

Lego EV3 Robotics

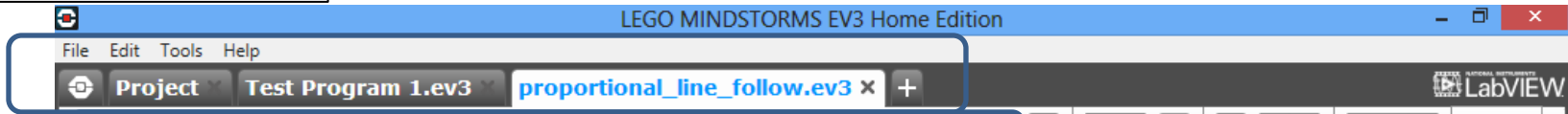
Georgia Institute of Technology

ICE Programs

Mr. Michaud

User Interface Windows

Program Tabs

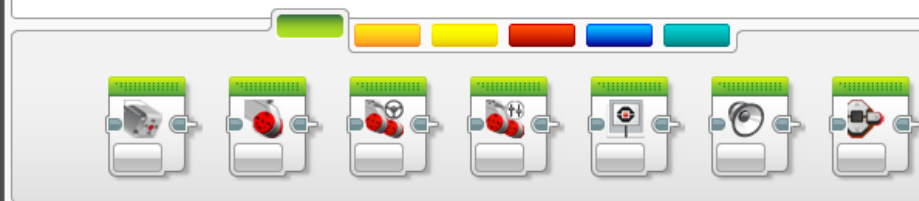


Programming Area:
Where the blocks are
assembled to create program.

