Robot Design Booklet

By : Geneworx Oughtamation

Table of Contents

Mechanical Design

Mechanical design includes the durability, and the mechanical efficiency of the robot. The following information will describe the mechanical design of the robot and its attachments.

Mechanization

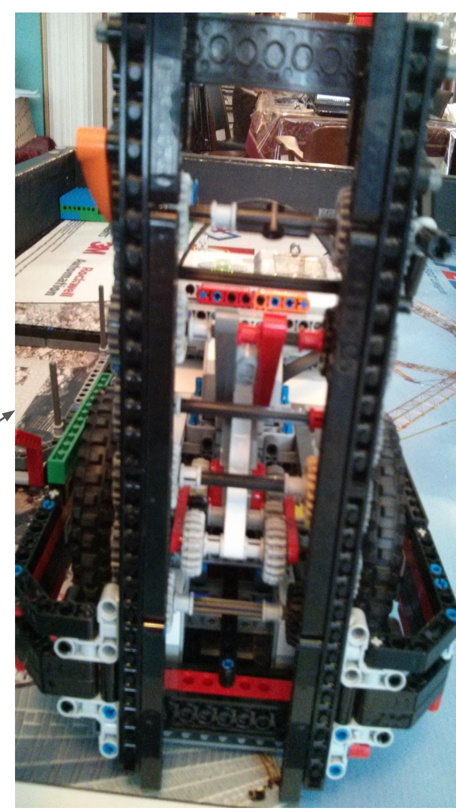
Using all five connecting pieces on the wheel

- The wheels are very strong. There is a large axle going through the middle of the motor, and four pins sticking into it as well. Because we connected the wheels to the four pins and the axle it becomes way stronger and won’t flex.

This is the wheel

Use of axles

- We use axles a *lot*. On the lift, the axles actually help attach the it to the robot, because lift arms are going from the robot to the axles. Also on the lift, axles go through the holes on the sides and attach the gears in place, so obviously, the axles are important because the gears are what make the whole lift spin and lift the attachment.



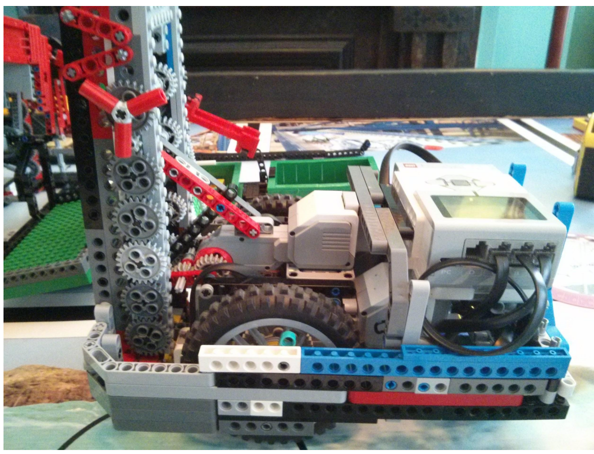
The lift is connected to the robot by axels

Durability

Use of frames to create strong parts

- Frames are strong because the apron is connected to the Robot. The apron connects on the wheels and on the brick to hold it together.

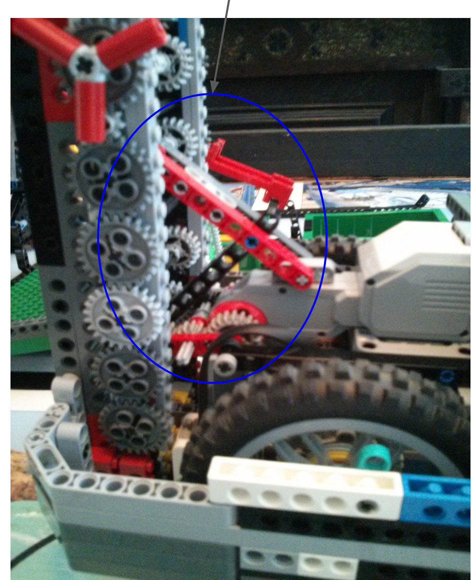
This is the apron, the apron is attached to the the wheels and other parts of the robot



Use of triangles in the robot

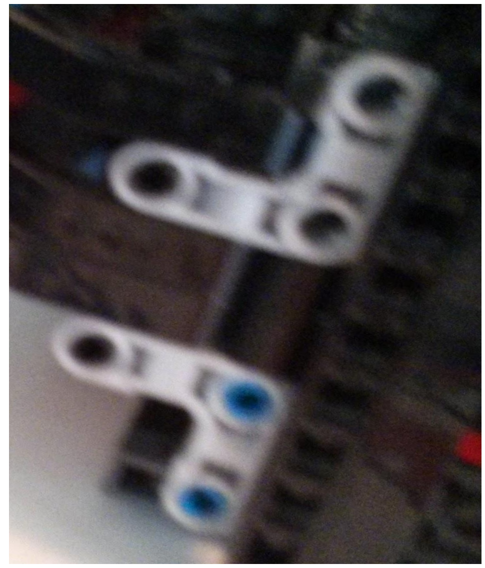
- Triangles are the strongest shape. If you look, you will find that we have used a few triangles in our robot. Including the third motor, the crosslinks on the lift and the bottom of the robot to connect the different frames. On the sides of the lift, there are triangle connectors and diagonal technic. This is important because this prevents the lift from breaking.

This is one triangle on the robot



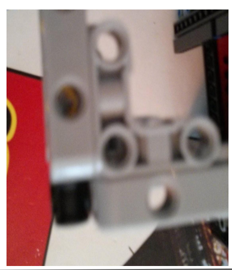
Durability

Use of hassen pins to make it redundantly sound



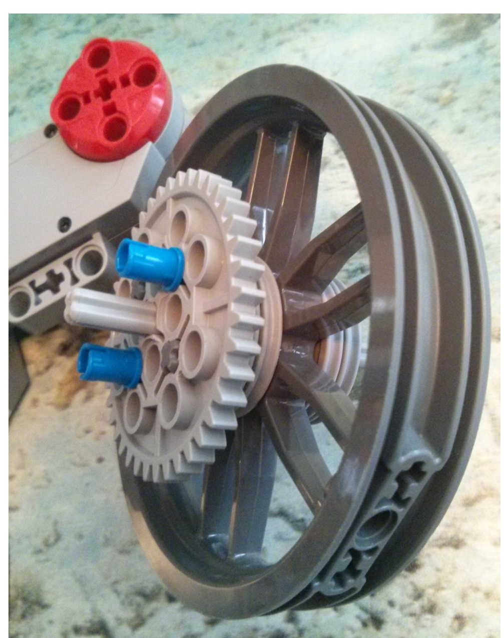
- Hassen pins are very strong and extremely good for building things. If you look, you will see that there are lots of hassen pins on our robot. (Hassen pins are the pieces that are angled and have 2 pins sticking out of each side.)

We use hassen pins to strengthen the attachments



We used hassen pins to secure the lift

Motors Locked to Wheels

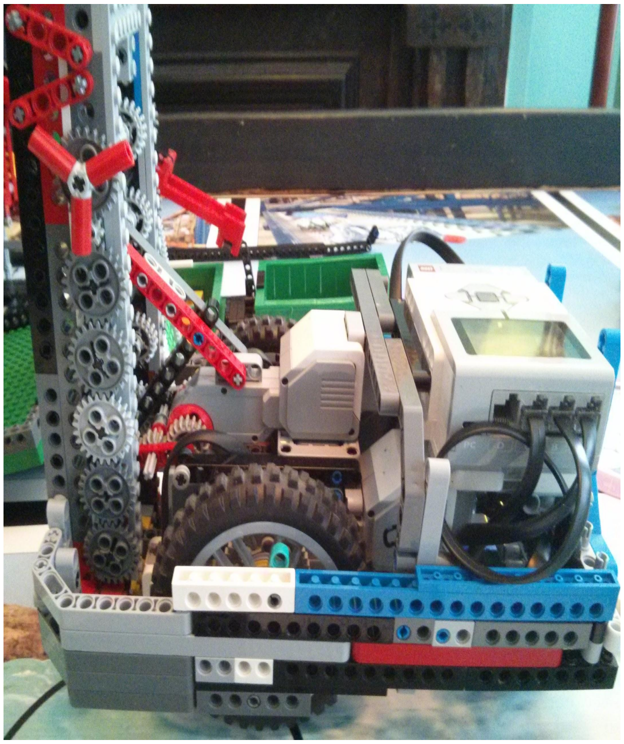


- The motor's and wheels are locked into the apron with pins and axles through lift arms.

The pins and axles will later also go through lift arms

Durability

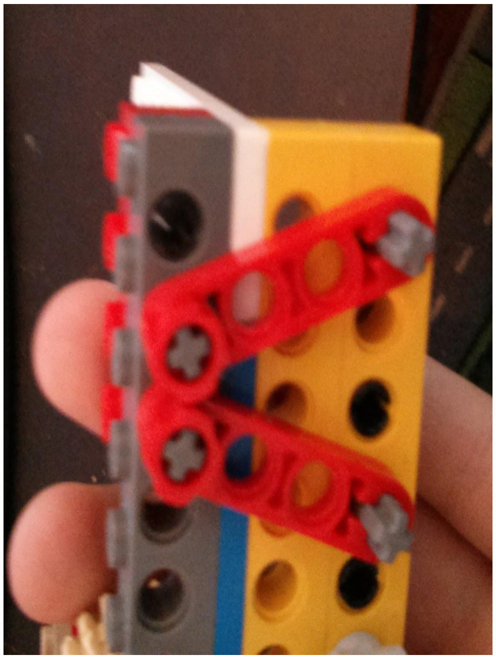
External wheel support



The axle connects the motors the apron

- As you can see, the apron has holes in it, and there are axles coming out of the wheels that go through the apron. This steadies the wheels.

