3958-GREEN MOUNTAIN GEARS

Cailin Fitzgerald

OVERVIEW

Introduction

Robot Game Goals

Mission Specific Designs

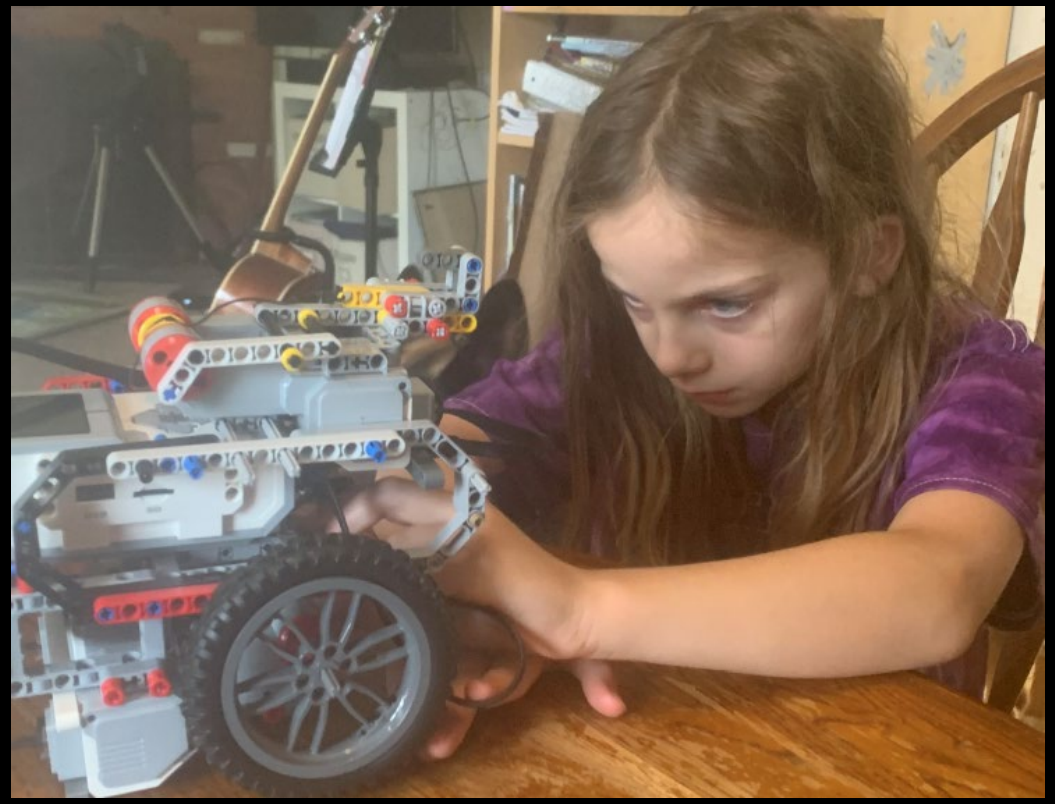
Autonomous Navigation

TEAM 3958

- Founded in 2011
- 2019-Core Values Award VT Championship
- 2016-Programming Award VT Championship
- 2015-VT State Champions, Semi-Finalist Global Innovations Award
- 2014-VT Qualifier Chamption, Robot Design Award VT/NH Regional Championship
- 2013-Judges Award VT/NH Regional Award

