



SPIKE Prime Code Examples

Explorers 64955

**Owen Tang, Audrey Zhang, Louis He,
Alexi Soto-Say, Evan Shen, Nick Wang**

This document provides a detailed description of the SPIKE Prime code developed by the Explorers FLL Team (64955) and used during the 2024 Arizona FLL State Championship.



Contact US!

Acceleration Deceleration

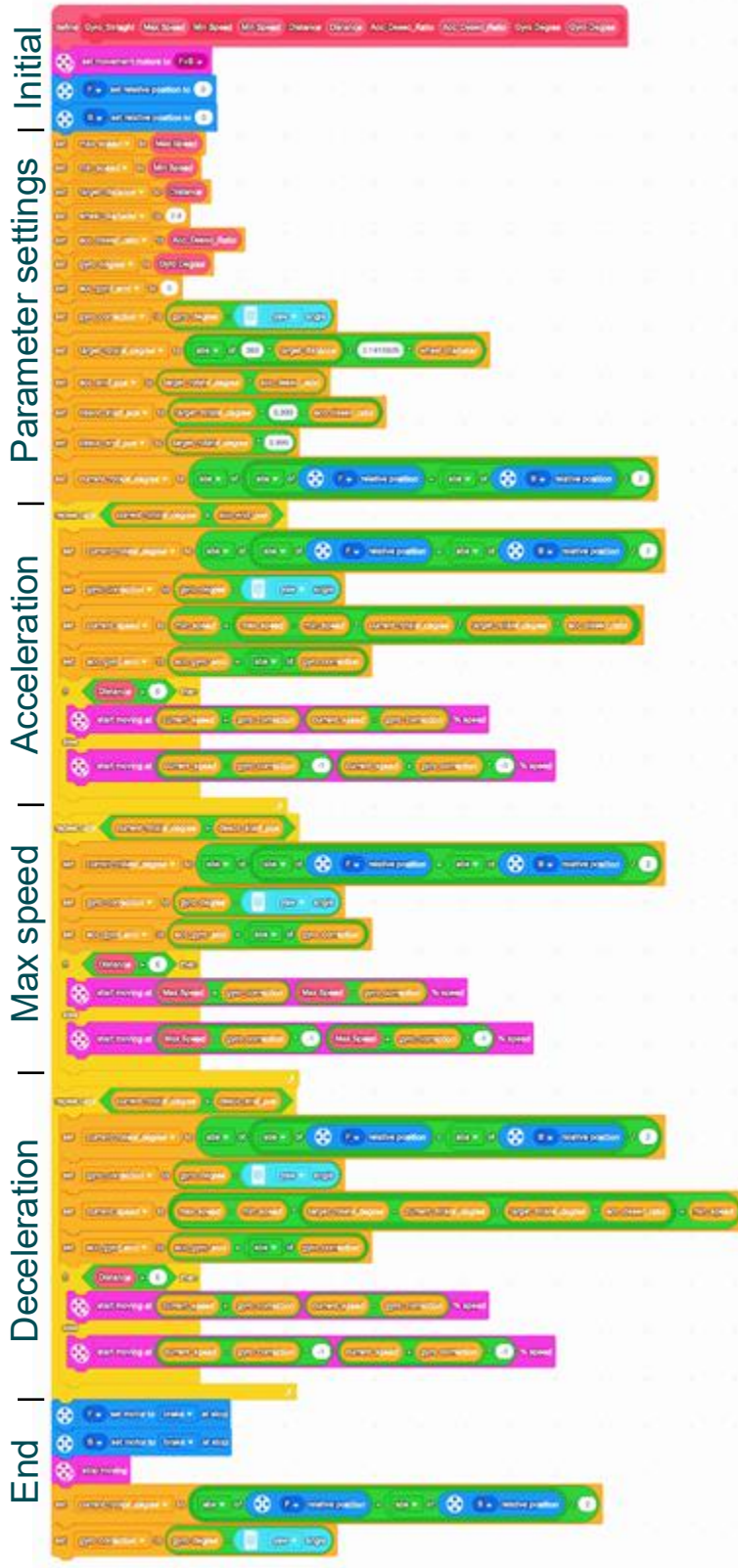
Gyro Straight

Introduction

The block ensures accurate straight-line movement in both forward and backward directions by utilizing acceleration and deceleration techniques.

