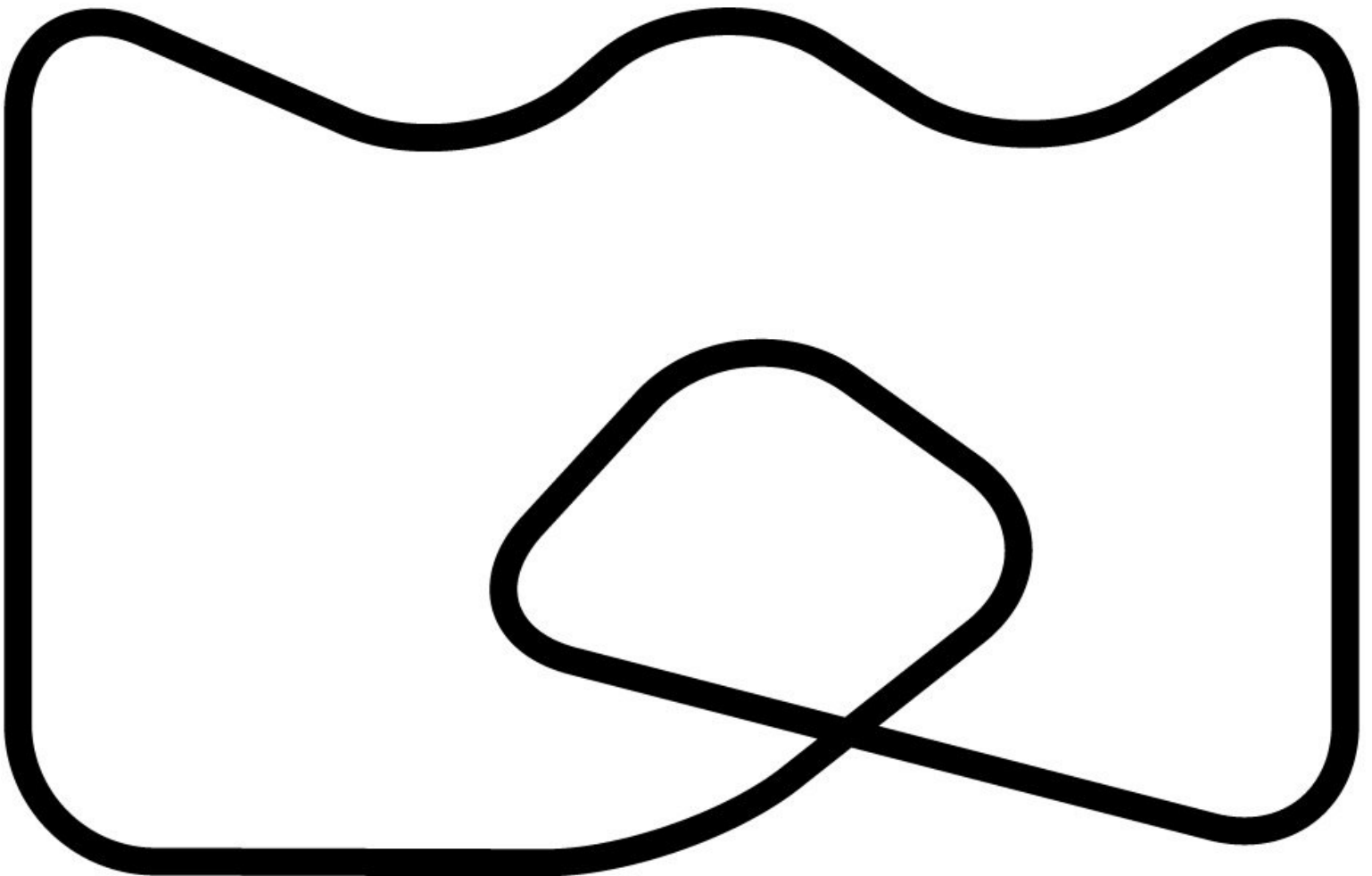


B - Follow the Path

1 line sensor is great, but we can't follow a complicated path - using two sensors however lets us do much more complex things!