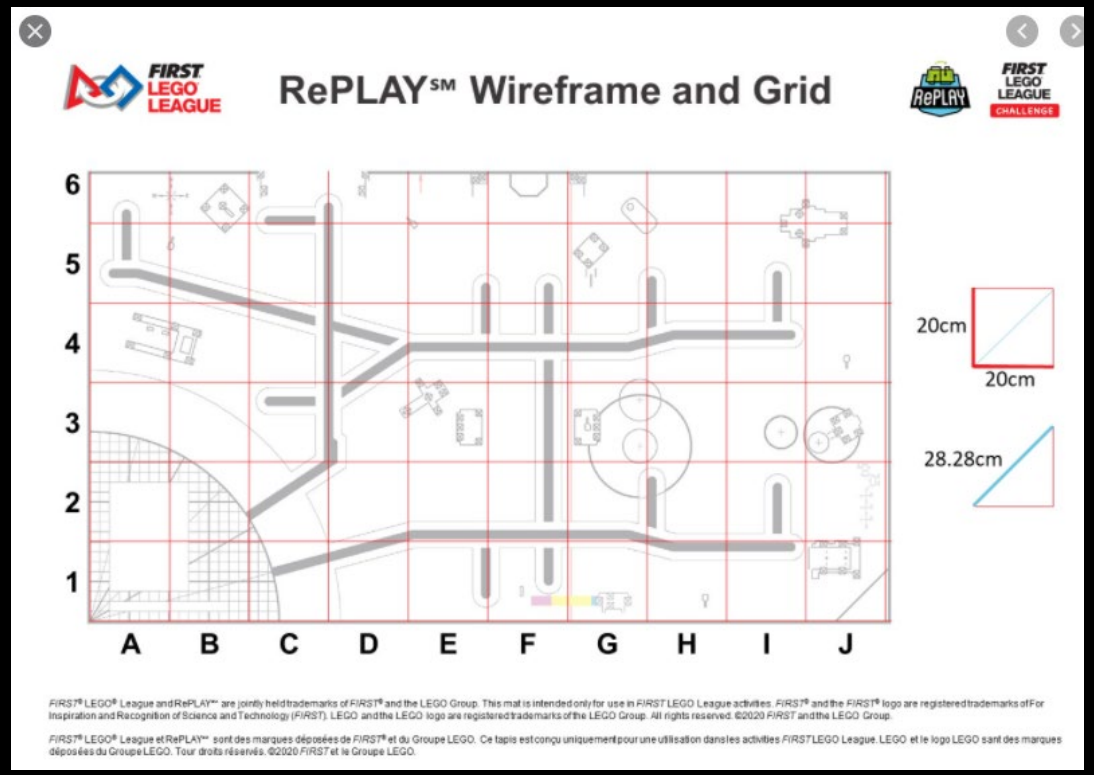
TEAM MEMBERS

- Paul Fitzgerald, Coach
- Luke Fitzgerald, Mentor
- Cailin Fitzgerald, member



MISSION SELECTION

- Close to base
- Easy way points for navigation
- Mission that can be done with simple robot designs
- Missions with high chance for success
- Small chance for penalties



