ADVANCED EV3 PROGRAMMING LESSON

Gyro Turns

By Droids Robotics and Construction Mavericks

Lesson Objectives

1. Learn what Gyro Lag is
2. Learn two ways to correct for this lag
3. Understand why it is important to explore alternative solutions to a problem

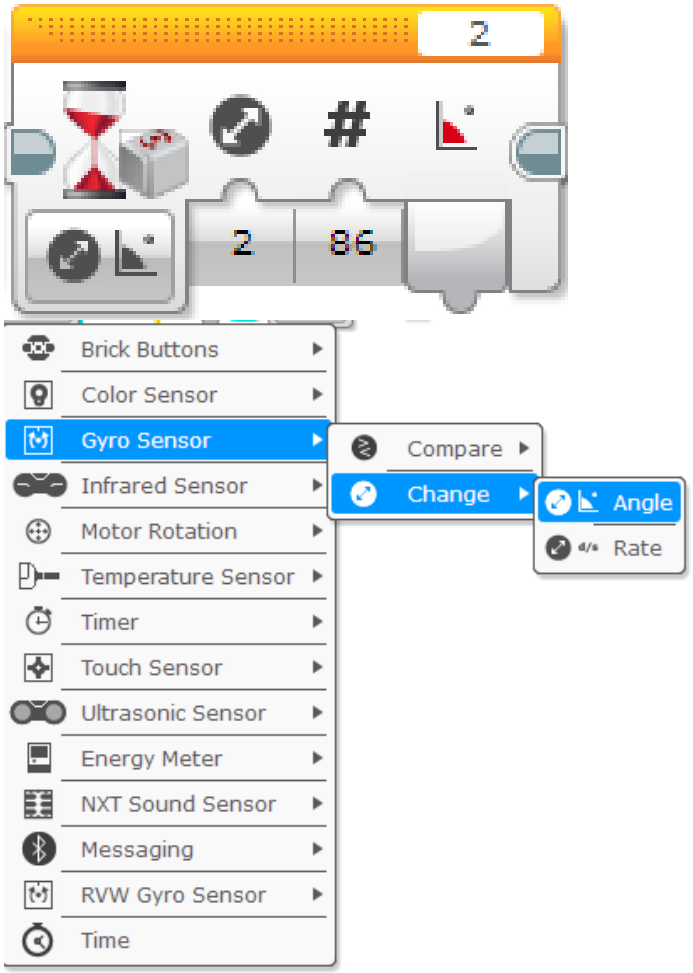
- Pre-requisites: My Blocks with Inputs and Outputs, Data wires, Math Blocks, Loops, Proportional Control

Gyro Problem 2: Lag

- What is lag?
  - The gyro sensor readings lag behind the true value sometimes
- When the turn starts, it takes time for the gyro to begin changing
- This lesson presents two ways to deal with lag in a turn
  1. Reduce the amount of angle that you turn to compensate for lag (slides 4-9)
  2. Use proportional control to continue performing your turn for a requested duration (slides 10-12)

Change Mode in Wait Block

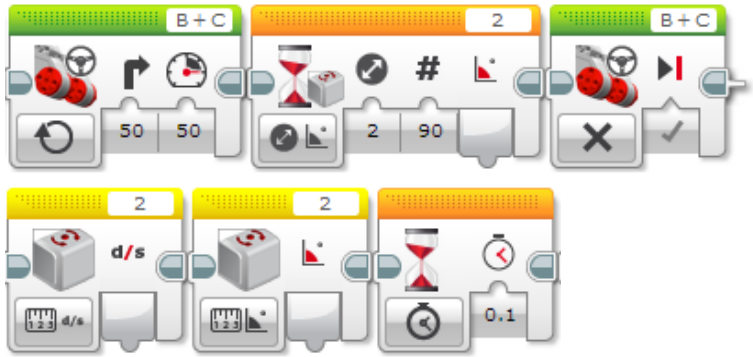
1. In this lesson we use the Wait Block (gyro sensor) in Change Mode
2. Advantages over Compare Mode:
   - You do not need to reset the gyro beforehand
   - You can measure if the value has changed the target degrees by both decreasing or increasing (no need to change the wait block for a left turn)
3. Direction (the first input) defines:
   - 0 – check if the value has increased the desired degrees
   - 1 – check if the value has decreased the desired degrees
   - 2 – check if the value has either increased or decreased the desired degrees