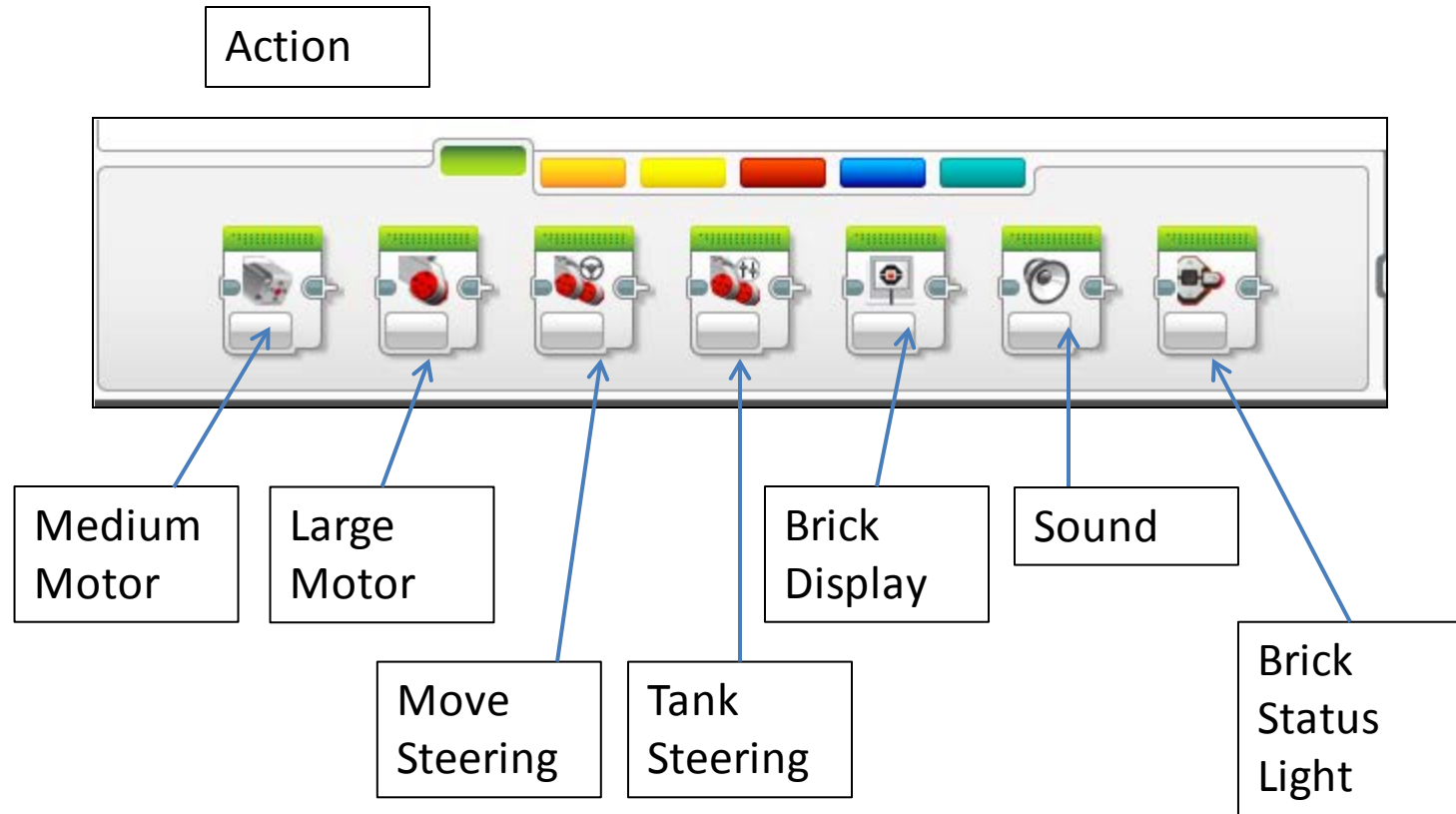
Active code tabs
within program

Block selection Window

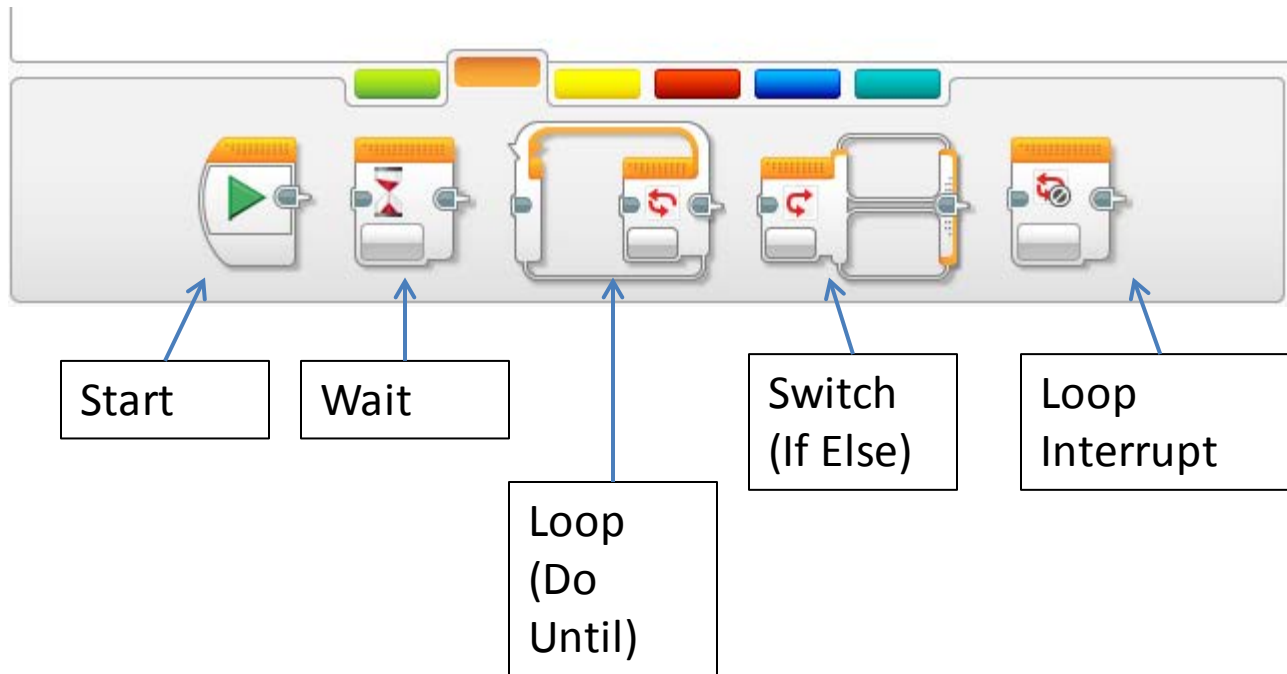
Brick
Communication
Window



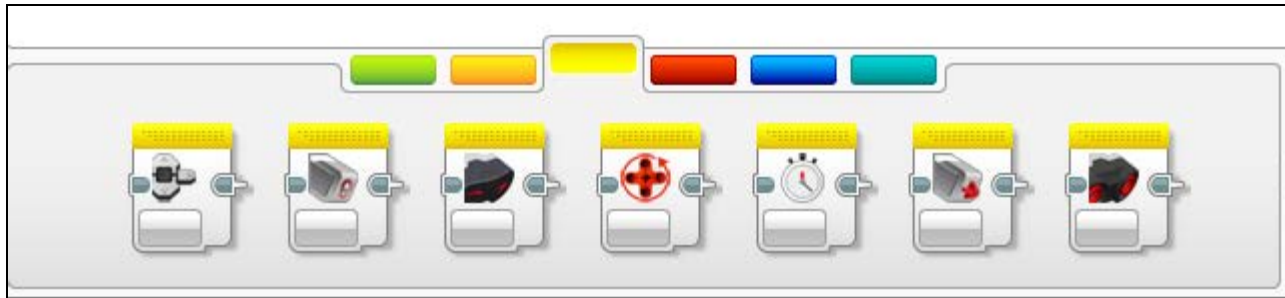
Action Blocks



Flow Control Blocks



Raw Sensor Data Blocks



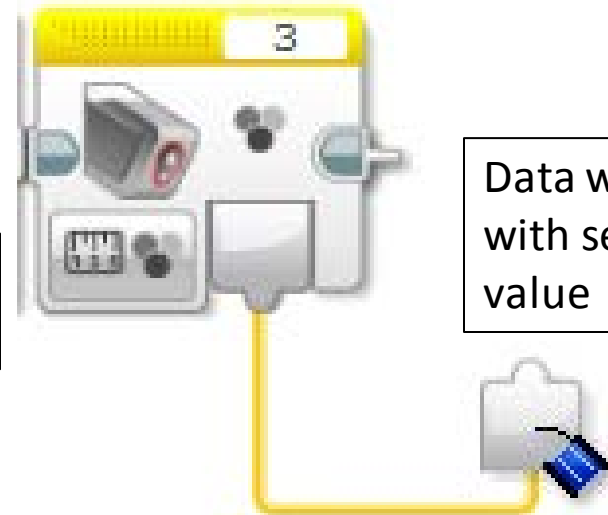
These blocks provide data output of the value of the indicated sensor.

