Follow solutions for around 1 second

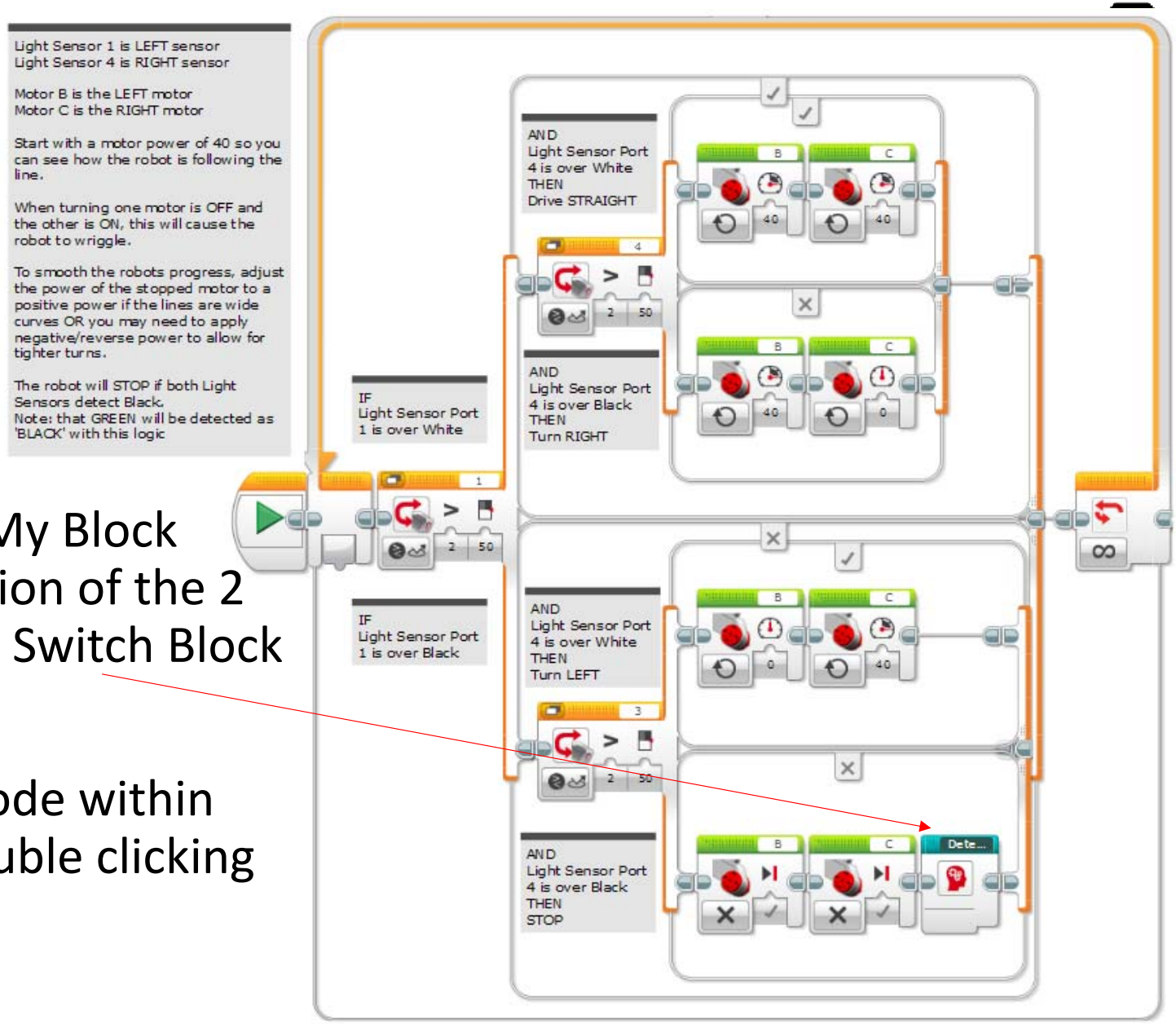
Using Switches for more options

- The example below is a solution using Switches.
- The advantage or possibly disadvantage of this method is we can program for the options of both Black or both white.
- But. What do we do if both are white?
- After testing, My Block this code



Line Following & Intersection Detection

- Include the new Detect Intersection My Block into the bottom section of the 2 Sensor Line Follower Switch Block
- Test your code
- You can adjust the code within the My Blocks by double clicking them



Test, Modify, Test and Modify some more!

- Now you have a basic Line Following program including Intersection detection
- Be aware that there are many variations on the Line Following algorithm so think the process through and come up with your own
- Note: that the physical design of your robot will influence how the robot can navigate the Rescue Course. Make sure your robot can physically navigate around the course e.g. get through doorways
- Change motor speeds, directions, sensor thresholds, one at a time
- Look at data logging to record what your sensors are detecting as the robot travels along the course

3rd Party Sensors and Actuators

- MTA – Everything Lego Education
  http://www.teaching.com.au
- Omni Wheels – RotaCaster designed for Lego
  http://www.rotacaster.com.au/
- Linear Actuators – Firgelli have NXT and EV3 models
  http://www.firgelli.com
- HiTechnic – Official 3rd party Lego Sensors
  http://www.hitechnic.com/
- MindSensor – Unofficial 3rd party Lego Sensors
  http://www.mindsensors.com/
- Dexter Industries – Advanced sensors
  http://www.dexterindustries.com