Crosslink technic support



As you can see the technic bricks are connected by the lift arms

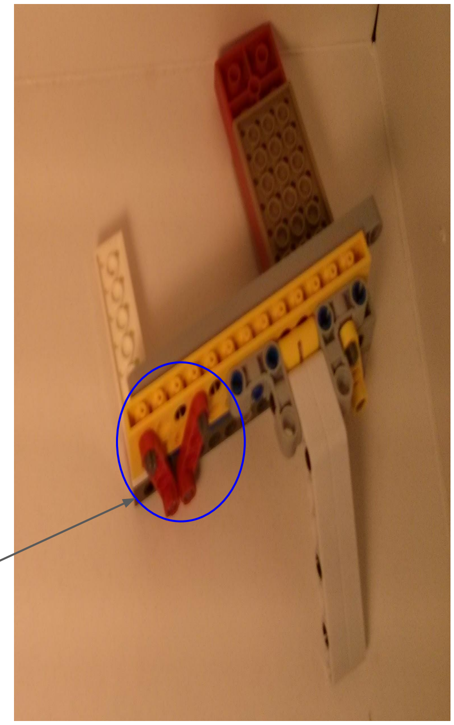
- Crosslink technic support is when technic beams connect technic bricks vertically. This strengthens the bricks a lot because then the bricks cannot separate at all without the lift arms coming off.

Mechanical Efficiency

Triple length pins to strengthen

- We use triple pins most of the time because they go through three layers instead of just two, making it stronger and more durable, and using less pieces.

There are triple pins here



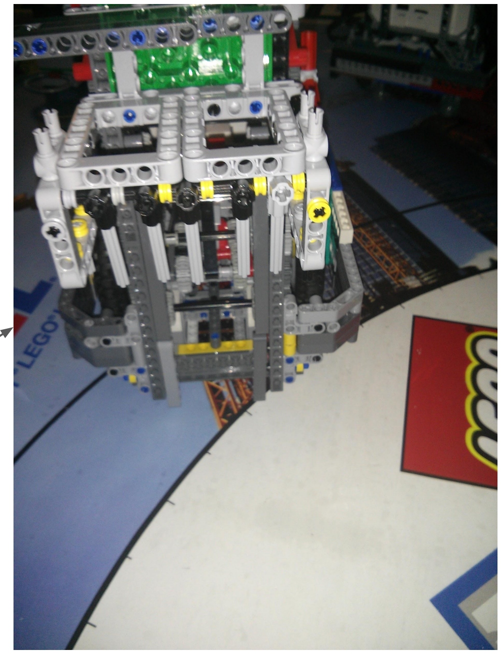
Efficient exchange (1.4 seconds)

- The lift has an open top, and it is very easy to switch out attachments because you can program the lift to rise on the way back to Base, so by the time it is in Base it is at the top and you can switch attachments without having to force the motor. Also, There are no pins you must snap into place, which would take time- a limited resource in FLL.

Mechanical Efficiency

Modular design / attachments (lift)

- Each attachment fits into the lift easily. Each attachment is also very quickly changed, because the attachments slip into the lift easily.

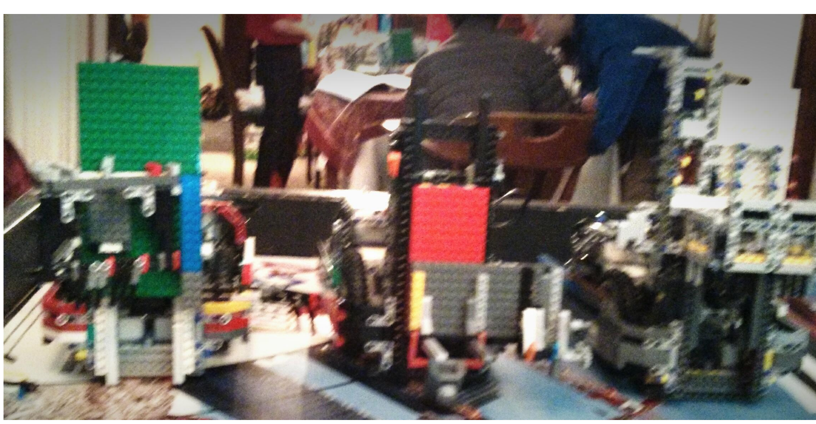


This is an attachment in the lift for the truck.

Parallel work

These are the three robots that are exact replicas these robots have different attachments in the lift

- We have multiple copies of the robot (Three), so multiple people can work on building and programming at the same time.



Strategy + Innovation

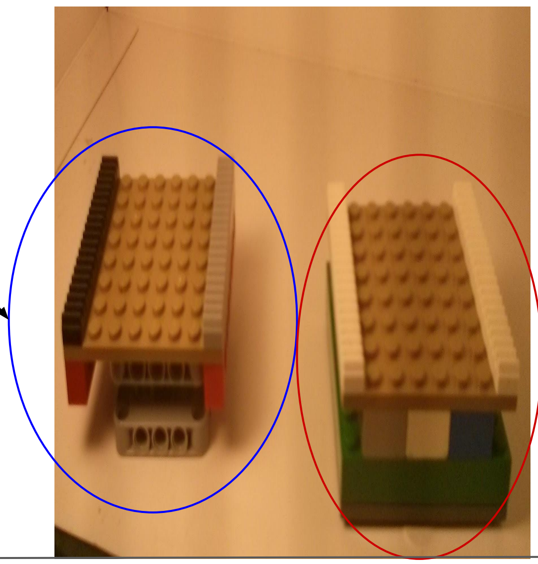
We have a very unique robot design, it has a very sturdy elevator and apron design that could “survive a fall off the Empire State Building” as a judge at the qualifiers said when we got our trophy. Our elevator allows us to change attachments in 1.4 seconds.

Innovation

Standard Interface

- Each attachment has the same backing, so each attachment can go into the lift. Part choice- 4 x 1 racks, 8 x 16 plates, brick, 12 x 24 brick.

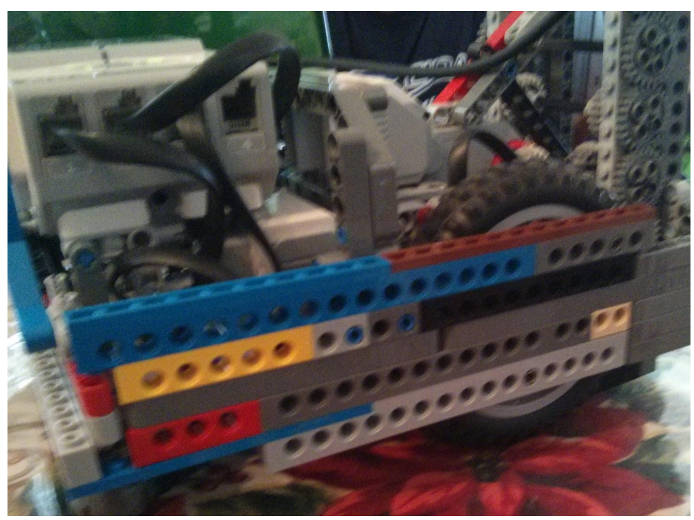
This is the old interface



This is the new interface

Frames are durable

- Frames are another strong element that we have used in the robot. This is because they are very durable. Frames are unique because you can build pieces as strong, but they are way heavier and you can build pieces as light, but they won't be nearly as strong.



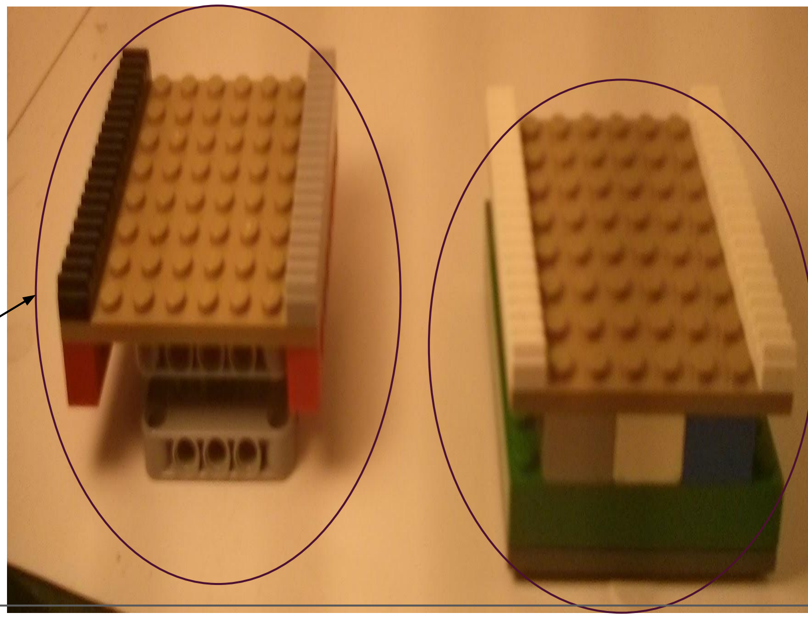
This is the apron a frame on our robot

Mission Strategy

Virtually no space wasted in lift

- No space is wasted on the lift because, the interface is small and can slip into to the lift easily with no time at all.