Sensor Type

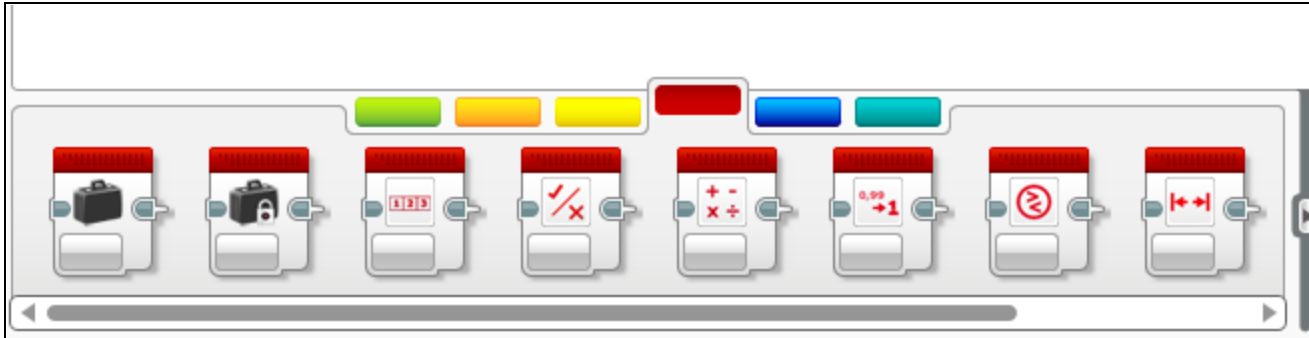
Sensor Port

Measurement Type

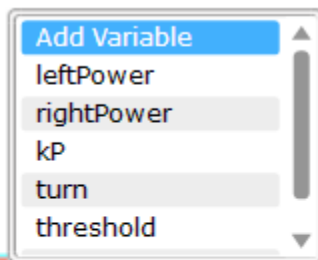
Data wire with sensor value



Data Operations Blocks



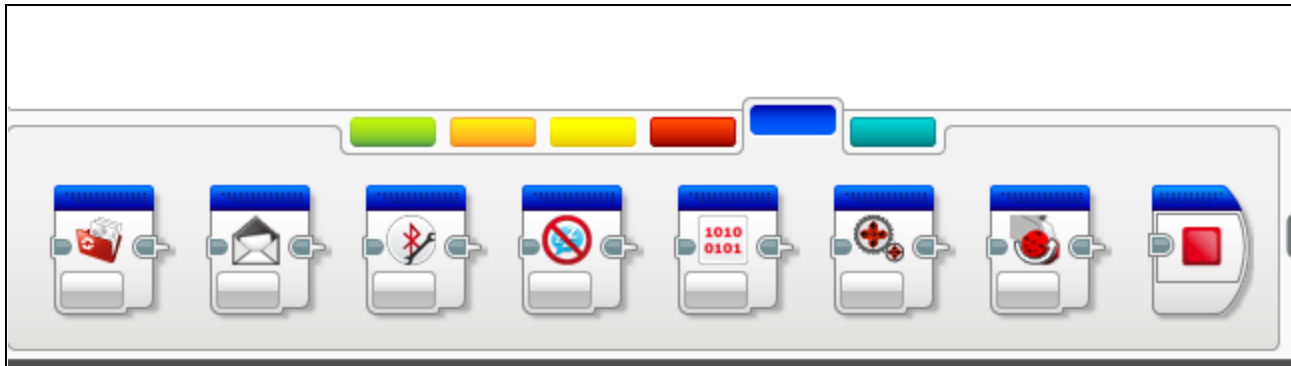
These blocks control variables, logic, math, random, text, and range operations.



To create a variable, drag a variable block to the work area and select "Add Variable"

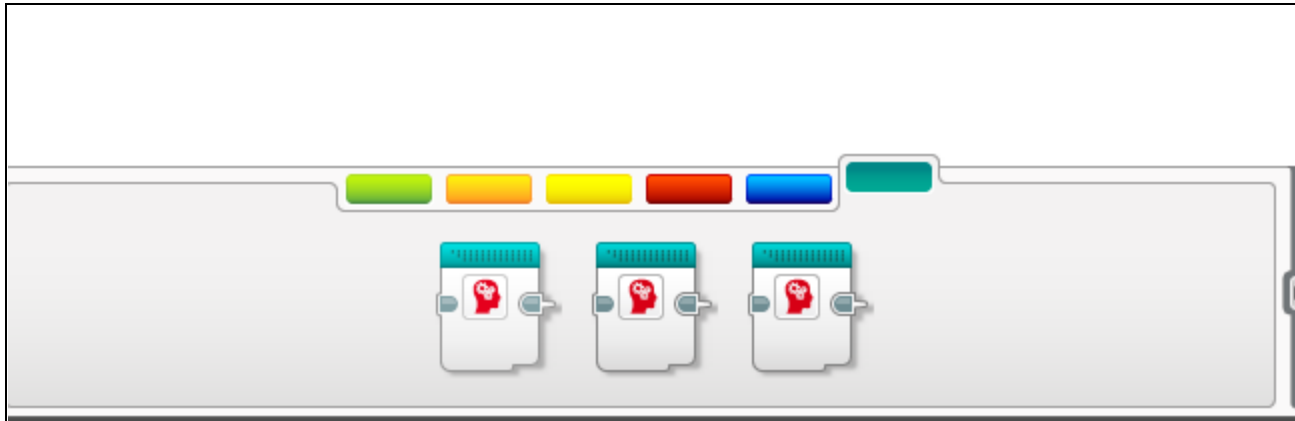


Advanced Blocks



Blocks for file access, communication, bluetooth, raw data, and advanced motor functions.

MyBlocks



These are blocks that you create. Similar to functions or procedures.

Brick Status Window

The image displays three stacked panels from the Brick Status Window, each with a vertical selection tab on the left and a control panel on the right. The top panel is in Brick Info Mode, the middle in Port Info Mode, and the bottom in Available Bricks mode. The right panel contains three buttons: a download icon, a play icon, and a combined download-and-play icon.