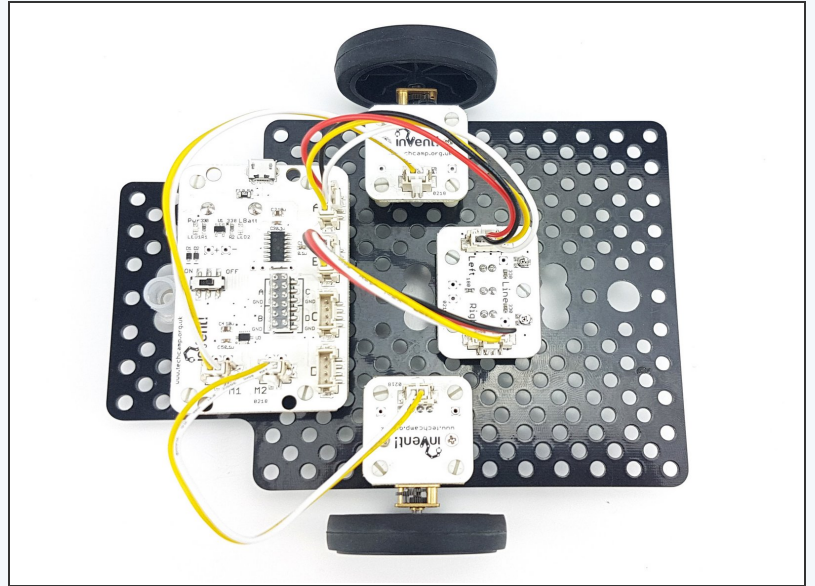
INTRODUCTION

1 line sensor is great, but we can't follow a complicated path - using two sensors however lets us do much more complex things!

Step 1

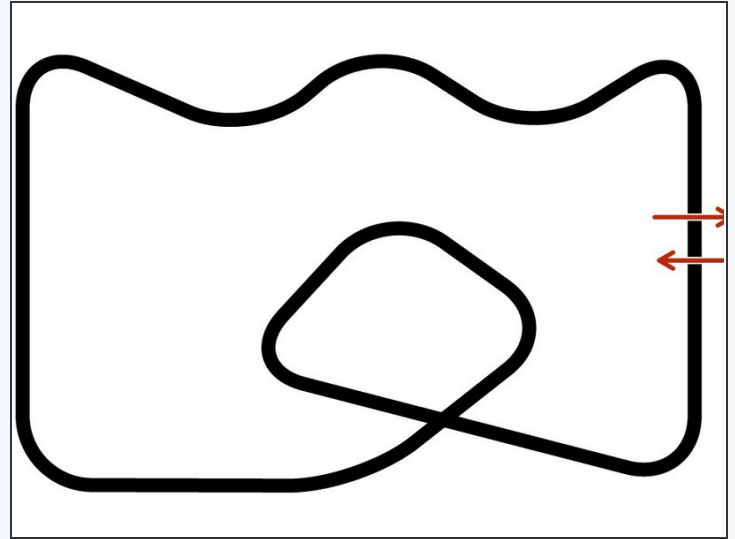
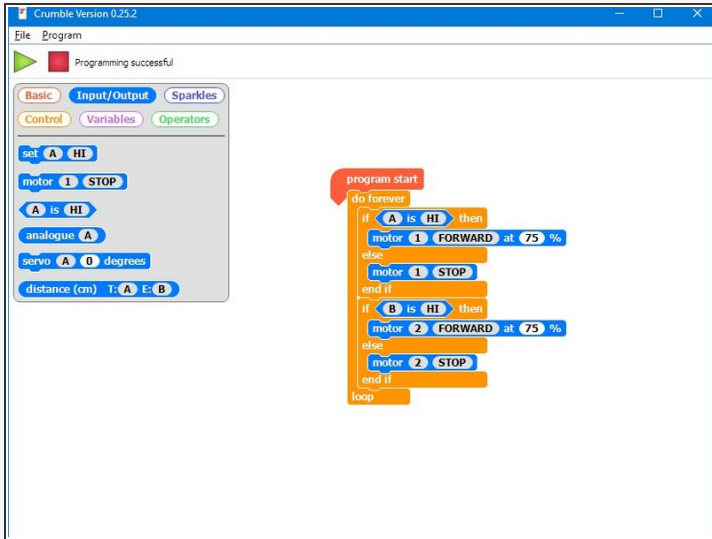
Two Line Sensors

- Assemble your robot like the picture - this time, plug **both the left and right** line sensors in!
- Plug the left one in **A**, and the right into **B**.