The old interface took up too much space. (3 brick heights)



This interface takes up little space(1 plate)

Large wheels speed up the robot

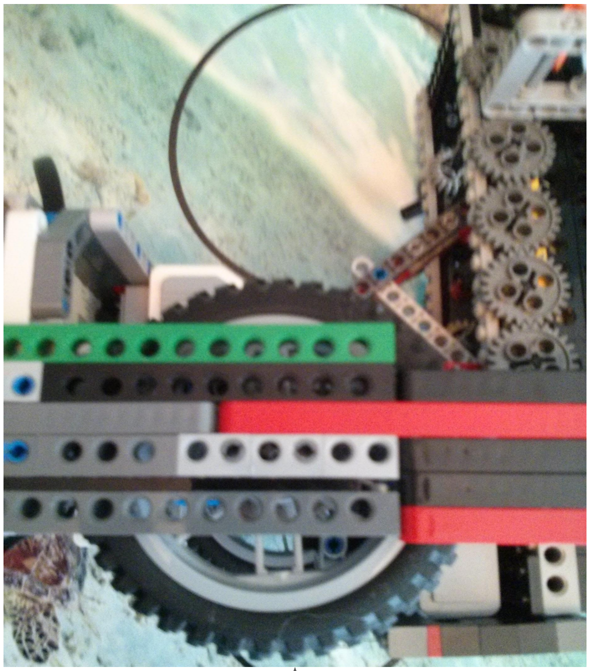
- The bigger the wheels are, the faster the robot goes. We have the biggest LEGO wheels that have treads.

This is our wheel (this is the second largest LEGO wheel you can buy, after the exclusive wheels of the HailFire Droid set.)

Innovation

Apron around robot

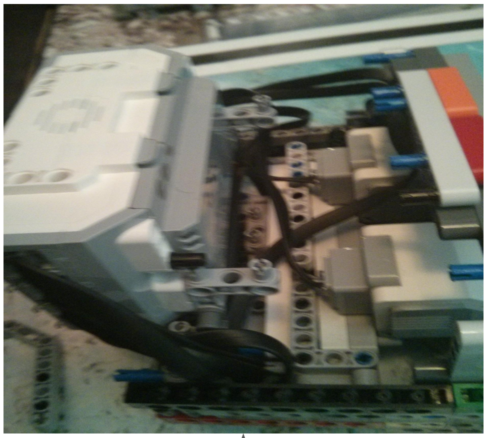
- The apron around the robot mostly acts as a shield around the robot. It protects the robot from damage if it drives into something. Also, it centers the wheels and makes sure that the wheels don't get damaged when the robot runs into the wall. An axle from each wheel goes through one of the holes in the apron. The apron also helps line up against the wall.



This is our apron.

Hinged brick allows user to replace batteries easily

- We actually originally had the brick sideways, but the brick was hard to use. So, we rebuilt the brick section so that it faced upward, making it way easier to use. But, it turned out that you could replace the batteries easily because the back can hinge.



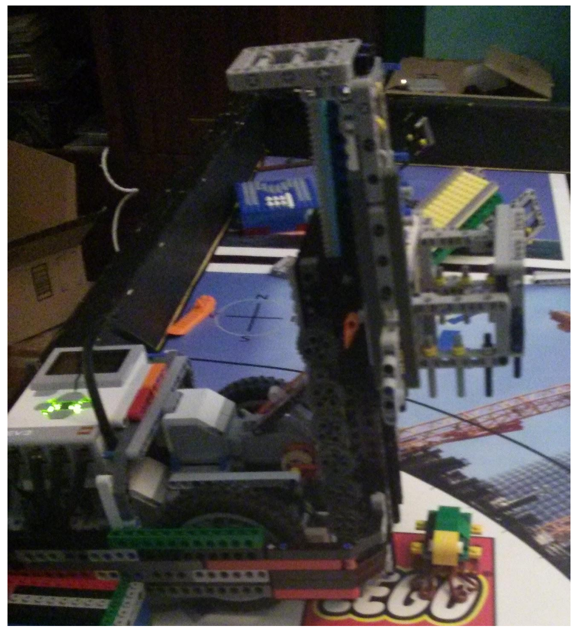
This is the robot unhinged from the robot after having the battery changed.

Design Process

One to one gear ratio gives strength to lift heavy attachments

- The gear ratio (1:1)is perfect to lift heavy attachments such as our attachment to knock over the building for demolition.

The robot is lifting the demolition attachment.



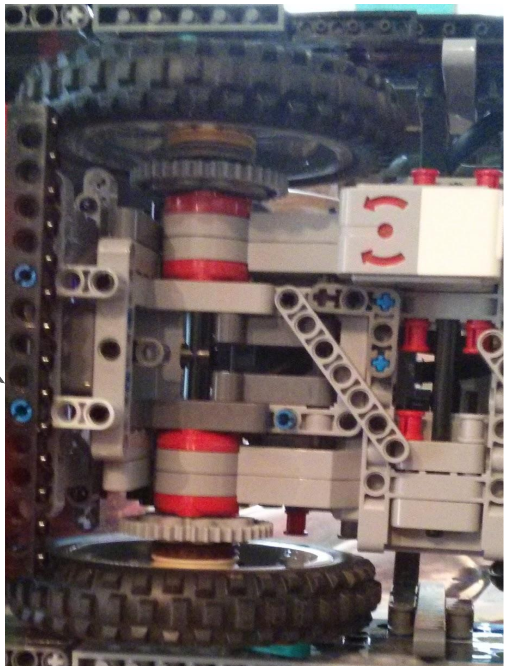
This is our wheel

Large wheels speed up robot

- We have the largest treaded wheels that LEGO makes. This makes our robot faster than others. Also, it gives more traction/ friction so that it doesn’t make the robot skid.

Design Process

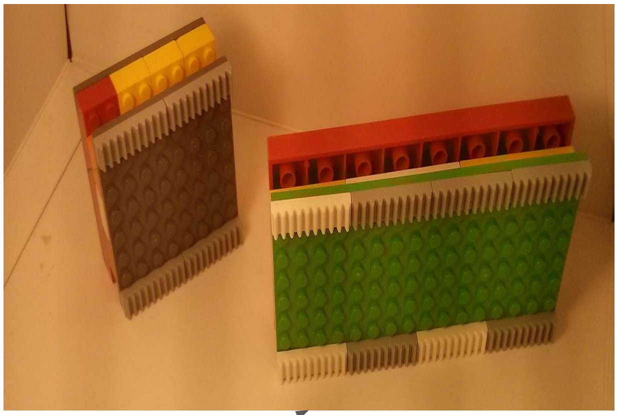
Wheels are also accurate- supported on outside and on motor



- Our wheels are accurate because they are held in place by the apron and the motors of the robot. (The way it's attached, the motors and the wheel are basicly like 1 giant part)

The wheels are strongly attached to the robot.

