



Following the Wall

Programming MARIO



1st - Build MARIO

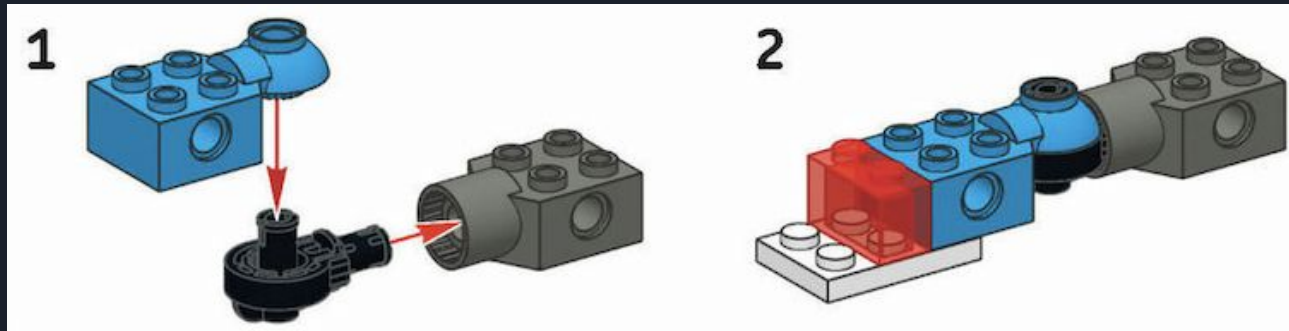
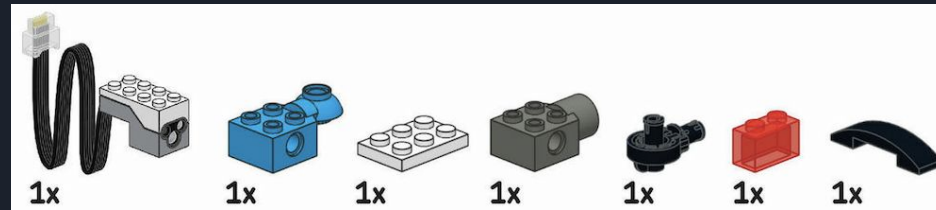
Find the directions in the activity
or on the
Enrichments Page -

<http://lowerschooltech.sites.da.org/enrichments/>

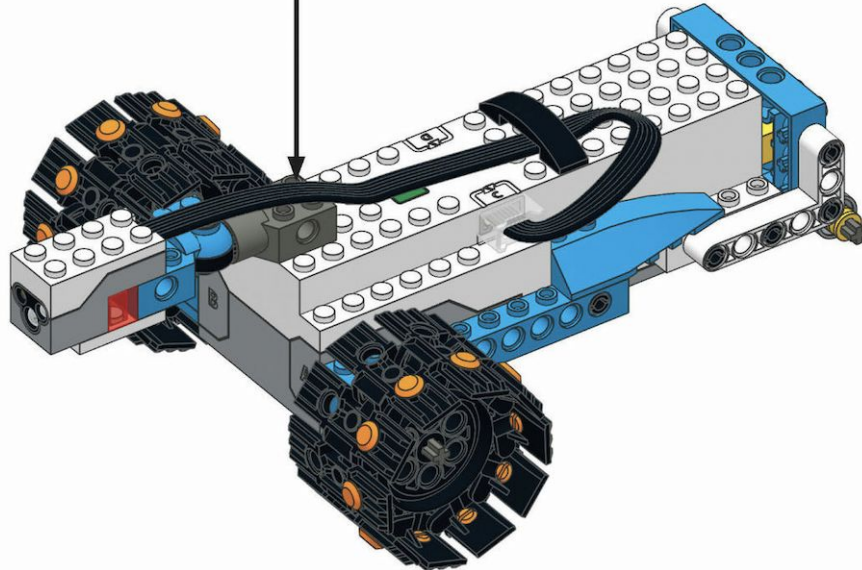
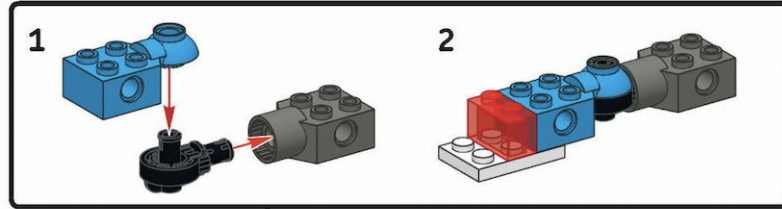
Then add the sensor.