Step 2

Test Both Sensors

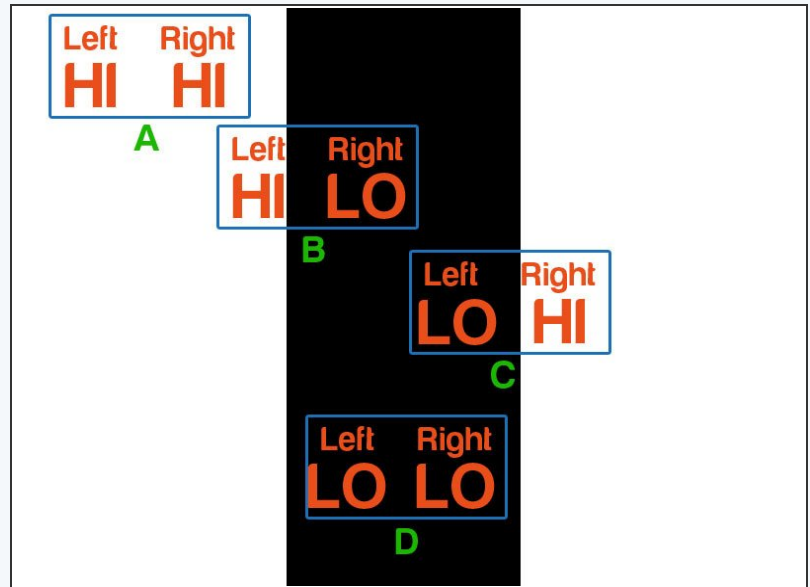


- Let's **test** both sensors so we know how they work.
- **Build** the test program in the picture - can you **guess** what it will do?
- On the other side of your activity mat, **program** your robot and **slowly** move the line sensor **side to side** across one of the lines.
- **What happens** to the motors? Does it do what you expected?

Step 3

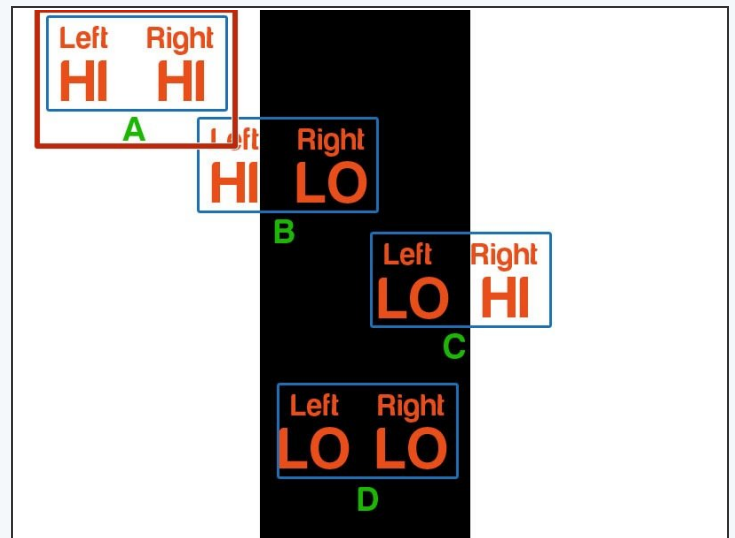
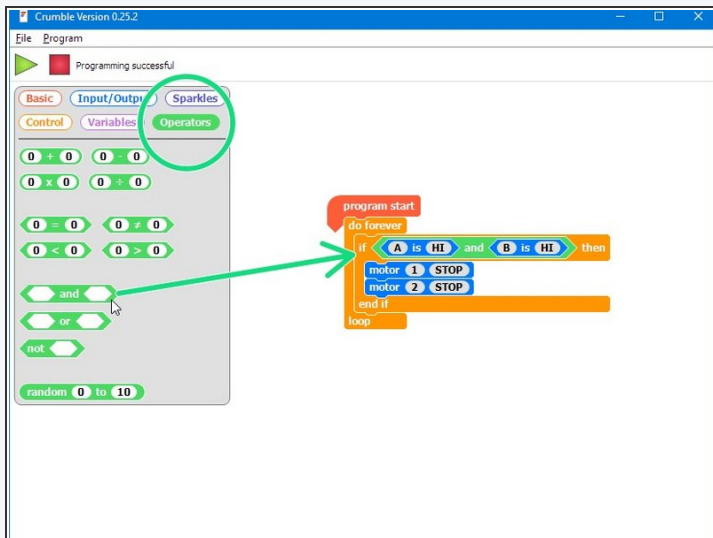
Using Both Sensors