ROBOT MISSIONS-PRECISION (60) + 70 + 65= 195

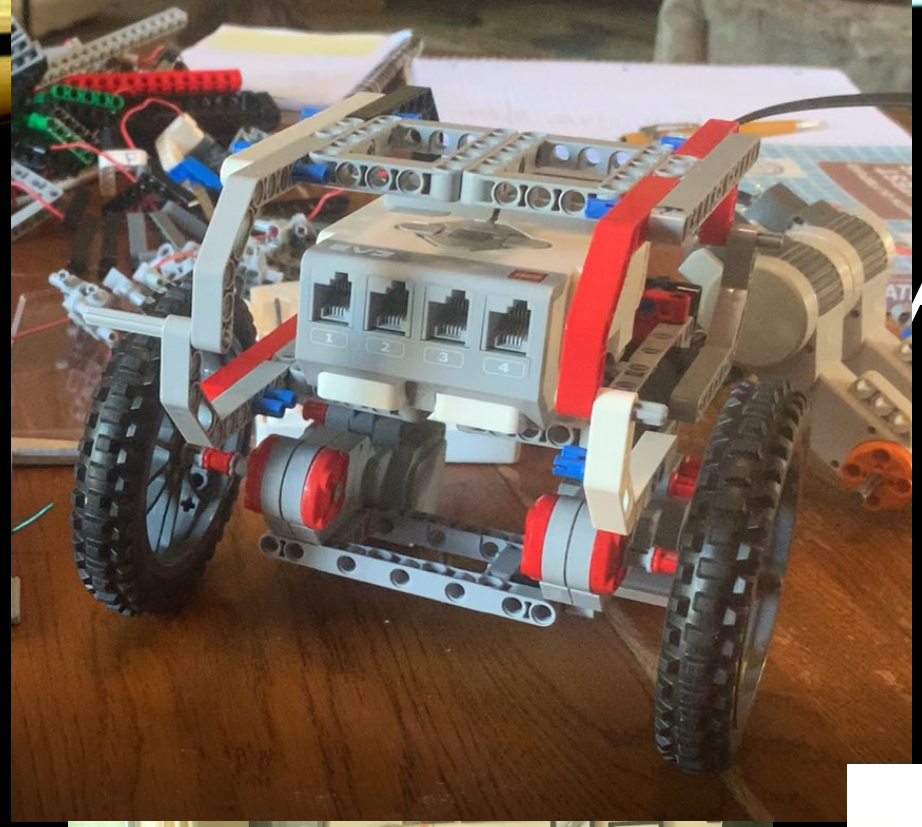
https://youtu.be/e41ABDgWdlw

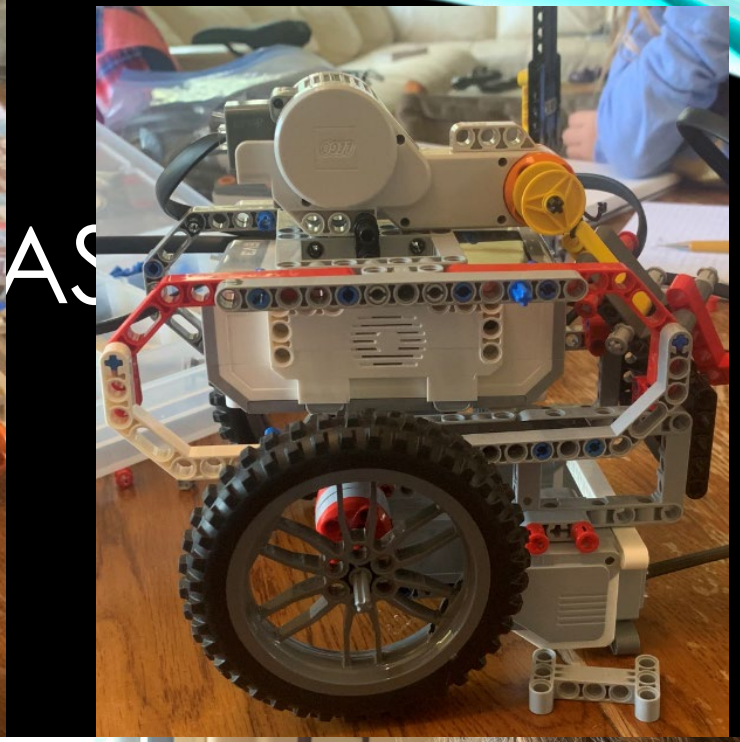
1. First Pass (uses attachment)
    1. Robot fits inside home base (25 pts.)
    2. Innovation (robot pushes innovation project to the bench) (20 pts)
    3. Bench(robot pushes down the bench) (10 pts)
    4. Back Rest(15 pts)
2. Second Pass (uses PID line following, PID straight and Ultrasonic Sensor)
    1. Step Counter (20 pts)
    2. Pass Through (15)
    3. Pull Up Bar (30 pts)