Selection Tabs (left side of the top panel):

- Brick Info Mode (selected)
- Port Info Mode
- Available Bricks

Brick Info Mode (top panel):

- EV3_02
- Firmware: V1.01H
- Connection Type: [Progress Bar]

Port Info Mode (middle panel):

A	B	C	D
	0	0	
1 0	2	3 0	4 8.6

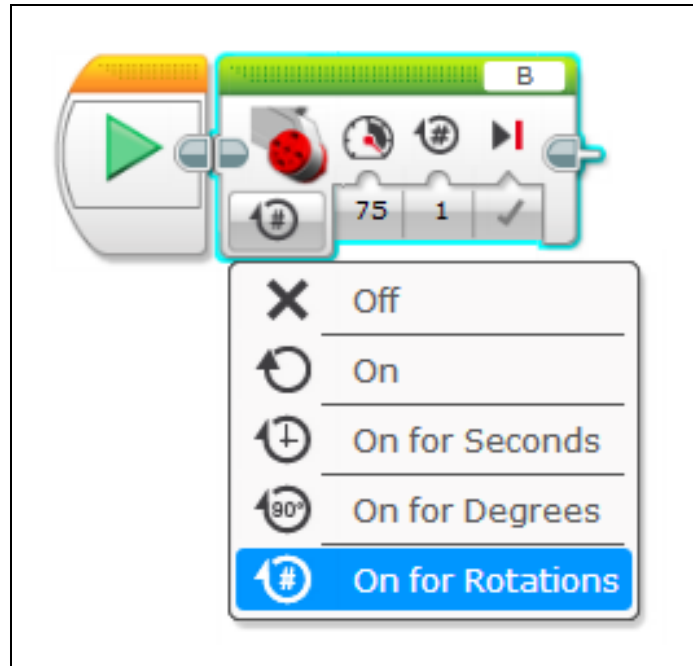
Available Bricks (bottom panel):

EV3_02	USB	Bluetooth	WiFi
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

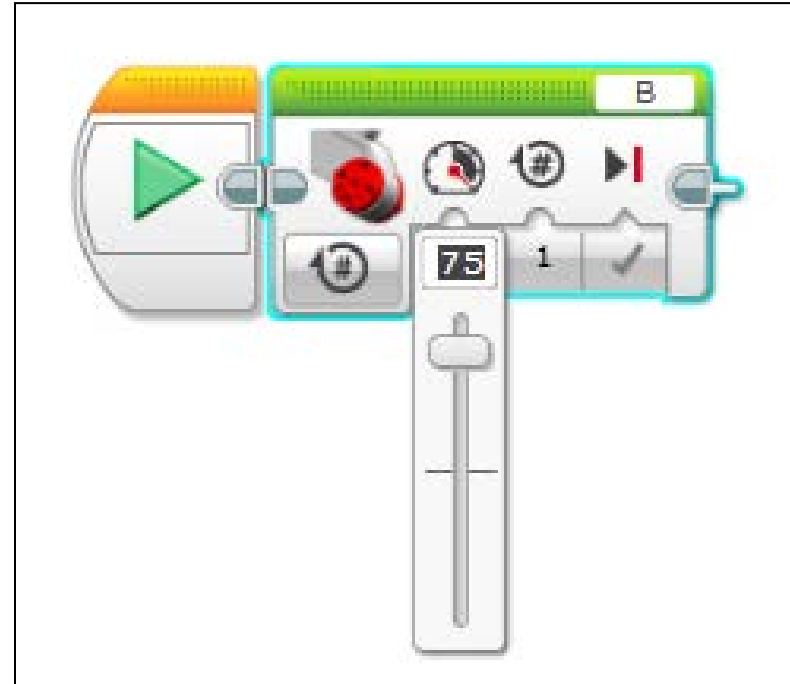
Actions (right side of the panels):

- Download Program (top button)
- Download and play Program (middle button)
- Download and play selected blocks (bottom button)