- We need to write a program using the two sensors that follows the **black track**.
- Let's consider **each of the possibilities** in turn, as shown in the diagram:
 - **A** - Off the track completely - **both sensors HI**
 - **B** - Slightly off to the left of the track - left sensor **HI**, right sensor **LO**
 - **C** - Slightly off to the right of the track - left sensor **LO**, right sensor **HI**
 - **D** - on the track, both sensors **LO**



Step 4

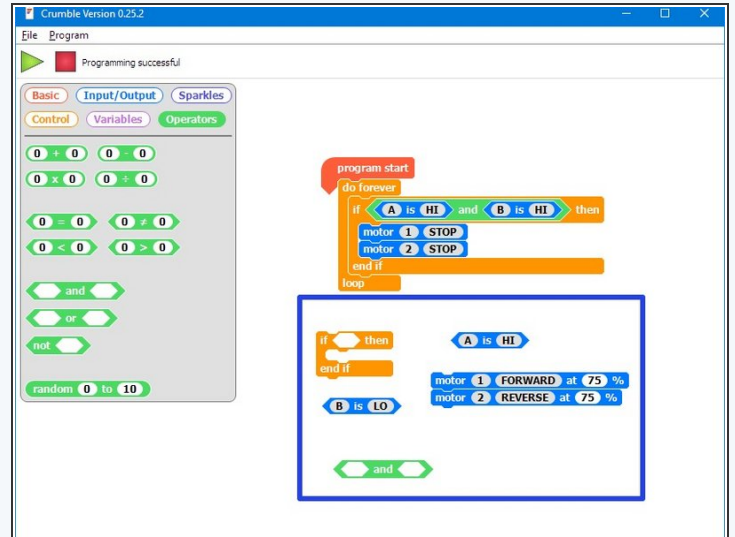
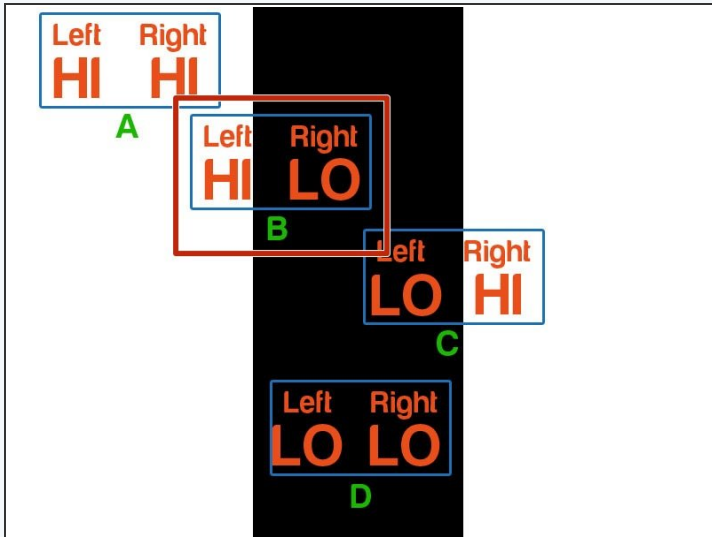
Off the Track



- For case **A**, if the robot goes off the track we need to make it **stop** so it doesn't drive off forever!
- Start your line following program by building the program in the picture.
- We need to check if A is HI **AND** if B is HI at the same time - we can do this with an **AND block**, which you can find in the **Operators** menu.