(Directions on next two pages.)

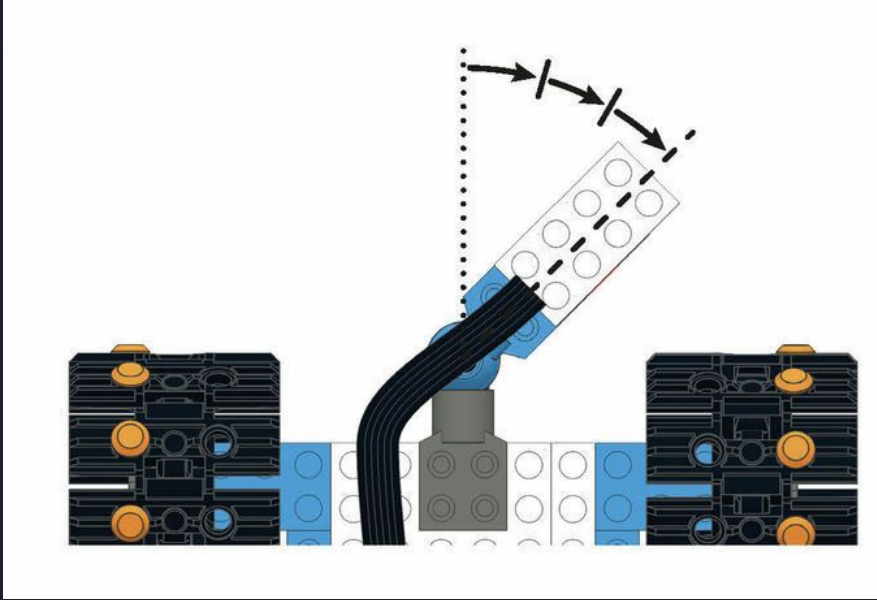
Building the Sensor Head Part 1



Building the Sensor Head Part 2

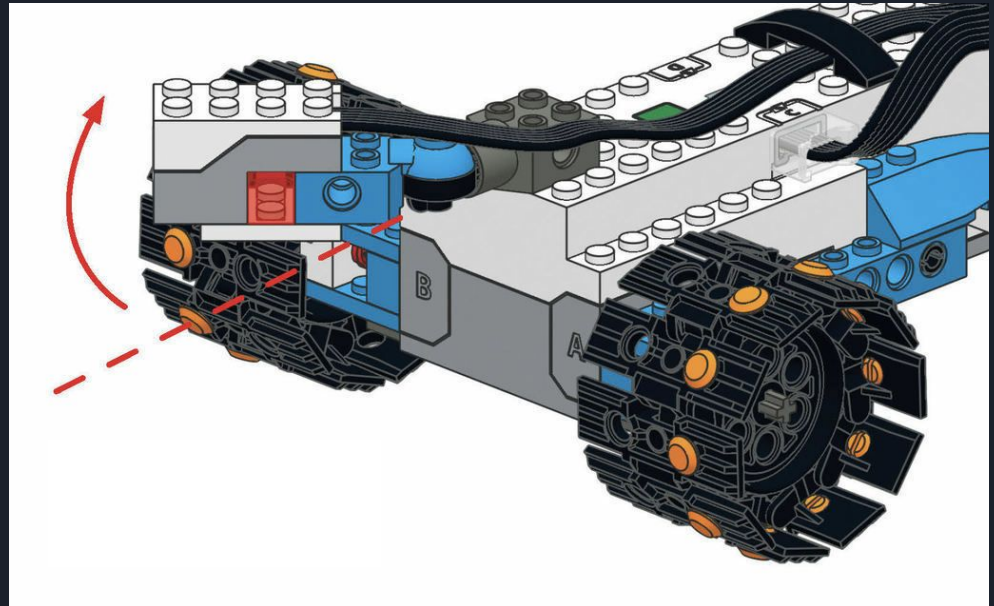
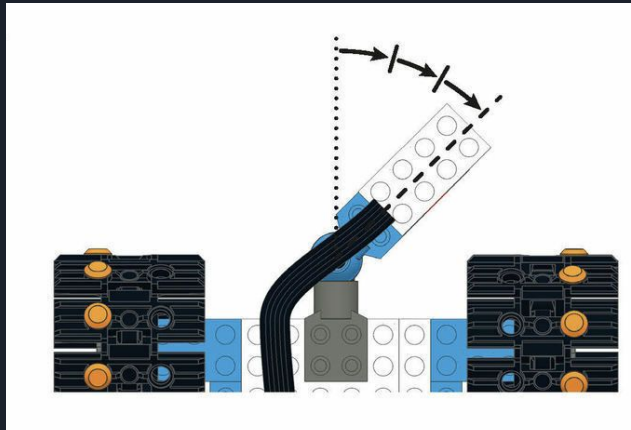


What MARIO Will Do