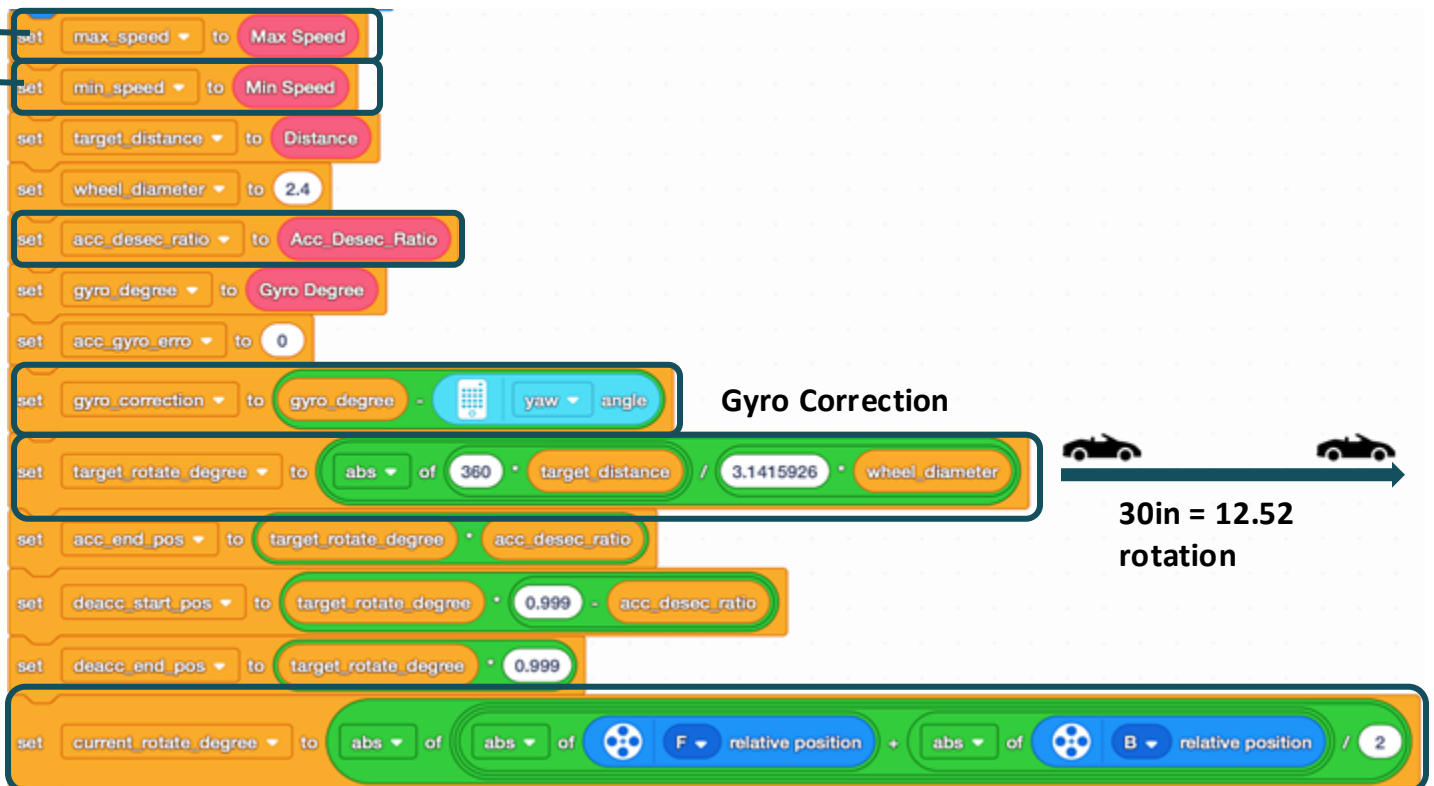
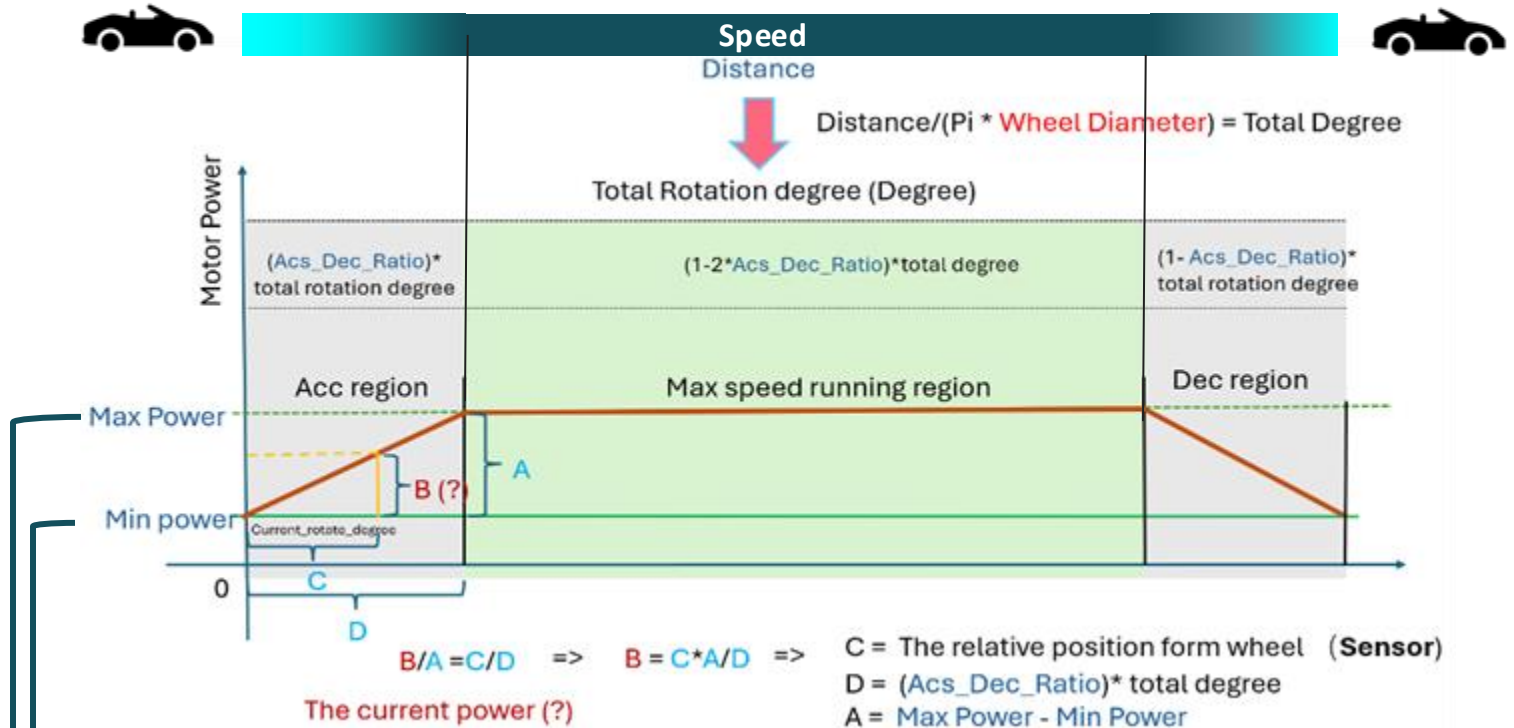
Key features

- Highly customizable to control the minimum and maximum speed, as well as the acceleration and deceleration ratios.
- Adjusting acceleration and deceleration increases friction, making movement distances more accurate.
- Low speed at stop points ensures the next turn is more precise.
- We use a Gyro Sensor to correct the robot's direction and keep it straight in real-time
- Build-in gyro error accumulation calculation make it easy to evaluate how straight the robot is going.