Gyro Turn in Four Easy Steps

STEP 1: Create a simple Gyro Turn program that turns 90 degrees using the Wait for Gyro block in Change Mode

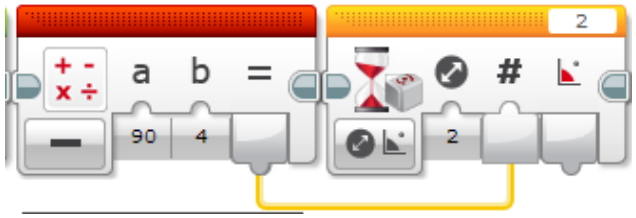
Remember to Calibrate the Gyro before the Wait For Block (see Gyro Lesson for help)



STEP 2: Compensate for Lag

A. Compensate for the lag by reducing the amount of angle to turn based on your robot (e.g 86 degrees instead of 90 degrees)

B. Use a Math Block to create an automatic calculator to compensate for lag



STEP 3: Create and Wire the My Block

STEP 4: Repeat the steps to make one for Left Turns vs. one for Right Turns.

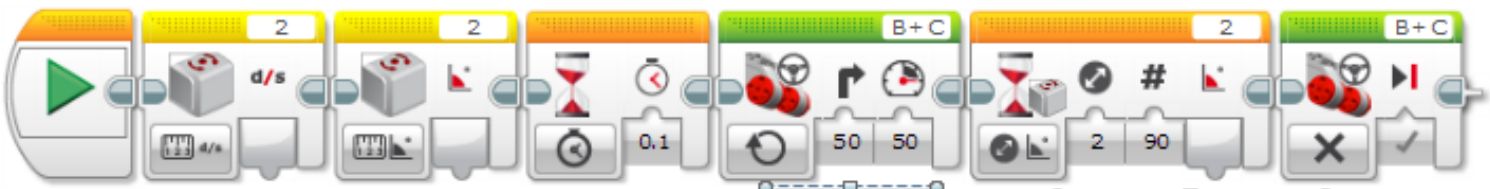
Step 1: Simple Gyro Turn

GOAL OF THE PROGRAM: Simple turn degrees using the gyro

This code is setup for the gyro being connecter to port 2 ; adjust as nedded.

Install tips: The gyro can be anywhere on your robot (even hidden or upside down is okay).

This program turns and waits for the gyro to read 90 degrees. This will make the robot turn 90 degrees to the right.



STEP 1: YOU NEED TO Calibrate THE GYRO: We have these two blocks here because the gyro sensor reading somtimes continues to go up even though the robot is still. By reading the rate of the gryo then the angle the gyro sensor gets recalibrated. Make sure to run these blocks when the robot is still.

We wait 0.1 seconds because it takes time for the gyro sensor to reset to zero.

START TURNING

This block waits until the gyro sensor reads 90.

This stops the motor so it does not keep moving after you reach 90 deg.

Step 2A: Dealing with Lag

Problem with the Step 1: You will find that the gyro does not go the degrees you want it to. If you set it to turn 90 degrees, sometimes it overshoots to 93. You need to make adjustments for your robot because of this. For ours, we needd to turn only 86 degrees in order to turn 90 degrees.

Program goal: A more precise gyro turn

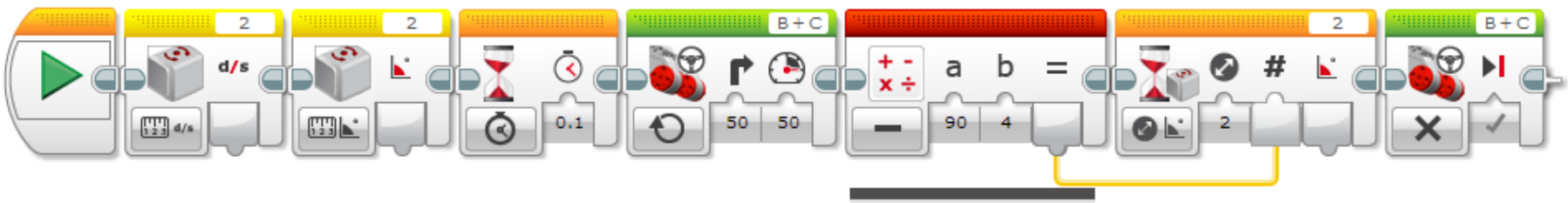
This program turns the robot a bit less than 90 degrees to reach exactly 90 degrees. This value will have to be changed for your robot. The reason the robot does not turn exactly 90 deg. when you type in 90 is because the gyro readings lag behind the robot's actual position.