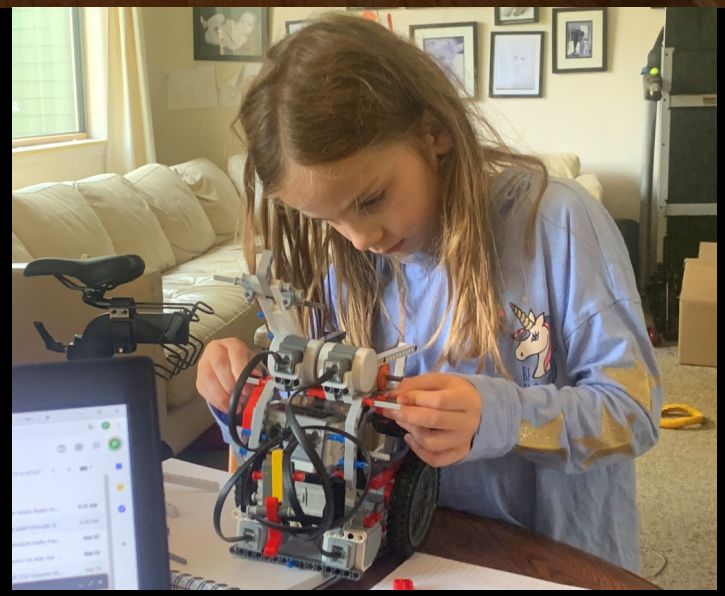
MISSION IMPOSSIBLE AUTOMATON (MIA)

- (2) large motors for driving (horizontal and low) for lower center of gravity
- Motorcycle tires for wheels to make robot faster, smoother
- (2) large motors for spatula and winch. Placed on top to reach the top of the bridge.
- Two light sensors, ultrasonic sensor and motor rotation sensors for navigation.
- Cage to hold motors and brick and sensors
- Brace to keep things "firm"
- A single attachment to complete the innovation, bench and back rest missions.

