BrainSTEM Botz

10703Z

ENGINEERING NOTEBOOK

CHANGE UP • 2020-2021

Robotics Engineering Notebook

REC Foundation

VEX

team name: BrainSTEM Botz

team number: 10703Z

season: 2020 - 2021

start date: 4/27/20

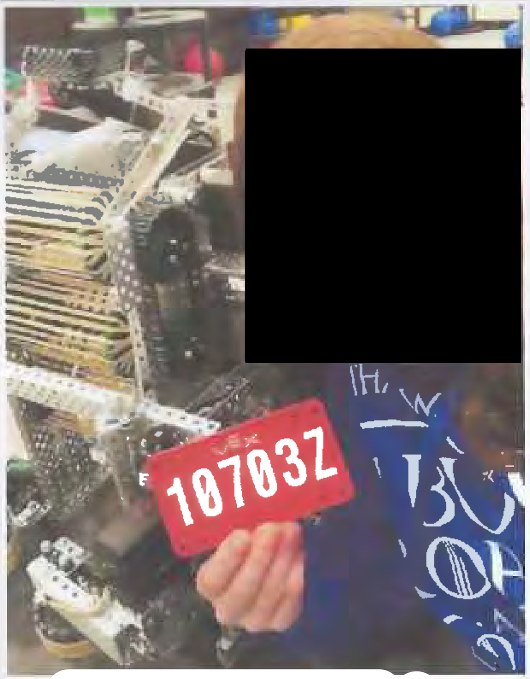
end date:

book number: of:

10
9
8
7
6
5
4
3
2
1 inch

Team Photo

Brainstorm



10703Z

BrainSTEM Botz

Hi, I'm Megan M , and this is my first season as a one-girl team! I'm 17 years old. I am homeschooled, and I am a senior this year.

This is my fourth season in VEX Robotics. I was on team 16859A for In the Zone and Turning Point, and I was on team 11656A for Tower Takeover.

I have experience journalling and programming, but building is new to me this season!

Team Profile

Color Coding:

- Working on the robot / programming
- Design Planning
- Resource Management
- Tournaments
- Goals

Robotics is all about iteration and improvement. This Design Cycle is what I follow as I build and modify my robot. This process is orderly and provides resources to look back on for future iterations.

Design Cycle

PROBLEM / CHALLENGE → BRAINSTORM → SELECT OPTION → BUILD / IMPLEMENT SOLUTION → TEST SOLUTION → PROBLEM / CHALLENGE

My Projects

project	date

Table of Contents

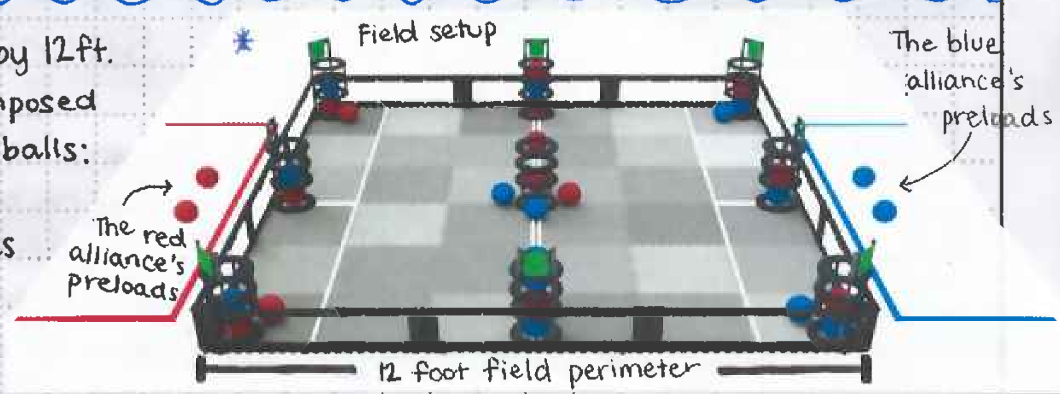
Continued →

My Projects

Continued in Book 2

THE GAME : CHANGE UP

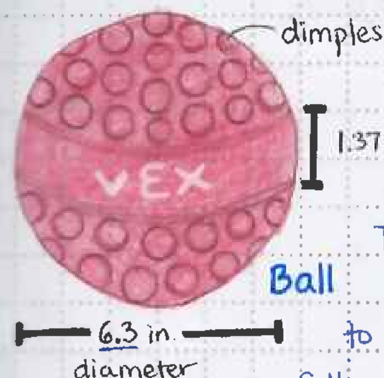
Change Up's 12 ft. by 12ft. playing field is composed of 9 goals and 32 balls: 16 red and 16 blue, including 2 preloads of each color.



THE GOAL:

The goal of Change Up is to work with your alliance partner to score more points than the opposing alliance in a 2 min. match (15 sec. of autonomous, 1 min. 45 sec. of driver control).

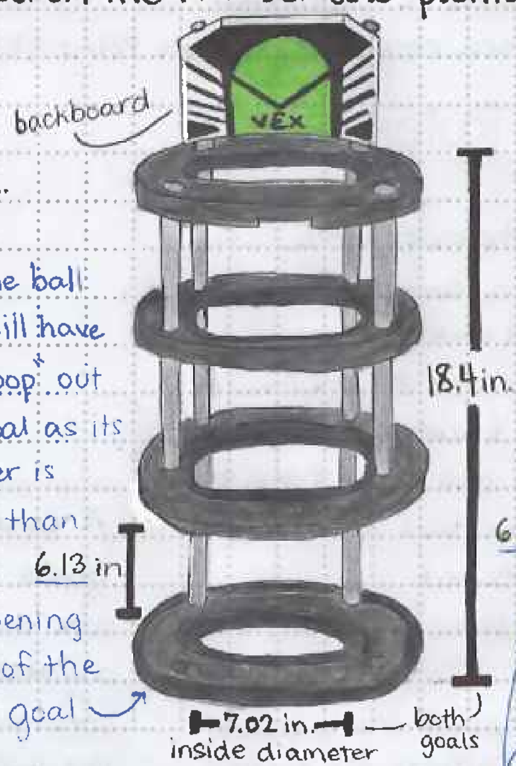
- Points are scored based on the number and position of balls in goals.

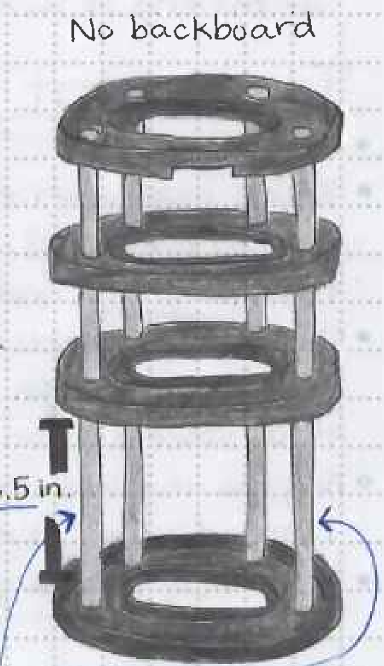


168 ± 10 grams (approx. 0.37 lbs.)

- Each ball is hollow, made of plastic, and covered in dimples.

The ball will have to "pop" out of the goal as its diameter is larger than the opening of the goal





These two extra poles are tricky with this goal. The ~~height of the opening~~ isn't an issue here.

- All of the goals except the one in the center of the field have backboards.

project designed by: witnessed by:

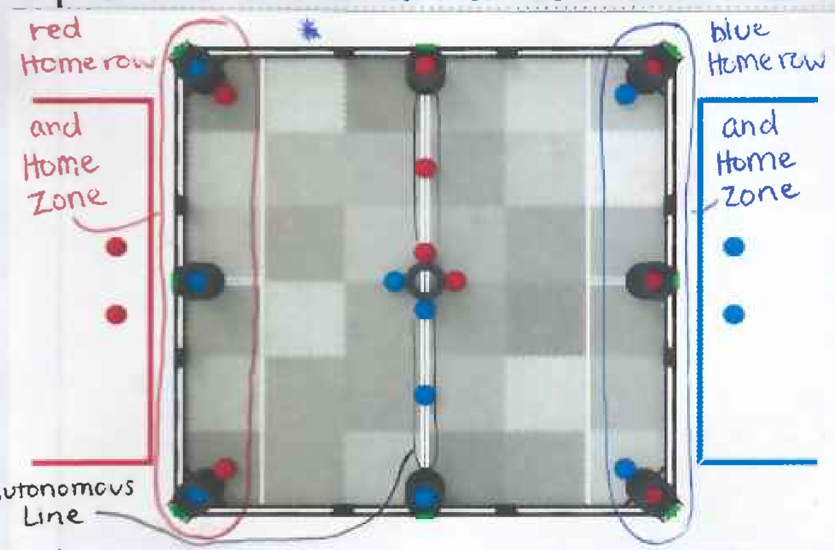
* Image from Change Up game manual

date: 4/27-28/20

Megan 4/27/20

4/27/20

THE GAME: CHANGE UP CONTINUED



Top view of Change Up's field setup

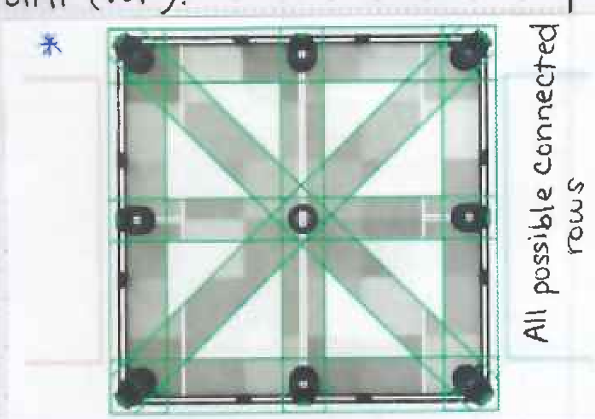
Scoring:

- Each ball scored in a goal is worth one point for its respective alliance.
- Each connected row is worth six points for the alliance that owns it.
- The alliance who scores the most points during auton. receives a six point bonus.

* An alliance who owns a connected row in their alliance home row at the end of auton. receives one Win Point (WP).

Important Rules:

- Robots must start the match fully within 18 in. by 18 in. by 18 in. and contacting their Home Zone and exactly 1 preload.
- Robots may not cross the Autonomous Line during auton. (neither can balls that start fully on one side of the field).
- Any ball that leaves the field will be returned where it exited.
- Balls may not be descored from the top of goals.
- A robot may not possess more than 3 of the opposing alliance's balls at a time.



A connected row is three goals in a straight line that are owned by the same alliance. There are 8 possible connected rows including the two alliance home rows.

project designed by: witnessed by:

* Images from Change Up game manual

date: 4/27-28/20

Megan M 4/27/20

4/27/20

THE GAME: CHANGE UP CONTINUED

Scoring Balls and Owning Goals

- The alliance whose color matches the color of the highest scored ball in a goal owns that goal.

Criteria for a ball to be considered scored in a goal:

- The ball must not be touching a robot of the same color as the ball.
- The ball must be fully or partially within the goal horizontally.
- The ball must be fully within the vertical volume of the goal.
- The ball must not be contacting the foam tiles outside the goal.

*

↔ Owned by the Blue alliance

↔ All balls are scored

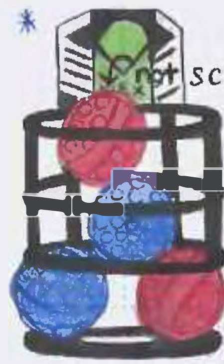
*

not scored

Owned by the Red ↔ alliance

All three red balls ↔ are scored

*



not scored

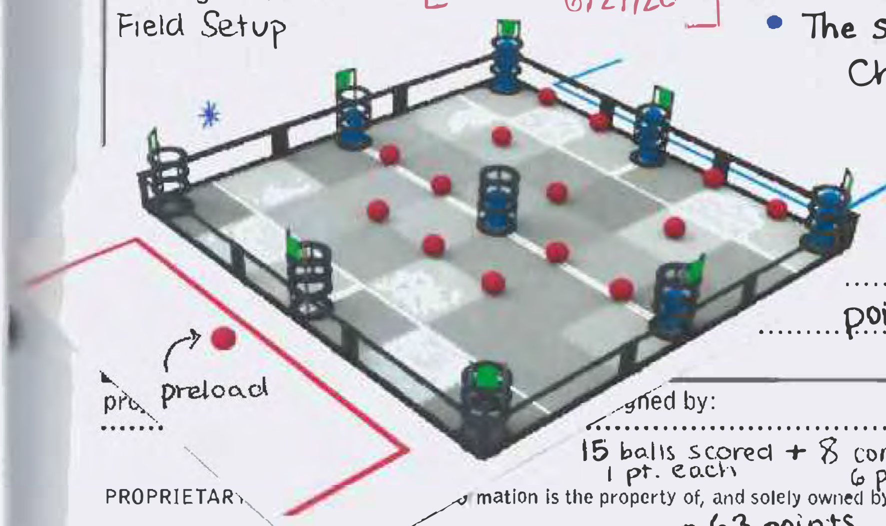
↔ Owned by the Blue alliance

↔ Both of the two balls on the bottom are scored (as is the middle blue ball)

[*See page 34! 6/27/20]

Change Up Skills Field Setup

*



preload

Robot Skills Challenge

- The scoring for the Robot Skills Challenge is the same as in a match; however, the initial setup is different.
- The highest possible score for one run is 63 points (126 points total).

changed 6/15/20 (new layout)

pro... ...gned by: witnessed by:

15 balls scored + 8 connected rows
1 pt. each 6 pts. each
= 63 points

date: 4/27-28/20

Megan M 4/27/20

* Images from Change Up game manual

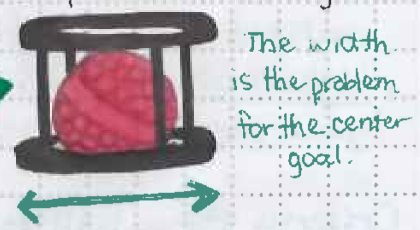
4/28/20

ANALYZING CHANGE UP

Game Challenges:

- Robots will need to be able to "pop" the ball out of all the goals except the center one.

close ups of the bottom of goals



- In order to descore from the center goal, robots will have to reach around the two additional poles at the bottom opening.
- The height of the goals is just over the 18" by 18" by 18" size limit at the start of the match. In order to score, robots must either launch the balls or lift up to place them in the goals.
- Descoring is necessary to achieve the highest possible skills score.

Goal height = 18.4 in.
* see page 1 for more dimensions

Robot Design Challenges:

- Space must be used wisely as the balls are large game elements.
- A high speed drivetrain is needed to reach goals and balls quickly (the field is relatively open).
- The robot must be able to score and descore all goals efficiently.
- The robot needs to be able to score without relying on the backboard (as the center goal does not have one).
- The robot needs to be able to hoard opponents' balls.

Strategy:

- The center goal is crucial - it is required for 4 of the 8 possible connected rows.
- With so few available game elements, hoarding will severely limit the opponent from scoring.
- Maximizing the number of connected rows greatly boosts your skills score (rather than the number of balls scored)

project designed by: witnessed by:

date: 4/28-29/20

Megan M 4/29/20

ANALYZING CHANGE UP CONTINUED

Autonomous Strategy:

- The Win Point for completing the home row in auton. is huge - this will make a team stand out, not to mention improving your ranking - so it's really important to have several different programs to work well with any alliance.
- A way to earn points in auton. is to descore the opponent's balls in the goals on your side - this has the same effect as scoring your balls.
- Save precious seconds by taking advantage of the preload and the ball sitting in front of the corner goal nearest you!

Sensors:

*

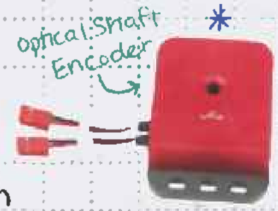

The Vision Sensor could be used to line up to a goal using the backboard (green), or it could be used to sort balls by color.

Line Tracker *


The Line Tracker could follow the white tape on the field to a goal. It could be used to determine if a ball is in the robot's claw / intakes / lift, etc.

*


The Potentiometer could be used to stop the motion of a lift or pivoting mechanism at exactly the right spot for scoring.

Optical Shaft Encoder *


The Optical Shaft Encoder is perfect for measuring distances on tracking wheels without slippage, etc.

*


*


The Bumper Switch and Limit Switch are perfect for stopping motion at a specific point. They could be used to signal hitting a wall, reaching a goal, or as a safeguard to protect electronics or a gear from grinding.

Inertial Sensor *


The Inertial Sensor is valuable for more accurate turning. It can be used to prevent tipping (however, this probably isn't necessary in Change Up as the goals are only slightly taller than the stowed robot).

project

designed by:

witnessed by:

* Images from vexrobotics.com

date: 4/28-29/20

Megan M 4/29/20

4/30/20

DESIGN CHECKLIST

After reading the game manual and analyzing Change Up, I compiled this list of "musts" and "wants" for the robot design. ~M. 4/30/20

Robot Design Constraints:

- Robot must be fully within 18" by 18" by 18" when stowed.
- A maximum of 8 V5 Smart Motors may be used.
- Any incorporated sensors requiring three wire ports are limited to a total of 8 ports (A-H on the V5 Brain).
- No more than a single 12" by 24" sheet up to 0.070" thick of polycarbonate may be used.

My Design Requirements for a Competitive Robot:

- Drivetrain geared for speed
- Ability to descore any goal
- Ability to hoard opponent's balls in accordance with <SG8> (3 max. at one time)
- Ball capacity is greater than 1
- Ability to score without relying on the backboard

My Design Preferences for a Robot in Change Up:

- 15" by 15" base size, 17" height
- Very efficient at scoring/descoring
- Ability to both score and descore from the same side of the robot (no turning required)
- Ball capacity is at least 3 (excluding the hoarding balls)
- Ability to strafe
- Ability to score and descore simultaneously
- Space availability for well-positioned vision sensor, tracking wheels, and other sensors as needed (see page 5)

Megan M 4/30/20

GOALS FOR CHANGE UP

GOALS FOR THE ROBOT:

- [ ] Design that demonstrates creativity
- [ ] Stability against defense and competition
- [ ] Efficiency in scoring and descoring all goals

GOALS FOR PROGRAMMING:

- [ ] Incorporation of sensors that improve the robot's performance
- [ ] Macros that increase the efficiency and accuracy of scoring and descoring in driver control
- [ ] Consistency in auton. and macros
- [ ] Routines that complement every alliance partner

GOALS FOR SKILLS:

See page 34 for updated skills goals with the new field layout 6/27/20

- [ ] Achieve 63 points (the highest possible score) in driver by the TN State Championship
  - ↳ requires scoring 15 red balls (1 point each), descoring 3 blue balls (to make room for red), and making 8 connected rows (6 points each)
- [ ] Achieve 57 points in programming skills by the TN State Championship
  - ↳ requires scoring 9 red balls (1 point each), descoring 3 blue balls (to make room for red), and making 8 connected rows (6 points each)

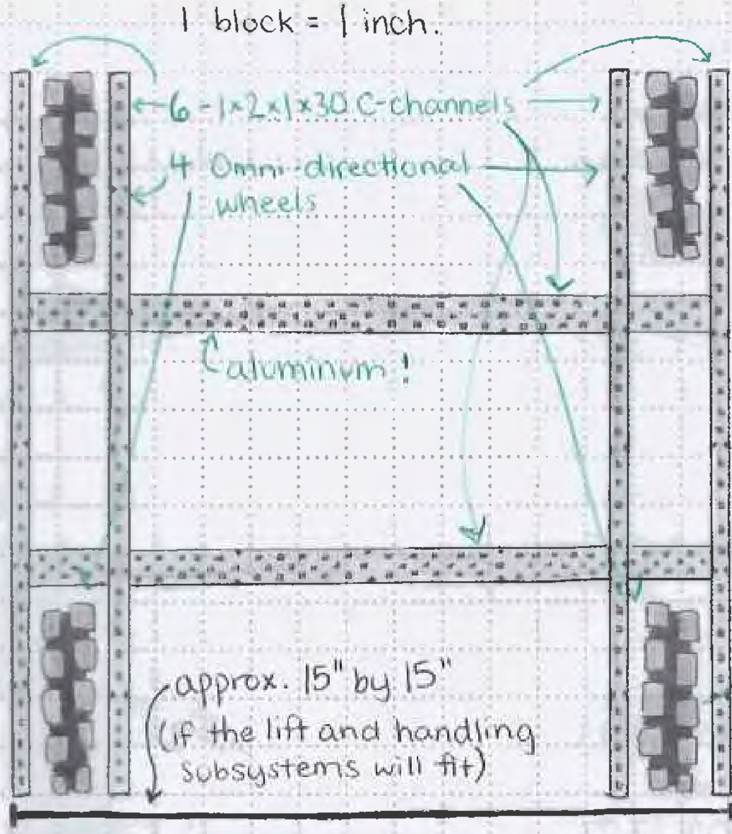
* This would equal a total score of 120 points (126 points is the highest possible total score)

Megan N. 4/30/20

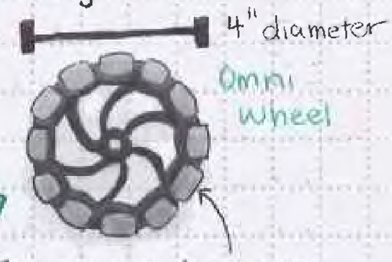
5/4/20

DRIVETRAIN OPTIONS

Drivetrain Option #1 : Omni Wheel Tank Drive



- Motors - attach directly to each wheel, or gear up for extra speed
- 2 or 4 motors, 200 rpm cartridges



- The rollers increase maneuverability, but causes the robot to be able to be pushed sideways easily.

Pros of Omni Wheel Tank Drive:

- Simple to build
- Option to use 2 or 4 motors depending on lift and handling systems
- Requires minimal space
- Ability to turn quickly

Cons of Omni Wheel Tank Drive:

- Cannot strafe without a fifth wheel and additional motor mounted in the center of the base
- Can be pushed sideways easily
- Middle speed among drivetrains

project designed by: witnessed by:

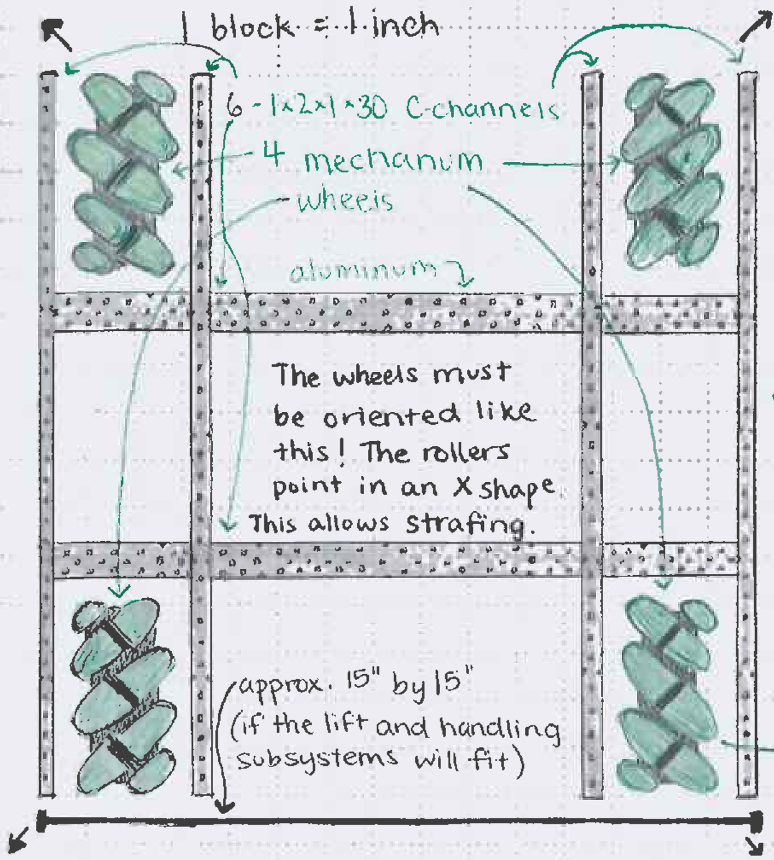
date: 5/4-5/20

Megan M 5/4/20

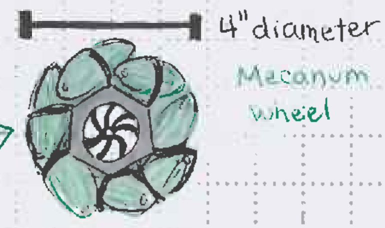
-/20

DRIVETRAIN OPTIONS CONTINUED

Drivetrain Option #2: Mechanum Drive



- Mechanum wheels require 1 motor per wheel - attach directly to each wheel, or gear up for extra speed
- 4 motors, 200 rpm cartridges
- Mechanum wheels are about twice the width of omni wheels, leaving less space for lift and handling mechanisms between them.



Pros of Mecanum Drive:

- Ability to strafe
- Decently simple to build
- Cannot be pushed around as easily as other drive-trains

Cons of Mecanum Drive:

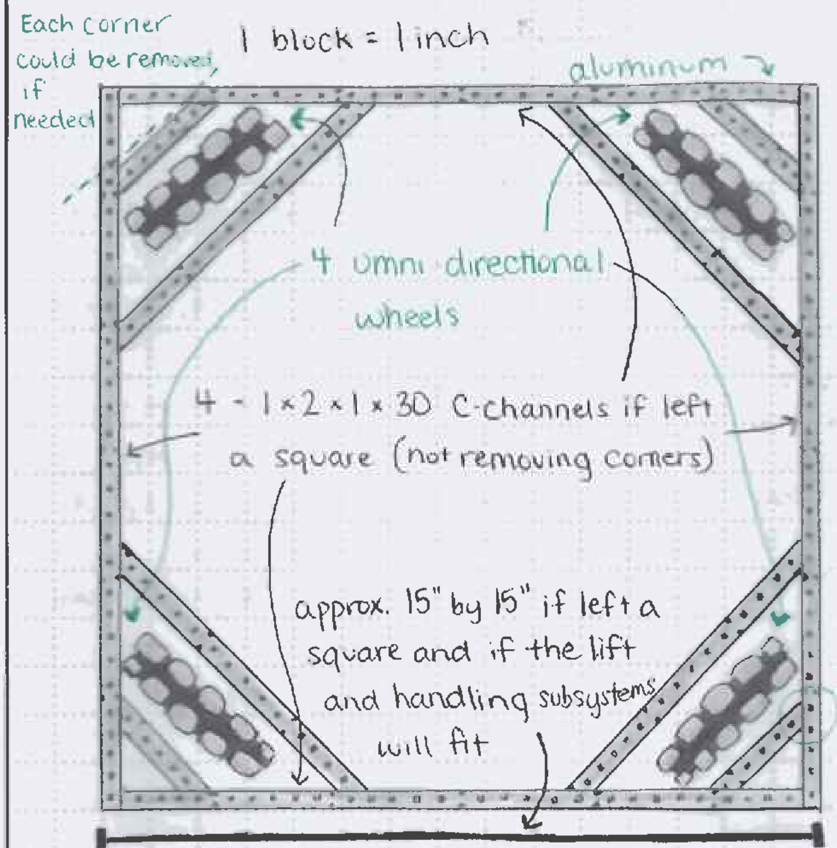
- Takes up more space due to the width of each wheel
- Requires 4 motors
- Lower speed than other drivetrains
- Slower turning

Megan M 5/4/20

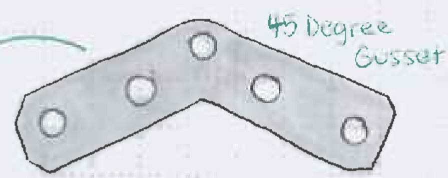
5/4/20

DRIVETRAIN OPTIONS CONTINUED

Drivetrain Option #3: Omni Base X Configuration (X Drive)



- X Drive requires 1 motor per wheel - attach directly to each wheel or gear for extra speed if space allows.
- 4 motors, 200 rpm
  Because the wheels are mounted at a 45° angle, X Drive is ≈1.4 times faster than a normal tank drive.
  ↳ Torque is decreased by the same amount



- 45 Degree Gussets are a great way to stabilize X-Drives.

Pros of X Drive:

- High speed
- Ability to strafe
- Ability to turn very quickly

Cons of X-Drive:

- Takes up a large amount of space
- Requires 4 motors
- Can be pushed around easily

project designed by: witnessed by:

date: 5/4-5/20

Megan M 5/4/20

5/4/20

SELECTED DRIVETRAIN

The criteria for this decision matrix came from the design constraints and requirements (see page 6) and the design challenges in Change Up (see page 4). ~M.M. 5/5/20

Drivetrain Options

Criteria (least to most important)	Scale	Omni Wheel Tank Drive (see page 8)	Mecanum Drive (see page 9)	Omni Base X configuration (X drive) (see page 10)
Simplicity to build	0 to 3	3	2	0
Ability to turn quickly	0 to 3	2	1	3
Low motor usage	0 to 3	2	1	1
High speed	0 to 5	4	2	5
Ability to withstand defense	0 to 5	2	4	0
Takes up little space	0 to 5	5	3	0
Ability to strafe	0 to 7	0	7	7
Total Score:		18	20	16

5/5/20 M.M.

- Based on the above decision matrix and the criteria specifically chosen to fulfill the game and design challenges (see page 4), mecanum wheels are the best drivetrain option.
- While the size of the robot's base may change due to the lift and handling systems, the drive system will remain 4 mecanum wheels, 4 200rpm motors, and any gears / sprockets and chain unless it becomes clear that another drivetrain would work better.

project

designed by:

witnessed by:

date: 5/4-5/20

Megan M 5/5/20

5/6/20

*Citations in this color

BRAINSTORMING and RESEARCH

In brainstorming various design options, I researched what other teams are doing to get ideas and learn anything I can. ~ M.M. 5/6/20

Change Up Robot in 24 Hours Reveal - MARC by Leifis on the VEX Forum

I like how this team has the ability to hoard (holding the balls below the tray). They also have a good ball capacity.

99999V C12D Reveal by Xenon27 on the VEX Forum

Having long intakes like these really helps to pull the balls out of the goals. This conveyor system is simple, but effective.

210Y VEX Change Up R13D Reveal by 210Y Robotics on YouTube

The hood is on a ratchet - I think that's a pretty cool idea! Again, long intakes with flaps. Instead of using tread with flaps for the conveyor, this team used rubberband rollers. It seems to work very well.

Vision sensor

5278C C17D Reveal by 5278C on the VEX Forum

This team uses a vision sensor to detect the color of the ball and to automatically sort it accordingly.

Long intakes seem to be popular this season; I wonder how they'll work with the middle goal?

project

designed by:

witnessed by:

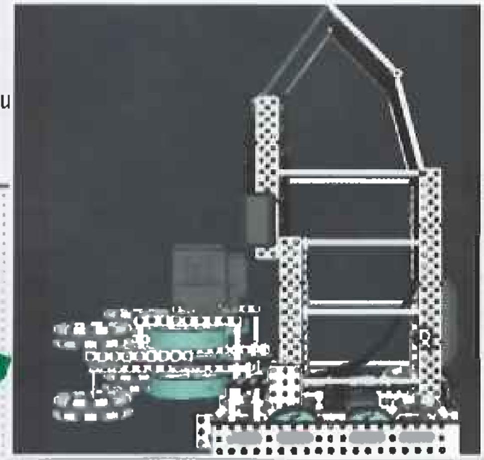
* citations in this color

date: 5/6-8/20

Megan M 5/7/20

7-LA VEX Change Up Ri3D Reveal on Youtube and from Helix on the VEX Forum

This team uses omniwheels rather than tread with flaps for intakes. It works well.



My cad reveal by CodeMaster on the VEX Forum

Also uses omni-wheels for intakes, spaced apart so as to pick up the ends of the ball (smaller)

Robot Concept - VEX Change Up on YouTube and from Codec on the VEX Forum

These intakes are different - probably have a better chance of not breaking; but how well would they work for descoring?

Added 5/15/20 Megan Miles

5090X Change Up Early Season Reveal on Youtube and from TaranMayer on the VEX Forum

This robot has a trapdoor to discard unwanted balls - but no hoarding bin. Perhaps some kind of trapdoor could release balls into a holding area?

Added 5/25/20
Megan

Change Up VEX YNOT Early Season Reveal 2020-2021 on Youtube

I really like the conveyor system on this robot. Also uses rubber bands instead of tread with flaps.

Almost all the robots I found in my research were "Snailbots". I think this has the potential to be a really good design, but it needs some upgrades (hoarding capabilities?) ~ M. 5/8/20

project

designed by:

witnessed by:

* citations in this color

date: 5/6-8/20

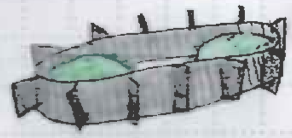
Megan M 5/8/20

5/11/20

INTAKE DESIGN OPTIONS

Intake Design Option #1: Rollers

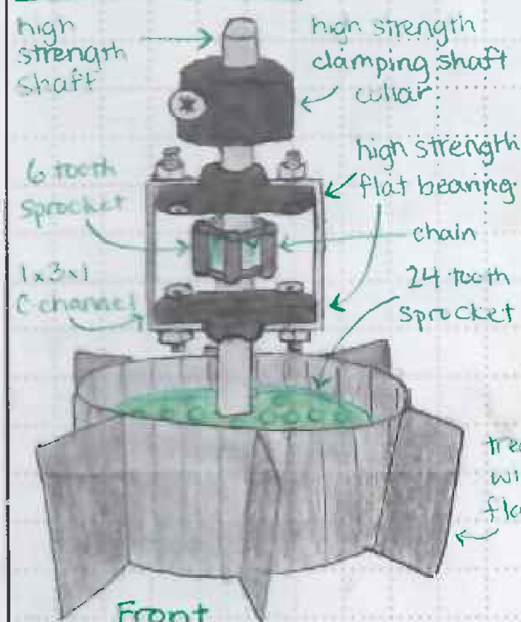
Many teams use long intakes like this, but it takes up a large amount of space and the chain could break.



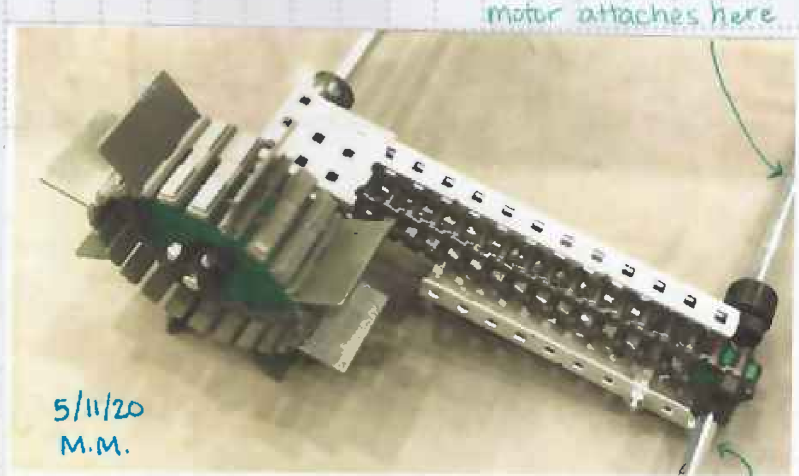
~M.M. 5/11/20

A single sprocket with tread would be a lot less likely to break (not to mention take up less space).

PROTOTYPING



motor attaches here



5/11/20 M.M.

Side view

attaches to the robot here, folds to fit within size limit

Pros of Intake Rollers:

- Potential to descore balls faster than other designs
- Very popular design, so resources available if help is needed
- Structure is very stable if high strength shaft is used (as shown above).

Cons of Intake Rollers:

- Take up large amount of space when stowed
- Possibility of tread breaking
- Must somehow open to fit around the middle goal
- Must be strong enough to "pop" the ball out of the outside eight goals (trade off of speed for torque)

project

designed by:

witnessed by:

date: 5/11-15/20

Megan M 5/11/20

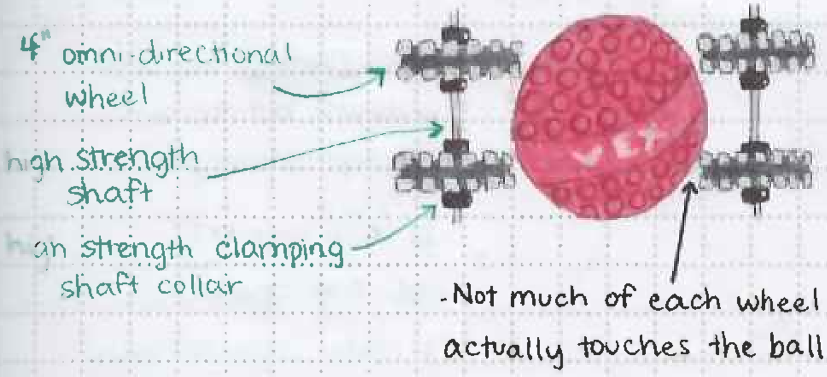
5/12/20

INTAKE DESIGN OPTIONS CONTINUED

Intake Design Option #2: Omni Wheels

This design seems to be less popular than long intakes with tread, but I did see several reveals in my research that uses this (see pg. 13) ~M.M. 5/12/20

PROTOTYPING



This concept works on the principle that the top and bottom of the ball is narrower than the diameter.

Pros of Omni Wheel Intakes:

- Easier to build than other designs
- Good use of resources (already have four omni wheels and don't have another use for them)
- Stable and not easy to break

The intake rollers and the omni wheel intakes are similar in a lot of ways; certain problems come with both - especially when trying to descore the middle goal.
~M.M. 5/12/20

Cons of Omni Wheel Intakes:

- Difficult to fit within the size limit and around corner goals
- Less grip on the balls than other designs
- Must open somehow to fit around the middle goal
- Must be strong enough to "pop" the ball out of the outside eight goals (trade off of speed for torque)

Megan P [redacted] 5/12/20

5/13/20

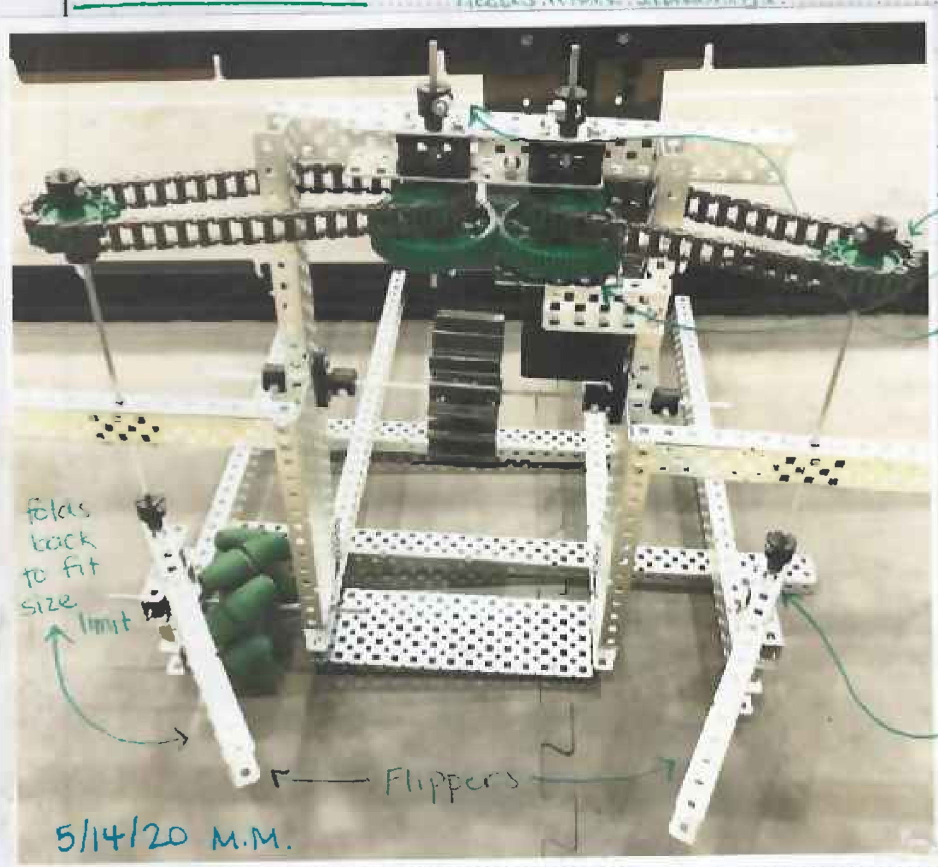
INTAKE DESIGN OPTIONS CONTINUED

Intake Design Option #3 : Flippers

This design is inspired by the concept of an inverse pinball machine. It offers solutions to several of the problems that come with intake rollers and omni wheel intakes. ~M.M. 5/14/20

PROTOTYPING

needs more stability!



This prototype has some problems, but the overall concept shows promise.

12 tooth sprocket

60 tooth gears

1 motor attaches here

This design really needs two motors, high strength shafts, and to be geared for torque. If this design is chosen, these changes will need to be made.

lock bars so the flipper pivots

Pros of Flippers:

- Multiple ways to descore any goal
- Opens naturally to descore the middle goal quickly
- Can easily have enough torque to "pop" a ball out of the goal
- Not easy to break, takes up little space, very unique design

Cons of Flippers:

- Spacing is challenging with a high strength shaft
- Must be built around the front wheels
- Potential problem keeping the flippers in sync when needed

project

designed by:

witnessed by:

date: 5/11-15/20

Megan M 5/14/20

5/15/20

SELECTED INTAKE DESIGN

The criteria for this decision matrix came from the design constraints and requirements (see page 6) and the design challenges in Change Up (see pg. 4). ~M.M. 5/15/20

Intake Design Options

Criteria (least to most important)	Scale	Intake Rollers see page 14	Omni Wheel Intakes see page 15	"Flippers" see page 16
Uniqueness	0 to 3	0	0	3
Simplicity to build	0 to 3	2	3	3
Ability to fit within the size limit	0 to 5	1	0	3
Few potential problems	0 to 7	2	2	5
Ability to descore all goals	0 to 7	6	5	7
Efficiency	0 to 7	6	6	5
	Total Score:	17	16	26

5/15/20 M.M.

- Based on this decision matrix and the potential that this design showed in prototyping (see previous page), flippers are the best intake design.
- Further prototyping will be necessary to make sure the flippers will work well with the rest of the robot design.

project designed by: witnessed by:

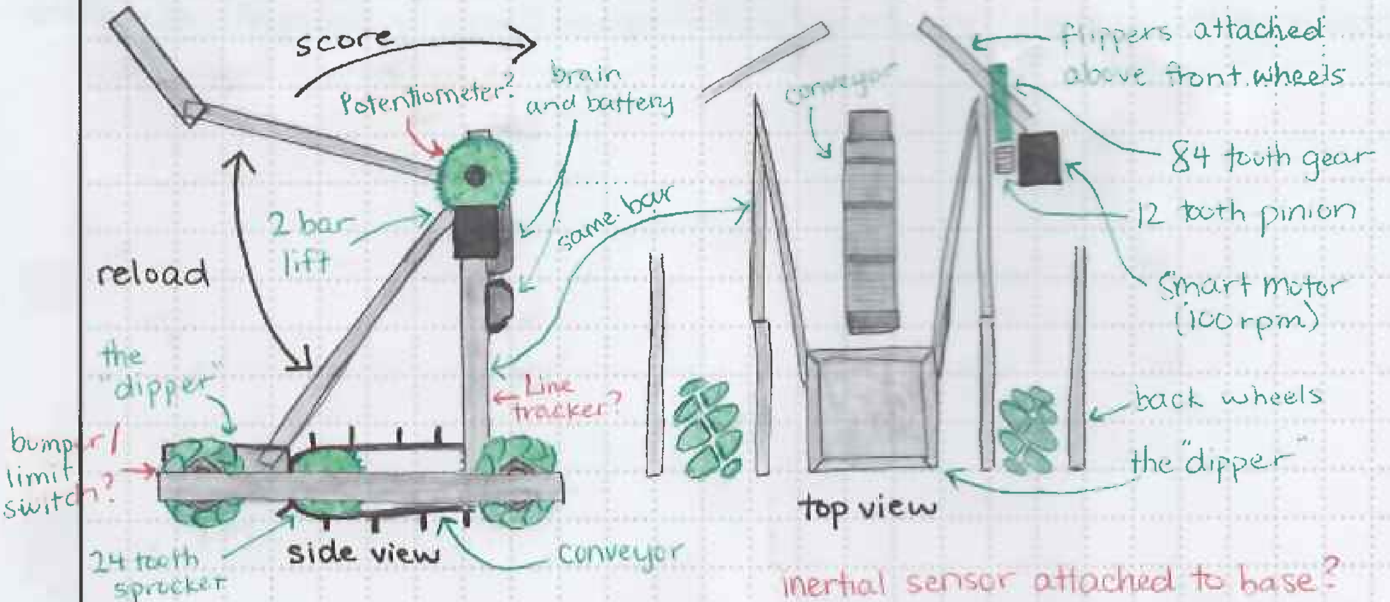
date: 5/11-15/20

Megan M 5/15/20

5/18/20

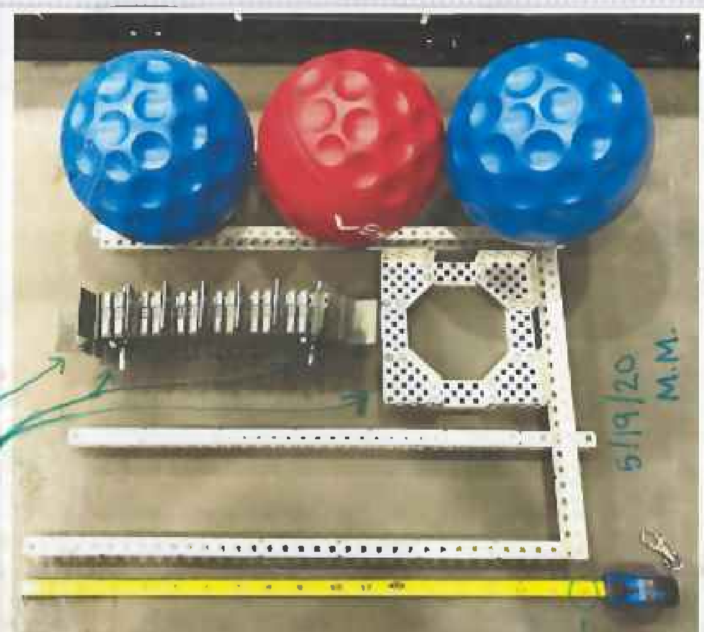
LIFT & HANDLING DESIGN OPTIONS

Lift and Handling Design Option #1 : "Big Dipper"



PROTOTYPING

Motor breakdown :
4 motor drive (200 rpm)
2 motor intake (100 rpm + geared)
1 motor conveyor (100 rpm ?)
1 motor lift (100 rpm and 1:7)



Ball capacity is not ideal in this design - only two can be held securely and no hoarding. ~M.M. 5/18/20

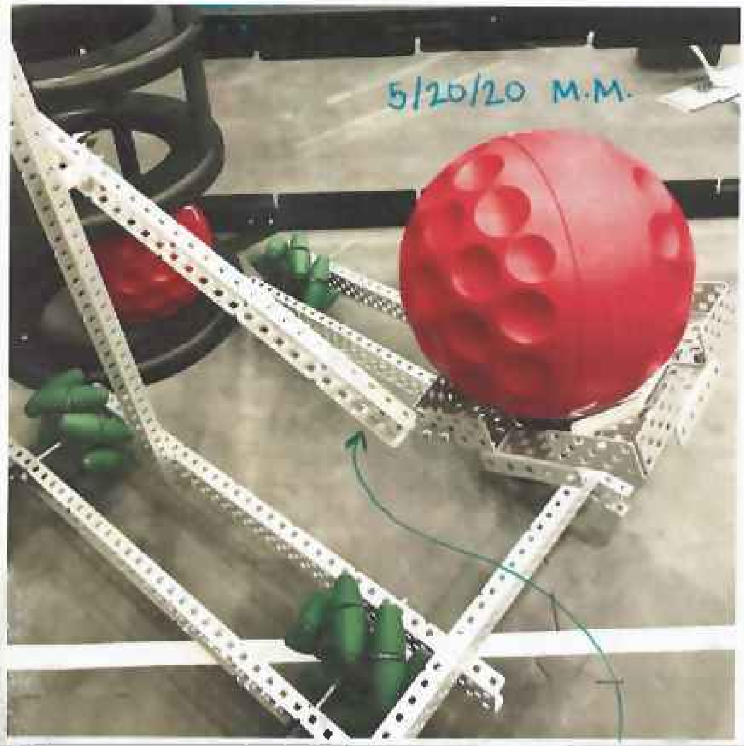
Megan M 5/18/20

LIFT & HANDLING DESIGN OPTIONS CONTINUED

"Big Dipper" Continued

Problems identified through prototyping:

- The robot is very front heavy when the dipper's arm is raised.
- There is no space to directly attach the four drive motors to the wheels.
- The dipper is attached at a bad angle for scoring - the ball falls out before it reaches the goal.
- Space is very tight everywhere
- Not very many sensors can be easily incorporated



The ball falls out when the arm is raised slightly higher than this (not into the goal)

Pros of "Big Dipper":

- Unique design
- Two available motors for the intakes
- If the ball didn't fall out of the "dipper", it should be consistently accurate at scoring
- Good spot for the brain and battery

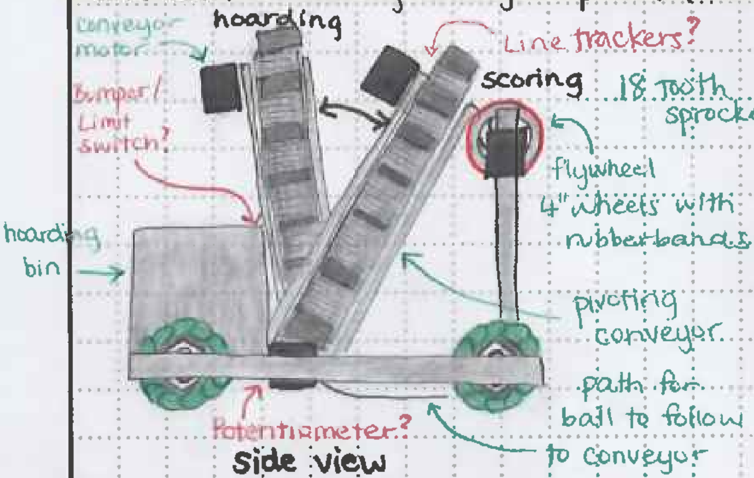
Cons of "Big Dipper":

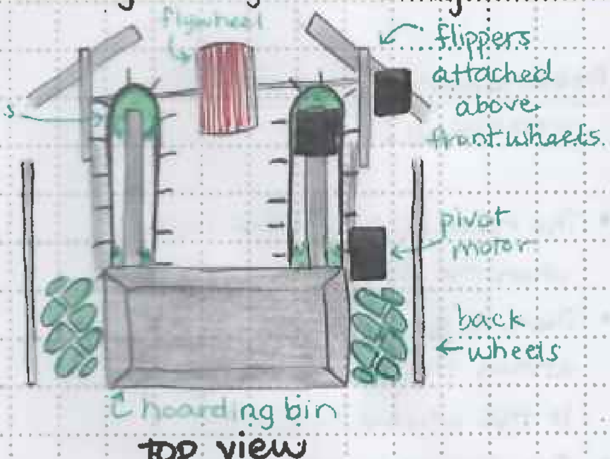
- Low ball capacity
- No hoarding capability
- Extremely inefficient
- Lots of very difficult issues to solve
- Even more potential challenges (such as getting the ball up onto the conveyor belt)

5/21/20

LIFT & HANDLING DESIGN OPTIONS CONTINUED

Lift and Handling Design Option #2: Pivoting Conveyor with flywheel

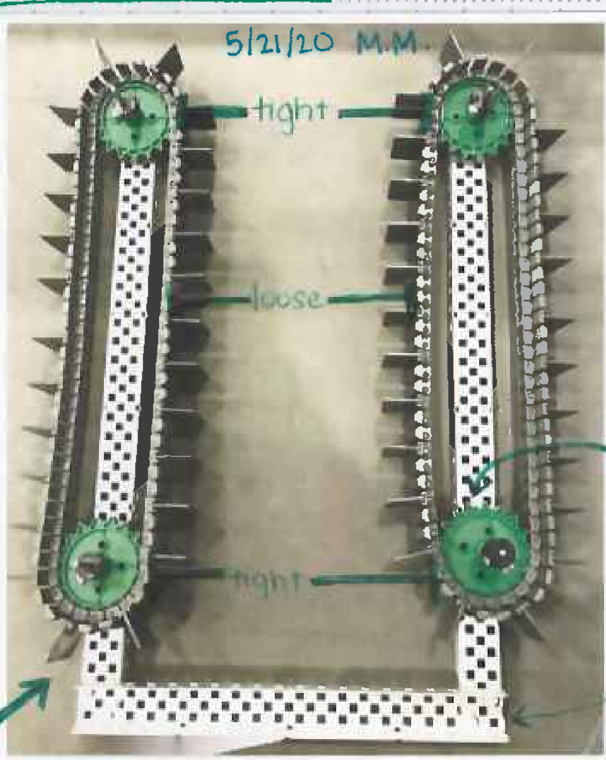




drive motors?

inertial sensor attached to base?

PROTOTYPING



18 tooth sprockets

The conveyor pivot would attach here

Motor breakdown:
- 4 motor drive (200 rpm)
- 1 motor conveyor (100 rpm?)
- 1 motor conveyor pivot (100 rpm + geared?)
- 1 motor intake (100 rpm + geared)
- 1 motor flywheel (600 rpm)

A flywheel seems a bit excessive to shoot the ball only a few inches, but just the conveyor probably isn't enough to get the ball in the goal. ~M.M. 5/21/20

The ball's path needs to be smooth all the way up - and it's not.

project

designed by:

witnessed by:

date: 5/21-23/20

Megan M 5/21/20

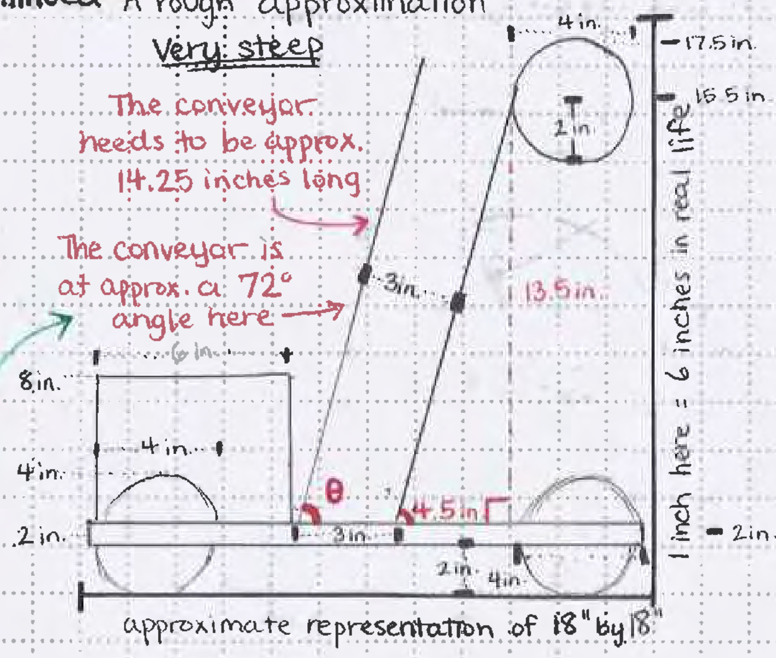
5/22/20

LIFT & HANDLING DESIGN OPTIONS CONTINUED

Pivoting Conveyor with Flywheel Continued

A rough approximation

Very steep



The conveyor needs to be approx. 14.25 inches long

The conveyor is at approx. a 72° angle here

Problems identified through prototyping:

- Only one motor and very little space for gearing is available for the intakes.
- Difficult to get the balls from the intakes into the conveyor
- Very steep path for the ball and no room for guard rails (potential for balls to fall out)
- Low ball capacity
- Hoarding is very difficult at the angle the conveyor is at (vertical)
- Spacial issues throughout

$a^2+b^2=c^2$

$(4.5)^2+(13.5)^2=c^2$

$c^2=202.5$

$c=14.23$ inches

Length of conveyor

$\tan\theta = \frac{opp}{adj}$

$\tan\theta = \frac{13.5}{4.5}$

$\theta = \tan^{-1}\left(\frac{13.5}{4.5}\right)$

$\theta = 71.57°$

Angle to reach flywheel

This means the conveyor has a 2 ball capacity

Pros of Pivoting Conveyor with Flywheel:	Cons of Pivoting Conveyor with Flywheel:
• Unique design	• Low ball capacity
• Ability to hoard	• Lots of very difficult problems to solve
• Relatively easy to fit within the size limit	• High possibility of further problems down the road that could break the design
• Potential to be fairly efficient	

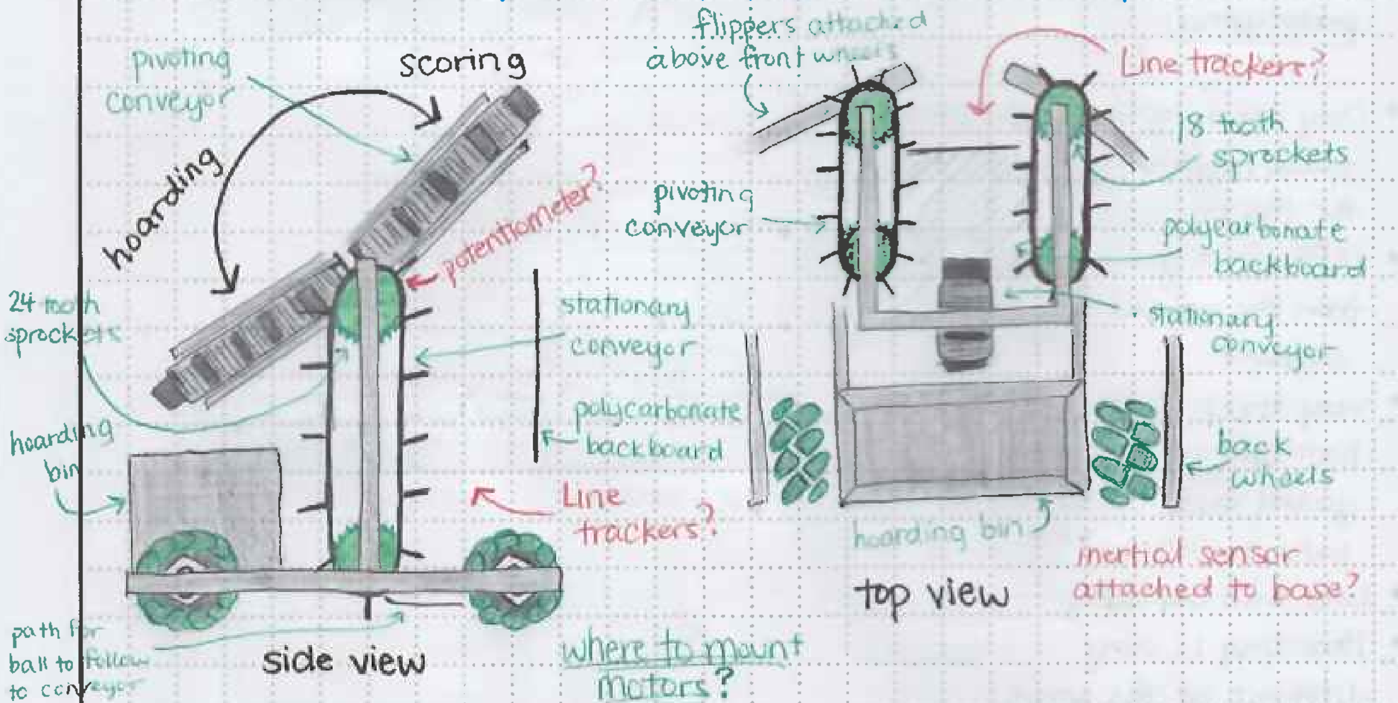
Megan M 5/23/20

5/25/20

LIFT & HANDLING DESIGN OPTIONS CONTINUED

Lift and Handling Design Option #3 : Pivoting Conveyor 2.0

This design is an improved version of Option #2 (see pages 20-21). It addresses some of the problems of a (hopefully) sound concept. ~M.M. 5/25/20



PROTOTYPING

Motor Breakdown:

4 motor drive (200rpm)
1 motor stationary conveyor (100 rpm?)
1 motor pivoting conveyor (100rpm?)
1 motor conveyor pivot (100 rpm + geared?)
1 motor intake (100 rpm+geared)

In attempting to prototype this design, I found more problems than previously anticipated, rendering this design practically unworkable as it is now.
~M.M.
5/26/20

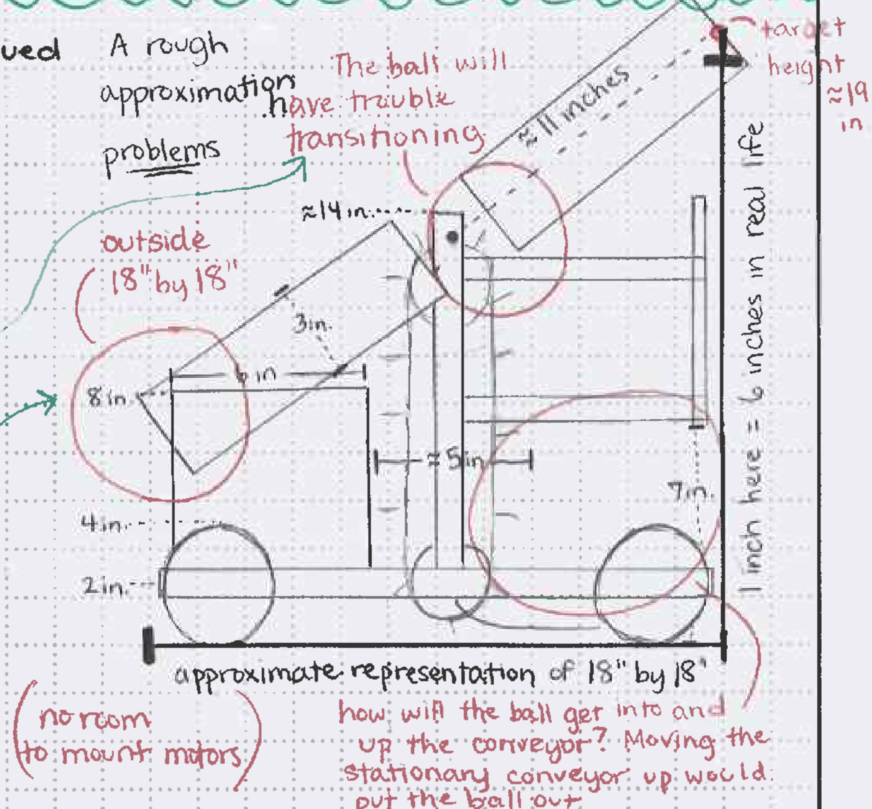
5/26/20

LIFT & HANDLING DESIGN OPTIONS CONTINUED

Pivoting Conveyor 2.0 Continued

Problems identified through prototyping:

- Trouble transitioning the ball from the stationary to the pivoting conveyor
- Fitting the 18" by 18" by 18" size limit
- Only one motor for both intakes and no space for gearing
- Potential for balls to fall out of the pivoting conveyor (no room for guard rails)



Pros of Pivoting Conveyor 2.0:

- 3-4 ball capacity
- Eliminates many of the issues in Option #2 (see pgs. 20-21)
- Unique design
- Potential to be fairly efficient
- Ability to hoard

Cons of Pivoting Conveyor 2.0:

- Modifications would be needed to fit the size limit, get the ball into the hoarding bin, etc.
- Lots of difficult problems to solve
- Spacial issues in places

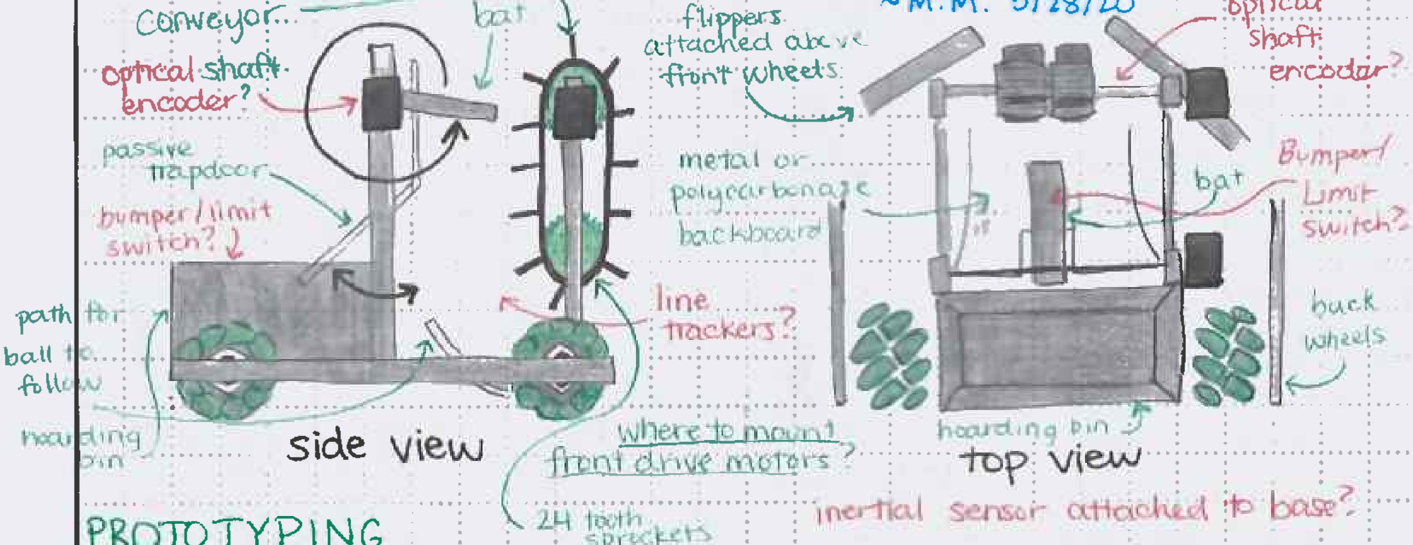
Megan M 5/27/20

5/29/20

LIFT & HANDLING DESIGN OPTIONS CONTINUED

Lift and Handling Design Option #4 : Enhanced "Snailbot"

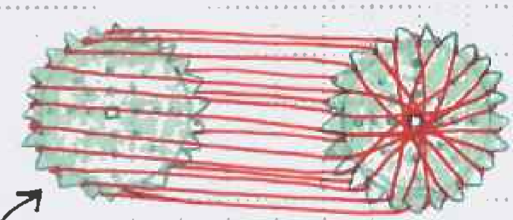
This concept started with the popular design of a snailbot (see pages 12-13). * Then, I took 5278C and 5090X 's idea of sorting the balls and modified it ~~to~~ be a hoarding bin with some sort of trapdoor on a snailbot.
~M.M. 5/28/20



PROTOTYPING

Motor breakdown:
- 4 motor drive (200 rpm)
- 2 motor intake (100 rpm + geared)
- 1 motor conveyor (200 rpm?)
- 1 motor bat

The bat pushes the ball up into the goal. No need for a hood like other snailbots, eliminating the challenge of the vertical limit.
~M.M. 5/28/20



Rubberbands work better than tread with flaps.
- Takes up less space
- Doesn't strain the motor
- Constant pressure on the ball all around
- Better grip on the ball

project   designed by: witnessed by:

*see pages 12-13 and articles on the VEX Forum

date: 5/28-29, 6/1-3/20

Megan M 5/29/20

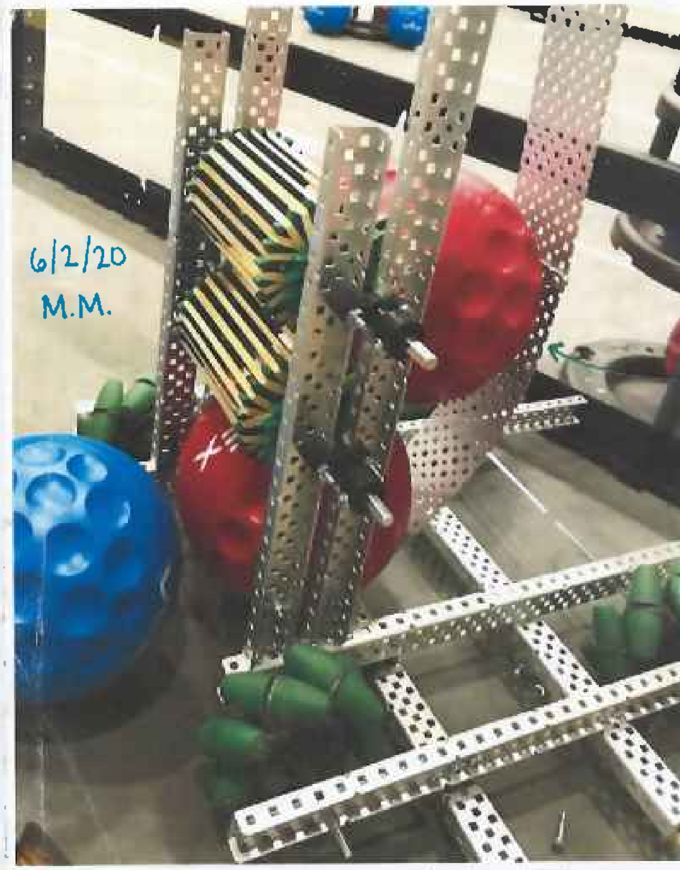
6/1/20

LIFT & HANDLING DESIGN OPTIONS CONTINUED

Enhanced "Snailbot" Continued

Problems identified through prototyping:

- The trapdoor has trouble both being stable for scoring and being able to open for hoarding.
- No great spot to mount the brain and battery
- The timing / coordination of the conveyor and bat is tricky.
- Low (ish) ball capacity
- The hoarding bin may have to expand slightly.



6/2/20 M.M.

possibly use polycarbonate here?