Single Motor Block Parameters

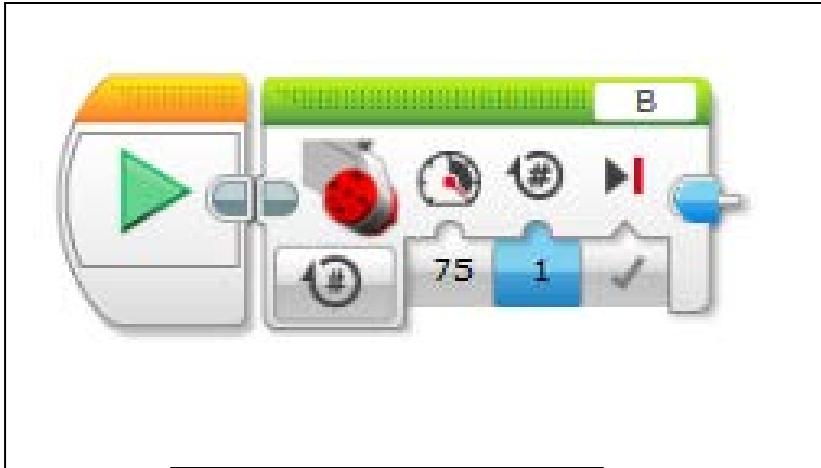


Motor Block Mode
Selection:
-Seconds
-Degrees
-Rotations

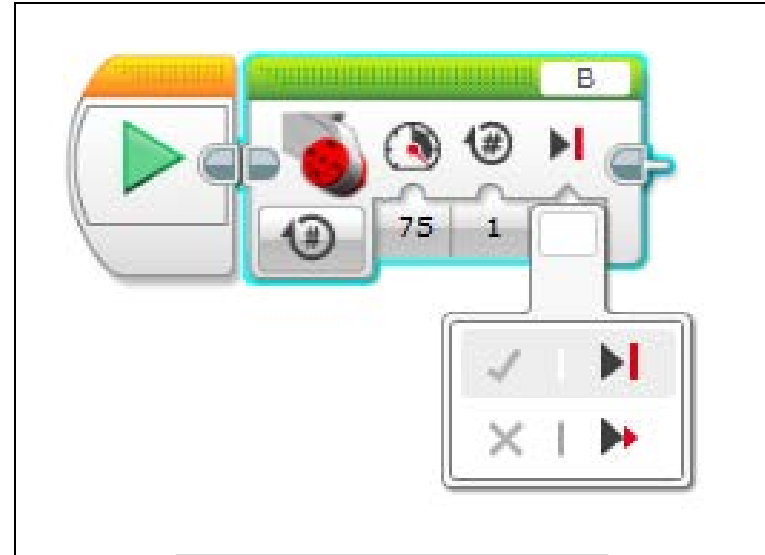


Motor Block
Power Selection

Single Motor Block Parameters

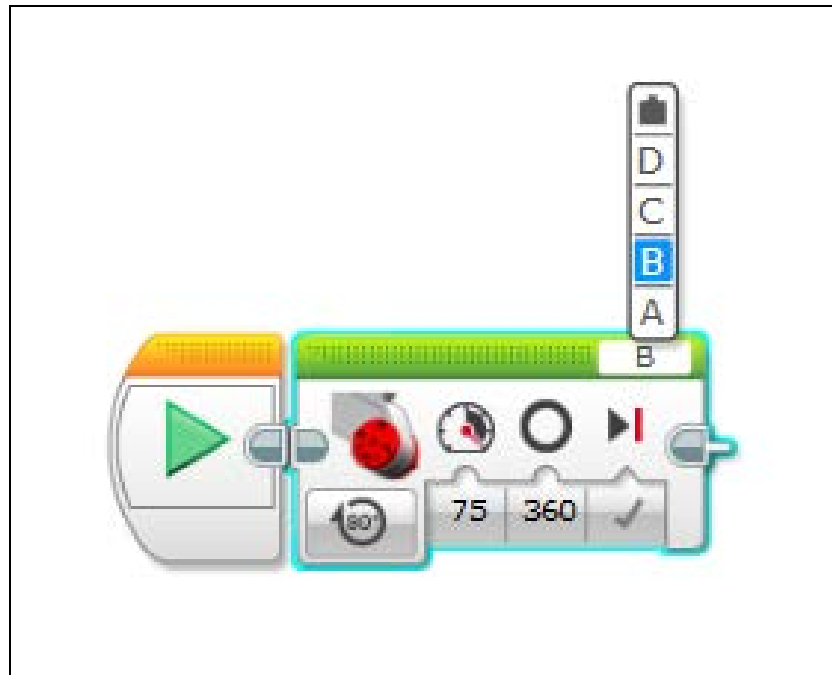


Select
Number of
Rotations
(Degrees, Time)



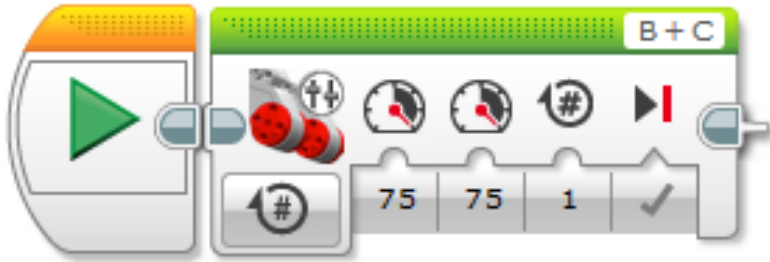
Select
Brake or Coast
Mode

Single Motor Block Parameters



Select
Motor Port

Dual Motor Blocks



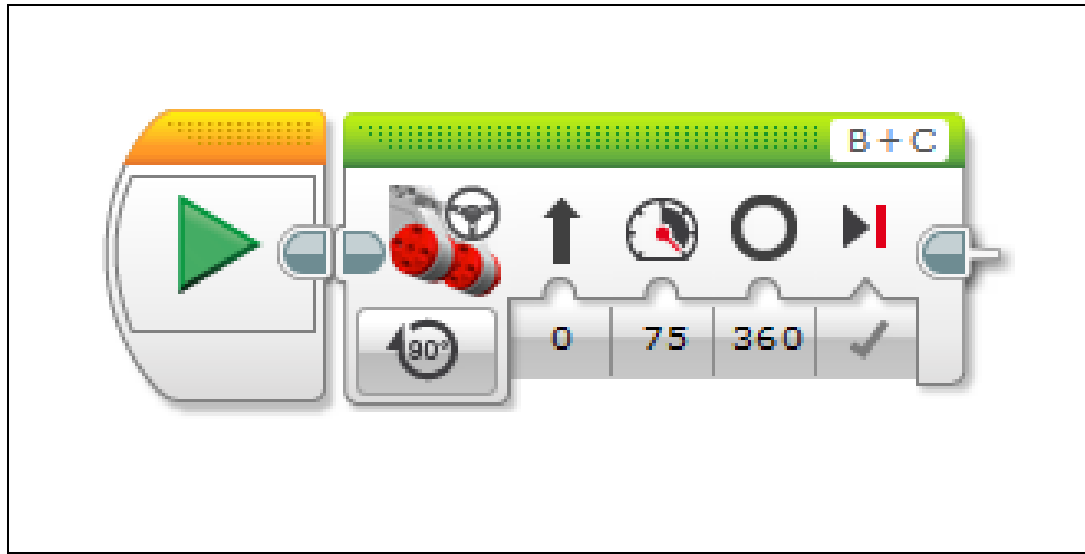
Tank Drive Settings (Motor power for two motors set separately).

Move Steering Settings. (Motors set according to arc (steering left and right)).

Dual motor blocks use PID steering algorithms to keep two drive motors in sync during program.