- Low motors
- Cage for brick





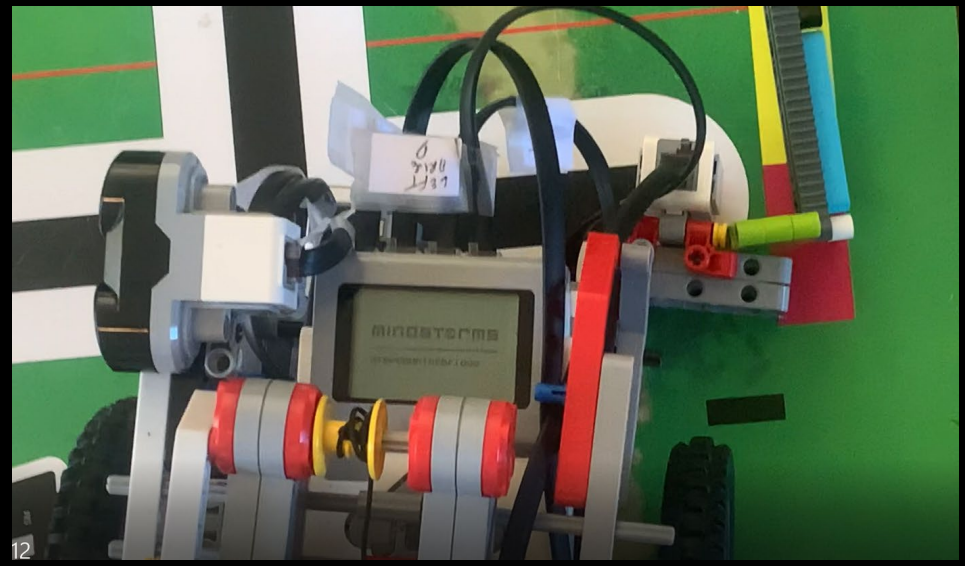


INNOVATION DESIGN/BENCH/BACK REST

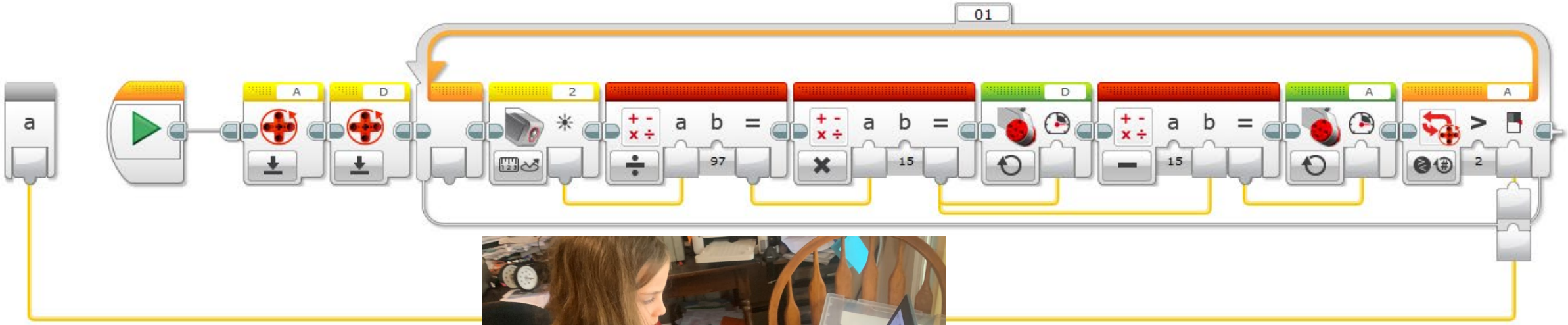
- Simple, removable pusher with a hook for back rest
- Innovation project looks like two spinners, representing project
- Simple out and back tank drive with some turns

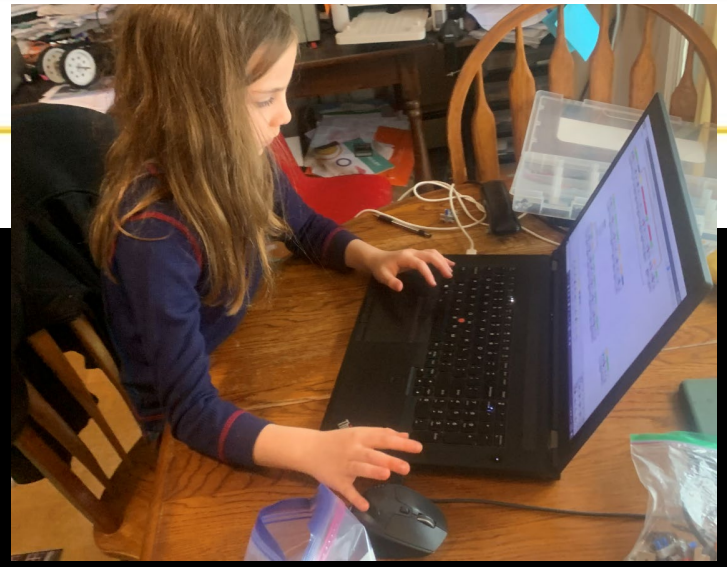
STEP COUNTER

- Many points, close to base, good way points, no need to return
- This was very hard because the dial could get stuck, the bar could slide out of position, the robot would drift
- Discovered that slow and steady works to keep ratchet mechanism from catching
- Lower power means that the motors tend to not have enough power to get to a specific rotation amount, so the motors need time as input to prevent getting stuck
- Needed a PID to keep robot straight
- Easily took 50+ trails to get it working!
- Hardware, software or person



LINE FOLLOW PID ALGORITHM





1. These blocks reset the rotation sensors for motors A and D. This is relevant because the next block uses the motor sensors for Sensor A to stop. If they were not reset then the loop might end too soon.

2. This block runs a loop that stops when the A rotation sensor is larger then the input parameter. This means that the program is able to be used repeatedly with different input values.

3. This block finds out how much light reflected back.

4. This block divide the reflected light by the maximum reflected light, 97.

5. This block multiplies the reflected light out of the max light by 15, I chose 15 because to be the maximum because it's not too fast, nor too slow. If the reflected power light is 97 then D would get 15 and A would get zero, if the light is 0 the vice versa.

6. This block plugs the light out of the max 97 multiplied by the max power into motor D.

7. this block subtracts 15 the max power from D to determine A.