Rubber band conveyors work much better than tread with flaps

Pros of Enhanced "Snailbot":

- Potential to be highly efficient
- Resources available for help if needed, but still has unique elements
- Ability to hoard
- Two motors available for the intakes
- Minimal issues fitting within the size limit

Cons of Enhanced "Snailbot":

- Several difficult problems to solve
- Much easier to compare to other robots if slower (lots of snailbots)
- 3 ball capacity max (excluding hoarded balls)

Megan M 6/3/20

6/4/20

SELECTED LIFT & HANDLING DESIGN

The criteria for this decision matrix came from the design constraints and requirements (see page 6) and the design challenges in Change Up (see page 4).
~M.M. 6/4/20

Lift and Handling Design Options

Criteria (least to most important)	Scale	"Big Dipper" See pages 18-19	Pivoting Conveyor with Flywheel see pages 20-21	Pivoting Conveyor 2.0 see pages 22-23	Enhanced "Snailbot" see page 24-25
Uniqueness	0 to 3	3	2	3	1
Simplicity to build	0 to 3	2	0	1	1
Good for sensors (use and mounting)	0 to 5	3	3	3	5
Ability to fit within the size limit	0 to 5	4	2	2	4
High ball capacity	0 to 5	2	2	3	3
Few potential problems	0 to 7	2	3	3	4
Ability to hoard	0 to 7	0	7	7	7
Efficient scoring	0 to 7	0	5	6	6
	Total Score:	16	24	28	31

M.M. 6/4/20

- Based on this decision matrix and the preliminary prototyping done on this design, the Enhanced "Snailbot" is the best solution.
- Now I can move to prototyping all three selected subsystems (Drivetrain page 11, intakes page 17, and lift/handling) to work out any issues before building the final design.

project designed by: witnessed by:

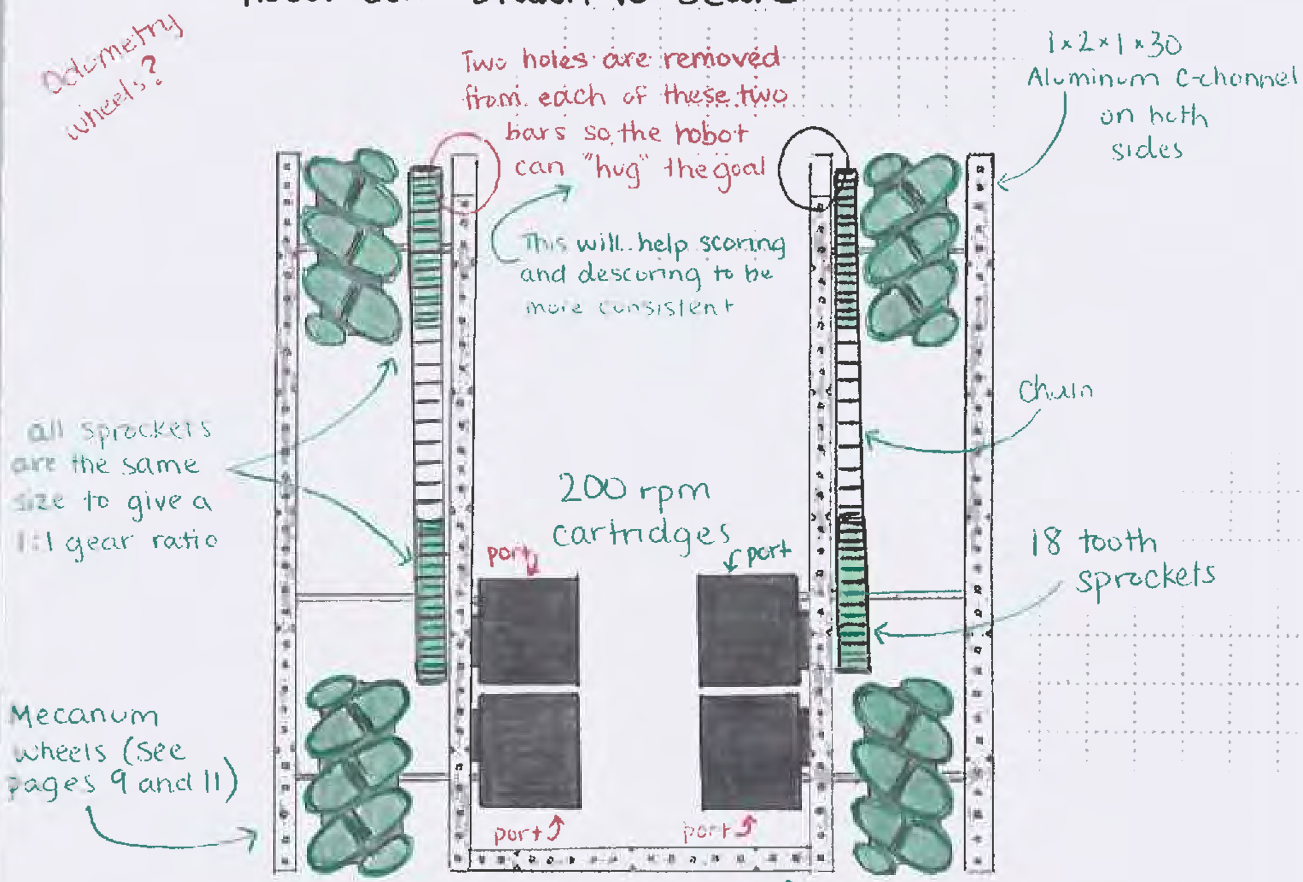
date: 6/4/20

Megan M 6/4/20

6/8/20

PROTOTYPING: BASE LAYOUT

Robot Base Drawn to Scale

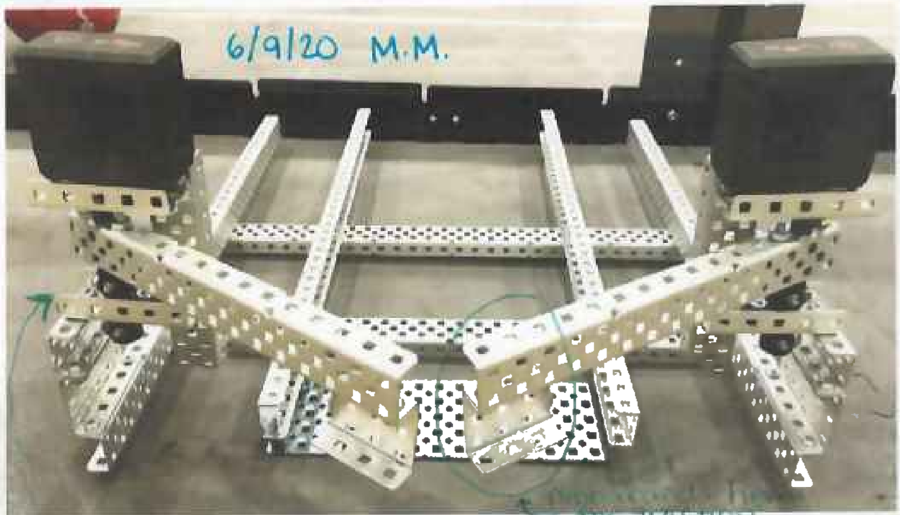


Top view
1 square = 1 inch

- This base is approx. 15" by 15".
- The front drive motors are unable to be attached directly to the front wheels due to the design of the lift and handling system.
- Additional support bars can be added into the structure of the overall design.
- This orientation of the drive motors allows each port to be easily accessible for wiring.

 6/9/20

PROTOTYPING: FLIPPERS



Perfect amount of clearance over the front wheels

First prototype of flippers with 2 motors

- 100 rpm motors still isn't enough torque
- This method of mounting still needs to be more stable.
- The motors need to be connected further back so as to not hit the goal.
- Definitely needs a high strength shaft

These flippers still show promise! Just a few bugs to be worked out. ~M.M. 6/10/20

6/11/20 M.M.

Flipper with High strength shaft

- Much more stable!
- Geared for torque, 1:3

$100\text{rpm} \times \frac{1}{3} = 33.\overline{3}$

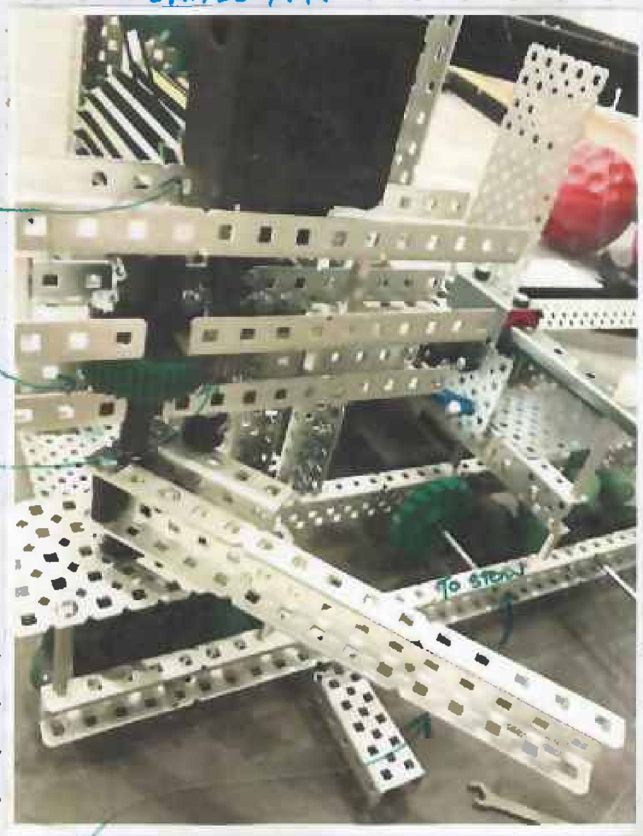
Max speed: 33 rpm

This should be more than enough speed

100 rpm motor

36 tooth gear

12 tooth pinion

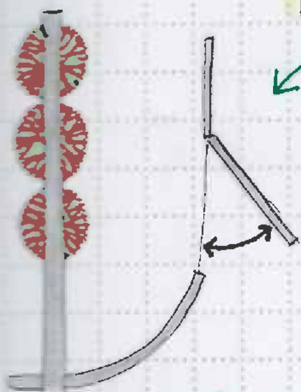


- The flipper has plenty of torque now; it has no problem "popping" a ball out of a goal.
- The only downside to this version is the large amount of space required.

Megan M[redacted] 6/11/20

6/12/20

PROTOTYPING: TRAPDOOR/HOARDING ~

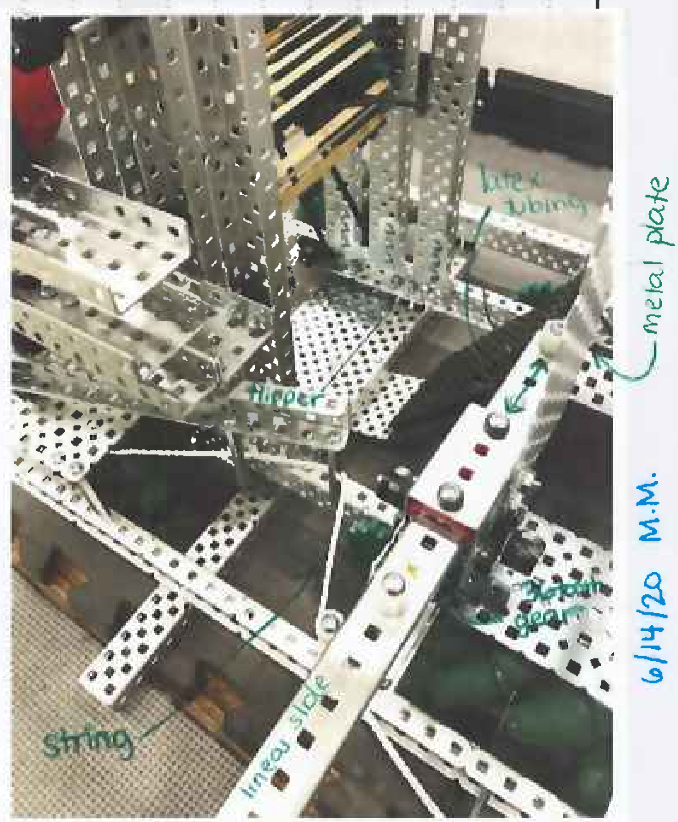


Popular Passive Trapdoor Design

(see pages 12-13)

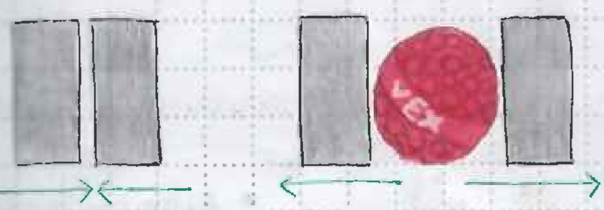
- I had trouble getting this design to both act as a sturdy backing for the ball and open to allow a ball to fall into the hoarding bin.
- Getting something to trigger the opening of the trapdoor is tricky - I only have the bat and the flippers available for this.

Linearly sliding doors design



6/14/20 M.M.

- This design uses a 36 tooth gear with gear racks in combination with a linear slide and an outer slide truck. It creates sliding doors.



Compression holds the doors together

A sort of pulley system activated by the flippers pulls the doors open.

Problems:

- Stability and being able to stay tight (string)
- The plates can be bent back - not good for the balls' backboard
- The string must be pushed very far for the doors to be opened wide enough for a ball to pass through.

When the flippers push the nylon string, the doors are opened. Once the string is no longer being pulled, compression from latex tubing causes the doors to snap closed.

project

designed by:

witnessed by:

date: 6/12-19/20

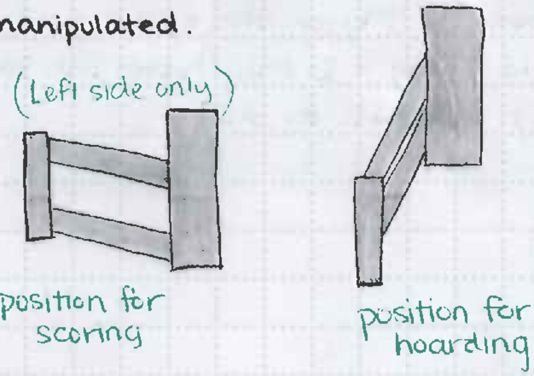
Megan M 6/13/20

6/18/20

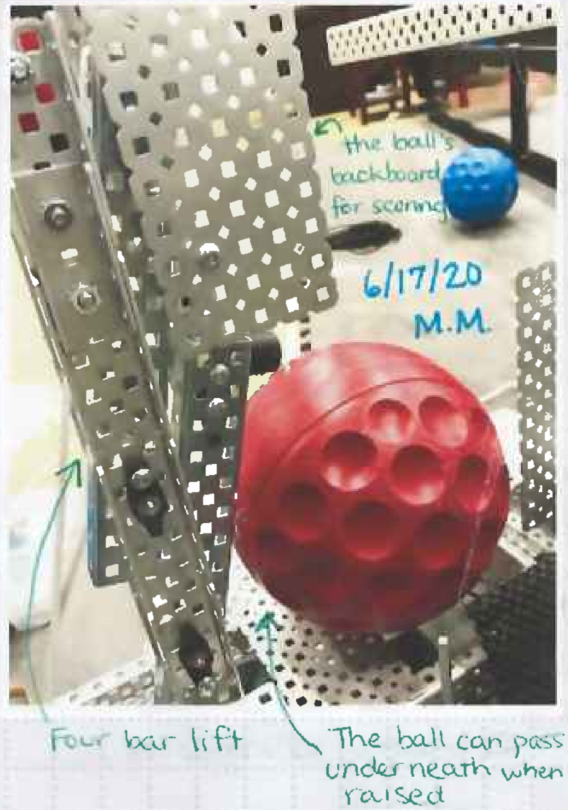
PROTOTYPING : TRAPDOOR / HOARDING CONTINUED

Four Bar Lifting Doors Design

- A four bar lift works on the principle that opposite sides of a parallelogram remain parallel even the parallelogram is manipulated.



- This design is fairly stable and opens up enough space for a ball to pass when even only one side is raised.

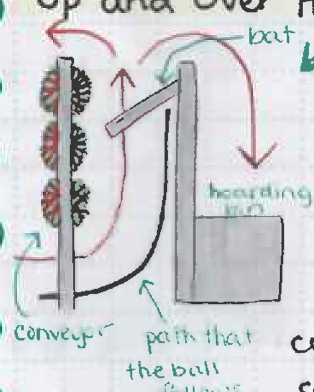


Problems:

- Very difficult to be raised by a motion of the flippers or the bat
- Potential to interfere with the bat
- Can still be bent back (although not as easily as the sliding doors design - see previous page)

This hoarding design is the best one yet. I really like having the control of being able to decide which side the ball goes in the hoarding bin. (by raising one or the other).
~ M.M. 6/17/19

Up and Over Hoarding Concept



- Although it needs to be tested, I had the idea to bring all balls to the top of the conveyor and then sort accordingly.
- This avoids many of the problems that came with having an actual moving door.
- The bat might even be able to help send the ball in the right direction.

project

designed by:

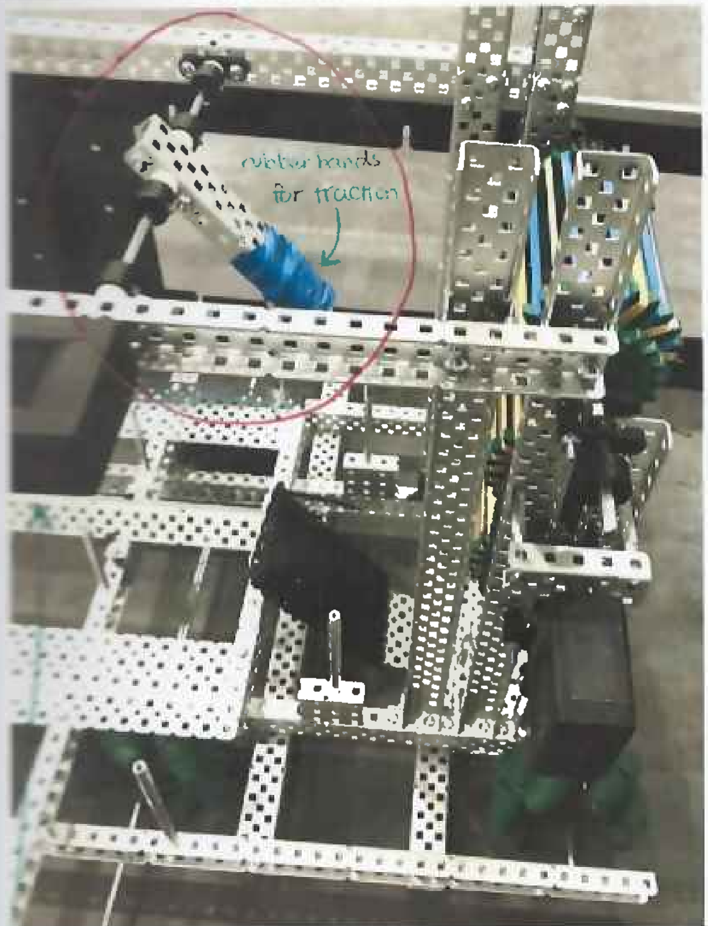
witnessed by:

date: 6/12-19/20

Megan M 6/19/20

6/22/20

PROTOTYPING: BAT / SCORING



-100 rpm motor

6/22/20 M.M.

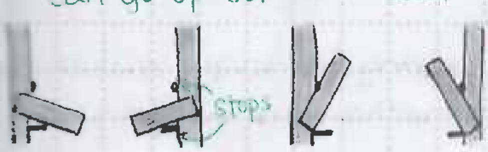
First prototype of the bat

- The bat can score balls in the outer and middle goals but not as efficiently as it needs to be.
- It needs to push the ball a lot harder - add either a lot of speed or a lot of torque.
- The bat really needs to be on a high strength shaft - this low strength axel is bending after minimal use.
- This is the perfect length (if any shorter, it wouldn't squarely hit the ball; if any longer, it would get stuck on the balls below the top one in the conveyor)

Overall, it worked quite well for a first try! I hope I can get it up to the efficiency level required to be competitive. ~M.M. 6/22/20

Launch Pad Idea

can go up but not down



resting position

position as a ball goes through

Once a ball is through and resting on the launch pad, reverse the conveyor slightly and then bat the ball into a goal

- This design is a simple one-way flap but it could help counter one of the challenges of using a bat - a consistent spot for balls to be hit from.
- The tricky part will be positioning it so that the ball naturally rests at the right height to be hit by the ball.

Megan M 6/23/20

6/24/20

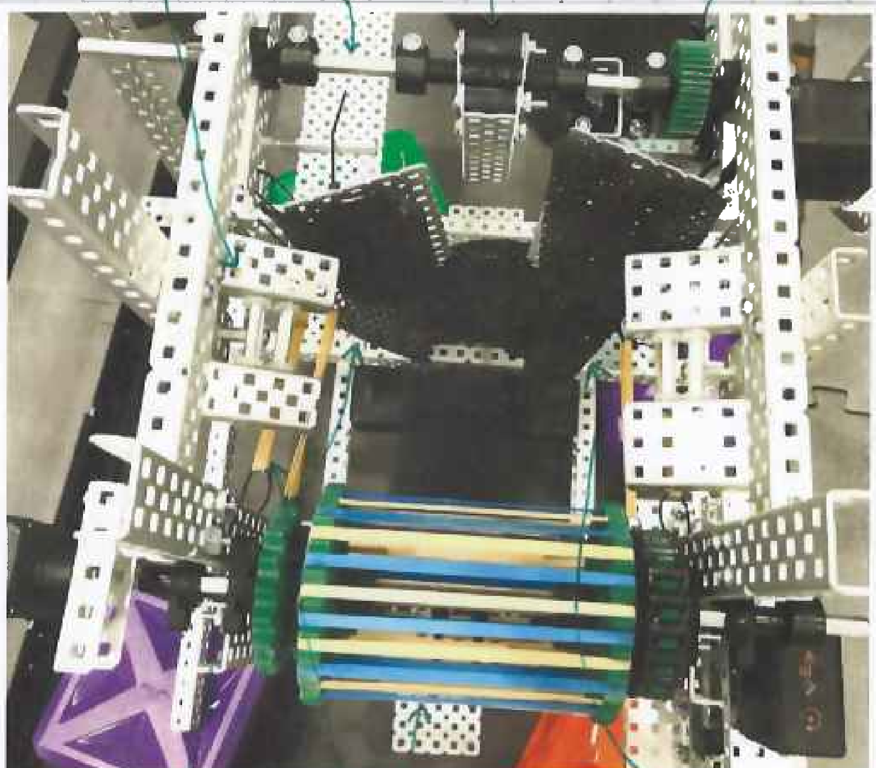
PROTOTYPING: BAT / SCORING CONTINUED

Improved Bat Design with Launch Pad and Conveyor backing

- The bat is much stronger with a high strength axel.
- The launch pad works well, although it needs to be tuned for more efficiency and possibly modified - it may offer the solution to hoarding balls.

if something could block the space above the conveyor (and be able to retract), the balls can go over bat and fall into the hoarding bin (see bottom of page 30).

see page 31
Launch pad
high strength axel
Bat
36 tooth gear above 12 tooth pinion



Conveyor backing
rubberbands needed for compression
conveyor
6/25/20 M.M.
anti-slip mat for traction

- The conveyor backing needs to be stabilized at the top. Other than that, it works well with the anti-slip mat.
- The 100 rpm motor on the bat with the 1:3 gearing for torque makes the bat way too slow. Perhaps a much faster mechanism would work better to score efficiently?

Megan M 6/25/20

6/26/20

DESIGN PLAN : THE PATH FORWARD

This is how I plan to implement the solutions that I found through research and prototyping. I will use this knowledge to construct a robot efficient at scoring, descoring, and hoarding, and with stability and unique elements.
~ M.M. 6/26/20

Drivetrain : Mecanum Drive

- 4 motors, 200 rpm
- back wheels are directly connected to motors
- front wheels are attached indirectly to motors through sprockets and chain (1:1 ratio)
- The axels go through the fourth hole in on the center row of the 1x2x1x30 c-channels.

Intakes : Flippers

- 2 motors, 100rpm, geared 1:3 for torque
- High strength axels
- Attached above front wheels
- 15 holes long with traction on the end

Conveyor

- 3 ball capacity
- Rubberbands stretched across 24 tooth sprockets (3 sets)
- 1 motor, 600 rpm
- High strength axels

Base

- approximately 15 inches by 15 inches
- 30 holes by 30 holes
- 16 holes long support bar across the back in the middle (between the back wheels)
- Additional support from the structures of scoring mechanisms

Bat with Launch Pad

- 1 motor, 200 rpm
- High strength axel
- Bat - 10 holes long (axel-second hole in) with traction on the end
- Launch Pad - each of the four pieces are 5 holes long

Hoarding

- 2 ball hoarding capacity
- No trapdoor
- expandable bin if needed
- Balls go up and over the bat with a retractable blocking piece over the conveyor (activated by flippers, snap back from compression)

While there may still be changes to be made, this is a good starting point. All of the prototyping I did gave me a good idea of what works and what doesn't. It will save me a lot of time and materials as I begin building.
~ M.M. 6/26/20

project designed by: witnessed by:

date: 6/26/20

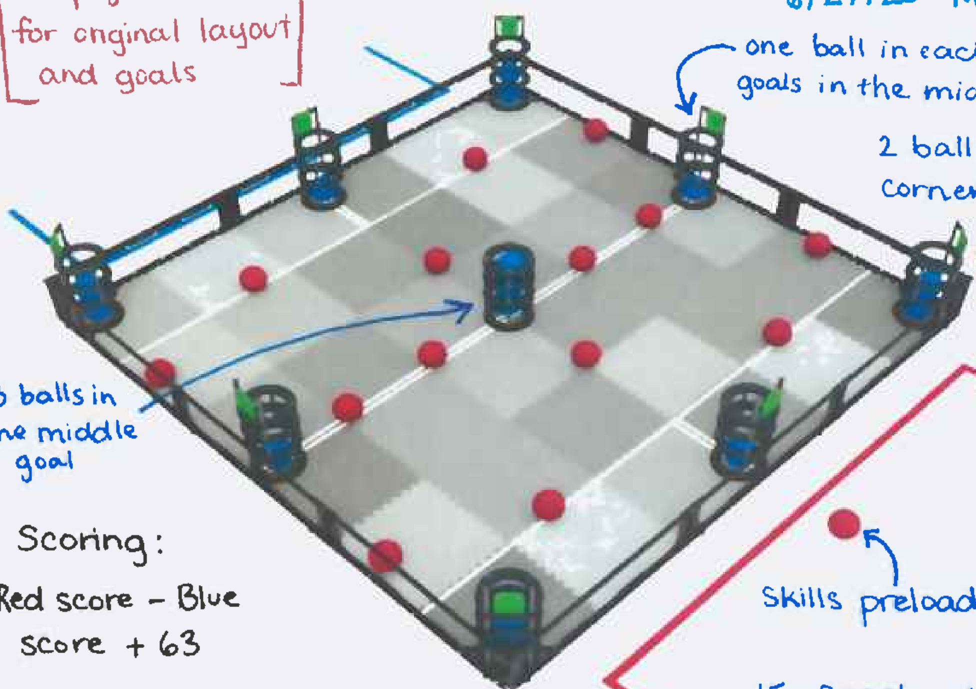
Megan M 6/26/20

6/27/20

UPDATED SKILLS FIELD LAYOUT AND GOALS ~

[see pages 3 and 7 for original layout and goals]

6/27/20 M.M.



one ball in each of the goals in the middle of the wall.

2 balls in each corner goal

3 balls in the middle goal

Skills preload

15 of each color of Ball used

Scoring:

Red score – Blue score + 63

The highest possible score is 126 points (252 combined max)

UPDATED GOALS FOR SKILLS:

- ☐ Achieve 106 points in driver skills by the TN State Championship (work towards the maximum score if qualified for Worlds)
- ☐ Achieve 106 points in programming skills by the end of the season

* This would equal a total score of 212 points.

→ Scoring 106 points requires scoring one red ball in each goal and descoring one blue ball from the center goal.

(9 scored red × 1 pt. each) + (8 connected rows × 6 pt. each) = 57 pts. RED SCORE
(14 scored blue × 1 pt. each) + (0 connected rows × 6 pt. each) = 14 pts. BLUE SCORE
57 - 14 + 63 = $\boxed{106 \text{ points}}$

project (see formula above) designed by: witnessed by:

date: 6/27/20

Megan M 6/27/20

6/27/20

TIMELINE : UNTIL FIRST COMPETITION

6/27/20
M.M.

BUILDING

Structure/Mobility
Base, Drivetrain
6/28 - 6/30

June

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Gameplay Mechanisms
Conveyor, Flippers, Bat, Hoarding Bin
7/1 - 7/20

July

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Improvements
Tuning, Sensors
7/21 - 7/31

July

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

This is my ideal plan of getting ready for competitions. Unexpected challenges may come - that's why I made sure to have some extra time just in case.
~M.M.
6/27/20

6/27/20
M.M.

PROGRAMMING

Driver Control Macros
8/1 - 8/15

August

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Autonomous Programming
(15 sec and 60 sec)
8/16 - 10/3

August

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

September

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

October

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Competition Prep
Auton Tuning
Driving Practice
10/4 - First Competition (?)

October

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

?

date: 6/27/20

Megan M 6/27/20

6/28/20

BUILDING THE BASE STRUCTURE

GOAL: Build a stable foundation for the robot that doesn't interfere with any subsystems that will be added in the future.

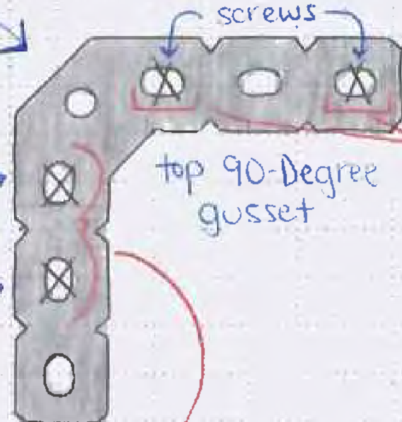
GOAL COMPLETED 6/29/20

Back support bar

90-Degree gussets for stability

This spacing is so as to not interfere with the back wheels

Built based on successful prototype (see pg. 27)

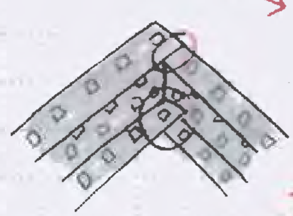


top 90-Degree gusset
steel washer
teflon washer
top lip of c-channel
$\frac{3}{8}$ inch spacer
$\frac{1}{2}$ inch spacer
bottom lip of c-channel
bottom 90-Degree gusset
nylock

The order of what goes through the $1\frac{1}{2}$ inch screw

* The washers change position in the two lists!

top 90-Degree gusset
top lip of c-channel
$\frac{3}{8}$ inch spacer
$\frac{1}{2}$ inch spacer
bottom lip of c-channel
teflon washer
steel washer
bottom 90-Degree gusset
nylock



The difference in order compensates for the overlap of the channels

- This method of attachment provides a good deal of stability over a single screw.

project

designed by:

witnessed by:

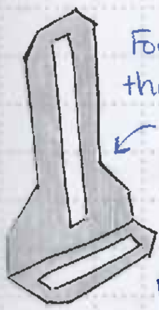
date: 6/28-29/20

Megan M 6/29/20

6/29/20

BUILDING THE BASE STRUCTURE CONTINUED

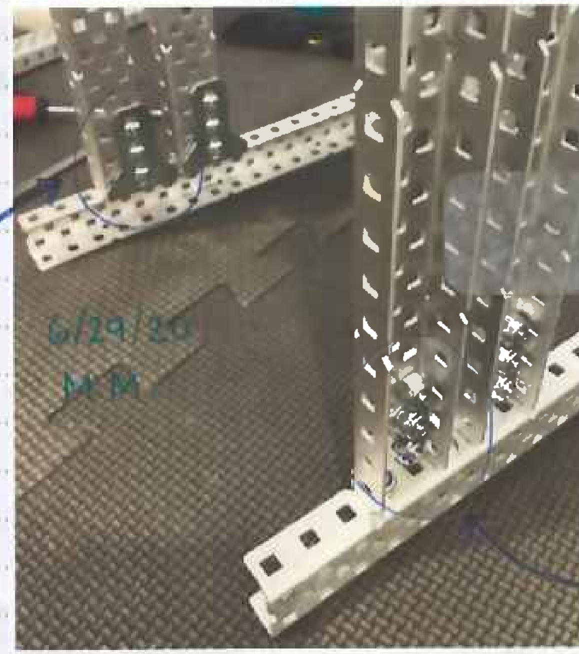
- I attached the four vertical posts that will hold the conveyor and provide support for the rest of the mechanisms on the robot.



Four of these were used

structural L-shaped gusset

Each gusset is attached with 5 - ½ inch screws and nylocks



back of attached gusset

- Using these gussets allows me to save the extra space that would be required to mount the post on the side of the base for other subsystems.

The finished base structure

Materials used in the Base Structure:

- 6 - 1×2×1×28 Aluminum C-channels
- 1 - 1×2×1×16 Aluminum C-channel
- 4 - 90-Degree gussets
- 4 - Structural L-shaped gussets
- 20 - 0.375 in. screws
- 8 - 1½ inch screws
- 8 - ⅜ inch spacers
- 8 - ½ inch spacers
- 8 - steel washers
- 8 - teflon washers
- 28 - nylocks

Megan M 6/29/20

6/29/20

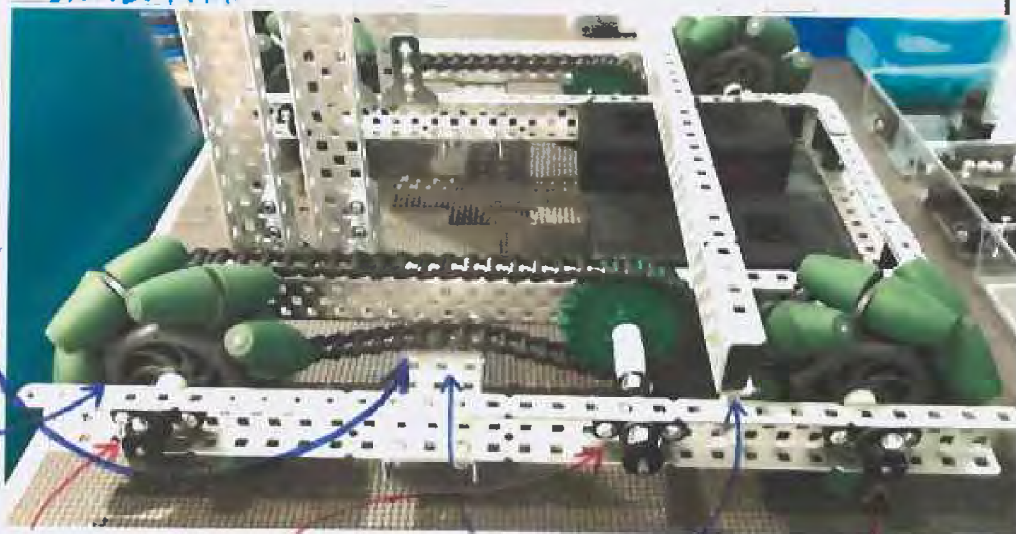
BUILDING THE DRIVETRAIN

GOAL: Build a stable mecanum drive so that the sides are symmetrical and the base of the robot is perfectly square.

GOAL COMPLETED 6/30/20

6/30/20 M.M.

- Today I built the drivetrain and added three support bars.



chain and sprockets to power the front wheels

mecanum wheels

Shaft goes through hole 4 in the center row

added support bars

Axel goes through hole 11 from the back end in the center row

Axel goes through hole 4 from the end in the center hole

Shaft through front wheel: outside → inside	Axel through sprocket: outside → inside	Axel through back wheel: outside → inside
clamping shaft collar	clamping shaft collar	clamping shaft collar
$\frac{1}{8}$ inch spacer	$\frac{1}{8}$ inch spacer	$\frac{1}{8}$ inch spacer
flat bearing	flat bearing	flat bearing
C-channel	C-channel	C-channel
teflon washer	$\frac{1}{4}$ inch spacer	teflon washer
$\frac{1}{2}$ inch spacer	clamping shaft collar	$\frac{1}{2}$ inch spacer
mecanum wheel	$\frac{1}{2}$ inch spacer	mecanum wheel
$\frac{1}{2}$ inch spacer	$\frac{3}{8}$ inch spacer	$\frac{1}{2}$ inch spacer
18 tooth sprocket with chain	$\frac{1}{2}$ inch spacer	clamping shaft collar
$\frac{1}{8}$ inch spacer	teflon washer	$\frac{1}{4}$ inch spacer
C-channel	18 tooth sprocket with chain	teflon washer
flat bearing	$\frac{1}{8}$ inch spacer	C-channel
$\frac{1}{8}$ inch spacer	C-channel	V5 Smart motor
clamping shaft collar	V5 Smart Motor	

project: designed by: witnessed by:

date: 6/29-30/20

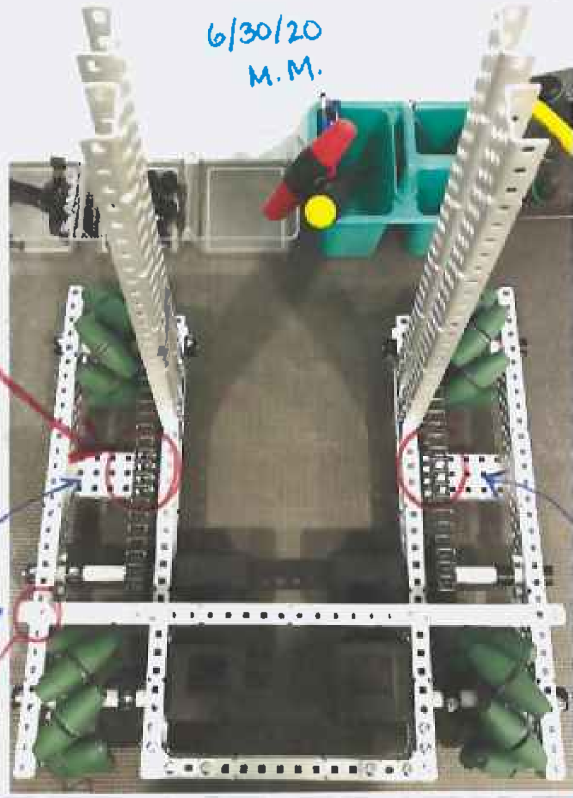
Megan M 6/30/20

6/29/20

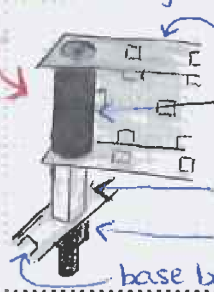
BUILDING THE DRIVETRAIN CONTINUED

- I plan to attach a small piece of polycarbonate to each of the two support bars beneath the chain.
  → This should allow the chain to slide over it smoothly.

6/30/20
M.M.



Newly added support bars



This method of attachment was necessary to clear the wheel and sprocket.

- The drivetrain and base are very sturdy now with the extra support bars.

Materials used in the Drivetrain:

- 20 - 0.375 in. screws
- 16 - $\frac{1}{2}$ in. screws
- 16 - nylocks
- 12 - thin nylocks
- 12 - clamping shaft collars
- 8 - flat bearings
- 8 - teflon washers
- 12 - $\frac{1}{8}$ inch spacers
- 4 - $\frac{1}{4}$ inch spacers
- 2 - $\frac{3}{8}$ inch spacers
- 12 - $\frac{1}{2}$ inch spacers
- 6 - cut shafts
- 4 - mecanum wheels
- 4 - 18 tooth sprockets
- 4 - V5 smart motors (200 rpm)
- chain

Materials used in the Supports:

- 1 - 1x2x1x30 Aluminum C-channels
- 2 - 1x3x1x8 Aluminum C-channels
- 4 - 2 inch screws
- 8 - 0.375 in. screws
- 4 - $\frac{3}{8}$ inch spacers
- 4 - $\frac{1}{2}$ inch spacers
- 4 - $\frac{1}{2}$ inch standoffs
- 12 - nylocks

project designed by: witnessed by:

date: 6/29-30/20

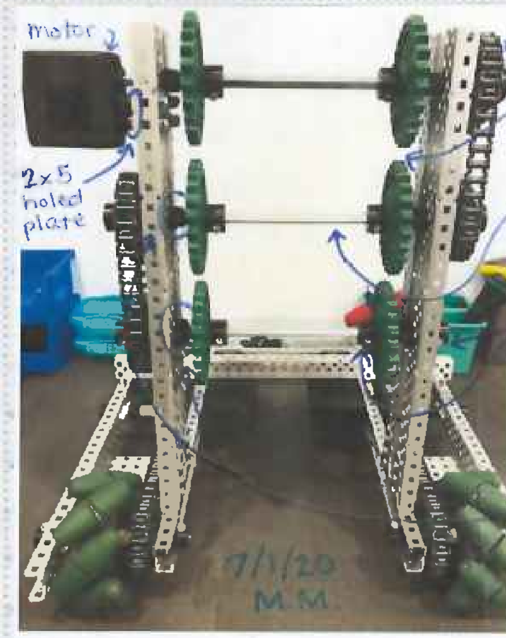
Megan M 6/30/20

7/1/20

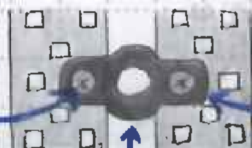
BUILDING THE CONVEYOR

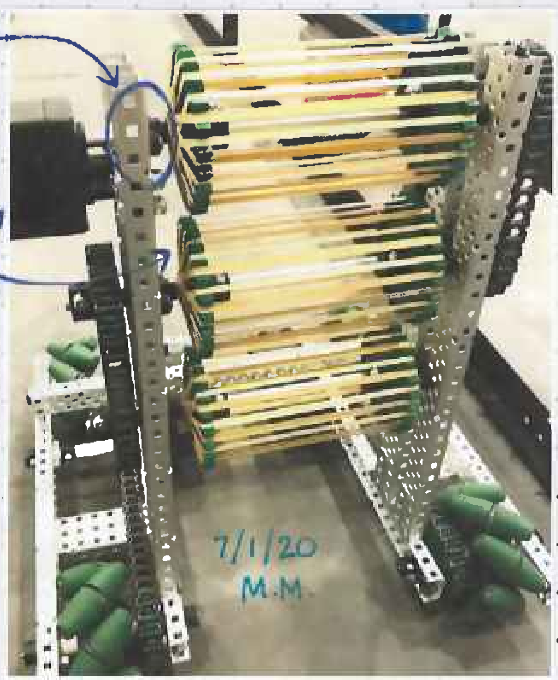
GOAL: Build a conveyor system with stability and minimal friction to allow for higher speeds without the motor overheating.

GOAL COMPLETED 7/2/20



- I built the conveyor onto the four vertical bars mounted on 6/29/20 (see page 37).
- The one hole gap between each set of bars allows me to use high strength shafts.





→ Materials used in the Conveyor: ←

7/9/20

- ~~4~~ 2 - 18 tooth sprockets
- 6 - 24 tooth sprockets
- 1 - V5 Smart motor ~~(600rpm)~~ 200rpm 7/2/20
- 36 - rubberbands
- 3 - cut high strength shafts
- 1 - 2x5 holed aluminum plate
- 6 - high strength flat bearings
- 11 - high strength clamping shaft collars
- 27 - nylocks
- 2 - 0.375 in. screws
- 23 - 0.625 inch screws
- ~~4~~ 2 - 0.875 inch screws 7/13/20
- 4 - $\frac{1}{2}$ inch spacers
- 6 - $\frac{1}{4}$ inch high strength spacers
- 5 - $\frac{1}{8}$ inch high strength spacers
- 5 - $\frac{1}{2}$ inch high strength spacers
- chain
- 2 - 1 inch screws 7/13/20

changes on page 48 7/9/20

7/13/20: (replaced 2 0.875 in for 1 in screws)

- 9 - 11 in. zipties (added 7/31/20)

project

designed by:

witnessed by:

date: 7/1-2/20

Megan N 7/1/20

7/2/20

TESTING THE CONVEYOR

GOAL: Verify if 600 rpm motor will overheat on the conveyor and make adjustments as needed.

GOAL COMPLETED 7/2/20

- I continually ran the 600 rpm motor until it stalled and recorded the temperature at various points along the way.
- I performed the test on three different ratios for comparison.

Testing at 300 rpm requires gearing

$100\text{rpm} \times \frac{3}{1} = \boxed{300\text{rpm}}$

7/2/20
M.M.

36 tooth gear
12 tooth pinion
5x5 holed steel plate

The code for getting the temperature of a motor:

7/2/20 M.M.

```
Controller1.Screen.clearLine();
Controller1.Screen.print("T: %.0f", ConveyorMotor.temperature(percent));
```

7/2/20
M.M.

Motor Speed	Time	Temperature (in percent)	Comments
600 rpm (600 rpm gear cartridge)	0	10	→ starting temperature
	1 min.	20	
	2 min.	40	
	3 min. 15 sec.	70	→ motor stalls due to overheating
300 rpm (100 rpm gear cartridge with 3:1 ratio)	0	10	→ starting temperature
	1 min.	20	lasted a little over double the time
	2 min.	30	
	5 min.	50	
	6 min.	70	
	7 min.	80	→ motor stalls due to overheating
200 rpm (200 rpm gear cartridge)	0	10	→ starting temperature
	1 min.	20	rather slow but the motor never stalled
	2 min.	20	
	5 min.	40	
	7 min.	50	
	10 min.	70	
	12 min. 45 sec.	80	→ testing ended

- Based on this test, 600 rpm overheats much too quickly.
- I decided to change the 600 rpm to a 200 rpm so it will work consistently.
- This will be much slower than I had wanted - improve this in the future?

Megan M 7/2/20

7/3/20

BUILDING : ADDING POLYCARBONATE

GOAL: Cut and attach the polycarbonate pieces for the conveyor backing and to go underneath the chain on the drivetrain. **GOAL COMPLETED 7/5/20**

It will be clear once the plastic is removed!

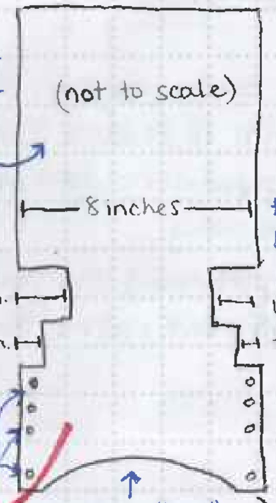
4 screws on each side through the inner base bars

Space to attach a support bar

plenty of extra to cut later after the bat is built.

extra space cut out for support bar

screws go through the inner base bars



to clear base bars

cut off to "hug" the goal

polycarbonate

Current length is $23\frac{3}{4}$ in (will be cut to size later)

Material used in conveyor backing:

- 1 – 8 in. by $23\frac{3}{4}$ in. polycarbonate
- 8 – 0.375 in screws
- 8 nylocks

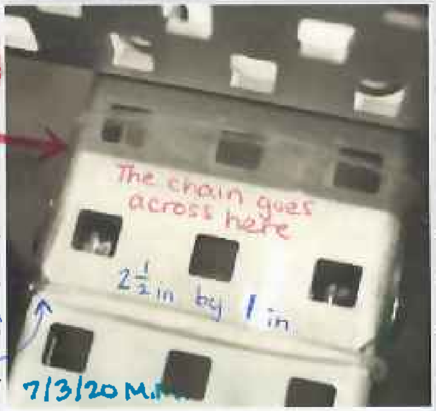
- The polycarbonate conveyor backing appears to work very well with a single layer of anti-slip mat.
- The small pieces of polycarbonate underneath the chain on the drivetrain make a big difference – the front wheels spin much smoother now.

(see page 39)

Material used on the base/drivetrain:

- 2 – $2\frac{1}{2}$ in. by 1 in. polycarbonate
- 4 – 0.375in screws
- 4 – nylocks

I used a hair dryer to heat the polycarbonate so I could bend it around the C-channel.



7/6/20

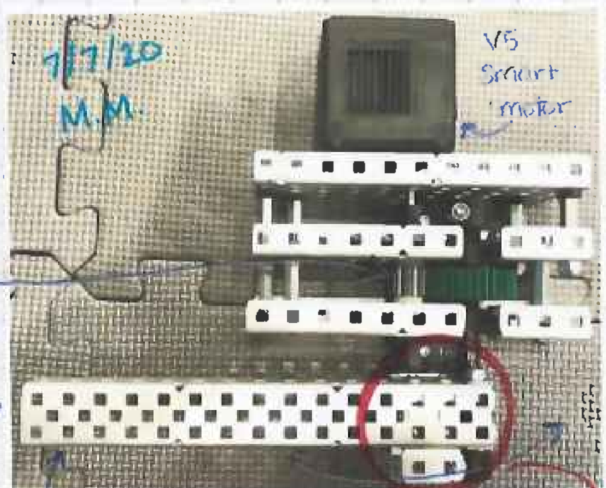
BUILDING THE FLIPPERS

GOAL: Construct the flippers to be very sturdy and be able to fit within the size limit above the front wheels. **GOAL COMPLETED 7/8/20**

Built based on prototyping (see page 28) but condensed slightly

gearing 1:3 (torque)

flipper



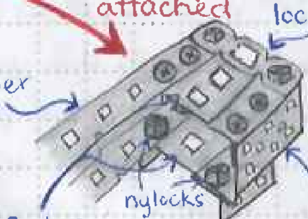
V5 Smart motor

stowed position

front of robot

- The structure of the flipper is quite large due to the usage and size of the high strength shaft (also flat bearings and clamping shaft collars).
- This design is very stable and fits nicely above the front wheels at a good height for descoring (with add-on, see page 64) 8/21/20

how the flipper is attached



Axel (regular strength):	High Strength Shaft:
V5 Smart Motor	H.S. clamping shaft collar
C-channel	1/4 inch H.S. spacer
clamping shaft collar	H.S. flat bearing
1/8 inch spacer	1/8 inch H.S. spacer
flat bearing	36 tooth H.S. gear
C-channel	1/4 inch H.S. gear
1/8 inch spacer	1/2 inch H.S. spacer
12 tooth metal pinion	H.S. lock bar
1/4 inch spacer	1/2 inch H.S. spacer x2
teflon washer x2	H.S. lock bar
flat bearing	1/8 inch H.S. spacer
c-channel	H.S. clamping shaft collar
teflon washer	
clamping shaft collar	

- Flat bearings and lock bars replace C-channels for a high strength shaft (this prevents having to drill holes through C-channels).

The order of what goes on the axels for each of the two flippers

7/7/20

BUILDING THE FLIPPERS CONTINUED

100 rpm gear cartridge

$\frac{1}{8}$ in. spacer

1 in. standoff

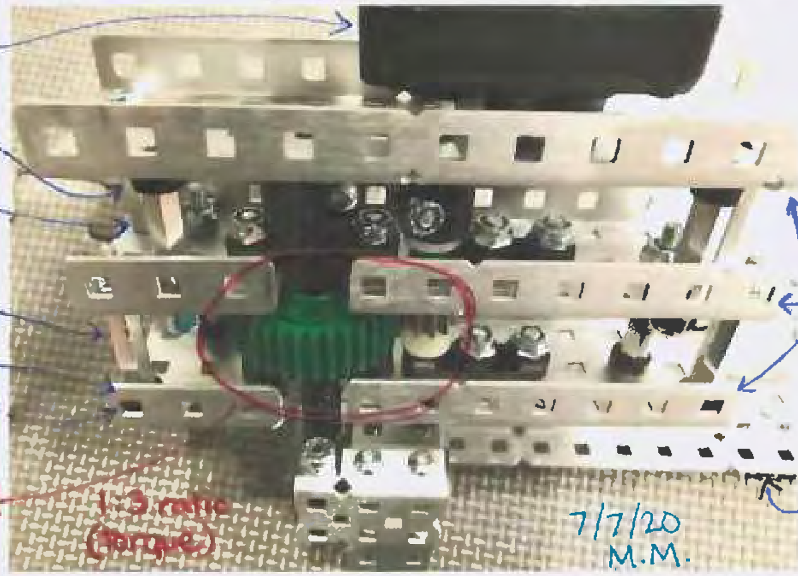
1 in. standoff

$\frac{1}{8}$ in. spacer

2 teflon washers

100 rpm x $\frac{1}{3}$ = 33.$\overline{3}$ rpm

Max speed: 33 rpm



The flippers will be mounted to the vertical bars (see page 37) from these c-channels.

The flipper

Materials used to build the Flippers:

- 2 - V5 Smart Motors (100 rpm)
- 2 - 36 tooth H.S. gears
- 2 - 12 tooth metal pinions
- 2 - metal pinion inserts
- 2 - 1x2x1x15 aluminum c-channels
- 4 - 1x3x1x7 aluminum c-channels
- 2 - 1x3x1x11 aluminum c-channels
- 4 - 1x3x1x3 aluminum c-channels
- 4 - 1x3x1x2 aluminum c-channels
- 2 - 1x2x1x3 aluminum c-channels
- 2 - cut shafts (regular strength)
- 2 - cut high strength shafts
- 4 - flat bearings
- 4 - H.S. flat bearings
- 4 - clamping shaft collars
- 4 - H.S. clamping shaft collars
- 16 - 1 inch standoffs
- 36 - nylocks
- 24 - thin nylocks
- 22 - nylon washers
- 24 - 0.375 inch screws
- 16 - $\frac{1}{2}$ inch screws
- 4 - 0.625 inch screws
- 4 - 1$\frac{1}{4}$ inch screws
- 8 - 1$\frac{1}{2}$ inch screws
- 8 - 1$\frac{3}{4}$ inch screws
- 20 - $\frac{1}{8}$ inch spacers
- 2 - $\frac{1}{4}$ inch spacers
- 4 - $\frac{3}{8}$ inch spacers
- 4 - $\frac{1}{2}$ inch spacers
- 4 - $\frac{1}{8}$ inch H.S. spacers
- 6 - $\frac{1}{4}$ inch H.S. spacers
- 6 - $\frac{1}{2}$ inch H.S. spacers

Megan M 7/7/20

3/20

BUILDING THE FLIPPERS CONTINUED

- Now that the flippers are built, I need to mount them in the right spot on the robot.
  - > Must be very stable to withstand pressure
  - > Must not contact the front wheels
  - > Must fit within the size limit
  - > Must be low enough to clear the goal
  - > Must be far enough forward to reach the back of the goal

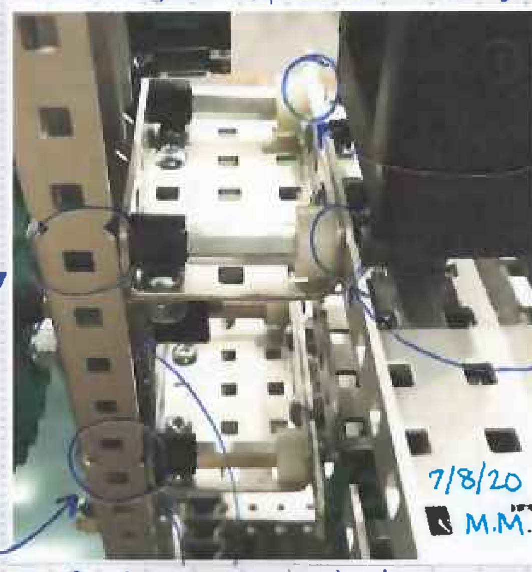
The Right Flipper Mounted

hole 15 up

hole 10 from the frontend

hole 6 from the frontend

hole 10 up



90-Degree gussets underneath for additional support

90-Degree gusset

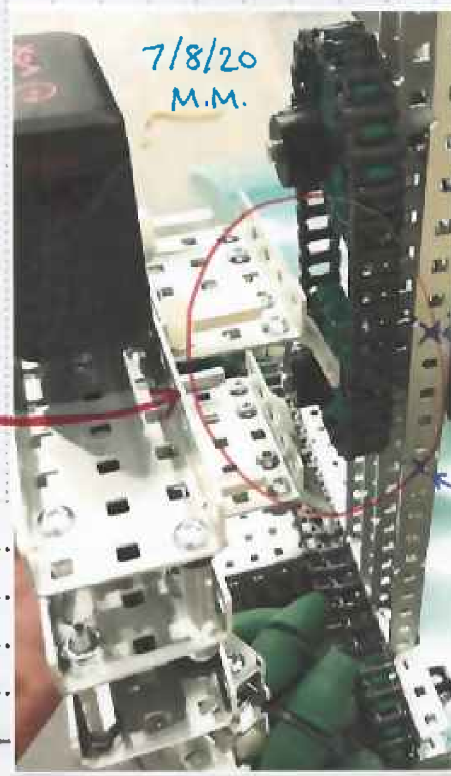
1x3x1x5 c-channels attached to the two outer channels on the flippers and the vertical bars (used for the conveyor as well)

- The c-channels by themselves aren't super stable but adding two 90-Degree gussets under each c-channel made a huge difference.

PROBLEM MOUNTING LEFT FLIPPER:

- Because of the different positioning of the sprockets / chain on each side of the conveyor, (see page 40) there isn't room to mount the left flipper the same way as the right one.

- The flippers must be mounted at the same height - and the position of the right flipper (see above) is perfect.



It would attach here

Megan M 7/8/20

7/8/20

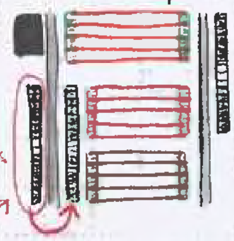
DESIGN CYCLE: FLIPPER MOUNTING PROBLEM

> **PROBLEM / CHALLENGE:** The conveyor prohibits the left flipper from being mounted in the same way as the right flipper (see previous page). page 45 for more detail!

pg. 45 for more detail!

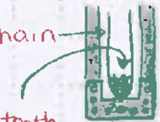
> **BRAINSTORM:** Possible solutions to the problem:

① Move the sprockets with chain to the inside of the vertical bars.



X Makes the ball's path off-center
X The ball could catch on the sprockets

② Downsize the sprockets and cut the c-channels on the mounts to fit around it.



X Makes the flipper unstable
X Lots of dremeling makes pieces unusable in the future

③ Slightly downsize the sprockets and add polycarbonate so the chain can go smoothly around the mount.

X Uses some of the limited amount of polycarbonate allowed.

> **SELECT OPTION:**

Left Flipper Mounting Possible Solutions — 7/8/20 M.M.

Criteria (least to most important)	Scale	① Move 18 tooth sprockets inward	② Chain goes *through* a dremeled section	③ Chain goes *around* with polycarbonate
Flipper Stability	0 to 5	5	2	5
Does not interfere with intaking balls	0 to 5	0	4	5
Does not prohibit conveyor function	0 to 5	3	4	4
	Total Score:	8	10	14

Having the chain go around the flipper mount is definitely the best solution. I will secure pieces of polycarbonate so the chain flows smoothly.

project — designed by: — witnessed by:

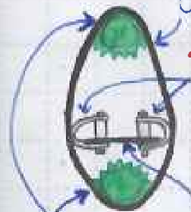
date: 7/8-9/20

Megan M 7/8/20

7/8/20

DESIGN CYCLE: FLIPPER MOUNTING CONTINUED

BUILD/IMPLEMENT SOLUTION:



chain

polycarbonate $2\frac{1}{2}$ in. by $1\frac{1}{2}$ in. pieces

12 tooth sprocket

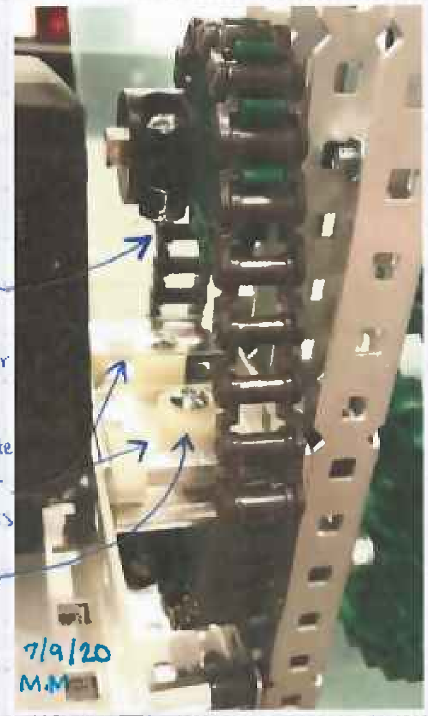
1x3x1x5 C-channel (part of left flipper mount)

- I mounted the left flipper at the same height as the right flipper using the solution of the chain going around the flipper mount.

12 tooth sprockets instead of 18 tooth like the right side of the conveyor

polycarbonate pieces for smoothness

$\frac{1}{2}$ inch spacer



Left Flipper mounted

TEST SOLUTION:

- The left flipper feels stable (equal to the right flipper). The mounting is able to withstand the pressure of the flippers opening / closing.
- The conveyor moves smoothly. The chain does not catch on anything, and all of the three rollers move at the same speed.

PROBLEM/CHALLENGE
Mounting the left flipper

BRAINSTORM
① Move sprockets inward
② Chain goes through
③ Chain goes around

SELECT OPTION
③ Chain goes around → most stability and least problematic

BUILD/IMPLEMENT SOLUTION
- Switched to 12 tooth sprockets
- Attached polycarbonate pieces

TEST SOLUTION
- Flipper is stable
- Conveyor runs smoothly

PROBLEM SOLVED 7/9/20

Design Cycle

designed by:

witnessed by:

date: 7/8-9/20

Megan N 7/9/20

7/9/20

DESIGN CYCLE: MOUNTED FLIPPERS

Materials used to mount the Flippers:

- 4 – 1×3×1×5 aluminum c-channels
- 8 – 90-Degree gussets
- 8 – $\frac{3}{4}$ inch standoffs
- 8 – $\frac{1}{4}$ inch spacers
- 8 – $\frac{1}{2}$ inch spacers
- 8 – $\frac{3}{8}$ inch spacers
- 38 – 0.375 inch screws
- 2 – $\frac{1}{2}$ inch screws
- 6 – 0.625 inch screws
- 8 – $1\frac{1}{4}$ inch screws
- 38 – nylocks

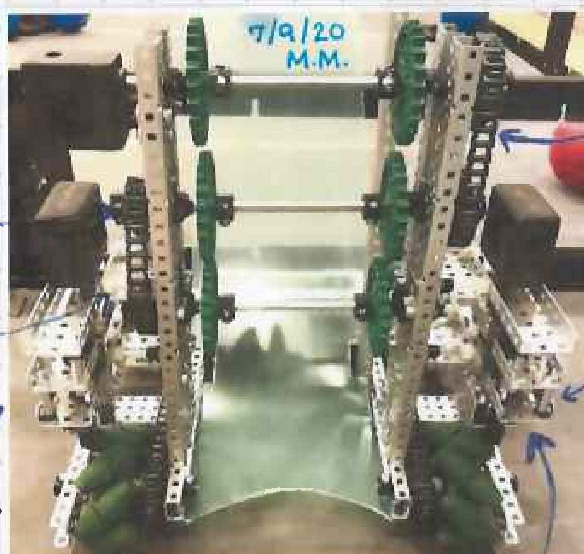
Materials that changed on the Conveyor:

(See initial list on page 40)

- 2 – $2\frac{1}{2}$ in. by $1\frac{1}{2}$ in. polycarbonate
- 2 – additional 0.875 inch screws
- 2 – additional $\frac{1}{2}$ inch spacers
- 2 – additional nylocks
- 2 – 4 inch zipties
- 2 – 12 tooth sprockets
- removed 2 – 18 tooth sprockets
- additional chain

18 tooth sprockets

12 tooth sprockets



Flippers in stowed position (folded back)

polycarbonate

Flippers Mounted

project

designed by:

witnessed by:

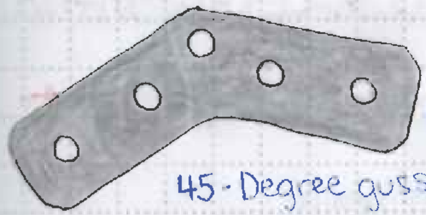
date: 7/9/20

Megan M 7/9/20

7/11/20

BUILDING: ADDING BUMPERS

GOAL: Add bumpers to the front of the robot at the right height, very securely, and without hindering the front wheels / sprockets. **GOAL COMPLETED 7/11/20**



45-Degree gusset

Bumpers

- These bumpers rest approx. 1 cm. off the ground - low enough to "bump" into the ground-level ring of each goal.

nylon washers

$\frac{3}{8}$ inch spacers

approx. 1 cm. off the ground

The edge of the 1x5x1x3 c-channel is sandwiched between two 45-Degree gussets

Materials used for the Bumpers:

- 2 - 1x5x1x3 steel c-channel
- 4 - 45-Degree gussets
- 4 - nylon washers
- 4 - $\frac{3}{8}$ inch spacers
- 4 - 0.375 inch screws
- 4 - 0.875 inch screws
- 8 - nylocks

- These bumpers have many benefits:
  - consistent position for scoring and descoring
  - protects the chain on the drivetrain
  - protects conveyor (rubberbands as well as sprockets) from getting caught on the goals

designed by: witnessed by:

date: 7/11/20

Megan M 7/11/20

7/12/20

BUILDING THE CONVEYOR BACKING SUPPORTS

GOAL: Build the supports for the polycarbonate (see page 42) to be stable and that allows ample space for mounting the brain and battery.

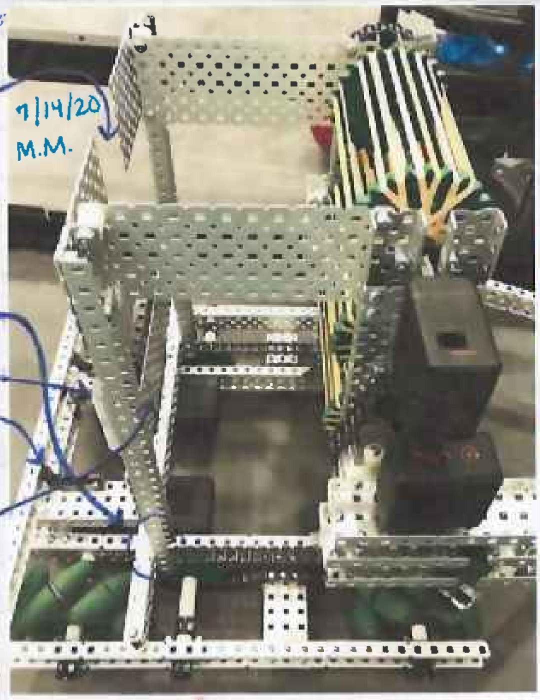
GOAL COMPLETED 7/14/20

This gap allows the bat to fit in between

1x2x1x27 C-channels attached to base support bar

Brain and battery clips

5x20 plate doubles as a support and a place to mount the brain



side view of conveyor backing support

- These supports also help to stabilize the conveyor / flippers with the base.

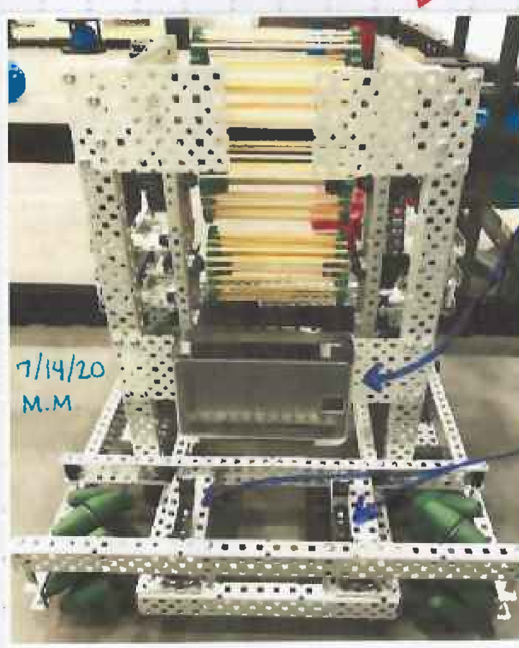
Materials used in polycarbonate supports:

- 1 - 5x20 aluminum plate
- 2 - 5x12 aluminum plates
- 2 - 5x8 aluminum plates
- 2 - 1x2x1x27 aluminum c-channels
- 4 - $\frac{1}{2}$ inch spacers
- 27 - 0.375 inch screws
- 4 - 0.875 inch screws
- 31 - nylocks

Materials to mount brain / battery:

- 1 - V5 robot brain
- 2 - V5 battery Clips
- 1 - 1x2x1x30 aluminum c-channel
- 2 - 1x3x1x9 aluminum c-channel
- 4 - 1 inch standoffs
- 4 - $\frac{3}{8}$ inch spacers
- 8 - $\frac{1}{2}$ inch spacers
- 4 - $\frac{1}{4}$ inch locking screws
- 8 - 0.375 inch screws
- 4 - 0.625 inch screws
- 4 - 0.875 inch screws
- 4 - $1\frac{1}{4}$ inch screws
- 12 nylocks

Back view of conveyor backing support



Brain mounted to the back of polycarbonate support

battery Clips on c-channels over back drive motors

additional support bar

project

designed by:

witnessed by:

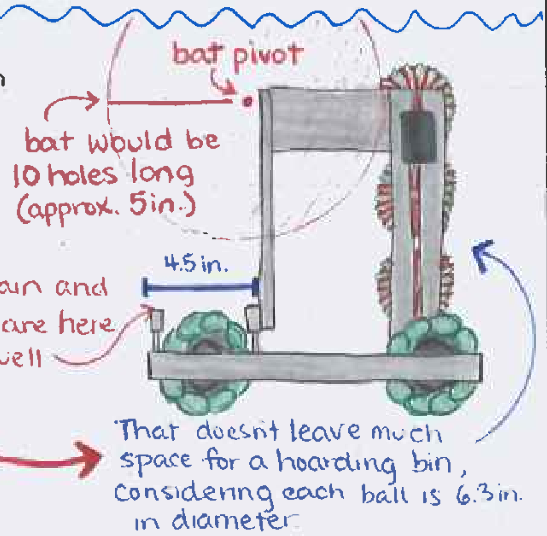
date: 7/12-14/20

Megan M 7/14/20

7/20/20

DESIGN CYCLE: BAT SPACING PROBLEM

> **PROBLEM / CHALLENGE:** There is not enough space to both mount the bat and have a hoarding bin (even an expandable one) due to the large area that must remain clear for the bat to turn 360°

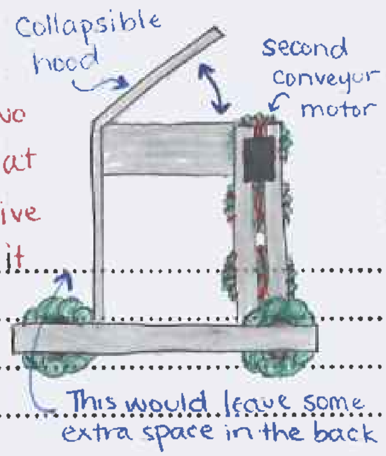


> **BRAINSTORM:**

Possible solutions to the problem:

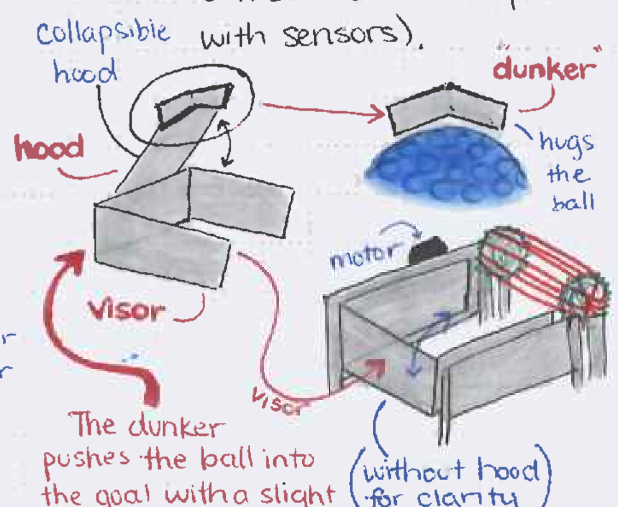
① Keep the bat the same and lose the hoarding bin. With no bin, there would be plenty of space for the bat.
- X Defeats the primary purpose for choosing the bat (pgs 24-26).
- X No hoarding

② Replace the bat with a hood like many robots have (see pgs 12-13). Use the leftover motor from the bat on the conveyor (switch to 600rpm?).
- X No hoarding
- X Rebuilding would be necessary to have a second conveyor motor.