Acceleration Deceleration Gyro Straight

Variables and Parameters Settings

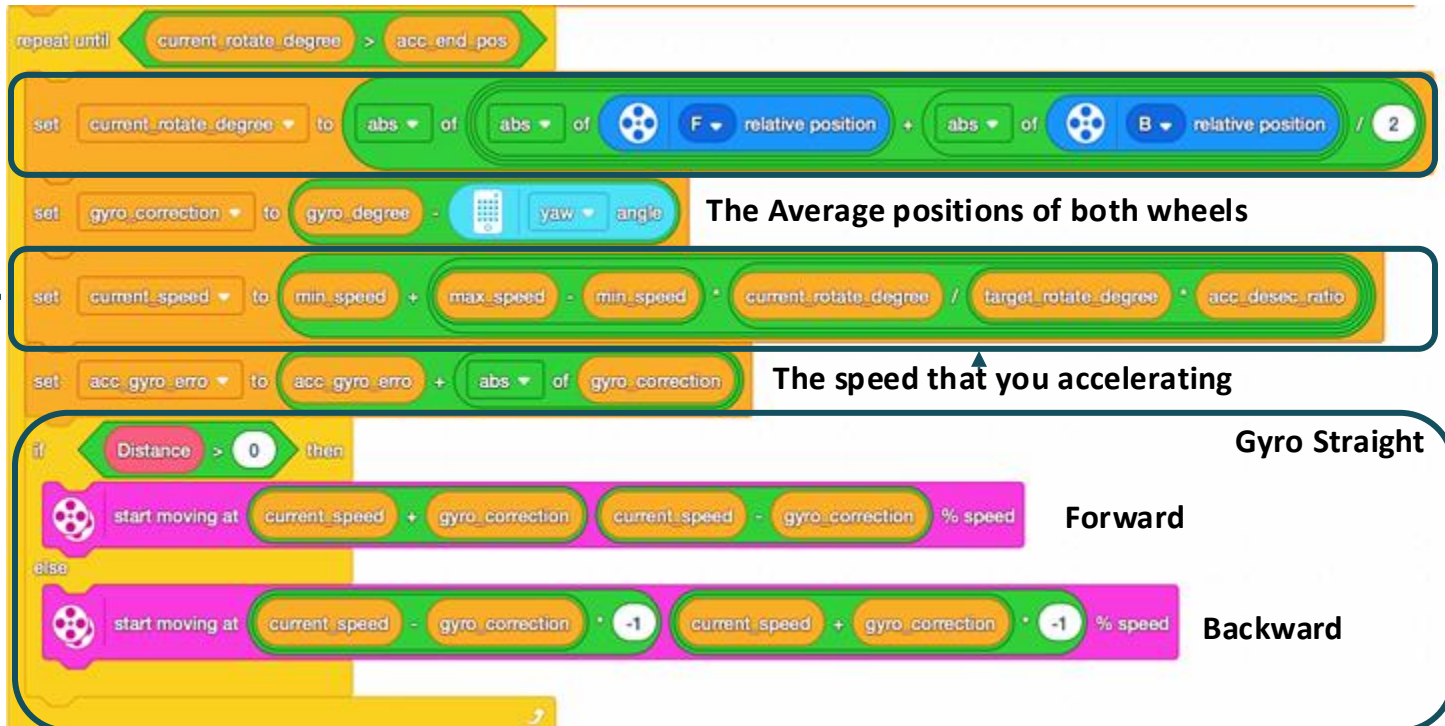


Wheel 1 + Wheel 2 = How many rotations has the wheels done?