8. This block plugs in the subtraction max power from D and plugs it into A.

9. This block has the parameter decide when the program stops.

Straight PID Algorithm Stepper

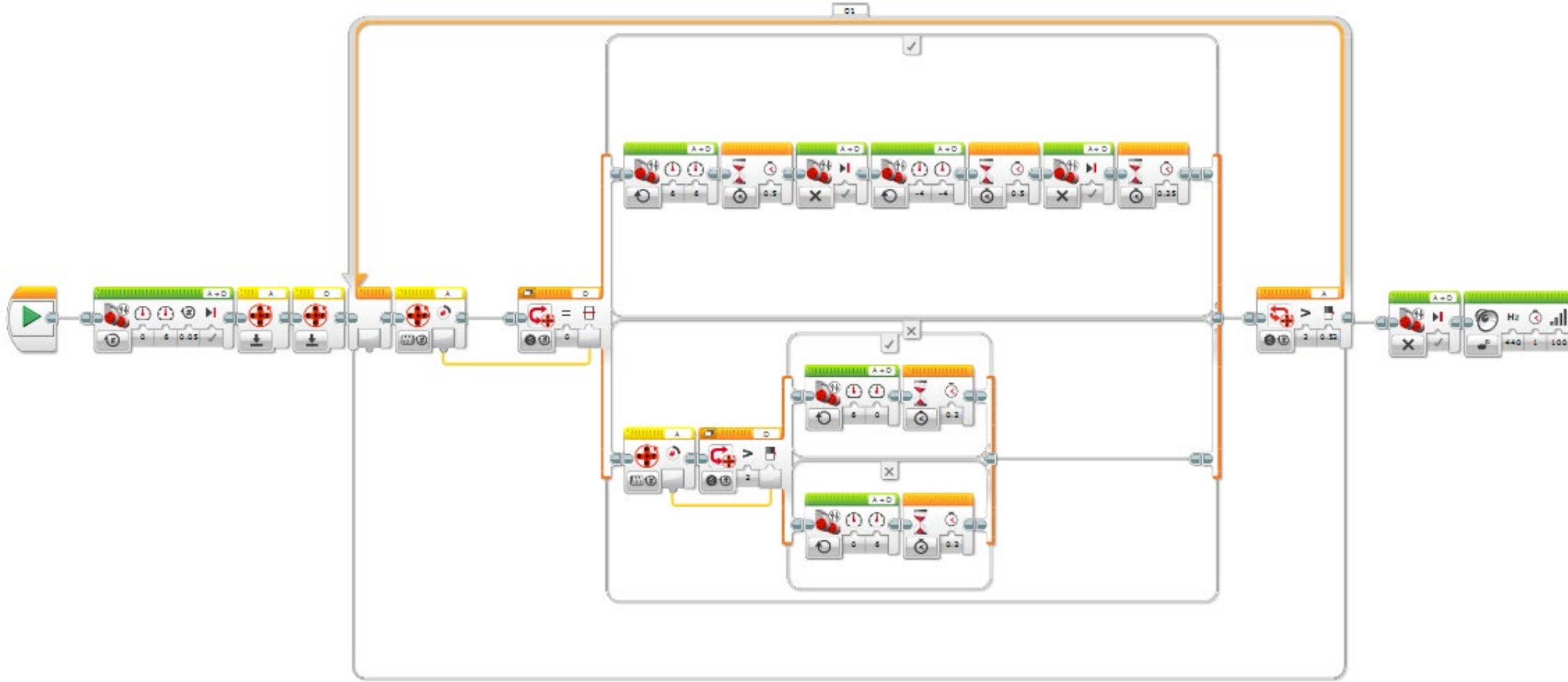


photo credit: Charles H. Willey

1. I discovered that the robot gets stuc on the flat of the step counter. That was because the was not going fora enough. To assess that problem we gave D power before the progra starts.