Step 5

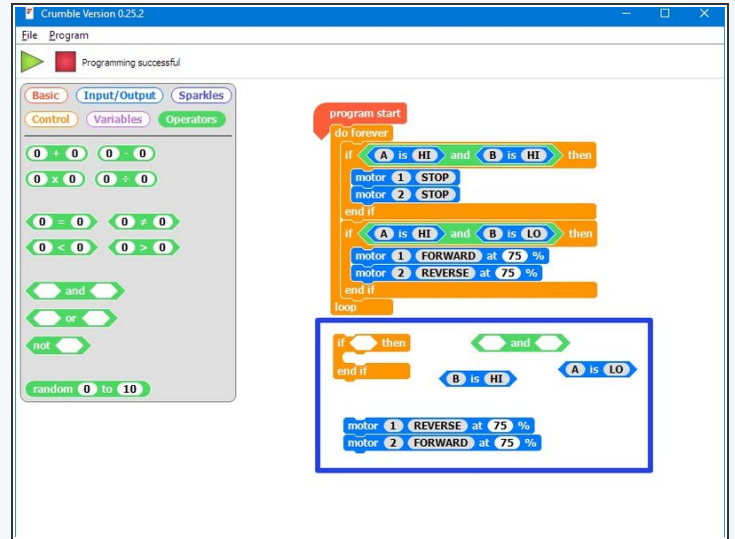
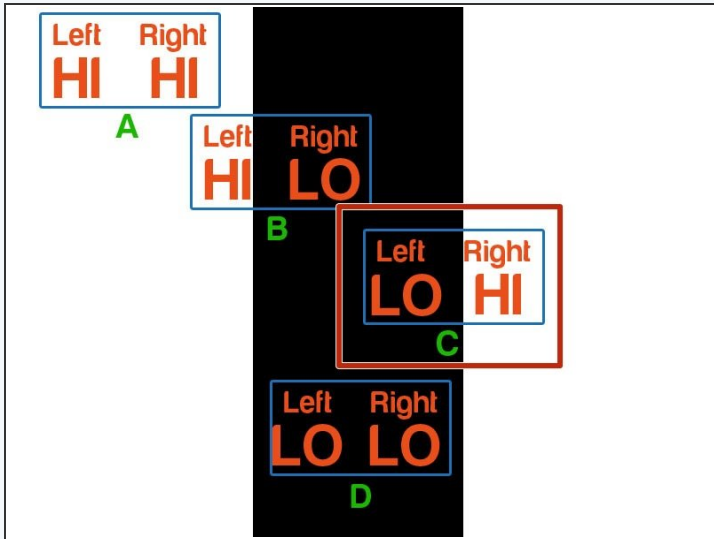
Left of the Track



- For case **B**, we are slightly too far left, so we need to **turn right** to get back on the line.
- **Add** some more blocks to check the sensors, and **turn right** if we are slightly to the left of the track.
- There are some **hint blocks** if you need them!

Step 6

Right of the track

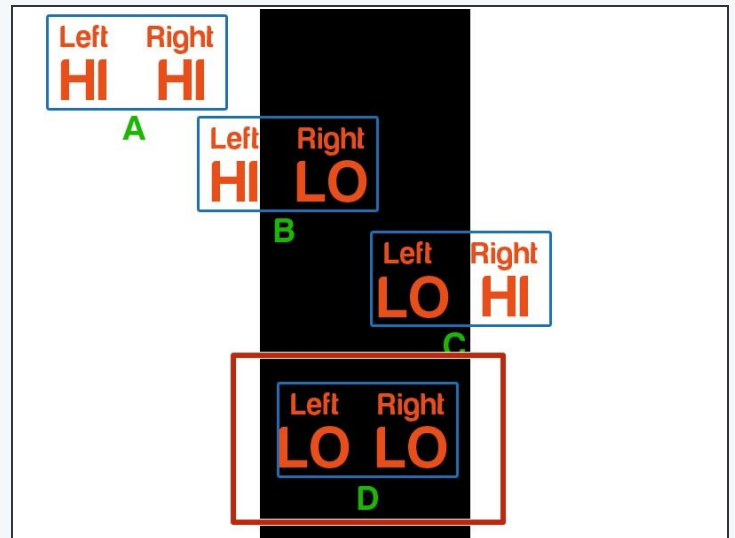


- For case **C**, we are too far right, so need to **turn left** to get back on the track.
- **Add** some more blocks to your program to **check the sensors** and **turn left** if we need to!
- There are some more **hint blocks** if you need them.

Step 7

The completed line follower

Challenge!



- Finally, we need to check for case **D** - both sensors are **LO** so we are **on the track**, and just need to go **forwards**.
 - **And some more blocks** to your program to complete it, and **test** your robot on the track.
 - It should be able to make it **all the way around on its own!**
- i** If you're robot keeps coming off the track, try **increasing the amount it turns**, or adding a **small wait** after the turn to make it turn even more.

Step 8

Find the Path

- Currently, if the robot goes **off the path completely** (or the path ends) it just **stops**.
- It would be more useful if the robot tried to **find the path again!**
- **Change** your program so that instead of stopping, the robot drives so that it might **find** the path again. You can make this **as complex as you like!**
- Some ideas:
 - **Reverse** in a straight line
 - Drive **forwards** whilst sweeping **left and right**
 - Drive in increasing size squares (**hard**)
 - Drive in an increasing size spiral (**v. hard!**)