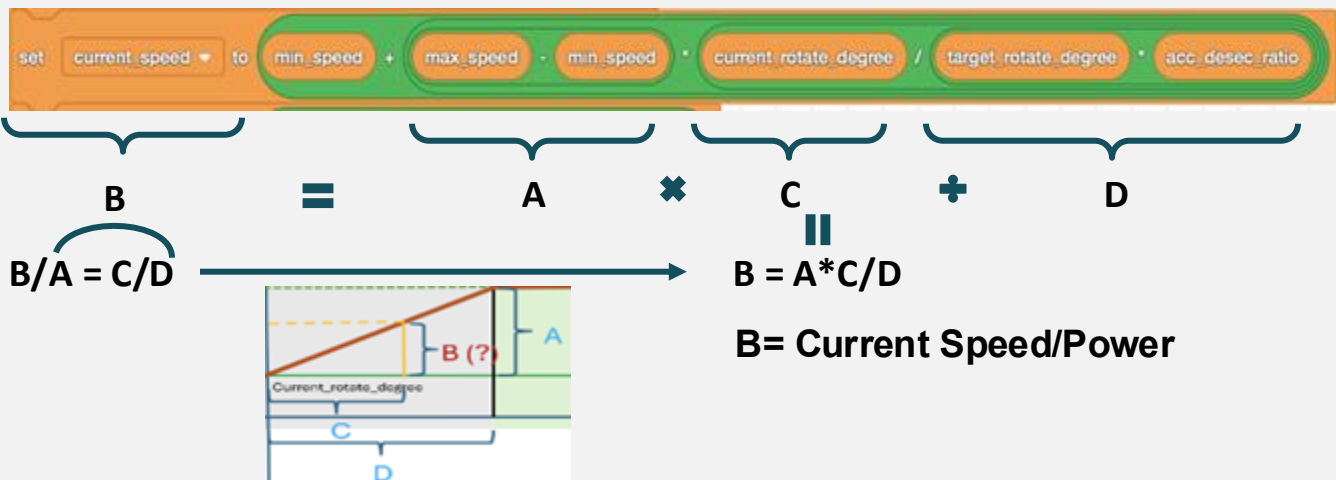
2

Acceleration Deceleration Gyro Straight

Acceleration Phase

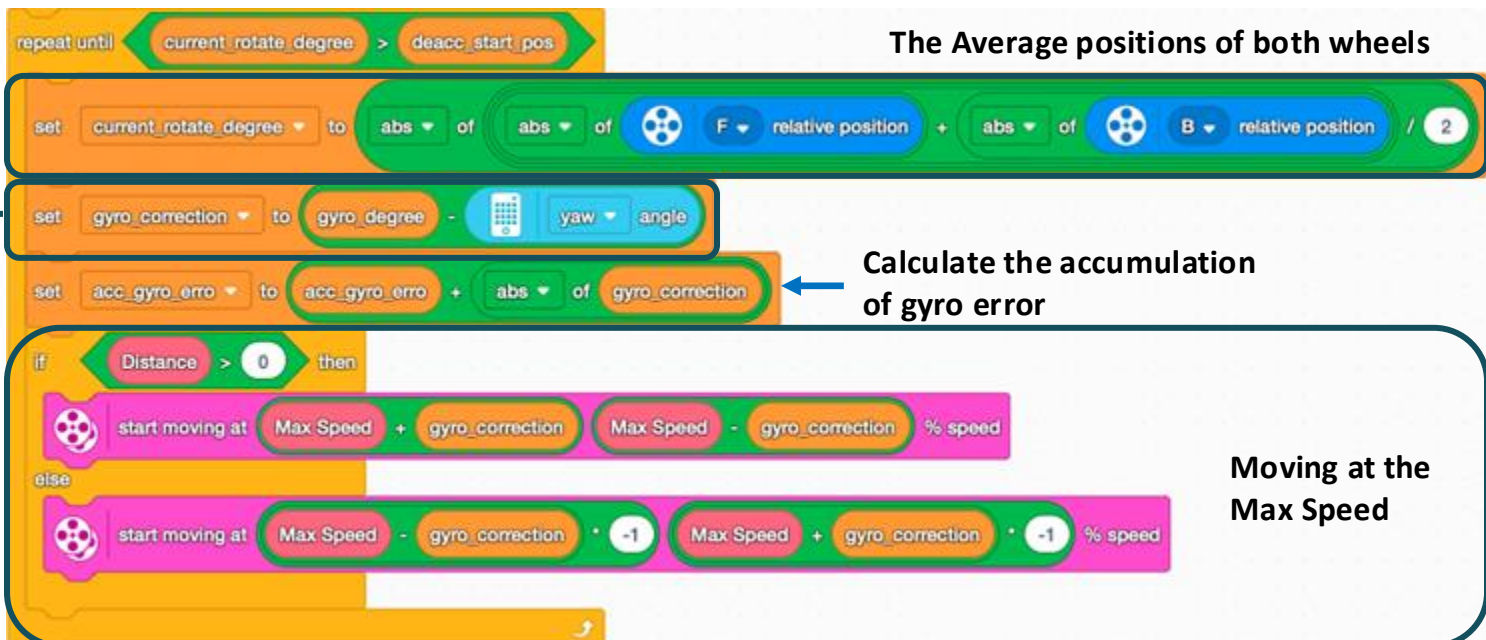
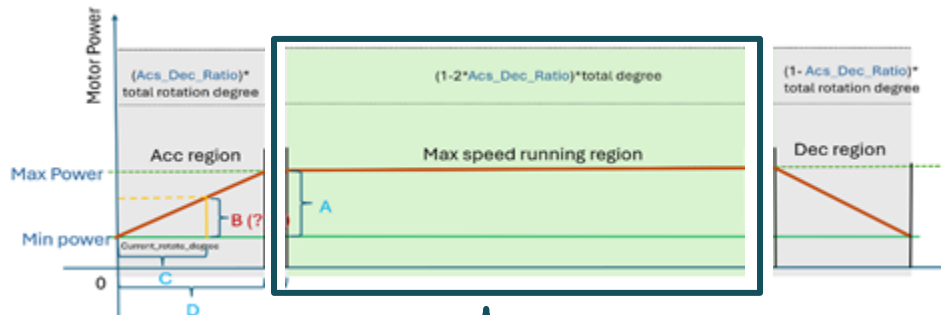


How to calculate the current speed?



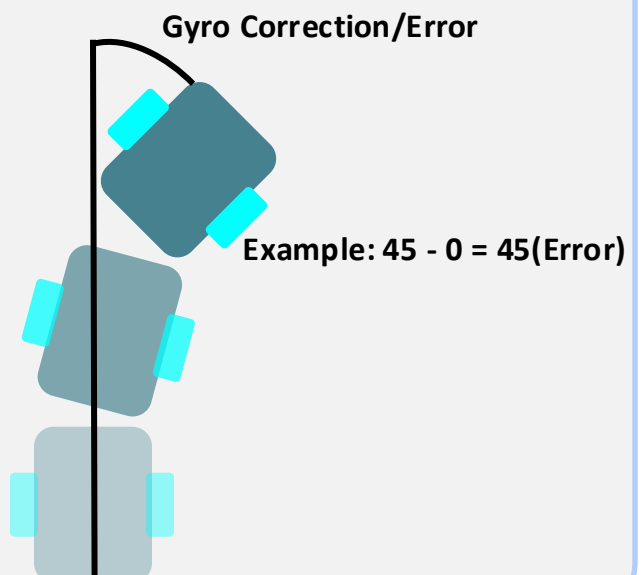
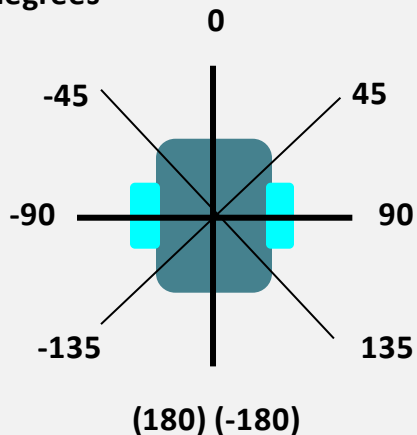
Acceleration Deceleration Gyro Straight

Max Speed Phase



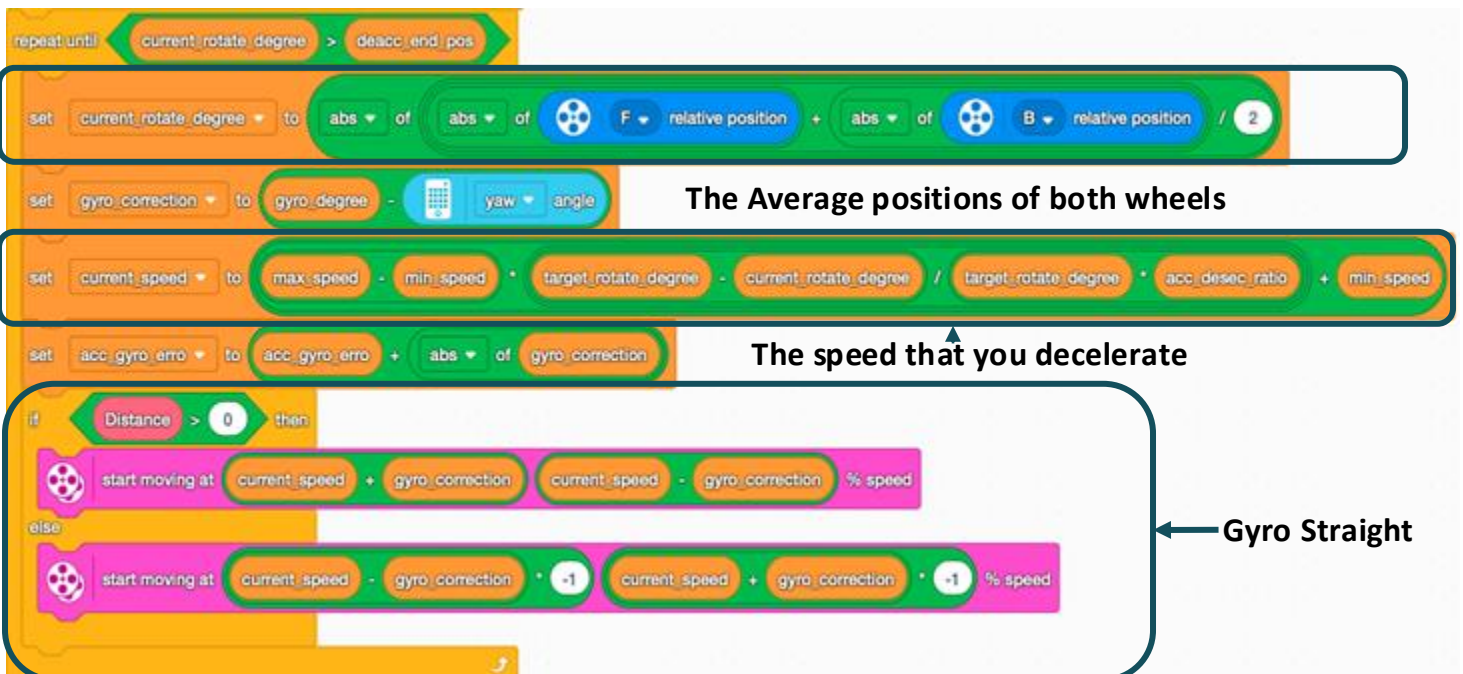
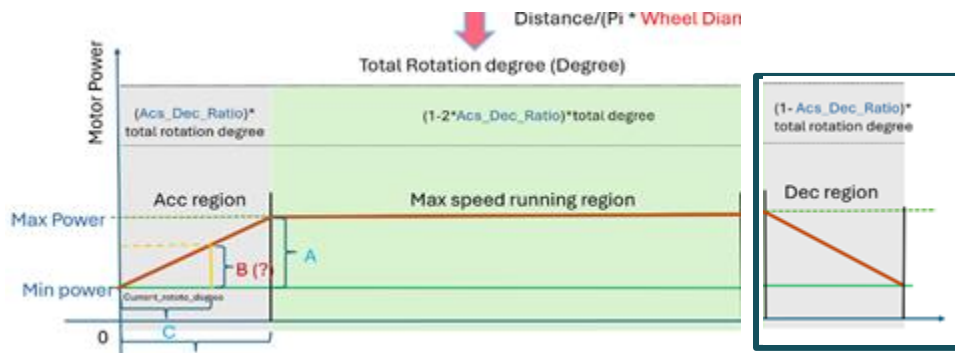
How to calculate gyro error?