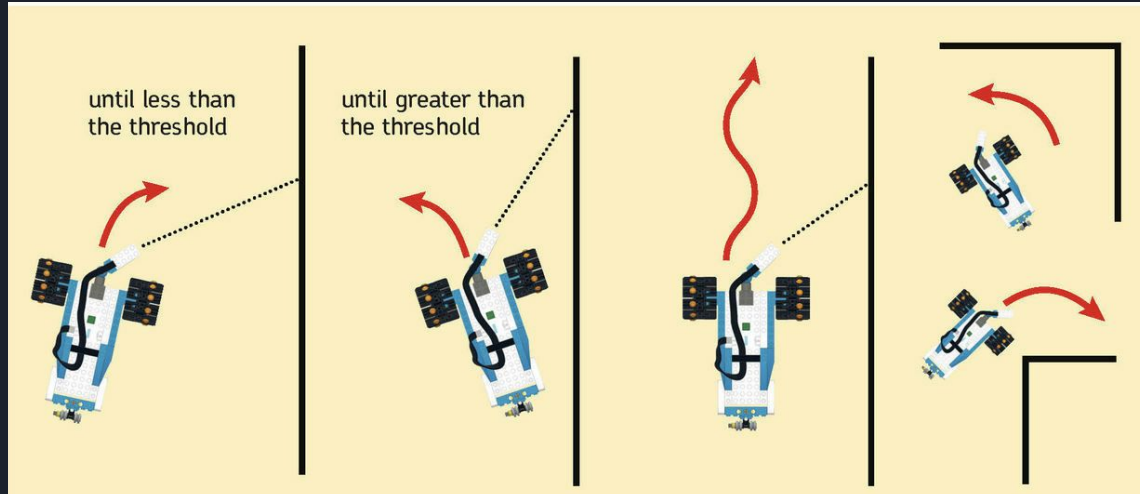
- Use the sensor to stay a constant distance from walls or other obstacles.
- Follow the wall on the RIGHT side.

Get MARIO Ready



Rotate the SENSOR HEAD 3 clicks to the right so that it points at a 45 degree angle.