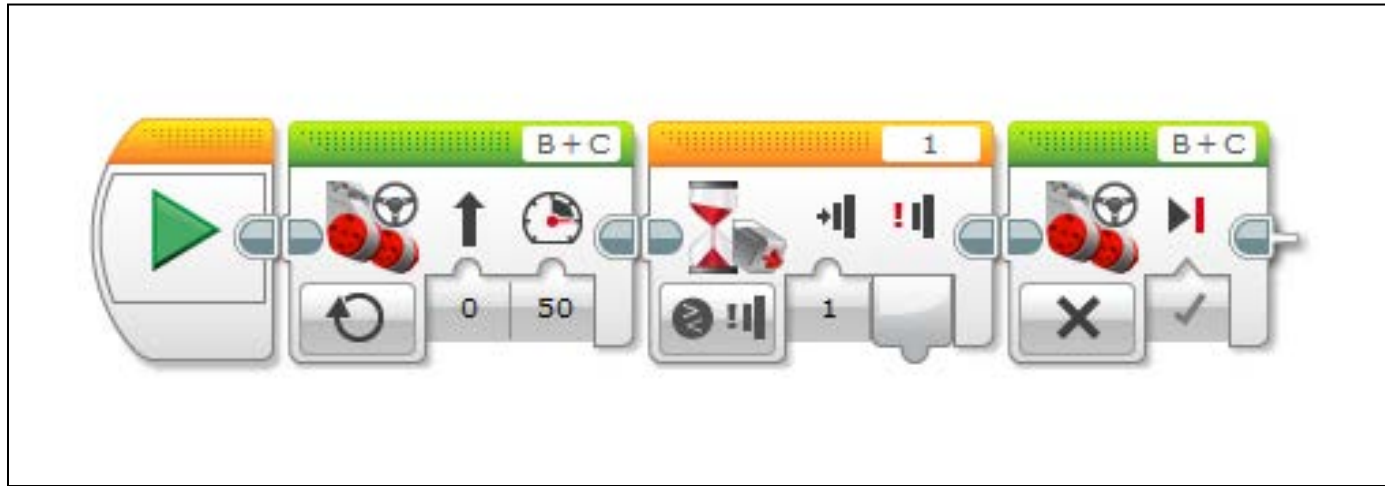


Program Samples: Forward for Degrees



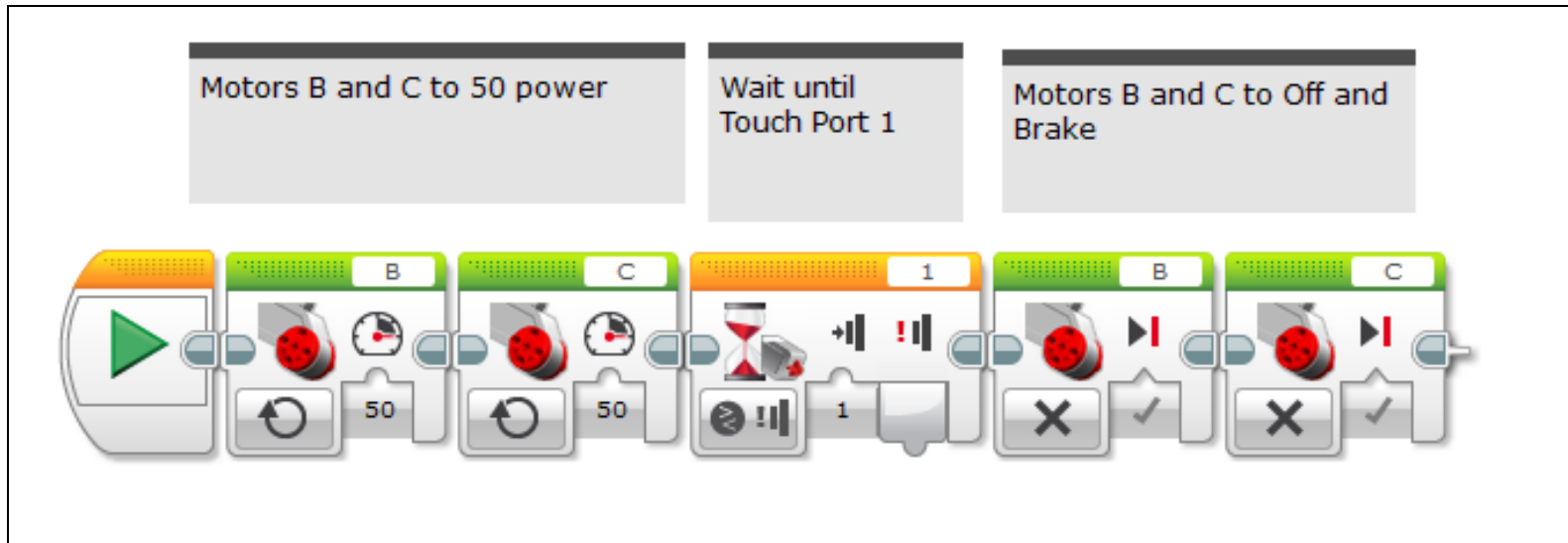
```
task main() {  
    steeringMotors(B, C, 0, 75, 360, brake);  
}
```

Program Samples: Forward and Wait for Touch



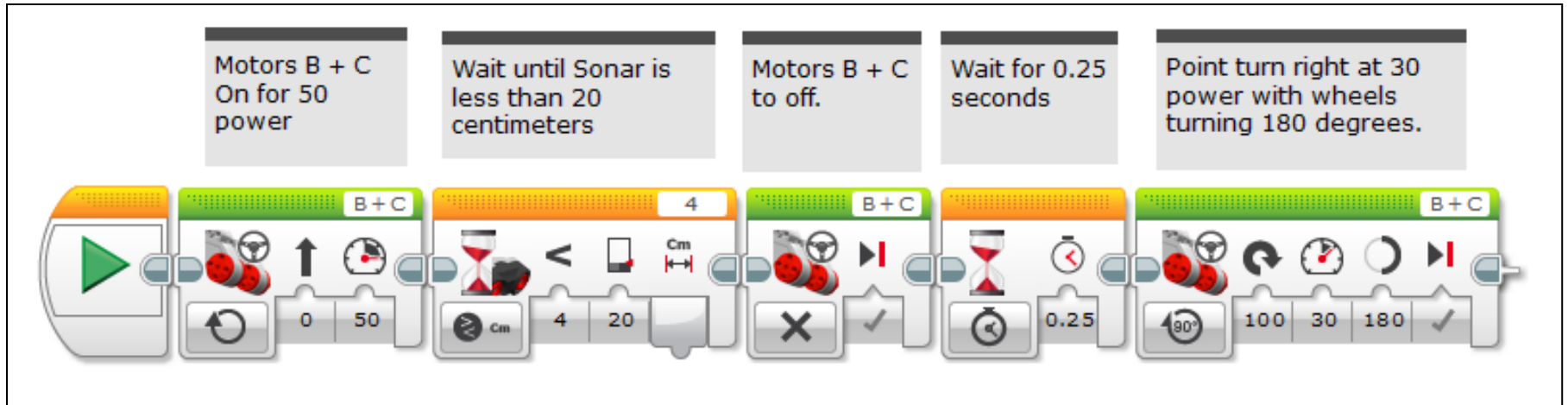
```
task main() {  
    steeringMotors(B, C, 0, 50);  
    while (SensorValue[touch] == 0) {  
    }  
    steeringMotors(B,C,0,0, brake);  
}
```