Gyro Degree (Target Angle) = 0 degrees



Acceleration Deceleration Gyro Straight

Deceleration Phase



$$B = \frac{A}{B/A = C/D} \times C + D \parallel B = C/D * A$$

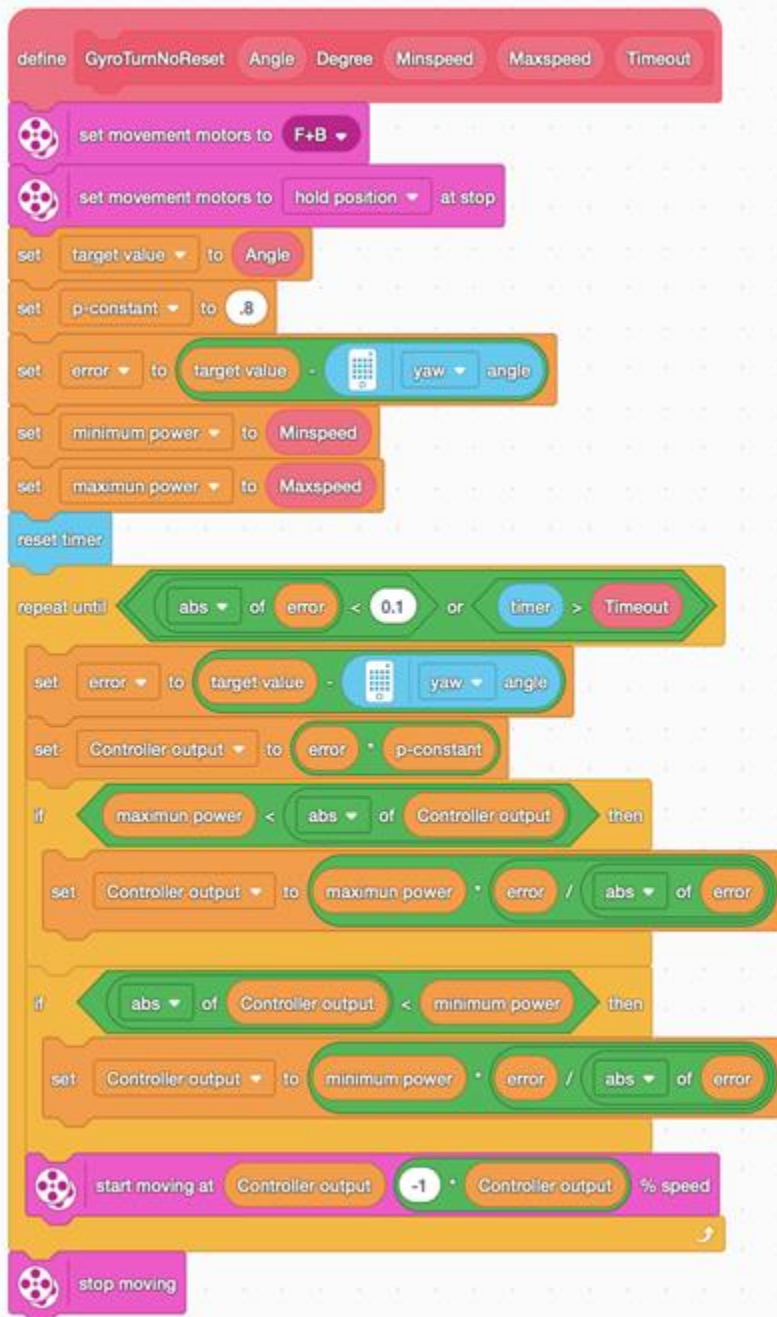
Gyro Pivot Turn

Introduction

The block ensures accurate pivot turn by using gyro sensor and utilizing proportionally turn algorithm.

Key features

- Highly customizable to control the minimum and maximum speed, as well as the deceleration ratios.
- Turn speed is proportionally to the error to maximize the speed and to prevent slipping.
- We also have a timeout setting to save time when stuck.