How Wall Following Works

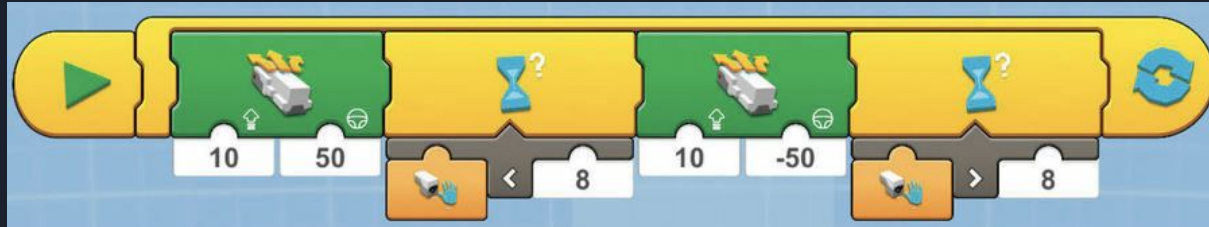


MARIO will use the sensor to measure its distance from the wall and choose an action based on what the sensor reports.

Using “Wait for True” Blocks

This program has MARIO wait for the distance between it and the wall to be above or below a set amount (the threshold) before taking action.

*Create a new program and call it WallFollow.



Test the program by placing MARIO near a wall, with the wall on the RIGHT side.

- What do you notice about how this program works?
- How well does MARIO deal with corners and edges?
- Try lowering the threshold from 8 to see how its movements change. Is it better or worse?



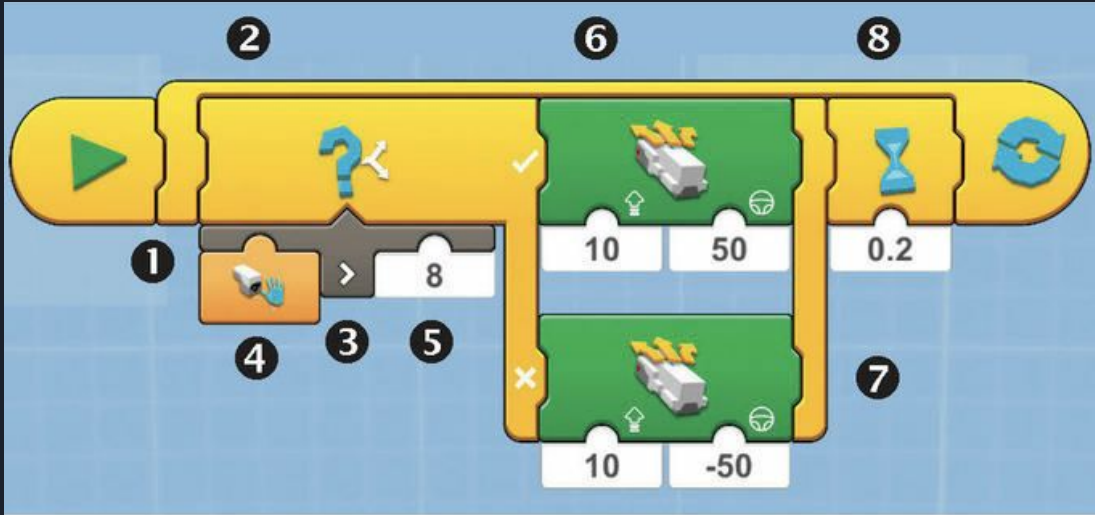
Write Multiple Sequences in One Project

*You will explore more than one way to write this program.

- You will keep all of these code sequences in the WallFollow Project.
- Move the other sequence up (or down) and add the new one.
- Start each new sequence with a Start block.
- You find the Start block in the yellow tab.
- Tap the Start block attached to the sequence you want to test.

Using "Switch" Blocks

*Compare Greater Than block is in the white tab.



1. Infinite Loop
2. Switch Block
3. Compare Greater Than Block
4. Sensor Distance Reporter
5. Threshold
6. Tells MARIO - Steer Right
7. Tells MARIO - Steer Left
Both are part of - Drivebase Move Steering Unlimited Block
8. Wait for Time Block

Create the sequence above, read the description on the next slide that explains how it works, and test your program.