Program Samples: Forward and Wait for Touch Separate Motors

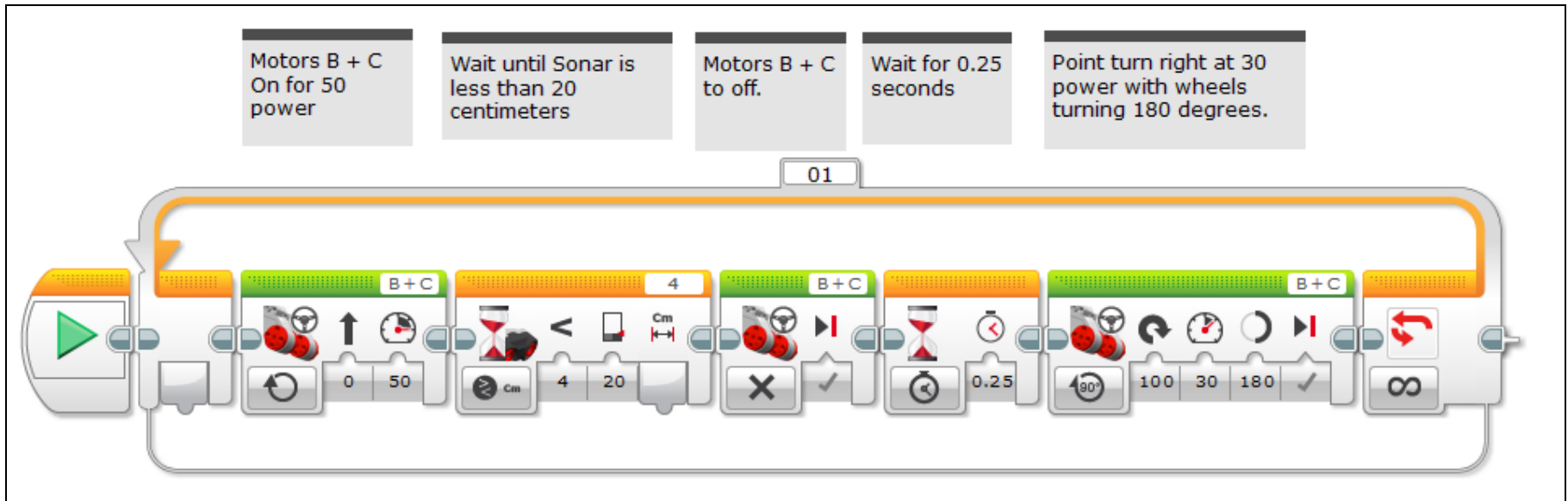


```
task main() {  
  motor[B]=50;  
  motor[C]=50;  
  while (SensorValue[touch] == 0) {  
  }  
  motor[B] = 0;  
  motor[C] = 0;  
}
```


Program Samples: Forward until Sonar and Turn



Programs: Forward and Turn with loop



Infinity symbol at end of loop means infinite loop.

Other Types of Loops

Motors B + C On for 50 power

Wait until Sonar is less than 20 centimeters

Motors B + C to off.

Wait for 0.25 seconds

Point turn right at 30 power with wheels turning 180 degrees.

01

B+C

4

B+C

B+C

0 50

4 20

0.25

100 30 180

∞

- Brick Buttons ▶
- Color Sensor ▶
- Infrared Sensor ▶
- Motor Rotation ▶
- Timer ▶
- Touch Sensor ▶
- Ultrasonic Sensor ▶
- Messaging ▶
- ∞ Unlimited**
- # Count
- Logic
- Time