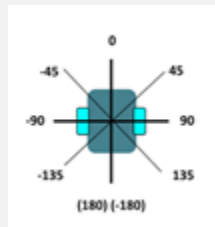
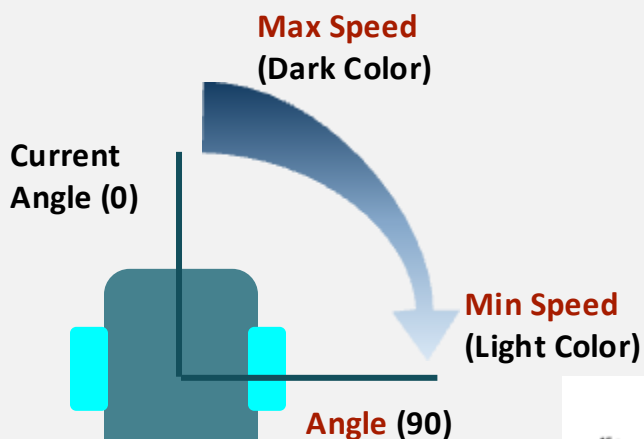


Gyro Pivot Turn

Parameters and Variables

define GyroTurnNoReset Angle Degree Minspeed Maxspeed Timeout

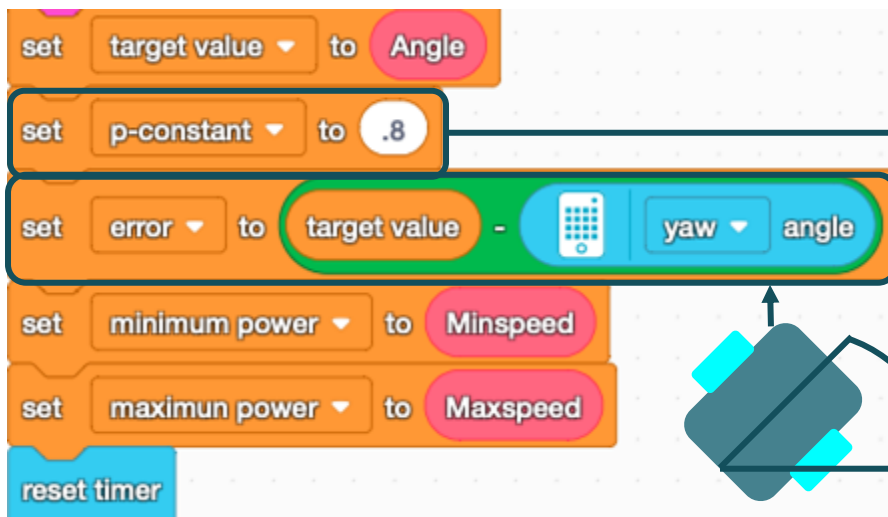
How does proportional turn work?



Get Stuck?

1...2...3...4...5...MOVE

After a set time it will move to for next move (Ex: 5sec)

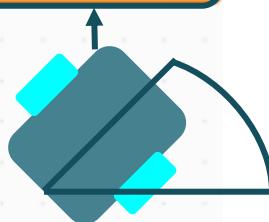


This speed changes based on your robot:

Bigger (Heavier): Larger

Smaller (Lighter): Less

Error (Target angle - Current angle)



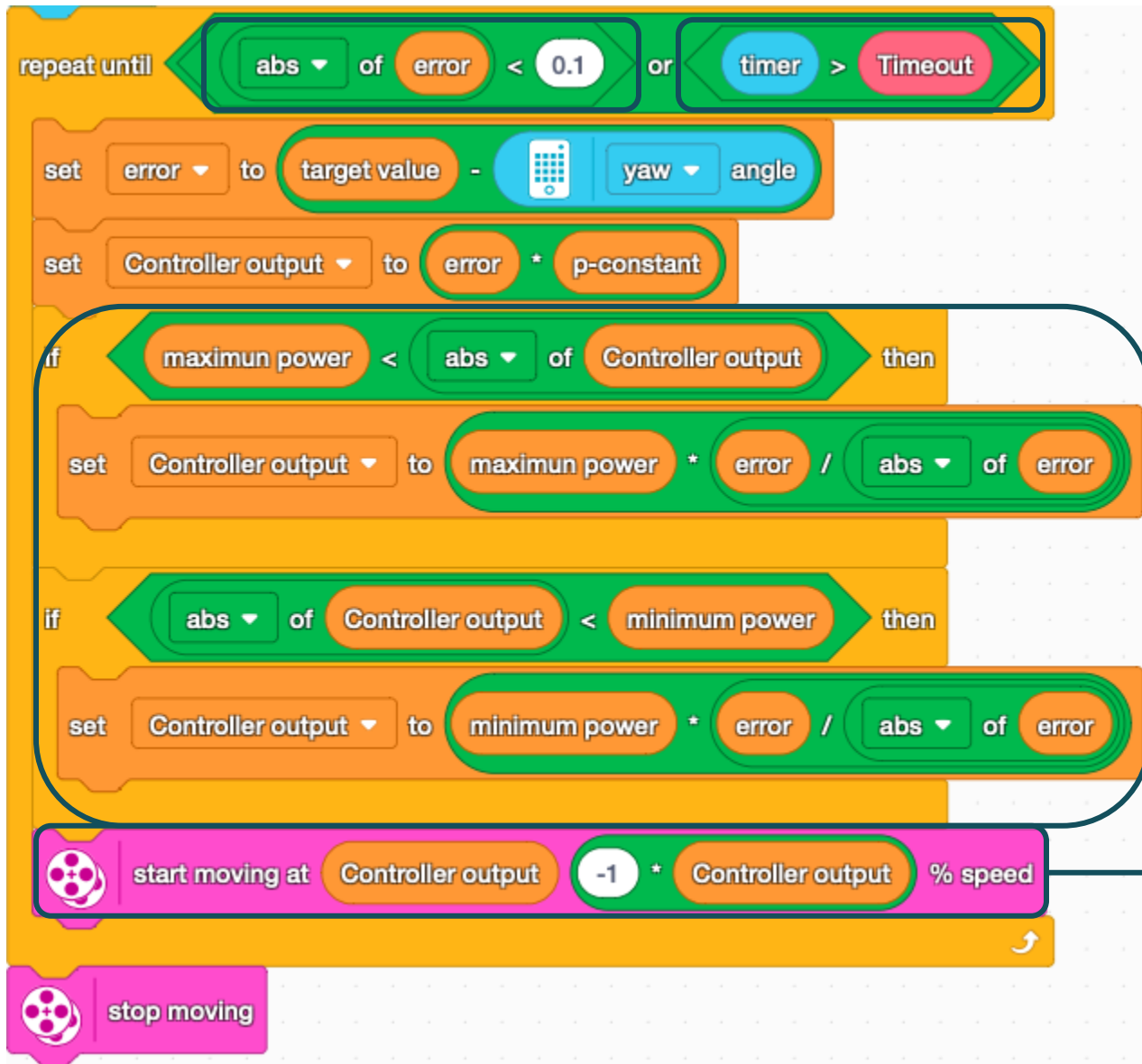
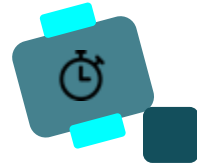
Example: $P\text{-constant}(0.8) * \text{Error} (\text{Target angle} - \text{Current angle}) = \text{Speed}$

Gyro Pivot Turn

Turning and Max, Min Speed Control

Sometimes it will skip the 0 so it is: < 0.1

Will skip to next code after certain time



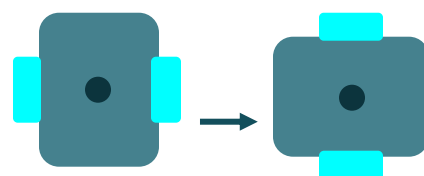
If it is **bigger** than the **Max Speed**?