The only change made from the previous step is that we are now waiting untill the Gyro reaches 86 degrees rather than 90 in this block. This will make a more precise turn

Step 2B: Automatically Correct for Lag

Program goal: subtract degrees automatically

We subtract 4 degrees from your desired degrees using a math block, so we do not need to type in 86 degrees to do a 90 degree turn

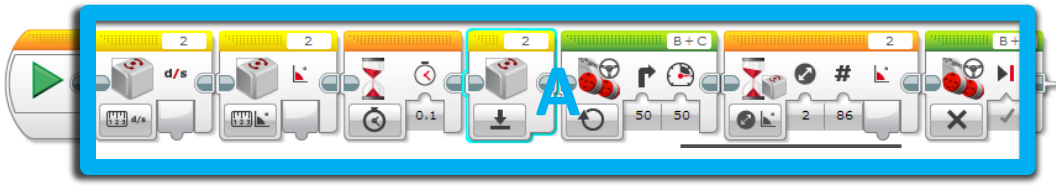


This block was added to automatically correct for lag.

The desired degrees is inputed into input a

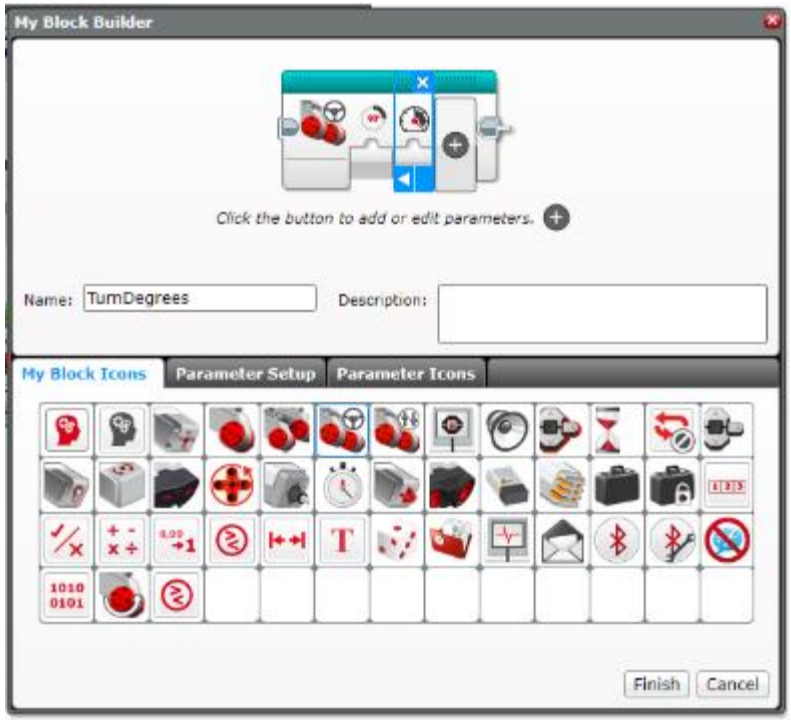
Step 3A: Create a My Block

A. Highlight all the blocks then go to My Block Builder



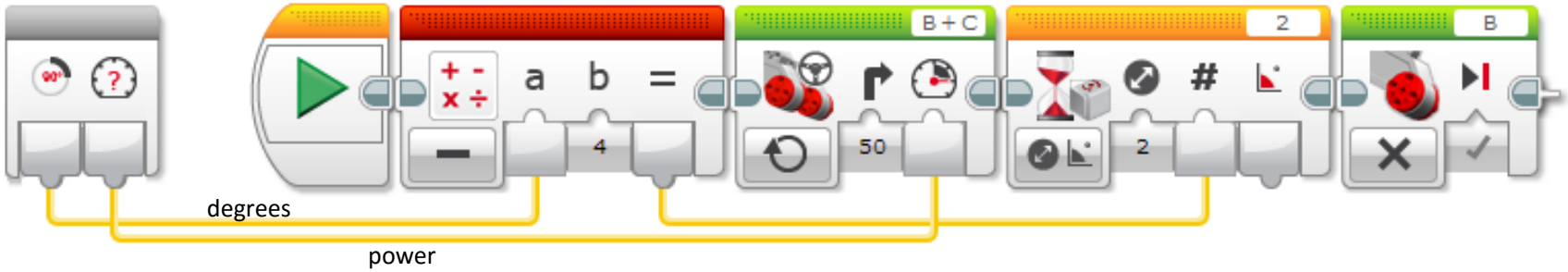
B. Add 2 inputs: one for power and one for and degrees

B