Use of pins to connect the base plates back on the attachments



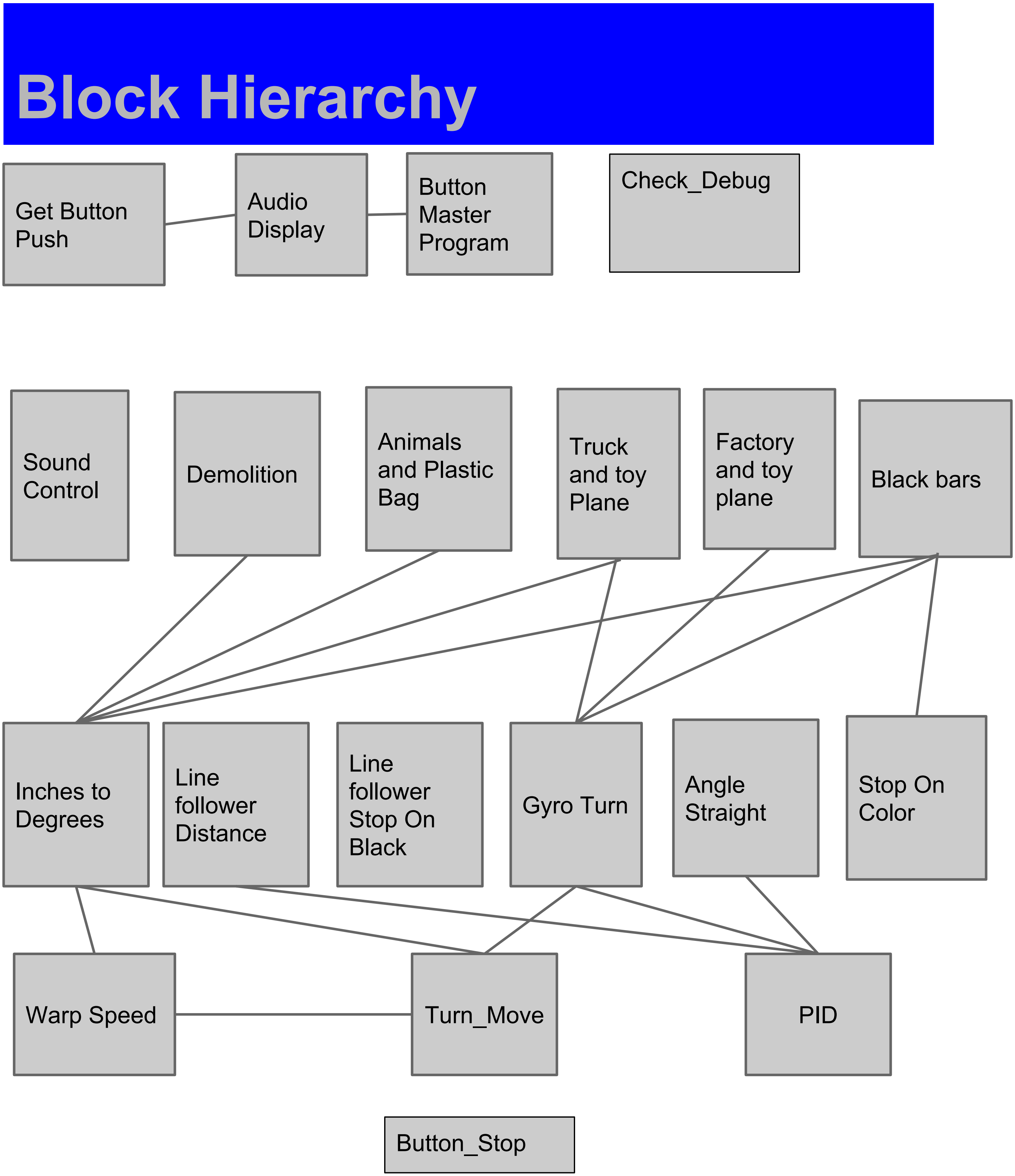
- We made the base plates easier to build on by using pins to attach two plates back-to-back. Originally, we had the plates snapped together so we had to build on backwards, which would have been possible but not convenient at all, so we built a long series of base plates with different pros and cons, but our current one is the most useful and stronger. Now all of the individual base plates are in a presentation we call the *Evolution of the Brick*.

These are two versions of the standard interface.

Programming

The programming we used includes the programming efficiency, and design process. The following information will describe the programming of the robot.

Programming Efficiency

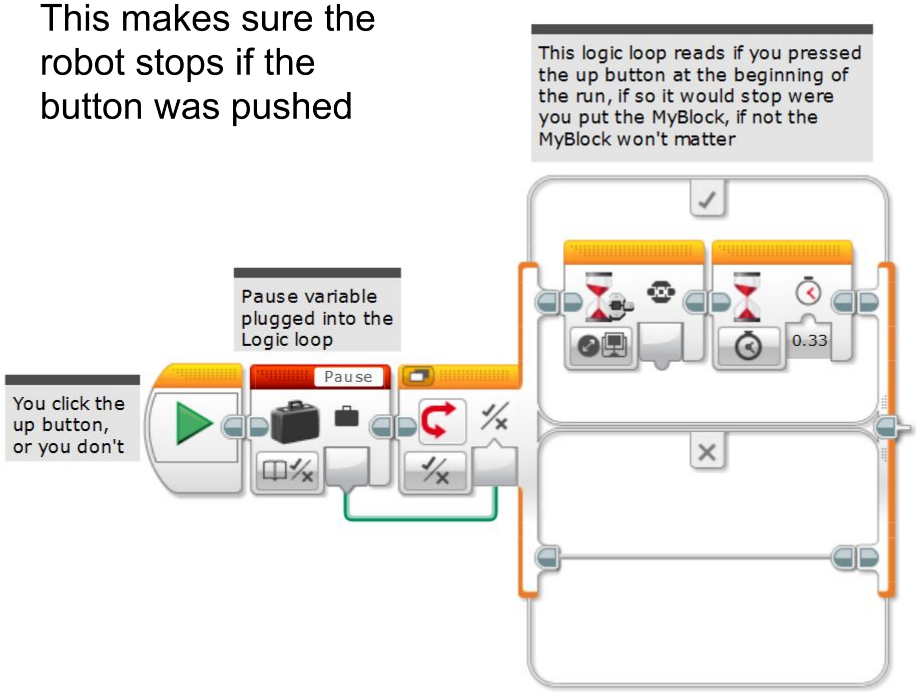


Rule of Five; Design process

The Rule of Five states that when you've made a program, it has to work five times in a row to be accepted as a part of the Robot Game routine. If it fails any of the five times, you have to fix it to make it work according to the Rule of Five.

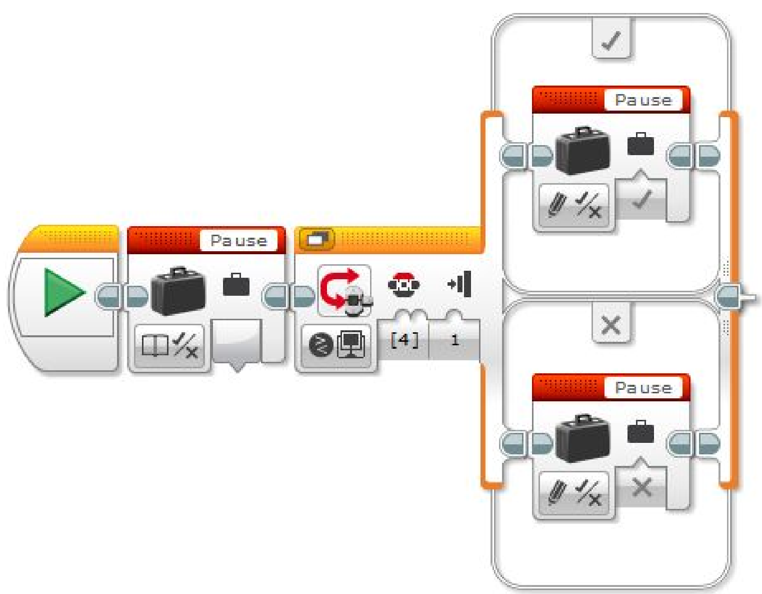
Debugging

This makes sure the robot stops if the button was pushed



This MyBlock is used at the beginning to lower the lift then stop, but can also be used when testing missions, we use this MyBlock in all missions to make the lift stop before starting the mission to make sure it is in base.

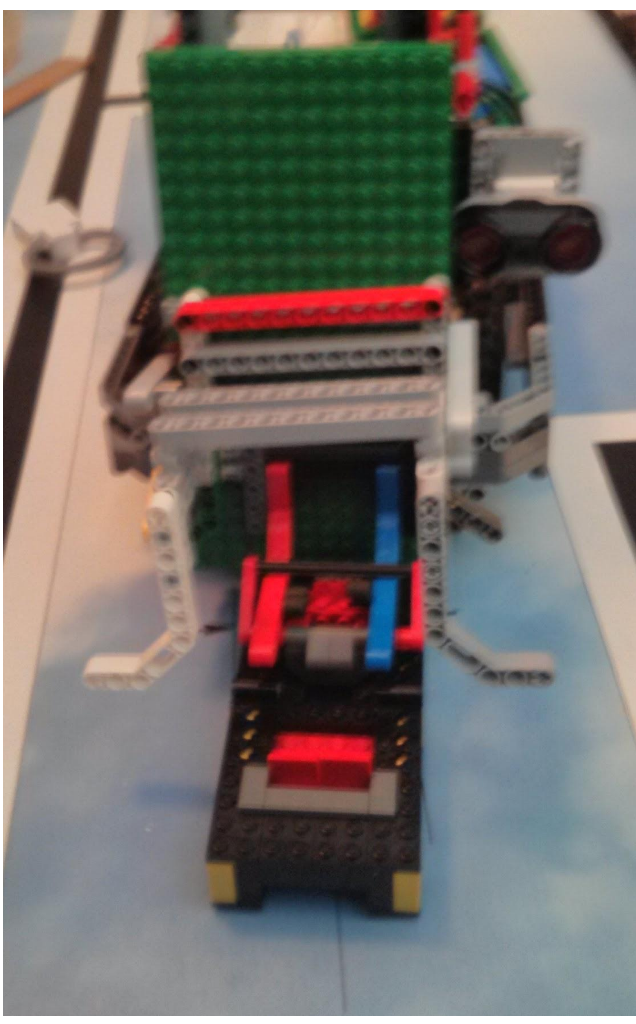
There are two parts to this mission this is to read if the button is pushed