Ideally having two conveyor motors at 600 rpm should give the ball the push it needs, however is overheating still an issue?

③ Replace the bat with a visor / hood with "dunker" combination. This would be more unique than a typical hood and allows hoarding.
- X Efficient scoring will be difficult (but not impossible with sensors).



The dunker pushes the ball into the goal with a slight lift of the visor. By raising the visor higher, balls can fall into the hoarding bin. The hood guides the ball to the dunker.

project

designed by:

witnessed by:

date: 7/20-28/20

Megan N 7/20/20

7/22/20

DESIGN CYCLE: BAT SPACING PROBLEM CONTINUED

> SELECT OPTION:

Alternative Scoring / Hoarding Methods: Possible Solutions

Criteria (least to most important)	Scale	① Bat with No Hoarding Bin	② Hood and Additional Conveyor Motor	③ Visor and Hood with Dunker
Minimal Rebuilding	0 to 3	3	1	3
Fits within the Size Limit	0 to 5	5	4	4
Ability to Hoard	0 to 7	0	0	7
Efficient Scoring	0 to 7	4	5	4
Total Score:		12	10	18

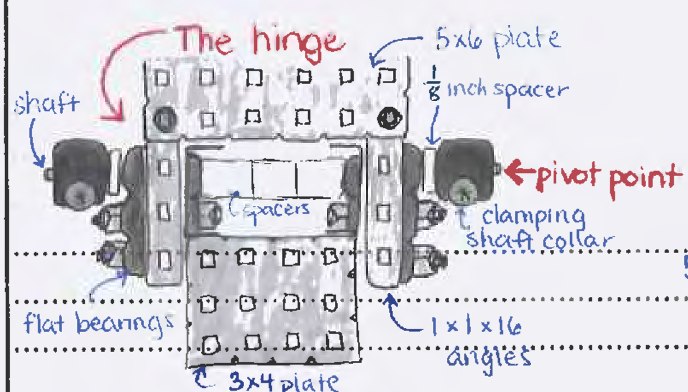
7/22/20
M.M.

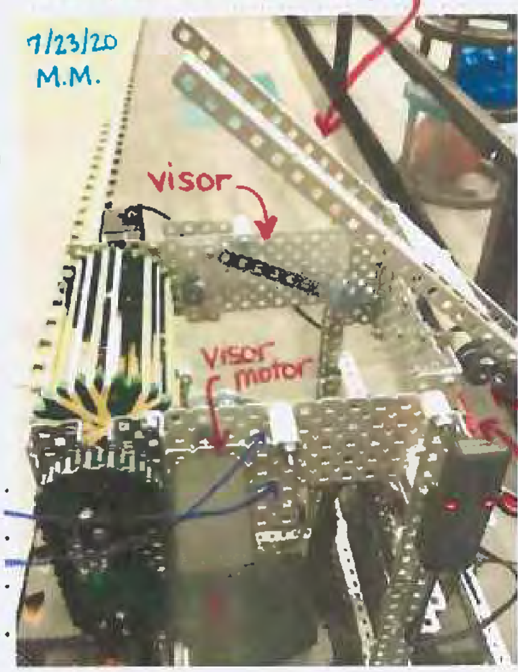
The visor / hood / dunker combination is the best of both worlds: scoring and hoarding. Building will have to be precise so the ball can fit just right.

> BUILD / IMPLEMENT SOLUTION:

7/23/20
M.M.

- The visor motor has a 100rpm gear cartridge.
- I attached a limit switch to make programming simpler (see page 53).





project

designed by:

witnessed by:

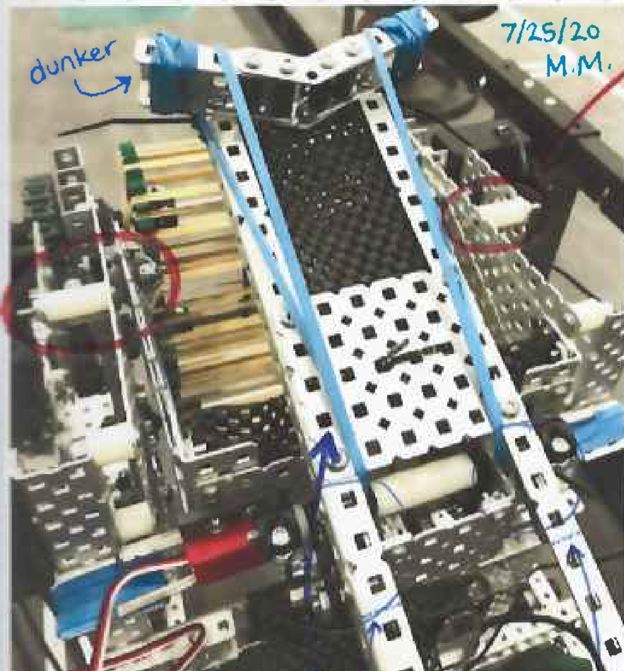
date: 7/20-28/20

Megan M 7/23/20

7/24/20

DESIGN CYCLE: BAT SPACING PROBLEM CONTINUED

> BUILD / IMPLEMENT SOLUTION CONTINUED:



will be cut here

Order on the shaft:
outside → inside

- clamping shaft collar
- $\frac{1}{8}$ inch spacer
- plate
- flat bearing
- $\frac{1}{2}$ inch spacer
- flat bearing
- plate
- washer (nylon)
- $\frac{1}{8}$ inch spacer
- lockbar
- polycarbonate plate
- lockbar
- clamping shaft collar

- I used 1×2×1×5 c-channels and 2 45-Degree gussets on the dunker, and I attached it to the end of the hood.
- I added rubber-bands for compression to keep the hood up right.

Order on the axel:
outside → inside

- 5 smart motor (100rpm)
- plate
- clamping shaft collar
- $\frac{1}{8}$ inch spacer
- flat bearing
- plate
- washer (nylon)
- $\frac{1}{8}$ inch spacer
- lockbar
- polycarbonate plate
- lockbar
- clamping shaft collar

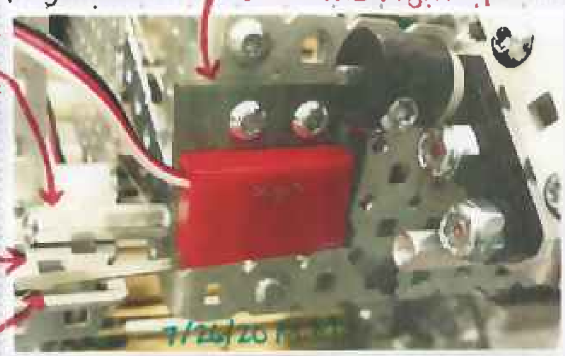
These rubberbands are wrapped around the standoffs used for mechanical stops.

This limit switch will be used to signal the visor motor to stop. It is positioned at just the right spot.

mechanical stop with standoff

2×2×3 angle

limit switch lever



I added a small piece of polycarbonate to make sure that the ball wouldn't catch on the clamping shaft collar

- Having polycarbonate for the sides and aluminum for the back of the visor works well. Sturdy, but lightweight.

Diagonal bar to support the polycarbonate siding

7/27/20

DESIGN CYCLE: BAT SPACING PROBLEM CONTINUED

> **TEST SOLUTION:** Visor and Hood with Dunker combination

- **THE TEST:** I attempted to score several times in both a corner goal and the middle goal with the hood and dunker. I also tried raising the visor and seeing if a ball could fall into where a hoarding bin will be.

- **THE RESULTS:**

  ✓ The hood with dunker scores quickly and accurately in all goals. with three zipties on the top roller to give the ball lift

  ✓ 100 rpm for the visor motor is fast, but does not overheat.

  ✓ The limit switch is positioned so the visor stops at the right height.

Needs improvement but shows promise:

- A ball can fall into the hoarding area, but needs to do so faster.
- The hood has difficulty staying folded down to fit the size limit. Also, the rubberbands keep ripping from the edge of the metal.
- More sensors are needed to efficiently control scoring / hoarding.

Design Cycle

PROBLEM/CHALLENGE

Not enough space for both the bat and hoarding bin

BRAINSTORM

1. Bat and no hoarding bin
2. Hood and second conveyor motor
3. Visor and hood with dunker

SELECT OPTION

3. Visor and hood with dunker

BUILD/IMPLEMENT SOLUTION

- Built and attached visor and hood with hinge
- Attached dunker to hood
- Added limit switch

TEST SOLUTION

Works well, but could use some improvements

PROBLEM SOLVED 7/28/20

project

designed by:

witnessed by:

date: 7/20-28/20

Megan 7/28/20

7/29/20

IMPROVING THE HOOD AND DUNKER

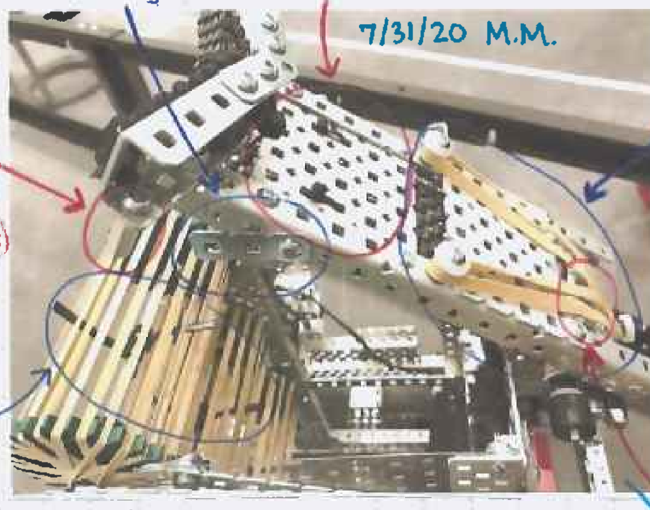
GOAL: Improve the hood and dunker by solving the issues found in testing (pg.54) without interfering with the visor or the rest of the robot.

GOAL COMPLETED 7/31/20

A 1x3 steel bar on each side to help the ball move more smoothly and efficiently

A second 5x6 plate to help the ball flow faster

A standoff on each side to stow easier (catch on the rubberbands)



A better way to attach rubberbands (spacers on a screw) - not stretched as far while still providing enough pull

Zipties on the middle as well as top roller for lift

A small piece of polycarbonate so the rubberbands don't snap on the metal edge

All of these changes work well ~M.M. 7/31/20

Materials used in the Visor:

- 1 - V5 smart motor (100rpm)
- 1 - Limit switch
- 2 - $5\frac{9}{16}$ in. by $2\frac{3}{4}$ in. polycarbonate
- 2 - $\frac{13}{16}$ in. by $2\frac{3}{4}$ in. polycarbonate
- 2 - 2x2x6 aluminum angles
- 2 - 2x2x3 aluminum angles
- 1 - 5x12 aluminum plate
- 2 - 5x5 aluminum plate
- 2 - 5x2 aluminum plate
- 2 - 1x11 steel bars
- 3 - flat bearings
- [illegible] - clamping shaft collars
- [illegible] - lockbars
- 2 - 1 in. standoffs
- [illegible] - nylon washers
- [illegible] - $\frac{1}{8}$ in. spacers
- [illegible] - $\frac{3}{8}$ in. spacers
- 3 - $\frac{1}{2}$ in. spacers
- [illegible] - 0.375 in. screws
- [illegible] - $\frac{1}{2}$ in. screws
- 2 - 0.625 in. screws
- [illegible] - $1\frac{1}{2}$ in. screws
- [illegible] - $1\frac{1}{4}$ in. screws
- [illegible] - thin nylocks
- 8 - nylocks
- 2 - cut shafts
- sections of anti-slip mat
- [illegible] - 4 in. zipties

Materials used in the hood and dunker:

- 1 - $3\frac{1}{4}$ in. by 2in. polycarbonate
- 2 - 1x2x1x5 aluminum c-channels
- 2 - 1x3x1x3 aluminum c-channels
- 2 - 1x1x16 aluminum angles
- 2 - 5x6 aluminum plates
- 1 - 3x4 aluminum plate
- 2 - 2x3 aluminum plates
- 2 - 1x3 steel bars
- 2 - 45-Degree gussets
- 4 - flat bearings
- 2 - clamping shaft collars
- 2 - $\frac{3}{4}$ in. standoffs
- 2 - 1 in. standoffs
- 4 - nylon washers
- 3 - 4.6 mm shaft spacers
- 1 - 8 mm shaft spacer
- 6 - $\frac{1}{8}$ inch spacers
- 6 - $\frac{3}{8}$ inch spacers
- 7 - $\frac{1}{2}$ inch spacers
- 4 - $\frac{1}{2}$ in. locking screws
- 26 - 0.375 in. screws
- 10 - $\frac{1}{2}$ in. screws
- 2 - 0.625 in. screws
- 1 - 0.875 in. screws
- 4 - $1\frac{1}{4}$ in. screws
- 2 - rubberbands
- 9 - thin nylocks
- 34 - nylocks
- 1 - cut shaft
- 5 - 4 inch ziptie
- 1 - 11 inch ziptie
- sections of anti-slip mat

designed by:

witnessed by:

date: 7/29-31/20

Short of adding sensors, the visor/hood/dunker is complete and seems to work wonderfully ~M.M. 7/31/20

Megan TV 7/31/20

8/5/20

IMPROVING THE CONVEYOR BACKING

GOAL: Improve the conveyor backing by recutting the polycarbonate to size with the new visor design (see pages 51-54) and adding guards to prevent the ball from getting stuck on the 18 tooth sprockets.

GOAL COMPLETED 8/5/20

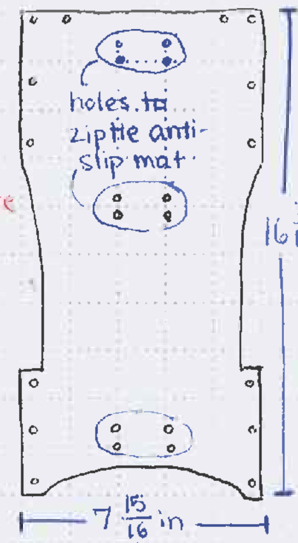
- Now that I no longer am using a bat for scoring, less polycarbonate is needed for the conveyor backing.

- I used spacers to position the guards the right amount in.

$\frac{1}{4}$ in. spacer

$\frac{1}{4}$ and $\frac{1}{2}$ in. spacers

See pg. 42 for initial polycarbonate usage!



over 7 $\frac{1}{2}$ in. of polycarbonate less than used previously

1x13 steel bars for guards

1x2x1x5 c-channels added here to support the polycarbonate

Updated Materials used in the conveyor backing: (see pg. 42 and 50)

- 1 - 7 $\frac{15}{16}$ in. by 16 $\frac{3}{16}$ in. polycarbonate
- 2 - 1x2x1x5 aluminum c-channels
- 2 - 1x13 steel bars
- 2 - $\frac{3}{8}$ in. spacers
- 4 - $\frac{1}{2}$ in. spacers
- 4 - $\frac{1}{4}$ in. spacers
- 6 - thin nylocks
- 16 - nylocks
- sections of anti-slip mat
- 16 - 0.375 in. screws
- 2 - $\frac{1}{2}$ in. screws
- 2 - 1 in. screws
- 2 - 1$\frac{1}{4}$ in. screws
- 8 - 4 in. zipties

from the changes on pg. 50

- removed 2 - 5x8 aluminum plates
- removed 4 - 0.375 in. screws
- removed 4 - nylocks

project | designed by: | witnessed by:

date: 8/3-5/20

Megan M 8/5/20

8/8/20

IMPROVING THE VISOR AND ZIPTIES ~

GOAL: Modify the conveyor (rubberbands, zipties) and visor to push the ball into the hoarding area faster (more force).

GOAL COMPLETED 8/17/20

- The zipties added to the top two rubberband rollers on 7/29-31/20 (see pg. 55) definitely help with some speed, but it still needs more speed / a greater force to push the ball.
- These zipties also got bent easily and move left to right on the shaft (didn't stay put).

These zipties allow for faster scoring and hoarding ∴

14 longer zipties on top roller; 6 longer and 8 shorter (to hold the zipties in place) on middle roller

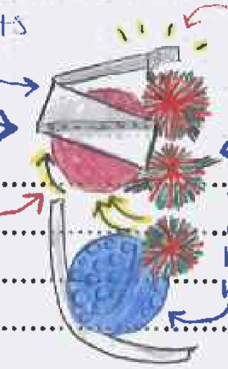
- I added more zipties so that there are 14 cut 11" zipties on each of the top two rubberband rollers (on the high strength shafts).

These zipties give the ball plenty of thrust to be scored in a goal ✔

- These zipties solve the problem of pushing the ball into the hoarding area, but they cause the visor to be unstable.

PROBLEM WITH VISOR INSTABILITY:

- The force from the zipties causes the visor to jolt upward when contacted by a ball, resulting in the ball to get stuck.



(The brakeType for the VisorMotor is already set to hold, but the small amount an axel can turn when on hold is multiplied over the distance from the pivot; what usually isn't a big deal matters a lot here.)

designed by:

witnessed by:

date: 8/8-17/20

Megan M 8/10/20

8/11/20

DESIGN CYCLE: VISOR JOLTING PROBLEM

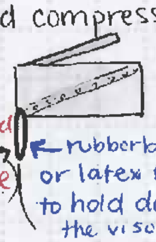
See pg 57 for more detail!

PROBLEM / CHALLENGE: The force from the zipties that allows balls to be scored and hoarded quickly also causes the visor to jolt upward.

See pg. 57 for more detail!

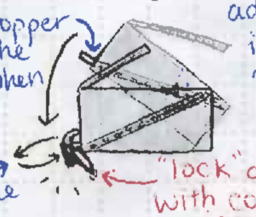
BRAINSTORM: Possible solutions to the problem:

① Add compression.



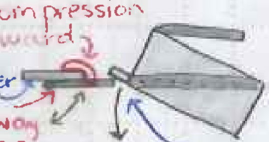
- X Could strain the motor
- X Would be ineffective because it would be tight only when pulled up
- X Causes problems with hoarding

② Add a locking mechanism.



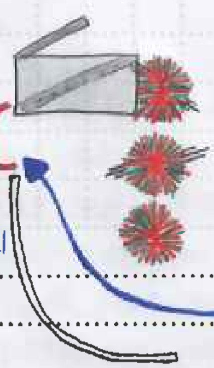
- X Either too easy for the ball to push up, or too difficult for the visor motor to "unlock"
- X Could strain the motor

③ Add a one-way flap.



- X Counter productive. Cannot raise visor to hoard
- X The nature of one-way flaps would mean the visor would have a bit of leeway before hitting the stopper.

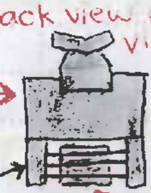
④ Add a structure to lessen the impact of the ball against the visor.



- X Prevents hoarding
- X The visor would still go up at least some, which causes the same issues as before.

This gap leaves room for a ball to be pushed into the hoarding bin

⑤ Add a shock absorber to dampen the force of the ball against the visor.



- X The visor will have to be raised higher to hoard a ball.

This would take the place of the gap

project

designed by:

witnessed by:

date: 8/8-17/20

Megan W. 8/12/20

8/11/20

DESIGN CYCLE: VISOR JOLTING CONTINUED

SELECT OPTION:

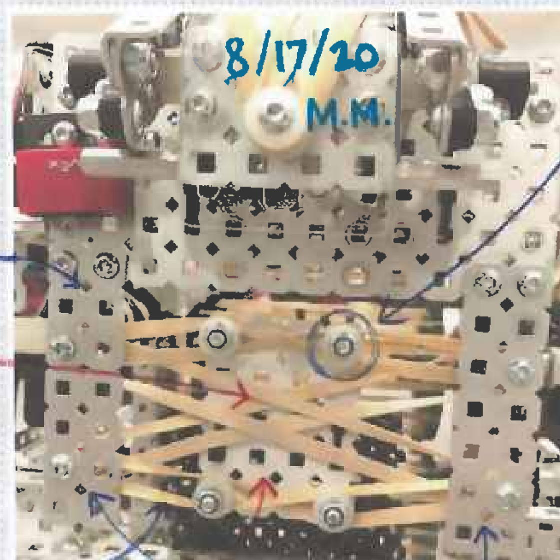
Visor Jolting Possible Solutions

Criteria (least to most important)	Scale	① Compression	② Locking Mechanism	③ One-Way Flap	④ Metal Structure	⑤ Shock Absorber
Does not strain the visor motor	0 to 3	1	1	2	3	3
Prevents the visor from jolting	0 to 5	0	3	1	2	4
Does not prohibit hoarding	0 to 5	1	3	0	1	4
Total Score:		2	7	3	6	11

8/12/20 M.M.

→ The shock absorber is the most viable solution by far.

BUILD / IMPLEMENT SOLUTION:



spacers of different diameters

prevents rubberband from falling off

- I attached two 2x7 plates to serve as the shock absorber structure, one on each side off the back of the visor.
- The shock absorber itself is a 5x4 plate suspended by rubberbands.
- I tried various rubberband configurations, but this one seemed to work and stay on the best.

shock absorber structure

Shock absorber suspended with rubberbands

too loose

doesn't stay on

rubberbands are stretched too far to have springiness

most successful rubberband configuration

designed by:

witnessed by:

date: 8/8-17/20

Megan T. 8/12/20

8/13/20

DESIGN CYCLE: VISOR JOLTING CONTINUED

BUILD / IMPLEMENT SOLUTION CONTINUED:

Materials used for the Shock Absorber:

- 2 - 2x7 aluminum plates
- 1 - 4x5 aluminum plate
- 4 - 0.375 in. screws
- 8 - 0.625 in. screws
- 12 - thin nylocks
- 8 - $\frac{1}{8}$ in. spacers
- 8 - 4.6 mm shaft spacers
- 3 - 4" zipties
- 4 - rubberbands
- one section of anti-slip mat
- 28 - 11" zipties (on the conveyor)

TEST SOLUTION: Does the shock absorber prevent the visor jolting?

- **THE TEST:** I ran a number of balls through the conveyor, both scoring and hoarding.
- **THE RESULTS:**
  - ✓ The shock absorber works as intended → the visor no longer is jolted upward when contacted by a ball.
  - ✓ Throughout testing, not a single ball got stuck in the conveyor/visor.
  - ✗ Balls are still able to be hoarded, but the visor has to be raised up much higher than before (possibly a good spot for a sensor?) so that it comes very close to the conveyor.

PROBLEM/CHALLENGE
The zipties on the conveyor cause the visor to jolt upward when contacted by a ball.

BRAINSTORM
1. Compression
2. Locking Mechanism
3. one-way flap
4. Metal structure
5. Shock Absorber

SELECT OPTION
5. Shock Absorber

BUILD/IMPLEMENT SOLUTION
- Built the shock absorber
- Experimented with rubberband configurations

TEST SOLUTION
Shock absorber solves the problem without preventing hoarding.
PROBLEM SOLVED 8/17/20

Design Cycle

project | designed by: | witnessed by:

date: 8/8-17/20

Megan T 8/16/20

8/19/20

MINOR HOOD & VISOR IMPROVEMENTS

GOAL: Improve the performance of the hood and visor when scoring and hoarding to make the most of recent modifications (see pgs. 57-60).

GOAL COMPLETED 8/22/20

- Now that I have a stronger force to thrust the balls into a goal or the hoarding area (see page 57) and the shock absorber to prevent the visor from jolting (see page 59-60), I decided to experiment with the pressure on the hood.
- I found that having pressure both upward and downward would cause the ball to "pop" out and have a higher launch than with pressure in only one direction.

See page 59

New rubberband configuration on hood

4:2 ratio

pressure up

pressure down

maximizes the force scoring the ball

6 rubberbands; 2 pulling downward, 4 pulling upward

Materials used / changed on the Hood from 7/31/20 (see page 55):

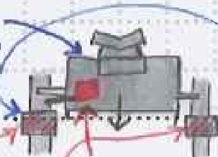
- removed 2 - 1x3 steel bars
- removed 2 - 0.375 in. screws
- removed 2 - nylocks
- 4 - additional rubberbands
- 2 - 0.625 in. screws
- 2 - $\frac{1}{8}$ in. spacers
- 2 - 4.6 mm shaft spacers
- 2 - thin nylocks

- With this new rubberband configuration, the dunker (see page 51-54) is no longer essential for scoring, but it can be used if needed to score a ball. (such as if a rubberband / ziptie breaks, or if there's "ball traffic".

- I added a small square of polycarbonate of each of the mechanical stops at the visor's resting position.

This ensures the limit switch's lever doesn't get caught in a hole on the angle

Back view of visor



These angles prevent the visor from going any lower

polycarbonate here

limit switch (see pg. 53)

also

It prevents the angles from getting beat up after repeated use

designed by:

witnessed by:

date: 8/19-22/20

Megan M 8/21/20

8/21/20

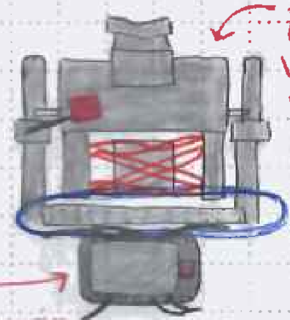
MINOR HOOD & VISOR IMPROVEMENTS CONTINUED

→ Materials used in adding polycarbonate to the visor's stops (see pg. 61): ←

- 4 - 0.375 in. screws
- 4 - nylocks
- 2 - 1 in. by $1\frac{7}{16}$ in. polycarbonate

Improvement #3

- I added a 2×12 plate to serve as a ledge over the hoarding area.
  - → The ball can roll off of this into the hoarding bin
  - → The ledge also doubles as wires-and-brain protection (it prevents a ball from hitting the brain)



→ Materials used on the ledge: ←

- 1 - 2×12 aluminum plate
- 2 - 0.375 in. screws
- 2 - nylocks

NEW SCHEDULE & DEADLINES 8/23/20

- My original timeline didn't go exactly according to plan, as I did not anticipate such major design changes, but thankfully I added some extra time, so I can still be ready for competitions.
  [see page 35 for initial timeline]

- Now that school is about to start and tournaments are being scheduled, I need a new timeline so I can plan my time to be ready for competitions.

project designed by: witnessed by:

date: 8/19-22/20

Megan M 8/22/20

8/23/20

NEW SCHEDULE & DEADLINES

August

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

September

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

October

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

8/23/20 M.M.

Timeline until the first competition of the season!

Calendar Key:

- robot work day
- possible robot work day if I can get my school work done
- robotics competition
- online challenge due

PROJECT TIMEFRAMES and DEADLINES:

- I still have a lot to finish in building the robot, but I need to be done by the beginning of October at the very latest.
- This leaves about a month for sensors, macros, autonomous (all the programming).
- I need to practice driving both for the matches and for driver skills as much as possible - a little each day as soon as the macros are finished.

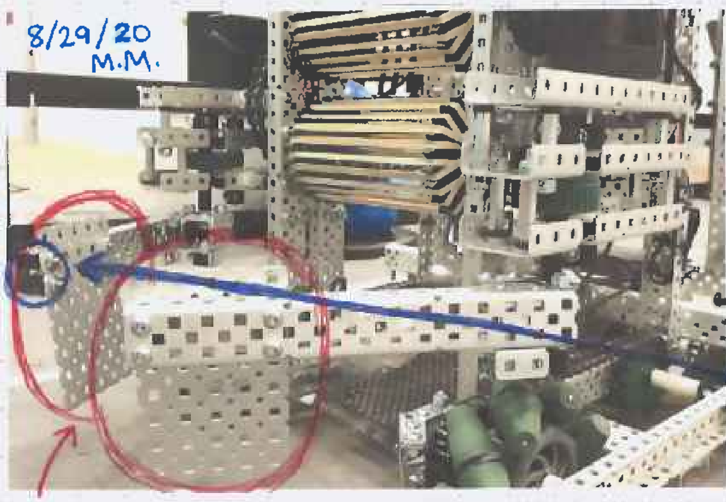
- Oct. 30-31 - competition in Brentwood
- Nov. 21 - competition in Mountain City -CANCELED-
- Dec. 12 - competition in Bluff City
- Jan. 22-23 - Kalahari Classic signature event in Ohio
- Oct. 31 - VRC Annotated Programming Skills online challenge due
- Feb. 10 - Encore online challenge due

Megan M 8/23/20

8/29/20

FLIPPER ADD-ONS AND PROBLEMS

GOAL: Finish building the flippers and test their effectiveness at descoring and picking up balls. **GOAL COMPLETED 8/29/20**



Flipper Addons, which extends the reach down to the level of the ball

- The only remaining part to build of the flippers is the add-ons (mounted and last worked on flippers 7/9/20, see pages 45-48).
- I used uneven spacers to mount these two plates at a slight angle, which helps to pop the ball out of the goal easier.

TESTING: I ran some simple tests with the flippers - descoring a side goal and the middle goal, and picking up a ball off the field.

- ✗ The flippers got caught on each other when folding inward to push a ball up to the conveyor.
- ✗ Could not descore the middle goal due to catching on the poles.
- ✗ Takes a very long time to descore a side goal, and cannot fit in the corner goals (hits the field perimeter).

8/30/20 Based on testing the flippers, I feel that it may be time for an intake design change. While modifications could be made to improve the flippers, it would be very difficult and time-consuming. These problems are things which I could not test in prototyping (see pages 16-17).

Megan T. 8/29/20

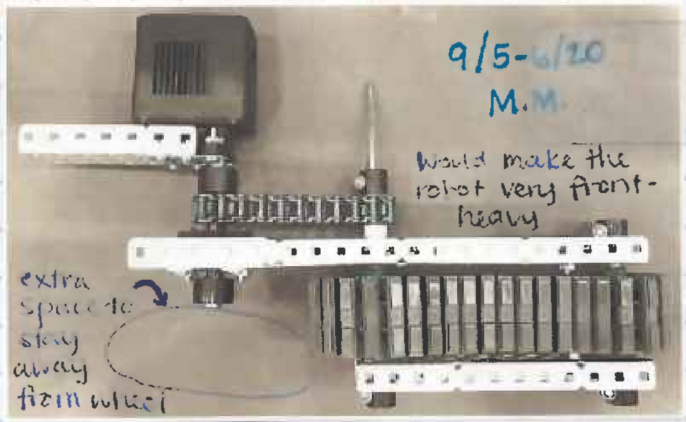
9/5/20 Intake Design Cycle : Brainstorm

PROTOTYPING INTAKE ROLLERS

GOAL: Find an intake design with rollers that will both fit within the size limit and perform the necessary functions of descoring and picking up balls off the field. **GOAL COMPLETED 9/19/20**

- Looking back at my original intake design decision matrix (see pg.17), my second place design is intake rollers - this is what I plan to build next.

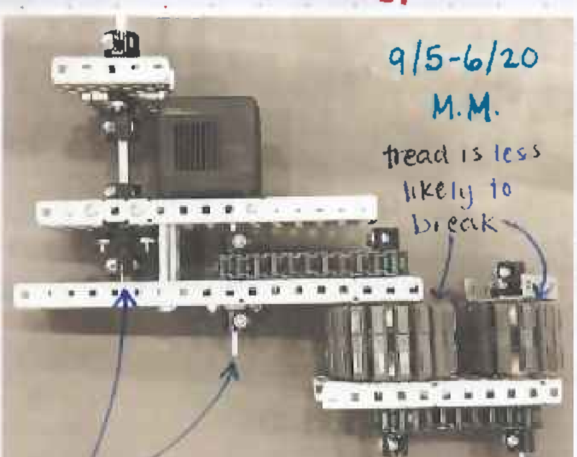
First Prototype:



- ✗ Too big to fit in the size limit
- ✗ Pivot to stow cannot be on the axel through the motor

- Intake rollers are tricky:
  - → Must reach far enough out to descore, but must be short enough to stow
  - → Must be low enough to reach the ball, but must be high enough to not hit the base when stowed
  - → Must be wide enough to go into a goal, but narrow enough to pick up a ball.

Second Prototype:



separate pivot point from motor axel

- Both of my first two prototypes were not functional, but I saw a lot of the issues that need to be addressed with any design.

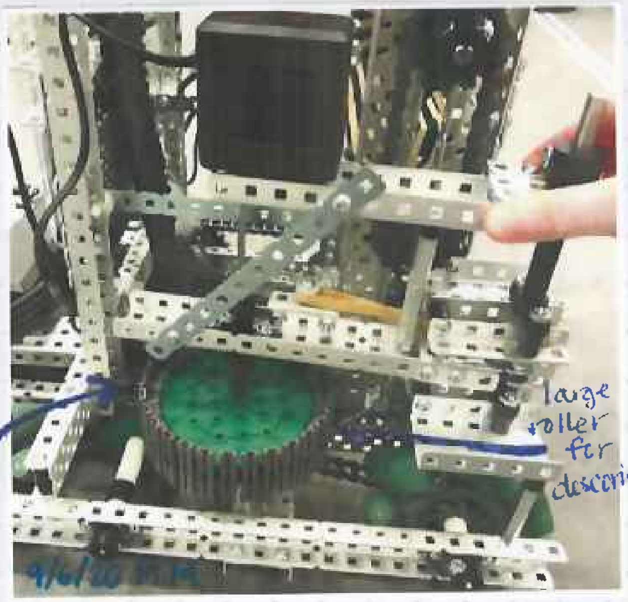
- ✗ Too big to fit in the size limit
- ✗ Too tall to fit in a goal
- ✗ The front roller is too small to reach a ball in a goal without the intake being at an extreme angle.

Megan N 9/6/20

 9/12/20 Intake Design Cycle: Brainstorm

PROTOTYPING INTAKE ROLLERS CONTINUED

Third Prototype:



large roller for descoring

Much better than the other prototypes so far in terms of length
(this is the first one that can stow)

✗ The motor hits the goal and prevents compression inward when descoring.

✗ Only a small portion of the roller actually fits inside the goal.

Fourth Prototype:

This design required additional structure but is the most stable by far.

compression inward and folds back to stow

medium sized roller

✗ Has some difficulty reaching balls in the back of goals

Intake Design Cycle: Select Option

Of my four prototypes, the last one is only one that shows promise and is able to fit within the size limit. Rollers intakes are tricky, and you have to find a balance that meets all the requirements - the fourth prototype seems to do just that.

project

designed by:

witnessed by:

date: 9/12-19/20

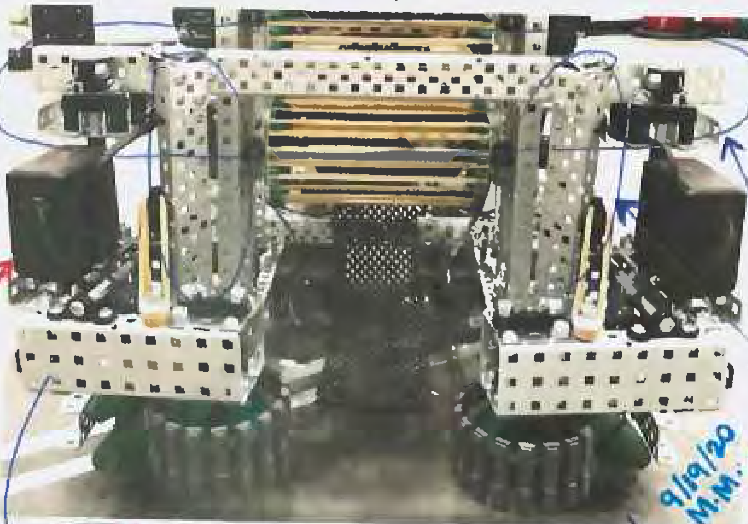
Megan M 9/19/20

9/19/20

INTAKES: BUILDING, TESTING, IMPROVING

GOAL: Finalize the intake rollers, and make improvements if needed after testing.

GOAL COMPLETED 9/20/20



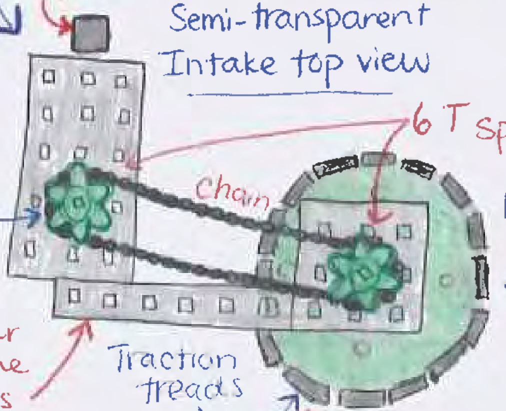
200 rpm motors

Intake Design Cycle: Build / Implement Solution

- I built the intakes very similar to the prototype (see page 66).
- The support structure can serve as a bumper against a goal.

This gusset helps to attach the vertical bars to the horizontal bar at an angle

The intakes have ≈270° of motion and fold around to stow



- Rubberbands provide compression inward (which helps with popping out from stowed position as well as intaking balls) and a metal stopper prevents the intakes from coming too far in.

Intake Design Cycle: Test Solution

- The traction treads don't grip the balls well, and they tend to severely scratch up the balls in the process of intaking
- The stopper is bending from the force of the compression slamming the intakes into it, so this needs to be replaced
- The intakes are the perfect height for descoring and the correct distance from the conveyor to pick up a ball.
- The rubber bands catch on the goal when descoring.

Megan M 9/20/20

9/20/20

INTAKES: BUILDING, TESTING, IMPROVING CONT.

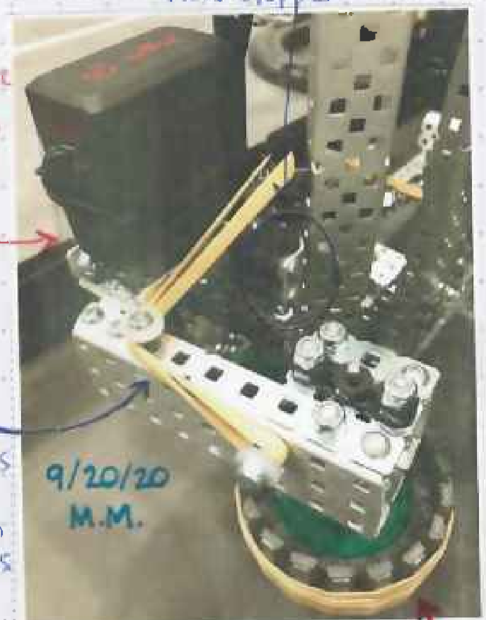
- After testing the intakes (see pg. 67), I need to find a method to hold them in stowed position for the start of a match until they need to be released, as well as make some improvements.

- Small improvements:
  - added a small polycarbonate guard on the outside of each intake to protect the motor from defense when descoring.
  - redirected the rubberbands that provide compression to stay away from the goal.
  - replaced the stopper (see pg. 67) with a 90-Degree gusset that shouldn't bend as easily
  - added rubberbands to the outside of the traction treads which provides a much better grip.

new stopper

added polycarbonate guards to protect the motors

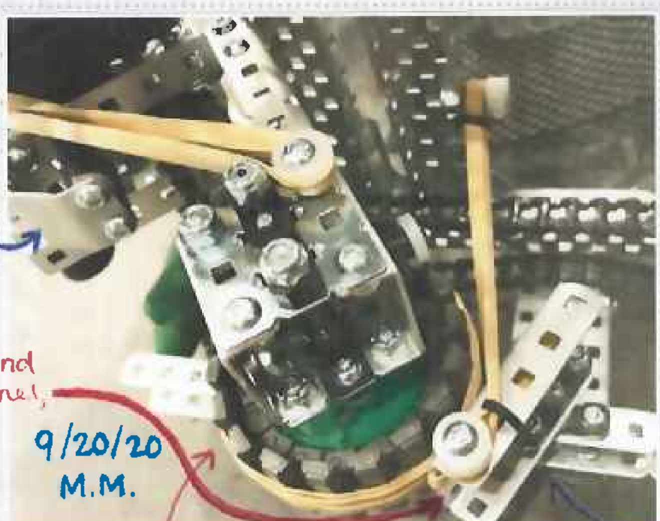
changed the path of the rubberbands to not catch on the goals



Added rubberbands around the traction treads for better grip

new stopper out of a 90° gusset

There is a rubberband wrapped around this c-channel, and the grip helps the intake stay stowed when needed



Intake in stowed position

- Stowing:
  - I tried to wrap a single rubberband around a tread link, but it could not hold.
  - Instead, I added a pivoting c-channel that has compression inward and essentially "hugs" the intake. To release, just outtake slightly, and it pops out easily.

- These improvements make a huge difference in intake performance.

project | designed by: | witnessed by:

date: 9/20/20

Megan M 9/20/20

9/20/20

ROLLER INTAKES CONTINUED ~

→ Roller Intakes Materials : ←

- 2 - V5 Smart Motors (200rpm)
- 1 - 1x2x1x31 aluminum c-channel
- 2 - 1x2x1x10 aluminum c-channels
- 2 - 1x2x1x5 aluminum c-channels
- 2 - 1x3x1x9 aluminum c-channels
- 2 - 1x3x1x3 aluminum c-channels
- 4 - 1x3x1x5 aluminum c-channels
- 2 - 2x2x2 aluminum angles
- 2 - 1x2x1x16 aluminum c-channels
- 4 - 1x2x2 aluminum angles
- 8 - 1x1x2 aluminum angles
- 2 - 1x1x1 aluminum angles
- 2 - 1x4 steel bars
- 2 - 1x2 steel bars
- 2 - 1x5 steel bars
- 4 - 2x2 steel plates
- 4 - 3x3 steel plates
- 4 - 1x3 H.S. lock bars
- 2 - 1x6 H.S. lock bars
- 8 - 1 in. standoffs
- 2 - pivot gussets
- 2 - 90-Degree gussets
- 4 - H.S. clamping shaft collars
- 6 - clamping shaft collars
- 4 - post hex nut retainer w/ flat bearing
- 4 - shafts (cut)
- 2 - H.S. shafts (cut)
- 10 - H.S. flat bearings
- 4 - flat bearings
- 2 - 18T sprockets
- 4 - 6T sprockets
- 36 - traction tread links
- 2 - Sections of chain
- 2 - 4" zipties
- 2 - 11" zipties
- 10 - rubberbands
- 2 - ≈4 in. x 2 1/2 in. polycarbonate
- 2 - ≈ 2 in. x 1/2 in. polycarbonate
- 94 - nylocks
- 32 - thin nylocks
- 74 - 0.375 in. screws
- 16 - 1/2 in. screws
- 4 - 1/2 in. locking screws
- 10 - 0.625 in. screws
- 10 - 0.875 in. screws
- 10 - 1 1/4 in. screws
- 4 - 1 1/2 in. screws
- 2 - 2 in. screws
- 20 - 1/8 in. spacers
- 6 - 1/4 in. spacers
- 12 - 3/8 in. spacers
- 18 - 1/2 in. spacers
- 18 - nylon washers
- 6 - H.S. plastic washers
- 2 - H.S. 1/8 in. spacers
- 4 - H.S. 1/4 in. spacers
- 10 - H.S. 1/2 in. spacers

Materials used to stow the Intakes on page 70 →

Megan M 9/20/20

9/20/20

ROLLER INTAKES CONTINUED

Materials used for stowing the intakes:

- 2 - 1x3x1x5 aluminum c-channels
- 6 - flat bearings
- 6 - rubberbands
- 4 - 4" zipties
- 8 - nylocks
- 6 - thin nylocks
- 4 - ½ in. screws
- 8 - 0.875 in. screws
- 2 - 1½ in. screws
- 6 - ¼ in. spacers
- 4 - 8 mm. shaft spacers
- 4 - nylon washers

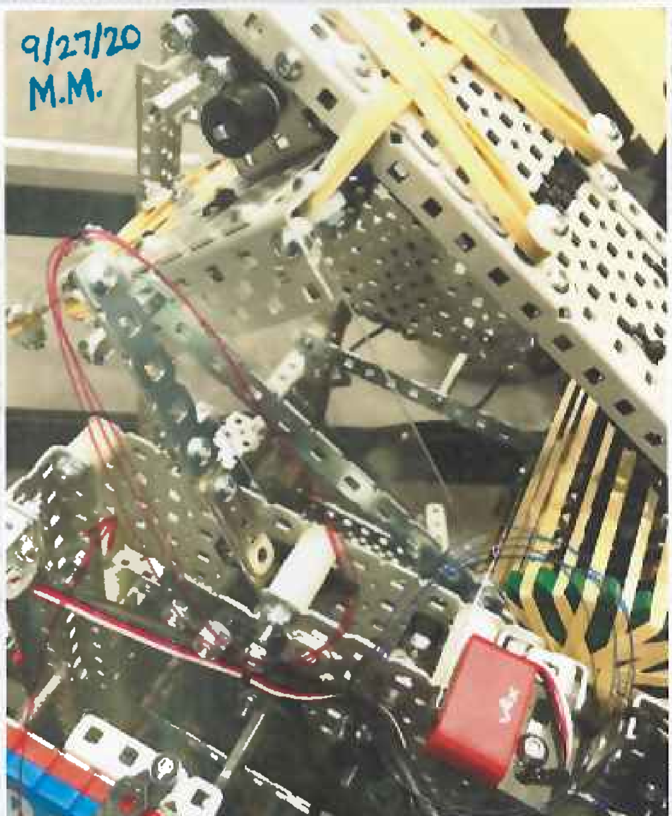
SENSORS AND OTHER IMPROVEMENTS

GOAL: Add sensors where needed and make any final improvements in preparation for programming.

GOAL COMPLETED 10/12/20

LIMIT SWITCH ON VISOR:

- The limit switch added on 7/26/20 (see pg. 53) takes care of the low end of the visor's movement, but a second one is needed for the upward movement.
- For extra protection, I added a mechanical stop as well. With enough force, it could get bent, but hopefully it will hold for a short time against unwanted upward motion outside the range the visor needs to be in.



mechanical stop, just in case

limit switch to stop upward motion

project

designed by:

witnessed by:

date: 9/20, 26-27

Megan T 9/27/20

9/27/20

SENSORS & OTHER IMPROVEMENTS CONTINUED

(see pg 70)

→ Materials used to mount limit switch: ←

- 1 - 1×2×3 aluminum angle
- 1 - Limit switch
- 1 - nylock
- 2 - thin nylocks
- 2 - nylon washers
- 2 - 0.375 in. screws
- 1 - 0.625 in screws

→ Materials used in visor's mechanical stop: ← (see pg 70)

- 1 - 1×1×10 aluminum angle
- 2 - lock bars
- 1 - 1×5 steel bar
- 1 - 3×4 aluminum plate
- 1 - 1in. standoff
- 1 - flat bearing
- 2 post hex nut retainer w/ flat bearing
- 2 nylon washers
- 1 - $\frac{3}{8}$ in. spacers
- 3 - $\frac{1}{2}$ in. spacers
- 7 - 0.375 in. screws
- 3 - $\frac{1}{2}$ in. screws
- 1 - 1$\frac{3}{4}$ in. screw
- 4 thin nylocks
- 6 nylocks

LICENSE PLATES & HOLDER:

To change color, simply turn these and flip the plate around. Then turn the retainers back to hold the plate in place.

- I made a license plate holder and attached one of each side of the robot using standoffs.
- This design allows for easy, fast flipping to the other color, while having little risk of becoming unintentionally flipped in a match
- The backing is an aluminum plate to comply with <R27a> in the game manual (opposite color is covered).

→ Materials used in license plates & holders: ←

- 2 - Red VRC License Plates
- 2 - Blue VRC License Plates
- 8 - VEX IQ pins from the VRC License Plate kit
- 4 - 1×1×4 aluminum angles
- 2 - 5×11 aluminum plates
- 2 - 1in. standoffs
- 2 - 3in. standoffs
- 8 - nylon washers
- 12 - 0.375 in. screws
- 6 - 0.625 in. screws
- 6 - $\frac{1}{2}$ in. locking screws
- 4 - post hex nut retainers
- 6 - $\frac{1}{4}$ in. spacers
- 4 - $\frac{1}{2}$ in. spacers
- 16 nylocks
- 4 thin nylocks
- 4 - post hex nut retainers w/ flat bearing

project designed by: witnessed by:

date: 9/27, 30/20

Megan M 9/30/20

9/30/20

SENSORS & OTHER IMPROVEMENTS CONTINUED

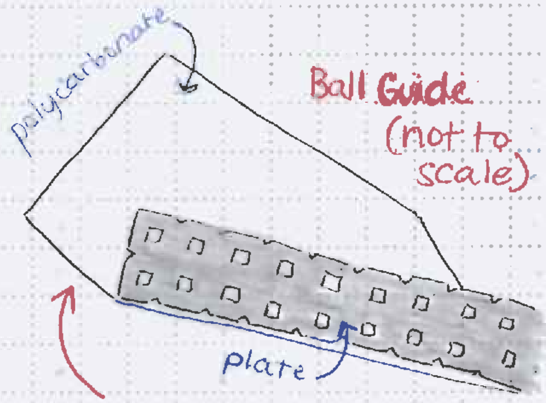
POLYCARBONATE BALL GUIDES:

Compression to stay up at this angle

Ball Guides after unstowing

holds nylon rope in place

- I added a pop-out ball guide on each side of the conveyor to help defend against defense when scoring.
- $\frac{1}{8}$ in. braided nylon rope (legal per <R8e>) holds the ball guides in stowed position until the intakes pop out, releasing the rope.
- These ball guides do not interfere with the hood or the visor at any time.



Shaped specifically to fit within the size limit when folded down

This angle helps to channel the rope

Ball Guide in stowed position

nylon rope wrapped in the intakes holds the guides in the size limit

→ Materials used on the Ball Guides: ←

- 2 - 10x2 aluminum plates
- 2 - 1x2 steel bars
- 1 - 2x2x5 aluminum angle
- 1 - 1x2x1x3 aluminum c-channel
- 2 - flat bearings
- 2 - pillow block bearings
- 2 - rubberbands
- 2 - 4" zipties
- 2 - $\frac{1}{2}$ in. standoffs
- 8 - nylon washers

- 2 - ≈$4\frac{1}{4}$ in. by $2\frac{1}{4}$ in. polycarbonate
- 2 - sections of $\frac{1}{8}$" nylon braided rope
- 6 - 0.375 in. screws
- 2 - 0.625 in. screws
- 4 - 0.875 in. screws
- 2 - 1 in. screws
- 1 - $\frac{3}{8}$ in. spacer
- 2 - $\frac{1}{2}$ in. spacers
- 2 - 8 mm shaft spacers
- 14 - nylocks

project

designed by:

witnessed by:

date: 9/30 - 10/1/20

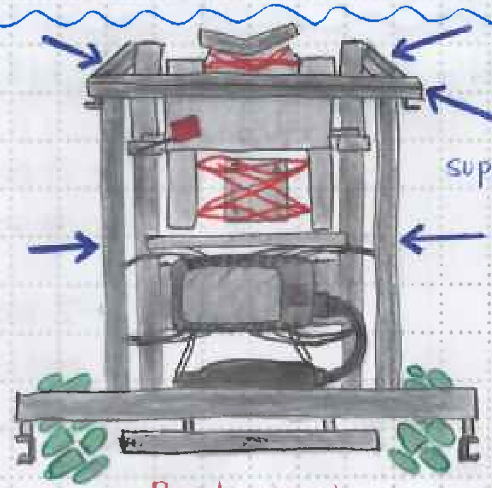
Megan M . 10/1/20

10/3/20

SENSORS & OTHER IMPROVEMENTS CONTINUED

BACK SUPPORT BARS:

- I added several support bars to the back of the robot.
  - Extra stability
  - Helps to counter-balance the weight on the front of the robot
  - Helps protect the brain, battery, and visor



support bars added

Back view of the robot

- These bars are spaced to allow a ball to be pushed out the back without interference.

These support bars form a frame around the back of the robot

Materials used on the Support bars:

- 1 - 1×1×22 aluminum angle
- 2 - 1×2×1×25 aluminum c-channels
- 2 - 1×2×1×10 aluminum c-channels
- 2 - Structural L-shaped gussets
- 8 - $\frac{1}{4}$ in. spacers
- 16 - 0.375 in. screws
- 4 - 0.875 in. screws
- 20 - nylocks

DISPOSING over HOARDING

Because the robot is functional as it is now, I decided to use the raising-the-visor-and-sending-a-ball-out-the-back for disposing an unwanted ball, rather than hoarding for now. This will leave more time for driving practice and programming before my first competition.

10/3/20 M.M.

I plan to revisit hoarding as a potential design strategy after the competition on Oct. 30-31 (see pg. 63)

project designed by: witnessed by:

date: 10/3/20

Megan M 10/3/20

 10/3/20

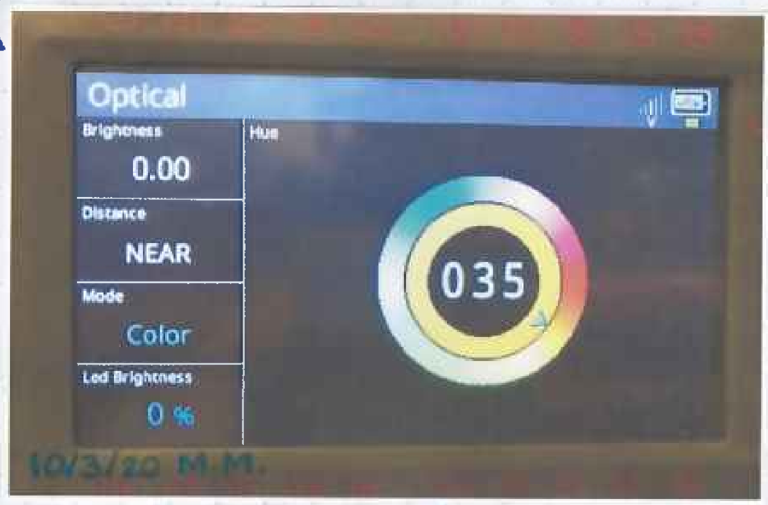
SENSORS & OTHER IMPROVEMENTS CONTINUED

TESTING THE OPTICAL SENSOR:

The positioning of the sensor when the brain displays this

* "a combination of light, color, proximity, and gesture sensor"

- Today I connected a new optical sensor* to the brain for the first time.
- I tested the optical sensor through the devices menu on the brain.



The optical sensor readings from a blue ball on the brain screen

- The optical sensor detects the correct hue for red balls, but it registers blue balls as yellow, light orange, or neon green.
- I tried using different lighting, angles, and changing the LED brightness on the sensor, but it never returned a blue hue.
- The optical sensor also does not detect gestures when in gesture mode on the devices menu on the brain.
- THE RESULTS: Based on the unpredictable results from experimenting with the optical sensor, I believe that it would be very difficult and time-consuming to implement effectively.

I hope to return to the optical sensor in the future, but there isn't enough time until the first competition to get it working well, so I plan to use distance sensors to find objects on the field instead. ~M.M. 10/3/20

Megan M 10/3/20

10/10/20

SENSOR STRATEGY

Tasks that need sensors (for driver control and especially autonomous)	Sensor options that can accomplish the task in question	Selected sensor
defined range of motion for the visor	potentiometer, optical shaft encoder, limit switch, bumper switch, inertial sensor?	limit switches (already implemented, see pages 59 and 70)
drive to goals around the field	line trackers, vision sensor	line trackers
score balls in goals	optical sensor, distance sensor, limit switch, inertial sensor	inertial sensor
find and intake ball on the field	optical sensor, vision sensor, distance sensor	distance sensor
ball positioning in conveyor and intakes	distance sensor, optical sensor, line trackers	distance sensor
dispose balls	distance sensor, bumper switch, limit switch	limit switch
maneuver the field 10/10/20 M.M.	motor encoders, vision sensors, distance sensor, inertial sensor, bumper switch	inertial sensor and distance sensors

— for use in autonomous
— for use in driver control

- The new V5 distance sensors will prove very useful, especially with the level of accuracy and precision that they provide.
- If a firmware update comes out and I have time to learn to use the optical sensor effectively, this would be a good option for several o these tasks listed above (see page 74)

Megan M 10/10/20

 10/10/20

IMPLEMENTING SENSORS

LINE TRACKERS:

Line trackers attached to the bottom of the robot to see / follow the tape on the field

- Line trackers can be programmed to return the reflectivity value of a surface (therefore able to distinguish a grey tile from white tape).
- They are the most accurate when mounted ≈ 3 mm above the surface that they are sensing.

Materials used to mount line trackers:

- 2 - Line trackers
- 1 - 1x1x24 aluminum angle
- 1 - 1x3 steel bar
- 5 - nylocks
- 2 - 0.375 in. screws
- 3 - 0.875 in. screws
- 2 - $\frac{1}{4}$ in. spacers
- 1 - $\frac{3}{8}$ in. spacer

INERTIAL SENSORS: HOOD AND BASE

Inertial Sensor attached to the base of the robot for accurate turns

- The inertial sensor is a combination of a gyroscope and an accelerometer.
- The yaw value can be used for accurate and consistent turning.

Inertial Sensor attached to the hood for accurate scoring

- The pitch value can be used to know if a ball has been scored. The hood always raises and then lowers once the ball has been launched.

project

designed by:

witnessed by:

date: 10/10-12/20

Megan M[redacted] 10/10/20

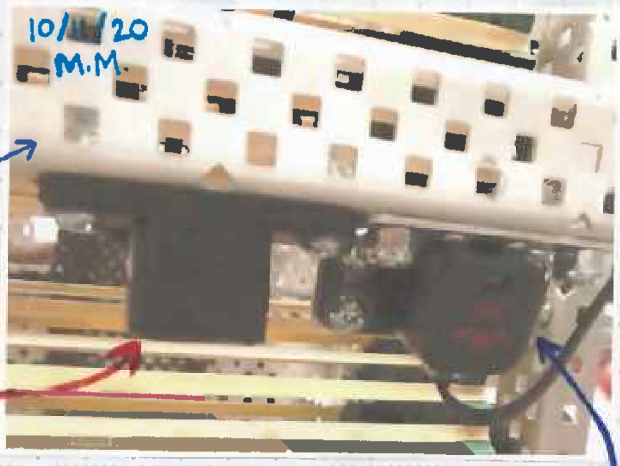
10/10/20

IMPLEMENTING SENSORS CONTINUED

→ Materials used to mount inertial sensors ← (see page 76)

- 2 - Inertial sensors
- 1 - 1×3×1×4 aluminum c-channel
- 2 - nylon washers
- 2 - 0.375 in. screws
- 2 - 0.875 in. screws
- 4 - nylocks

DISTANCE SENSORS : BALL POSITIONING AND LOCATING :



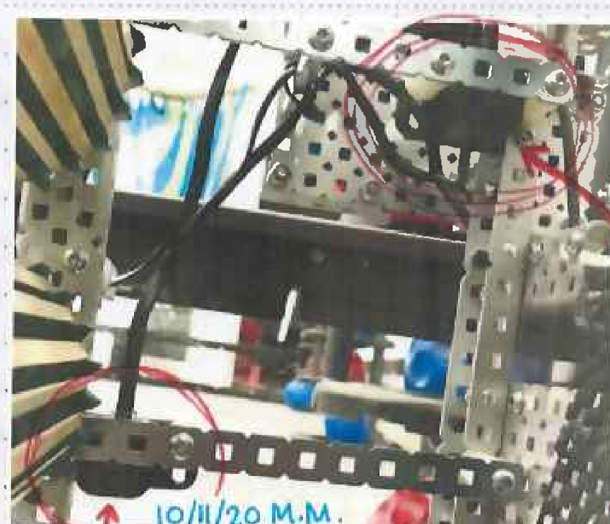
- The new V5 Distance Sensors are the best choice for determining ball positioning in the conveyor. The data returns distance to the mm, which allows me to evaluate not only if a ball is present, but also what part of the ball is in front of the sensor (edge or center).

Measures distance to determine if and how far a ball is into the intakes. Can be used to know if the robot is at a goal.

Measures distance to the nearest object in front of the robot (goal, ball, field perimeter). Can be used to line up with a ball and drive to a goal.

[**5** Distance Sensors]

Mounted to the back of the robot. Measures distance to the nearest object behind the robot. Can be used to ensure accurate driving foward.



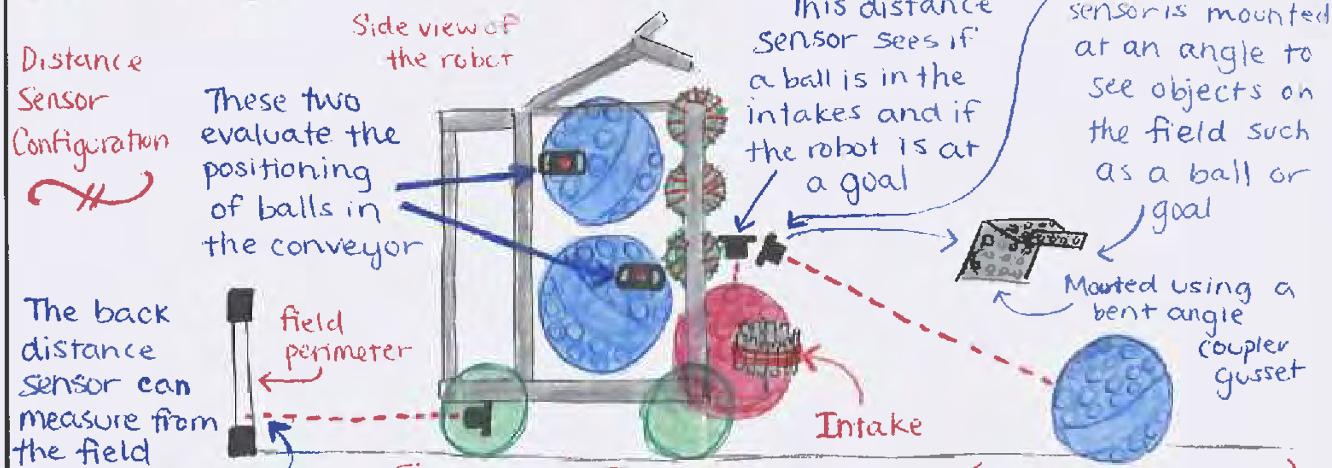
Measures distance to determine if and what part of the ball (ie. edge, center) is in the middle or top of the conveyor, respectively

Megan M 10/11/20

10/11/20

IMPLEMENTING SENSORS CONTINUED

DISTANCE SENSORS CONTINUED:



[see page 77] (not drawn to scale)

Materials used to mount the distance sensors:

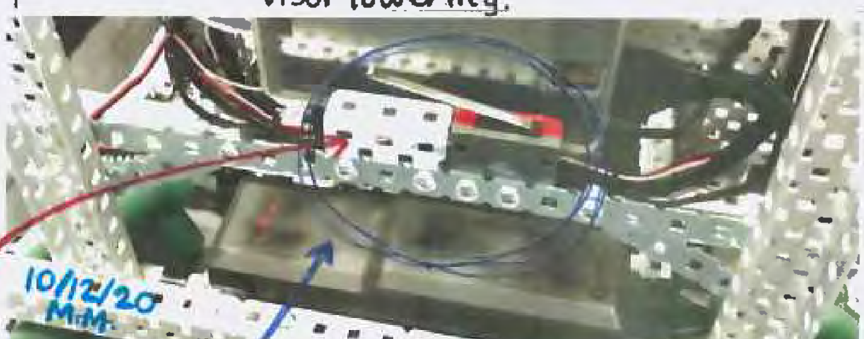
- 5 - Distance Sensors
- 1 - Angle Coupler gusset
- 16 - nylon washers
- 4 - nylocks
- 4 - $\frac{1}{4}$ in. spacers
- 4 - $\frac{3}{8}$ in. spacers
- 6 - $\frac{1}{2}$ in. spacers
- 11 - thin nylocks
- 2 - 0.375 in. screws
- 5 - $\frac{1}{2}$ in. screws
- 2 - 0.625 in. screws
- 4 - $1\frac{1}{4}$ in. screws
- 2 - $1\frac{1}{2}$ in. screws

LIMIT SWITCH FOR DISPOSING:

- I attached a limit switch on the back of the robot for disposing (high enough that every ball will press it when disposed, regardless of speed).
- I can program the pressing of this limit switch to initiate the conveyor stopping and visor lowering.

Materials used to mount this limit switch:

- 1 - Limit Switch
- 1 - 1 x 20 steel bar
- 2 - 1x6 steel bars
- 1 - 1 x 2 x 3 aluminum c-channel
- 7 - 0.375 in. screws
- 2 - $\frac{1}{2}$ in. screws
- 2 - nylon washers
- 7 - nylocks
- 2 - thin nylocks



Limit Switch for disposing

This c-channel (cut) prevents the limit switch's lever arm from being bent down

witnessed by:

date: 10/10-12/20

Megan M 10/12/20

10/13/20

PROGRAMMING A DRIVE/STRAFE TOGGLE ~

GOAL: Program driver control macros to help driving be more efficient and intuitive.

GOAL COMPLETED 10/18/20

Tasks I would like a macro for:

- ☐ Toggle between driving / strafing
- ☐ Bringing in balls
- ☐ Disposing balls
- ☐ Scoring with automatic dunking if needed
- ☐ Cycling balls at goals (intaking, scoring descoring)
- ☐ Pop-out for Driver Skills only

```
void strafeLeftAndRight() {
  /* joystickPositionLimit specifies the range of values returning from the
  Controller.Axis.position() function in which the wheels are signaled to brake */
  int joystickPositionLimit = 5;

  // ButtonLeft toggles between tank drive and strafing
  // Axis1 is the horizontal axis on the right joystick
  if(Controller.ButtonLeft.pressing() == true && Controller.Axis1.position()) {
    DriveMotorLeftFront.spin(directionType::fwd,
    Controller.Axis1.value(),velocityUnits::rpm);
    DriveMotorLeftBack.spin(directionType::rev,
    Controller.Axis1.value(),velocityUnits::rpm);
    DriveMotorRightFront.spin(directionType::rev,
    Controller.Axis1.value(),velocityUnits::rpm);
    DriveMotorRightBack.spin(directionType::fwd,
    Controller.Axis1.value(),velocityUnits::rpm);

    /* This specifies the range of joystick values to register as unchanged
    (a broader range prevents the robot from continuing to move if the joystick isn't
    exactly at 0)*/
    wait(10, msec);
    if(Controller.Axis1.position() <= joystickPositionLimit &&
    Controller.Axis1.position() >= -joystickPositionLimit) {
      DriveMotorLeftFront.stop(brakeType::brake);
      DriveMotorLeftBack.stop(brakeType::brake);
      DriveMotorRightFront.stop(brakeType::brake);
      DriveMotorRightBack.stop(brakeType::brake);
    }
    if(Controller.ButtonLeft.pressing() == false) {
      DriveMotorLeftFront.stop(brakeType::brake);
      DriveMotorLeftBack.stop(brakeType::brake);
      DriveMotorRightFront.stop(brakeType::brake);
      DriveMotorRightBack.stop(brakeType::brake);
    }
  }
}
```

- Here is the strafing function for my first macro, a toggle between strafing and driving forward/backward.

While I'm holding the left button on the controller, the horizontal axis on the right joystick controls strafing, and driving forward/backward is unable to execute.

10/13/20
M.M.

date: 10/13/20

Megan M 10/13/20

10/13/20

DRIVE / STRAFE TOGGLE CONTINUED

```
void driveForwardAndBackward() {
  /* joystickPositionLimit specifies the range of values returning from the
  Controller.Axis.position() function in which the wheels are signaled to brake */
  int joystickPositionLimit = 5;

  // ButtonLeft toggles between tank drive and strafing
  // Axis2 is the right joystick
  if(Controller.ButtonLeft.pressing() == false && Controller.Axis2.position()) {
    DriveMotorRightFront.spin(directionType::fwd,
    Controller.Axis2.value(),velocityUnits::rpm);
    DriveMotorRightBack.spin(directionType::fwd,
    Controller.Axis2.value(),velocityUnits::rpm);

    /* This specifies the range of joystick values to register as unchanged
    (a broader range prevents the robot from continuing to move if the joystick isn't
    exactly at 0)*/
    wait(10, msec);
    if(Controller.Axis2.position() <= joystickPositionLimit &&
    Controller.Axis2.position() >= -joystickPositionLimit) {
      DriveMotorRightFront.stop(brakeType::brake);
      DriveMotorRightBack.stop(brakeType::brake);
    }
    if(Controller.Axis3.position() <= joystickPositionLimit &&
    Controller.Axis3.position() >= -joystickPositionLimit) {
      DriveMotorLeftFront.stop(brakeType::brake);
      DriveMotorLeftBack.stop(brakeType::brake);
    }
  }

  // ButtonLeft toggles between tank drive and strafing
  // Axis3 is left joystick
  if(Controller.ButtonLeft.pressing() == false && Controller.Axis3.position()) {
    DriveMotorLeftFront.spin(directionType::fwd,
    Controller.Axis3.value(),velocityUnits::rpm);
    DriveMotorLeftBack.spin(directionType::fwd,
    Controller.Axis3.value(),velocityUnits::rpm);

    /* This specifies the range of joystick values to register as unchanged
    (a broader range prevents the robot from continuing to move if the joystick isn't
    exactly at 0)*/
    wait(10, msec);
    if(Controller.Axis3.position() <= joystickPositionLimit &&
    Controller.Axis3.position() >= -joystickPositionLimit) {
      DriveMotorLeftFront.stop(brakeType::brake);
      DriveMotorLeftBack.stop(brakeType::brake);
    }
    if(Controller.Axis2.position() <= joystickPositionLimit &&
    Controller.Axis2.position() >= -joystickPositionLimit) {
      DriveMotorRightFront.stop(brakeType::brake);
      DriveMotorRightBack.stop(brakeType::brake);
    }
  }
}
```

This checks to see if strafing is being used.

I control the right and left sides of the drivetrain separately. This gives me more manipulation options when driving.