Refer to the My Blocks with Inputs & Outputs lesson if you need help setting up the My Block

Stage 3B: Wire the My Block



Connect the degrees value into the math block and the power into the move steering block

Stage 4: Using the My Block

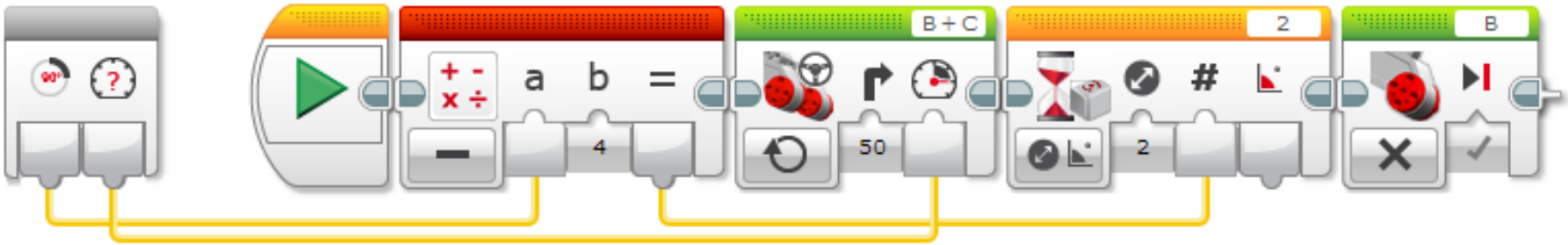
Here is our final stage, it is the same as Step 3, but converted into a my block. It has two inputs, degrees and power. Double click on the my block(s) to see inside.

Here two different my blocks that have been made turn left and right.

DO NOT SELECT the gyro calibrate blocks while making the My Block

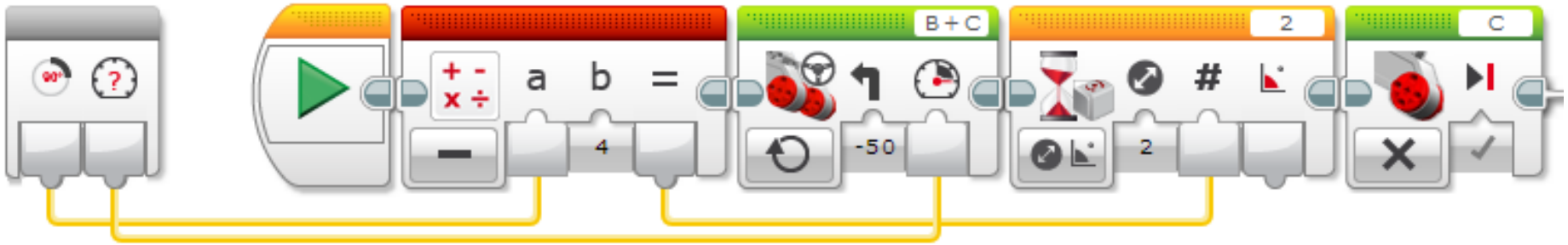
Step 4: Turn Degrees Right

This program is the same as Step 2 other than it is a my block. It was made with two numeric inputs: Power and degrees