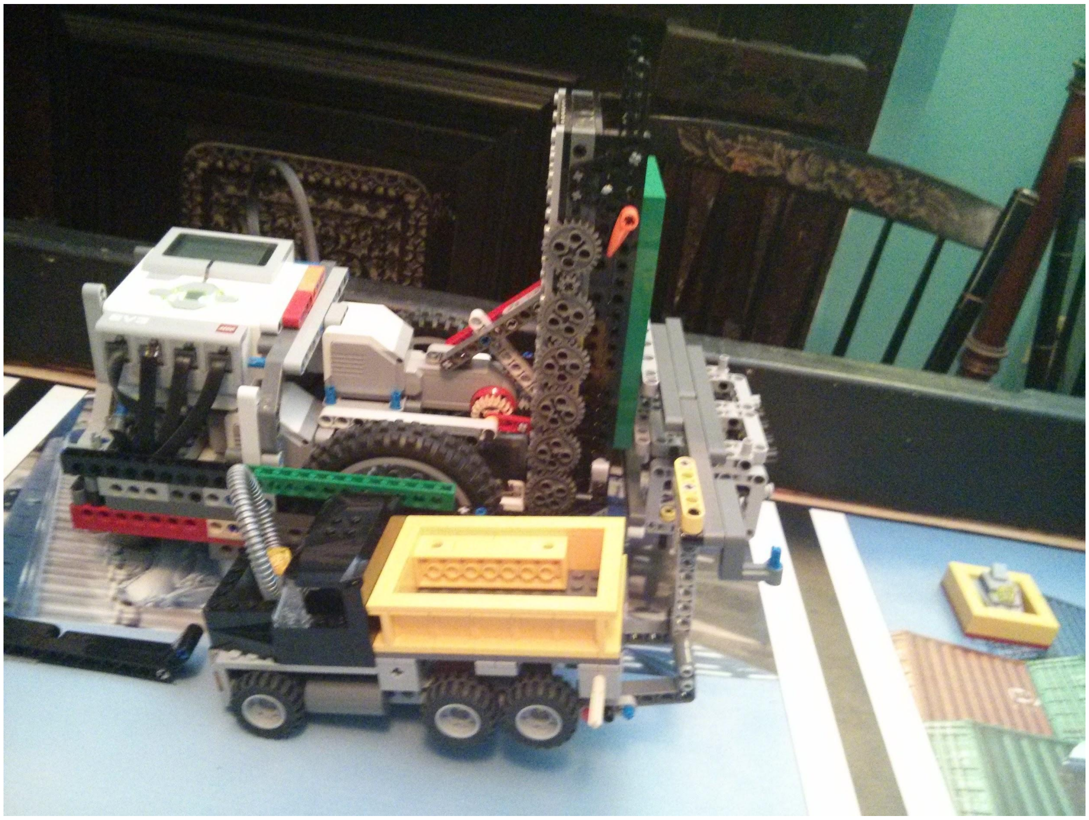
Engine Mission

The engine mission includes the truck mission, the animals mission, plastic bag mission, demolition mission, and factory mission. The following information will describe the missions we are performing.

This is the robot after it has finished the engine mission



Truck Mission

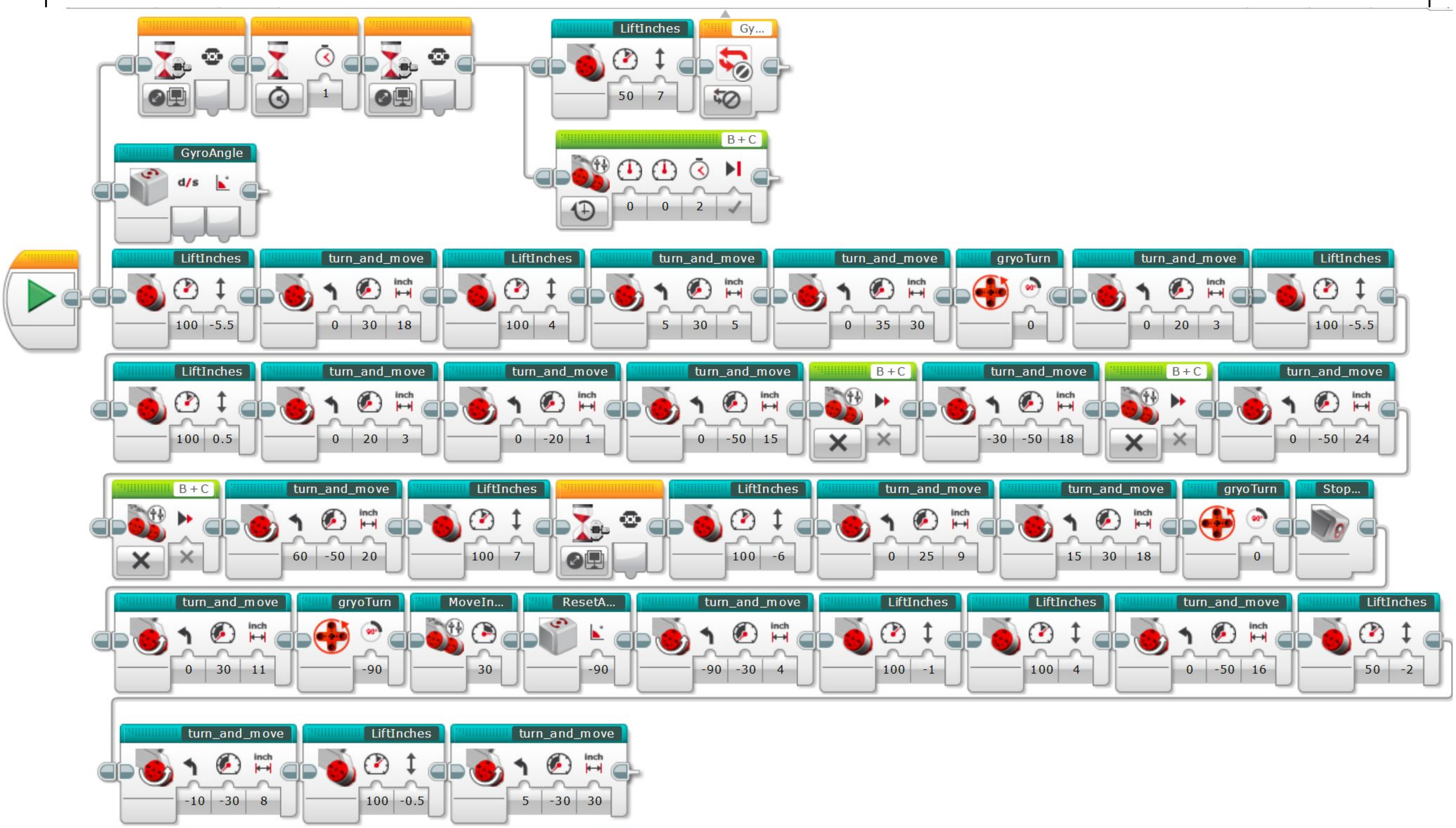


This is the robot after it after the truck mission, but before it grabs the toy plane.

This run gets the truck with the yellow bin on it,then drops the minifigure, lifts the truck and sets it down west of the guide. Then it collects the toy plane, backs into the wall, and heads for base, carefully avoiding the animals in the circle.

Animals, Plastic Bag and Demolition

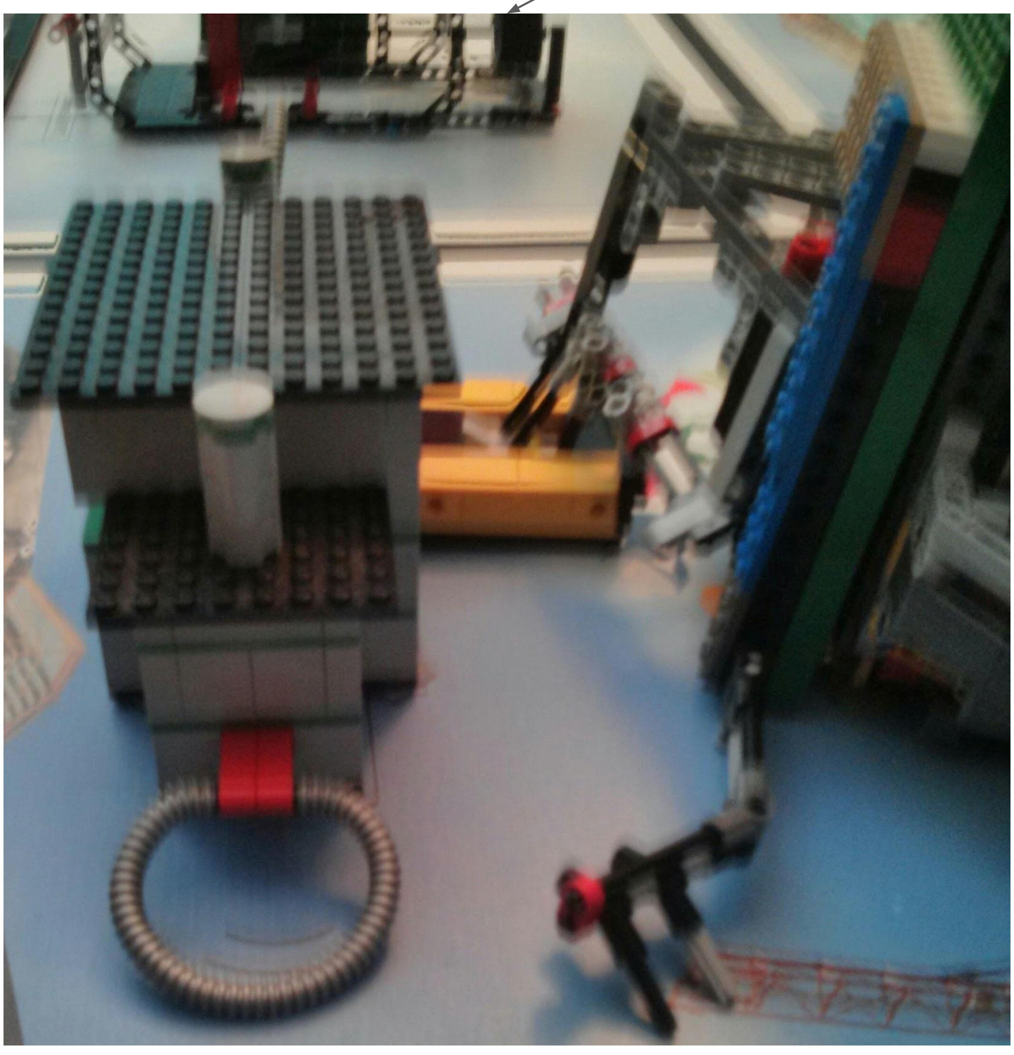
This mission moves a box out to catch the building as it falls, ejects the compost, collects the bag and the turtle, and heads back to base. Then the robot waits for a button push before putting the animals it the right hand circle and knocks down the building into the box, and bringing the valuables and the building parts back to base.