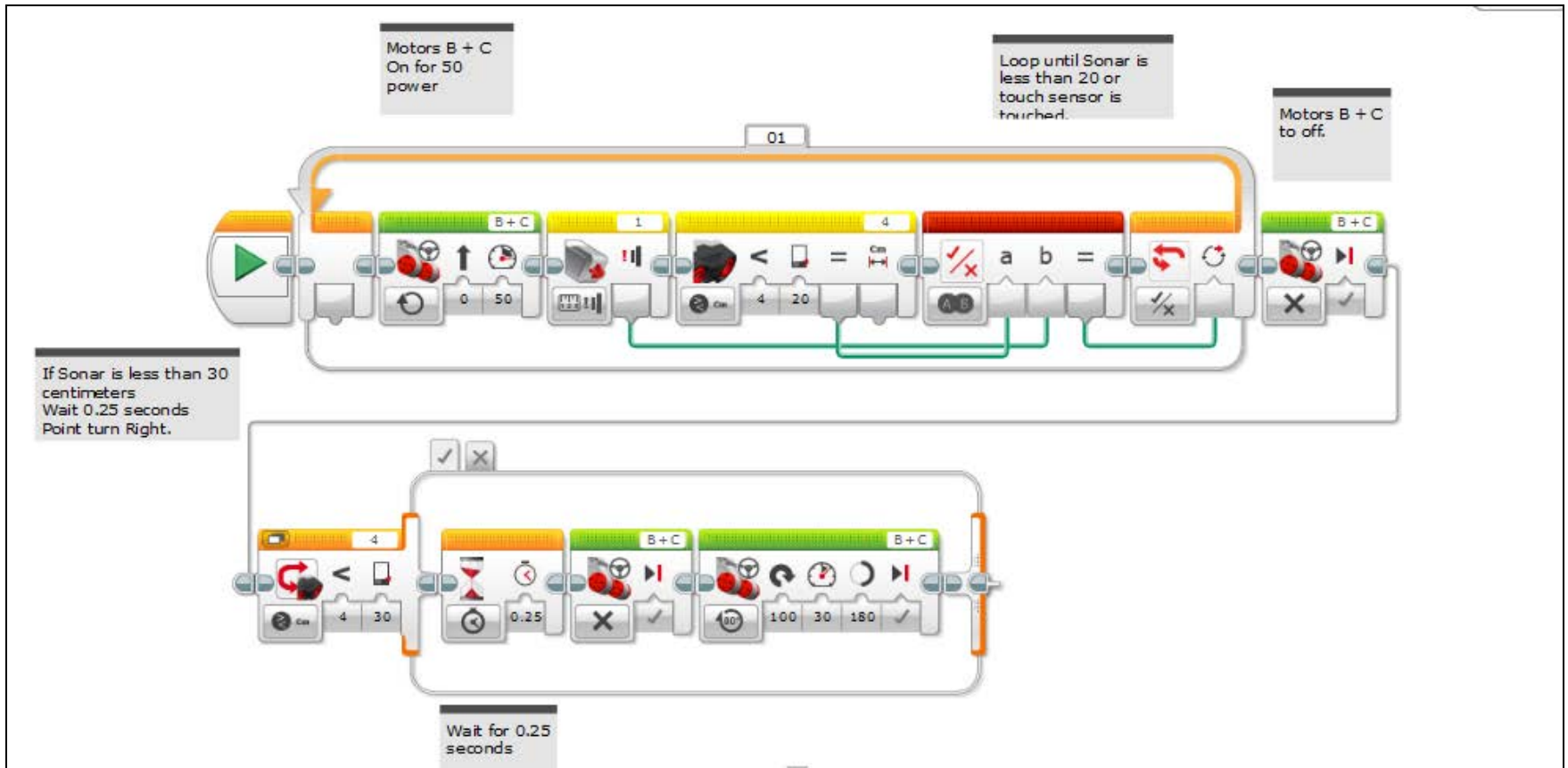
A 1 0

B 747

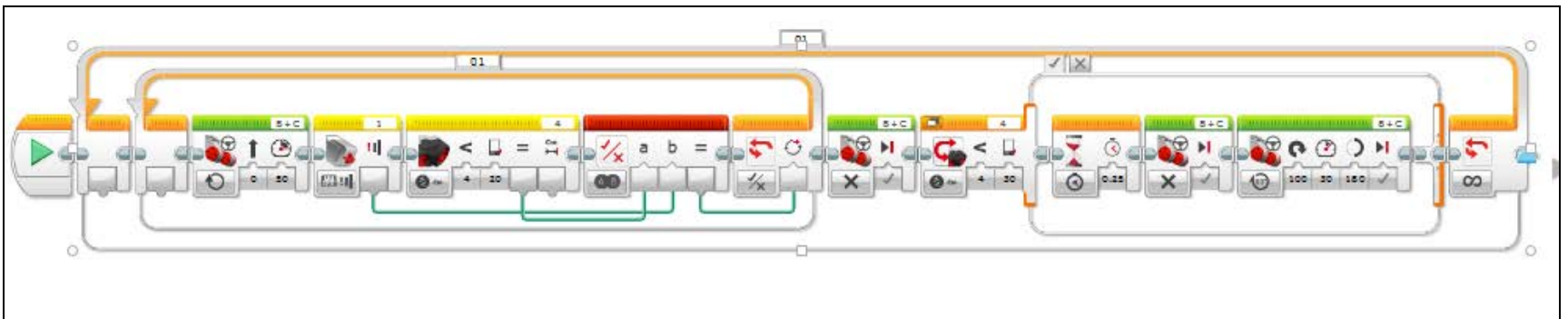
2

3

Example with Loop and Logic Blocks



Loop within Loop



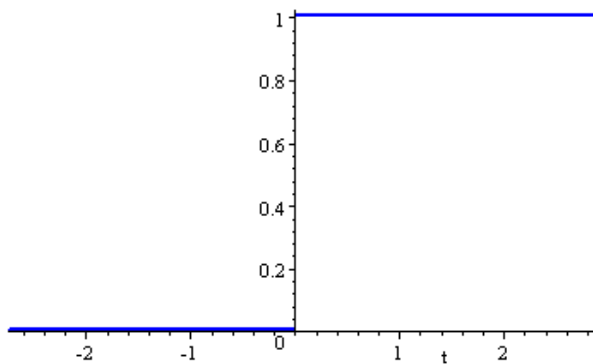
Objectives

- Linear Programming: Maze Navigation
- Sensor / Event Driven Programming
 - Touch Stop
 - Sonic Stop
 - Line Following
 - Heaveside
 - Proportional

Line Follow Algorithms

- Heaviside

$$H[n] = \begin{cases} 0, & n < 0, \\ 1, & n \geq 0, \end{cases}$$



- Proportional

```
float kP = 0.5;
int threshold = 40;
int lightValue = 0;
int error = 0;
int power = 50;
int turn = 0;
int leftP = 0;
int rightP = 0;

while (true) {
    error = lightValue - threshold;
    turn = error * kP;
    leftP = power + turn;
    rightP = power - turn;
    motor[right] = rightP;
    motor[left] = leftP;
}
```

Artbotics