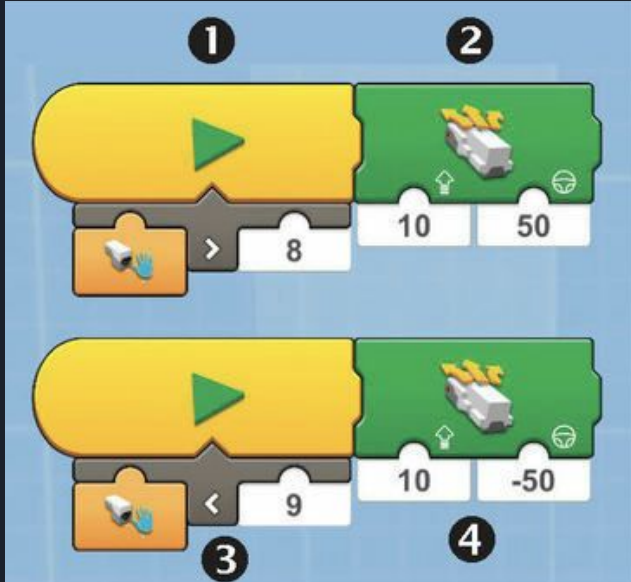


Using “Switch” Blocks

*The sequence explained.

- The SWITCH block makes MARIO decide how to move based on the result of a comparison.
- The COMPARE GREATER THAN block uses the value reported by the SENSOR DISTANCE REPORTER and the threshold (you set) to make that comparison.
- The comparison - Is MARIO is greater than 8 away from the object?
- If the result is true, the DRIVEBASE MOVE STEERING UNLIMITED block tells MARIO to Steer Right.
- If the result is false, the DRIVEBASE MOVE STEERING UNLIMITED block tells MARIO to Steer Left.
- The WAIT FOR TIME block creates a tiny pause to keep the app from processing the directions too quickly and becoming overwhelmed.

Using “Start on True” Blocks



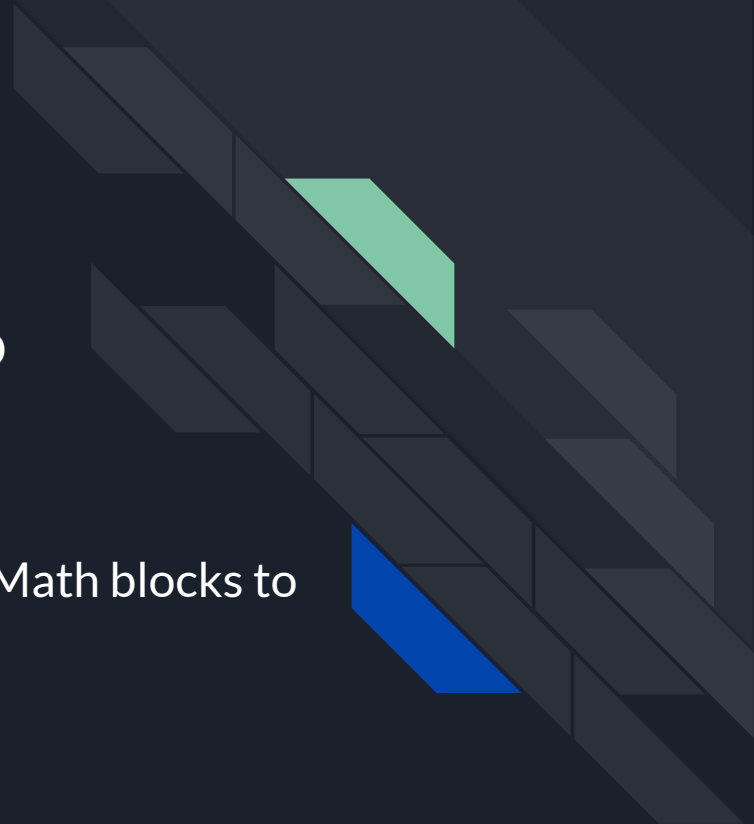
1. Start on True block waits for the sensor to measure a distance greater than 8.
2. Then the Drivebase Move Steering Unlimited (DMSU) block moves MARIO toward the wall (to the right).
3. The other Start on True block waits for the sensor to measure a distance less than 9.
4. Then the DMSU block moves MARIO away from the wall (to the left).