Factory Mission

The factory mission pulls the red handle on the side of the factory, then it collects the toy and grabs the plastic bag out of the sorter.

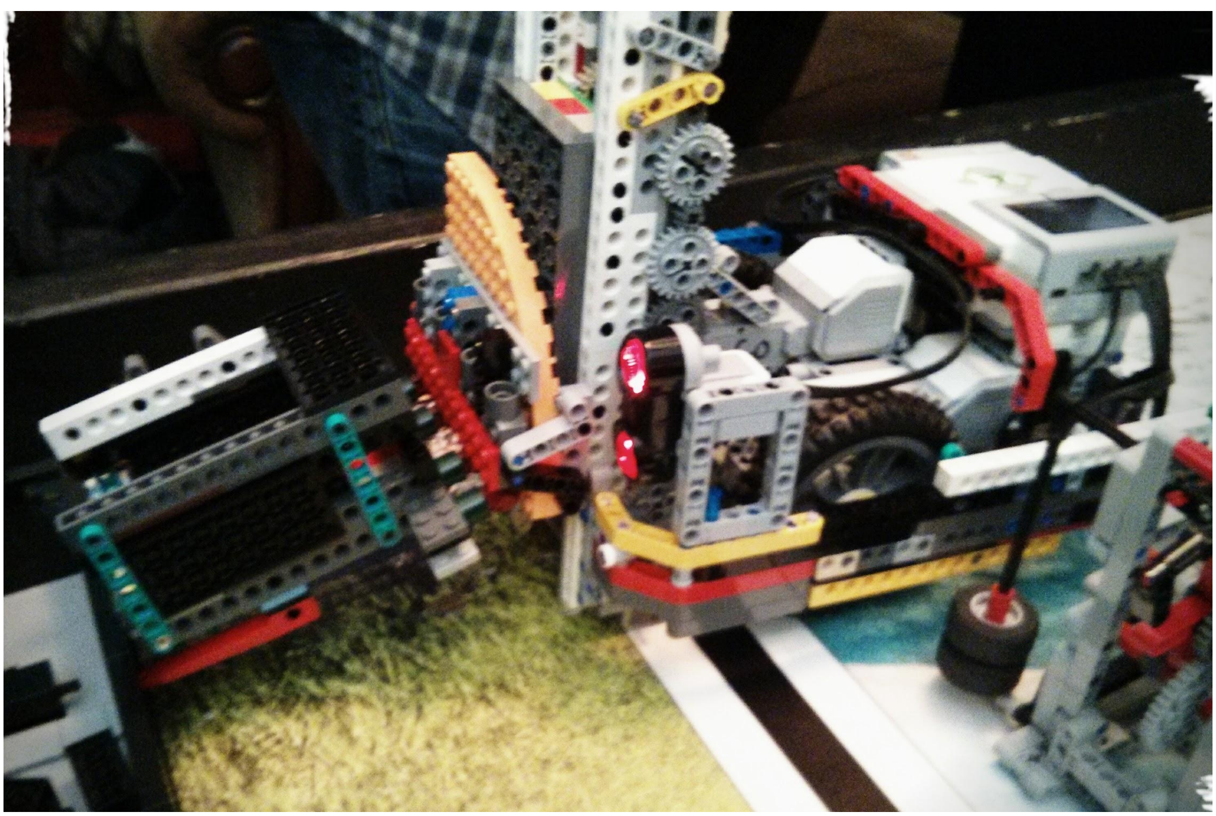
This is the complete factory Mission



Black Bars

The black bars mission uses the stop MyBlocks and turn_move Myblocks and the lift inches Myblock is also used and is very helpful. The mission is the final mission so we leave it where it is.

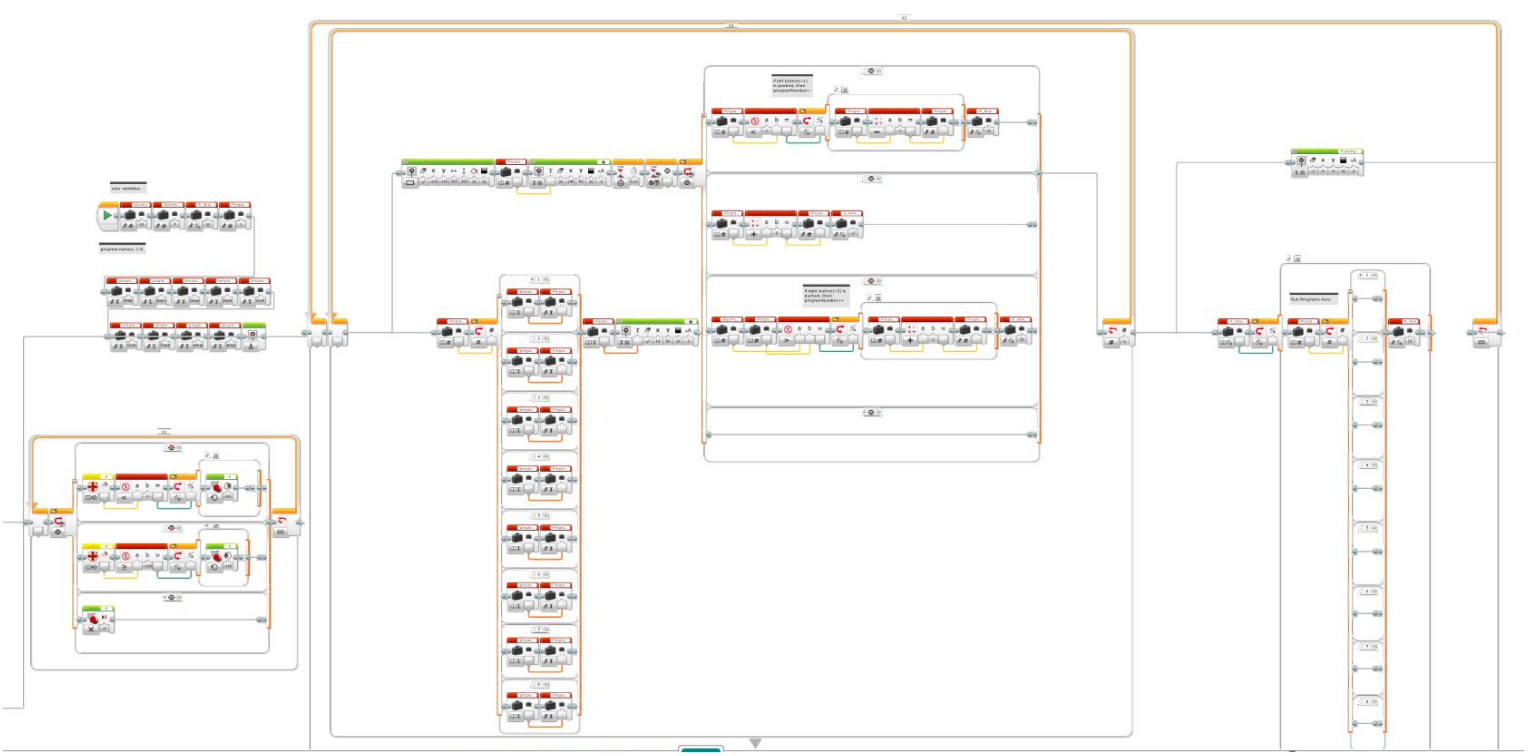
This is the Black Bars at the end



<u>Automation/Navigation</u>

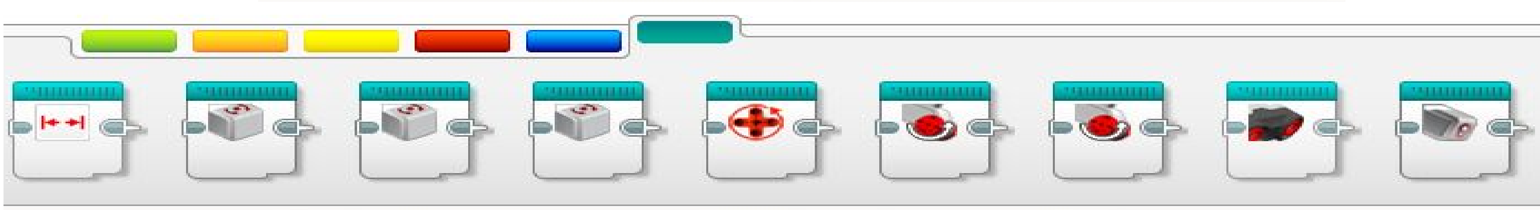
Master Button program

The master button program makes sure that while we run the mission, it makes it easier to run the mission in a row. It makes it easier because of instead of searching for the program, you can click the button.