- This code works well!
- Being able to toggle between driving and strafing will be a huge help in Driver Skills

10/13/20 M.M.

project | designed by: | witnessed by:

date: 10/13/20

Megan M 10/13/20

10/14/20

COLLECTING BALLS MACRO

- I plan to use the three distance sensors in the conveyor (see page 77-78) to evaluate the position of the balls prior to motor movement in my macro to bring in the balls.
- Using the data from the devices menu on the Brain, I compiled these thresholds for each distance sensor on the conveyor:

The low sensor can be used for goal alignment as well!

(in mm)	No Ball	Edge of Ball	Center of Ball	Aligned with Goal
Low Position 1	230-240	130-140	50	20-30
Middle Position 2	170-175	120-130	35-45	--
High Position 3	170-180	120	10-15	--

10/14/20 M.M.

These are the distance measurements returned by each sensor when the balls are in various positions;

- In order to have a fully functional collecting balls macro, I need to code specific actions for every possible ball position.

- There are eight possible arrangements for balls in the conveyor: (where the 0 or 1 stands for false / no ball and true / ball respectively, and the order is low, middle, high)
  - **000**: bring incoming ball to position 2
  - **100**: bring ball to position 3 and incoming ball to position 2
  - **010**: bring ball to position 3 and incoming ball to position 2
  - **001**: bring incoming ball to position 1; do not move the conveyor
  - **110**: bring balls to positions 3 and 2 and incoming ball to position 1
  - **011**: bring incoming ball to position 1; do not move conveyor
  - **101**: vibrate the controller once
  - **111**: vibrate the controller once

This alerts me to score or dispose before intaking more (I would accidentally eject balls otherwise)

low 1 middle 2 high 3

Megan M 10/15/20

 10/15/20

COLLECTING BALLS MACRO CONTINUED

- Here is the code that determines what arrangement the balls are in using the thresholds from the table on page 81:

This snippet of code shows how the ball positions are evaluated

```
// ButtonR2 initiates collecting balls
else if(Controller.ButtonR2.pressing() == true) {
  collectBallsArray[6] /*collectBallsActive */ = true;
  // Get the current data from the three distance sensors on the conveyor
  double lowConveyorSensorData = DistanceSensorConveyorLow.objectDistance(mm);
  double middleConveyorSensorData = DistanceSensorConveyorMiddle.objectDistance(mm);
  double highConveyorSensorData = DistanceSensorConveyorHigh.objectDistance(mm);

  // Determines whether there is a ball in each position based on thresholds in
  distance sensor readings
  if(lowConveyorSensorData < 220) {
    collectBallsArray[1] /* positionOne (low distance sensor) */ = true;
  }
  if(middleConveyorSensorData < 160) {
    collectBallsArray[2] /* positionTwo (middle distance sensor) */ = true;
  }
  if(highConveyorSensorData < 160) {
    collectBallsArray[3] /* positionThree (high distance sensor) */ = true;
  }
  // indicates that the positions of balls in the conveyor have been evaluated
  collectBallsArray[0] /* evaluatePosition */ = false;
}
```

10/14/20 M.M.

the "if" statement before this checks to see if the emergency button is pressed

- With macros dependent on a sensor value being returned, I wanted to have an emergency stop button in case something goes wrong and the value is never reached - I designated Button Down to serve this purpose.

- The code for each position (see page 81), along with this evaluation work by itself, but the drive motors in the functions from 10/13/20 cannot move at the same time this is executing - this is a big problem. I cannot operate the robot without being able to move multiple groups of motors at once.

When either of these functions is running, the program flow is stuck in a while loop which prevents anything else from executing.

The only way to fix this is to restructure the flow of my code

project

designed by:

witnessed by:

date: 10/15/20

Megan T. 10/15/20

10/16/20

RESTRUCTURING THE CODE

- After researching and running very simple tests, I have come to the conclusion that using different tasks or threads to multitask will not work for my application. → this is because I need the various functions to run separately, but then rejoin, and tasks and threads aren't made to do this

- So, I developed my own solution to the problem: a new style of programming which dynamically checks functions for execution and completion.
  ↳ this took a lot of trial and error and testing

What I learned: In order to allow several functions to execute at once, you must use if statements - you cannot use while loops inside the subroutines. And to avoid sending conflicting commands to motors, you must use flagging.

This code is in the usercontrol function which is called in the Main

This is how I call functions with dynamic programming

(... calls ...nction ...Left passes ...active ...to ...ted ...on ...the ...ers)

```
if(Controller.ButtonLeft.pressing() == false) {
  if(Controller.Axis3.position()) {
    driveLeftActive = true;
  }
  if(driveLeftActive == true) {
    driveLeft(driveLeftActive); // forward and reverse motion in the left wheels when
    left joystick is changed
  }
  if(Controller.Axis2.position()) {
    driveRightActive = true;
  }
  if (driveRightActive == true) {
    driveRight(driveRightActive); // forward and reverse motion in the right wheels
    when right joystick is changed
  }
}

else if(Controller.ButtonLeft.pressing() == true) {
  if(Controller.Axis1.position() && Controller.ButtonLeft.pressing() == true) {
    strafeActive = true;
  }
  if(strafeActive == true) {
    strafe(strafeActive); // sideways motion when ButtonLeft is pressed and right
    joystick is changed horizontally
  }
}
```

flagging

flagging

flagging

This is an extra "stop point" because occasionally the order of releasing the joystick/button would cause it not to stop

```
if(driveLeftActive == false && driveRightActive == false && strafeActive == false) {
  DriveMotorLeftFront.stop(brakeType::brake); // stop the wheels if neither of the
  joysticks are changed
  DriveMotorLeftBack.stop(brakeType::brake);
  DriveMotorRightFront.stop(brakeType::brake);
  DriveMotorRightBack.stop(brakeType::brake);
}
```

flagging

10/16/20 M.M.

How the code changed : pg. 79-80 ← sequential

pg. 83-84 ← dynamic

10/16/20

RESTRUCTURING THE CODE CONTINUED ~

similar function for driveLeft

```
int driveRight(bool driveRightActive) {
  /* joystickPositionLimit specifies the range of values returning from the
  Controller.Axis.position() function in which the wheels are signaled to brake */
  int joystickPositionLimit = 5;
```

Dynamic programming requires using a value returning function instead of a void function

```
  // ButtonLeft toggles between tank drive and strafing
  // Axis2 is the right joystick, controls the wheels on the right side
  DriveMotorRightFront.spin(directionType::fwd,
  Controller.Axis2.value(),velocityUnits::rpm);
  DriveMotorRightBack.spin(directionType::fwd,
  Controller.Axis2.value(),velocityUnits::rpm);
```

driveForwardAndBackward() must be split into driveRight and driveLeft (the sides of the drivetrain)

```
  /* This specifies the range of joystick values to register as unchanged
  (a broader range prevents the robot from continuing to move if the joystick isn't
  exactly at 0) */
  if((Controller.Axis2.position() <= joystickPositionLimit &&
  Controller.Axis2.position() >= -joystickPositionLimit)
  || Controller.ButtonLeft.pressing() == true) {
    DriveMotorRightFront.stop(brakeType::brake);
    DriveMotorRightBack.stop(brakeType::brake);
    driveRightActive = false; // flag to "unlock" other functions that call the drive
    motors
  }
  return driveRightActive; // returns the values to use in the next iteration
}
```

↳ resets the flag when no longer running

returns the flag

10/16/20 M.M.

```
int strafe(bool strafeActive) {
  /* joystickPositionLimit specifies the range of values returning from the
  Controller.Axis.position() function in which the wheels are signaled to brake */
  int joystickPositionLimit = 5;

  // ButtonLeft toggles between tank drive and strafing
  // Axis1 is the horizontal axis on the right joystick, controls strafing of all
  four wheels
  DriveMotorLeftFront.spin(directionType::fwd,
  Controller.Axis1.value(),velocityUnits::rpm);
  DriveMotorLeftBack.spin(directionType::rev,
  Controller.Axis1.value(),velocityUnits::rpm);
  DriveMotorRightFront.spin(directionType::rev,
  Controller.Axis1.value(),velocityUnits::rpm);
  DriveMotorRightBack.spin(directionType::fwd,
  Controller.Axis1.value(),velocityUnits::rpm);
```

No wait times in dynamic programming

```
  /* This specifies the range of joystick values to register as unchanged
  (a broader range prevents the robot from continuing to move if the joystick isn't
  exactly at 0)*/
  if((Controller.Axis1.position() <= joystickPositionLimit &&
  Controller.Axis1.position() >= -joystickPositionLimit)
  || Controller.ButtonLeft.pressing() == false) {
    strafeActive = false; // flag to "unlock" other functions that call the drive
    motors
  }
  Controller.ButtonLeft.released(driveMotorsStop); /* ensures that the drive motors
  stop as intended,
  no matter if the joystick or the button is released first */
  return strafeActive; // returns the values to use in the next iteration
}
```

↳ this allows other functions with drive motors to execute if called

returns the flag

10/16/20 M.M.

10/16/20

Megan M 10/16/20

10/17/20

PROGRAMMING OTHER MACROS and buttons to control manually

- I programmed a disposing macro similarly to the collectBalls macro (see pg. 82) but using dynamic programming and flagging as in the rewritten functions (see pg. 83-84). The limit switch for disposing (see pg. 78) works well.
- I programmed manual controls for the intakes and conveyor, and a one-time use pop-out for Driver Skills, all using dynamic programming.
- later I hope to add a scoring macro, but for now I'll use a manual dunking button when needed (see below).

These are found in usercontrol ()

Initialized variables and array to pass to the dunkManual function when called

```
int stepCompleted = 0;
bool dunkManualActive = false;

// Array with the needed parameters to manually dunk balls
int dunkManualArray[2] = {stepCompleted, dunkManualActive};


int dunkManual(int dunkManualArray[2]) {
  // Check that the emergency stop button isn't being pressed
  if(Controller.ButtonDown.pressing() == true) {
    VisorMotor.stop(hold);
    dunkManualArray[0] /* stepCompleted */ = 0;
    dunkManualArray[1] /* dunkManualActive */ = false;
  }

  if(Controller.ButtonL2.pressing() == true) {
    dunkManualArray[1] = true; // "locks" all other functions that use the visor motor
  }
  if(dunkManualArray[1] /* dunkManualActive */ == true) {
    if(LimitSwitchVisorHigh.pressing() == false && dunkManualArray[0] /*
    stepCompleted */ == 0) {
      VisorMotor.spin(directionType::fwd,100,velocityUnits::rpm);
    }
    if((InertialSensorHood.pitch() <= 0 || LimitSwitchVisorHigh.pressing() == true)
    && dunkManualArray[0] == 0 /* stepCompleted */) { // prevents visor from going
    too high
      VisorMotor.stop(hold);
      dunkManualArray[0] /* stepCompleted */ = 1;
    }
    if(dunkManualArray[0] /* stepCompleted */ == 1) {
      VisorMotor.spin(directionType::rev,60,velocityUnits::rpm); // once ball has
      been dunked
    }
    if(dunkManualArray[0] == 1 /* stepCompleted */ && LimitSwitchVisorLow.pressing()
    == true) {
      VisorMotor.stop(hold); // stop visor once lowered
      dunkManualArray[0] /* stepCompleted */ = 0;
      dunkManualArray[1] /* dunkManualActive */ = false; // "unlocks" all other
      functions that use the conveyor
    }
  }
  return dunkManualArray[2]; // returns the values to use in the next iteration
}
```

One button push will raise and then immediately lower the hood and visor again.

For dunking, the visor doesn't have to raise all the way up - just past horizontal

resets the variables

10/18/20 M.M.

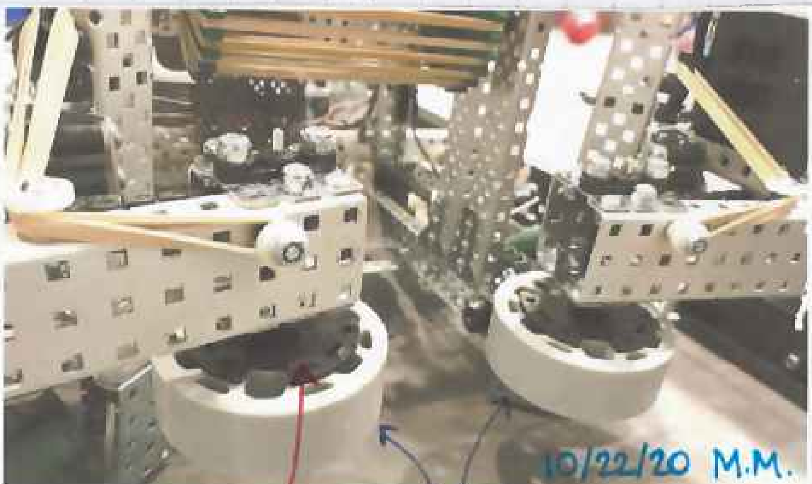
date: 10/17-18/20

Megan M 10/17/20

10/22/20

FLEX WHEELS & SKILLS PATH

GOAL: Experiment with the newly-legal-for-VRC Flex Wheels on the intakes and start planning a driver skills path. **GOAL COMPLETED 10/22/20**



1/2" Hex Versahub   Flex wheels

- The 3" OD, 30A Straight Flex Wheels are about the same size as the traction tread with rubberbands around an 18 T sprocket (see pg. 67).
- The Flex Wheels have better grip on the balls and can intake, descore, and outtake much faster than the traction treads.

- I plan to keep the Flex Wheels - they still fit within the size limit and do not affect my driver macros (see pages 79-85).

- I plan to follow this driver skills path for the Halloween competition.

10/22/20 M.M.

Red score − Blue score + 63

5 balls + (2 rows · 6 points each) = 17
15 balls + (0 rows · 6 points each) = 15

17 − 15 + 63 = $\boxed{\text{65 points}}$

- I replaced the standoffs on the end of the hood (see pg. 55) with a cut 11" ziptie to slide under the rubberbands to stow.
- These shouldn't wear out the rubberbands on the conveyor as much.

zipties for stowing

dunker

project   designed by: witnessed by:

* Field diagram from VEX Forum

date: 10/22/20

PROPRIETARY INFORMATION all information is the property of, and solely owned by the Designer.

Megan M 0/22/20

10/24/20

PROGRAMMING AUTONOMOUS SKILLS

GOAL: Program autonomous skills dynamically with lots of sensors.

GOAL COMPLETED 10/24/20

- Programming dynamically in autonomous is different than in usercontrol.

All within autonomous()

PSEUDOCODE to show the code flow of a dynamic autonomous

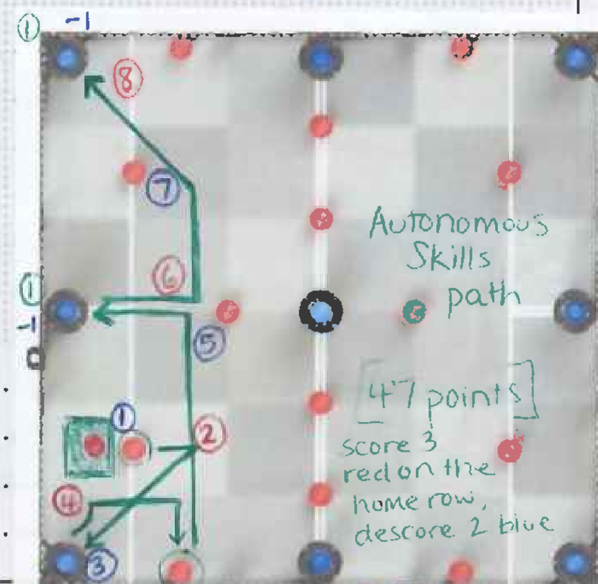
```
while(true) {
  // STEP #: description of robot's movements
  if(routineIndex == #) {
    assign any variables that need to be passed into the function rather than
    evaluated inside the function
    call functions and pass in needed variables in an array (if more than one value
    to send)

    if(all boolean completion flags signal true) {
      reassign the routineIndex to value of the next step
      reset variables passed into the function
    }
  }
}
```

Again, one while(true) here and all if statements in the functions

10/24/20
M.M.

- Instead of flagging function based on motor usage, I have an index system based on the value of the variable routineIndex.
- A group of functions (or even just one) needed to accomplish a movement are called together in a step (or index value).
- The program cannot leave that step until all of the functions within that step have completed.

1. Intake ball with pop-out.
2. Drive forward with back distance sensor and turn with inertial sensor.
3. Stop at goal with low distance sensor.
4. Pivot turn with inertial sensor; strafe to ball and intake with front distance sensor.
5. Drive with back distance sensor and turn with inertial sensor.
6. Drive backward with front distance sensor
7. Turn with inertial sensor; Drive forward with back distance sensor; Turn with inertial sensor
8. Stop at goal with low distance sensor.



10/24/20 M.M.

project: Scoring with inertial sensor, and disposing with limit switches

designed by:

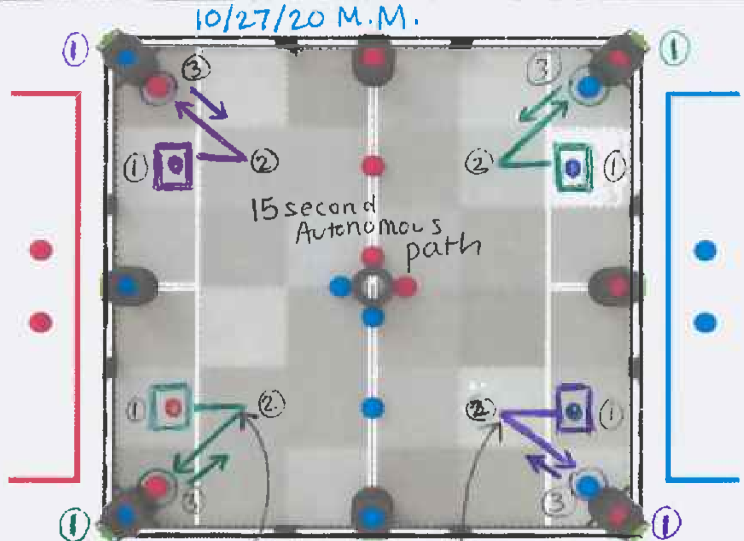
date: 10/24-25/20

Megan M 10/24/20

10/26/20

PROGRAMMING AUTONOMOUS (15 seconds)

10/27/20 M.M.



- Since I'm running short on time until the first competition, I decided to only program one 15-second autonomous routine that scores the preload in a corner goal. The field is symmetrical, which means I only have to change the direction of the angle for the mirrored side.

① Drive forward with back distance sensor
② Turn with inertial sensor
③ Stop at goal with low distance sensor

Right goal : angle = +118
Left goal : angle = −118
(scoring with inertial sensor)

[back up with front distance sensor]

SEQUENTIAL vs. DYNAMIC

SEQUENTIAL	DYNAMIC
• Each step is initiated programmatically	• Each step is initiated by joystick / button (in driver) or Boolean flags (in autonomous)
• Each step must wait for the previous step to complete with no overlap.	• Each step must wait for a flag or sensor reading to execute and may potentially overlap
• uses lots of loops	• uses one main loop that encompasses many if statements
• calls one function at a time	• all functions are available and "listen" for commands
• Code may be reused but often requires modification	• Functions may be reused easily without modification
• Inefficient because motors cannot execute at the same time based on different sensor values	• Very efficient because multiple actions can execute at once and flagging can limit the number of "listening" functions.

10/28/20

FINAL PREPARATIONS before the Halloween COMPETITION

10/28/20

- In programming autonomous and skills, I found that the polycarbonate ball guides don't pop out a significant portion of the time.
  - ↳ When this happens, I cannot get close enough to the goal to score, so I removed them. I plan to look into reattaching them in the future.

10/29/20

- I completed my submission for the VRC Annotated Programming Skills online challenge.

POLYCARBONATE MAP :

- I cut pieces of paper approximately the size of each section of polycarbonate to make sure I am within the single sheet limit (12" by 24")
- This is an identical sheet to the one I've used on my robot

POLYCARBONATE CUTTING TIP:

I've tried saws, exacto knives, you name it. But I found that **scissors work the best**, at least with 0.062" thick.

10/29/20 M.M.

Megan M 10/29/20

 10/29/20

10703Z for the BRENTWOOD HALLOWEEN Competition

Controls left side moving forward/reverse

Message prints once inertial sensors have calibrated

Controls right side moving forward/reverse

Toggle for strafing

Emergency stop button for macros

Manual outtaking from intakes

Buttons on front (left side)
top: manual conveyor moving upward
bottom: manual dunking

Controller Button layout

Controls strafing when toggled

disposing macro

Manual conveyor moving downward

Buttons on front (right side)
top: manual intaking
bottom: collecting balls macro

SUCCESSES:

- Implementing a number of sensors
- Ability to score and descore all goals
- Ability to dispose unwanted balls
- Collecting and disposing macros
- Developing dynamic programming

THINGS TO IMPROVE:

- Programming a wider variety of autonomous programs to work well with any alliance partner
- Increasing speed for more efficiency
- Program more macros for scoring and automatic dunking
- Driving proficiency

MY GOALS FOR THE BRENTWOOD HALLOWEEN COMPETITION:

- ☑ Get a second interview.
- ☑ Score at least 65 points in Driver Skills
- ☒ Score 47 points in Autonomous Skills
- ☑ Rank in the top half after Qualifications

Completed 10/31/20

project

designed by:

witnessed by:

date: 10/29/20

Megan N 10/29/20

10/31/20

BRENTWOOD HALLOWEEN COMPETITION ANALYSIS

10703Z Stats:

- Ranked 7th after qualifications
- Ranked 9th in skills with 87 points total
- 5 wins, 2 losses, 1 tie in qualifications

- WP = 13
- AP = 30
- SP = 91
- OPR = 15.5
- DPR = 9.4
- CCWM = 6.1

- My 15-second autonomous programs (score preload in corner goal, slight variation for left and right sides) worked perfectly every single time except for my first match, where it did not run, and the Ethernet cable was replaced after the match.

- I won the Design Award, which qualifies me for the TN State Championship in March!

10/31/20 M.M.

Brentwood Skills Results

Driver / Auton	Score	Analysis
Driver run #1	65	Scored 5 red balls, one in each goal on the path drawn on pg. 86; had a few seconds remaining and almost scored one more goal.
Driver run #2	67	Scored 5 red balls as in Driver run #1, but also descored 2 blue balls from the final goals.
Driver run #3	$\boxed{68}$	Scored 6 red balls as in Driver runs #1 and #2, but scored 2 in fifth goal; descored 2 blue balls from the final goals
Auton run #1	0	The rubberbands on the hood weren't broken in, so the first ball got stuck and the program could not continue.
Auton run #2	$\boxed{19}$	The robot scored the preload in the corner goal, but the criteria from the front distance sensor was not met, so the robot kept going
Auton run #3	19	Same as Auton run #2, the robot did not stop when it reached the ball on the field perimeter. (drawn on pg. 87)

project designed by: witnessed by:

date: 10/30-31/20

Megan M 10/31/20

COMPETITION ANALYSIS CONTINUED

10/31/20

10/31/20 M.M.

Brentwood Qualification Match Results

Alliance Partner	Opponents	Win / Loss	Score	Analysis
9364V	663E 61391Z	Win	18-15	Ethernet cable did not work (was replaced after the match) → auton. did not run. very little strategy
663A	73972B 9364U	Win	69-8	Won autonomous ; good strategy, owned all goals good defense
99060K	663C 73973B	Win	27-10	Won autonomous good strategy good defense of center goal
9364Y	73972A 57249A	Win	64-6	won autonomous good strategy, owned all goals need to work on picking up balls
9364T	663X 9364D	Loss	15-18	won autonomous very little strategy need to work on center goal defense
9364B	663B 663G	Tie	15-15	lost autonomous decent strategy only owned one row
73973A	9364X 9364F	Win	21-14	won autonomous very little strategy and communication good scoring center goal
9364E	2775J 9364C	Loss	8-48	lost autonomous very little strategy no center goal scoring on my alliance

Brentwood Elimination Match Results 10/31/20 M.M.

Alliance Partner: 663B

Opponents	Win / Loss	Score	Analysis
57249A 663E	Win	24-13	R16 #3-1 lost autonomous good strategy, but the chain on my left intake broke
73973B 97934U	Loss	6-14	QF #2-1 lost autonomous chain on my left intake broke again and alliance's drive stopped working

date: 10/31/20

Megan M 10/31/20

10/31/20

BRENTWOOD COMPETITION ANALYSIS CONTINUED

INSPIRATIONS FROM OTHER TEAMS:

- **2775J:** self-actuating intakes (a ratchet allows the intakes to open)
  rubberbands stretched across their metal conveyor backing so the ball is coming up smoothly and pushed into the conveyor
  autonomous that scores ball in two goals and pushes a ball into the center goal
- **663A:** small, light robot with very fast drive motors and cycling, but no way to dispose (this didn't matter as much)
- A number of teams including **9364A** and **9364B** had long intakes; many were very floppy, but effective.
- Several teams including **9364Y** and **73973B** had an autonomous that scored balls in two goals, completing the home row when their alliance could score one

WHAT I LEARNED:

- Having to back up from a goal in order for the balls to fall down is problematic - it's difficult to tell when I've descored a goal on the far side of the field.
- There's not as much use for hoarding the opponent's balls as I had originally thought (see pages 4 and 6), but disposing is definitely useful, both in a match and in skills.
- Other robots weren't able to cycle the center goal, but pushing them out was faster anyway most of the time.
- Many of the hood bots had to have a lot of momentum in order to score (they couldn't bring a ball to the top of their conveyor unless they were about to score)
- There was less defense than I suspected - many teams could simply go and descore several goals in the time it would take to prevent another team from scoring or descoring.
- The autonomous bonus and owned rows are equally important. A connected row is worth the same six points as the autonomous bonus, and if opposing alliances have equal rows / autonomous bonus, it comes down to number of balls scored (see qualification match where I tied).

Megan M[redacted] 10/31/20

11/1/20

*Citations in this color

DESIGN CHANGES AFTER BRENTWOOD COMPETITION

PROBLEMS TO ADDRESS / IMPROVEMENTS NEEDED:

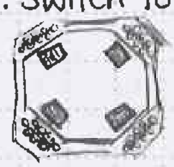
1. Drivetrain needs more speed. travel time is a big issue
2. More efficient and reliable disposing. The visor got stuck occasionally
3. Higher ball capacity. Three is ideal, all fully in the conveyor
4. Greater functionality and stability for the intakes. preferably with gears

BRAINSTORMING / RESEARCH:

Possible solutions to each problem:

① Faster Drivetrain

1. Lose some of the weight of the robot.
   - X Less stable
   - X Wouldn't gain much speed
2. Gear the mecanum wheels to a higher speed.
   - X Not enough space to add gears to back wheels
3. Change the drive motor cartridges to 600rpm (blue)
   - X Too fast to control
   - X High potential of burning out the motors
4. Switch to X-Drive
   - X Requires complete rebuild
   - X Little pushing power.



② Improved Disposing

1. Modify the visor
   - X no guarantee the problems are able to be fixed
2. Switch to a rubberband roller
   - X requires complete rebuild

3. Switch to a different type of trapdoor (like 48952E)
   - X requires complete rebuild
   - X many potential problems

Research Notes:

- Most hoodbots with a 3+ ball capacity have the ball follow a C-shaped path defined by their rollers (95E | 94999E Yokai VEX Change Up Early Season Reveal on Youtube)
- The only method of disposing that I saw at the Brentwood competition was using rubberband rollers. 48952E has a more unique "lever" to dispose, which seems to work well. (VRC Annotated Programming Skill Challenge 2020 | Team 48952E (Best Ghosts) on Youtube)

- The most functional style of intakes seems to be like 9185A has - long, but spaced rollers so as to be able to descore the center goal.

(9185A Early Season Vex Change Up Reveal (JarJar) on Youtube)

Megan M[redacted] 11/6/20

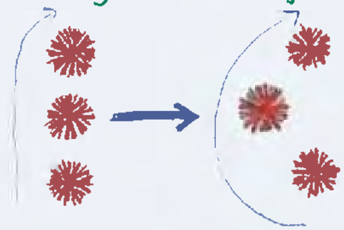
11/6/20

DESIGN CHANGES AFTER COMPETITION CONTINUED

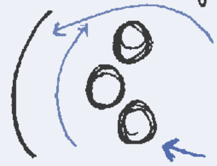
BRAINSTORMING CONTINUED: Possible solutions to each problem:

③ Higher ball capacity while improving speed / maintaining efficiency.

1. Reposition the current rubberband rollers to form a C-shaped path.
   - X Rubberband rollers require a lot of compression to bring balls up (cannot be high speed due to motor burnout)
   - X Requires complete rebuild



2. Use the space used for disposing to make a longer path (thus be able to hold more balls)
   - X Lose ability to dispose
   - X Requires complete rebuild



allows for a more compact backing

more room to form a C-shape

3. Switch to flex wheels on the conveyor in a C-shape.
   - X Potentially lower ball capacity than other options
   - X Requires complete rebuild

allows for speed without tight compression → 30a flex wheels have excellent traction (see pg. 86)

④ Improved intakes

1. Same design but with gears instead of chain (chain broke twice at tournament, see pg. 92)
   - X Does not fix collecting / descoring issues
   - X Not many options for gear configurations
2. Long intake with rollers spaced to descore the center goal (like 9185A).
   - X potential stowing issue
   - X requires complete rebuild

pushes ball out of center goal

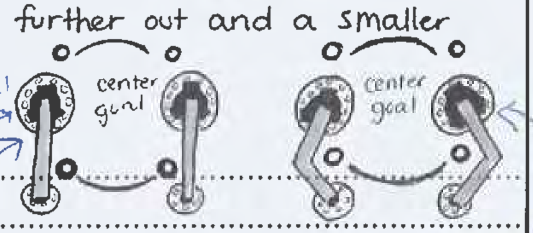
would still use flex wheels

fits around center goal

with either gears or chain one way or both?

Would this allow the robot to be all the way up to the center goal?

3. Straight or angled intake with one roller further out and a smaller one further back.
   - X potential issue stowing and staying clear of the wheel
   - X potential interference with other parts of the design
   - X requires complete rebuild



same concept, allows robot to be all the way up to the center goal, the shape will be determined by gears/chain

project | designed by: | witnessed by:

date: 11/1-6/20

Megan N 11/6/20

 11/6/20

DESIGN CHANGES AFTER COMPETITION CONTINUED

> SELECT OPTION :

Problem #1

Options to Increase Drivetrain Speed

Criteria (least to most important)	Scale	1 Remove weight where possible	2 Gear the mecanum drive	3 600rpm on mecanum drive	4 Rebuild with X-Drive
Retains ability to withstand defense	0 to 3	3	3	2	1
Low risk of motor burnout	0 to 5	5	5	0	5
Viable for the space available	0 to 7	7	0	7	6
Higher drivetrain speed	0 to 7	1	7	7	7
Total Score:		16	15	16	19

Even though switching to X-Drive requires a complete rebuild, the additional speed will be worth it, both in skills and in matches.

Problem #2

Options to Improve Disposing

Criteria (least to most important)	Scale	1 Modify the visor	2 Rubberband roller	3 Different type of trapdoor
Requires small amount of space	0 to 3	3	2	2
Few potential problems	0 to 5	0	5	0
Efficiency	0 to 7	2	7	3
Total Score:		5	14	5

Many teams successfully use a rubberband roller to dispose. It's simple, can also be used with the conveyor, and is highly efficient.

project designed by: witnessed by:

date: 11/1-6/20

Megan M 11/6/20

VEX "Never doubt that a small group of thoughtful, committed citizens can change the world. Indeed, it is the only thing that ever has." - Margaret Mead.

11/6/20

DESIGN CHANGES AFTER COMPETITION CONTINUED

> SELECT OPTION CONTINUED:

Problem #3

Options for Higher Ball Capacity

Criteria (least to most important)	Scale	(1) Rubberband rollers in a C-shape	(2) Larger C-shape without ability to dispose	(3) Flex Wheels in a C-shape
Low risk of motor burnout	0 to 5	3	5	5
Minimum ball capacity of 3	0 to 5	4	5	3
Retains ability to dispose	0 to 7	7	0	7
Efficiency and Speed	0 to 7	3	6	7
Total Score:		17	16	22

The flex wheels worked very well as intakes before - good traction.

Problem #4

Options to Improve the Intakes

Criteria (least to most important)	Scale	(1) Keep current design but with gears	(2) Long intake to push ball out of center goal	(3) Rollers spaced to descore center goal
Viable in the space available	0 to 5	2	3	3
Few potential problems	0 to 5	0	3	3
Ability to descore all goals	0 to 7	4	7	7
Efficiency	0 to 7	3	5	7
Total Score:		9	18	20

Being able to cycle the center goal could come in very handy.

project

designed by:

witnessed by:

date: 11/1-6/20

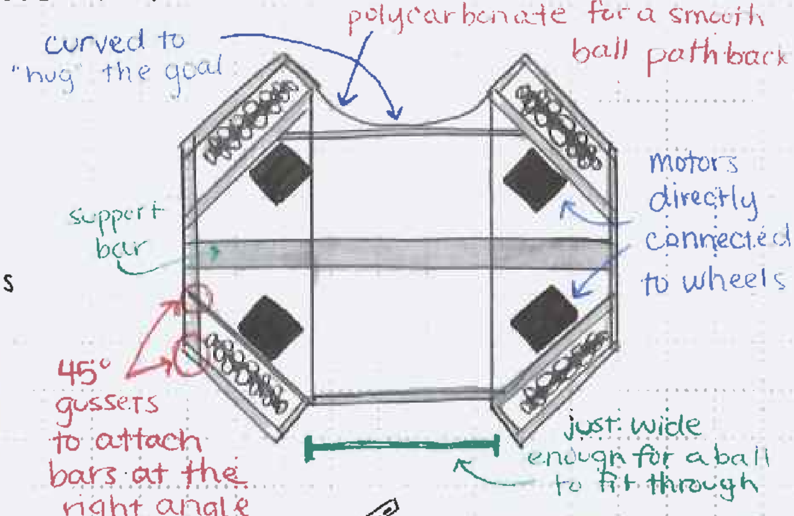
Megan M 11/6/20

 11/6/20

DESIGN PLAN : Build #2

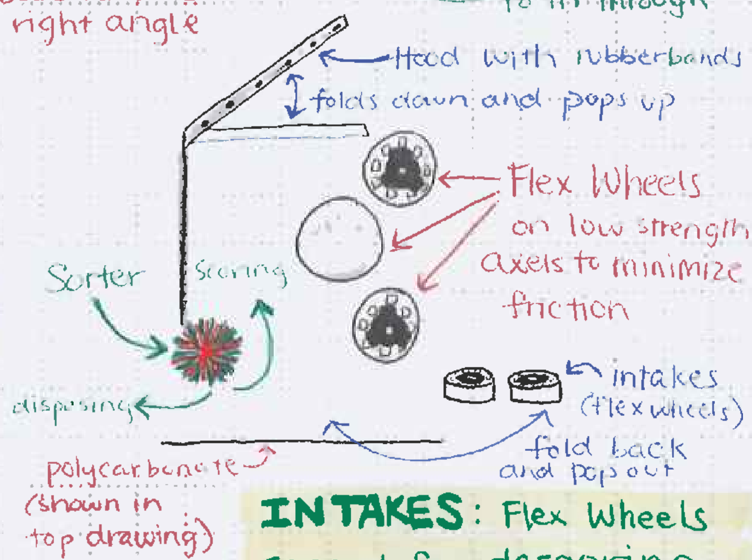
- My second design will be faster and more efficient, with fewer issues than I had with the first robot. Based on the problems I observed at the Brentwood competition (see pg. 91-93), and the best solutions to these problems (identified through decision matrices, pg. 96-97), this design should be a big improvement.

DRIVETRAIN: X-Drive
- 4 motors, 200 rpm cartridges (X-configuration makes this higher speed)
- 4" OD omni-directional wheels



CONVEYOR: Flex Wheels defining a C-shaped path
- 1 motor, 600 rpm cartridge
- chain / sprockets to connect the different rollers to one motor
- 3" OD and 4" OD 30a flex wheels

SORTER: Rubberband roller for disposing or sending balls to be scored
- 1 motor, 600 rpm cartridge
- 24T sprockets, rubberbands every other tooth



HOOD: Aluminum Frame with rubberbands across
- Stationary once popped up (held in place by additional rubberbands)
- the ball will only contact rubberbands

INTAKES: Flex Wheels spaced for descoring
- 2 motors, 200 rpm cartridges
- gears or chain / sprocket to connect the flex wheels
- folds backward to fit size limit (rubberbands to pop out)

project | designed by: | witnessed by:

date: 11/1-6/20

Megan P 11/6/20

11/7/20

TIMELINE : UNTIL SECOND COMPETITION

I don't have much time, considering that I'm completely rebuilding the robot, but I think I can push and get it done (although probably not in time for as many sensors and autonomous programming as I would like).

~M.M. 11/7/20

11/7/20
M.M.

BUILDING

Structure/Mobility
Base, X-Drive,
Structure
11/8 - 11/10

November

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

today

Gameplay Mechanisms
Conveyor, Hood,
Disposing, Intakes,
Improvements
11/11 - 12/5

November

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

December

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

PROGRAMMING

Programming and Practice
Sensors, Macros,
Autonomous, Skills
12/5 - 12/11

December

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Calendar Key :

- robot work day
- possible robot work day if I can get my school work done
- robotics competition in Bluff City

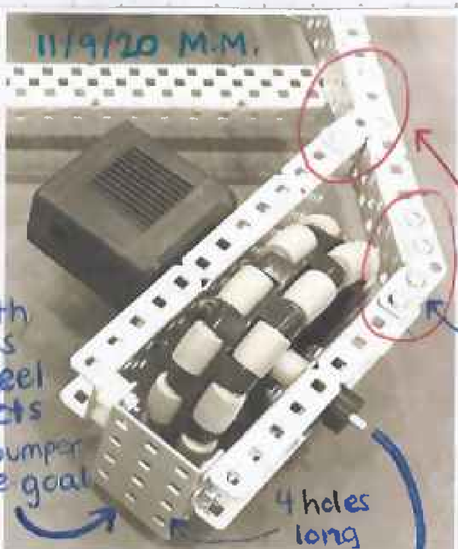
Megan M 11/7/20

11/8/20

BUILDING THE BASE AND X-DRIVE

GOAL: Build the X-Drive and base structure so as to fit within the size limit and leave enough space for the balls to enter the conveyor.

GOAL COMPLETED 11/10/20



This both guards the wheel and acts as a bumper on the goal

- X-Drive is tricky to build - you have to make sure all the angles are just right in order to minimize friction and motor overheating.

45-Degree gussets

For each connection point, I used two of the gussets, one on top and one on the bottom

- The base is 17 in. by 17in., with three support bars across.

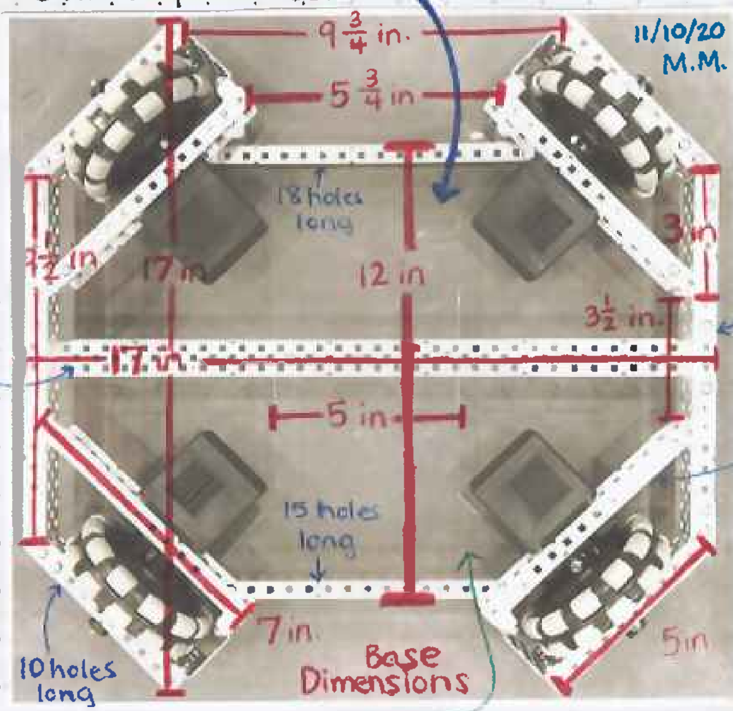
Order on the axel:

V5 Smart Motor (200rpm)
post hex nut retainer w/ flat bearing
c-channel
nylon washer
clamping shaft collar
1/8 inch spacer
omni-wheel
3/8 inch spacer
c-channel
flat bearing
nylon washer
clamping shaft collar

This is the same for all four wheels

- This polycarbonate gives the ball a smooth path back.

34 holes long



18 holes long

13 holes long

There will be anti-slip mat here for traction

project

designed by:

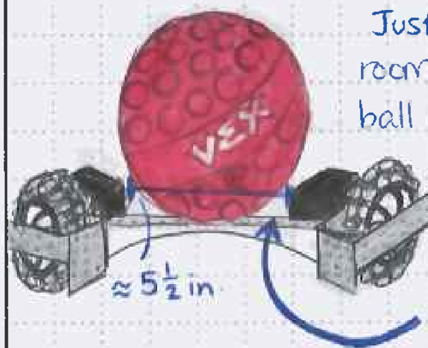
witnessed by:

date: 11/8-10/20

Megan M 11/8/20

11/10/20

BUILDING THE BASE AND X-DRIVE CONTINUED



Just enough room for the ball to pass without hitting the drive motors

Materials used on the base and X-Drive:

- 4 - omni-directional wheels
- 16 - 45-Degree gussets
- 1 - ≈ 13 in. by $7\frac{1}{2}$ in. polycarbonate
- 4 - shafts (cut)
- 1 - section of anti-slip mat
- 8 - 4" zipties
- 16 - nylon washers
- 6 - $\frac{1}{8}$ inch spacers
- 12 - $\frac{1}{4}$ inch spacers
- 8 - $\frac{3}{8}$ inch spacers
- 96 - thin nylocks
- 4 - V5 Smart motors (200rpm)
- 1 - 1×2×1×34 aluminum c-channel
- 4 - 1×2×1×10 aluminum c-channels
- 4 - 1×2×1×13 aluminum c-channels
- 2 - 1×2×1×18 aluminum c-channels
- 4 - 1×3×1×4 aluminum c-channels
- 1 - 1×1×18 aluminum angle
- 1 - 1×1×15 aluminum angle
- 4 - flat bearings
- 4 - post hex nut retainers w/ flat bearing
- 8 - clamping shaft collars
- 84 - 0.375 inch screws
- 8 - $\frac{1}{2}$ inch screws
- 8 - 0.625 inch screws
- 4 - 1 inch screws

TESTING:

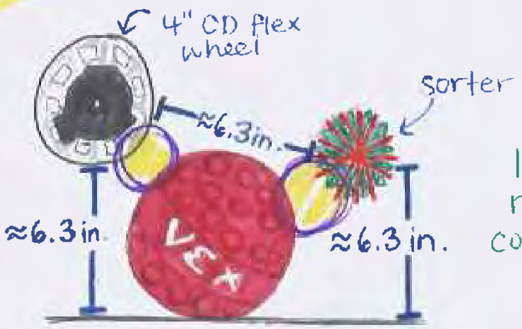
- The new drivetrain seems much faster than the mecanum drive was (see pg. 38-39) but also much easier to be pushed.
- The bumpers (see pg. 100) line up with the goals perfectly!

Megan M 11/10/20

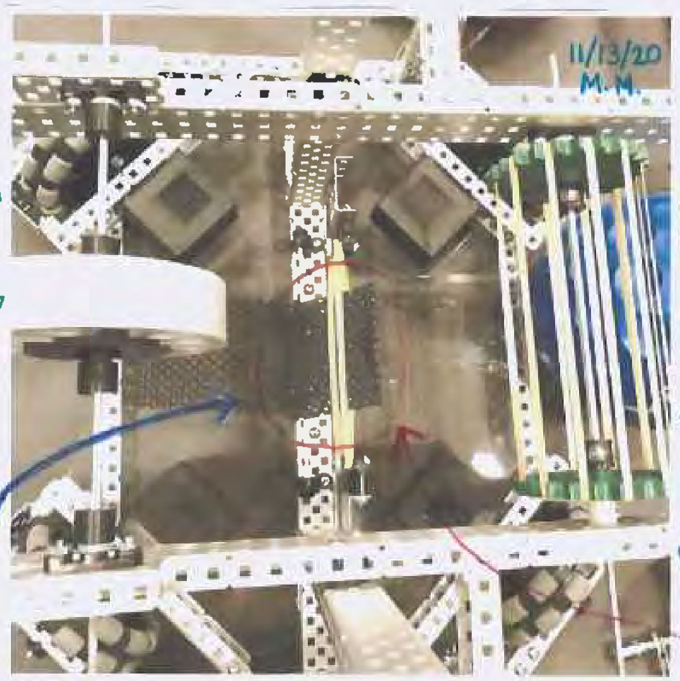
11/13/20

PROTOTYPING THE CONVEYOR & SORTER

GOAL: Find a configuration of rollers that allows for scoring, disposing, and that has a 3+ ball capacity; all while being within the size limit. GOAL COMPLETED 11/15/20



lowest roller on conveyor



I found that in order for the ball to make it back to the sorter, I need some sort of mechanism to raise it to the right level.

First prototype - with a small ramp to raise the ball to the right level

These locations for the lowest roller on the conveyor and the sorter are good

Problems with the ramp:

- Difficult to mount stably
- Must raise / pivot for the ball to be disposed (could interfere with the next ball)
- Must be curved or shapely exactly right, or could put pressure on the flex wheel / axel; potential to overheat the conveyor motor

- Other options instead of the ramp:
  - reattach the polycarbonate to have a bubble in the center (not ideal)
  - attach a permanent metal piece that doesn't move (problems disposing)
  - attach a rubberband across

Sprockets are an easy way to have ~~a rubberband~~ across the conveyor.

This works well and allows the ball to be scored/disposed with the sorter

project

designed by:

witnessed by:

date: 11/13-15/20

Megan M 11/13/20

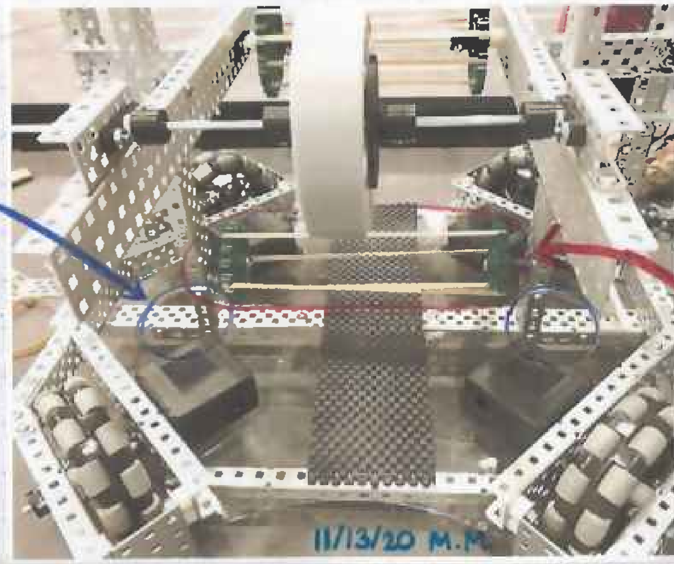
11/14/20

PROTOTYPING THE CONVEYOR & SORTER CONTINUED

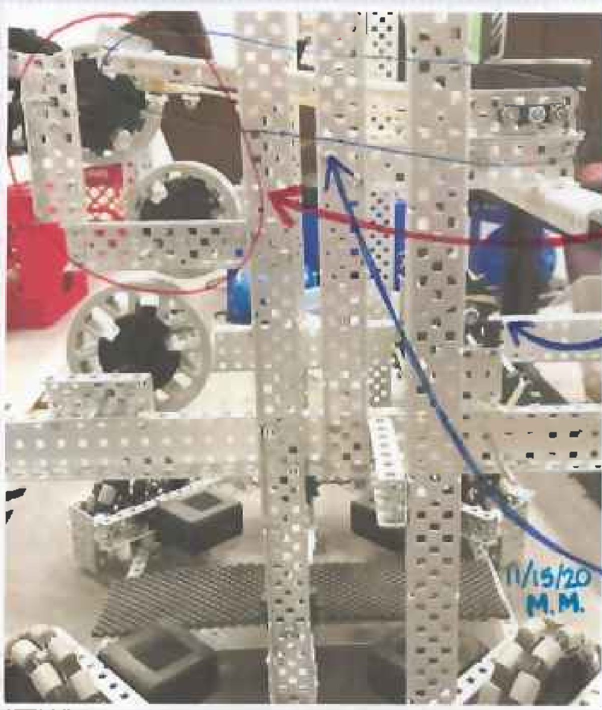
- These 90-Degree gussets seem to work well for attaching the vertical supports for the conveyor.

- The stationary rubberband-holding sprockets work so well that I added another set above the disposing roller to reduce the distance that the sorter contacts the ball.

This limits the interference from other balls in the robot when disposing



Prototype with stationary sprockets for rubberbands across



a second stationary set of 12T sprockets for rubberbands to be stretched across

- Adding two 3" OD flex wheels above the 4" flex wheel comes up to 17"-17.5" and has room to hold 3 balls in the conveyor, plus one, half in the intakes.

- The prototyped hood (frame with rubberbands) works well, but needs to be shorter in order to be able to fold down and still be within the size limit.

Final prototype with 3 rollers on the conveyor, rubberband roller as the sorter, and hood

- Overall, the conveyor/sorter is much faster and works much better than the previous design.

Megan M[redacted]/15/20

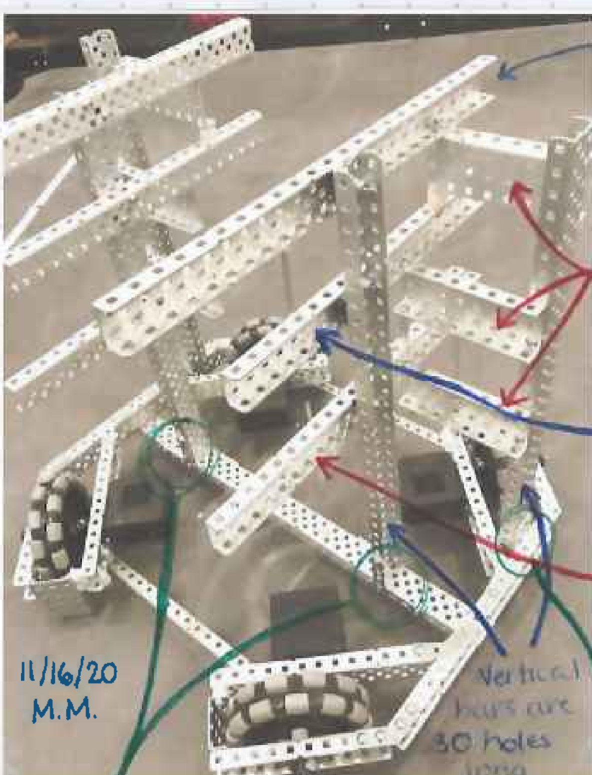
11/16/20

FINALIZING THE CONVEYOR STRUCTURE

GOAL: Based on the placement of the rollers found through prototyping (see pg. 102-103), build a stable structure for the conveyor and sorter with support bars where needed for mounting the rollers.

GOAL COMPLETED 11/16/20

- Today I built the conveyor with three horizontal bars (one for each flex wheel).
- There are two vertical bars on each side with three small horizontal supports connecting them on each side.



bars for top roller are 24 holes long

The supports are 8 holes long

bars for middle roller are 20 holes long

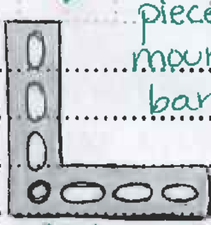
bars for bottom roller are 23 holes long

11/16/20
M.M.

Materials used in the conveyor structure

- 4 - 1x2x1x30 aluminum c-channels
- 2 - 1x2x1x24 aluminum c-channels
- 2 - 1x2x1x23 aluminum c-channels
- 2 - 1x2x1x20 aluminum c-channels
- 6 - 1x3x1x8 aluminum c-channels
- 2 - 1x2x1x1 aluminum c-channels
- 2 - 1x2x2 aluminum angles
- 4 - 90-Degree gussets
- 4 - 3" standoffs
- 2 - 2" standoffs
- 2 - nylon washers
- 2 - $\frac{1}{4}$ inch spacers
- 2 - $\frac{3}{8}$ inch spacers
- 14 - $\frac{1}{2}$ inch spacers
- 6 - $\frac{1}{2}$ inch locking screws
- 42 - 0.375 inch. screws
- 16 - 0.875 inch screws
- 50 - thin nylocks

I used these pieces to securely mount the vertical bars to the base



90-Degree gusset

1x2x2 aluminum angle

- I used standoffs to directly connect the back end of the lowest horizontal bar to the base.

11/16/20
M.M.

connected in hole 3

project

designed by:

witnessed by:

date: 11/16-20/20

Megan M 11/16/20

11/17/20

BUILDING THE CONVEYOR & SORTER

GOAL: Attach the flex wheels and motor on the conveyor and add the sprockets / rubberbands and motor for the sorter with as little friction as possible.

GOAL COMPLETED 11/20/20

Sorter

24T sprockets with rubberbands every other tooth

→ Materials used in the sorter ←

- 1 - V5 smart motor (600rpm)
- 2 - 1x3x1x6 aluminum c-channels
- 2 - 24T sprockets
- 2 - 90-Degree gussets
- 1 - shaft (cut)
- 1 - flat bearing
- 3 - clamping shaft collars
- 3 - nylon washers
- 11 - 0.375 inch screws
- 2 - $\frac{1}{2}$ inch screws
- 2 - $\frac{1}{4}$ inch spacers
- 2 - $\frac{1}{2}$ inch spacers
- 11 - thin nylocks
- 6 - rubberbands

- Only having rubberbands on every other tooth makes it easier to compress and dispose the ball.
- I added two of the sets of stationary sprockets with rubberbands (see pg. 102-103), one defining position one, the second for position two.

These sprockets are stationary. The rubberband bridges the gap between the conveyor and sorter (see pg. 102-103)

→ Materials used in stationary sprockets (two sets): ←

- 4 - 1x9 steel bars
- 4 - 12T sprockets
- 4 - rubberbands
- 2 - $\frac{1}{4}$ in. spacers
- 8 - $\frac{1}{2}$ in. spacers
- 4 - 0.875 in. screws
- 2 - $1\frac{1}{2}$ in. screws
- 2 - $1\frac{3}{4}$ in. screws
- 4 - nylocks
- 4 - thin nylocks

- To attach the second set of stationary sprockets I used 9 holes long steel bars on the low and middle horizontal bars.

(see picture on next page)

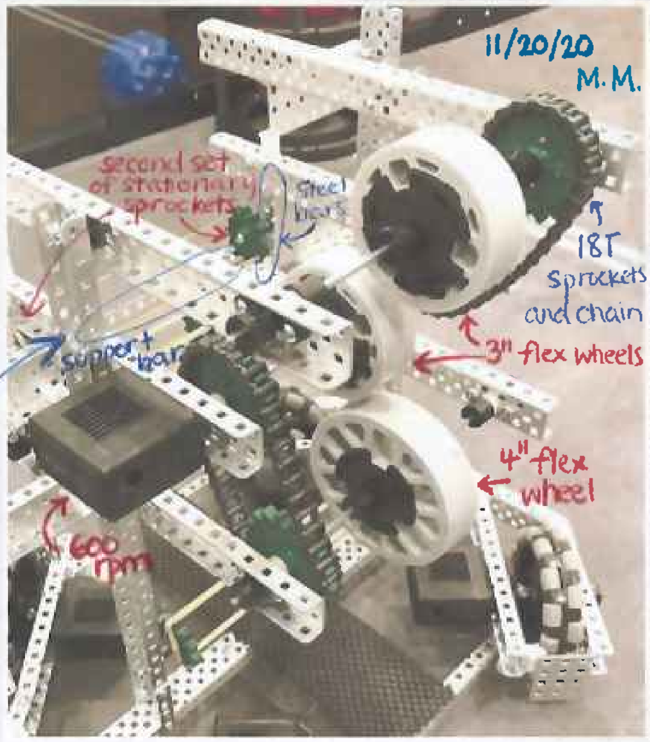
project	designed by:	witnessed by:

date: 11/16-20/20

Megan N 11/17/20

11/20/20

BUILDING THE CONVEYOR CONTINUED



- I mounted the flex wheels in the same positions as in the prototype (see pg. 103).

→ Materials used on the Conveyor: ←

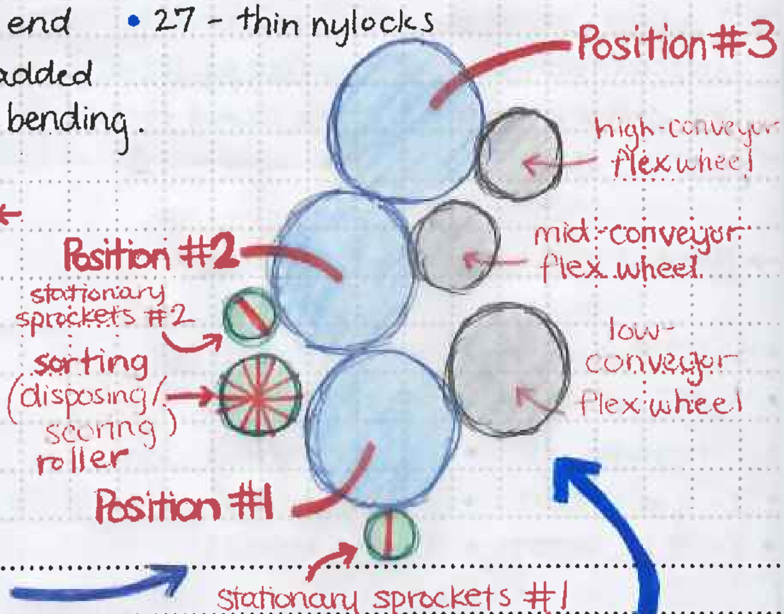
- 1-V5 smart motor (600rpm)
- 1 - 4" OD 30a flex wheel
- 2 - 3" OD 30a flex wheels
- ~~4~~ 3 - 18T sprockets
- 1-12T sprocket 11/22/20
- 3 - shafts (cut)
- 5 - flat bearings
- 17 - clamping shaft collars
- 10 - nylon washers
- 6 - $\frac{1}{4}$ inch spacers
- 19 - 0.375 inch screws
- 10 - $\frac{1}{2}$ inch screws
- 27 - thin nylocks
- 2 - sections of chain

- The second set of stationary sprockets puts compression on the back end of the horizontal bars, so I added a support bar to prevent any bending.

→ Materials used in support bar: ←

- 1 - 1x1x18 aluminum angle
- 2 - 0.375 inch screws
- 2 - thin nylocks

- In addition to these three positions, there should be room to hold a ball between the intakes and lowest conveyor flex wheel.



These are the three positions for the balls with this design.

project
designed by: see pg. 78 and 81 (for 1st design positions)
witnessed by:
date: 11/16-20/20

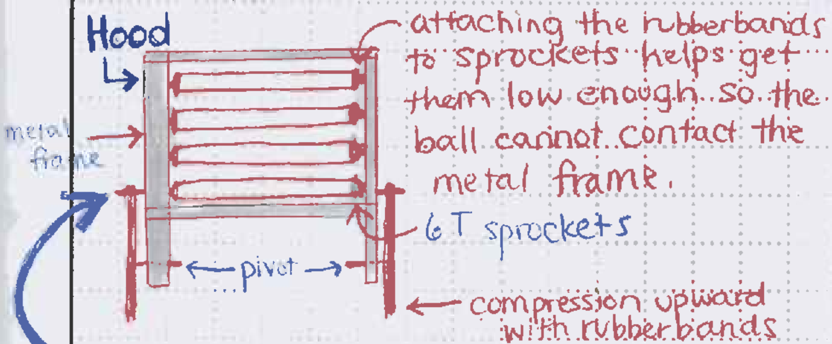
Megan M 11/20/20

11/21/20

BUILDING THE HOOD and finishing the conveyor backing

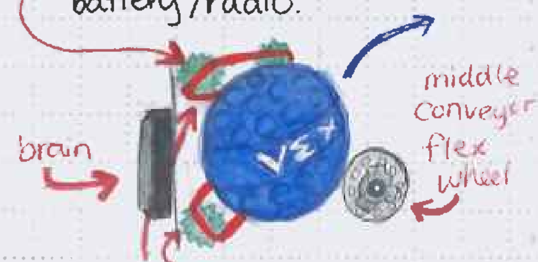
GOAL: Build and mount a hood in a way that can fold down to fit the size limit but is long enough to guide the ball into the goal. Finish the conveyor backing so the ball travels all the way through the robot.

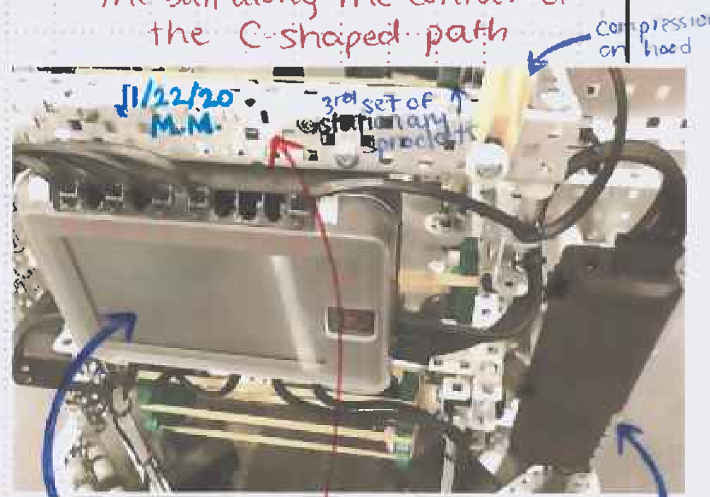
GOAL COMPLETED 11/22/20



- Hood requirements:
  - → must be framed with metal on all four sides so the compression from the rubberbands won't cause it to bend
  - → must be wide enough so the ball only contacts the rubberbands
  - → must be short enough to fold down and be within the size limit, but long enough for the ball to make it in the goal
  - → must be specifically shaped so the ball doesn't catch on it too early
  - → The pivot must be attached far enough back so a ball can fit through while constantly maintaining contact with the flex wheels.
  - → must be parallel to top two flex wheels when popped up

- Once I mounted the hood, I was able to mount the third set of stationary sprockets and the brain / battery/radio.





project | designed by: | witnessed by:

date: 11/21-22/20

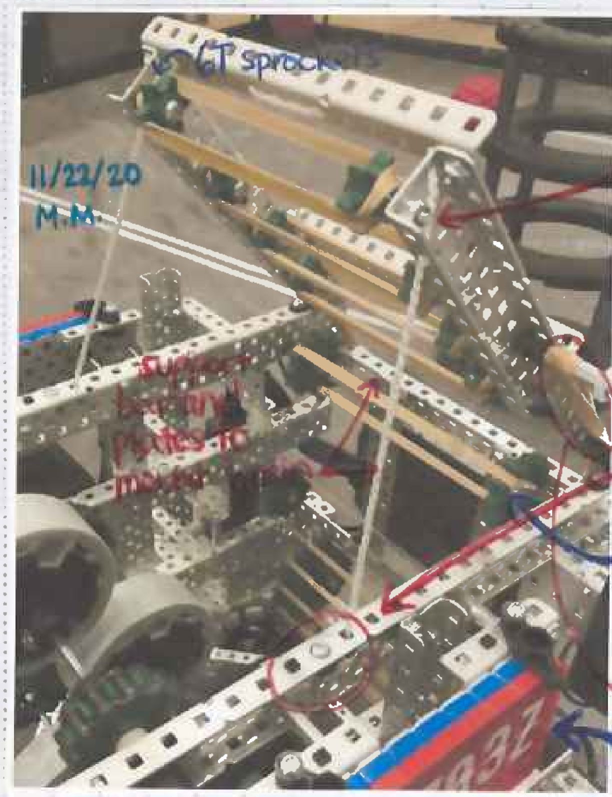
Megan 11/21/20

11/22/20

BUILDING THE HOOD CONTINUED

Materials used to mount the brain / battery / radio:

- 1 - V5 robot brain
- 1 - V5 robot radio
- 2 - V5 battery clips
- 1 - 1x2x1x18 aluminum c-channel
- 2 - 2x11 aluminum plates
- 2 - $\frac{3}{4}$ inch standoffs
- 4 - $\frac{1}{4}$ inch locking screws
- 4 - 0.375 inch screws
- 6 - 0.875 inch screws
- 4 - $1\frac{1}{4}$ inch screws
- 2 - $1\frac{1}{2}$ inch screws
- 2 - $\frac{1}{4}$ inch spacers
- 10 - $\frac{1}{2}$ inch spacers
- 4 - $\frac{3}{8}$ inch spacers
- 14 - thin nylocks



This nylon string acts as a stopper for the hood wh... causing pro... fitting in the size limit

post, hex nut retainer

how the string is attached

third set of stationary sprockets

The other end of the rubberband hook onto the...

I attached the same license plate holders that I used before (see page 71)

Materials used to mount license plate holders:

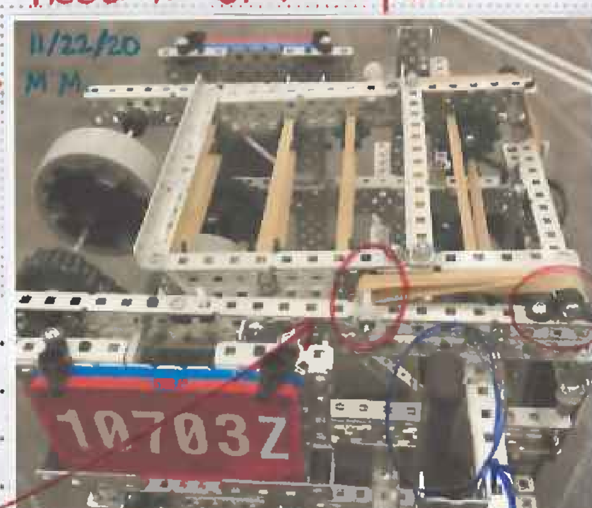
*see pg. 71 for holder materials list *

- 4 - 2 inch standoffs
- 8 - $\frac{1}{2}$ inch locking screws

Materials used in stationary sprockets (set #3):

- 2 - 12T sprockets
- 2 - $\frac{1}{4}$ inch spacers
- 2 - $\frac{3}{8}$ inch spacers
- 2 - $1\frac{1}{2}$ inch screws
- 2 - nylocks
- 2 - rubberbands

Hood in stowed position



The first ball scored pushes up the hood, and it stays up after that due to compression from the rubberbands.

This prevents the hood from going further down and getting stuck

radio mounted here

project:

designed by:

witnessed by:

date: 11/21-22/20

Megan M[redacted] 11/22/20

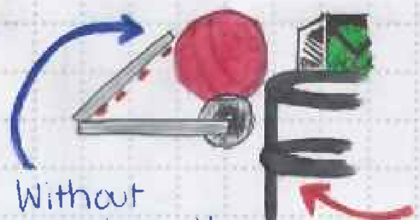
11/22/20

BUILDING THE HOOD CONTINUED

Materials used on the Hood:

- 2 - 1x2x1x11 aluminum c-channels
- 2 - 1x1x14 aluminum angles
- 2 - 1x1x7 aluminum angles
- 8 - 6T sprockets
- 2 - 1 inch standoffs
- 2 - pillow block bearings
- 14 - rubberbands
- 1 - Section of $\frac{1}{8}$" nylon braided rope
- 4 - $\frac{1}{8}$ inch spacers
- 8 - $\frac{1}{4}$ inch spacers
- 4 - $\frac{1}{2}$ inch spacers
- 2 - 8mm shaft spacers
- 6 - nylon washers
- 10 - 0.375 inch screws
- 2 - 0.875 inch screws
- 10 - 1$\frac{1}{4}$ inch screws
- 2 - 2 inch screws
- 14 - thin nylocks
- 10 - nylocks

>TESTING:



Without momentum, the ball ends up here

Problem #1:
- The ball moves all the way through the robot quickly, but doesn't quite make it to the goal when shot from the top of the conveyor.

Problem #2:
- The combined weight of the battery and conveyor motor on the same side makes the robot drive crooked and more prone to tip on that side

HOOD & CONVEYOR IMPROVEMENTS

- I switched the 18T sprocket on the top flex wheel to a 12T sprocket for extra speed to make it into the goal with minimal momentum required.

$$\text{(motor gearing) } 600\text{rpm} \times \frac{18T}{12T}\left(\frac{\text{driver}}{\text{driven}}\right) = \frac{10800}{12} = \boxed{900\text{rpm}}$$

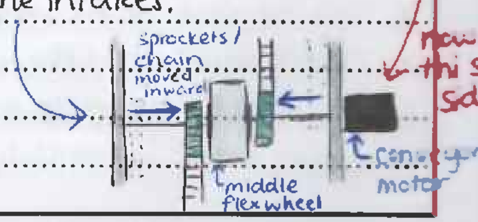
↳ the top conveyor roller now spins at 900 rpm

Addressing Problem #2

- I moved the conveyor motor to the other end of the axel (the side opposite the battery) for a more equal weight distribution.

Preventative measure:
- I moved the sprockets / chain on the conveyor to the middle of the axels (around the flex wheels) to give more room for the intakes.

THE RESULT: The robot drives straighter and the additional speed on the top roller helps score easier with less momentum needed.



Megan M[redacted] 11/22/20

11/23/20

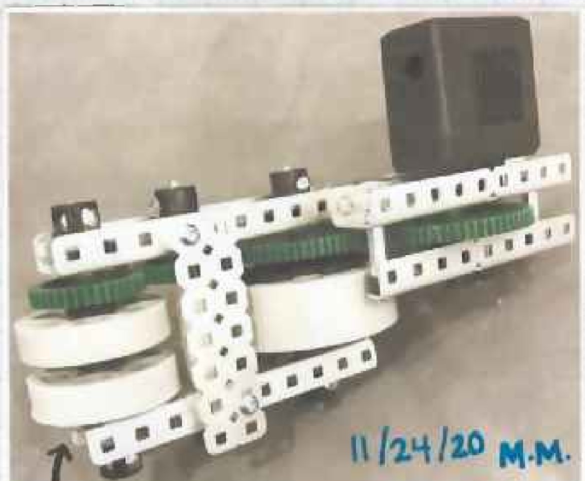
PROTOTYPING THE INTAKES

GOAL: Find a configuration of flex wheels, gears, sprockets / chain that will fit in the size limit (when folded back) and be able to pick up balls / descore.