If it is **smaller** than the **Min Speed**?

Turning **Different Directions**



Turn at the **center of the wheels**



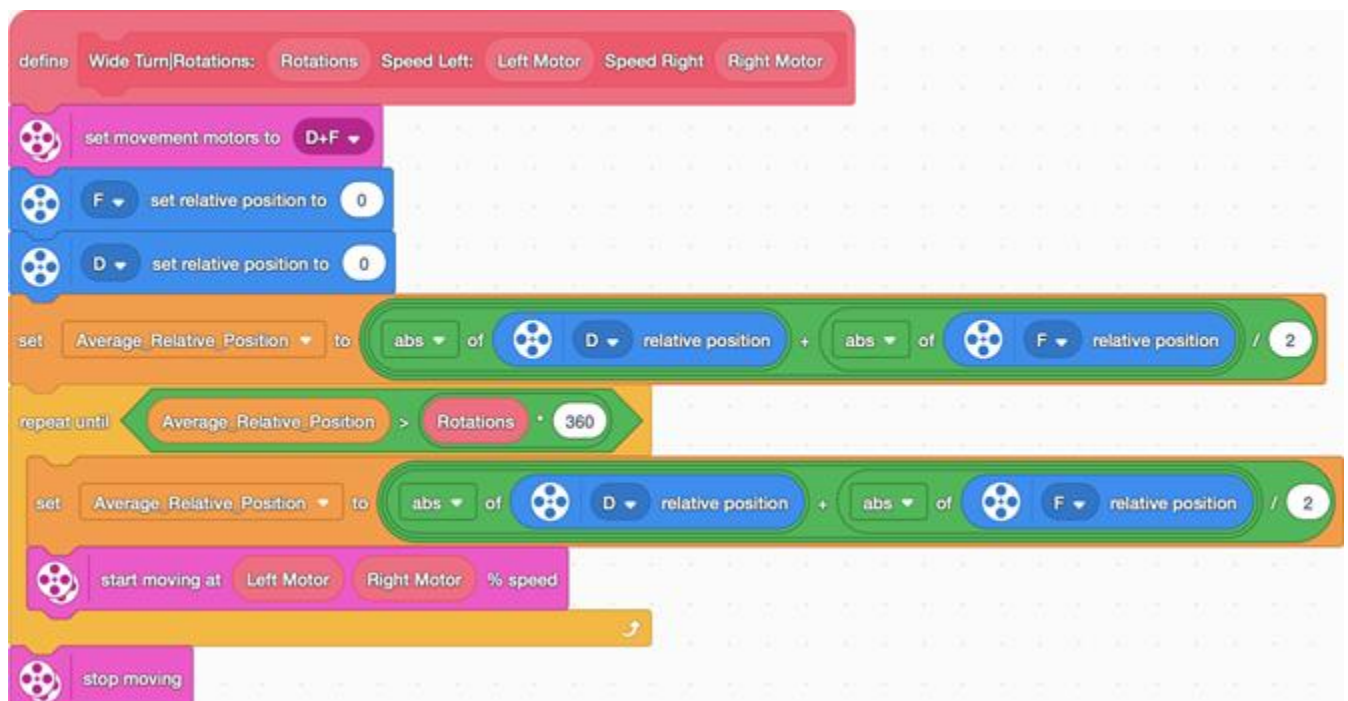
Wide Turn

Introduction

The block makes robot do a wide turn by let the wheels rotate in different speeds but the same direction.

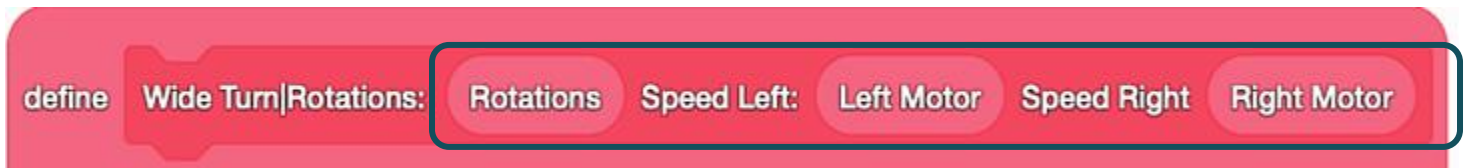
Key features

- No stop or deceleration before turning, therefore the transition is more smooth and quick than a regular pivot turn.
- Wide turn follows a curve path and easy to combine with proportion algorithm.
- The turn accuracy in degree is less than pivot turn.
- The block usually was used in the situation when turning doesn't need high accuracy but higher speeds.