MyBlocks

Our MyBlocks do a wide variety of things such as, pausing in the middle of a mission for testing a mission. Line following of PID. Efficient turn makes sure the robot goes the fastest route to the angle you want. Our stop n line is a great MyBlock, we use it as much as we can in every run. Our InchesToDegrees and CmToDegrees both move a measured distance and is way easier to go a distance than when you use rotations or other ways that MindStorms gives you like second or degrees, we use the degrees part which is simply type in inches then change the move block to degrees and plug them into each other.



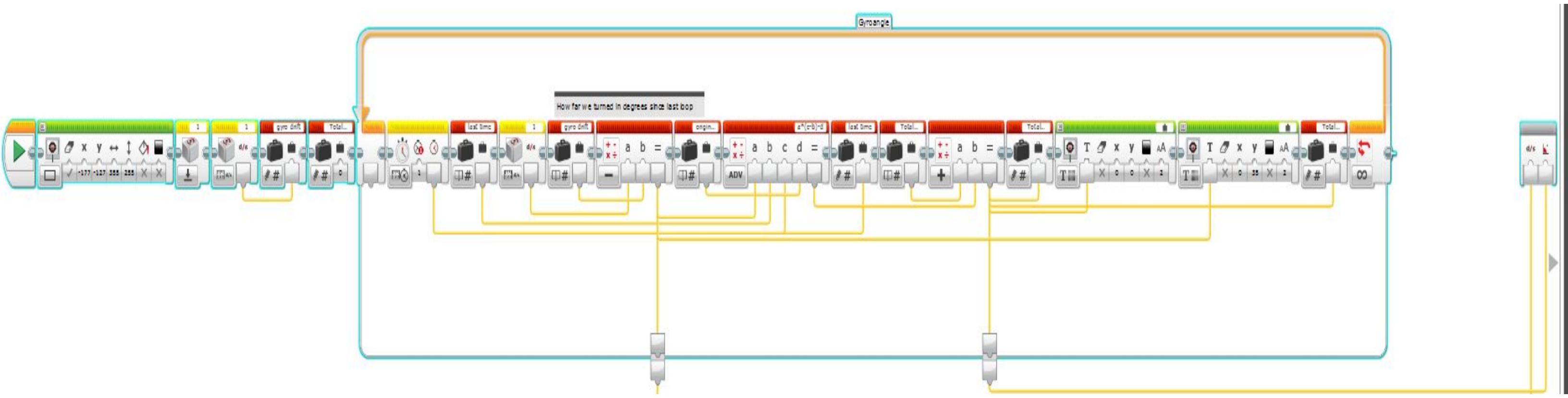
Programming Efficiency

BuildingBlocks

All of the MyBlocks made as infrastructure are in a set of files called BuildingBlocks. The BuildingBlocks are named so that you don't need a tutorial to use them. Also, the BuildingBlocks themselves are annotated so that in the programs, you don't really need to annotate as well.

<u>Programming Efficiency</u>

Gyro-Angle

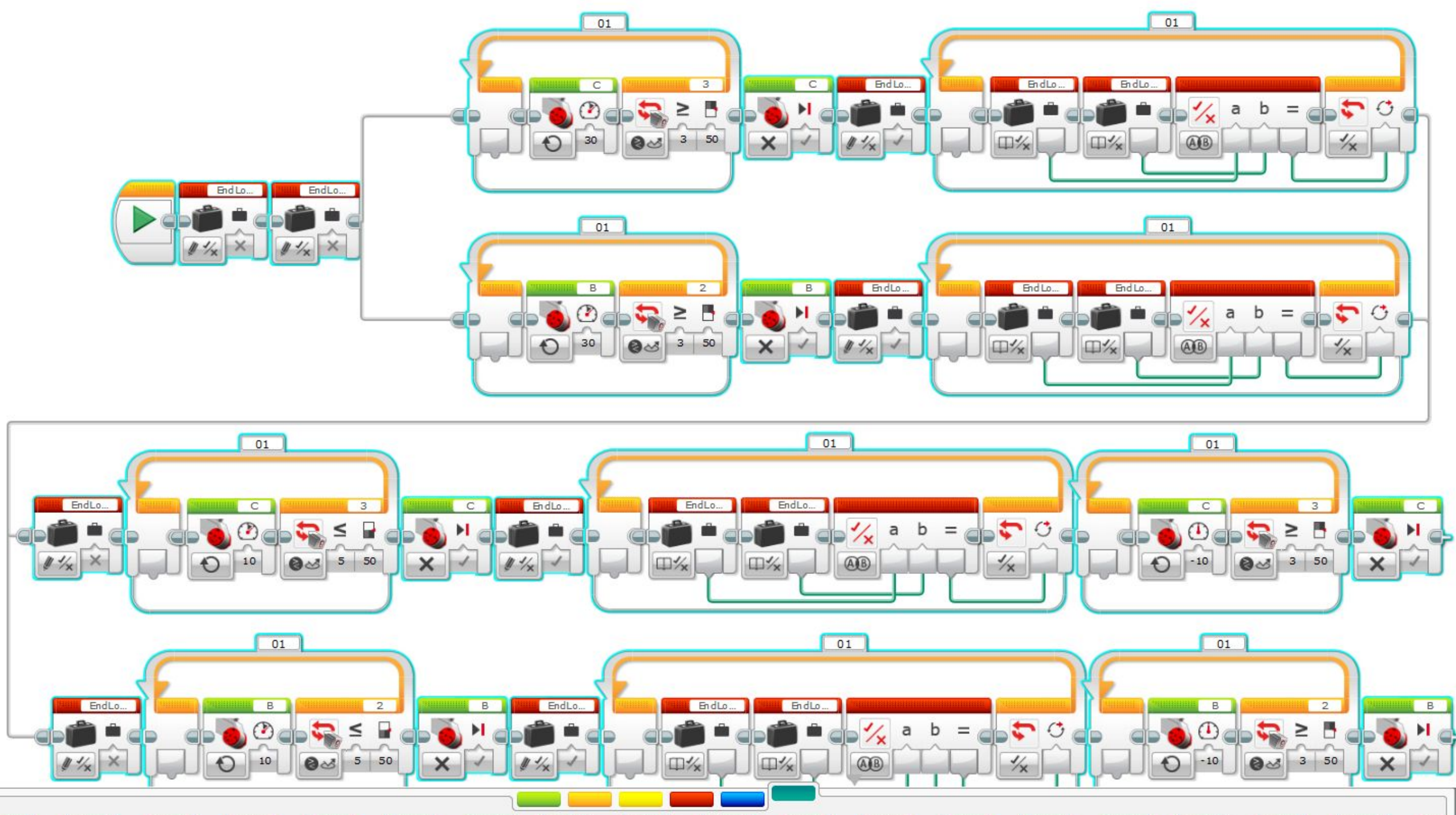


The gyro sensor measures rate, not angle. It uses an integrator to calculate the angle, but the integrator can be saturated easily. So, we wrote our own angle-integrator to calculate the position of the robot, with a built in feature to reset the gyro drift (which it then subtracts out of the rate) to reduce error.

Programming Efficiency

Stop on Line

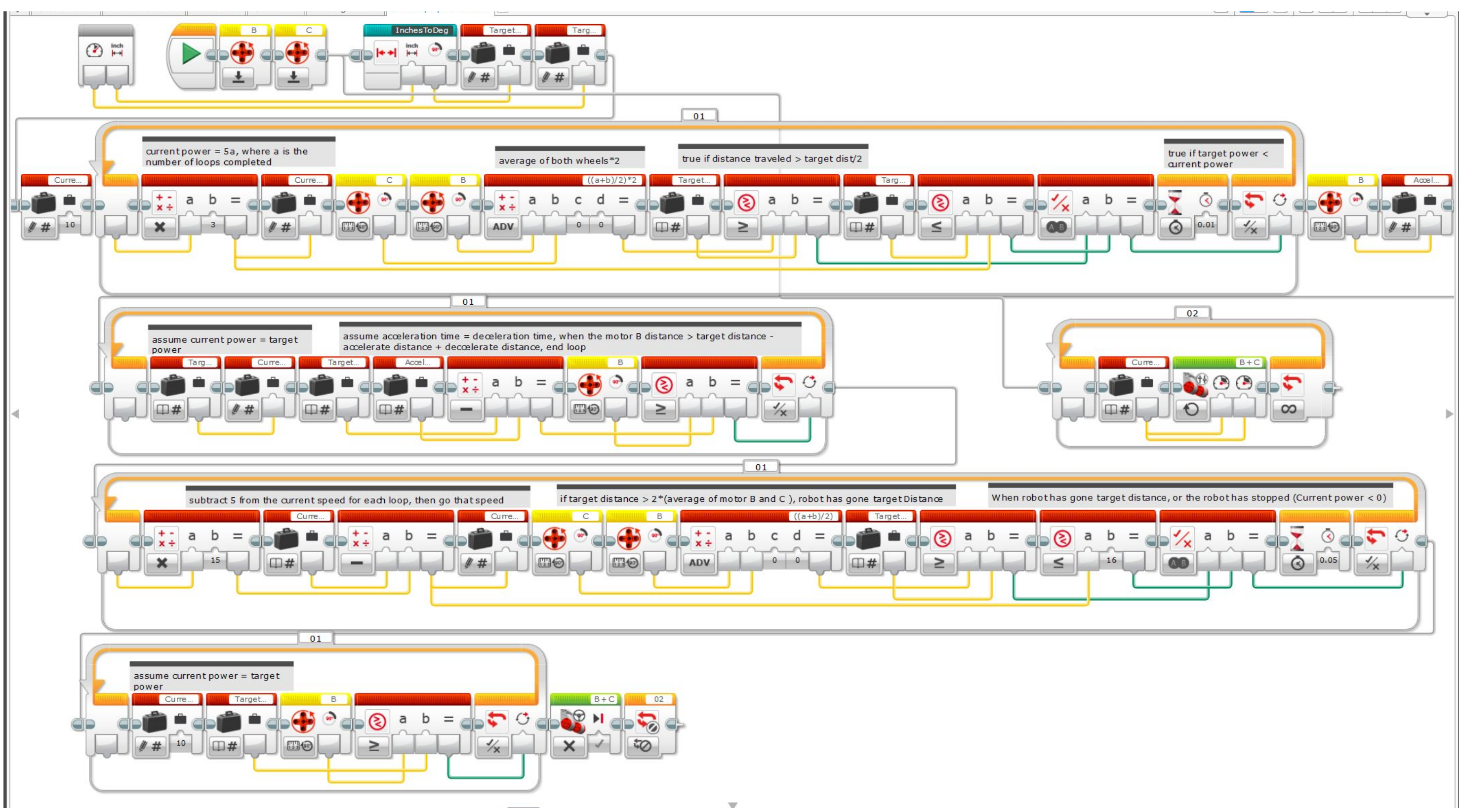
StopOnLine is an easy MyBlock to use.