2. In order for robot to venture the right distance, the encoders have t be reset.

3. This block of code determines whet motors A and D are equal.

4 If the A D motors ARE equal then the robot drives forward for half a seco then drives back with less power for half a second, this causes it to take five 5 steps forward, 1 step back.

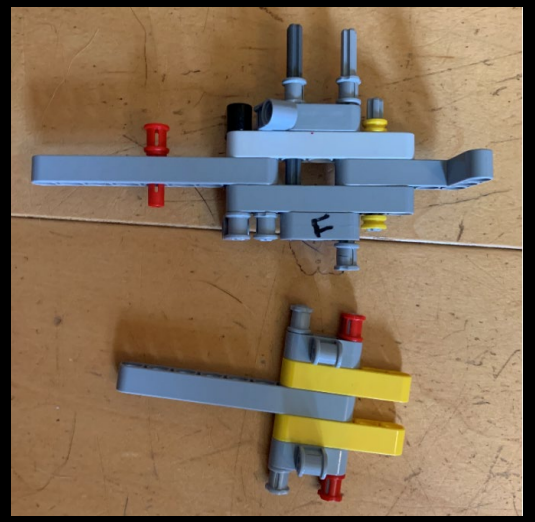
5. If the motors are not equal than these figures out which one has an larger value, then the blocks add power to the minority.

6. the loop determines when A's motor rotations reach .56 then stop

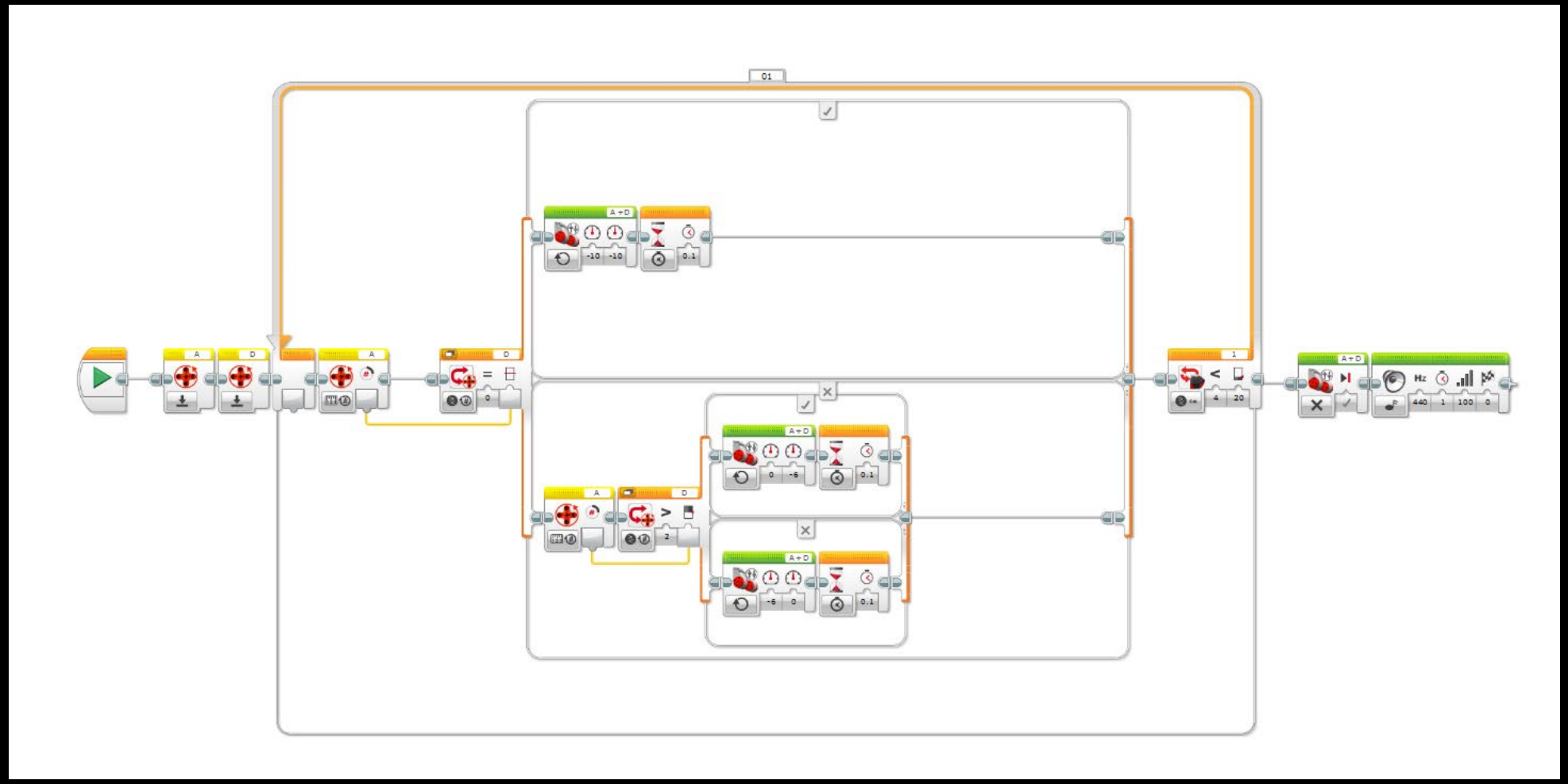
7. The robot makes a sound at the end so we know it's not stuck.