GOAL COMPLETE 11/28/20

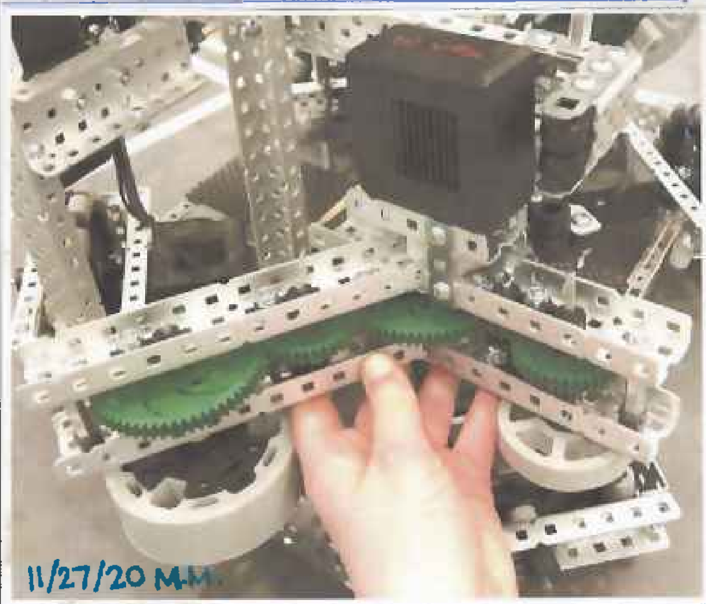
Ideal for 2" OD flex wheels

✓ Small enough to fit in the size limit
✗ No good spot for the motor
✗ Cannot reach into center goal to descore

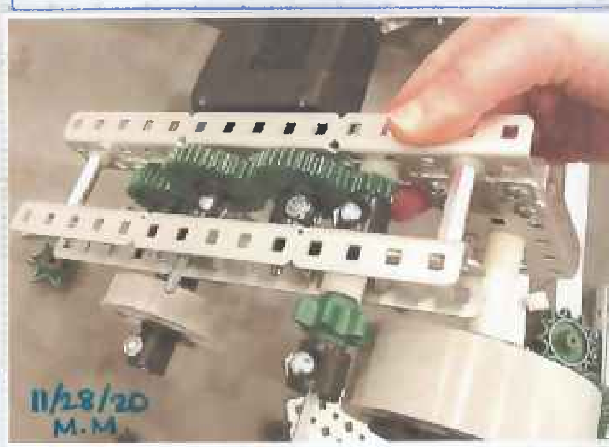


too tall to stow

✓ Very stable and no chain
✗ Cannot fit in the size limit
✗ Cannot descore center goal
✗ Cannot bring ball into conveyor



✓ The motor fits without hitting the goal.
✗ Too angled to fit in the size limit
✗ Flex wheels at different speeds
✗ Difficult for gears to mesh at an angle and be stable



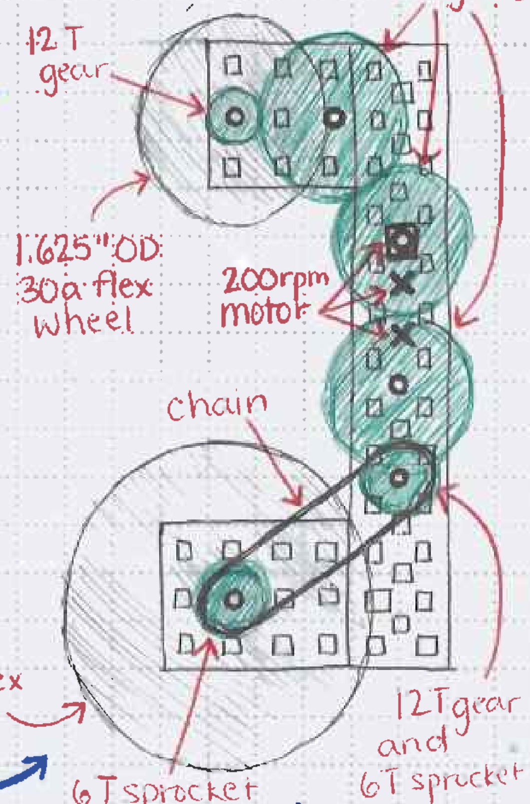
✓ Shaped to descore center goal
✗ Too tall and too wide to fit in the size limit
✗ Chain is exposed and vulnerable

Megan M 11/23/20

11/28/20

PROTOTYPING THE INTAKES CONTINUED

- Based on what I learned from prototyping these different designs, I have a good idea of what the finalized intakes need to look like:
  - The flex wheels need to be spread apart (one inside the goal, one outside when descoring)
  - Small gears only! The ball will catch on large ones.
  - The motor needs to be on a part of the intake that is built out from the rest so it doesn't interfere with descoring.
  - As thin as possible! It must get over the wheel when popping out
  - The smaller roller needs to be close enough to the conveyor to be able to bring the ball in



- From what I can tell, This design should be able to fit in the size limit, bring balls all the way up into the conveyor, and descore all goals (including center goal).

$$200\text{rpm}\left(\frac{\text{motor}}{\text{gearing}}\right) \times \frac{36}{12}\left(\frac{\text{driver}}{\text{driven}}\right) = \boxed{600\text{rpm}}$$

With this design, the intakes will spin at 600 rpm

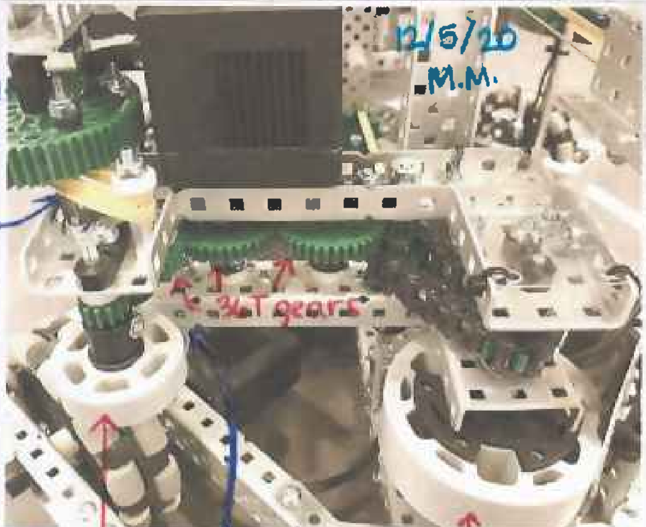
- I will need to build a mount on which the intake can pivot to stow and pick up balls, as well as a stopper with compression inward.

11/29/20

BUILDING THE INTAKES

GOAL: Build and mount the intakes designed on 11/28-29 (page 111) and test its functionality and efficiency. **GOAL COMPLETED 12/6/20**

rubberbands for compression inward



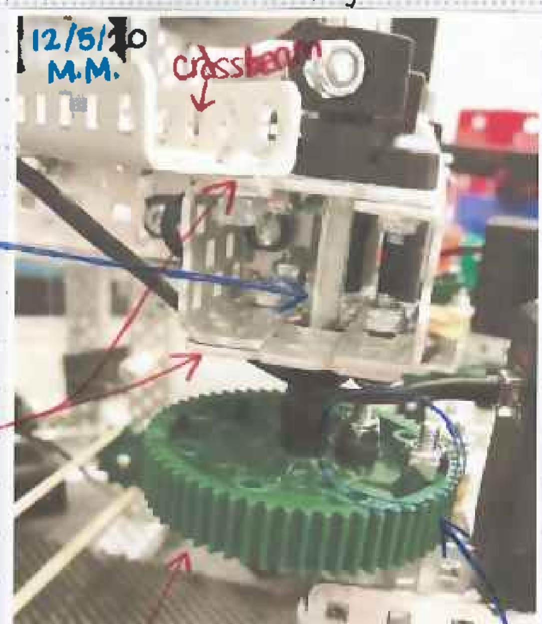
1.625" OD 30a flex wheel

steel plate to conserve space vertically

3" OD 30a flex wheel

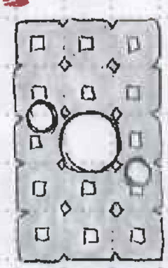
- I was able to follow my design on page 111, using 2 holes wide c-channels for the main structure where the motor is mounted and 3 holes wide c-channels / plate for the add-ons.

- I used a 60T High strength gear for a swing pivot - the center of the gear is what the intake pivots around, but it is mounted further out to swing around.



pivot is a high strength axel

- I specially cut two aluminum plates on each side for the high strength flat bearing to attach cleanly.



5x3 plate

- I attached an 8 holes long c-channel as a cross beam between the intake mount and the lowest horizontal bar on the conveyor (where the mount is connected)

60T gear for swing pivot

mounted to the gear here

project designed by: witnessed by:

date: 11/29-12/6/20

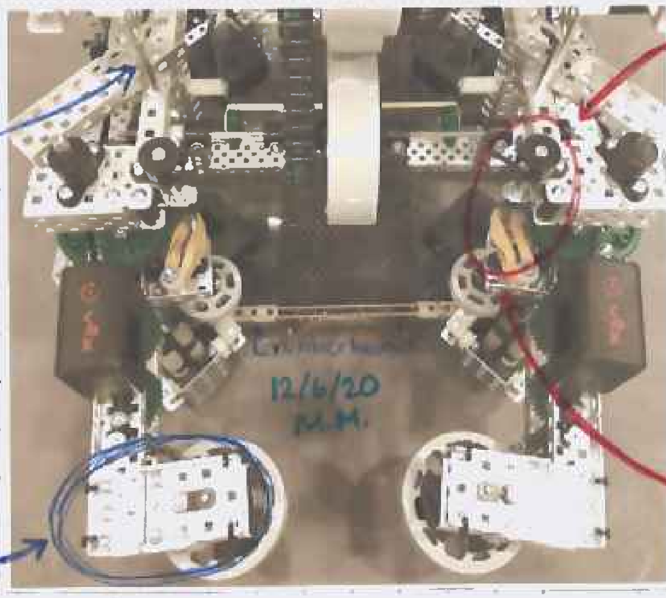
Megan M 12/1/20

12/4/20

BUILDING THE INTAKES CONTINUED

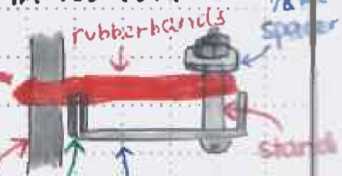
Standoff between low and mid horizontal conveyor bars for additional support

- I attached an approximately 3 inch by 1.5 inch piece of polycarbonate to prevent the intake from getting caught on a goal.



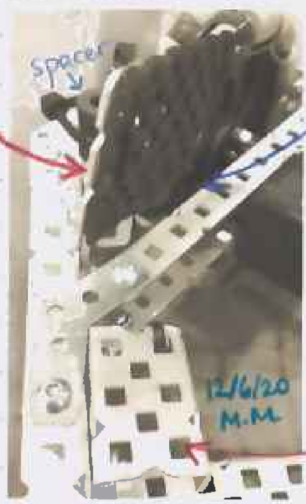
Stopper

- I used a high strength shaft with a lock bar as the stopper to prevent the intakes from coming in too far.



This eliminates the need for a separate compression point.

- In order for the intakes to stow, I ziptied a 3x4 plate to the outer vertical support bar on each side with a $\frac{1}{2}$ inch spacer to keep it at the right angle.



The flex wheel must slightly compress to be stowed, but they then can pop out easily when the intake is spun in reverse.

small square of polycarbonate so the intake can't get stuck in any of the holes

- I found that the intakes have a little bit of trouble getting the balls up the edge of the path, so I added a single rubberband to help bump it up.

rubberband wrapped around a ½ in. standoff

stretched across the front of the robot (see picture above)

Materials used in intake stowing:

- 2 – 1x2x1x3 aluminum c-channels
- 2 – 3x4 aluminum plates
- 2 – ≈ 1.5 in. by 1.5 in. polycarbonate
- a number of 4" zipties
- 2 – $\frac{1}{2}$ in. spacers
- 2 – $\frac{1}{2}$ in. screws
- 2 – thin nylocks

project | designed by: | witnessed by:

date: 11/29 – 12/6/20

Megan M[redacted] 12/6/20

12/6/20

BUILDING THE INTAKES CONTINUED

Materials used in the intakes:

- 2 - V5 Smart motors (200rpm)
- 2 - 3" OD 30a flex wheels
- 2 - 1.625" OD 30a flex wheels
- 4 - 12T gears
- 6 - 36T gears
- 4 - 6T sprockets
- 2 - sections of chain
- 4 - 1x2x1x13 aluminum c-channels
- 4 - 1x3x1x4 aluminum c-channels
- 2 - 1x3x1x3 aluminum c-channels
- 2 - 1x2x1x8 aluminum c-channels
- 2 - 1x2x1x4 aluminum c-channels
- 2 - 1x2x1x3 aluminum c-channels
- 2 - 2x5 aluminum plates
- 4 - 3x5 aluminum plates
- 2 - 5x3 steel plates
- 6 - 3x3 steel plates
- 2 - 60T high strength gears
- 4 - high strength shafts (cut)
- 12 - shafts (cut)
- 2 - high strength lock bars
- 4 - 1x2 steel bars
- 2 - 2inch standoffs
- 8 - 1 inch standoffs
- 14 - clamping shaft collars
- 6 - high strength clamping shaft collars
- 14 - flat bearings
- 2 - shaft collars
- 2 - rubber shaft collars
- 4 - rubberbands
- 2 - 3" by 1.5" polycarbonate
- 21 - nylon washers
- 16 - $\frac{1}{8}$ inch spacers
- 10 - $\frac{1}{4}$ inch spacers
- 2 - $\frac{3}{8}$ inch spacers
- 8 - $\frac{1}{2}$ inch spacers
- 8 - high strength flat bearings
- 4 - high strength washers
- 2 - high strength $\frac{1}{8}$ inch spacers
- 2 - high strength $\frac{1}{2}$ inch spacers
- 2 - $\frac{1}{2}$ inch locking screws
- 42 - 0.375 inch screws
- 14 - $\frac{1}{2}$ inch screws
- 14 - 0.625 inch screws
- 6 - 0.875 inch screws
- 2 - $1\frac{1}{2}$ inch screws
- 4 - $1\frac{3}{4}$ inch screws
- 6 - 2 inch screws
- a number of 4" zipties
- 14 - nylocks
- 66 - thin nylocks

Materials used in the bridge between intakes and conveyor:

- 2 - $\frac{1}{2}$ inch standoffs
- 2 - 0.375 inch screws
- 1 - rubberband

Overall, the intakes work well for descoring and picking up balls. Not as stable as I would like.

~M.M. 12/6/20

project

designed by:

witnessed by:

date: 11/29 - 12/6/20

PROPRIETARY INFORMATION all information is the property of, and solely owned by the Designer.

Megan M 12/6/20

12/7/20

BALL GUIDES, INERTIAL SENSOR, PROGRAMMING

GOAL: Make any final building improvements, attach any essential sensors, and program driver control.

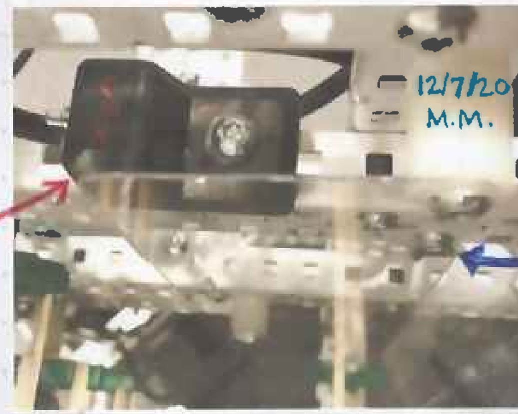
GOAL COMPLETED 12/9/20

- With time running short until the Bluff City competition (Dec. 12), I need to prioritize to make sure the robot is able to compete.

- In terms of programming, I'm only adding an inertial sensor, which will speed up programming turns. I plan to use the built-in motor encoders for now.

- I attached a 10" by 2" polycarbonate strip to serve as ball guide on each side of the conveyor.

inertial sensor



ball guide

→Materials lists:←

- 1-V5 inertial sensor
- 1-nylon washer
- 1-thin nylock
- 1-0.875 in. screw
- 2-0.875 in. screws
- 4-1 1/4 in. screws
- 10-1/2 in. spacers
- 6-thin nylocks
- 2-10" by 2" polycarbonate

My driver control functions are all basic, except for this one

```
int sorterManual(int sortingArray[2]) {
  if(Controller.ButtonL1.pressing() == true && Controller.ButtonR2.pressing() ==
  true) { if both buttons are pressed, disposing overrides scoring
    sortingArray[1] /* sorterOutActive */ = true; // "locks" all other functions that
    use the sorter motor
    SortingMotor.spin(directionType::rev,600,velocityUnits::rpm); // dispose ball
  }
  else if(Controller.ButtonL1.pressing() == true) {
    sortingArray[0] /* sorterInActive */ = true; // "locks" all other functions that
    use the sorter motor
    SortingMotor.spin(directionType::fwd,600,velocityUnits::rpm); // bring ball up
    conveyor  The conveyor motor spins foward with L1 being pressed as well
  }
  else if(Controller.ButtonR2.pressing() == true) {
    sortingArray[1] /* sorterOutActive */ = true; // "locks" all other functions that
    use the sorter motor
    SortingMotor.spin(directionType::rev,600,velocityUnits::rpm); // dispose ball
  }
  if(Controller.ButtonL1.pressing() == false && Controller.ButtonR2.pressing() ==
  false) {
    SortingMotor.stop(brake);
    sortingArray[0] /* sorterInActive */ = false; // "unlocks" all other functions
    that use the sorter motor
    sortingArray[1] /* sorterOutActive */ = false;
  }
  return sortingArray[2]; // returns the values to use in the next iteration
}
```

12/9/20 M.M.

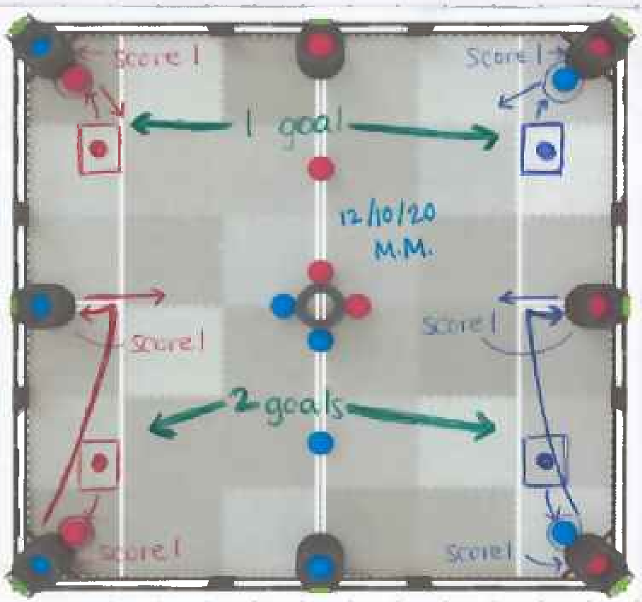
date: 12/7-9/20

Megan M 12/9/20

 12/10/20

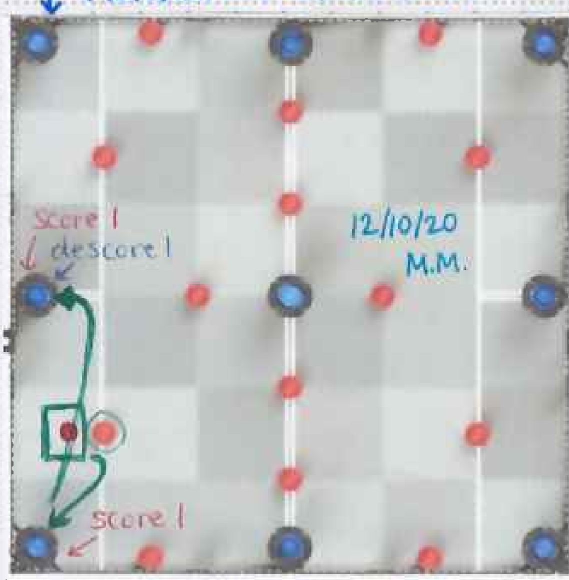
PROGRAMMING AUTON. & SKILLS

GOAL: Program basic autonomous programs to score in one and two goals using the motor encoders and inertial sensor. **GOAL COMPLETED 12/10/20**



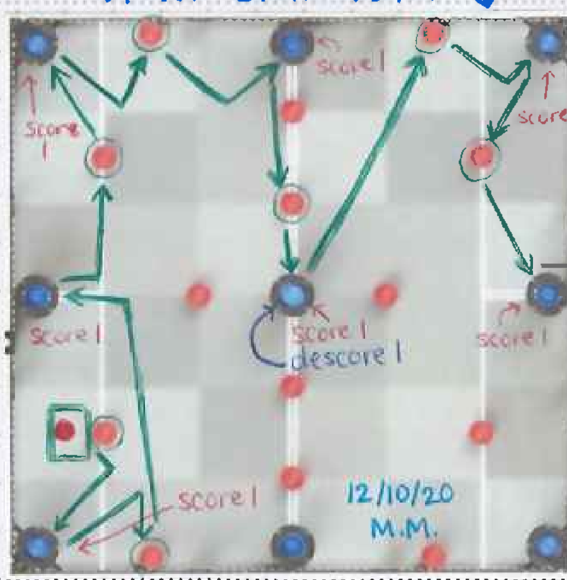
- I programmed four autonomous (15 second) routines: right side of the field, scoring one or two goals; and left side of the field, scoring one or two goals.
- These will enable me to work well with my alliances at the competition.
- I programmed a two-goal routine for autonomous skills as well.

Autonomous Skills Path



$2 - 38 + 63 = \boxed{27 \text{ points}}$

Driver Skills Path

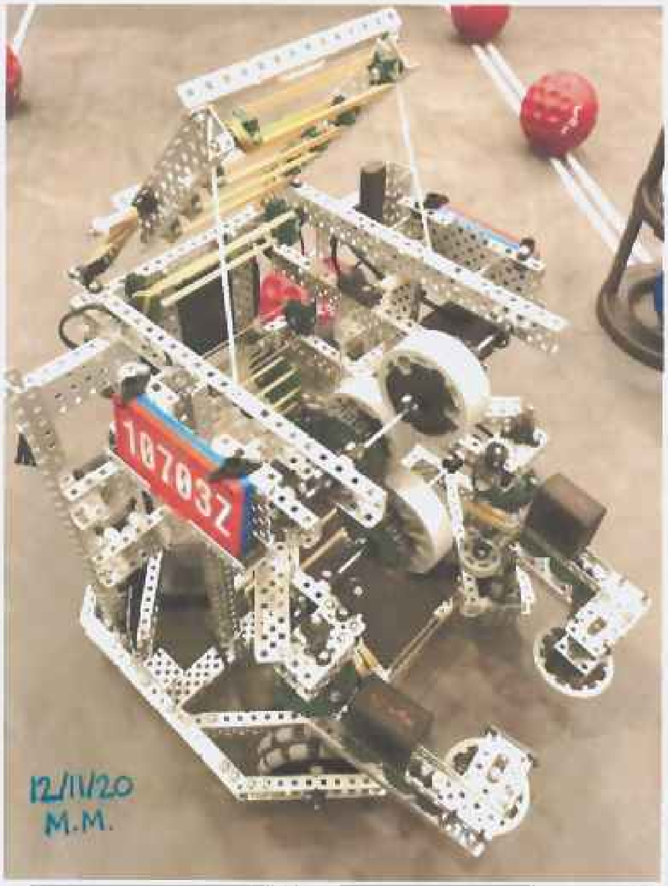


$31 - 14 + 63 = \boxed{80 \text{ points}}$

(combined score would be 107 points)

12/11/20

10703Z for the Bluff City Competition



SUCCESSES:

- Finishing a complete rebuild
- Ability to score and descore all goals
- Higher speed and efficiency with the intakes and conveyor
- Very functional method of disposing
- One and two goal autonomous runs

THINGS TO IMPROVE:

- Implementing more sensors
- Higher scoring autonomous skills
- More driving practice and macros
- More consistency in autonomous

MY GOALS FOR THE BLUFF CITY COMPETITION:

☑ Get a second interview. [Completed 12/12/20]

☑ Rank in the top half after qualifications.

☒ Score 80 points in Driver Skills ✓ and 27 points in Autonomous Skills ✗ for a total of 107 points. ✗

Megan M 12/11/20

12/12/20

BLUFF CITY COMPETITION ANALYSIS

10703Z Stats:

- Ranked 3rd after qualifications
- Ranked 4th in skills, with a total score of 99 points
- 7 wins, 3 losses in qualifications

- WP = 17 ⟵ 3 WP from completing the home row in autonomous
- AP = 36
- SP = 101 ← highest SP in the tournament!

Autonomous Routines Success Rates:

- Left side, 2 goals : used 8 times
  - fully worked : 63% (5 times)
  - partially worked : 25% (2 times)
  - completely failed : 12% (1 time)
- Right side, 2 goals : used once
  - fully worked : 0 %
  - partially worked : 100%
  - completely failed : 0 %
- Left side, 1 goal : used once
  - fully worked : 100%
  - partially worked : 0 %
  - completely failed : 0 %
- Right side, 1 goal : used 2 times
  - fully worked : 0%
  - partially worked : 0%
  - Completely failed : 100%

Bluff City Skills Results 12/12/20 M.M.

Driver / Auton	Score	Analysis
Driver run #1	80	Scored 7 red balls, one in each goal on the path drawn on pg. 116
Driver run #2	66	Scored 5 red balls (one in each goal to complete the home row and left diagonal). Descored 1 blue ball. Missed a goal.
Auton run #1	19	The robot scored the preload in the corner goal, but caught on the second goal, unable to complete a command
Auton run #2	19	Same as Auton run #1, a boolean wait for completion command could not be met, so the program did not continue.

project | designed by: | witnessed by:

date: 12/12/20

Megan 12/12/20

12/12/20

BLUFF CITY COMPETITION ANALYSIS CONTINUED

Bluff City Qualification Match Results 12/12/20 M.M.

Alliance Partner	Opponents	Win / Loss	Score	Analysis
63303C	16859B 97934U	Loss	8-19	Right, 2 goal auton scored 1 point. decent strategy good descoring
24816H	19589A 24816M	Win	16-12	Left, 2 goal auton scored 2 points good center goal scoring decent strategy
63303V	16859B 19589A	Win	36-7	Left, 2 goal auton scored 2 points good scoring and strategy
63303V	24816M 63303A	Win	50-11	Left, 2 goal auton scored 0 points good strategy and center goal scoring
98709C	663C 24816V	Loss	6-27	Right, 1 goal auton scored 0 points bad defense and scoring
98709C	663B 24816T	Win	12-8	Right, 1 goal auton scored 0 points good strategy bad center goal defense
63303A	97934X 24816T	Win	19-7	Left, 2 goal auton scored 2 points good scoring and defense
24816H	98709C 97934Y	Loss	11-22	Left, 2 goal auton scored 2 points bad scoring and defense
97934X	63303V 663B	Win	22-21	Left, 1 goal auton scored 1 point good strategy bad scoring
97934Y	663C 63303A	Win	20-10	Left, 2 goal auton scored 2 points decent strategy decent scoring

date: 12/12/20

Megan P 12/12/20

12/12/20

BLUFF CITY COMPETITION ANALYSIS CONTINUED

Bluff City Elimination Match Results

12/12/20
M.M.

Alliance Partner: 63303V

Opponents	Win / Loss	Score	Analysis
24816H 63303A	Win	23-10	QF #4-1 Left, 2 goal auton scored 1 point. 63303A played good defense good strategy
24816V 97934U	Loss	1-43	SF #2-1 Left, 2 goal auton. scored 1 point. 24816V and 97934U were able to out-cycle us

- I won Excellence and the Sportsmanship award (also 2nd place for best ugly Christmas sweater)!

INSPIRATIONS FROM OTHER TEAMS:

- **663B:** differential gearing as an alternative to a ratchet to control separate mechanisms with a single motor
- **24816V** and **97934U** had mobile backboards to increase accuracy when scoring, especially for the center goal - this also allowed them to easily defend a goal from scoring.
- Many teams including **97934X**, **98709C**, and **63303V** focused on goals in an L shape, which guarantees a win if all five goals are owned by your alliance.

WHAT I LEARNED:

- **The significance of the Win Point for completing the home row in autonomous in regards to rankings**: I had a 7-3-0 win-loss-tie record, but I was above a 7-2-1 team because of win points (which were tied, but I had a higher AP). My alliance partner, 63303V, had a 6-4-0, but were ranked above 3 7-3-0 teams.

Megan N 12/12/20

12/14/20

DESIGN CHANGES AFTER BLUFF CITY COMPETITION

PROBLEMS TO ADDRESS / IMPROVEMENTS NEEDED:

1. The balls sometimes get jammed in the conveyor needs a better flow / path
2. More stability for the intakes by mounting on the base
   ↳ mounting the intakes on the base requires switching to 3.25" OD omni-wheels from the current 4" OD wheels (less distance with the same speed)

BRAINSTORMING / RESEARCH: Possible solutions to each problem:

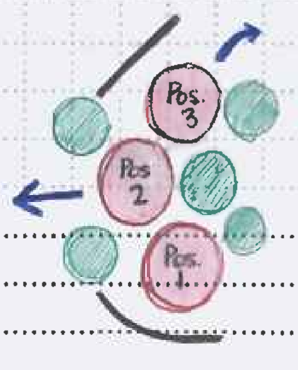
1. Improved conveyor flow

1. Remove disposing altogether
- X Less functional
- X Makes the robot front heavy

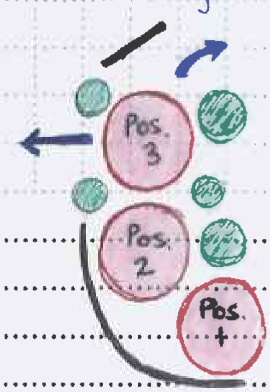
2. Restructure to dispose from position 2
- X Potentially lower ball capacity
- X Not efficient with motor power

3. Restructure to dispose from position 3
- X not very viable with size restrictions
- X Cannot score and dispose simultaneously
- X Lower ball capacity

Option One is the most simple by far but not functionally ideal



Option 2



Option 3

Research Notes:

- In order to have a straight intake while being able to cycle the center goal, the diameter of the wheels has to be fairly large (most teams use 4" OD flex wheels or other roller)
- 662B's intakes work very well with a 4" and a 3" roller on each intake. They also have a unique way of attaching the flex wheel to the axel. (using standoffs) (662B Kepler - End of 2020 Reveal - Change Up on YouTube and the VEX Forum)
- Potentiometers can be used on intakes that fold out to measure the angle of each (I saw this on an online challenge entry a while back but I couldn't find it again to give credit to the team)

Megan M 12/15/20

12/15/20

DESIGN CHANGES AFTER COMPETITION CONTINUED

BRAINSTORMING / RESEARCH CONTINUED:

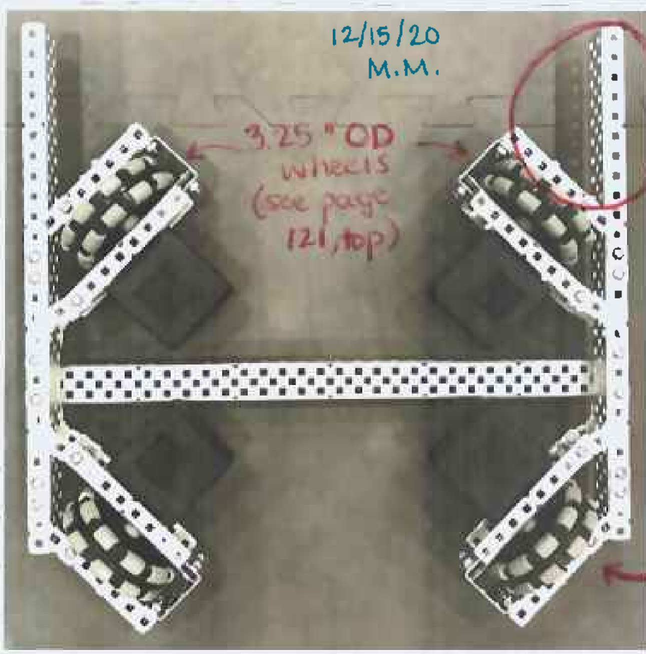
② Improved intake stability

1. Keep the design from pgs. 111-113 (used at Bluff City competition) but mount to the base instead of the conveyor.
- X shape is not conducive for folding to fit size limit
- X interference with front drive wheels
- X not very efficient at intaking and descoring

2. Build stationary intakes with a smaller base so intakes **do** not have to fold in to fit the size limit (mount to base)
- X smaller base = smaller conveyor = lower ball capacity
- X intakes must reach to the back of goals (stationary intakes have to be fairly short)

3. Build a straight intake with large rollers like 662B Kepler (see pg. 121) but have them fold in to fit the size limit and mounted to the base.
- X large rollers cause difficulty in fitting the size limit.
- X folding intakes are less stable by nature.

Prototyping for Option 2:

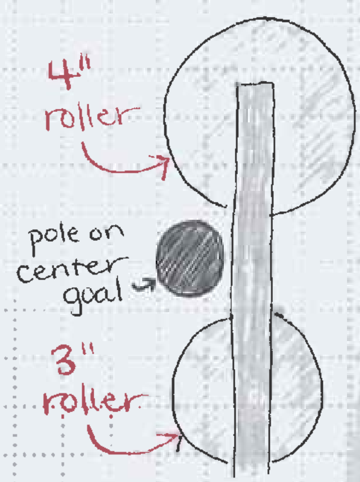


intakes would be mounted here, angled in

this is not conducive for having 2 rollers per intake to descore

The remaining space is not enough to dispose

Option 3



In order for the intakes to be able to descore balls from the center goal into the robot, the rollers have to be large enough to reach around the poles.

project designed by: witnessed by:

The wheels form a rectangle, not a square, which makes turning slower due to more friction

date: 12/14-19/20

Megan M 12/15/20

Kevlar was invented by Stephanie Kwolek for Du Pont in 1963. A new branch of synthetic materials were created called liquid crystalline polymers that were five times stronger than steel. It is used for lightweight body armor.

DESIGN CHANGES AFTER COMPETITION CONTINUED

> SELECT OPTION:

Problem #1 — Options to Improve Conveyor Flow — 12/19/20 M.M.

Criteria (least to most important)	Scale	① Remove disposing	② Restructure with disposing in position 2	③ Restructure with disposing in position 3
Few potential problems	0 to 3	3	2	0
Viable with space limitations	0 to 5	5	4	0
Efficiency and Speed	0 to 5	2	5	4
	Total Score:	10	11	4

Many teams have done this successfully, so this is a good option to improve the path and increase efficiency.

Problem #2 — Options to Improve Intake Stability — 12/19/20 M.M.

Criteria (least to most important)	Scale	① Same design as before but mounted to base	② Stationary intakes mounted to base	③ Straight folding out intakes mounted to base
Few potential problems	0 to 3	0	1	2
Viable with space limitations	0 to 5	1	1	4
Increased stability	0 to 5	2	5	3
Ability to descore all goals	0 to 7	5	4	7
	Total Score:	8	11	16

As long as I make sure that the rollers fit in the size limit, these intakes should work very well with relatively few issues to deal with.

project — designed by: — witnessed by:

date: 12/14-19/20

Megan M[redacted] 12/19/20

12/21/20

DESIGN PLAN : Build #3

- My third design is specifically targeting the issues I've encountered in the first two design processes. The concept is very similar to build #2 (see page 98) but improved for better, more stable intakes as well as a more efficient and smoother conveyor path.

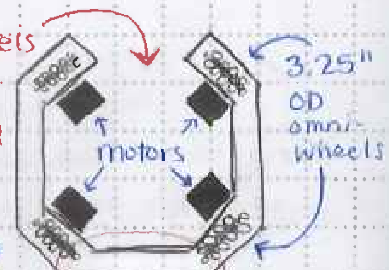
DRIVETRAIN : X-Drive

- 4 motors, 200 rpm cartridges (x-configuration increases this speed by ≈1.4)
- 3.25" OD wheels (omni-directional)

Having smaller wheels allows more room for the intakes while retaining a small base

- I plan to maximize drivetrain speed by making the robot as lightweight as possible:
  - ✓ good weight distribution
  - ✓ no flex wheels on conveyor
  - ✓ square x-drive for faster turning

The smaller wheels allow the base to be 16" wide and still allow the ball to fit (good for intake stowing)

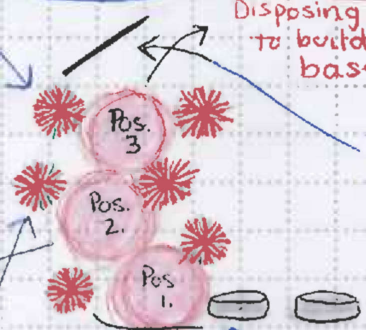


Disposing higher allows me to build a more stable base structure

CONVEYOR : Rubberband rollers on both sides of the ball

- 1 motor, 600 rpm cartridge
- chain / sprockets and gears to connect the rollers and spin the correct direction

Rubberband rollers are significantly lighter than flex wheels, and having rollers on both sides of the ball gives it more speed and thrust



HOOD:

- aluminum frame with anti-slip mat

SORTER : one rubberband roller to sort disposing/scoring

- 1 motor, 600 rpm cartridge
- same as build #2 (see pg. 98) but disposes a ball from higher up

INTAKES : straight, folding with large rollers

- 2 motors, 200 rpm cartridges
- 3" OD and 4" OD 30a flex wheels connected with chain
- spaced for descoring center goal

project designed by: witnessed by:

date: 12/21/20

Megan M[redacted] 12/21/20

12/21/20

TIMELINE : UNTIL THIRD COMPETITION

When classes start up again in January, I won't have as much time to work, so this is the timeline of absolute deadlines. The building has to be done by January 28th at the latest, leaving not long to add sensors and programming. ~ M.M. 12/21/20

12/21/20
M.M.

Structure/Mobility
Base, X-Drive, Structure
12/28 - 12/30

December

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	~~22~~	~~23~~	~~24~~	~~25~~	~~26~~
~~27~~	28	29	30	31		

today

Christmas break

Gameplay Mechanisms
Conveyor, Hood, Disposing, Intakes, Improvements
12/31 - 1/28

December

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

January

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Programming and Practice
Sensors, Macros, Autonomous, Skills
1/29 - 2/5

January

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

February

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

FRA competition

Robo Rumble
—CANCELED—

Calendar Key:

days with time to work on the robot

tournament

project designed by: witnessed by:

date: 12/21/20

Megan M 12/21/20

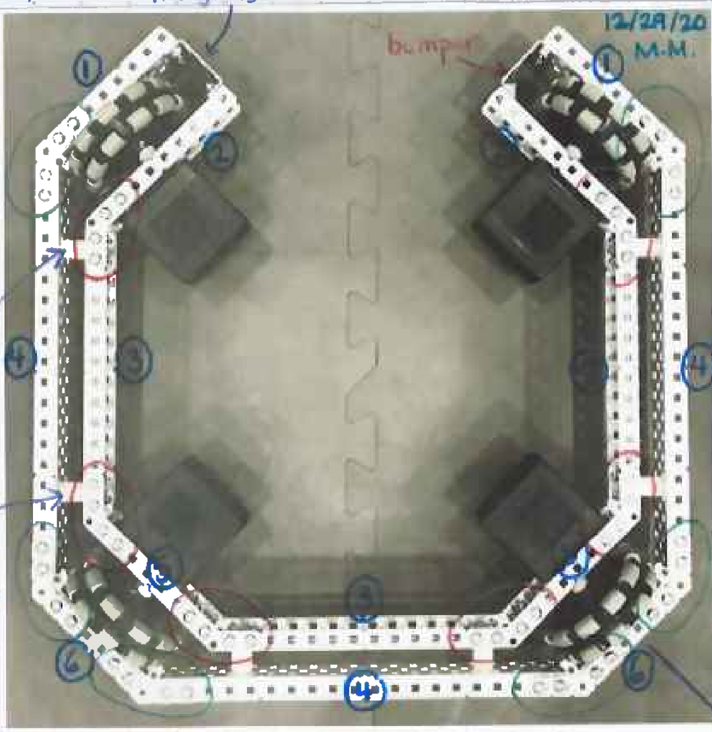
12/28/20

BUILDING X-DRIVE & BASE STRUCTURE

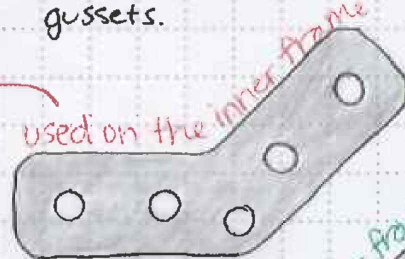
GOAL: Build a 16" by 16" x-drive with as perfect 45° angles as possible. Add support bars to serve as the base structure to attach the conveyor and intakes to. **GOAL COMPLETED 12/30/20**

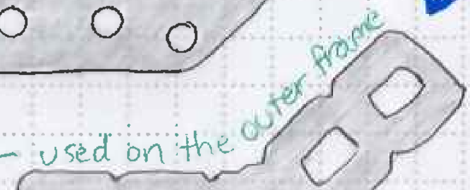
This 3-hole wide c-channels serves as a bumper for the goals

5/8 in. of spacers



- VEX sells two kinds of 45-Degree gussets (same angle, different holes). I found that the inner and outer x-drive frames line up with different 45-Degree gussets.





Dimensions: (numbered above)

1. 8 holes long, 4.5 inches to corner of gusset
2. 9 holes long, 5 inches to corner of gusset

These sections are longer than 6 and 5, respectively, to allow the 3 hole wide c-channel (bumper) to contact the goal at the right angle.

3. 14 holes long, 7 3/8 inches from corner to corner of the gussets
4. 20 holes long, 11 inches from corner to corner of the gussets
5. 8 holes long, 4.5 inches from corner to corner of the gussets
6. 6 holes long, 3 3/4 inches from corner to corner of the gussets

Order on the axel:

V5 Smart Motor (200 rpm)
c-channel
1/4 inch spacer
shaft collar
omni-wheel
shaft collar
1/8 inch spacer
c-channel
flat bearing

(same for all four wheels)

project | designed by: | witnessed by:

date: 12/28-30/20

Megan T 12/28/20

12/29/20

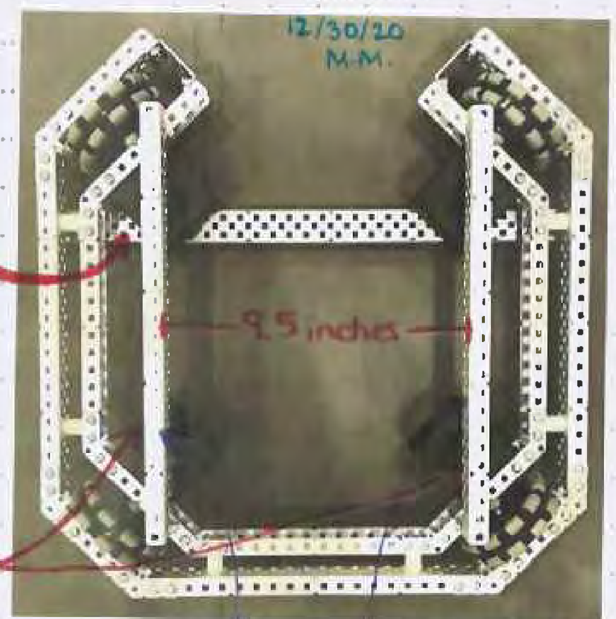
BUILDING X-DRIVE & BASE STRUCTURE CONTINUED

- I added one support bar across the bottom of the x-drive for stability in the front.

32 holes long, 16 inches

- In order to attach vertical bars for the conveyor far enough inward to allow the intakes to fold into the size limit, I added two bars on $\frac{1}{2}$ inch standoffs (to avoid the wheels) across the base.

→ 24 holes long, 12 inches



The intakes and vertical bars on the conveyor will be mounted to these bars

→ Materials used on the x-drive and base structure: ←

- 4 - V5 Smart motors (200rpm)
- 4 - 3.25" OD omni-wheels
- 2 - 1x2x1x24 aluminum c-channels
- 1 - 1x2x1x32 aluminum c-channel
- 3 - 1x2x1x20 aluminum c-channels
- 3 - 1x2x1x14 aluminum c-channels
- 2 - 1x2x1x9 aluminum c-channels
- 4 - 1x2x1x8 aluminum c-channels
- 2 - 1x3x1x3 aluminum c-channels
- 24 - 45-Degree gussets
- 4 - axels (cut)
- 4 - $\frac{1}{2}$ inch standoffs
- 4 - flat bearings
- 8 - shaft collars
- 4 - nylon washers
- 8 - $\frac{1}{8}$ inch spacers
- 14 - $\frac{1}{4}$ inch spacers
- 6 - $\frac{3}{8}$ inch spacers
- 104 - 0.375 inch screws
- 4 - $\frac{1}{2}$ inch screws
- 8 - 0.625 inch screws
- 4 - 0.875 inch screws
- 6 - 1 inch screws
- 118 - thin nylocks

project

designed by:

witnessed by:

date: 12/28-30/20

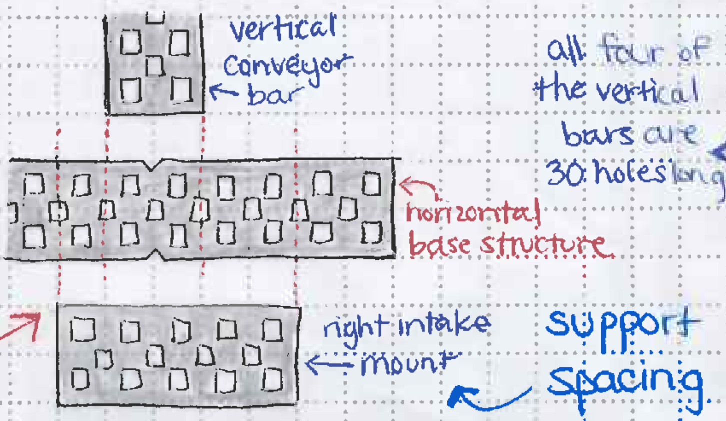
Megan M 12/30/20

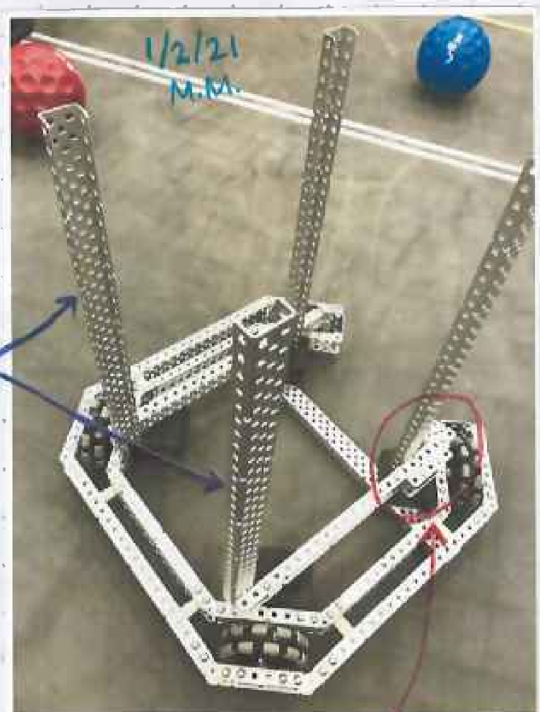
 12/31/20

ADDING THE CONVEYOR SUPPORT BARS

GOAL: Attach the four vertical bars that the conveyor will be built on and build a mount for the intakes. **GOAL COMPLETED 1/2/21**

- I attached the back two vertical bars to the last two holes on the horizontal base structure to maximize space.



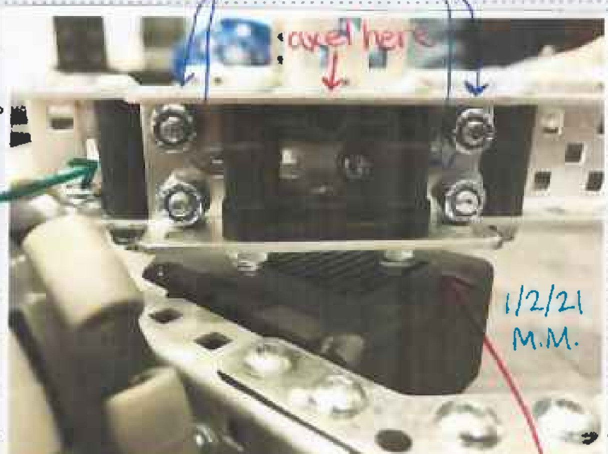


right intake mount

- The front vertical bars are four holes in due to the intakes and so that the top roller won't hit the goal.

$\frac{1}{2}$ in. standoffs

this prevents bending.



left intake mount (axel through center hole)

Materials used in the conveyor structure and intake mounts:

- 4 - 1x2x1x30 aluminum c-channels
- 2 - 1x2x1x5 aluminum c-channels
- 10 - $\frac{1}{2}$ in. standoffs
- 4 - flat bearings
- 4 - $\frac{1}{4}$ inch locking screws
- 8 - 0.375 inch screws
- 6 - $\frac{1}{2}$ inch screws
- 8 - 0.875 inch screws
- 10 - $1\frac{1}{4}$ inch screws
- 26 - thin nylocks
- 10 - $\frac{3}{8}$ inch spacers
- 4 - nylon washers

project | designed by: | witnessed by:

date: 12/31/20 - 1/2/21

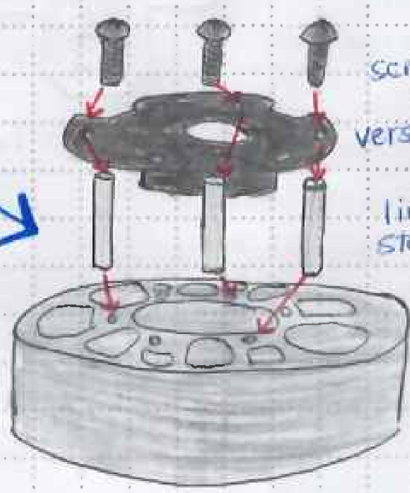
Megan M 1/2/21

BUILDING THE INTAKES

GOAL: Build the intakes with correct spacing for descoring the center goal, being mounted to the base / supported by the conveyor, and for folding in the size limit. **GOAL COMPLETED 1/8/21**

- My intakes are inspired by 662B Kepler's intakes. This team uses only one versa hub with their flex wheels and then standoffs to hold the flex wheel in place.

662 B Kepler - End of 2020 Reveal - Change Up on YouTube and VEX Forum



$$\frac{200\text{ rpm}}{1} \times \frac{12\text{ T}}{6\text{ T}} = \boxed{400\text{ rpm}}$$

The front roller spins twice as fast as the back one.

12 T sprocket

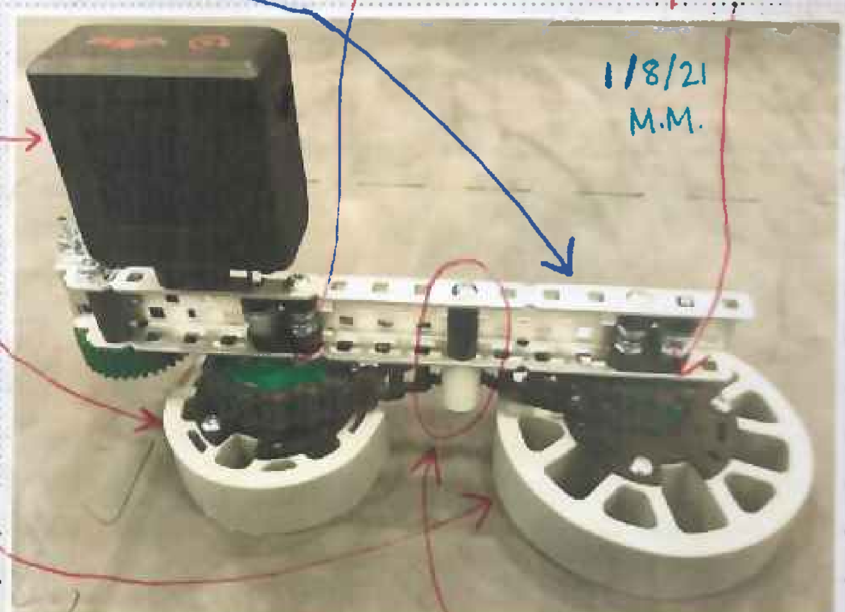
6 T sprocket

- The intake arm is 17 holes long, with the motor directly connected to the back roller in the 6th hole from the end.

200 rpm cartridge

3" OD 30a Flex wheel

4" OD 30a flex wheel



1/8/21 M.M.

My left intake

this holds the chain out of the way of the poles on the center goal

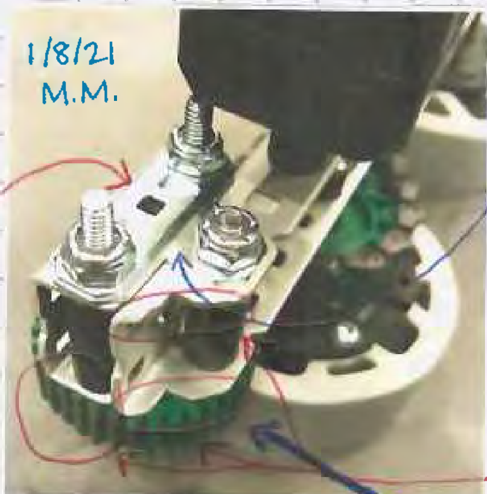
project designed by: witnessed by:

date: 1/4-8/21

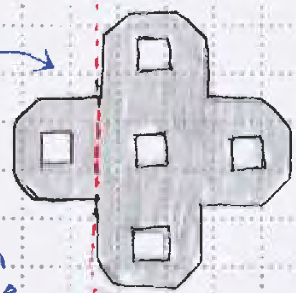
Megan 1/4/21

BUILDING THE INTAKES CONTINUED

- In order for the motor to fit next to the vertical conveyor bar when mounted (see page 128), the intake mounting / pivot point has to be offset a hole from the arm. This allows for a good distance between the back rollers as well.



under the lockbar, there is a cut plus gusset to connect the intake arm to the 3 holes long c-channel



these cuts and the filed portion on the 36T gear allow the intake to clear the vertical conveyor bar when folding in and out with the pivot

cut here

left intake pivot point

The 36T gear provides a lot of stability

→ Materials used in building the intakes: ←

- 2 - V5 smart motors (200 rpm)
- 2 - 4" OD 30a flex wheels
- 2 - 3" OD 30a flex wheels
- 2 - 12T sprockets
- 2 - 6T sprockets
- 2 - 1x2x1x17 aluminum c-channels
- 2 - 1x2x1x3 aluminum c-channels
- 2 - plus gussets
- 2 - lockbars
- 2 - 36T gears
- 12 - 1 inch standoffs
- 4 - axels (cut)
- 8 - flat bearings
- 4 - clamping shaft collars
- 4 - shaft collars
- 20 - nylon washers
- 12 - $\frac{3}{8}$ inch spacers
- 8 - $\frac{1}{2}$ inch spacers
- 20 - 0.375 inch screws
- 8 - $\frac{1}{2}$ inch screws
- 2 - 1$\frac{3}{4}$ inch screws
- 6 - 2 inch screws
- 22 - thin nylocks
- 2 - sections of chain

project designed by: witnessed by:

(Continued in Book 2)

date: 1/4-8/21

Megan M 1/8/21

Robotics Engineering Notebook

team name: BrainSTEM Botz

team number: 10703Z

season: 2020 - 2021

start date: 1/11/21

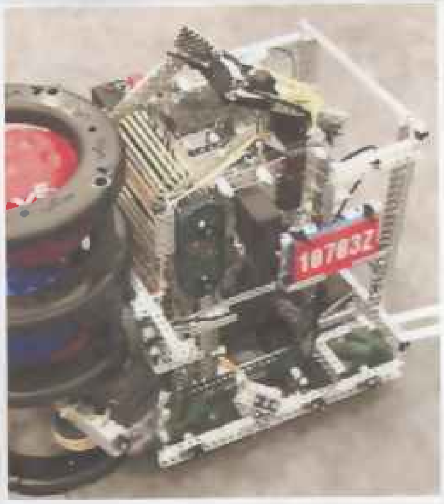
end date:

book number: 2 of:

Team Photo

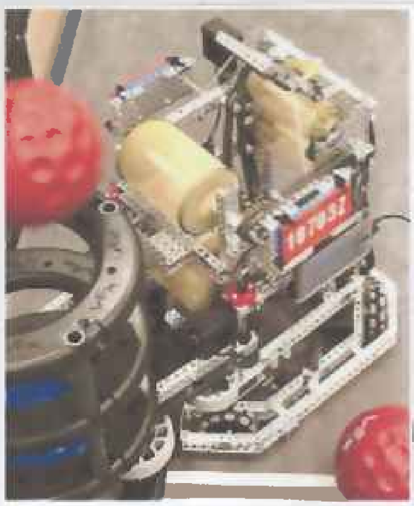
10703Z • BrainSTEM Botz

Build #1



Build #2

Build #3, Version #1



Build #3, Version #2

*see this page in Book One for team profile, color coding key, and the design cycle!

Awards this season:

- Design Award at BA Halloween competition
- Excellence Award at Bluff City SEMS Christmas competition
- Sportsmanship Award at Bluff City SEMS Christmas competition
- Design Award at FRA competition
- Tournament Finalist at FRA competition
- Design Award at Bolt Up competition in Chattanooga
- Design Award at Bluff City SEMS St. Patrick's Day competition
- Robot Skills Champion at TN State Championship
- Excellence Award at TN State Championship
- Excellence Award at the Live Remote World Championship

[Continued from Book 1]

My Projects

page	project	date

TABLE OF CONTENTS

Continued →

My Projects

1/11/21

PROTOTYPING THE CONVEYOR

GOAL: Find a configuration of rollers that provide the ball a smooth flow for both scoring and disposing.

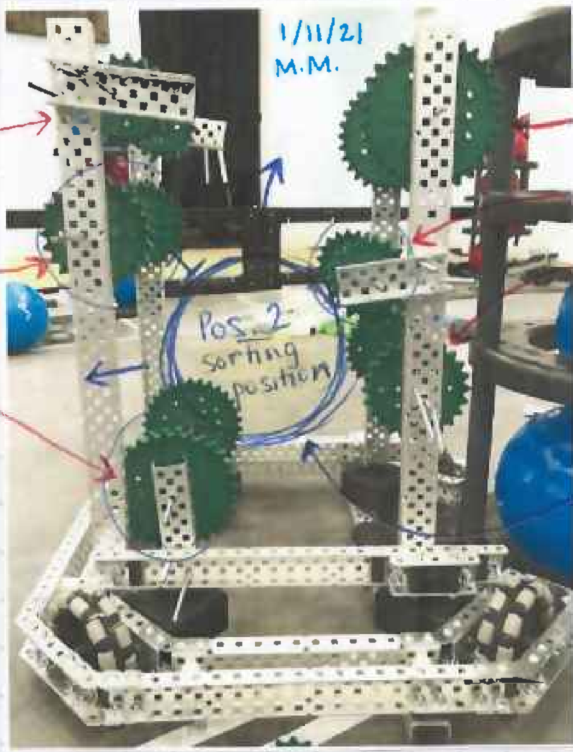
GOAL COMPLETED 1/12/21

- The trick is to find a path with the right amount of compression all the way through (once rubberbands are added).

18T sprocket

24T sprocket

30T sprocket



← Early Configuration

30T sprocket

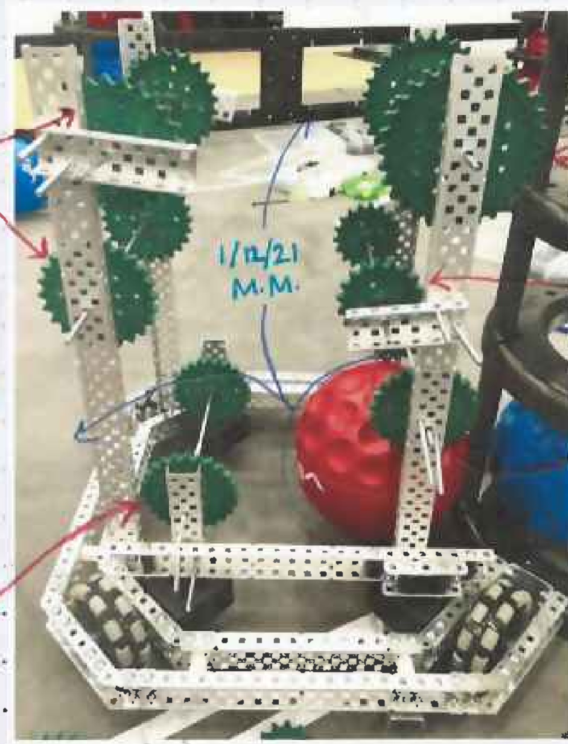
18T sprocket

24T sprocket

- This one is close to working, but there isn't enough compression for the sorting position.

- It is important that the sorting position allow a ball to contact all three of the closest rollers (circled)

24T sprocket



30T sprocket

18T sprocket

24T sprocket

- For this configuration, I switched several of the sizes of the sprockets.

- There is still a bit more space around the sorting position than I would like, but it might work (I'll have to test it).

24 T sprocket

Final Configuration

project

designed by:

witnessed by: (with prototyping)

date: 1/11-12/21

Megan T 1/11/21

1/13/21

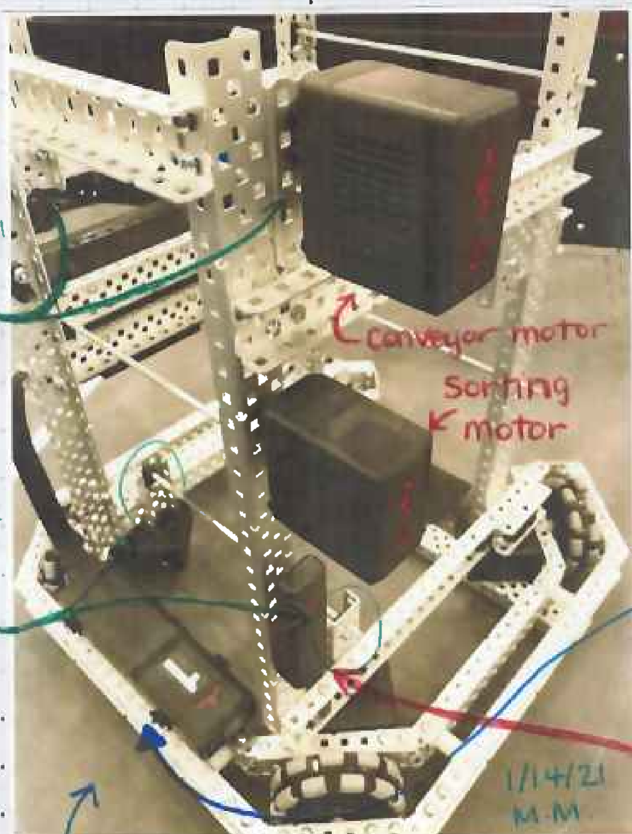
BUILDING THE CONVEYOR

GOAL: Attach the brain, battery, radio, motors, and necessary horizontal support bars / bearings / attachment c-channels to offset rollers from the vertical support bars, in order to test the sprocket configuration from 1/12/21 (see pg. 1). **GOAL COMPLETED 1/20/21**

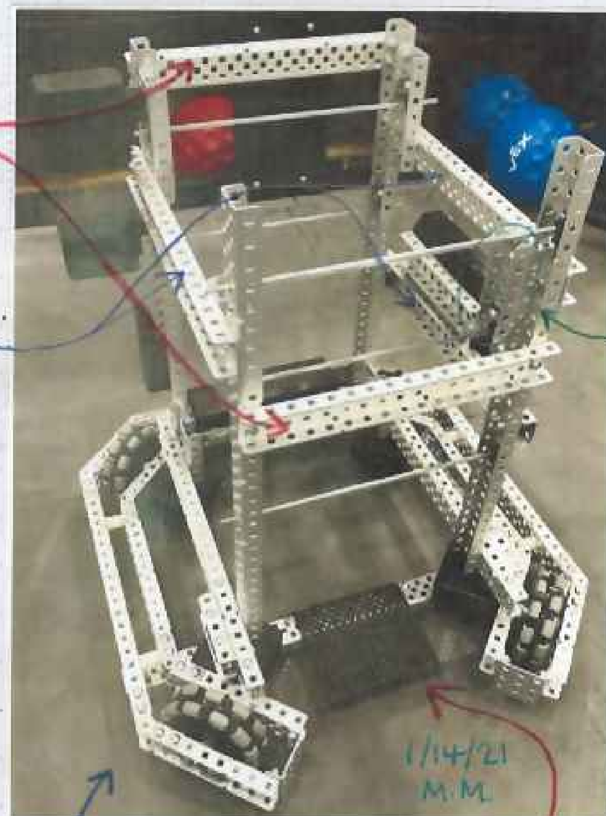
- The horizontal support bars across the front and back of the robot are 17 holes long.
- The horizontal support bars across the sides of the conveyor are 20 holes long.
- I attached the conveyor and sorting motors to the top two back rollers.

7 holes long c-channel to offset top back roller

5 holes long c-channel to offset bottom roller



Back view



7 holes long c-channel to offset front middle roller

Front view

This polycarbonate with anti-slip mat "floor" is for testing and will be finalized later.
↳ see page 5 - 1/20/21

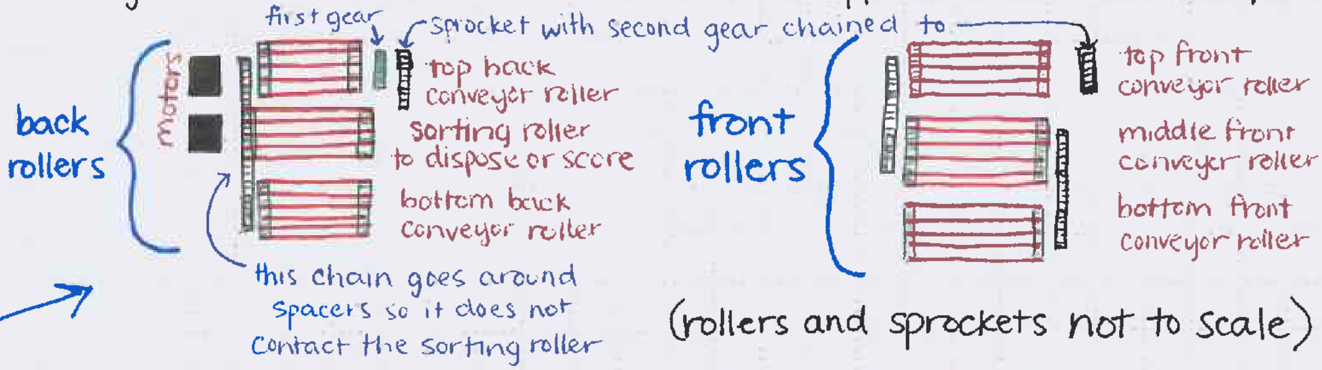
- I attached two battery clips directly to the back end of the base.
- The brain is attached on the opposite side as the motors for a more even weight distribution.
- I attached the radio here.

Megan M 1/14/21

1/15/21

BUILDING THE CONVEYOR CONTINUED

- In order to have one motor devoted to the single scoring/disposing roller, I need to have the remaining 5 rollers connected to one motor using chain / sprockets and gears to reverse the direction of rollers on opposite sides of the ball path.



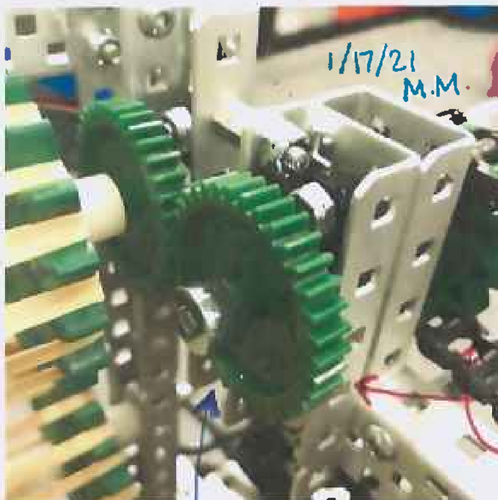
(rollers and sprockets not to scale)

l holes
ong
channe
offset
ront
iddle
roller

Chain layout

- I added the sprockets / chain and the gearing based on the layout above and the roller positioning prototype from 1/12/21 (see pg. 1)

- Now that the initial build of the conveyor is complete, I can test the flow of the ball.



Here are the gears which connect the top back conveyor roller to the top front conveyor roller.

These gears reverse the direction and this chain connects to the sprocket on the top front roller

36 T gears (low strength)

TESTING : I attempted to run the ball through both path (i.e., disposing or scoring) but hit some spacing issues that need to be addressed.

Problem #1 : As I suspected, there is too much space in the sorting position (see page 1).

Problem #2 : There is actually too much compression between the two top rollers, which will overheat the motor after repeated scoring.

project | designed by: | witnessed by:

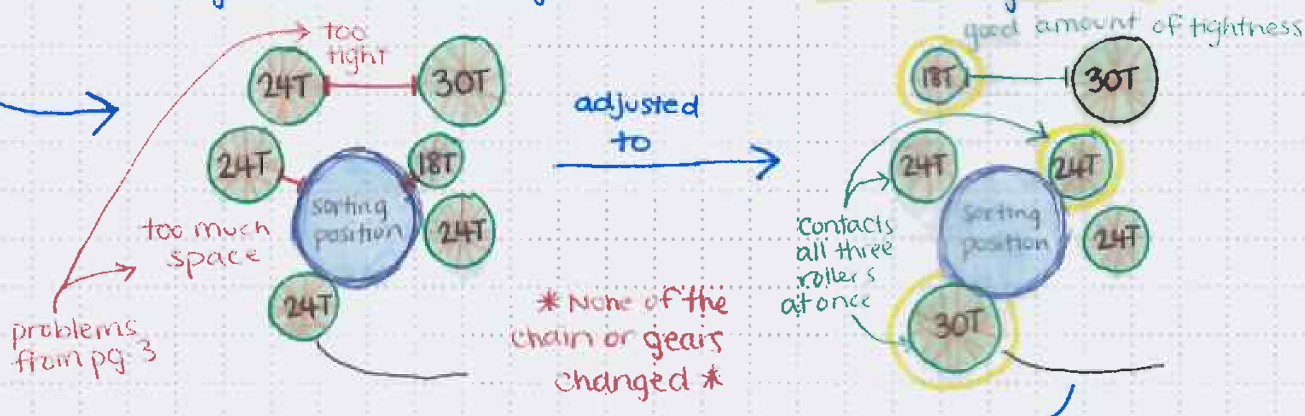
date: 1/13-20/21

Megan M 1/17/21

1/17/21

BUILDING THE CONVEYOR CONTINUED

- To solve the problems found in testing, (see page 3) I adjusted the configuration of the rollers on the conveyor.