Wide Turn

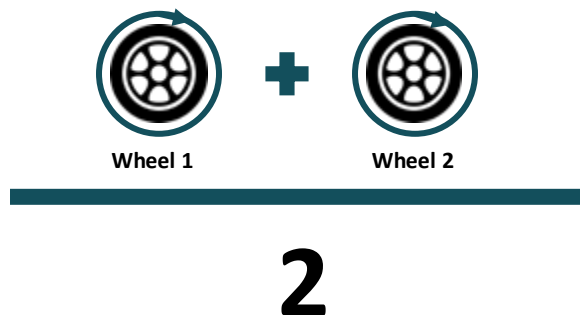
CODE Details



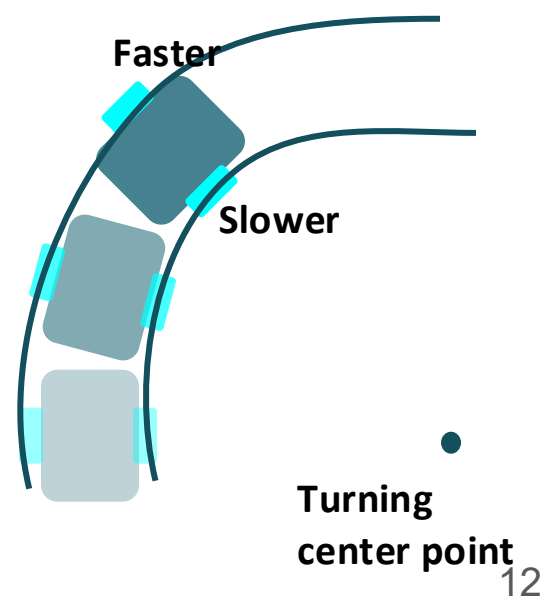
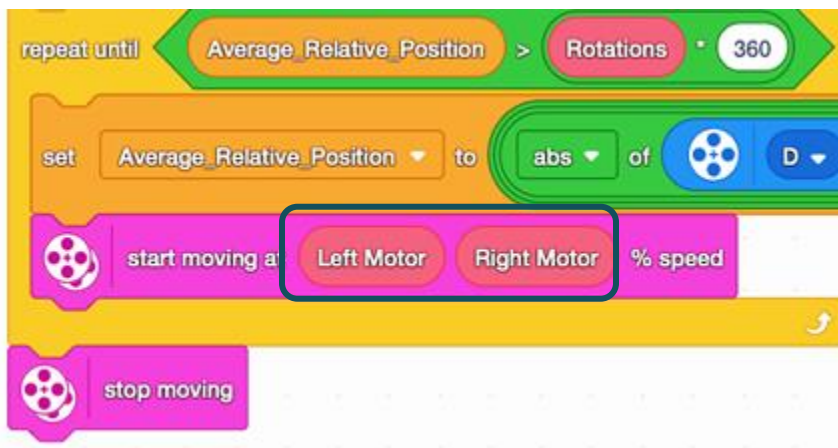
How far you want to go  ? Left motor speed Right motor speed



Finds the Average of the degree of the 2 wheels.
(This is also used in the Gyro Straight Myblock)



Current wheel degree > Converting Rotations to Degree



Line Following

CODE Details

Speed of both wheels Ex: 40:60, 50:40

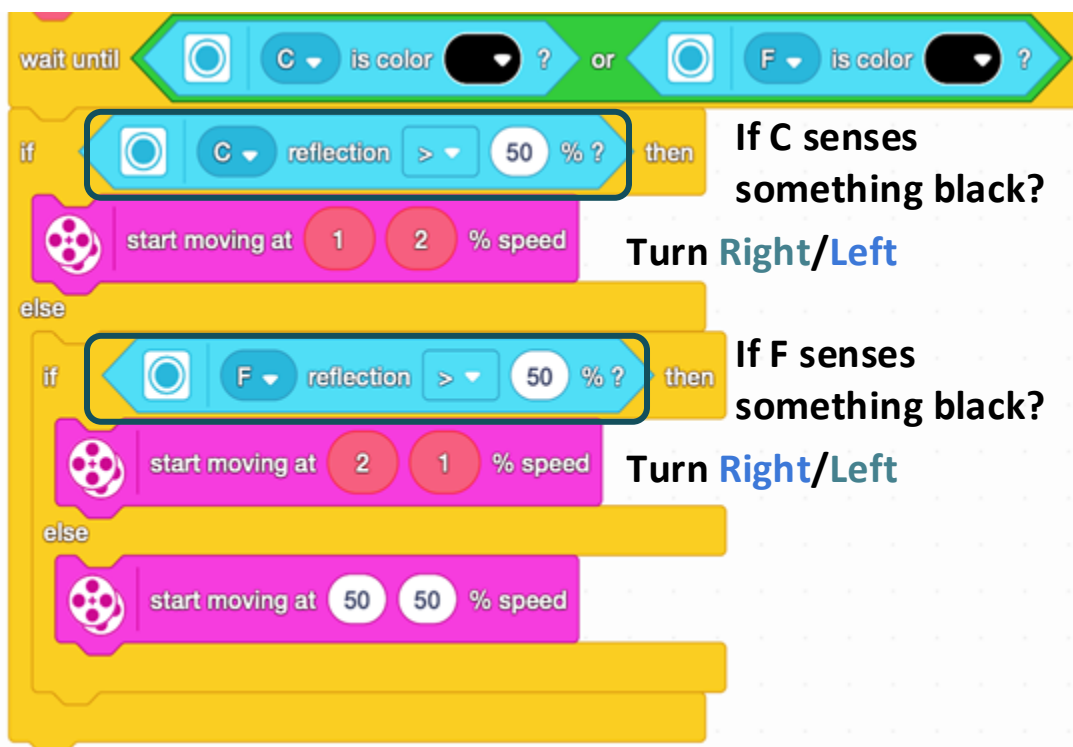
THE REAL GOAL IS TO BE IN
THE MIDDLE OF THE LINE
(50%)

define line following | Speed Ratio

1

2

Black(0%) White (100%)

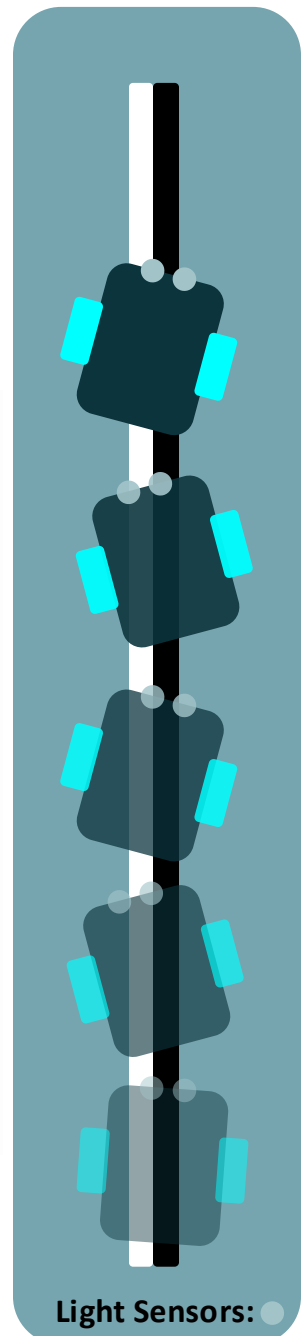


If C senses
something black?

Turn Right/Left

If F senses
something black?

Turn Right/Left



Light Sensors: ●

Line Following

Introduction

A line-following algorithm enables a robot to follow a line on the ground, typically a black line on a white surface. The robot uses one or two color or light sensors, to detect the line and adjust its path accordingly. The basic idea is to keep the sensor on the edge of the line, making small adjustments to stay on course. There are several methods to achieve this, including:

1. Two-State Line Following: The robot moves left and right to stay on the edge of the line.