PASS THROUGH AND PULL UP BAR

- Many way points, high score, close to base, no need to return
- Set top height
- Needed two medium motors to place hook and winch robot
- Used many prototypes including scissors linkage
- Hook underwent many designs



TROLL-BACKWARDS PID UNDER BRIDGE

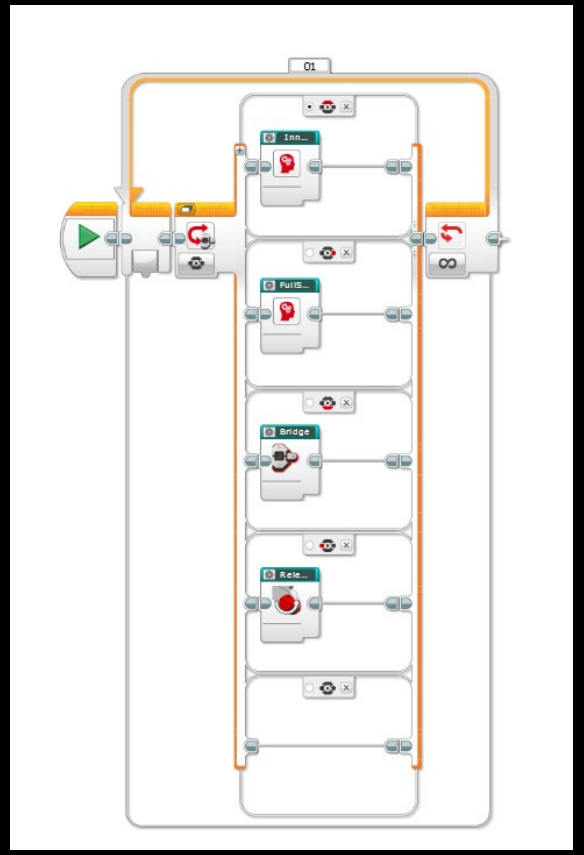


MASTER CONTROLLER

This allows me to select a specific program
Without having to search for a program on the Brick

Each mission is mapped to a specific button

Top is first mission (innovation/bench)
Right is second mission (stepper/bridge)
Bottom is just bridge (assuming stepper done)
Left is to release winch when done



AUDIO CUES

- Audio cues help me make decisions about whether or not the robot is stuck.
- First program uses audio countdown to remove the attachment and place robot.
- The stepper uses an audio signal to tell me if it is done or not.
- The troll uses an audio signal to let me know if it detected the bridge or not.