TESTING THE NEW CONFIGURATION :

⟶ Both paths (scoring and disposing) flow well. Disposing has a good bit more compression than I'd like, but it gives the ball a big thrust across the field. The rubberband rollers must have rubberbands on every tooth (can't just do every other tooth) in order for there to be the correct amount of space in the sorting position.

The adjusted back rollers

1/20/21
MJM.

spacers to channel the path of the chain around the sorting roller

- Both the conveyor and sorter motors have 600 rpm cartridges
- I used 12T sprockets on all rollers, except the top front conveyor roller, which has 6T sprockets.

$600\text{rpm} \cdot \frac{12T}{6T} = \boxed{1200\text{ rpm}}$

1200rpm should be plenty fast for scoring.

Megan M 1/19/21

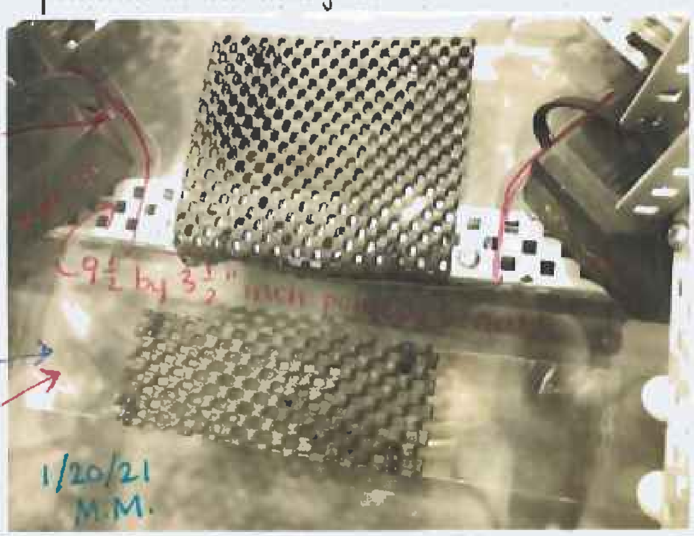
1/20/21

BUILDING THE CONVEYOR CONTINUED

- Spacing on each axel:
  - shaft collar
  - nylon washer
  - c-channel
  - flat bearing
  - nylon washer
  - $\frac{1}{4}$ inch spacer
  - 12T (or 6T) sprocket
  - nylon washer

then whatever size of sprockets for the x9 rubberband roller – with 9 $\frac{1}{2}$ inch spacers in between.
the other end of the axel is identical if there are two sprockets to connect chain. Otherwise, there's an extra spacer in place of a sprocket.

- I finalized the polycarbonate "floor" on the conveyor.



9 $\frac{1}{2}$ inch by 1 $\frac{1}{2}$ inch polycarbonate

The flexibility of the polycarbonate allows it to conform to the shape of the ball

→ Materials used to build conveyor, mount the brain, radio, etc. from 1/13-20/21: ←

- 1 - 9 $\frac{1}{2}$" by 3 $\frac{1}{2}$" polycarbonate (sections cut)
- 1 - 9 $\frac{1}{2}$" by 1 $\frac{1}{2}$" polycarbonate
- 3 - 1x2x1x20 aluminum c-channels
- 2 - 1x2x1x17 aluminum c-channels
- 1 - 1x2x1x10 aluminum c-channel
- 2 - 1x2x1x8 aluminum c-channels
- 2 - 1x2x1x7 aluminum c-channels
- 2 - 1x2x1x5 aluminum c-channels
- 1 - 1x2x1x6 aluminum c-channel
- 1 - 1x2x1x4 aluminum c-channel
- 1 - 1x2x1x3 aluminum c-channel
- 2 - pillow block bearings
- 4 - $\frac{1}{4}$ inch locking screws
- 2 - V5 Smart motors (600 rpm)
- 2 - battery clips
- 1 - V5 Brain
- 1 - V5 radio
- 6 - 12T sprockets
- 2 - 6T sprockets
- 2 - 18T sprockets
- 2 - 36T gears
- 6 - 24T sprockets
- 44 - nylon washers
- 4 - 30T sprockets
- 1 - $\frac{1}{8}$ inch spacer
- 4 - $\frac{1}{2}$ inch standoffs
- 11 - $\frac{1}{4}$ inch spacers
- 2 - 1 inch standoffs
- 14 - $\frac{3}{8}$ inch spacers
- 12 - flat bearings
- 73 - $\frac{1}{2}$ in. spacers
- 6 - axels (cut)
- 30 - 0.375 inch screws
- 14 - shaft collars
- 12 - $\frac{1}{2}$ inch screws
- chain
- 7 - 0.875 inch screws
- 2 - sections of anti-slip
- 6 - 1 $\frac{1}{4}$ inch screws
- 75 - rubberbands
- 5 - 1 $\frac{1}{2}$ inch screws
- 10 - 4" zipties
- 56 - thin nylocks

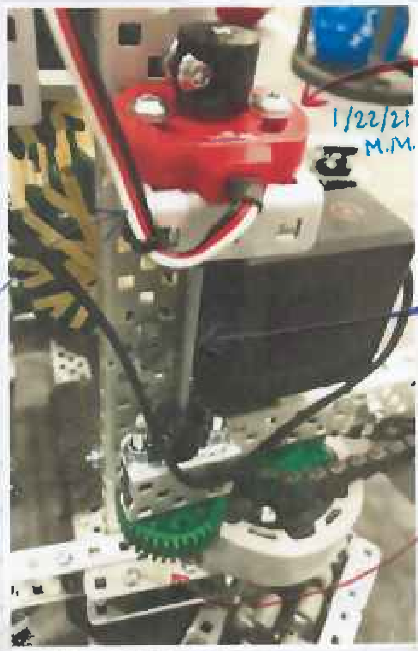
Megan M. 1/20/21

1/21/21

MOUNTING THE INTAKES

GOAL: Mount the intakes to the base and stabilize them against the conveyor. Add compression and stowing. **GOAL COMPLETED 1/23/21**

- I already built the intake mount on 1/2/21 (see Book One, page 128), so now I need to add support from the conveyor.



Potentiometer

long shaft that goes through mount on base

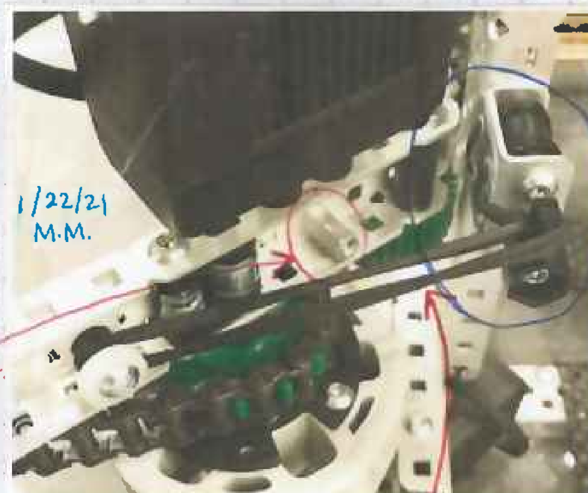
spacers to reach the correct height

1x3x1x3 aluminum c-channel

- As well as providing stability, the potentiometer allows me to know the angle of the intakes (opened, stowed, etc.)

- Having the intakes mounted to both the base and conveyor work very well. They seem very stable, much moreso than on the first two builds.

Intake Compression + Stopper



Stopper

- For a stopper, I mounted a 1x2x1x2 aluminum c-channel to the front vertical bars of the conveyor with polycarbonate as padding.

this standoff provides another point of contact with the stopper

- I lowered the 4" front flex wheels with $\frac{5}{8}$ inches in spacers to be at a better height for descoring and collecting balls.

3" loop of latex doubled

A rubberband would not be as effective here due to its size.

project

designed by:

witnessed by:

date: 1/21-23/21

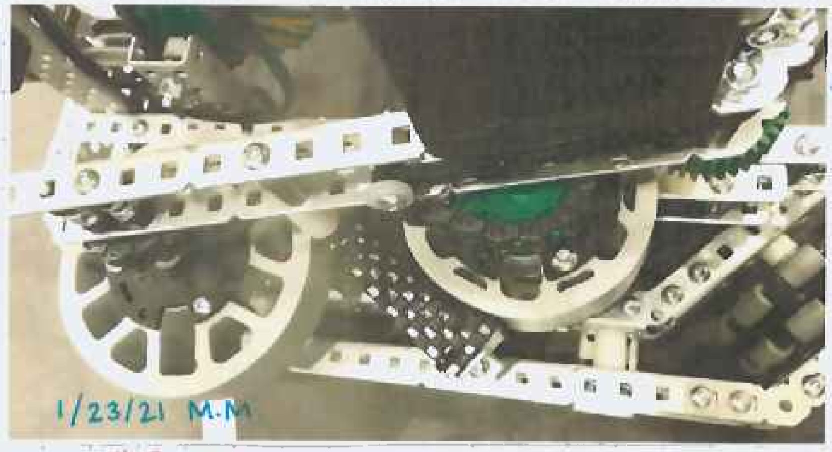
Megan M 1/21/21

1/23/21

MOUNTING THE INTAKES CONTINUED

Intake Stowing →

- I used the method of stowing that I used for the first two designs again here.
- With both intakes stowed, the width of the robot from 4" flex wheel to 4" flex wheel is ≈ 17.5 in. so it fits in the size limit!



right intake stowed

The 3" flex wheel rolls onto a 1x3x1x5 c-channel and a spacer ($\frac{1}{4}$ inch) on a screw holds it in place until the intake reverses.

→ Materials used to mount the intakes, stoppers, stowing: ←

- 2 - potentiometers
- 2 - 1x3x1x5 aluminum c-channels
- 2 - 1x3x1x3 aluminum c-channels
- 2 - 1x2x1x2 aluminum c-channels
- 2 - shafts (cut)
- 4 - 1 inch standoffs
- 2 - $\frac{1}{2}$ inch standoffs
- 2 - 1-post hex nut retainers
- 2 - flat bearings
- 8 - clamping shaft collars
- 10 - nylon washers
- 12 - 4" zipties
- 2 - 1" by 1" polycarbonate
- 2 - sections of anti-slip mat
- 4 - $\frac{1}{8}$ inch spacers
- 6 - $\frac{1}{4}$ inch spacers
- 8 - $\frac{3}{8}$ inch spacers
- 6 - $\frac{1}{2}$ inch spacers
- 2 - 8 mm shaft spacers
- 10 - $\frac{1}{2}$ inch locking screws
- 12 - 0.375 inch screws
- 4 - 0.875 inch screws
- 2 - $\frac{1}{2}$ inch screws
- 2 - 1 inch screws
- 4 - $1\frac{1}{4}$ inch screws
- 2 - $1\frac{3}{4}$ inch screws
- 26 - thin nylocks
- 2 - 3" loops of latex

project designed by: witnessed by:

date: 1/21-23/21

Megan M 1/23/21

1/25/21

HOOD, LICENSE PLATES, BUMPER

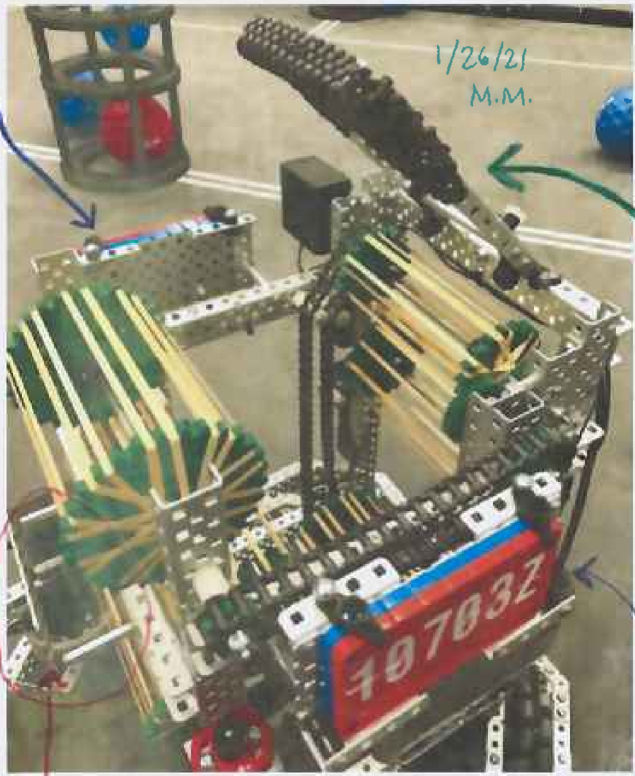
GOAL: Finish building (except for sensors!) by adding a hood, attaching the license plates, and adding a bumper to align with goals.

GOAL COMPLETED 1/27/21

- I'm using the same license plate holders from 9/30/20 (see Book 1, pg. 71) attached the same as on 11/22/20 (see Book 1, pg. 108).

- I attached a 4x9 aluminum plate on 2" stand off with an $\frac{1}{8}$ inch spacer to the front horizontal conveyor support.
  ↳ this helps me line up with goals both in driver and autonomous
  plus, if a goal is outside the legal tolerance, the bumper can push it up enough to descore.

First Hood Design: first prototype



bumper

license plate holder

The hood is attached with pillow block bearings and is 15 holes long

→ Materials used on the bumper: ←
- 1 - 4x9 aluminum plate
- 3 - 2" standoffs
- 6 - $\frac{1}{2}$ inch locking screws
- 3 - $\frac{1}{8}$ inch spacers

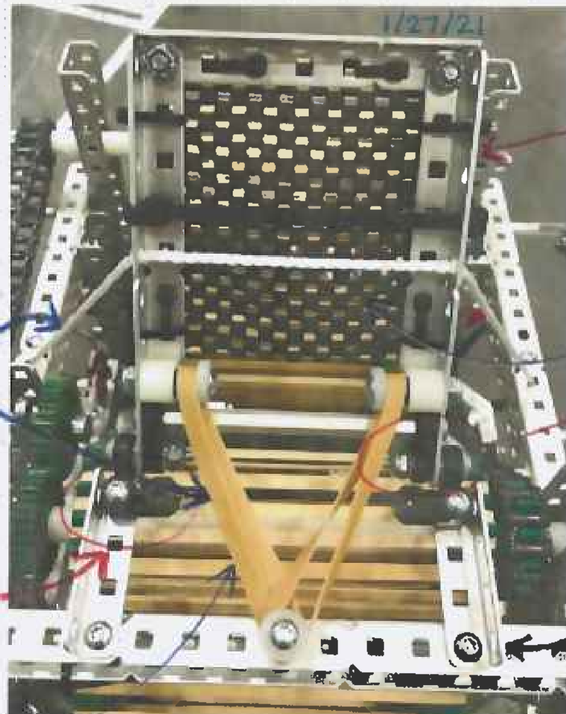
Testing the Hood: The hood needs to be at a higher angle to score consistently, but in the current location, the ball does not contact the hood if it is at a higher angle.

Megan M[redacted] 1/25/21

1/26/21

HOOD, LICENSE PLATES, BUMPER CONTINUED

FIXING THE HOOD: In order to move the hood further inward (on top of the back 18T rubberband roller) so the ball contacts it all the way out, I attached the hood to two 4-holes long aluminum angles.



rubberbands for compression upward, nylon string for a stop (keeps the hood from going higher than the right angle for scoring)

These flat bearings act as a stop to prevent the hood from folding down too far.

The preload pushes the hood up when scored and the tension from the rubberbands keep it up.

I shortened the aluminum frame to 10 holes long from 15 holes.

anti-slip mat for traction

The hood pivots and folds down to fit in the size limit

attached to the back horizontal support bar for the conveyor

Improved Hood

→ Materials used on the hood: ←

- 2 - 1x1x10 aluminum angles
- 2 - 1x1x4 aluminum angles
- 1 - 1x1x7 aluminum angle
- 4 - flat bearings
- 2 - pillow block bearings
- 1 - 3" standoff
- 2 - post hex nut retainer w/ flat bearing
- 6 - 0.375 inch screws
- 5 - 0.625 inch screws
- 4 - 1 inch screws
- 6 - nylon washers
- 3 - 8 mm shaft spacers
- 3 - $\frac{1}{8}$ inch spacers
- 2 - $\frac{3}{8}$ inch spacers
- 13 - thin nylocks
- 8 - 4" zipties
- 2 - 11" zipties
- 2 - rubberbands
- 1 - section of $\frac{1}{8}$" nylon braided rope

TESTING THE HOOD:

The higher angle is much better. Unfortunately, a ball cannot be scored consistently from position 3 (see Book One, page 124). Due to the shortage of time until the next competition, I plan to only utilize positions 1 and 2 for now. I will definitely revisit this later.

Megan M. 1/27/21

1/28/21

ADDING SENSORS:
- inertial sensor, distance sensors,
- optical sensor, limit switch

GOAL: Attach sensors that will be used for driver macros and / or autonomous functions (i.e., scoring, intaking, disposing)

GOAL COMPLETED 1/29/21

INERTIAL SENSOR:

- This will help with accurate turning in autonomous.

1x2x1x4 c-channel attached to the base support bar (see pg. 127, Book One)

→Materials used to mount inertial sensor:←
- 1 - V5 inertial sensor
- 2 - 0.375 inch screws
- 1 - 0.875 inch screw
- 1 - 1x2x1x4 aluminum c-channel
- 1 - nylon washer
- 3 - thin nylocks

LIMIT SWITCH:

sorting roller

bottom back conveyor roller

- When the limit switch is pressed, it indicates that a ball is being disposed.
- I can use this as the signal for the motors to stop running after disposal.

attached to the c-channel that supports the Brain

→Materials used to mount limit switch:←
- 1 - limit switch
- 1 - 2x5 aluminum plate
- 2 - nylon washers
- 2 - $\frac{1}{4}$ inch spacers
- 2 - $\frac{1}{2}$ inch screws
- 2 - 0.625 inch screws
- 4 - thin nylocks

project

designed by:

witnessed by:

date: 1/28-29/21

Megan M 1/28/21

1/29/21

ADDING SENSORS CONTINUED

DISTANCE SENSOR:

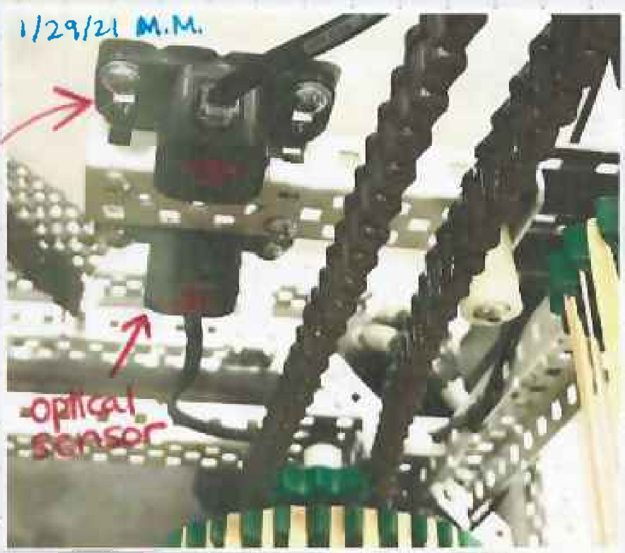
mounted to the front conveyor support bar

- This distance sensor is mounted behind the bumper added 1/25/21 (see page 8).
- It detects whether or not there is a ball in position one and can work with the potentiometers (see page 6) to determine an exact ball positioning.

→ Materials used to mount distance sensor in position one: ←

- 1 - 4x4 aluminum plate
- 1 - V5 distance sensor
- 2 - nylon washers
- 2 - 0.375 inch screws
- 2 - ½ inch screws
- 4 - thin nylocks

DISTANCE SENSOR AND OPTICAL SENSOR:



distance sensor

- I mounted a second distance sensor as well as an optical sensor to detect balls in the sorting position (position 2).

→ Materials used to mount distance sensor and optical sensor in pos. 2: ←

- 1 - V5 distance sensor
- 1 - V5 optical sensor
- 4 - nylon washers
- 2 - ½ in. spacers
- 8 - thin nylocks
- 1 - 1x2x1x11 alum. c-chan.
- 1 - 4x6 alum. plate
- 2 - 0.375 in. screws
- 4 - ½ in. screws
- 2 - 0.875 in. screws

On 10/3/20 (see pg. 74, Book One), I tested the optical sensor and found that it doesn't work at a far distance, but is fairly consistent at a close range. I hope to use this for auto-sorting.

project designed by: witnessed by:

date: 1/28-29/21

Megan M 1/29/21

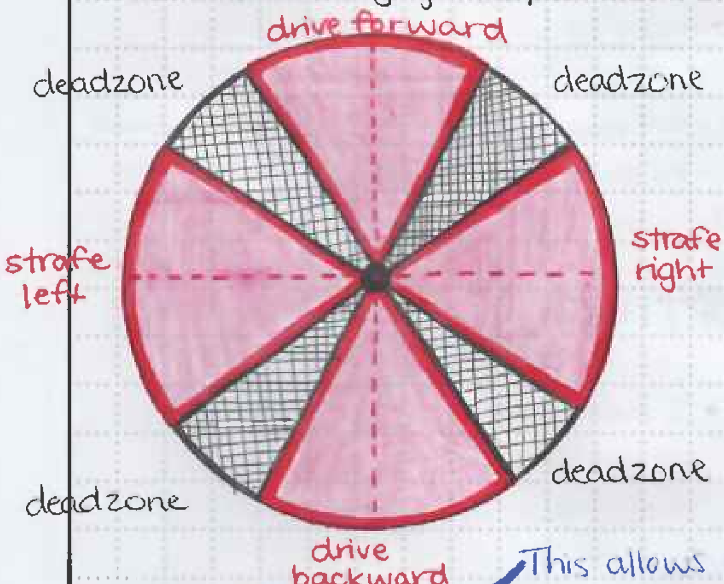
1/29/21

PROGRAMMING USERCONTROL

GOAL: Program manual controls for driver, as well as macros for intaking, scoring, and descoring; use auto-sorting and have toggle options for auto-sorting / manual. **GOAL COMPLETED 2/1/21**

Drivetrain movement:

- I want the horizontal axis on the left joystick to control turning left/right, and the right joystick to control straight forward / backward driving and strafing left / right on the vertical and horizontal axes, respectively.
- I found in testing that it is necessary to set zones for different motions on the same joystick, as the axes overlap.



- I wrote this code to determine what zone is being triggered (driving or strafing)
- I use the deadzone to mathematically if the joystick position is within the set span of one axis or not.
- Turning has priority over driving and strafing.

This allows me to set the span of each axis

```
// Turning may be active at the same time as either driving or strafing
if(Controller.Axis4.position()) {
  drivetrainArray[1] /* turnActive */ = true;
}

int deadzoneRange = 20;
int joystickPosition = abs(Controller.Axis2.position()) -
abs(Controller.Axis1.position());

// Determine if driving, strafing, or neither is active
if(abs(joystickPosition) <= deadzoneRange) {
  drivetrainArray[3] /* deadzone */ = true;
}
else if(joystickPosition > 0) {
  drivetrainArray[0] /* driveActive */ = true;
}
else if(joystickPosition < 0) {
  drivetrainArray[2] /* strafeActive*/ = true;
}
```

1/29/21
M.M.

This is a part of my flagging system which prevents the possibility of conflicting commands

This code makes a big difference!

project

oy:

date: 1/29 - 2/1/21

Megan M[redacted] 1/29/21

1/29/21

PROGRAMMING USERCONTROL CONTINUED

Ball Positioning Evaluation:

- Using the distance sensors at position 1 and position 2 (see page 11), I am able to determine where, if any, balls are in the robot.

```
int evaluatePosition(int positionArray[3]) {
  if(DistancePos1.objectDistance(mm) <= distancePos1Unoccupied) {
    positionArray[1] /* positionOneOccupied */ = true;
  }
  else {
    positionArray[1] /* positionOneOccupied */ = false;
  }
  if(DistancePos2.objectDistance(mm) <= distancePos2Occupied) {
    positionArray[2] /* positionTwoOccupied */ = true;
  }
  else {
    positionArray[2] /* positionTwoOccupied */ = false;
  }
  positionArray[0] /* needs to evaluate */ = true;
  return positionArray[3];
}
```

constants for the distance detected when no ball is in the robot

1/29/21
M.M.

- I found that the floppyness of the intakes, not to mention the poles on the middle goal, make the potentiometers not very consistent at identifying where a ball is, but I hope to use these specifically for the middle goal in the future.

- I can use the knowledge of what positions are occupied for more versatile macros.

  Collect/Intake: bring a ball to the furthest available position

  Score/Descore: score a ball if already in robot, otherwise descore until you can score a ball.

- Using this evaluation code along with automatic sorting code (see pg. 14) opens a lot of possibilities for macros and autonomous routines.

1/30/21

PROGRAMMING USERCONTROL CONTINUED

Automatic Sorting:

- I use the flags autoSort and redAlliance to toggle disposing red, disposing blue, and disabling auto-sorting macros (manual controls only).

```
void autoSortToggle() {
  if(autoSort == false) {
    redAlliance = true;
    autoSort = true;
    Controller.Screen.setCursor(4, 1);
    Controller.Screen.print("RED            ");
  }
  else if(autoSort == true) {
    if(redAlliance == true) {
      redAlliance = false;
      Controller.Screen.setCursor(4, 1);
      Controller.Screen.print("BLUE           ");
    }
    else if(redAlliance == false) {
      autoSort = false;
      Controller.Screen.setCursor(4, 1);
      Controller.Screen.print("OFF            ");
    }
  }
  wait(200, msec);
}
```

1/30/21
M.M.

this wait time decreases the sensitivity of the toggle button

- In pre-auton, the optical sensor evaluates the color of the preload to determine what color to sort out (dispose the opposite alliance's color).

```
void identifyColor() {
  int ballColor = Optical.color();
  bool colorIdentified = false;
  int numberOfChecks = 0;

  while(numberOfChecks <= 20) {
    wait(10, msec);
    ballColor = Optical.color();
    if(ballColor == red || ballColor == blue) {
      colorIdentified = true;
      break;
    }
    numberOfChecks += 1;
  }

  if(colorIdentified == true) {
    if(ballColor == red && redAlliance == true) {
      dispose = false;
    }
    else if(ballColor == blue && redAlliance == true) {
      dispose = true;
    }
    else if(ballColor == red && redAlliance == false) {
      dispose = true;
    }
    else if(ballColor == blue && redAlliance == false) {
      dispose = false;
    }
  }
  else {
    dispose = true;
    Controller.rumble("-");
  }
  Optical.setLight(ledState::off);
}
```

1/30/21
M.M.

Over $\frac{1}{5}$ of a second, the color is checked up to 20 times if needed

Disposing or scoring is decided based on the color of the ball and the flags I have set

if the color cannot be identified within 200 msec, default to dispose that ball, and vibrate the controller to alert me

- The optical sensor works consistently at a close range, especially when the LED light (built into the sensor) is on at 25%.

- My sorting code worked 9 out of 10 runs, where the one failed to evaluate in time and defaulted to disposing.

witnessed by:

date: 1/29 - 2/1/21

Megan M[redacted] 2/1/21

2/1/21

PROGRAMMING USERCONTROL CONTINUED

Macros: collectBall, scoreBall, descoreBall:

- The combination of my evaluatePosition function (see page 13) and my identifyColor function (see page 14) allows for accurate driver macros.

- For my collectBall function, I have code for each of the four possible situations:

false, false	false, true	true, false	true, true
(both positions are empty)	(only position 2 is occupied)	(only position 1 is occupied)	(both positions are occupied)
↓	↓	↓	↓
bring the new ball to position 2, check for disposing, either dispose or end function.	check for disposing, either dispose and bring the new ball up to position 2 (check for disposing again), or bring new ball to position 1 and keep initial ball in position 2.	bring the ball up to position 2 and check for disposing, either dispose or end function.	check for disposing, either end the function or dispose and bring the ball from position 1 up and check for disposing again.

this works for picking ball up off the field but also descoring as well

These two possibilities don't actually collect a new ball, just adjust the current ones in the robot

- My scoreBall function and descoreBall function are simpler than collectBall because they only use position 2 to determine when to stop the motors, so there are only two options each (true or false).

- I found that for intaking and disposing, using the limit switch and distance sensor data will trigger actions too soon (such as stopping the intakes before the ball reaches the conveyor or is disposed).

```
// encoder constants for scoring, intaking, and disposing (used alongside other
sensors)
double encoderDegreesToScore = 1200; // in degrees
double encoderDegreesToIntake = 180; // in degrees
double encoderDegreesToDispose = 720; // in degrees
```

I use these sensors as the signal to turn the motor this specific degree constant to ensure the action is completed

Megan T[redacted] 2/1/21

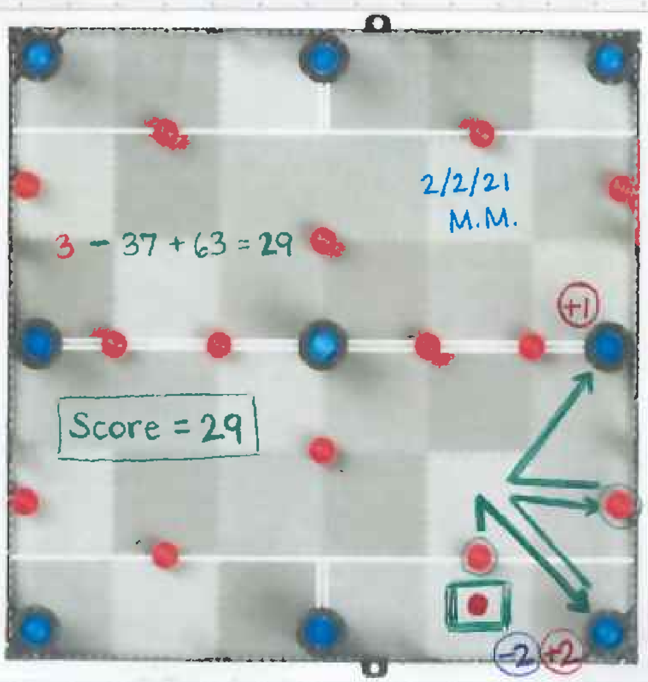
 2/2/21

PROGRAMMING AUTONOMOUS

GOAL: Program an autonomous skills routine and 15 second autonomous routines for one and two goals that utilize as much of the code used in the macros (see pages 13-15) as possible.

GOAL COMPLETED 2/3/21

Autonomous Skills:



- Here are the paths for the autonomous routines.
- The autonomous skills routine uses the collectBall and scoreBall with slight adjustments, as well as evaluatePosition and identifyColor (see pages 13-15).
- I hope to add more to this routine and score a third goal to complete a row after the competition on Feb. 6th.

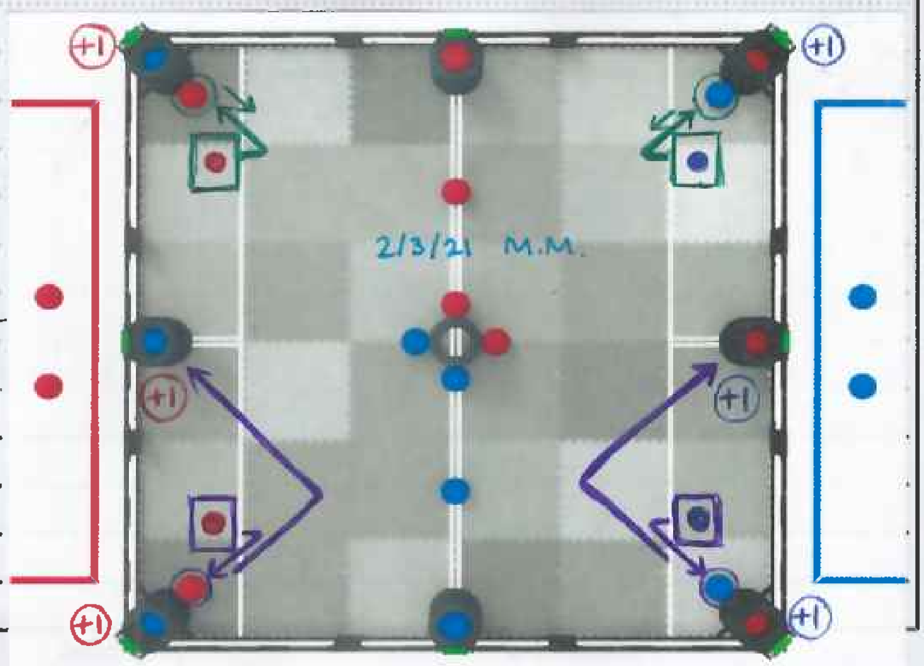
15-Second Autonomous:

- All of the starting positions for the 15-sec. routines are the same – the robot faces the field perimeter

this saves time with a shorter turn

one goal (left and right)

two goals (left and right)



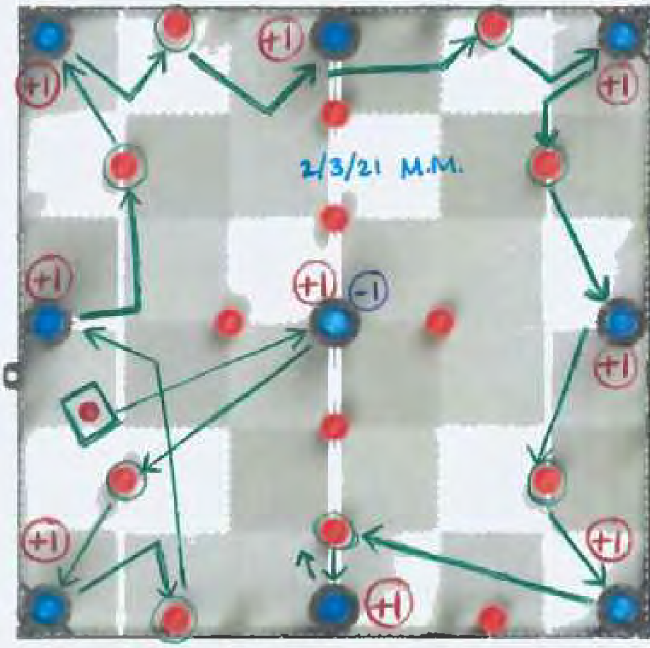
project

date: 2/2-3/21

Megan M 2/3/21

2/3/21

DRIVER SKILLS PATH & PRACTICE



- Here is my strategy for driver skills. For now, I'm aiming to score one ball in each goal, but I hope to add descoring later.
- If I can score one red in each goal (requires descoring one from the center goal), it would be 106 points.

57 - 14 + 63 = $\boxed{\textbf{106} \text{ points}}$

	Total Score	Rows Owned
1.	**82**	4
2.	**81**	4
3.	**80**	4
4.	**78**	3
5.	**96**	6
6.	**94**	6
7.	**110**	8
8.	**80**	4
9.	**93**	6
10.	**81**	4

2/3/21 M.M.

	Total Score	Rows Owned
11.	**107**	8
12.	**95**	6
13.	**94**	6
14.	**93**	6
15.	**93**	6

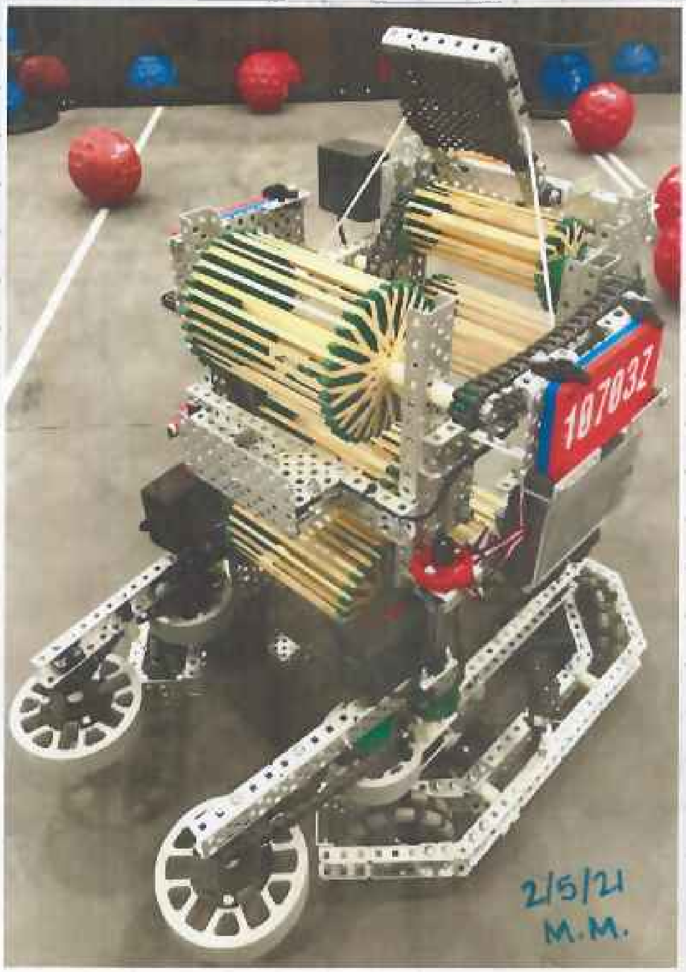
2/4/21 M.M.

- Even though I was only able to successfully score all nine goals twice, there were a number of times that I was really close and would have scored that last goal with one or two more seconds

Megan M 2/4/21

2/5/21

10703Z for the FRA Competition



SUCCESSES:

- Completing build #3 with stable and functional intakes, a good conveyor flow, and minimal problems
- Ability to score and descore all goals with macros
- Automatic sorting and ball position evaluation code
- One and two goal autonomous routines and autonomous skills

THINGS TO IMPROVE:

- Three ball positions
- Higher scoring autonomous skills
- More driving practice

MY GOALS FOR THE FRA COMPETITION:

- ☑ Get a second interview. [Completed 2/6/21]
- ☒ Rank in the top half after qualifications.
- ☑ Score at least 106 points in Driver Skills.
- ☒ Score 29 points in Autonomous Skills

project

designed by:

witnessed by:

date: 2/5/21

Megan M 2/5/21

2/6/21

FRA COMPETITION ANALYSIS

10703Z Stats:

- Ranked 13th after qualifications
- Ranked 7th in skills, with a total score of 134
- 2 wins, 5 losses in qualifications

- WP = 7
- AP = 30
- SP = 73

3 WP from completing the home row in autonomous

- OPR = 6.7
- DPR = 9.7
- CCWM = -3.1

FRA Skills Results 2/6/21 M.M.

Driver / Auton	Score	Analysis
Driver run #1	68	I got hung up on the center goal. Completed 2 rows, scored 5 red balls, descored 3 blue balls
Driver run #2	80	Still took too long on center goal. Completed 4 rows, scored 7 red balls, descored 1 blue ball
Driver run #3	$\boxed{106}$	Much faster with center goal. Changed path to score corner goal before center goal. Completed all 8 rows, scored 9 red, descored 1 blue
Auton run #1	22	Scored the first two red and descored two blue correctly. Did not drive forward enough to score second goal.
Auton run #2	$\boxed{28}$	Much better than auton run #1. Perfect, except descored one red ball when backing up from first goal.
Auton run #3	28	Same as auton run #2. Pulled red ball out of corner goal.

Megan M 2/6/21

2/6/21

FRA COMPETITION ANALYSIS CONTINUED

Autonomous Routines Success Rates:

- Left side, 2 goals: used 1 time
  - fully worked: 100%
  - partially worked: 0%
  - completely failed: 0%
- Right side, 2 goals: used 5 times
  - fully worked: 80% (4 times)
  - partially worked: 20% (1 time)
  - completely failed: 0%
- Left side, 1 goal: used 2 times
  - fully worked: 100%
  - partially worked: 0%
  - completely failed: 0%
- Right side, 1 goal: used 3 times
  - fully worked: 100%
  - partially worked: 0%
  - completely failed: 0%

FRA Qualification Match Results 2/6/21 M.M.

Alliance Partner	Opponents	Win / Loss	Score	Analysis
663E	98709A 98709C	Loss	23-16	Left, one goal auton scored 1 point poor communication and strategy good center goal scoring
97934Y	97934Z 9364C	Loss	41-4	Right, two goal auton scored 1 goal (just missed corner) decent strategy too slow poor center goal scoring
663C	9364A 9364Y	Loss	16-9	Left, one goal auton. scored 1 point decent strategy too slow good center goal scoring
663B	9364X 24816V	Win	19-10	Right, two goal auton scores 2 goals good strategy and communication good center goal scoring
97934X	97934U 92715A	Win	53-6	Left, two goal auton. scored 2 goals good strategy and communication good rows
37167R	96504R 663A	Loss	17-13	Right, two goal auton scored 2 goals decent strategy poor center goal scoring
96504P	2775J 92715A	Loss	17-15	Right, two goal auton scored 2 goals poor strategy and communication poor center goal scoring

Megan M - 2/6/21

2/6/21

2/6/21

FRA COMPETITION ANALYSIS CONTINUED

FRA Elimination Match Results 2/6/21 M.M.

Alliance Partner: 2775J

Opponents	Win / Loss	Score	Analysis
37167R 37167H	Win	52-7	R16 #2-1 Right, 2 goal auton scored 2 goals good strategy and communication all rows but one goal owned
98709C 98709A (seed #1)	Win	41-8	QF #1-1 Right. 1 goal auton scored 1 goal good strategy and communication good center goal scoring
663A 24816V (seed #4)	Win	21-11	SF #1-1 Right, 1 goal auton scored 1 goal good strategy and communication good center goal scoring
663B 663C (seed #3)	Loss	37-10	Final #1-1 Right. 1 goal auton scored 1 goal good strategy and communication too slow poor center goal scoring

- I won the Design award and Tournament Finalist!

WHAT I LEARNED:

- **Strategy can beat speed**, especially when you get the autonomous bonus. Neither my robot nor 2775J's robot are fast, but we were able to defeat the 1st seed and the 4th seed (both of whom are very fast) with strategy.

- **Guaranteed Win Strategies:** if you win the autonomous bonus, you only need to own a diagonal row to win the match. if you lose autonomous, you can still win if you own five goals in an L shape.

Megan N 2/6/21

 2/8/21

DESIGN CHANGES AFTER FRA COMPETITION

> PROBLEM TO ADDRESS / IMPROVEMENT NEEDED:

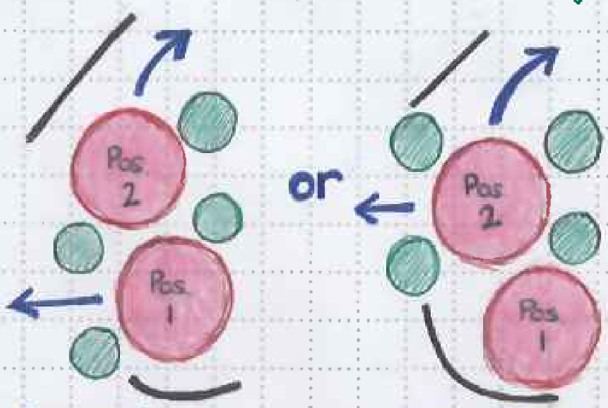
→ **Higher ball capacity** (while maintaining accurate scoring)

↳ this was anticipated (see page 9, testing the Hood results)

- Even being able to hold two balls in the conveyor would open many options for more efficient autonomous routines and driver skills path.

> BRAINSTORMING: Possible solutions to the problem:

① Two ball capacity with disposing



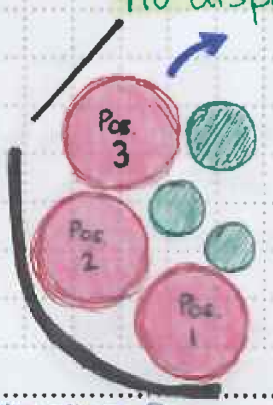
Dispose from position one

Dispose from position two

✓ plenty of space to work with
✓ potential for more efficient motor distribution

X not ideal to only have a two ball capacity in the conveyor
X requires fair amount of rebuilding

② Three ball capacity, no disposing

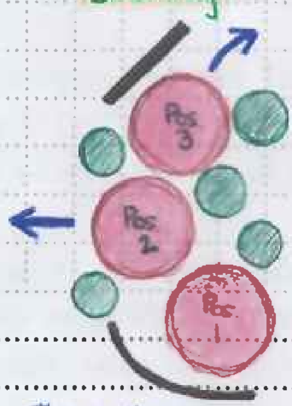


✓ good ball capacity
✓ more efficient motor distribution

X not ideal for programming
X requires a large amount of rebuilding

No rollers for disposing opens up more space for a higher ball capacity

③ Three ball capacity with disposing, less consistent scoring



✓ good ball capacity

X not reliable
X inconsistencies are not ideal for programming
X requires some rebuilding

This is the closest to what is currently built.

project | designed by: | witnessed by:

date: 2/8-9/21

Megan M[redacted] 2/8/21

2/9/21

DESIGN CHANGES AFTER COMPETITION CONT.

> SELECT OPTION:

Options to Increase Ball Capacity 2/9/21 M.M.

Criteria (least to most important)	Scale	① Two ball capacity with disposing	② Three ball capacity, no disposing	③ Three ball capacity with disposing, less consistent scoring
Few potential problems	0 to 3	3	1	0
Viable with space limitations	0 to 5	5	4	2
Ideal for macros, auton, and driving	0 to 5	4	2	3
Total Score:		12	7	5

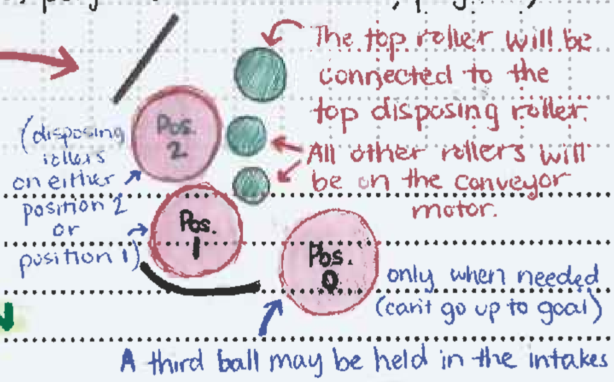
This is the best option as it does have a higher ball capacity while retaining the ability to dispose, consistent scoring, and flexible spacial options.

DESIGN PLAN: Build #3, Version #2

- My third design (see Book One, page 124) works well overall, so this version only modifies the conveyor and sorter to hold two balls instead of one.
- The base, x-drive, intakes, and hood will be the same as was built from 12/28/20 to 1/27/21 (see Book One, page 126 - Book Two, page 9)

NEW CONFIGURATION OF RUBBERBAND ROLLERS

- positions for two balls to sit far enough down to have force for scoring



MORE EFFICIENT MOTOR DISTRIBUTION

- more torque for scoring

Megan M 2/9/21

2/9/21

TIMELINE : UNTIL UPCOMING COMPETITIONS

2/9/21 M.M.

Building Changes
Conveyor, Other Improvements
2/10 - 2/13

today

February

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
~~14~~	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

Programming
Sensors, Macros, Autonomous, Skills
2/15 - 3/19

February

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
~~14~~	15	16	17	18	19	20
21	22	23	24	25	~~26~~	27
28						

March

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	~~12~~	13
~~14~~	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Driving Practice
Skills runs
2/22 - 3/19

February

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	~~26~~	27
28						

March

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	~~12~~	13
~~14~~	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

3/15-19/21
↑
I only have one week between my 4th and 5th competitions. I plan to make any programming changes needed but mainly practice.
~M.M. 2/9/21

I hope to be approaching the end of my building season and move into programming.
I need high scoring and consistent autonomous routines for State!
~M.M. 2/9/21

Calendar Key:

days with time to work on the robot

/ days I probably won't have time to work on the robot

tournament

DEADLINES:

- March 13 - Bolt Up in Chattanooga
- March 20 - SEMS St. Patrick's Day Competition in Bluff City

project

designed by:

witnessed by:

date: 2/9/21

Megan N 2/9/21

2/10/21

PROTOTYPING CONVEYOR CONFIGURATION

GOAL: Find a new configuration of rubberband rollers with positions for two balls and a smooth path for disposing and scoring.

GOAL COMPLETED 2/10/21

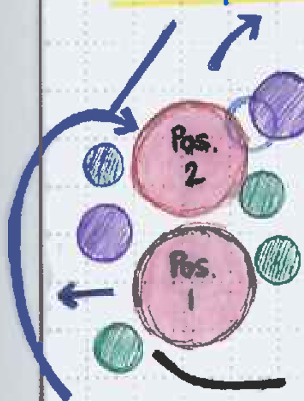
- With two balls in the conveyor instead of three, the spacing for the rollers is a lot more flexible.

- Based on where I attach the disposing rollers, the ball will be disposed from either position one or position 2.

rollers connected to the sorting motor

rollers connected to the conveyor motor

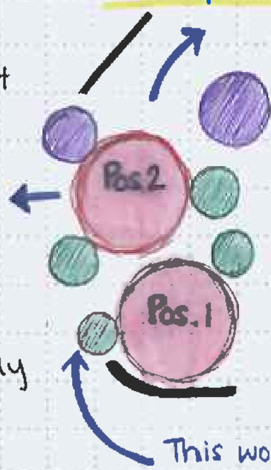
Option 1: Dispose from position one



✓ much more efficient if working ideally
✓ more conducive for driving / skills
✗ much harder to program
✗ balls will most likely hit mid-air
✗ potential problems with motor distribution

The ball in position 2 can not touch the top roller

Option 2: Dispose from position two



✓ much easier to program
✓ improved motor distribution
✓ less chance of balls hitting mid-air
✗ less efficient flow for auton. and driving

This would probably have to be either a stationary roller or the stationary sprockets used in build #2 (see Book One, pg 105).

- Neither of these options are able to score and dispose simultaneously due to the direction of the top roller when the top disposing roller runs in reverse (disposing).

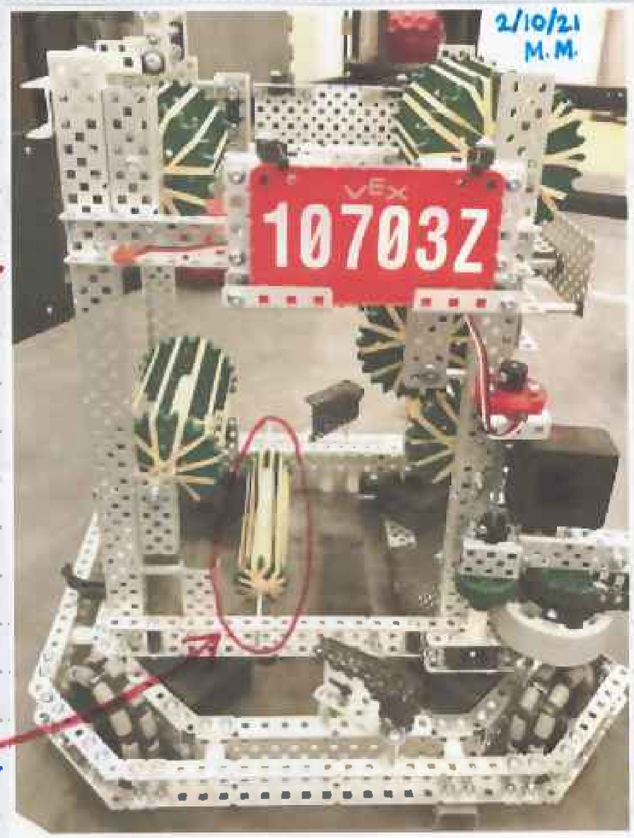
- Based on the pros and cons of disposing from position one or two, (see above) I have decided to choose Option Two: Disposing from position 2. It is more viable with fewer potential issues to work through.

Megan N 2/10/21

PROTOTYPING CONVEYOR CONFIGURATION CONT.

- In order to accurately prototype my chosen conveyor configuration, I took apart most of my initial design down to the supports.

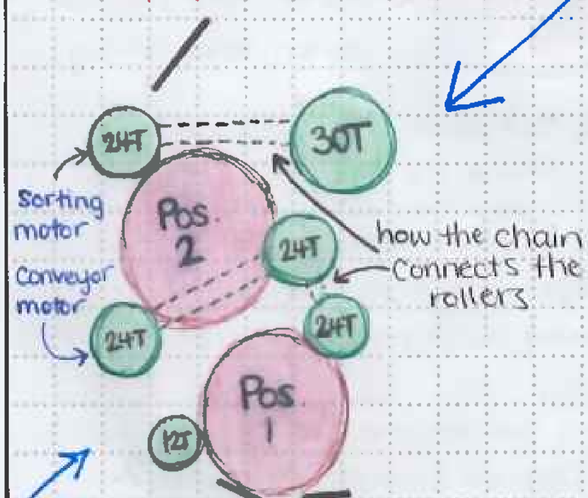
Here is the new arrangement →



Final Conveyor Spacing

The front three rollers ended up being the same as before (see page 4, new configuration), but all three back rollers changed.

This 12T roller is stationary, but I found that the ball hits the spacers, so the final build will just be the sprockets and no shaft.



Sprocket Sizes

- Depending on how easily the ball in position two is able to move up the conveyor to be scored, I may need to add more rubberbands (either every tooth, or double up on every other tooth).

- This configuration has nice resting positions for the balls - there is less compression which allows the ball to naturally come to each position.

project designed by: witnessed by:

date: 2/10/21

Megan M [redacted] 2/10/21

2/11/21

BUILDING NEW CONVEYOR CONFIGURATION

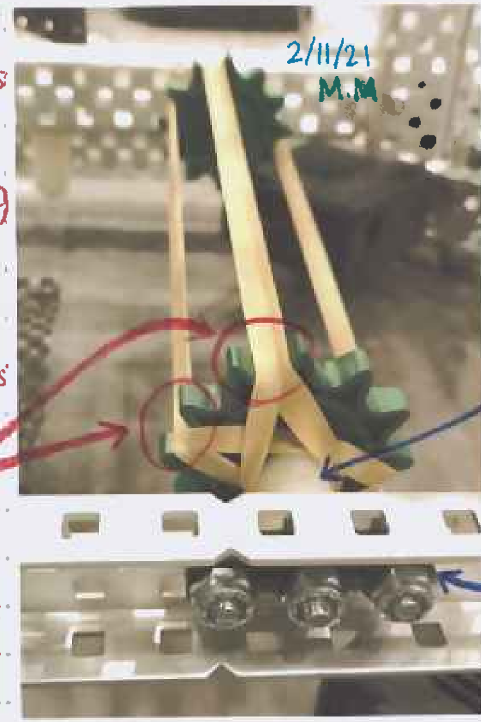
GOAL: Finalize the conveyor structure and rubberband rollers from the chosen configuration (see pages 25-26) with motors, gears, and sprockets and chain.

GOAL COMPLETED 2/12/21

- I attached this stationary set of 12T sprockets in place of the stationary axel from prototyping (see pg. 26) in the center row of holes on the horizontal support bar on the base, 8 holes from the back end (same location as in photo on pg. 26).

four rubberbands total (doubled on two places on the sprocket)

front of the robot



rubberbands spaced to contact the ball when needed without having too much compression on the base.

$1\frac{1}{2}$ in. screw

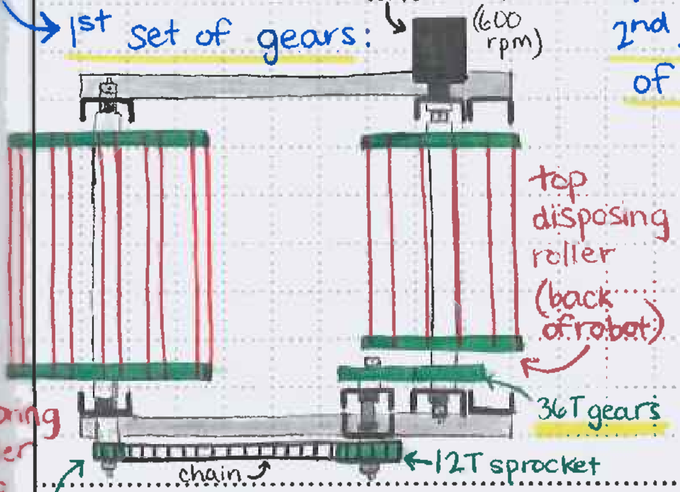
flat bearing for added stability

Stationary 12T sprockets (see page 26)

- Due to the new motor distribution (see pg. 25-26), I will need two sets of gears to reverse the direction of the front rollers (unlike version #1, see pg. 3).

1st set of gears:

sorter motor (600 rpm)



top disposing roller (back of robot)

scoring roller (top, front of robot)

36T gears

chain

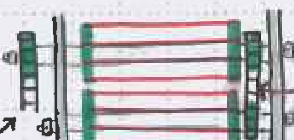
12T sprocket

6T sprocket for 1200 rpm

Top View (not to scale)

[see pg. 26 for roller sizes]

2nd set of gears:

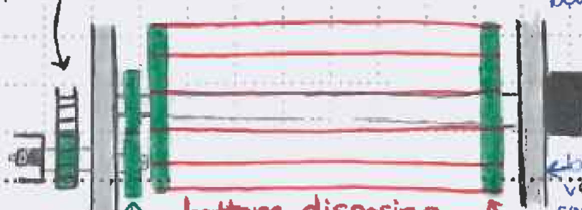


chain connecting bottom two conveyor rollers

front vertical conveyor support bars

12T sprockets

chain connects 12T sprockets here



conveyor motor (600 rpm)

back vertical conveyor support bars

bottom disposing roller (back of robot)

36T gears

Back View (not to scale)

project

designed by:

witnessed by:

date: 2/11-12/21

Megan N 2/11/21

2/12/21

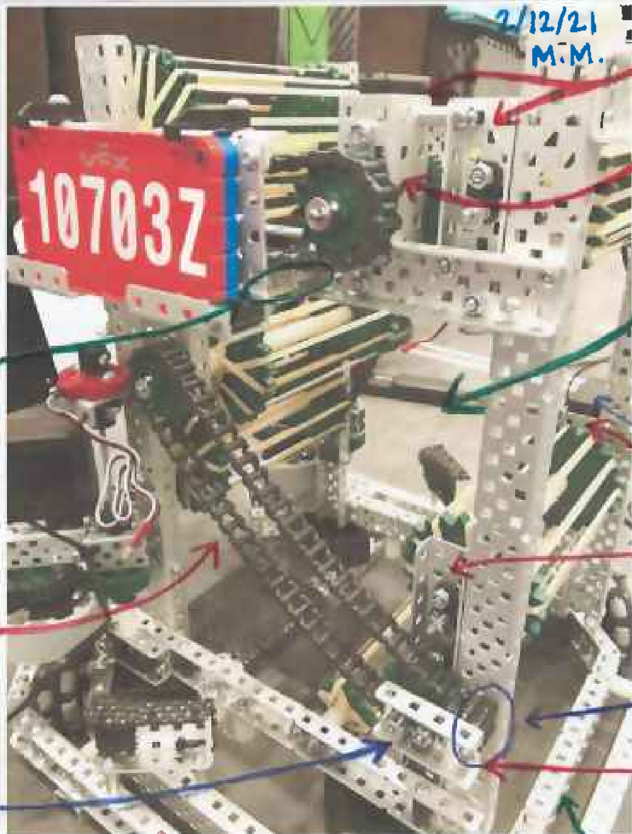
BUILDING NEW CONVEYOR CONFIGURATION CONT.

Finalized Conveyor

I had to move this standoff on the license plates holder diagonally down one hole to avoid the chain

This chain does not get in the way of the intake stowing

This c-channel provides a second point of contact for the axel and this minimizes slop significantly (same for top set of gears as well)



1x2x1x6 c-channels
attached to vertical and horizontal support bars

1x2x1x4 C-channel

This motor distribution requires the placement of the brain to be adjusted slightly (chain interferes).

conveyor motor

1x2x1x13 C-channels
attached to vertical and horizontal support bars

attached to vertical bar with screws and spacers

1x2x1x4 C-channel

The battery's position is unaffected

Materials used to build new conveyor configuration:

- 2 - V5 Smart motors (600 rpm)
- 2 - 1x2x1x13 aluminum c-channel
- 3 - 1x2x1x6 aluminum c-channel
- 2 - 1x2x1x4 aluminum c-channel
- 1 - $\frac{1}{2}$ inch standoff
- 7 - shafts (cut)
- 14 - flat bearings
- 14 - shaft collars
- 4 - 36 T gears
- 2 - 30T sprockets
- 7 - 12T sprockets
- 8 - 24T sprockets
- 1 - 6T sprocket
- 25 - nylon washers
- 3 - $\frac{1}{8}$ inch spacers
- 10 - $\frac{1}{4}$ inch spacers
- 10 - $\frac{3}{8}$ inch spacers
- 60 - $\frac{1}{2}$ inch spacers
- 10 - 0.375 inch screws
- 3 - 0.875 inch screws
- 10 - $\frac{1}{2}$ inch screws
- 4 - $1\frac{1}{4}$ inch screws
- 2 - $1\frac{1}{2}$ inch screws
- 29 - thin nylocks
- chain
- 40 - rubberbands

project

designed by:

witnessed by:

date: 2/11-12/21

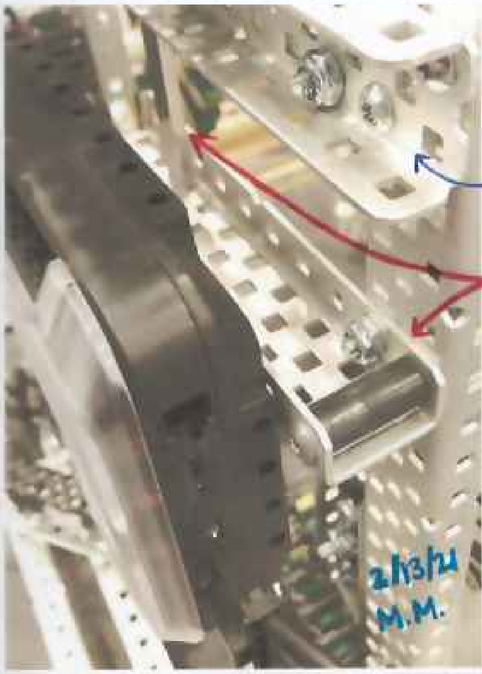
Megan W[redacted] 2/12/21

2/13/21

MINOR BUILDING IMPROVEMENTS

GOAL: Mount the Brain, radio, and battery clips. Make any final repairs or improvements needed before sensors / programming.

GOAL COMPLETED 2/13/21



This is a little higher and further toward the back of the robot than it was in version #1 (see pg. 2)

MOUNTING THE BRAIN:

this support bar can be seen in the photo on pg. 28

1x2x1x10 c-channel attached to back vertical support bar and attached with a 2" standoff to horizontal support bar

- I attached the Brain with a sideways c-channel to leave room for the chain on the conveyor to pass behind it.
- The Brain is on the opposite side as the conveyor / sorter motors.

this helps to have an even weight distribution

→ Materials used to mount Brain: ←

- 1 - V5 Brain
- 1 - 1x2x1x10 aluminum c-chan.
- 1 - 2 inch standoff
- 1 - $\frac{3}{8}$ inch spacer
- 1 - $\frac{1}{2}$ inch spacer
- 2 - $\frac{1}{4}$ inch locking screws
- 2 - $\frac{1}{2}$ inch locking screws
- 1 - 0.375 inch screws
- 1 - $1\frac{1}{4}$ inch screw
- 2 - thin nylocks

MOUNTING THE BATTERY CLIPS AND THE RADIO:

- I attached the battery clips at an angle to allow for easier access and better protection.
- I mounted the radio to the right back vertical support bar between the conveyor and sorter motors.

Battery clips re-attached to the back of the base, this time at a angle

Megan N 2/13/21

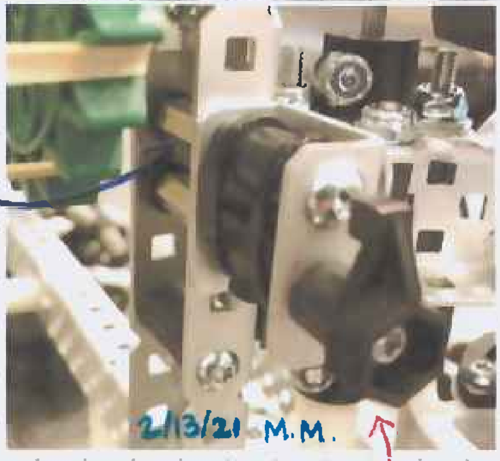
2/13/21

MINOR BUILDING IMPROVEMENTS CONTINUED

STABILIZING THE INTAKE STOPPERS:

(see page 6 for stoppers built initially):

$\frac{3}{4}$ inch standoff and $\frac{1}{8}$ inch spacer



Left intake stopper

- I noticed that the vertical support bars were bending from the force of the intakes unstowing, so I added standoffs to stabilize them.

→ Materials used to stabilize stoppers: ←

- 4 - $\frac{1}{2}$ inch screws
- 2 - $1\frac{1}{4}$ inch screws
- 4 - $\frac{1}{8}$ inch spacers
- 4 - $\frac{3}{4}$ inch standoffs

HOOD IMPROVEMENTS:

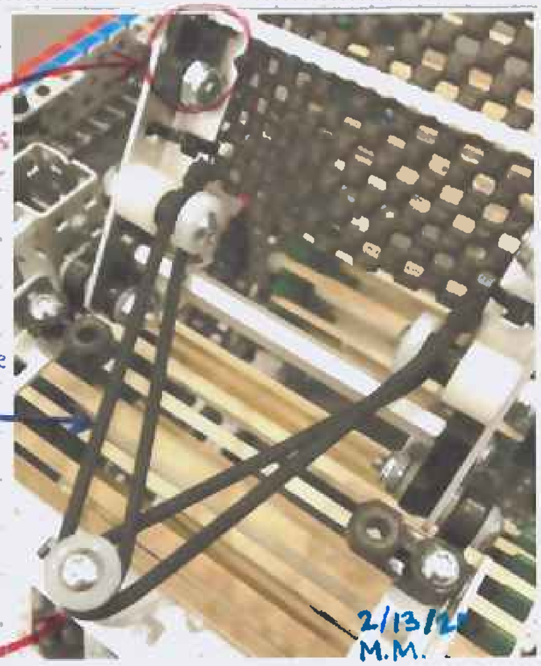
- The string was starting to tear, and it was caused by rubbing on the metal on the hood, so I added flat bearings for the string to pass through.

flat bearings for a smoother path for the string

Latex instead of rubberbands for slightly more compression

two 3" loops

I added a 4.6 mm spacer here to hold the latex better



This is the same hood from build #3, version #1 (see page 9)

→ Materials used to improve the hood: ←

- 2 - $\frac{1}{2}$ inch screws
- 1 - 0.875 inch screw
- 1 - 4.6 mm shaft spacer
- 2 - flat bearings
- 2 - thin nylocks
- 2 - 3 inch loops of latex

- I replaced the rubberbands used for compression with latex to last longer.

project

designed by:

witnessed by:

date: 2/13/21

Megan M 2/13/21

2/15/21

STRATEGY: SENSORS AND AUTONOMOUS

GOAL: Think through what sensors are needed for autonomous programming and for driver macros to accomplish the necessary functions successfully. **GOAL COMPLETED 2/15/21**

AUTONOMOUS ROUTINES TO WORK ON (15 sec.):

- ☐ one goal (corner goal) cycle, with and without scoring the center goal
- ☐ two goal (corner goal and middle goal on home row) cycle, with an option to score the center goal, if there's time in the 15 seconds.
- ☐ home row (score each of the three goals on the home row), with cycling, if there's time in the 15 seconds.
- ☐ various options to score the center goal and the side goal on the center row initially owned by the opposing alliance, with and without 1-2 goals on the home row ← I don't plan to work on these until the other 15 sec. routine options, as well as autonomous skills, are complete.