Programming Efficiency

WarpSpeed

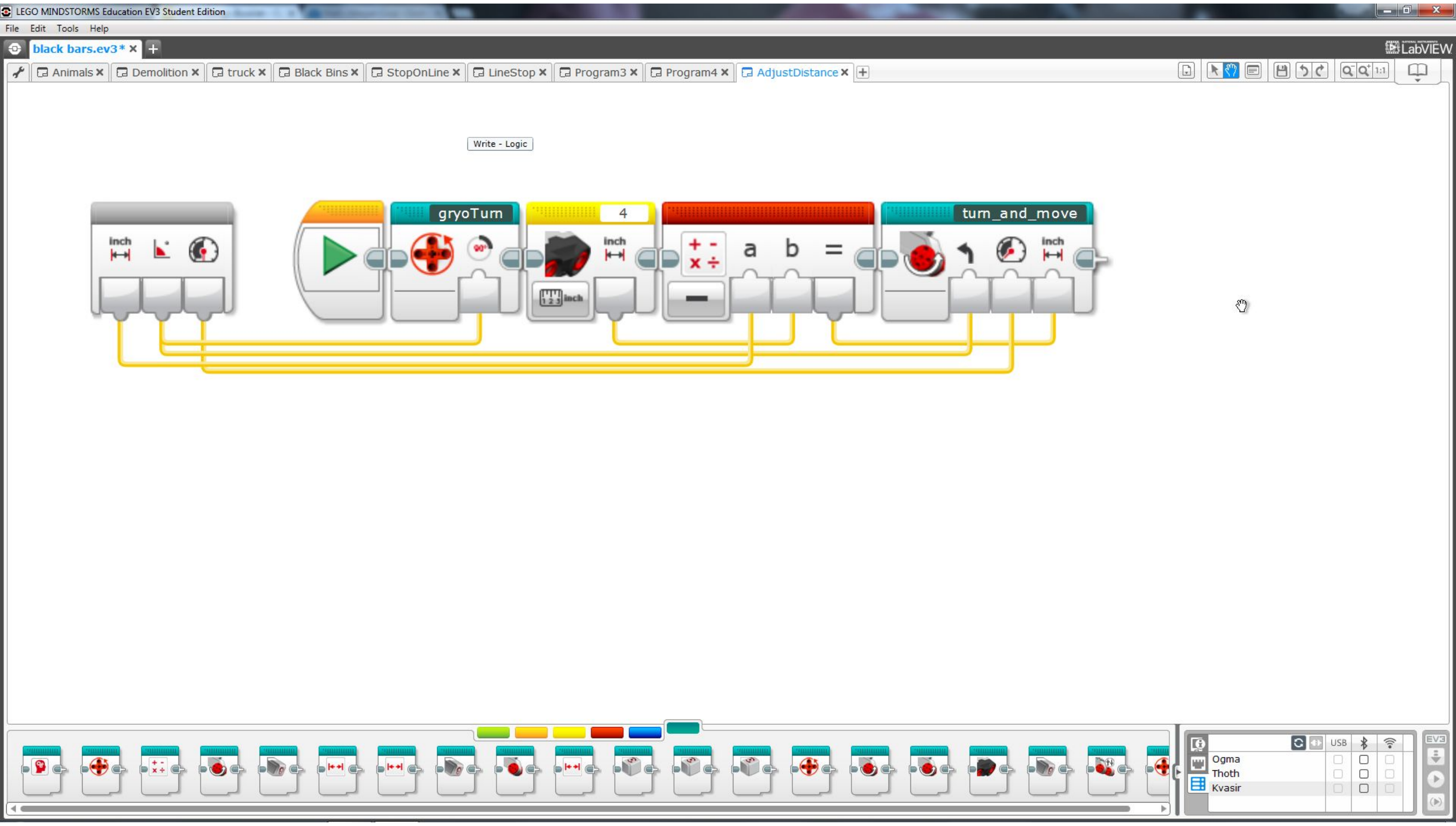
WarpSpeed is a BuildingBlock that accelerates to a chosen maximum speed and decelerates as it nears the target distance. The original plan was for the robot to accelerate exponentially and accurately estimate the decelerate time in order to stop at the correct distance, but this required calculus and the software couldn’t support the equation and so the WarpSpeed block uses linear acceleration.



Programming Quality

AdjustDistance

The AdjustDistance Block uses the ultrasonic sensor to position the robot a certain distance away from any object, given it's positioned in front of the robot.



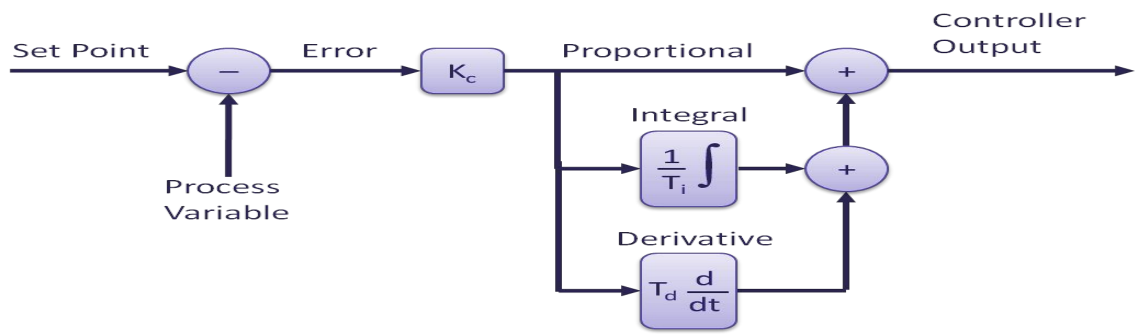
Oughtamation/Navigation

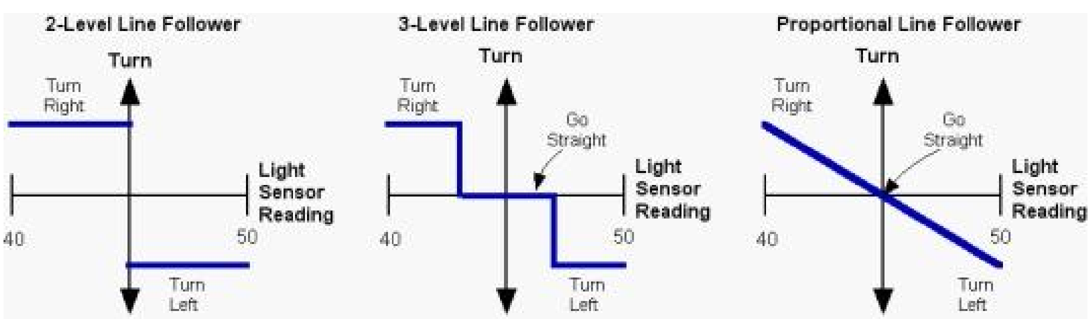
PID: Proportional Integral Derivative

Proportional: $m = \Delta y / \Delta x$

$\text{Integral}_{\text{New}} = \text{Integral}_{\text{Old}} + \text{Error}$

Derivative= (Current Error) - (Previous Error)





Automation/Navigation

What is PID?

With the 2 range line follower, Robot never drives straight, even if it is perfectly aligned with line's edge and the line is straight. With a 3 range, the robot goes straight for the light values near the target value, but curves more the further away you are. With the proportional line follower it has, effectively, an infinite number of ranges, the further off the robot is from the line, the more it turns, the turn is proportional to the error.

Proportional means a graph of the variables against each other produces a line

Integral is the running sum of the error.

Derivative is the change in the **error** between two consecutive points.

Programming Efficiency