To start this program, tap both Start on True blocks at the same time (1 and 3).
This program is only running 1 sequence at a time.

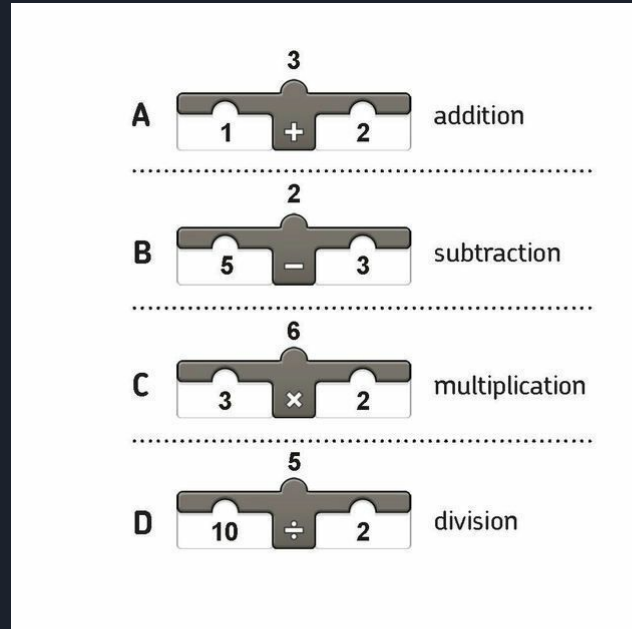
A Smoother Wall Following Program with Math

The programs you have tried so far switch between 2 steering directions depending on the distance MARIO was from the wall.

To make MARIO's movements smoother, we will use Math blocks to write a program for that uses "proportional steering."