NEEDED DRIVER MACROS:

- ☐ collecting balls
- ☐ score one ball
- ☐ score two balls
- ☐ score one and descore one simultaneously
- ☐ unstowing the intakes (driver skills only)

FUNCTIONS NEEDED:

- ☐ three-ball positioning
- ☐ color identification for position 2 / auto-sorting
- ☐ acceleration / deceleration
- ☐ brain screen autonomous routine selector

SENSORS:

- ☐ ball positioning: 3 distance sensors, 1 of which is still attached from version #1 (see page 11) ← 2 new sensors
- ☐ automatic sorting: 1 optical sensor in position 2 ← 1 new sensors
- ☐ disposing: distance sensor for ball positioning can be used for this
- ☐ scoring: distance sensor mounted at the top of the robot ← 1 new sensor
- ☐ consistent turning in autonomous: inertial sensor, still attached (pg. 10)

potential additions if needed:

- line trackers to align with tape
- bumper or limit switches to line up with goals or field perimeter

2/16/21

ATTACHING SENSORS

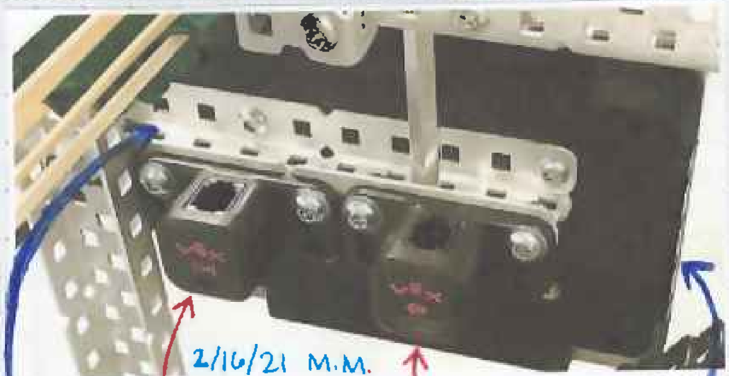
GOAL: Add the sensors needed to accurately perform all necessary tasks in autonomous and driver macros, as determined on 2/15/21 (see pg. 31) **GOAL COMPLETED 2/16/21**

POSITION TWO DISTANCE AND OPTICAL SENSORS:

see pg. 29

- The 10-hole long c-channel that provides a mount for the Brain is the perfect place to attach the two position two sensors, one distance sensor to detect if a ball is present, and one optical sensor to identify the color if present.

back of robot



Distance Sensor

Optical Sensor

10 holes long c-channel

V5 Brain

- It is important for the distance sensor to be closer to the back end of the robot, and the optical sensor to be further in
  - this allows the distance sensor to detect once a ball has been disposed
  - this way, the optical sensor sees the center of the ball rather than the edge (higher level of consistency)

Materials used to attach sensors:

all materials used for sensors on pgs 32-33

- 3 - V5 Distance Sensors
- 1 - V5 Optical Sensor
- 1 - 1x2x1x8 aluminum c-channel
- 8 - nylon washers
- 2 - $\frac{1}{2}$ inch spacers
- 8 - $\frac{1}{2}$ inch screws
- 2 - 0.875 inch screws
- 10 - thin nylocks

project

designed by:

witnessed by:

date: 2/16/21

Megan T 2/16/21

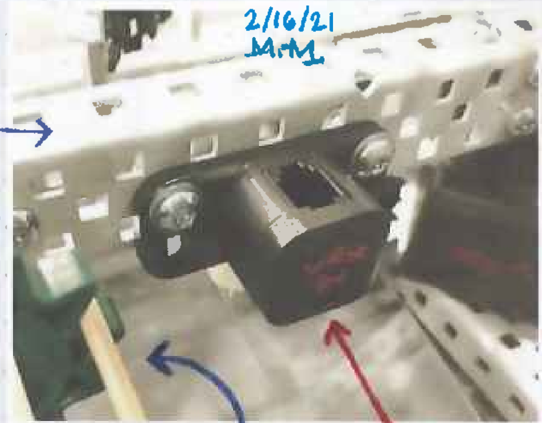
2/16/21

ATTACHING SENSORS CONTINUED

POSITION ONE DISTANCE SENSOR:

- The easiest location to mount the position one distance sensor is the horizontal support bar across each side of the base.
  - ↳ this is low on the ball, but it avoids the intakes (stowed) and doesn't require additional metal.

horizontal support bar; left side (see Book 1, pg 127)

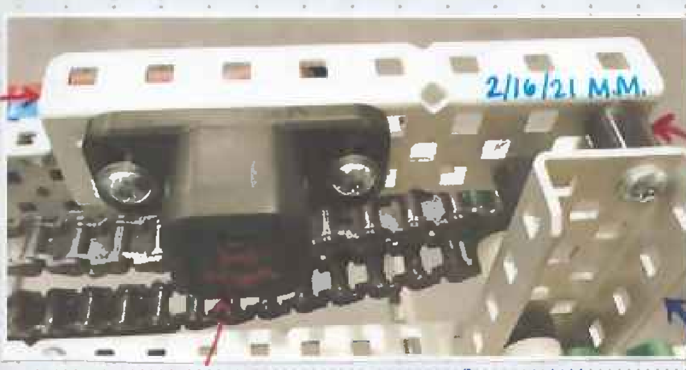


Stationary 12T sprocket (see pg. 27)

Distance Sensor

- This distance sensor is attached two holes over from the stationary 12T sprocket.

DISTANCE SENSOR FOR SCORING:



Distance Sensor

front left vertical conveyor support bar (see Book 1, page 128

- I mounted this distance sensor as high as possible without exceeding the size limit or requiring large amounts of metal to mount.

$\frac{1}{2}$ inch spacers with 0.875 inch screws

- Based on the position of this sensor, for scoring in autonomous, I will have to check that the sensor detects a ball on its way to being scored, but wait to stop the motors until the sensor no longer detects a ball (it would stop just before shooting the ball otherwise).

- I mounted the distance sensor upside-down for more conducive wiring.

Megan M 2/16/21

2/17/21

PROGRAMMING USERCONTROL

GOAL: Modify the driver control programming from 1/29 - 2/1 (build #3, version #1, see pgs. 12-15) to be faster, have more effective manual controls, and accommodate the increased ball capacity in version #2. **GOAL COMPLETED 2/26/21**

Increasing Drivetrain Speed:

- After researching and talking to teams at competitions including **97934U**, I decided to try to increase drivetrain speed by using volts instead of rpm.
  - The result: The drivetrain is definitely faster when using volts, but the Controller.Axis.value() function which allows for a range of speed (based on the joystick's position) only moves at the highest speed (12 volts) rather than the gradual increase / decrease with rpm.

I need the extra speed, but I also need the option to move at a slower velocity, so I set a threshold for whether to use volts (faster, but constant) or rpm (slower, but based on joystick position).

```
if(abs(Controller.Axis2.value()) < speedControlThreshold) {
   spin commands using velocityUnits::rpm
}
else {
   spin commands using voltageUnits::volt
}
```

2/17/21
M.M.

pseudocode for determining volts or rpm

- Controller.Axis.value() returns a number (the speed based on the position of the joystick) from -127 to +127.
  - → I set my speedControlThreshold for 100 (it takes the absolute value of the speed for determining rpm / volts).

- All other usercontrol code for the drivetrain is the same as it was for build #3, version #1 (see pg. 12).

2/17/21

PROGRAMMING USERCONTROL CONTINUED

Programming Manual Controls:

Controller configuration to allow easiest access to driver macros with somewhat easy to access manual controls:

Controller Configuration

- With the new motor distribution in build #3, version #2 (see pg. 23,25-26), there are several functions that I need manual controls for in the event that a macro fails (or to use with the center goal)

Needed manual functions:

- intaking (both intakes spin inward) – Button R2
- outtaking (both intakes spin in reverse) – Button Left
- scoring (conveyor motor and sorting motor move upward) – Button Up
- disposing (conveyor upward, sorter downward) – Button Right
- reversing / lowering (conveyor and sorter downward) – Button Down

- The manual controls for intaking and outtaking are used when cycling the center goal, but all other manual controls will only be used to correct a failed macro.

project

designed by:

witnessed by:

date: 2/17/21

Megan T 2/17/21

2/18/21

PROGRAMMING USERCONTROL CONTINUED

Programming ball position evaluation: The code to evaluate if ball(s) are present in the robot is the same flow as in build #3, version #1 (see code on pg. 13). Now it has an extra if statement so that all three positions (0, 1, and 2) are occupied or not (positions, page 23).

Programming automatic sorting functions: Again, the code I used in build #3, version #1 worked with minimal modifications (see code on pg. 14). With the new positioning of the optical sensor, having the built-in LED light set to 35% brightness instead of 25% is more consistent. autoSortToggle and identifyColor did not change.

Programming collectBall macro:

- Now that I have 3 ball positions, there are 8 possible combinations for the ways balls are in the robot (see build #3, version #1 code on pg. 15). Code for each position: (position 0, position 1, position 2)

① false, false, false AND true, false, false: bring ball to pos. 2. Check color, keep or dispose

② false, true, false AND true, true, false: intake and bring both balls to pos. 1 and pos. 2, check color, keep both, or dispose one and then bring second ball up to pos 2 and keep or dispose

③ false, false, true AND true, false, true: intake and bring lower ball to pos. 1, check color, keep both, or dispose one and then bring second ball up to pos. 2 and keep or dispose

④ false, true, true AND true, true, true: intake until ball is in pos. 1, check color, keep all or dispose one, bring balls up, and check again, repeat until robot is empty, or ball is to be kept (my alliance's color)

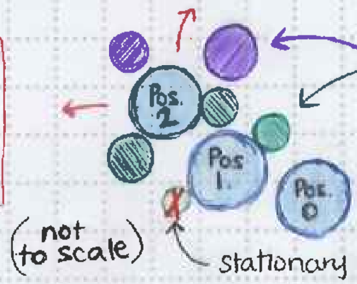
[8 possibilities, 4 options to program]

2/19/21 DESIGN CYCLE:

COLLECT BALL MACRO DISPOSING ISSUE

PROBLEM: In testing my collect Ball macro, I found that when a ball is in position 2 and the conveyor motor (bottom three rollers) is running to bring a ball into position one, the ball in position 2 is disposed by accident (not programmed to do so).

only an issue when dealing with more than one ball at a time



conveyor up, sorter up: scoring
conveyor up, sorter down: disposing
conveyor up, sorter stopped: slowly dispose (unintentionally)

BRAINSTORMING: Possible solutions to the problem:

1. Make position 1 the primary position
   - X must constantly bring balls up and down
   - X very complicated to code efficiently
2. Add another optical sensor - this one in position one - so that position one can be the primary position more efficiently.
   - X requires a lot of re-programming
   - X still a chance of accidental disposing
3. Bring position two ball up and hold it in the sorter when intaking a second ball.
   - X risk of disposing and scoring accidentally
   - X potential for balls to be out of position if reset

SELECT OPTION:

Options to prevent disposing in collectBall macro — 2/19/21 M.M.

Criteria (least to most important)	Scale	① Make pos. 1 the primary position	② Add an optical sensor for pos. 1	③ Hold pos. 2 ball in sorting rollers
Not overly complicated to code	0 to 3	0	3	2
Efficiency and speed	0 to 5	1	5	2
Solves problem of disposing	0 to 5	5	3	1
	Total Score:	6	11	5

Megan M 2/19/21

2/19-23/21

2/20/21

COLLECT BALL MACRO DESIGN CYCLE CONTINUED

> IMPLEMENT SOLUTION :

front of robot

position one distance sensor attached 2/16 (see pg. 33)

NEW position one optical sensor

back of robot

- I mounted the position one optical sensor on the opposite side of the position one distance sensor, but further forward by one hole to have more consistent color detection.

Materials used to mount the position one optical sensor:

- 1 - V5 optical sensor
- 2 - $\frac{1}{2}$ in. screws
- 2 - nylon washers
- 2 - thin nylocks

New Flow for collectBall macro :

- While the actions for each combination of positions are mostly the same as before (see page 36), the execution is now much more efficient.

I modified identifyColor (see pg. 14) to check the color of both positions one and two at the same time, and the code for each combination of positions deals with both balls at once.

↳ no more checking, bring the next one up, checking, and so on

- Position one as the primary position : even with this second optical sensor, there is still a chance of accidental disposing. To minimize this, whenever there is only one ball in the robot, it is brought down to position one (unless being disposed).

Megan M 2/20/21

2/20/21

COLLECT BALL MACRO DESIGN CYCLE CONTINUED

IMPLEMENT SOLUTION CONTINUED:

- IdentifyColor now returns two values: the status of position one ball (keep, dispose, or unknown) and the status of position two ball. The actions performed by my collectBall macro are based on these values.

Here's how it evaluates it (after balls have been put into the right positions):

- if there's one ball in the robot: keep, or dispose (very simple)
- if there's two balls in the robot: if the highest ball is to keep, nothing else to do. otherwise, dispose one or two balls as needed.
- if there's three balls in the robot: same as for two balls, but the ball in the intakes must be brought in if one or both of the higher balls are disposed.

- if any ball is unknown (i.e. the optical sensor was unable to identify the color), the ball is treated as a "keep"

TEST SOLUTION:

- The addition of the position one optical sensor as well as making position one the primary position definitely reduced the problem of accidental disposing.

- The larger benefit of increased efficiency for autonomous and driver macros would be worth it, even if it hadn't helped with the disposing issue.

There are still occasional accidental disposed balls, but it is much better than it was. To help further reduce these instances, I added the code on the next page to bring the pos. 2 ball up to the sorter only if needed to prevent disposing.

Megan M 2/20/21

COLLECT BALL MACRO CONTINUED

```
if(count == 3) { // stop once balls are in positions one and two
  if(DistancePos2.objectDistance(mm) < distancePos2Unoccupied
  && DistancePos1.objectDistance(mm) > distancePos1Unoccupied
  && stopFromDisposing == false) {
    sorterSpin(true, 225);
    stopFromDisposing = true;
  }
  if(stopFromDisposing == true) {
    if(DistancePos2.objectDistance(mm) >= distancePos2Unoccupied) {
      sorterLock();
    }
  }
  if(DistancePos1.objectDistance(mm) < distancePos1Unoccupied) {
    intakesStop();
    conveyorStop();
    count = 4;
    if(stopFromDisposing == true) {
      sorterStop();
      sorterSpin(false, 600);
    }
  }
}
if(count == 4) {
  if(stopFromDisposing == true && DistancePos2.objectDistance(mm) <=
  distancePos2Occupied
  && !(DistanceScoring.objectDistance(mm) < distanceScoringBallSeen)) {
    sorterStop();
    stopFromDisposing = false;
    count = 5;
  }
  else if(stopFromDisposing == false) {
    count = 5;
  }
}
```

slower speed to reduce chance of accidental scoring

the brakeType is set to hold to keep the ball in place

stopFromDisposing is my flag to initiate action to prevent unintended disposing

bring ball back down to pos. 2

2/23/21
M.M.

I added this code to further reduce accidental disposing

MINOR BUILDING MODIFICATIONS

- I added standoffs in front of the intake motors to help protect the wires from getting caught on a goal.

Materials used on standoffs:

- 2 - 2 inch standoffs
- 2 - $\frac{1}{2}$ inch locking screws

- I also moved the flat bearings added to the hood on 2/13 (see pg. 30) to the outside of the metal frame for less strain on the string.

project

designed by:

witnessed by:

date: 2/19-23/21

Megan Y. 2/23/21

2/24/21

PROGRAMMING USERCONTROL CONTINUED (see pgs. 34-40)

Programming ScoreBall macro:

- The evaluation process for my scoreBall macro is similar to collectBall, but different in several aspects:
  - "Unknown" balls: balls that are unable to be identified as red or blue by the optical sensors are to be disposed rather than scored (collectBall keeps unknowns rather than disposing)
  - The goal or ending point of scoreBall is to score one ball - so if there is already a right-colored ball in positions one or two, it scores that. Otherwise, it descores / disposes until a ball can be scored or I stop it with Button A (reset).
  - A more efficient flow: when there are two balls in the robot (positions one and two), having two optical sensors allows me to check both colors and then deal with both accordingly.

Combinations for positioning with scoreBall macro: (position 0,1,2)

1 -, false, false: descores and disposes until a ball is scored

2 -, true, false AND -, false, true: score, or dispose and keep descoring and disposing until a ball is scored.

I do not use position 0 in this macro due to using it at a goal (this could cause false positives).

3 -, true, true:
- a -, dispose*, dispose*: dispose both
- b -, score, score: score one, keep one
- c -, score, dispose*: dispose, then score
- d -, dispose*, score: score, then dispose

* or unknown

each of these combinations ends when one ball has been scored and the remaining ball has been dealt with appropriately.

- This version of the scoreBall macro is much more efficient and beneficial for use in a competition than the first (see pg. 15).

Megan T 2/24/21

2/25/21

PROGRAMMING USERCONTROL CONTINUED

Programming score And Descore macro:

- At the FRA competition (see pgs. 19-21), I found that I needed to score and descore simultaneously when (a) the goal has 3 balls, and (b) when my robot's capacity is full of balls.
- I have programmed Button L2 (see pg. 35) to score one ball and descore one ball simultaneously - due to these actions occurring at the same time, this macro does not have automatic sorting or color identification.

The Flow for each position combination: (position 0,1,2)

① —, false, false: nothing; vibrate the controller to alert me (cannot score and descore if there isn't a ball to score - use collectBall instead).

② —, true, false AND —, false, true AND —, true, true: intake until a ball is in position zero, then score one as ball in position zero is brought into robot. ends when ball from position zero is in position one, or reset button is pressed.

Programming scoreTwo into scoreBall macro:

This code is located at the evaluation for scoreBall to be activated

```
if(Controller.ButtonL1.pressing() == true) {
  int numberOfChecks = 0;
  scoreBallArray[0] = true;

  if(positionArray[3] /* evaluated */ == false) {
    while(numberOfChecks <= 25) {
      wait(4, msec);
      if(Controller.ButtonL2.pressing() == true) {
        scoreTwo = true;
        break;
      }
      numberOfChecks += 1;
    }
  }
}
```

checks for 100 msec ($\frac{1}{10}$ of a second)

2/26/21 M.M.

- Occasionally, I may want to score two balls but don't have time to wait for color identification.
- I built in this option to scoreBall to check if both L1 and L2 are pressing. If so, score two with no automatic sorting.

project designed by: witnessed by:

date: 2/25-26/21

Megan M 2/26/21

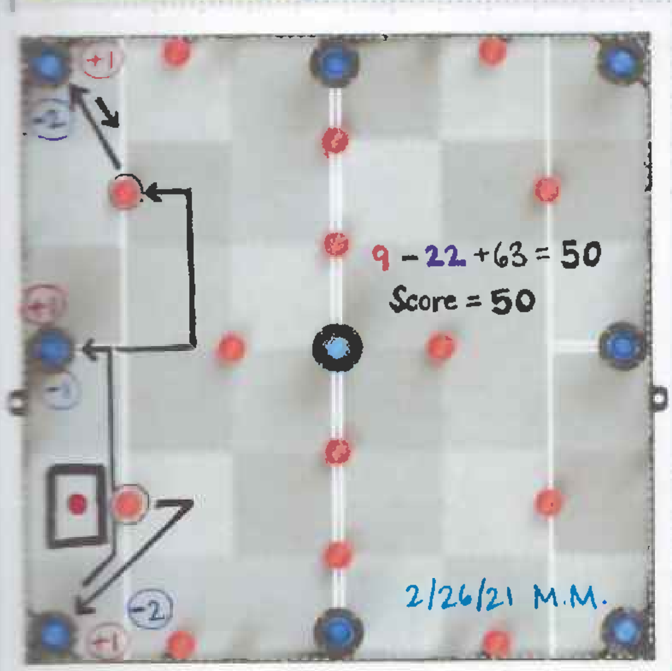
2/26/21

AUTONOMOUS PATHS & PRIORITIZING

GOAL: Determine the simplest and most consistent path for the needed 15 second autonomous routines and autonomous skills. Prioritize what routines need to be completed first.

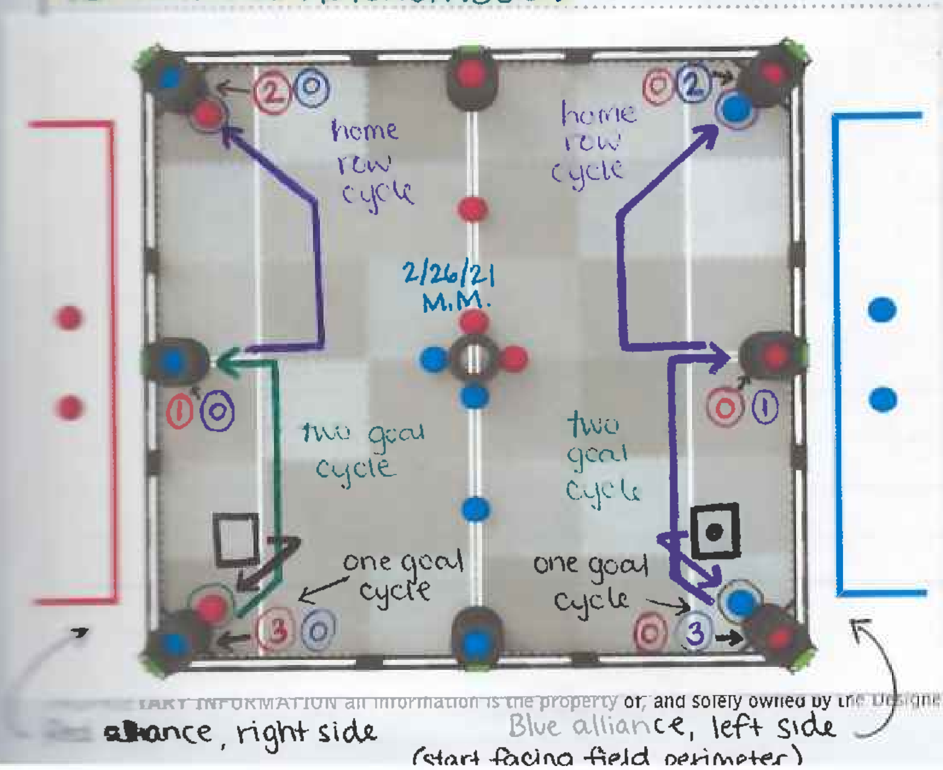
GOAL COMPLETED 2/26/21

Autonomous Skills:



- An improved autonomous skills score from my last competition (see pgs. 19-21) is my top priority for programming.
- Due to time limitations, I plan to only do a row for now (I will add more after my 15 second routines if I have time).
- By completing the home row and descoring all 5 blue balls in those goals, I can score 50 points.

15-second Autonomous:



alliance, right side

Blue alliance, left side (start facing field perimeter)

1. Most important 15 sec. routines: one goal cycle and two goal cycle, left and right.
2. If I have time, I will do a home row cycle, building off the two goal cycle.
3. Lower priority (for later competitions): center goal with 1-2 goal cycle

witnessed by:

date: 2/26/21

Megan M 2/26/21

2/27/21

44 BRAIN SCREEN AUTONOMOUS SELECTOR

```
while(ready == false) {
  // toggle Left on
  if(Brain.Screen.xPosition() > leftCoordinatesArray[0] && Brain.Screen.xPosition() <
  leftCoordinatesArray[1]
  && Brain.Screen.yPosition() > leftCoordinatesArray[2] && Brain.Screen.yPosition() <
  leftCoordinatesArray[3]
  && leftSelected == false) {
    leftSide = true;
    leftSelected = true;
    rightSelected = false;
    // code to print to the brain screen
  }
```

This allows me to toggle between left and right

(see pg. 45)

```
  // toggle Right on
  else if(Brain.Screen.xPosition() > rightCoordinatesArray[0] &&
  Brain.Screen.xPosition() < rightCoordinatesArray[1]
  && Brain.Screen.yPosition() > rightCoordinatesArray[2] && Brain.Screen.yPosition()
  < rightCoordinatesArray[3]
  && rightSelected == false) {
    leftSide = false;
    leftSelected = false;
    rightSelected = true;
    // code to print to the brain screen
  }
```

I reference leftSide to determine which code to run; the code for left if true, and right if false

```
  // flag for calibration to begin
  else if(Brain.Screen.xPosition() > readyCoordinatesArray[0] &&
  Brain.Screen.xPosition() < readyCoordinatesArray[1]
  && Brain.Screen.yPosition() > readyCoordinatesArray[2] && Brain.Screen.yPosition()
  < readyCoordinatesArray[3]
  && (leftSelected == true || rightSelected == true)) {
    ready = true;
    // code to print to the brain screen
  }
```

As soon as I select "ready", calibration of the inertial sensor and color identification of the preload begin – no more toggling is enabled

```
  // toggle Center
  else if(Brain.Screen.pressing() == true) {
    centerToggled = true;
  }
  if(centerToggled == true && Brain.Screen.pressing() == false) {
    // toggle Center on
    if(Brain.Screen.xPosition() > centerCoordinatesArray[0] &&
    Brain.Screen.xPosition() < centerCoordinatesArray[1]
    && Brain.Screen.yPosition() > centerCoordinatesArray[2] &&
    Brain.Screen.yPosition() < centerCoordinatesArray[3]
    && scoreCenter == false) {
      scoreCenter = true;
      centerToggled = false;
      // code to print to the brain screen
    }
```

Whereas, for left and right, I'm toggling two separate buttons, for "center", I'm pressing the same button several times, requiring different flagging.

```
    // toggle Center off
    else if(Brain.Screen.xPosition() > centerCoordinatesArray[0] &&
    Brain.Screen.xPosition() < centerCoordinatesArray[1]
    && Brain.Screen.yPosition() > centerCoordinatesArray[2] &&
    Brain.Screen.yPosition() < centerCoordinatesArray[3]
    && scoreCenter == true) {
      scoreCenter = false;
      centerToggled = false;
      // code to print to the brain screen
    }
  }
}
```

(xPosition() and yPosition() return the most recently pressed coordinates)

This is why I first check that the screen is pressed, then released

2/27/21 M.M.

2/27/21

Megan M 2/27/21

BRAIN SCREEN AUTONOMOUS SELECTOR CONT.

GOAL: Program a simple user-interface on the V5 Brain Screen to allow for additional routines (more than just the 8 slots available for programs to be downloaded into) as autonomous options.

GOAL COMPLETED 2/27/21

- For each routine, I need to have the option to start on the left or right side of the field, so I will need a toggle for that.
- In preparation for future routines scoring the center goal as well as (hopefully!) 1-2 goals on the home row, I need an optional center-scoring button as well.
- The Brain Screen is 480 × 240 pixels.

If the screen is pressed in the coordinates within a rectangle, a ~~toggle is changed~~

2/27/21 M.M.

```
int width = 160; // in pixels
int height = 75; // in pixels

/* left x, right x, top y, bottom y */
int leftCoordinatesArray[4] = {60, 60 + width, 25, 25 + height};
int rightCoordinatesArray[4] = {260, 260 + width, 25, 25 + height};
int centerCoordinatesArray[4] = {60, 60 + width, 125, 125 + height};
int readyCoordinatesArray[4] = {260, 260 + width, 125, 125 + height};
```

all rectangles are the same size

I use the Brain.Screen.drawRectangle() function, which requires these four values

- Once selected, I have programmed the color of the words / rectangle to be inverted to show what has been selected.

- Safeguards: only left or right can be selected, not both. Ready cannot be selected until either left or right is selected.

Right side selected, ready not yet selected on autonomous selector

2/28/21

COLLECTING AND SCORING AUTON. FUNCTIONS

GOAL: Write code for collecting and scoring functions based on the driver macros from 2/18-26 (see pgs. 36-42)

GOAL COMPLETED 3/2/21

- CollectBallAuton and scoreBallAuton are the same as my collectBall and scoreBall macros (see pgs. 36-41) except:
  - They are not based on a button press, but on a function call.
  - There is no disposing - only the return of the flag wrongColor as true ← this is so that I can dispose of a ball when it is convenient to do so without interfering with the routine

scoreAndDescoreAuton is quite different: this allows for smooth autonomous cycling.

```
int scoreAndDescoreAuton(int scoreAndDescoreArray[1]) {
  if(count == 0) {
    intake(200);
    count = 1;
  }
  if(count == 1) {
    if(DistancePos0.objectDistance(mm) < distancePos0Occupied) {
      conveyorSpin(true);
      sorterScore();
      count = 2;
    }
  }
  if(count == 2) {
    if(DistanceScoring.objectDistance(mm) <= distanceScoringOccupied) {
      scoring = true;
    }
    else if(scoring == true && DistanceScoring.objectDistance(mm) >
    distanceScoringOccupied) {
      scoring = false;
      scoreCount += 1;
    }
    if(scoreCount == 2) {
      count = 3;
    }
  }
  if(count == 3) {
    if(DistancePos1.objectDistance(mm) < distancePos1Unoccupied
    || DistancePos2.objectDistance(mm) < distancePos2Unoccupied) {
      count = 4;
    }
  }
  if(count == 4) {
    intakesStop();
    conveyorStop();
    sorterStop();
    count = 0;
    scoreAndDescoreArray[0] /* stepsCompleted */ = true;
  }
  return scoreAndDescoreArray[1];
}
```

this increments as each ball is scored

I modified this code to also have a descoreCount and to have both scoreCount and descoreCount as parameters in the array so I specify the amounts.

3/2/21 M.M.

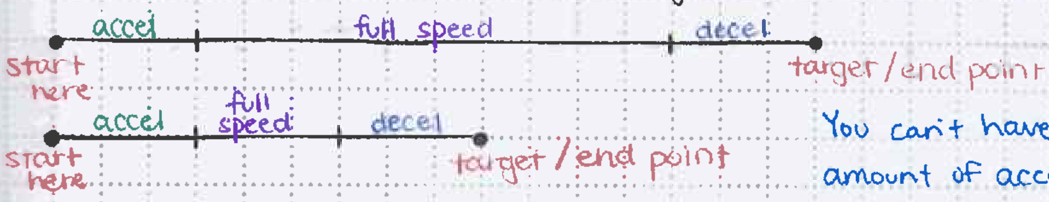
date: 2/28 - 3/2/21

Megan M 2/28/21

3/4/21

ACCELERATION AND DECELERATION

GOAL: Program autonomous functions for driving, turning, and strafing that have proportional acceleration and deceleration for higher consistency and to reduce jerking. **GOAL COMPLETED 3/7/21**



You can't have a set amount of accel / decel.

- Acceleration / deceleration is difficult because it depends how far you need to go, what you want the highest speed to be, and how fast / slow you want it to accelerate or decelerate.
- In order to have a highly versatile acceleration / deceleration, I programmed based on percentages, not set numbers.
- Each iteration, I have the program calculate these values:

```
// get the current encoder value in degrees and determine how much further
currentDegree = DriveMotorRightFront.rotation(deg);
degreesRemaining = driveArray[3] /* degreesToTravel */ - currentDegree;

// number of degrees to accelerate / decelerate based on what percentage of the
total numbers of degrees to travel I specify as an argument in the driveArray */
degreesToAccel = (double(driveArray[4]) /* percentToAccel */ / 100) *
double(driveArray[3]) /* degreesToTravel */;
degreesToDecel = (double(driveArray[5]) /* percentToDecel */ / 100) *
double(driveArray[3]) /* degreesToTravel */;

// percentage of degrees to accelerate / decelerate completed as a fraction
percentAccelComplete = currentDegree / degreesToAccel;
percentDecelComplete = (degreesToDecel - degreesRemaining) / degreesToDecel;
```

← it's all based on the distance traveled

3/6/21 M.M.

By using these values, I know exactly how far into the run it is, how much more it needs to accelerate to get to the highest / target speed, and what percent of the acceleration is complete.

degreesToAccel = (percentToAccel / 100) × degreesToTravel

degreesToDecel = (percentToDecel / 100) × degreesToTravel

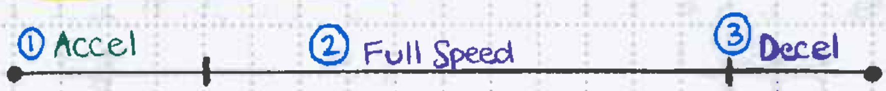
Megan M 3/7/21

3/7/21

ACCELERATION AND DECELERATION CONT.

- Based on the calculations (see previous page), the "zone" that the robot is currently in is determined.

The three "zones":



The distance traveled determines the current zone

Code for determining the current zone

```
if(currentDegree <= degreesToAccel) {
  accel = true;
  fullSpeed = false;
  decel = false;
}
else if(currentDegree > degreesToAccel && currentDegree < (driveArray[3] /*
degreesToTravel */ - degreesToDecel)) {
  accel = false;
  fullSpeed = true;
  decel = false;
}
else if(currentDegree >= (driveArray[3] /* degreesToTravel */ - degreesToDecel)) {
  accel = false;
  fullSpeed = false;
  decel = true;
}
```

3/7/21 M.M.

- Once the current zone is determined, the speed can be set proportionally to the zone and the distance traveled, for the specified amount of acceleration, deceleration, and the highest speed.

Code for calculating speed

```
if(accel == true) {
  setSpeed = percentAccelComplete * driveArray[2] /* speed */; // accelerate
  proportional to distance traveled
}
else if(fullSpeed == true) {
  setSpeed = double(driveArray[2] /* speed */); // set to the target speed when not
  accelerating or decelerating
}
else if(decel == true) {
  setSpeed = (1 - percentDecelComplete) * driveArray[2] /* speed */; // decelerate
  proportional to distance traveled
}
```

3/7/21 M.M.

- setSpeed is then passed to my drive motors in spin commands with the correct direction of turning each wheel.
- I wrote similar functions as this for moving forward/backward, strafing left/right, and turning left/right (which used the Inertial sensor).

Megan W 3/7/21

3/8/21

INCORPORATING TRIGONOMETRY

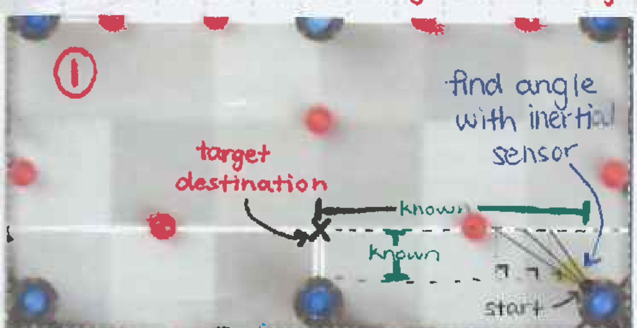
GOAL: Incorporate trigonometric calculations to ensure accuracy regardless of the angle of the robot at the goal. **GOAL COMPLETED 3/10/21**

- Even with acceleration and deceleration, movements based soley on a motor encoder are approximate, not exact - this means that if the robot isn't quite centered on a goal, it can be knocked to an inaccurate angle for the traveling after scoring / descoring that goal.

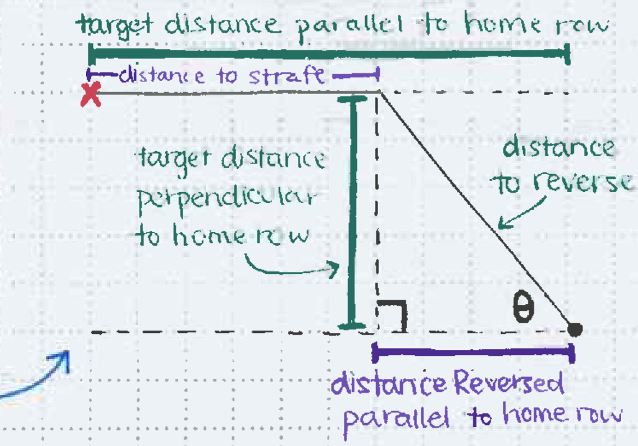
With the inertial sensor, I know what angle the robot is at when it is at a goal, and with some basic trigonometry, I can solve for the distances needed after the goal based on the robot's actual angle.

see path on pg. 43

Two instances for trigonometry in Autonomous Skills:



3/10/21 M.M.



$$\sin\theta = \frac{\text{opposite}}{\text{hypotenuse}}$$

$$(\text{distance to reverse})\ \sin\theta = \frac{\text{target distance perpendicular to home row}}{\text{distance to reverse}} \cdot \cancel{\text{distance to reverse}}$$

$$\frac{\text{distance to reverse} \cdot \cancel{\sin\theta}}{\cancel{\sin\theta}} = \frac{\text{target distance perpendicular to home row}}{\sin\theta}$$

$$\boxed{\text{distance to reverse} = \frac{\text{target distance perpendicular to home row}}{\sin\theta}}$$

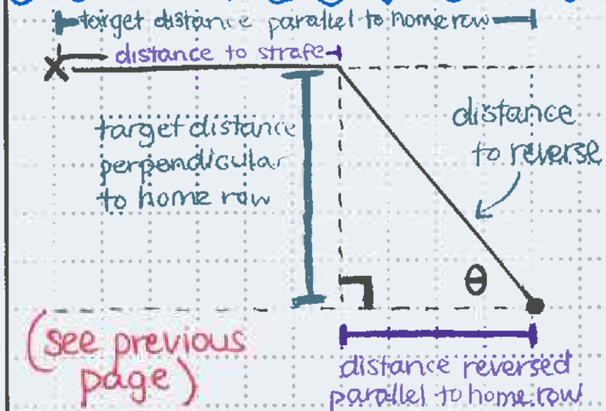
project designed by: witnessed by:

date: 3/8-10/21

Megan M 3/10/21

3/9/21

INCORPORATING TRIGONOMETRY CONTINUED



(see previous page)

$\tan \theta = \dfrac{\text{opposite}}{\text{adjacent}}$

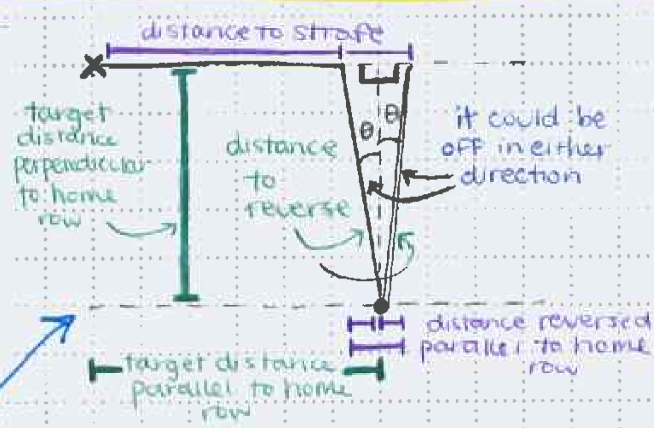
$\tan \theta = \dfrac{\text{target distance perpendicular to home row}}{\text{distance reversed parallel to home row}}$

I need to find this value in order to solve for distance to strafe

$\left(\text{distance reversed parallel to home row}\right) \tan \theta = \text{target distance perpendicular to home row}$

$$\boxed{\text{distance reversed parallel to home row} = \dfrac{\text{target distance perpendicular to home row}}{\tan \theta}}$$

$$\boxed{\text{distance to strafe} = \text{target distance parallel to home row} - \text{distance reversed parallel to home row}}$$



$\cos \theta = \dfrac{\text{adjacent}}{\text{hypotenuse}}$

$\cos \theta = \dfrac{\text{target distance perpendicular to home row}}{\text{distance to reverse}}$

$$\boxed{\text{distance to reverse} = \dfrac{\text{target distance perpendicular to home row}}{\cos \theta}}$$

project

designed by:

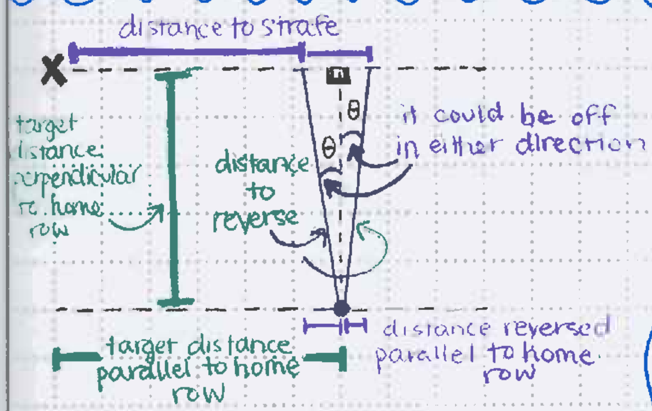
witnessed by:

date: 3/8-10/21

Megan M 3/10/21

3/9/21

INCORPORATING TRIGONOMETRY CONTINUED



distance reversed parallel to home row = $\tan\theta$ × target distance perpendicular to home row

This is the positive distance from the 90° angle made from the goal. Depending which direction the robot is off, this needs to be added or subtracted from the target distance parallel from home row to find the distance to strafe.

(see previous page)

$$\tan\theta = \frac{\text{opposite}}{\text{adjacent}}$$

$$\tan\theta = \frac{\text{distance reversed parallel to home row}}{\text{target distance perpendicular to home row}}$$

✱ NOTE : the sin(), cos(), and tan() functions require the argument to be in radians – must convert the triangle-angle degrees with this formula:

$$\text{angle in degrees} \times \left(\frac{\pi}{180}\right) = \text{angle in rads}$$

ALSO, distances in inches must be converted to encoder degrees:

$$\text{distance in inches} \times \left(\frac{360° \text{ per revolution}}{\text{circumference of wheel}} \cdot \sqrt{2}\right) = \text{encoder degrees}$$

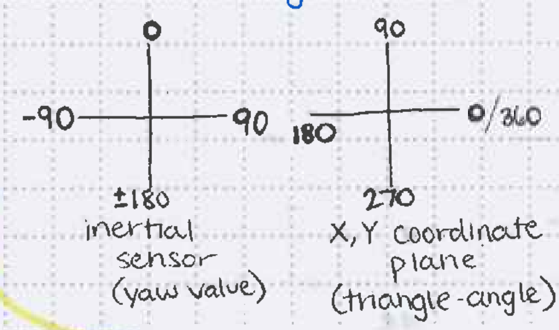
where circumference of wheel = 3.25in × π

diameter of wheel

this takes the X-drive into account

This number can be directly passed to my driving with acceleration / deceleration functions (see pgs. 47-48)

Finding the triangle-angle from the inertial angle:



There isn't one formula to convert from one to the other – you have to figure it out on a case-by-case basis depending on the range needed in the triangle.

INCORPORATING TRIGONOMETRY CONTINUED

- This trigonometry has made my autonomous routines much more consistent, but only when the robot is within a range of degrees at the goal.
  - ↳ to solve this, I have the robot correct its angle while at the goal to be within the working range of values for my trigonometric calculations.

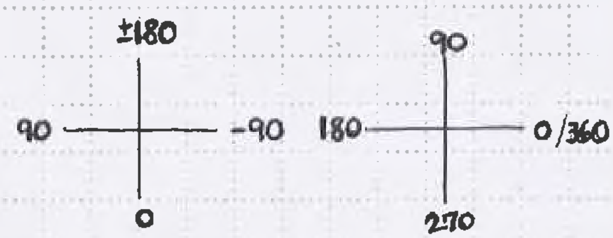
```
if(angleAtGoal < 135) {
  DriveMotorLeftBack.spin(directionType::fwd,100,velocityUnits::rpm);
  DriveMotorRightBack.spin(directionType::rev,100,velocityUnits::rpm);
  waitUntil(Inertial.yaw() >= 135);
  angleAtGoal = Inertial.yaw();
}
else if(angleAtGoal > 145) {
  DriveMotorLeftBack.spin(directionType::rev,100,velocityUnits::rpm);
  DriveMotorRightBack.spin(directionType::fwd,100,velocityUnits::rpm);
  waitUntil(Inertial.yaw() <= 145);
  angleAtGoal = Inertial.yaw();
}
```

range of values: 135° - 145°

This correction code is part of the trig. function for instance ① (see pg. 49-50) in autonomous skills

3/10/21 M.M.

- I can use very similar trig. calculations for the same goals as instances ① and ② (see pgs. 49-51) in my 15-second autonomous routines with modifications due to field set up and the orientation of the inertial sensor for calibration.



inertial sensor (yaw value)

X,Y coordinate plane (triangle angle

↑ I start my 15-second routines facing the field perimeter (opposite of autonomous skills) so the inertial sensor's values are oriented differently (see autonomous skills orientation, pg. 51)

- The only reason for my autonomous routines to fail in sections with trigonometry would be translational drifting of the X-drive due to an uneven weight distribution, etc. Because there is no turning when the robot drifts, the yaw value from the inertial sensor can't help.

project

designed by:

witnessed by:

date: 3/8-10/21

Megan M 3/10/21

FINAL AUTONOMOUS MODIFICATIONS

GOAL: Finalize all autonomous routines needed for Bolt Up (in Chattanooga).

GOAL COMPLETED 3/11/21

- To counter translational drifting, I added an offset variable. For any movement forward, backward, left, or right (not turning), I can specify exactly how much offset (thus drifting, no drifting, or a small amount of drifting) I want there to do - essentially, using it to my advantage.

```
DriveMotorLeftFront.spin(directionType::fwd,setSpeed,velocityUnits::rpm);
DriveMotorLeftBack.spin(directionType::fwd,setSpeed + offset,velocityUnits::rpm);
DriveMotorRightFront.spin(directionType::fwd,setSpeed + offset,velocityUnits::rpm);
DriveMotorRightBack.spin(directionType::fwd,setSpeed,velocityUnits::rpm);
```

I found that the robot tends to drift left, so passing a negative value as the offset to these wheels decreases drift

- In order to maximize points in my autonomous routines, in the event that certain conditions fail to be met (i.e., the robot misses a ball, or accidentally disposes one that was supposed to be scored), I added time conditions to allow the program to continue.

one of the more common mishaps is when attempting to pick up the red ball on the home row line (see path on pg. 43); this extra condition causes the robot to still descore the far corner goal on the home row for 43 points rather than 29.

pseudocode

if (<ball has been collected> **or** Brain.Timer.value ><time it should take>), then: continue to specified step

- By 3/11/21, I have completed a one-goal cycle, two goal cycle, and a home row (no cycling, not enough time in 15 seconds) routines as well as the 50 point autonomous skills routine I was aiming to complete. After Bolt Up, I can look into some routines that score the middle goal.

DRIVER SKILLS : PRACTICE SCORES, PATH, IMPROVEMENT

3/11/21

Date	Total Score	Rows Owned
2/23/21	80	4
	80	4
	75	3
	80	4
	87	5
2/24/21	107	8
	93	6
	74	3
	75	3
	80	4
	86	4
2/28/21	64	2
	90	5
	93	5
	83	4
	94	6
3/1/21	87	4
	85	4
	85	4
	101	6
	86	4
	98	6
	69	2
	84	4
3/2/21	107	8
	93	6
	94	6
	81	4
3/3/21	110	8
	109	8
	94	6

3/11/21 M.M.

Date	Total Score	Rows Owned
3/5/21	94	6
	110	8
	110	8
	97	6
	96	6
	88	5
	81	4
	93	6
	93	6
	95	6
	112	8
	111	8
	98	6
3/6/21	95	6
	94	6
	93	6
	112	8
	97	6
3/7/21	107	8
	90	5
	93	5
	94	6
	97	6
	112	8
	109	8
	111	8
	110	8
	110	8
	81	4
	113	8

3/11/21 M.M.

I programmed the controller to vibrate as a countdown to help me know how long I have left.

```
// warning that time is almost out
if(Brain.Timer.value() >= 58 && Brain.Timer.value() < 58.5 && rumbleCount == 0) {
  Controller.rumble(".");
  rumbleCount = 1; // vibrate once when there are two seconds left
}
else if(Brain.Timer.value() >= 59 && Brain.Timer.value() < 59.5 && rumbleCount == 1) {
  Controller.rumble(".");
  rumbleCount = 2; // vibrate once when there is one second left
}
else if(Brain.Timer.value() >= 59.5 && Brain.Timer.value() < 59.8 && rumbleCount ==
2) {
  Controller.rumble(".");
  rumbleCount = 3; // vibrate once when there is half a second left
}
```

3/11/21 M.M.

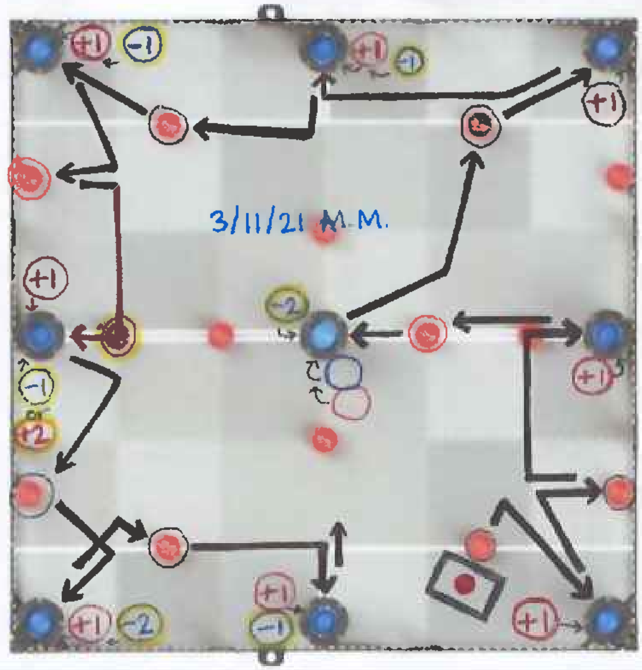
3/11/21 Megan M . 3/11/21

3/11/21

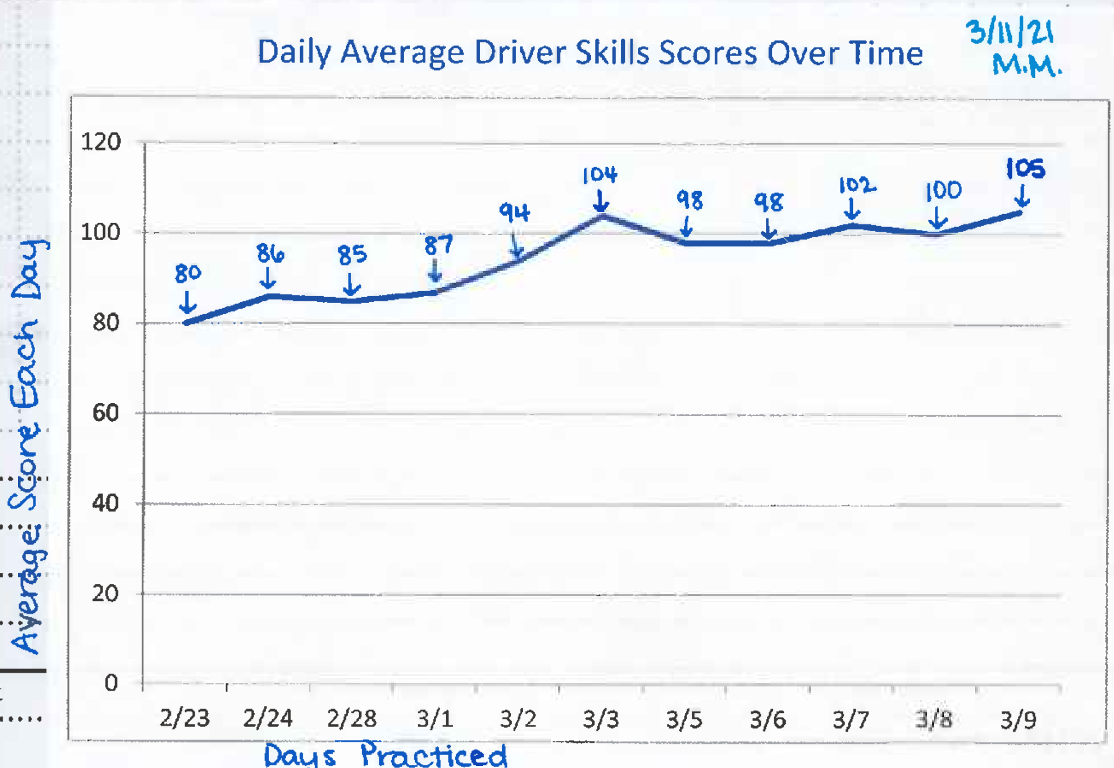
DRIVER SKILLS FROM 2/23 to 3/9

Date	Total Score	Rows Owned
3/8/21	97	6
	95	5
	114	8
	97	6
	101	6
	114	8
	91	4
	84	4
	101	6
	110	8
3/9/21	85	4
	112	8
	94	6
	113	8
	114	8
	100	6
	96	5
	114	8
	116	8

3/11/21 M.M.



Here is the path I follow to efficiently score with minimal driving time/most points (yellow circles indicate what I try to do, but only if there's enough time)



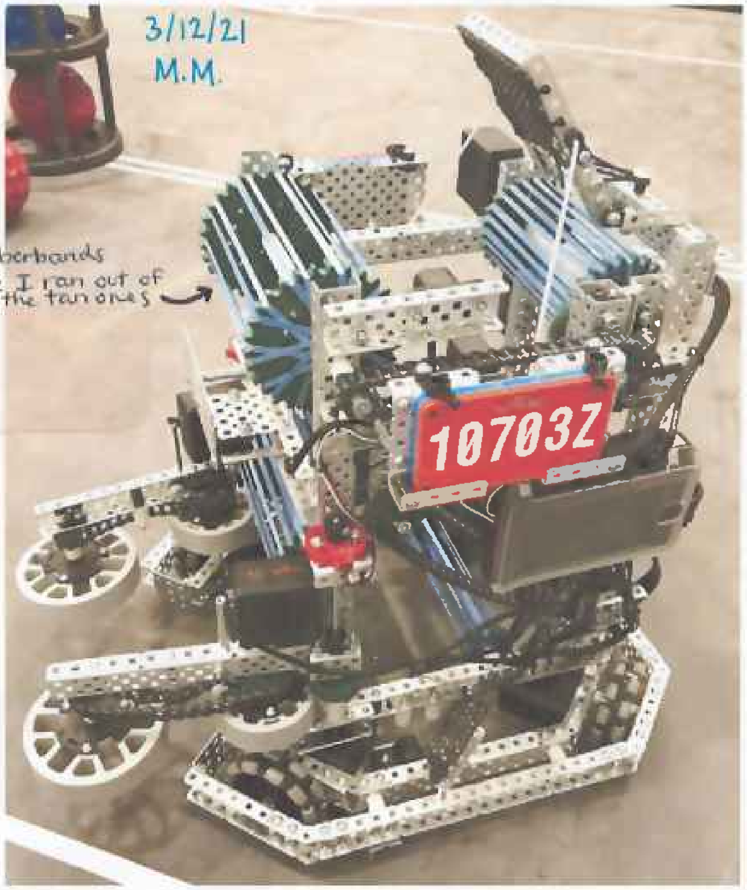
project

3/11/21

Megan M 3/11/21

3/12/21

10703Z for the Bolt Up Competition



Blue rubberbands because I ran out of the tan ones

SUCCESSES:

- Completing build #3, version #2 with an improved ball capacity; a better and more efficient motor distribution
- More efficient driver macros for scoring
- One and two goal auton. routines with cycling, as well as a home row routine and 3 goal skills auton. program
- Improving driver skills scores

THINGS TO IMPROVE:

- Higher scoring auton. skills program
- 15 sec. auton. options to score the center goal
- More driving practice. Consistency in scoring all nine goals.

MY GOALS FOR BOLT UP:

- [x] Get a second interview.
- [x] Rank in the top half after quals
- [x] Score at least 110 points in Driver Skills
- [ ] Score at least 43 points in Autonomous Skills (cycle two goals and descore third goal)

Completed 3/13/21

project

designed by:

witnessed by:

date: 3/12/21

Megan M 3/12/21

3/13/21

BOLT UP COMPETITION ANALYSIS

10703Z Stats :

- Ranked 9th after qualifications
- Ranked 4th in skills, with a total score of 157
- 5 wins, 3 losses after qualifications

- WP = 12
- AP = 36
- SP = 61
- OPR = 4.9
- DPR = 8.3
- CCWM = -3.4

2 WP from completing the home row in autonomous

Bolt Up Skills Results

3/13/21 M.M.

Driver / Auton	Score	Analysis
Driver run #1	$\boxed{115}$	Scored 11 red balls, descored 8 blue balls Owned 8 rows (disposed one red) Good speed and macros only failed once
Driver run #2	101	Scored 9 red balls, descored 8 blue balls Owned 6 rows (disposed Not fast enough, macros failed several times
Driver run #3	57	Got hung up on center goal and second corner goal 1 owned row Scored 6 red balls, descored 3 blue balls
Auton. run #1	$\boxed{42}$	Scored one red ball, descored 5 blue balls wrong angle / not enough correction for first corner goal ; missed red ball on home row line
Auton. run #2	0	did not turn toward corner goal - drove straight across field ↳ inertial sensor not calibrated correctly?
Auton. run #3	42	Same as auton. run #1 The wait time serving as fail safes (see pg. 53) allowed me to score 42 instead of 20 or 28 even though it messed up

Megan M 3/13/21

3/13/21

BOLT UP COMPETITION ANALYSIS CONTINUED

3/13/21 M.M.

Bolt Up Qualification Match Results

Alliance Partner	Opponents	Win / Loss	Score	Analysis
24816T	9364G 99060N	Win	37-2	Left, 1 goal cycle auton works perfectly good speed, decent strategy
663F	38555B 97934X	Win	24-4	Left, 2 goal cycle auton. works perfectly decent speed, decent strategy
9848W	24816H 3231S	Loss	5-17	Right, home row auton. scores 1 goal alliance partner's robot was having difficulty good speed, poor scoring
30207A	663G 9364E	Win	13-7	Left, 2 goal cycle auton works perfectly good speed, good strategy
97934U	663C 663Y	Loss	12-20	Left, 2 goal cycle auton almost works perfectly my auto-sorting fails several times
73973B	12876A 663B	Win	20-15	Left, 2 goal cycle auton mostly works (1 red in each goal) my auto-sorting fails several times
73973A	663W 663E	Win	18-12	Left, 2 goal cycle auton almost works perfectly my auto-sorting fails at least twice good defense
91028A	2775J 38502A	Loss	4-68	Left, 2 goal cycle auton completely fails (scores 1 ball) all macros fail poor scoring and strategy

- I suspect my auto-sorting and ball positioning macros may have failed partially because of interference from sunlight (open air pavilion) and partially due to different rubberbands that seem to wear out faster (see page 56, blue rubberbands).

project designed by: witnessed by:

date: 3/13/21

Megan M 3/13/21

3/13/21

BOLT UP COMPETITION ANALYSIS CONTINUED

Autonomous Routines Success Rates:

- Left side, 2 goal cycle: used 8 times
  - fully worked: 25% (2 times)
  - partially worked: 62.5% (5 times)
  - completely failed: 12.5% (1 time)
- Right side, home row: used once
  - fully worked: 0%
  - partially worked: 100%
  - completely failed: 0%
- Left side, 1 goal cycle: used once
  - fully worked: 100%
  - partially worked: 0%
  - completely failed: 0%

Autonomous routines I did not use:

left side, home row; right side, one goal cycle; right side, two goal cycle; no auton.

Bolt Up Elimination Match Results

3/13/21
M.M.

Alliance Partner: 73973A

Opponents	Win / Loss	Score	Analysis
9848A 663X	Win	14-7	R16 #3-1 Left, 2 goal cycle auton partially works (good corner goal, scores opposite color on center on home row) my auto-sorting macros fail several times
99060K 97934X	Loss	7-61	QF #2-1 Left, 2 goal cycle auton. partially works (good corner goal, gets stuck on center goal on home row) my robot gets entangled with my alliance partner

- I won the Design award!

WHAT I LEARNED:

- Mobile backboards are helpful for consistent scoring if you're right up to a goal, but they cause problems if you're even a few inches back.
- My two goal cycle autonomous works well with an alliance who scores the center goal on the home row by popping up their hood (97934U).

project designed by: witnessed by:

date: 3/13/21

Megan M 3/13/21

3/14/21

PROGRAMMING CHANGES AFTER BOLT UP

- Problems to address:
  - Driver macros not consistently working
  - Home row auton. did not score two corner goals.
  - Auton. skills missed first corner goal and red ball on home row line.

- Improvements to be made:
  - Add more correction and wait time condition failsafes
  - Try to use the accelerometer for a more reliable drift correction
  - A disposing macro (no auto-sorting)

- Because I only have one week until my next competition (see page 24 for Timeline), I need to prioritize how to spend my time: increasing consistency in my driver macros and my autonomous routines, rather than starting on new routines.

- I plan to tune my collectBall macro so that it works with the new blue rubber bands (see pg. 56), both when fresh and worn.

- I also plan to add a new macro, disposeBall so that I can more easily recover from an optical sensor failing to identify the color.
  ↳ At Bolt Up, my auto-sorting would fail when the ball was generally in the correct position with the distance sensor, but just out of sight or at a bad angle for the optical sensor.

- IDEA FOR A MORE RELIABLE METHOD OF DRIFT CORRECTION: I had the idea to check the accelerometer's values (built into the inertial sensor) and correct when the axis opposite to the intended path of motion registers over a certain threshold. I don't know if it will work, but it's worth a try. [See pg 53 for current drift mitigation]

I searched for accelerometers on the VEX FORUM and found very little - very few teams find accelerometers useful, and the only application has been tipping-prevention. ~ M.M. 3/14/21

3/15/21

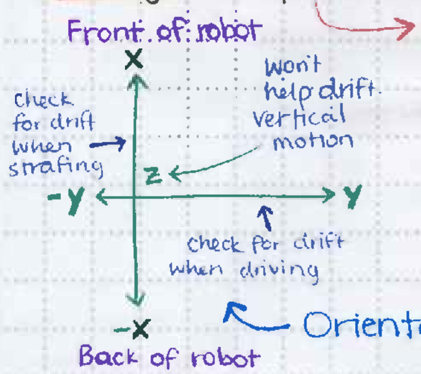
USING ACCELEROMETER FOR DRIFT CORRECTION

GOAL: Test to see if the inertial sensor's built in accelerometer is a viable way to correct translational drift, and compare its reliability to my current drift mitigation (see pg. 53).

GOAL COMPLETED 3/15/21

- The inertial sensor's accelerometer has three axes (x, y, z); based on the orientation of the sensor, these are used to measure different directions: the change in speed.

This means it will only correct when the robot is accelerating or decelerating. The drift is constant when the robot is moving at a constant speed - thus, the drift will not register in the accelerometer's data / readings.



Orientation of inertial sensor on my robot

My drift correction code using the inertial sensor's accelerometer:

this global variable is the threshold for correction

```
double driftRange = 0.02;
double leftOffset;
double rightOffset;
```

3/15/21 M.M.

```
if(accel == true || decel == true) {
  if(Inertial.acceleration(yaxis) >= -driftRange
  && Inertial.acceleration(yaxis) <= driftRange) {
    leftOffset = 0;
    rightOffset = 0;
  }
  else if(Inertial.acceleration(yaxis) < -driftRange) {
    rightOffset = -setSpeed / 4;
    leftOffset = 0;
  }
  else if(Inertial.acceleration(yaxis) > driftRange) {
    leftOffset = -setSpeed / 4;
    rightOffset = 0;
  }
}
else {
  leftOffset = 0;
  rightOffset = 0;
}
DriveMotorLeftFront.spin(directionType::fwd,setSpeed + leftOffset,velocityUnits::rpm);
DriveMotorLeftBack.spin(directionType::fwd,setSpeed + rightOffset,velocityUnits::rpm);
DriveMotorRightFront.spin(directionType::fwd,setSpeed +
rightOffset,velocityUnits::rpm);
DriveMotorRightBack.spin(directionType::fwd,setSpeed + leftOffset,velocityUnits::rpm);
```

no offset in the full speed zone (see pg. 47-48) or when there is no drift (within threshold)

this is when the robot is drifting left (corrects to the right)

when the robot drifts right, it corrects to the left.

I found that the robot drives the smoothest when the correction reduces the speed of two wheels by 25%

date: 3/15/21

Megan T 3/15/21

3/15/21

ACCELEROMETER FOR DRIFT CORRECTION CONT.

- This code (see previous page) is inside my acceleration / full speed / deceleration functions (see pgs. 47-48) for driving forward and backward, and for strafing left and right.

TESTING MY DRIFT CORRECTION CODE:

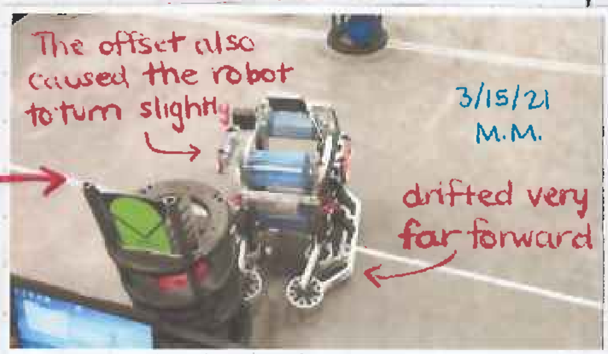
- In order to test my drift correction algorithm, I took out one of the most prone-to-drift-and-cause-inconsistencies parts of my autonomous routines: strafing from the corner goal to the center goal on the home row.

Without the accelerometer

(only offset from 3/11/21 - see pg. 53)

Starting point:

Ended here:

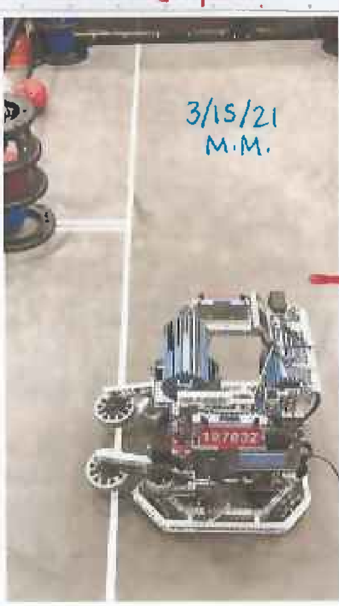


[compared to:]

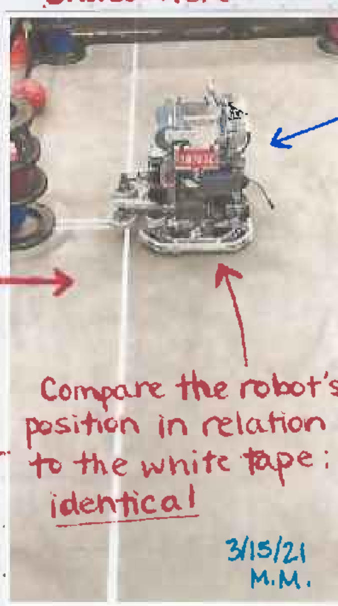
With new drift correction Code

(see previous page)

Starting point:



Ended here:



Compare the robot's position in relation to the white tape: identical!

The accelerometer caused the robot to strafe almost perfectly straight!

In order to have a straighter run, I have to increase the percentage of acceleration in order to have more correction.

project | designed by: | witnessed by:

date: 3/15/21

Megan [redacted] 3/15/21

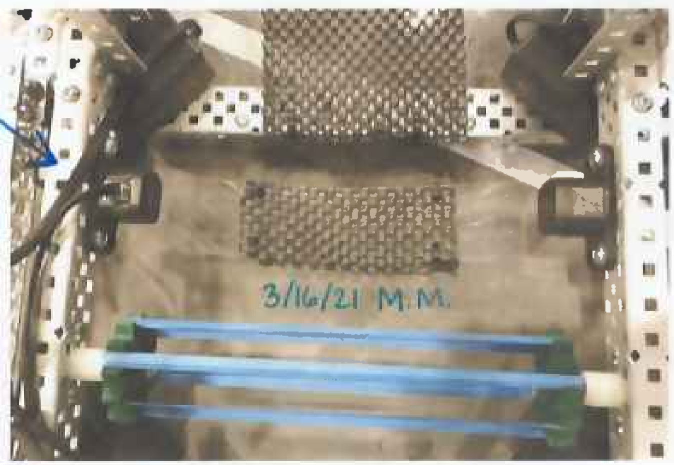
3/16/21

AUTONOMOUS AND MACROS: CHANGES

GOAL: Improve consistency by tuning autonomous routines, programming a disposeBall macro, and sensor adjustments.

GOAL COMPLETED 3/18/21

- In order for the balls to be brought into position one and stopped at the right place for color identification, I moved the position one distance sensor one hole toward the front of the robot.



Now, the optical and distance sensors in position one are lined up, rather than one hole apart.

(See pg. 33 for original location)

- With the new positioning of this sensor, plus the numerous times the balls were unable to be identified at my last competition, I tuned the constant thresholds for each distance sensor.

OTHER IMPROVEMENTS:

- Previously, scoreBall would default to dispose an unknown ball. Now, it will score a ball of the right color and an unknown color.
- I tweaked travel distances, and percentages to accelerate / decelerate with my new drift correction (see pgs. 61-62) for all autonomous routines and for autonomous skills (I only have time to fix the home row routine on the right side of the field, but every other routines has two now-working sides).

See pg. 43 for path of auton routines

- disposeBall macro, see pg. 64

Tuned distance thresholds:

3/16/21 M.M.

```
// ball positioning constants and variables
double distancePos0Unoccupied = 290; // in millimeters
double distancePos0Occupied = 165; // in millimeters
double distancePos1Unoccupied = 165; // in millimeters
double distancePos1Occupied = 25; // in millimeters
double distancePos2Unoccupied = 190; // in millimeters
double distancePos2Occupied = 30; // in millimeters
double distanceScoringOccupied = 85; // in millimeters
double distanceScoringBallSeen = 100; // in millimeters
```

This is used when stopFromDisposing is equal to true

project | designed by: | witnessed by:

date: 3/16-18/21

Megan [redacted] 3/16/21

3/17/21

AUTONOMOUS AND MACROS: CHANGES CONT.

```
int disposeBall(int disposeBallArray[1]) {
  if(Controller.ButtonX.pressing() == true) {
    disposeBallArray[0] = true;
  }
  if(disposeBallArray[0] == true) {

    if(positionArray[3] /* evaluated */ == false) {
      evaluatePosition(positionArray);
    }

    /* false-false-false */
    if(positionArray[0] == false && positionArray[1] == false && positionArray[2] ==
    false) {
      disposeBallArray[0] /* disposeBallActive */ = false;
      positionArray[3] /* evaluated */ = false;
    }

    /* true-false-false */
    else if(positionArray[0] == true && positionArray[1] == false && positionArray[2]
    == false) {

      if(count == 0) {
        intake(200);
        count = 1;
      }
      if(count == 1) {
        if(DistancePos0.objectDistance(mm) <= distancePos0Occupied) {
          conveyorSpin(true, 600);
          sorterSpin(false, 600);
          count = 2;
        }
      }
      if(count == 2) { // stop once ball is in the conveyor
        if(DistancePos1.objectDistance(mm) < distancePos1Occupied) {
          intakesStop();
          count = 3;
        }
      }
      if(count == 3) {
        if(DistancePos2.objectDistance(mm) <= distancePos2Unoccupied) {
          count = 4;
        }
      }
      if(count == 4) {
        if(DistancePos2.objectDistance(mm) > distancePos2Unoccupied) {
          conveyorStop(); // stop once ball has been disposed
          sorterStop();
          count = 5; // reset and exit
        }
      }
      if(count == 5) {
        disposeBallArray[0] /* disposeBallActive */ = false;
        count = 0;
        positionArray[3] /* evaluated */ = false;
      }
    }
```

- I also added more failsafes - Conditions that allow the program to continue if a value is not met

I also have code for –, true, false and –, false, true and –, true, true (very similar to true-false-false)

This macro only evaluates the positions of the balls, not the color.

Code Snippet of new macro: disposeBall

3/17/21 M.M.

date: 3/16-18/21

Megan T. 3/17/21

3/19/21

DRIVER SKILLS PRACTICE SCORES (3/16-19/21)

	Total Score	Rows Owned
1.	99	6
2.	110	8
3.	100	6
4.	113	8
5.	80	3
6.	96	6
7.	100	6
8.	102	6
9.	102	6
10.	114	8
11.	94	6
12.	111	8
13.	115	8
14.	111	8
15.	111	8
16.	115	8
17.	112	8
18.	113	8

3/19/21 M.M.

- In order to get a higher score, I've been trying to score and descore more balls along the way.
  - ↪ I need to work on time management and having an idea of how much time I have left because many of these lower scores were just a second or two away from owning all eight rows.
- My improved macros from tweaks made 3/16-18/21 definitely help - if a ball is unable to be identified, it costs several seconds for me to adjust it manually.
- New trick for an extra two points: after scoring the last goal, if there is a red ball behind the robot, back straight into the center goal - the red ball is pushed in and descores a blue ball in the process.