Step 4: Turn Degrees Left

This program is the same as Step 2 other than it is a my block. This has been converted to turn left. It was made with two numeric inputs: Power and degrees



Proportional Gyro Turns
by The Construction Mavericks

- The remainder of this lesson is a contribution by The Construction Mavericks.
- This method improves over the simple overshoot correction mechanism from earlier by using proportional control
  - If you are unfamiliar with proportional control, please see the advanced lesson on proportional control before continuing.
  - The basic idea is to use the current gyro position and where it wants to point to determine how to set the motor power.
- Note from Construction Mavericks: It's not perfect, but we have had much better success with these blocks than the overshoot-corrected ones.
- Tip from Construction Mavericks: Try to set the outer loop to an infinite loop. Once the robot settles into place, pick it up and rotate it and watch it try to get back to where it wants to be.

Sample: Proportional Gyro Right Turn

1) Read the gyro value

2) Compute Error: Subtract the gyro value from the target amount of degrees to turn. Use scaling (multiply to tweak how much to adjust the power) if necessary.

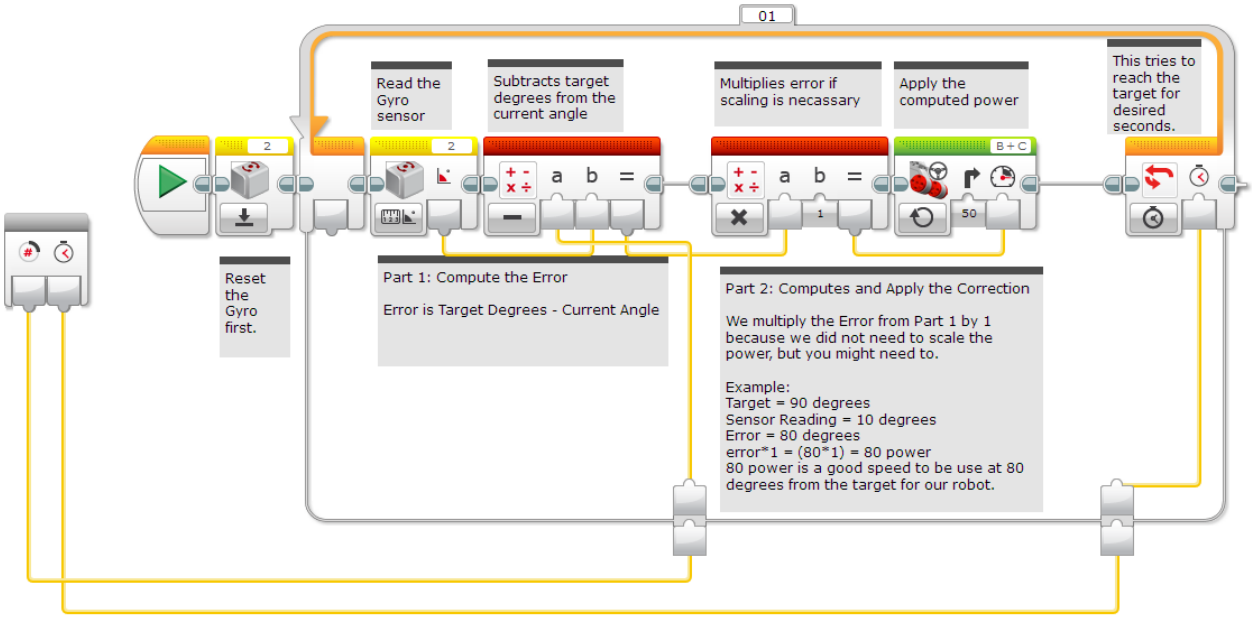
3) Apply Correction: Use data wires to wire the result from the previous step to the Move Steering block's power input, with steering 50

4) Repeat for the specified duration

Proportional Right Turn

The goal of this program is to create a proportional right pivot turn that ends after a amount of seconds. Thank You Construction Mavericks for the original code that we modified! :-)