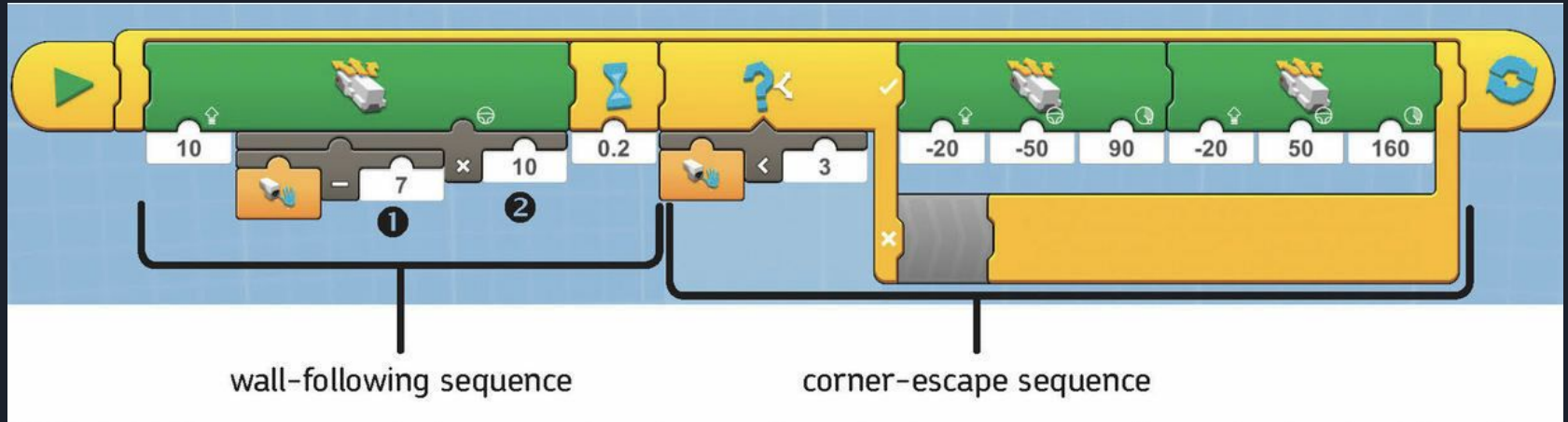


Math Blocks - White Tab



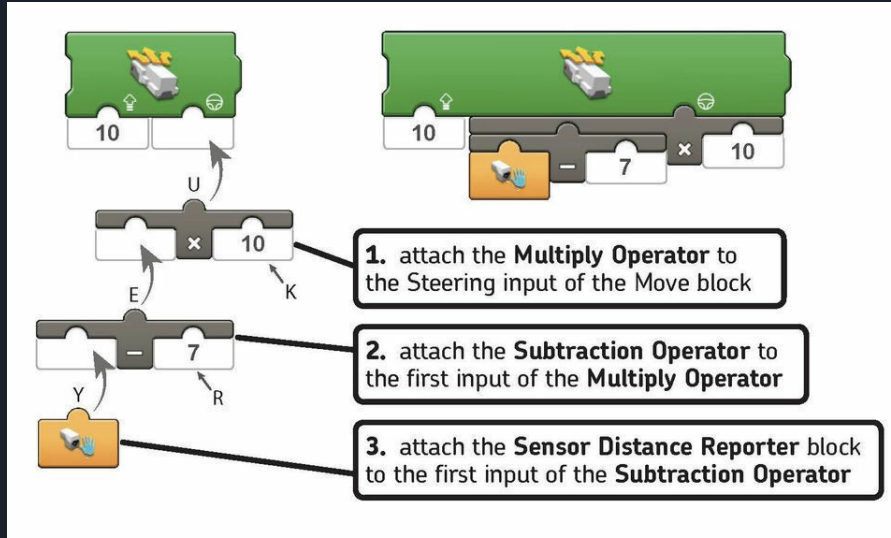
You will use Math blocks and Compare blocks to compute the value of Steering Input for a Move block using the value from the sensor. *Don't worry, I will show you how.

Proportional Steering



Create the sequence above and test your program. The next 2 slides will explain how this sequence of code works.

Creating a Proportional Controller



This piece of code is called a proportional controller because it produces a command for a robot that is proportional to the reading of the sensor.

“Proportional” means that the steering level is big if the robot is far from the desired distance from the wall and there is very little steering when the robot is close to the desired distance from the wall.

We use the number 10 to control the strength of the robots steering.

Backing Out of Corners



With the wall following sequence written as shown, MARIO should be able to turn without getting its right wheel stuck in the corner of the wall.

HOWEVER, when it comes to corners, the proportional controller might not be enough. MARIO might hit the wall in front of it before the steering command becomes strong enough to turn MARIO sharply away from the wall.

To solve this problem, we use a Switch block to activate an escape maneuver when the distance measured by the sensor block is less than 3.

The code above uses 2 Drivebase Move Steering for Angle blocks to make MARIO back out of the corner while turning left.



What You Have Learned

- How to program a robot to explore an enclosed space while following walls at a safe distance.
- You did this 4 different ways:
 - Using **Wait for True** blocks
 - Using a Switch block
 - Using Start on True blocks
 - Creating a Proportional Feedback Controller
- How to use Math blocks to do math operations, which you used to control MARIO's steering level based on its distance from the wall.



Challenges - Do you want to try more?

- Rotate the sensor to look ahead to the left and change the programs to make your robot follow walls on its left.
- Make a program that allows you to control a robot without touching it. For example, when you place your hand in front of a sensor, the robot should keep a distance of 5 from your hand. (Hint - use a proportional controller that uses a speed input for a Move block.)