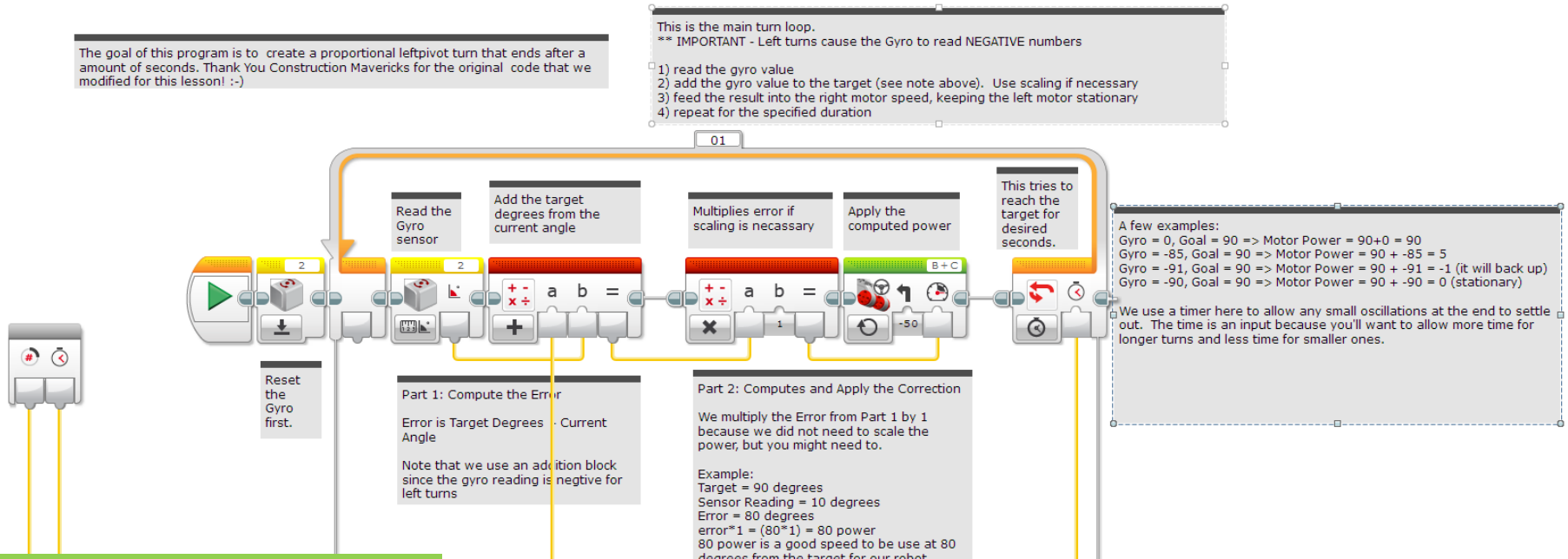
This is the main turn loop.
1) read the gyro value
2) subract the gyro value from our target. Use scaling if necessary.
3) feed the result into the left motor speed, keeping the right motor stationary
4) repeat for the specified duration



A few examples:
Gyro = 0, Goal = 90 => Motor Power = 90-0 = 90
Gyro = 85, Goal = 90 => Motor Power = 90 - 85 = 5
Gyro = 91, Goal = 90 => Motor Power = 90 - 91 = -1 (it will back up)
Gyro = 90, Goal = 90 => Motor Power = 90 - 90 = 0 (stationary)

We use a timer here to allow any small oscillations at the end to settle out. The time is an input because you'll want to allow more time for longer turns and less time for smaller ones.

Proportional Left Turn



What is going on with the math block in Left Pivot Turn? You always calculate TARGET/GOAL MINUS CURRENT VALUE. So why an Addition Math Block? When you make a Left turn, the gyro always returns negative degrees. From math, we know that adding a negative number is the same as subtracting the number. So, that is why we use the Addition Math block in a Left Gyro Turn.

Discussion

1. What is gyro lag?
   Ans. The gyro sensor's reading lags behind the true reading

2. What is the difference between the two solutions presented in this lesson?
   Ans. The first way was to reduce the amount of angle that you turn to compensate for lag. The second way was to use proportional control to continue performing your turn for a requested duration

Credits

- This tutorial was written by Sanjay Seshan and Arvind Seshan from Droids Robotics using code shared by The Construction Mavericks (http://fllmavericks.wix.com/fllmavericks)
- More lessons at www.ev3lessons.com