* I am following the same path as I used at Bolt Up (see pg. 55 for field diagram with path drawn)

project

designed by:

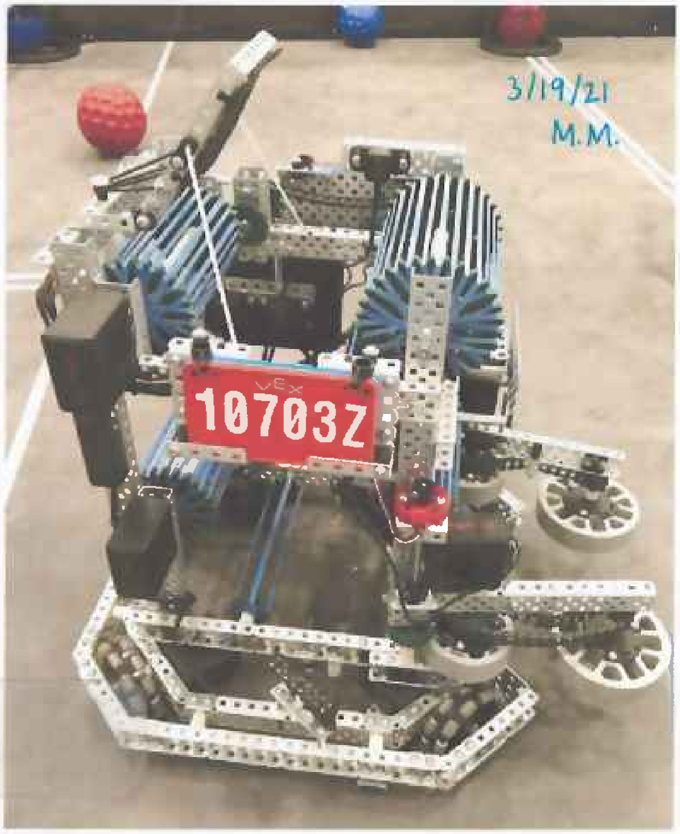
witnessed by:

date: 3/16-19/21

Megan T 3/19/21

3/19/21

10703Z for the St. Patrick's Bluff City Competition



SUCCESSES:

- Unique incorporation of the inertial sensor's accelerometer to successfully correct translational drift.
- Improved macros, including a new dispose Ball macro.
- Increased consistency and correction in autonomous routines.

THINGS TO IMPROVE:

- Home row autonomous routine starting from the left side
- Higher scoring autonomous skills program.
- A more versatile selection of autonomous (15 sec.) routines, including options to score the center goal.
- More driving practice. More consistency in scoring all eight goals.

MY GOALS FOR THE ST. PATRICK'S DAY BLUFF CITY COMPETITION:

- ☑ Get a second interview.
- ☑ Rank in the top half after quals.
- ☒ Score at least 115 points in Driver Skills.
- ☒ Score 50 points in Autonomous Skills (must work perfectly in order to achieve this score)

[Completed 3/20/21]

Megan T 3/19/21

3/20/21

ST. PATRICK'S BLUFF CITY COMPETITION ANALYSIS

10703Z Stats:

- Ranked 2nd after qualifications
- Ranked 2nd in skills, with a total score of 150
- 6 wins, 1 loss, 1 tie after qualifications

- WP = 18
- AP = 48
- SP = 37

5 WP from completing the home row in autonomous

- OPR = 20.1
- DPR = 5.6
- CCWM = 14.5

St. Patrick's Day Bluff City Skills Results

3/20/21 M.M.

Driver / Auton	Score	Analysis
Driver run #1	100	Missed several goals (had to go back), was unable to score one goal. Scored 9 red balls, descored 7 blue balls; 6 rows
Driver run #2	114	Got hung up on center goal. scored 9 red balls, descored 9 blue balls 8 rows
Driver #3	101	Bad time management - just barely missed the last goal. scored 9 balls, descored 8 blue balls, 6 rows
Auton. run #1	30	Missed first corner goal and blue ball got knocked back into goal. Worked perfectly on second two goals
Auton run #2	36	first goal works perfectly, but accidentally disposes red ball on the way to second goal (thus doesn't descore) worked perfectly for last goal.
Auton. run #3	29	first two goals work perfectly. Misses red ball on home row line and does not make it to third goal.

project | designed by: | witnessed by:

date: 3/20/21

Megan M. 3/20/21

 3/20/21

BLUFF CITY COMPETITION ANALYSIS CONT.

St. Patrick's Day Bluff City Qualification Match Results 3/20/21 M.M.

Alliance Partner	Opponents	Win / Loss	Score	Analysis
16859A	63303C 24816T	Tie	11-11	Right, 2 goal cycle auton. works perfectly. decent strategy, poor speed, poor center goal defense
19589A	24816H 16859B	Win	37-2	Right, 2 goal cycle misses second goal good strategy and descoring, poor center goal scoring
1691A	19589A 63303C	Win	28-4	Right, 1 goal cycle mostly works good strategy and speed, poor center goal scoring
63303V	16859B 24816V	Win	38-1	Left, 2 goal cycle works perfectly good strategy, owned all but two goals
24816T	16859B 63303A	Win	48-2	Left, 2 goal cycle works perfectly good strategy and speed good defense
63303A	63303V 1691A	Loss	10-59	Left, 2 goal cycle works perfectly my battery got unplugged by being knocked just right
24816V	24816H 24816T	Win	26-7	Left, 1 goal cycle works perfectly decent strategy and speed
1691A	16859A 24816H	Win	69-0	Right, 2 goal cycle works perfectly good strategy and speed

- My only two losses (one during quals, the other in the final) were the matches where my battery got unplugged by a robot knocking it just right - I definitely need to fix this before my next competition (State)!

project designed by: witnessed by:

date: 3/20/21

Megan T 3/20/21

3/20/21

BLUFF CITY COMPETITION ANALYSIS CONT.

Autonomous Routine Success Rates:

- Left side, two goal cycle: used 3 times
  - fully worked: 100%
  - partially worked: 0%
  - completely failed: 0%
- Right side, two goal cycle: used 5 times
  - fully worked: 40% (2 times)
  - partially worked: 60% (3 times)
  - Completely failed: 0%
- Left side, one goal cycle: used once
  - fully worked: 100%
  - partially worked: 0%
  - completely failed: 0%
- Right side, one goal cycle: used once
  - fully worked: 0%
  - partially worked: 100%
  - Completely failed: 0%
- I did not use my home row routine, or my no auton. option.

St. Patrick's Day Bluff City Elimination Match Results

3/20/21
M.M.

Alliance Partner: 1691A

Opponents	Win / Loss	Score	Analysis
19589A 63303C	Win	29-4	SF #1-1 Right, 2 goal cycle almost works perfectly good strategy, decent speed
24816V 24816T	Loss	1-39	Final #1 Right, 2 goal cycle fails on second goal (leaves wrong color) my battery got unplugged in the first 10 seconds of driver control

- I won the Design Award!

WHAT I LEARNED:

- Cycling the goals on the home row is worth more points than scoring a ball in the center goal (you get points for scoring, descoring, and the home row if completed; whereas the center goal is only 1 point - it rarely completes any rows). I need to prioritize cycling in auton. over the center goal.

project

designed by:

witnessed by:

date: 3/20/21

Megan T. 3/20/21

 3/21/21

PLAN TO IMPROVE FOR STATE (Mechanical fixes and programming)

- Problems to address:
  - ☐ The front of each intakes needs stability to prevent twisting
  - ☐ I need to add cable protection for the wires on the Brain and to prevent the battery from being able to be unplugged.
  - ☐ More efficiency and reliability for my driver macros

- Options to stabilize intakes:
  1. add a standoff in the front hole on each
  2. replace c-channel and potentially shaft on front roller

- **PROGRAMMING PRIORITY:**

Because I only have about three-and-a-half weeks until State, I need to prioritize what has to get done. For the mechanical fixes, the fastest options are to add a standoff to each intake (1 to stabilize intakes), and to change the battery's angle, plus add polycarbonate guards (1 and 3 for cable protection). This will solve the mechanical problems while leaving the most time available for tweaking driver macros and working on autonomous and autonomous skills.

- Improvements to be made:
  - ☐ More consistency in autonomous and autonomous skills
  - ☐ Higher scoring autonomous skills routine
  - ☐ 15-second autonomous option to score or descore in center row after 1 goal cycle
  - ☐ Home row autonomous that starts on left side **OR** improved reliability for right side routine

- Options for Cable protection:
  1. Change the angle of the battery so the cable is more inward
  2. Flip the battery to face the other way and use a longer power cable run inside the conveyor.
  3. Attach polycarbonate to guard the battery cable and smart cables attached to the Brain

- Autonomous Routines Priority:
  1. Improving consistency of two goal cycle
     ↓
  2. Consistency and higher scoring in auton. skills
     ↓
  3. One goal cycle with center row option
     ↓
  4. Consistency for home row auton.

project designed by: witnessed by:

date: 3/21/21

Megan T , 3/21/21

3/21/21

TIMELINE: UNTIL STATE CHAMPIONSHIP

3/21/21
M.M.

Building Fixes
Intakes, Cable Protection
3/22 - 3/23

today

March

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Programming
Macros, Autonomous, Skills
3/23 - 4/14

March

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

April

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Driving Practice
Skills runs
3/24 - 4/16

March

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

April

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Live Remote Skills-only Tournament the day before State - this is good because it ensures I have to get everything done in plenty of time before State ~M.M. 3/21/21

Calendar Key:

days with time to work on the robot

competition

DEADLINES:

- April 15 - Live Remote Skills Tournament
- April 16-17 - TN State Championship

3/22/21

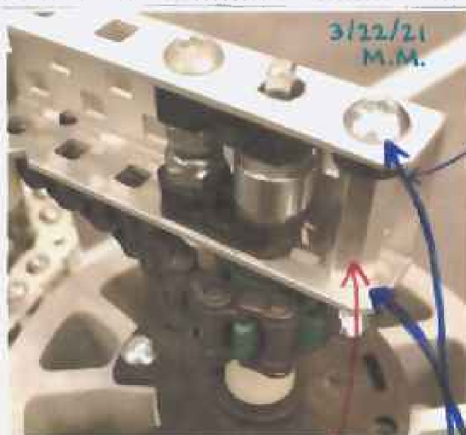
BUILDING FIXES (Intake Stability and Cable Protection)

GOAL: Finish the minor building fixes from 3/21/21 (see pg. 70) so the intakes won't twist and the wires / battery cable won't come unplugged.

GOAL COMPLETED 3/23/21

STABILIZING THE INTAKES:

- In order to help prevent the intakes bending due to banging against the field perimeter or being slammed by scored balls in a goal, I added a standoff on the first hole of each intake.



$\frac{1}{8}$ inch spacer

Left Intake

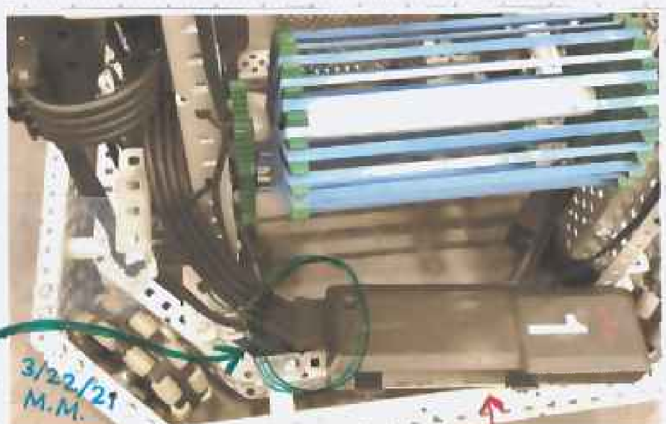
$\frac{3}{4}$ inch standoff added for stability

$\frac{1}{2}$ inch screws

0.375 inch screws

→ Materials used to stabilize intakes: ←

- 2 - $\frac{3}{4}$ inch standoffs
- 2 - $\frac{1}{8}$ inch spacers
- 2 - 0.375 inch screws
- 2 - $\frac{1}{2}$ inch screws



New battery position

much more difficult to be unplugged when knocked

ANGLING THE BATTERY IN:

- After having my battery unplugged twice at the St. Patrick's Day SEMS Bluff City Competition (see pg. 68-69) I want to make sure that there is no way for it to be knocked out again.

- I moved the battery clips mounted on 2/13/21 (see pg. 29) so that the battery plug points inward instead of outward.

project

designed by:

witnessed by:

date: 3/22-23/21

Megan M 3/22/21

3/23/21

BUILDING FIXES CONTINUED

POLYCARBONATE CABLE PROTECTION:

- While I was already ensuring my battery wouldn't get unplugged, I wanted to also add a precautionary polycarbonate guard so all of my wiring is safe from being pulled / entangled.

- I positioned the section of polycarbonate so that it does not interfere with the left intake's stowing.

Bottom left corner is ziptied loosely to the horizontal base support bar

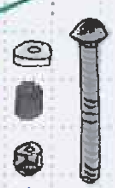
2 in standoffs

6" by 4.5" polycarbonate smart cable protection

$1\frac{1}{2}$" by $2\frac{3}{4}$" polycarbonate power cable protection

4" ziptie to attach polycarbonate to base

small latex loop



on 0.875in. screw

(not to scale)

$\frac{1}{8}$ inch. spacer

8mm shaft spacer

thin nylock

- Even with the battery at an inward angle (see previous page), I still wanted to add a small piece of polycarbonate to make extra sure that the battery cannot be unplugged during a match.

To change the battery, I lift the polycarbonate (latex stretches) and it pivots on the ziptie.

Materials used in polycarbonate cable protection:

- 1 - 4.5" by 6" polycarbonate
- 1 - $2\frac{3}{4}$" by $1\frac{1}{2}$" polycarbonate
- 2 - 2 inch standoffs
- 4 - 0.375 inch screws
- 1 - 0.875 inch screws
- 1 - $\frac{1}{8}$ inch spacer
- 1 - 8 mm shaft spacer
- 1 - thin nylock
- 3 - 4" zipties
- 1 - small latex loop

project | designed by: | witnessed by:

date: 3/22-23/21

Megan T 3/23/21

3/24/21

USERCONTROL PROGRAMMING IMPROVEMENTS

GOAL: Make adjustments to the usercontrol coding and the macros to maximize usability, reliability, and efficiency.

GOAL COMPLETED 3/26/21

SMOOTHER TRANSITIONING FROM DRIVING TO STRAFING AND VICE-VERSA:

- The code I wrote on 1/29/21 (see pg. 12) to control drivetrain motion works well for preventing conflicting motor commands, but the deadzones make switching between driving and strafing jerky and the robot comes to a complete stop, which is slow and inefficient.

- Today I wrote a much simpler algorithm to determine which motion to trigger, keeping the execution code the same as from 2/17/21 (see pg. 34).

```
// Turning has priority over driving and strafing
if(abs(Controller.Axis4.value()) > joystickStopRange) {
  drivetrainArray[0] /* driveActive */ = false;
  drivetrainArray[1] /* turnActive */ = true;
  drivetrainArray[2] /* strafeActive*/ = false;
}
else if(abs(Controller.Axis2.value()) > joystickStopRange
|| abs(Controller.Axis1.value()) > joystickStopRange) {

  if(abs(Controller.Axis2.value()) > abs(Controller.Axis1.value())) {
    drivetrainArray[0] /* driveActive */ = true;
    drivetrainArray[1] /* turnActive */ = false;
    drivetrainArray[2] /* strafeActive*/ = false;
  }
  else if(abs(Controller.Axis2.value()) <= abs(Controller.Axis1.value())) {
    drivetrainArray[0] /* driveActive */ = false;
    drivetrainArray[1] /* turnActive */ = false;
    drivetrainArray[2] /* strafeActive*/ = true;
  }
}
else {
  drivetrainArray[0] /* driveActive */ = false;
  drivetrainArray[1] /* turnActive */ = false;
  drivetrainArray[2] /* strafeActive*/ = false;
  drivetrainStop();
}
```

Turning left/right

Driving forward/backward

Strafing left/right

no deadzones

3/24/21
M.M.

- This works much better, and switches very smoothly.

project | designed by: | witnessed by:

date: 3/24-26/21

Megan 3/24/21

3/24/21

USERCONTROL IMPROVEMENTS CONTINUED

INCREASING THE USABILITY OF ScoreTwo:

(see pg. 42 for original scoreTwo code)

- Having the option to score two balls at once is super useful, especially for skills, but with the current code, if I push L2 even a few seconds before L1, scoreAndDescore executes instead of scoreBall → scoreTwo.

- I added this evaluation code to scoreAndDescore so that I can press L1 and L2 in either order to activate scoreTwo (as long as they are pressed within 1/10 of a second of each other).

```
if(Controller.ButtonL2.pressing() == true && scoreAndDescoreArray[0] == false &&
scoreTwo == false) {
  int numberOfChecks = 0;
  scoreAndDescoreArray[0] = true;

  if(positionArray[3] /* evaluated */ == false) {
    while(numberOfChecks <= 25) {
      wait(4, msec);
      if(Controller.ButtonL1.pressing() == true) {
        scoreTwo = true;
        scoreAndDescoreArray[0] = false;
        break;
      }
      numberOfChecks += 1;
    }
  }
}
```

only runs if L2 is pressed first, and neither scoreBall nor scoreAndDescore is active

runs scoreTwo from scoreBall if L1 is pressed instead of scoreAndDescore

checks for 100 milliseconds

3/24/21 M.M.

ALLOWING MANUAL INTAKE CONTROL WHEN SCORING

- I added a flag, allowManualIntakingWhileScoring, that allows manual control of the intakes while scoreBall is active and the macro is not sending commands to the intake motors (this only happens when false-false is running)

```
if(allowManualIntakingWhileScoring == true) {
  scoreBall(scoreBallArray); // allows manual intaking / outtaking when scoring as
  long as the
  intakesManual(intakesManualArray); // intakes are not in use in the scoring macro
}
```

Here is the code in usercontrol() to allow more versatile scoring

```
if(scoreTwo == true && scoreBallArray[0] == false) {
  scoreBallArray[0] = true; // in case L1 and L2 aren't pressed exactly at the same
  time
}
```

The only time this is true is if I press L2, then L1, running the first segment of code on this page.

3/24/21 M.M.

date: 3/24-26/21

Megan M. 3/24/21

3/25/21

USERCONTROL IMPROVEMENTS CONTINUED

IMPROVING THE RELIABILITY OF COLLECTBALL:

- Previously when bring balls to positions one and two simultaneously, I had stopped the rollers based on position one, which often caused accidental disposing even with my stopFromDisposing code (see pg. 40).

- Now, I have programmed collectBall, scoreBall, scoreAndDescore, and disposeBall to stop based on position two (where applicable).

- The problem: When the higher ball reaches position two, oftentimes the lower ball will be halfway between positions one and ~~two~~ zero, and the distance sensors in both of these positions register as occupied; thus, collectBall is unable to bring the lower ball to the correct position.

- The solution: To rectify this issue, I added a positionOneHalfOccupied flag to check the values of the distance sensors in positions one and ~~two~~ zero once false-true-false and false-false-true (initial positioning) is completed.

```
if(DistancePos1.objectDistance(mm) >= distancePos1Unoccupied) {
  positionOneHalfOccupied = true; // check that balls are positioned correctly
}
```

3/25/21 M.M.

```
if(positionArray[3] /* evaluated */ == false) {
  evaluatePosition(positionArray);
  if(positionOneHalfOccupied == true) {
    correctPositionOneHalf = true;
  }
  else {
    correctPositionOneHalf = false;
  }
}
```

3/25/21 M.M.

The next time I press L1 to activate collectBall (assuming I don't use any other macros in between), this flag triggers for correction code to run (see next page).

- Pressing the reset button or using any other macros or manual controls resets this flag to false.

Megan T. 3/26/21

3/26/21

USERCONTROL IMPROVEMENTS CONTINUED

IMPROVING THE RELIABILITY OF COLLECT BALL CONTINUED:

```
/* correction for positionOneHalfOccupied */
if(correctPositionOneHalf == true) {
  if(count == 0) {
    sorterLock();
    conveyorSpin(false, 600);
    count = 1;
  }
  if(count == 1) {
    if(DistancePos2.objectDistance(mm) > distancePos2Unoccupied) {
      positionOneHalfOccupied = true;
      conveyorStop();
      intake(200);
      conveyorSpin(true, 600);
      count = 2;
    }
  }
  if(count == 2) {
    if(DistancePos2.objectDistance(mm) < distancePos2Unoccupied) {
      intakesStop();
      conveyorStop();
      positionOneHalfOccupied = false;
      correctPositionOneHalf = false;
      collectBallArray[0] /* collectBallActive */ = false;
      count = 0;
      positionArray[3] /* evaluated */ = false;
    }
  }
}
```

bring ball in position two down so as to not dispose it

← bring both balls up, stopping when the highest ball again reaches position two

Here is the correction code to try to get the ball in between positions zero and one up to one.

I have the flags reset once this correction runs once

3/26/21 M.M.

- Sometimes the "positionOneHalf" ball is still unable to make it to position one by the time the higher ball reaches position two, but if I don't reset the flags, it can get stuck only repeating this code if the ball never makes it.

See pgs. 74-77 → for macro improvements

While my macros still aren't perfectly consistent (and never will be), the changes I've made the past few days have greatly improved the performance of the robot, and it definitely has made driving the robot easier.

~M.M. 3/26/21

Megan M 3/26/21

3/26/21

MORE CONSISTENCY FOR TWO-GOAL AUTON.

GOAL: Increase the consistency of the two-goal cycle 15 sec. autonomous routine and apply these changes to other autonomous programs as needed / where applicable. **GOAL COMPLETED 3/31/21**

UPDATING THE BRAIN SCREEN AUTONOMOUS SELECTOR:

New display - no option for center on Two Goal program and Home - Row program

- Because my two-goal cycle routine takes almost the full 15 seconds (when going at a speed to work reliably), I removed the option for Center from the autonomous selector (only for two-goal and home-row). See pg. 44-45 for code and display with center option
- I do hope to add an optional center goal scoring to my one-goal cycle, but it is lower priority (see pg. 70)

MORE DRIFT CORRECTION: FOR FULL SPEED

- See pg. 61-62 for accelerometer drift correction when accelerating or decelerating!

- Using the x and y axes of the inertial sensor's accelerometer works very well for translational drift correction when accelerating or decelerating. But, accelerometers measure a change in velocity, so it won't register drifting when the robot is moving at a constant speed (thus, constant level of drift).
- I can increase the percent of acceleration and / or deceleration for more drift correction, but this takes longer as the robot would be moving slower for longer. Instead, I want to write an algorithm that corrects during full speed based on the encoder values of the wheel motors.

See pg. 47-48 for acceleration / deceleration before correction with accelerometer (pg. 61-62))

project designed by: witnessed by:

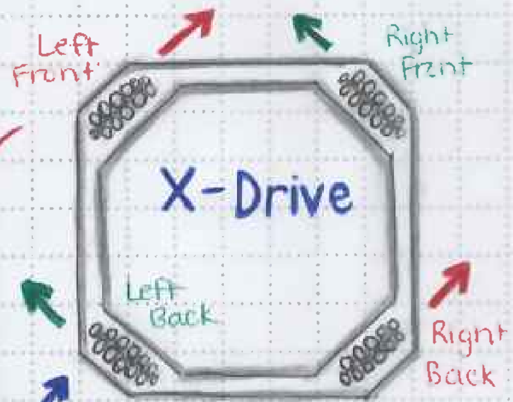
date: 3/26-31/21

Megan M 3/28/21

3/28/21

MORE CONSISTENCY FOR AUTON. CONTINUED

DRIFT CORRECTION FOR FULL SPEED CONTINUED:



not to scale of my robot

(see Book One, pg 126 for dimensions of my x-drive)

- Due to the orientation of the wheels on an x-drive, when drifting left, the right front and left back wheels should have a higher encoder value than left front and right back (vice-versa for drifting right).
- By comparing the pairs of encoders, I can generally determine the translational drift (doesn't account for slippage).

```
else if(fullSpeed == true) {
  leftEncoderAverage = (fabs(DriveMotorLeftFront.rotation(deg)) +
  fabs(DriveMotorRightBack.rotation(deg))) / 2;
  rightEncoderAverage = (fabs(DriveMotorRightFront.rotation(deg)) +
  fabs(DriveMotorLeftBack.rotation(deg))) / 2;

  if(leftEncoderAverage > rightEncoderAverage) {
    driftError = (leftEncoderAverage - rightEncoderAverage) / leftEncoderAverage;
    leftDriftOffset = setSpeed * driftError;
    rightDriftOffset = 0;
  }
  else if(leftEncoderAverage < rightEncoderAverage) {
    driftError = (rightEncoderAverage - leftEncoderAverage) / rightEncoderAverage;
    rightDriftOffset = setSpeed * driftError;
    leftDriftOffset = 0;
  }
  else {
    leftDriftOffset = 0;
    rightDriftOffset = 0;
  }
}
```

larger on the bottom

These drift offsets are subtracted from setSpeed on the appropriate wheels when sent to the motors

lowers speed proportionally to the amount of drift

if equal, no drift offset

Full speed drift correction based on pairs of encoder values

3/29/21 M.M.

```
void determineHighestEncoderValue() {
  highestEncoderValue = fabs(DriveMotorLeftFront.rotation(deg));
  if(fabs(DriveMotorLeftBack.rotation(deg)) > highestEncoderValue) {
    highestEncoderValue = fabs(DriveMotorLeftBack.rotation(deg));
  }
  if(fabs(DriveMotorRightFront.rotation(deg)) > highestEncoderValue) {
    highestEncoderValue = fabs(DriveMotorRightFront.rotation(deg));
  }
  if(fabs(DriveMotorRightBack.rotation(deg)) > highestEncoderValue) {
    highestEncoderValue = fabs(DriveMotorRightBack.rotation(deg));
  }
}
```

With this new drift correction, I don't know which encoder to base the distance traveled on, so now it uses whichever is highest each iteration.

absolute value for easier comparison when reversing or strafing

3/29/21 M.M.

this is now used in my driveForward, driveBackward, strafeLeft, and strafeRight functions

date: 3/26 - 31/21

Megan M 3/29/21

3/30/21

MORE CONSISTENCY FOR AUTON. CONTINUED

IMPROVED CYCLING CODE:

3/31/21 M.M.

(See pg. 81)

```
int scoreAndDescoreAuton(int scoreAndDescoreArray[1]) {
  if(count == 0) {
    ballScored = false;
    scoring = false;
    intake(200);
    startTime = Brain.Timer.value();
    count = 1;
  }
  if(count == 1) {
    if((DistancePos1.objectDistance(mm) <= distancePos1Occupied &&
    DistancePos1.isObjectDetected() == true)
    || (DistancePos2.objectDistance(mm) < distancePos2Unoccupied &&
    DistancePos2.isObjectDetected() == true)) {
      conveyorLock();
      count = 2;
    }
    else {
      conveyorSpin(true, 600);
    }
  }
  if(count == 2) {
    if(DistancePos0.objectDistance(mm) < distancePos0Occupied &&
    DistancePos0.isObjectDetected() == true) {
      intakesStop();
      conveyorSpin(true, 600);
      sorterScore();
      count = 3;
    }
  }
  if(count == 3) {
    if(ballScored == false && scoring == false && DistanceScoring.isObjectDetected()
    == true
    && DistanceScoring.objectDistance(mm) <= distanceScoringBallSeen) {
      if(lastBallToDescore == false) {
        intake(200);
      }
      conveyorStop();
      scoring = true;
    }
    else if(ballScored == false && scoring == true &&
    (DistanceScoring.isObjectDetected() == false
    || DistanceScoring.objectDistance(mm) > distanceScoringBallSeen)) {
      if(lastBallToDescore == false) {
        intake(200);
      }
      wait(100, msec);
      sorterStop();
      scoring = false;
      ballScored = true;
    }

    if(ballScored == true || Brain.Timer.value() >= startTime + 1.5) {
      if(lastBallToDescore == false) {
        intake(200);
        conveyorSpin(true, 600);
      }
      else if(lastBallToDescore == true) {
        intake(100);
        conveyorSpin(true, 600);
      }
      count = 4;
      startTime = Brain.Timer.value();
    }
  }
```

brings a ball to either position one or two (or immediately moves to the next step if already there)

brings a second ball into the intakes (so that if the goal is full, the ball being scored will be less likely to fall off the top)

score highest ball

Start bringing the ball in the intakes into the conveyor

lastBallToDescore is a flag that I pass in when the function is called. It causes intaking to start later if it is the last one so it doesn't descore one too many.

Code continues on next page

3/26-31/21

Megan N 3/31/21

3/31/21

IMPROVED CYCLING CODE CONTINUED:

MORE CONSISTENCY FOR AUTON. CONTINUED

← Continued from previous page

```
  if(count == 4) {
    if((DistancePos1.objectDistance(mm) < distancePos1Unoccupied &&
    DistancePos1.isObjectDetected() == true)
    || Brain.Timer.value() >= startTime + 1) {
      if(lastBallToDescore == false) {
        intakesStop();
      }
      else if(lastBallToDescore == true) {
        outtake(200);
      }
      conveyorLock();
      sorterStop();
      count = 0;
      scoreAndDescoreArray[0] /* stepsCompleted */ = true;
    }
  }
  return scoreAndDescoreArray[1];
}
```

The functions ends when the ball being descored is in position one

Like the original, this function does not evaluate the color of the balls (no auto-sorting)

3/31/21 M.M.

- One of the ways to increase consistency in autonomous is by ensuring the robot cycles correctly - i.e. scores and descores the correct number, all balls make it into the goal, ensuring the goal is at least owned by your alliance color, even if the other balls don't end up exactly right.

- This revised function, scoreAndDescoreAuton (see pg. 46 for original code), is designed for there to be a slighter larger gap between the balls being scored so that there is less chance of a collision where either one or neither ball make it to the goal.

The only problem with the performance of this function is that if a ball, for whatever reason, falls back onto the conveyor (not making it into the goal), the robot will continue to descore as the scoring distance sensors sees the ball (and the other balls won't have enough force to push it out of the robot).

→ if this becomes a problem (happening often) or if I have extra time, I will look into some correction code to more appropriately handle this situation.

project designed by: witnessed by:

date: 3/26-31/21

Megan M 3/31/21

 3/31/21

MORE CONSISTENCY FOR AUTON. CONTINUED

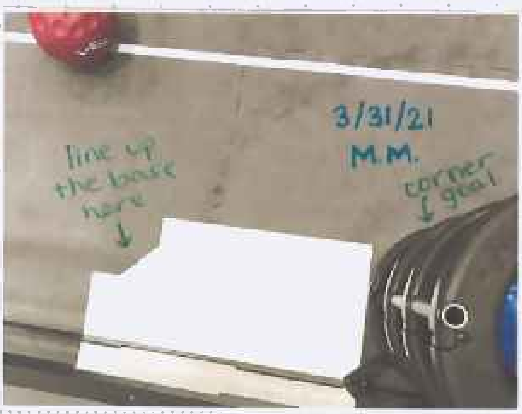
DEALING WITH FIELD DIFFERENTIATION: TEMPLATES:

- All of these improvements this past week have helped my two-goal cycle autonomous routine be more consistent, but when I run the program with the robot on the opposite side as usual (to simulate a different field, like there will be at the State Championship), it doesn't work as consistently.

→ The angles are weird; the robot doesn't seem to go the same distances, etc.

I realized that if I line up the robot with the foam tiles, it isn't always the same because of where the field perimeter rests on the tiles. In order to line up with the goals and account for any angle differences in the field perimeter, I measured and cut these templates:

Autonomous Skills Template



15 sec. Autonomous Template, right side (template can be flipped and used on the left side of the field as well)

- Not only do these templates help minimize issues due to field differences, but they allow me to program around any robot weight distribution differences (etc.) and ensure an exactly accurate starting angle.

- My two-goal cycle and one-goal cycle (just a shorter version of the two-goal cycle) work on all sides of my field with the templates.

project

designed by:

witnessed by:

date: 3/26-31/21

Megan M 3/31/21

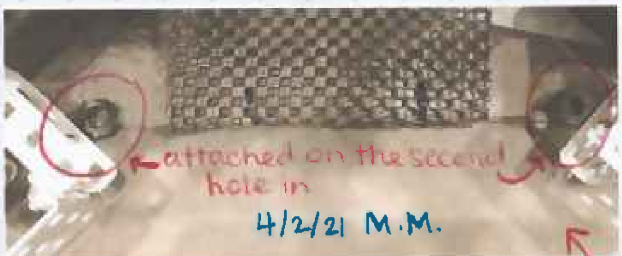
4/1/21

MINOR BUILDING ADJUSTMENTS

GOAL: Adjust the rubberbands and anti-slip around the ~~front~~ of the conveyor to make descoring ~~an~~ smoother and to prevent a ball from getting stuck in position one.

GOAL COMPLETED 4/2/21

- Having the balls flow smoothly is highly dependent on the level of compression provided by the rubberbands and the traction on the antislip on the floor of the conveyor, as well as the tension on the latex on the intakes.



→ Materials used to attach high strength flat bearings: ←

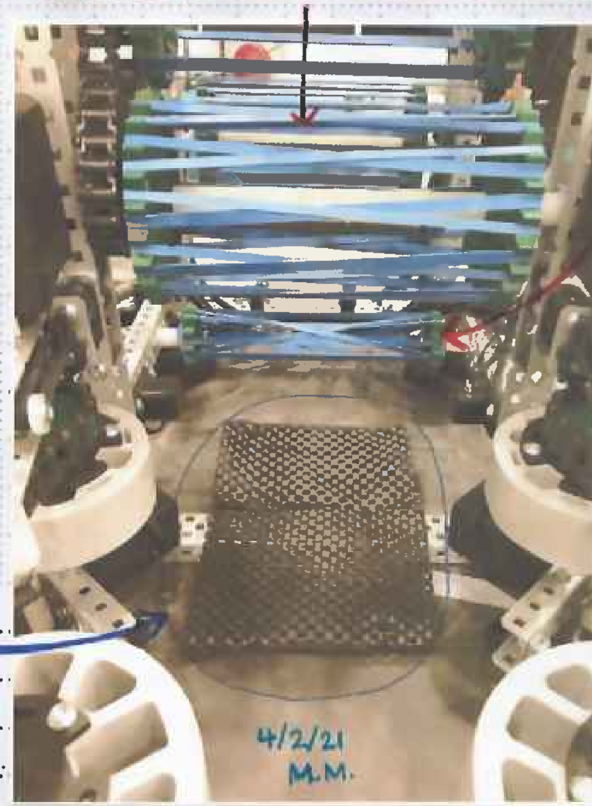
- 2 - high strength flat bearings (cut)
- 2 - $\frac{1}{2}$ inch screws
- 2 - thin nylocks
- also 4 - sections of anti-slip mat (see below)

- Sometimes it seems that the balls catch on the polycarbonate, so I added two high strength flat bearings to lower the polycarbonate slightly.

high strength flat bearing

one end cut

crossed for more tension in the middle



- I also added a section of anti-slip mat extending from where the first section ended to the strip of polycarbonate across the base.
  - ↳ the extra layers give more traction and more compression on the rubberbands

- I crossed the rubberbands on the lowest conveyor roller and the stationary sprockets.

Megan M. 4/1/21

4/3/21

MONITORING MOTOR TEMPERATURES

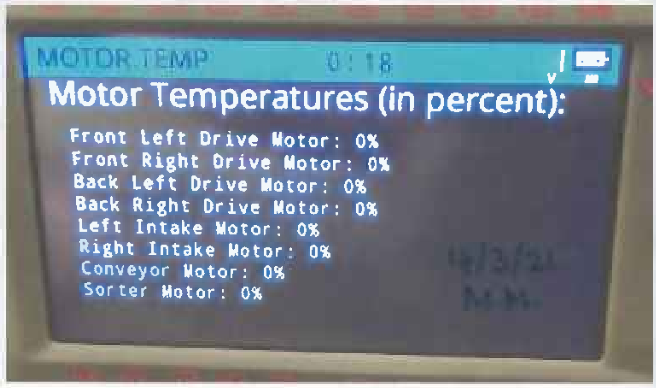
GOAL: Write a program to check the temperature of each motor.

GOAL COMPLETED 4/3/21

- After running the robot for awhile (about 7-10 minutes straight, or half an hour off and on), the robot will start driving very crooked and soon only two wheels will turn at all (so the robot goes in a circle slowly), and the drive motors are very warm to the touch.

- Having an objective way to check the temperature of the motors would be great for competitions and general troubleshooting, so I wrote this program.

70% is when the motors start acting strangely (80% is when it drives in a circle)



Brain Screen display when motors are cool

```
void getMotorTemps() {
  Brain.Screen.setFont(propL);
  Brain.Screen.setCursor(1, 2);
  Brain.Screen.print("Motor Temperatures (in percent):");

  Brain.Screen.setFont(monoM);
  Brain.Screen.setCursor(3, 4);
  Brain.Screen.print("Front Left Drive Motor: %.0f%%%%",
  DriveMotorLeftFront.temperature(pct));
  Brain.Screen.setCursor(4, 4);
  Brain.Screen.print("Front Right Drive Motor: %.0f%%%%",
  DriveMotorRightFront.temperature(pct));
  Brain.Screen.setCursor(5, 4);
  Brain.Screen.print("Back Left Drive Motor: %.0f%%%%",
  DriveMotorLeftBack.temperature(pct));
  Brain.Screen.setCursor(6, 4);
  Brain.Screen.print("Back Right Drive Motor: %.0f%%%%",
  DriveMotorRightBack.temperature(pct));
  Brain.Screen.setCursor(7, 4);
  Brain.Screen.print("Left Intake Motor: %.0f%%%%", IntakeMotorLeft.temperature(pct));
  Brain.Screen.setCursor(8, 4);
  Brain.Screen.print("Right Intake Motor: %.0f%%%%",
  IntakeMotorRight.temperature(pct));
  Brain.Screen.setCursor(9, 4);
  Brain.Screen.print("Conveyor Motor: %.0f%%%%", ConveyorMotor.temperature(pct));
  Brain.Screen.setCursor(10, 4);
  Brain.Screen.print("Sorter Motor: %.0f%%%%", SortingMotor.temperature(pct));
}
```

motor.temperature(pct) returns the data I need for this

%%%% is needed to print one % as seen above

4/3/21
M.M.

Function to get and print motor temperatures

date: 4/3/21

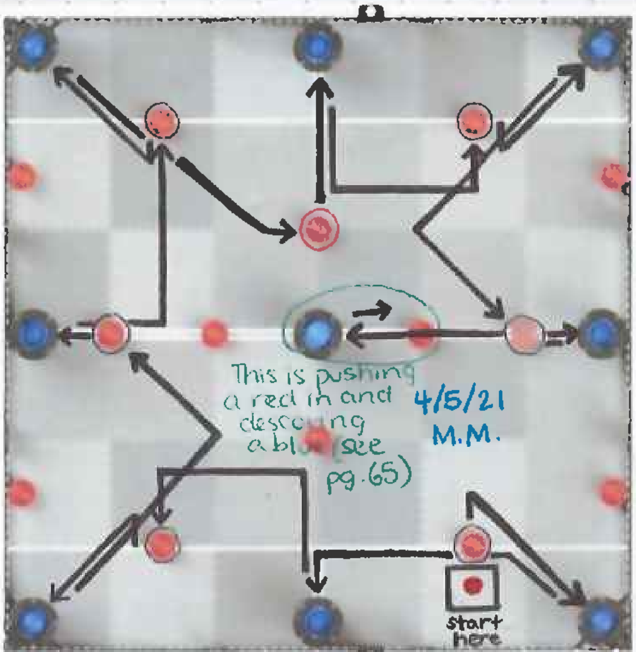
Megan M. 4/3/21

4/5/21

AUTONOMOUS SKILLS STRATEGY

GOAL: Plan a path that will be high scoring, yet consistent.

GOAL COMPLETED 4/5/21



Autonomous Skills Path

- Building on the program for 50 points (if works perfectly) that I used for my last competition (see pg. 43 for field diagram), this is the logical path.

This path is conducive for using more trigonometry (see pg. 49-52)

- Assuming I have time to score one red in each outside goal, descore the 12 blue balls from the outside goals, and push one red ball into the center (descoring a blue in the process), the score would be:

$33 - 2 + 63 = \boxed{94 \text{ points}}$

This seems unrealistic to complete in one minute and rows are worth more points

So, if I keep the same path, but don't descore any blue balls from the outside goals, the score would be:

$33 - 14 + 63 = \boxed{82 \text{ points}}$

This is more doable in one minute, but risky, because if the robot misses a red ball, there could easily still be blue rows

- Based on these scores, and the challenges with each, I will have two options for autonomous skills, both with the same path and scoring (see above). One will descore along the way, and the other won't descore.

I added another Brain Screen autonomous ~~selector; this time for skills~~

project designed by: witnessed by:

"Descore" scores and descores; "center" only scores in hopes of making it to the center goal by the end of the run

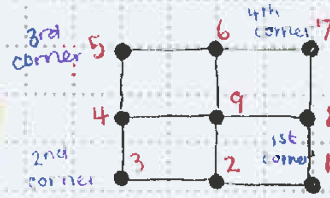
date: 4/5/21

Megan M 4/5/21

4/6/21

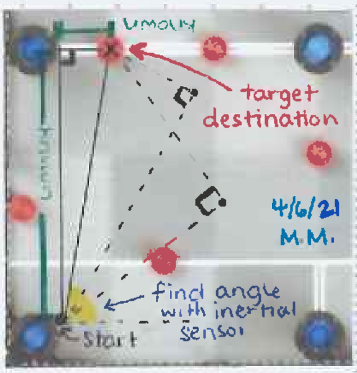
WORKING ON AUTONOMOUS SKILLS

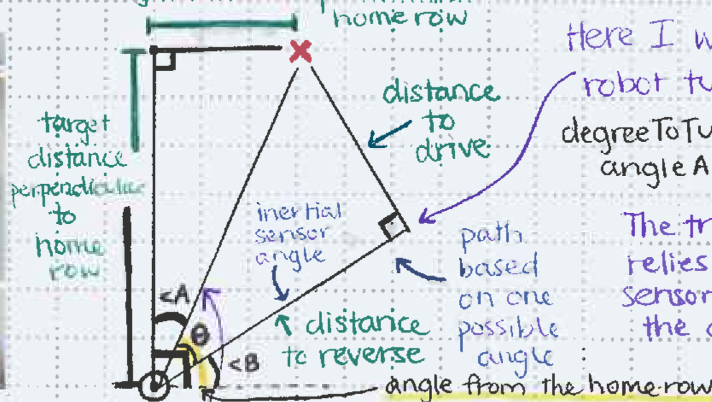
GOAL: Program the autonomous skills path based on the strategy plan from 4/5/21 (see pg. 85) with the option to descore as well as score, or only score.



INCORPORATING MORE TRIGONOMETRY:

- So far I've used trigonometry to move consistently from the first goal to the second goal, and from the second goal to a ball. The next step in the run is from the second corner goal to the ball in front of the fourth goal.





Here I will have the robot turn 90°

degreeToTurnTo = angleAtGoal + 90

The trigonometry relies on the inertial sensor turning to the correct angle

distance from second corner goal to ball

I can find the distance from the second goal to the ball using the Pythagorean theorem:

Assuming the corner of the field is a perfect right angle:

$$\boxed{\theta = 90^\circ - (\angle A + \angle B)}$$

$$\boxed{\text{distance from second corner goal to ball} = \sqrt{\left(\text{target distance parallel to home row}\right)^2 + \left(\text{target distance perpendicular to home row}\right)^2}}$$

Using this information, I can calculate the distance To Reverse and Drive:

$$\boxed{\text{distanceToReverse} = \text{distance from second corner goal to ball} \times \cos\theta}$$

$$\boxed{\text{distanceToDrive} = \text{distance from second corner goal to ball} \times \sin\theta}$$

I can get <B from the inertial sensor.

$$\boxed{\angle A = \tan^{-1}\left(\frac{\text{target distance parallel to home row}}{\text{target distance perpendicular to home row}}\right)}$$

project: Math functions:

$\sqrt{\ }$ = sqrt()

$\#^2$ = pow(#, exponent; in this case 2)

designed by:

witnessed by:

date: 4/6-8/21

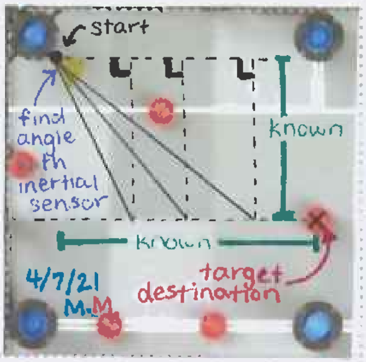
Megan M 4/6/21

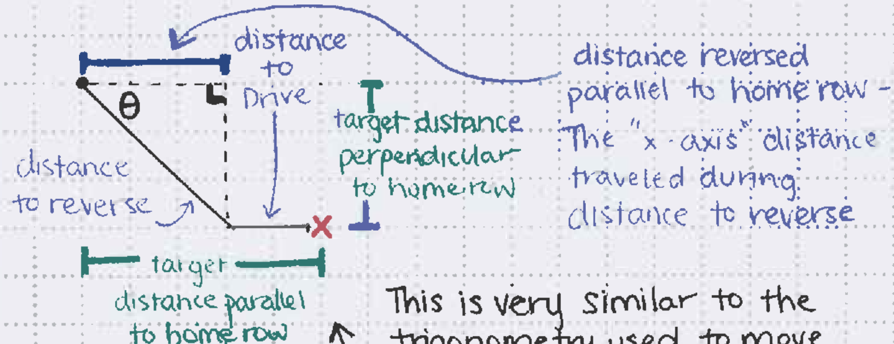
4/6/21 System Engineers are concerned with the overall process of defining, developing, operating, maintaining and the replacement of systems. Where other engineers concentrate on the details of a specific aspect of a system the system engineers are concerned with the integration of all of these aspects into a coherent and effective system.

WORKING ON AUTON. SKILLS CONTINUED

INCORPORATING MORE TRIGONOMETRY CONTINUED:

- Moving from the fourth goal to the third corner is fairly simple, no trigonometry needed here. But moving from the third corner to the sixth goal after picking up a ball needs some trig.





I can get θ from the inertial sensor

This is very similar to the trigonometry used to move from the first goal to the second goal (see pg. 49-50)

$\sin\theta = \frac{\text{opposite}}{\text{hypotenuse}}$

$\tan\theta = \frac{\text{opposite}}{\text{adjacent}}$

$\boxed{\text{distance to Reverse} = \text{target distance Perpendicular to home row} \,/\, \sin\theta}$

$\boxed{\text{distance reversed parallel to home row} = \text{target distance Perpendicular to home row} \,/\, \tan\theta}$

$\boxed{\text{distance To Drive} = \text{target distance parallel to home row} - \text{distance reversed parallel to home row}}$

I can use this to find how much further I need to have the robot drive

- The trigonometry from the sixth to the seventh goal after picking up a ball is the same as the trigonometry from 3/9/21 (see pg. 50-51).

- The trigonometry from pg. 86 also works on the fourth corner.

***NOTE:** sin(), cos(), and tan() functions require radians!

$\text{degrees} \times \left(\frac{\pi}{180}\right) = \text{radians}$

$\text{radians} \times \left(\frac{180}{\pi}\right) = \text{degrees}$

- Distance in inches must be in degrees when passed to the drive function!

$\text{inches} \times \left(\frac{360^\circ \text{ per rev.}}{\text{circumference of wheel} \cdot \sqrt{2}}\right) = \text{encoder degrees}$

$\text{circumference of wheel} = \underbrace{3.25 \text{ in}}_{\text{diameter}} \times \pi$

project designed by: witnessed by:

date: 4/6-8/21

Megan 4/7/21

4/8/21

WORKING ON AUTON. SKILLS CONTINUED

FIELD DIFFERENTIATION:

- Using the trigonometry (see pg. 49-52, 86-87) I was able to finish programming the autonomous skills path from pg. 85.

- Based on what is selected on the Brain Screen in pre-auton (see pg. 85), a boolean variable descore is assigned either true or false. I use this flag to know what needs to happen (thus, what steps to run) at each goal.

  the flow will skip over steps with descoring if descore is false.

- The Problem: the trigonometry gets the robot to the specified location in relation to the field perimeter. The balls are placed based on the field tiles, so if the tiles and perimeter are not exactly centered or square, the robot will tend to miss the ball.

  To help reduce the effects of this, I added some correction code so that if there is no ball in the conveyor when the robot reaches the goal, it does not attempt to score, saving time.

  - Because the robot can make it to the goals on any side of the field with the trigonometry (only has issues picking up the balls), I will add a third option if I have time: just descoring.

    This isn't as many points, but I would rather have a consistent, lower scoring routine than an unreliable one that fails and ends up scoring less.

Megan M 4/8/21

4/9/21

OCCASIONAL SCORING PROBLEM

GOAL: Eliminate, or at least reduce, the number of times a ball fails to shoot with enough force to make it in the goal.

GOAL COMPLETED 4/10/21

- **The Problem:** Often in autonomous, rarely in driver, when a ball is shot, it doesn't go far enough to make it into the goal. Instead, it bounces back from the rim of the goal onto the top conveyor roller where the scoring distance sensor (see pg. 33) detects it.
  → this causes my scoreAndDescoreAuton function (see pg 80-81) to execute incorrectly.

- **My Solution:**
  - Replacing the rubberbands on the conveyor
  - Replacing the anti-slip mat on the hood
  - Adding a small section of polycarbonate to the hood behind the anti-slip to provide slightly more compression and traction on the ball.
  - Correction code in my Two-goal and one-goal cycling autonomous routines.

New polycarbonate on the hood

→ Materials used on hood: ←
- 1 - 3.5" by 3.5" polycarbonate section
- 4 - 0.375 in. screws
- 4 - thin nylocks

```
void correctionForBallNotMakingItIntoTheGoal() {
  if(correctionForFailedScoring == false) {
    wait(500, msec);
    if(DistanceScoring.objectDistance(mm) <= distanceScoringBallSeen &&
    DistanceScoring.isObjectDetected() == true) {
      sorterSpin(false, 600);
      startTime = Brain.Timer.value();
      waitUntil((DistancePos2.objectDistance(mm) < distancePos2Unoccupied &&
      DistancePos2.isObjectDetected() == true)
      || Brain.Timer.value() >= startTime + 0.8);
      sorterLock();
      conveyorLock();
      correctionForFailedScoring = true;
    }
  }
}
```

take the ball back down to position two and try again

4/10/21 M.M.

flag that it has corrected once, can no longer be called again

This code only runs once so that it doesn't keep doing the same thing over and over if it is unable to score.

These greatly reduce the problem.

date: 4/9-10/21

Megan M 4/10/21

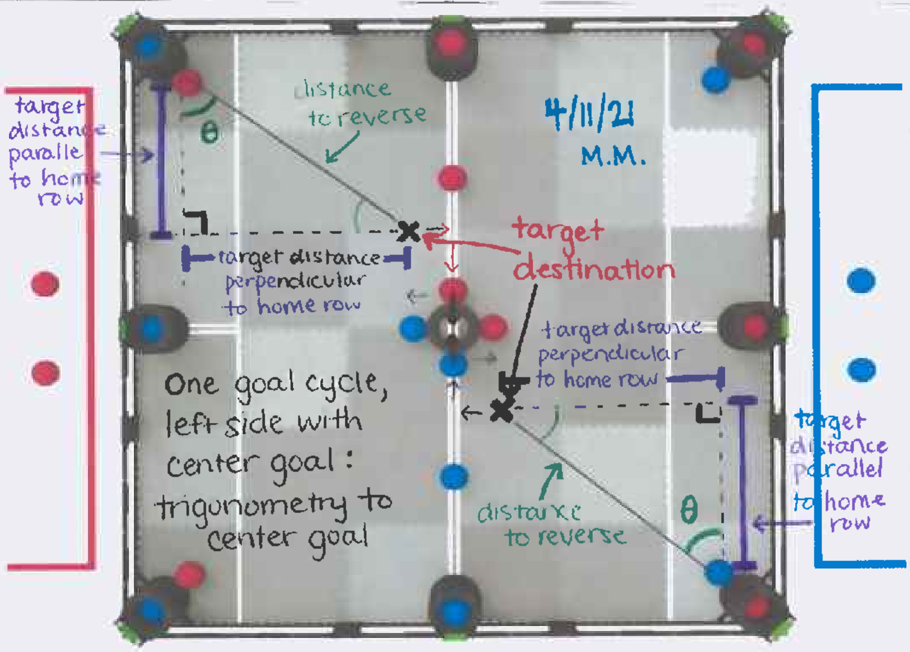
4/11/21

PROGRAMMING CENTER GOAL AUTON.

GOAL: Write an autonomous program (building on one-goal cycle) that scores a ball in the center goal.

GOAL COMPLETED 4/12/21

- After measuring, I found that is not feasible for my robot to reliably descore a ball from a side goal on the center row, so now the option for pushing a ball into the center goal will be the only center row option (and only available on the left side)



- Using trigonometry to get the robot to the center goal is very simple compared to previous motions (see pg. 49-52, 86-87)
- This time, however, I don't base the calculations on the angle of the robot at the goal - I determine what angle the robot needs to be at in order to reach the target destination and then correct to that angle.

Like I did on pg. 86, I can use the Pythagorean Theorem to find the length of the hypotenuse:

$$\text{distanceToReverse} = \sqrt{\left(\begin{array}{c}\text{target distance}\\ \text{perpendicular}\\ \text{to home row}\end{array}\right)^2 + \left(\begin{array}{c}\text{target distance}\\ \text{parallel to}\\ \text{home row}\end{array}\right)^2}$$

target Angle at goal

$$\theta = \tan^{-1}\left(\frac{\text{target distance Perpendicular to home row}}{\text{target distance parallel to home row}}\right)$$

Then, I use θ and convert it to the angle the inertial sensor needs to correct to. ⟶

$$\text{inertial angle to correct to} = 90 - \left(\theta \cdot \left(\frac{180}{\pi}\right)\right)$$

convert θ to degrees from radians

project designed by: witnessed by:

date: 4/11-12/21

Megan M 4/11/21

4/12/21

PROGRAMMING CENTER GOAL AUTON. CONT.

- After running the routine with the trigonometry on the previous page, followed by generic drive-for-a-certain-number-of-encoder-degrees function, I found that the robot did not end up in the exact same spot when run on both sides of the field (scored the center both times, but crossed the autonomous line once).

- To reduce the risk of forfeiting the autonomous bonus by crossing the autonomous line, I added two line trackers, one on the front of each front wheel base structures.

Two new line trackers

Attached to the very end hole on each side

Line trackers work best when ≈ 3 mm from the surface,* so I lowered them from the base with $\frac{1}{8}$ in. spacers (one on each side)

Materials used to mount line trackers:

- 2 - line trackers
- 2 - $\frac{1}{8}$ ~~nylo~~ in. spacers
- 2 - nylon washers
- 2 - 0.625 inch screws
- 2 - thin nylocks

- I kept the trigonometry (see previous page) the same, but for the short drive to the tape (autonomous line), movement is based on the level of reflectivity of the line trackers (higher = lighter surface *).

```
double lineTrackerReflectivityThreshold = 60;
  Stop driving forward if either sensor sees tape
if(LeftLineTracker.reflectivity() > lineTrackerReflectivityThreshold
|| RightLineTracker.reflectivity() > lineTrackerReflectivityThreshold
|| Brain.Timer.value() >= startTime + 2) {
  drivetrainStop();
}
```

This is another waitTime failsafe (see pg. 53)

4/12/21 M.M.

- The line trackers consistently stop the robot at the right point on both sides of the field, as long as the wheels are moving slow (50 rpm works well) so that it detects the tape in time.

* from the VEX Knowledge Base

4/13/21

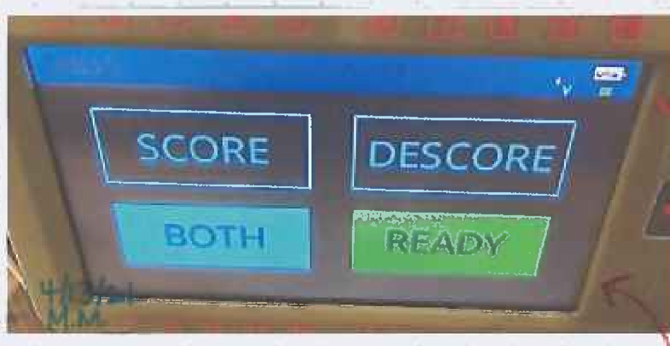
FINAL PROGRAMMING FOR STATE

GOAL: Finish any programming not yet completed so I have as many consistent and high scoring routines as needed.

GOAL COMPLETED 4/19/21

AUTONOMOUS SKILLS:

- I redid the Autonomous Skills Brain Screen Selector from 4/5/21 (see pg. 85) to have three options:
  1. Score only
  2. Descore only
  3. Score and descore (Both)



Autonomous Brain Screen ~~See~~ selector for Skills with "Both" and "Ready" selected

- I realized that in order to program the descore option (see pg. 88), it would take a lot of time as the positioning is different. It also wouldn't be many points, plus the red balls would likely interfere.
  - ↳ So, I think the best bet is going to be "Both", scoring and descoring.

15 SECOND AUTON. HOME ROW OPTION:

- Due to the limited time left until State, I don't have enough time to do both right and left side options for completing the home row, but hopefully this won't be a big deal in the event that I need to use this routine, so I am only working on the right side.

- The code is very similar to before (see pg. 52 and 63), but tweaked with the new drift correction (see pg. 78-79).

- To save a few milliseconds, I no longer have the balls' colors be identified in autonomous. If there is a ball, it will just score it. If there's no ball, it immediately continues on in the routine.

Megan T. 4/14/21

4/15/21

DRIVER SKILLS FROM 3/25 to 4/15

Date	Total Score	Rows Owned
3/25/21	113	8
	114	8
	114	8
	115	8
	113	8
	115	8
	115	8
	117	8
3/28/21	104	6
	104	6
	105	6
	103	6
3/29/21	101	6
	101	6
	103	6
	119	8
	101	6
	102	6
	116	8
3/30/21	104	6
	102	6
	104	6
	102	6
	113	8
4/1/21	104	6
	116	8
	102	6
	118	8
	94	5
	103	6
	104	6
4/15/21 M.M.	116	8
	103	6

Date	Total Score	Rows Owned
4/5/21	118	8
	101	6
	104	6
	116	8
	105	6
	117	8
	100	6
4/6/21	116	8
	117	8
	101	6
	114	8
4/7/21	115	8
	111	8
	115	8
4/8/21	109	8
	114	8
	116	8
	100	6
	101	6
	103	6
	116	8
	101	6
	99	6
4/9/21	117	8
	113	8
4/10/21	118	8
4/11/21	114	8
	120	8
	101	6
4/15/21 M.M.	105	6
	101	6

My goal for State is 120 points. I've been working on a modified path to get more points (see pg 94)

project designed by: witnessed by:

date: 3/25 – 4/15/21

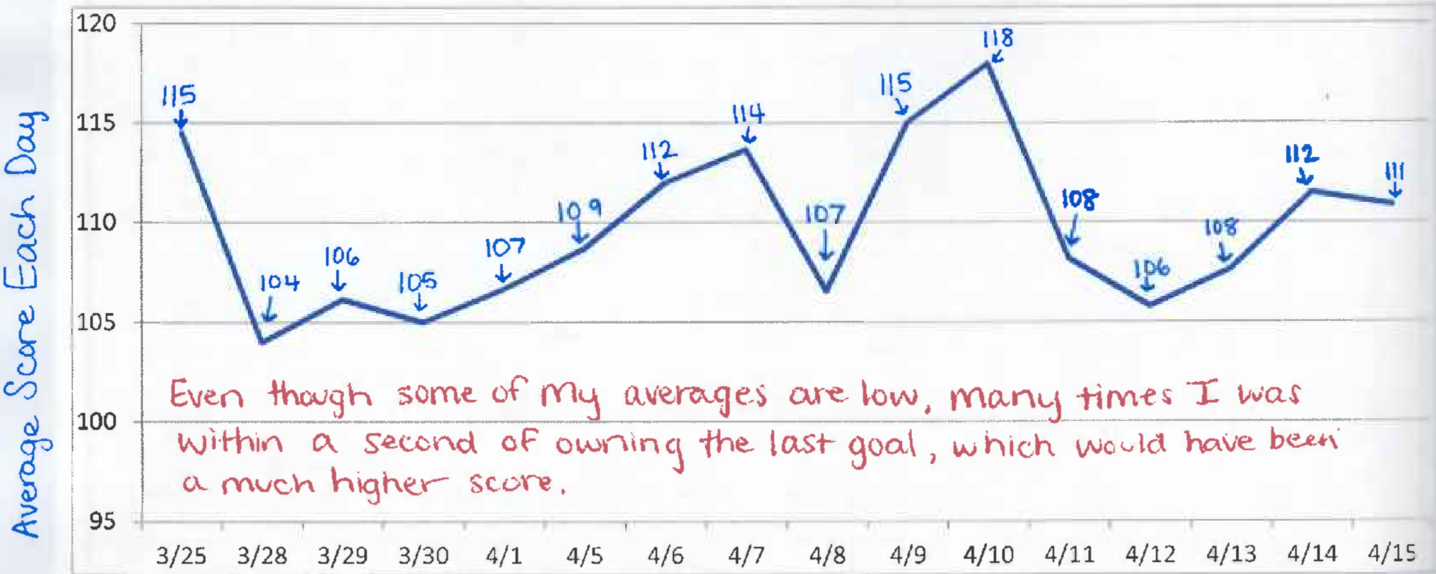
Megan M. 4/15/21

4/15/21

DRIVER SKILLS FROM 3/25 to 4/15 CONTINUED

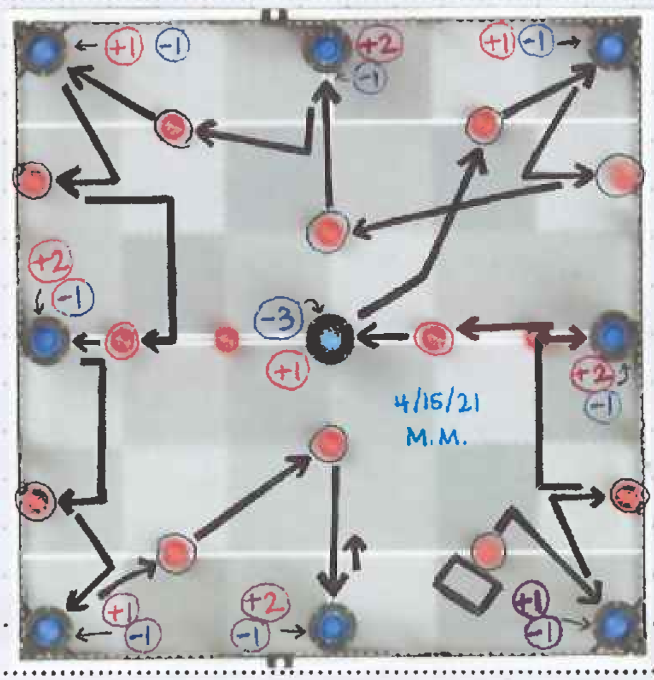
Daily Average Driver Skills Scores Over Time

4/15/21
M.M.



Date	Total Score	Rows
4/12/21	104	6
	105	6
	114	8
	104	6
	102	6
4/13/21	117	8
	105	6
	101	6
4/14/21	115	8
	115	8
	119	8
	102	6
	99	6
	119	8
4/15/21	115	8
	99	6
	118	8
	104	6
	116	8
4/15/21 M.M.	100	5
	116	8
	119	8

Days Practiced



This is the path I've been aiming for: this is 120 points; 4 scored blue balls and 2 unscored red balls remain.

ned by:

witnessed by:

date: 3/25 - 4/15/21

Megan T 4/15/21

4/15/21

LIVE REMOTE SKILLS ANALYSIS

MY GOALS FOR THE LIVE REMOTE SKILLS-ONLY TOURNAMENT:

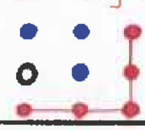
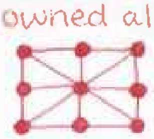
- [x] Become familiar with the process of LRS (in hopes of qualifying for Worlds)
- [x] Score higher than my top Driver Skills at a competition (score more than 115 points - that's what I scored at Bolt Up, see pg. 57)
- [ ] In autonomous skills, score 75+ points (this is what it would get if it works perfectly up until the time runs out, owning and descoring six goals)

[Completed 4/15/21]

10703Z Stats:

- Total skills score of 190 points
- Ranked 4th out of 22 teams

Live Remote Skills Results 4/15/21 M.M.

Driver / Auton	Score	Analysis
Auton run #1	73	scoring and descoring. Successfully descored  7 blue and scored 5 red. missed descoring one blue and picking up one red. ran out of time before 7th goal
Driver run #1	117	owned all 8 rows, descored 8 blue, scored  13 red. Got hung up on descoring first goal, but recovered. Need to work on speed.

- Because I scored higher in autonomous and driver skills than at my other competitions on the first attempt, I opted to not use my last two attempts of each inorder to have more time to prepare for the State Championship.

*This skills score bumped me up to 3rd in the state of Tennessee in the World Skills Standings; hopefully I can increase my score even more at the State Championship. ~M.M. 4/15/21

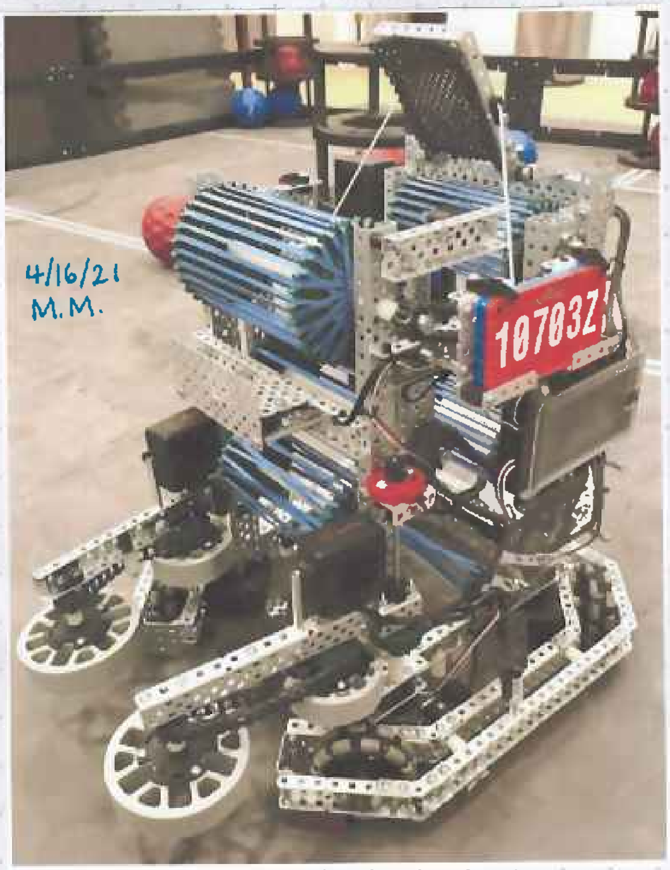
project designed by: witnessed by:

date: 4/15/21

Megan M 4/15/21

4/16/21

10703Z for the TN State Championship



4/16/21
M.M.

SUCCESSES:

- A number of macros for more efficient scoring and driving - Collect Ball, Score Ball with Score Two option, Score And Descore, and dispose Ball
- Drift correction using the drivetrain motor encoders and the inertial sensor's accelerometer
- A variety of 15 sec. autonomous routines to work with any team: one goal cycle, right and left; one goal cycle with scoring the center goal (left side only); two goal cycle, right and left; home row, right side only
- The incorporation of trigonometry to increase consistency in the robot's positioning
- Actively using 9 sensors (4 kinds), excluding motor encoders

MY GOALS FOR STATE:

- ☑ Get a second interview.
- ☒ Score at least 75 points in autonomous skills
- ☑ Score higher than 117 points in Driver Skills
- ☑ Rank in the top third of teams after qualifications

THINGS TO IMPROVE:

- Higher scoring and more consistency for autonomous skills
- More driving practice
- Higher scoring and more routine options for 15 sec. autonomous

Completed 4/17/21

project

designed by:

witnessed by:

date: 4/16/21

Megan M 4/16/21

4/17/21

TN STATE CHAMPIONSHIP ANALYSIS

10703Z Stats:

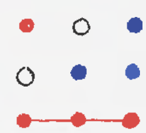
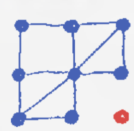
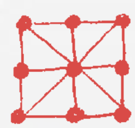
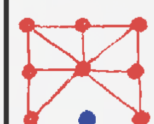
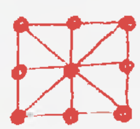
- Ranked 9th after qualifications
- Ranked 1st in skills, with a total score of 185 points
- 6 wins, 4 losses after qualifications

- WP = 16
- AP = 48
- SP = 74

4 WP from completing the home row in autonomous

- OPR = 9.2
- DPR = 4.2
- CCWM = 5.0

TN State Championship Skills Results 4/17/21 M.M.

Driver / Auton	Score		Analysis
Auton. run #1	$\boxed{67}$		scoring and descoring: scored 4 red balls, descored 9 blue balls. missed picking up 2 red balls and ran out of time before 7th goal
Auton. run #2	19		scoring and descoring: scored 1 red ball, got stuck descoring first goal. Stopped early
Driver run #1	117		scored 9 red balls, descored 12 blue balls. scoreTwo macro failed several times.
Driver run #2	105		Scored 10 red balls, descored 11 blue balls within a second of scoring last goal for 117+ points
Driver run #3	$\boxed{118}$		scored 10 red balls, descored 12 blue balls; just barely scored last goal in time.

- Because I was already 1st and did not anticipate my autonomous skills scoring higher than 67, I opted to not run my third attempt for autonomous skills and give my motors time to cool down.

project

designed by:

witnessed by:

date: 4/16-17/21

Megan M. 4/17/21

4/17/21

TN STATE CHAMPIONSHIP ANALYSIS CONTINUED

TN State Championship Qualification Match Results 4/17/21 M.M.

Alliance Partner	Opponents	Win / Loss	Score	Analysis
73973B	97934W 24816T	Win	8-5	Right, 2 goal cycle auton works perfectly in the first goal but fails to shoot second not very fast, good strategy
9364B	663A 98709C	Win	10-5	Right, 2 goal cycle auton works perfectly good scoring and speed decent strategy
92715A	9364A 663E	Loss	5-32	Left, 2 goal cycle auton fails to cycle corner goal, works on second goal poor speed and strategy
97934Y	24816T 97934U	Loss	11-13	Left, 2 goal cycle auton works perfectly decent strategy, poor speed against defense
57249A	37215A 9364D	Loss	13-14	Left, 2 goal cycle auton works perfectly decent speed, decent strategy
663C	24816M 19589A	Win	43-5	Right, 2 goal cycle auton works perfectly good speed, good strategy and scoring
9364G	5999A 5999J	Win	11-9	Right, 2 goal cycle auton works perfectly good speed and decent strategy
97934Z	12876B 96504R	Win	18-3	Right, 2 goal cycle auton leaves 1 red in corner goal, works on second goal good speed. Left intake chain broke.
663B	9364C 5999S	Loss	11-18	Left, 2 goal cycle auton works on first corner goal, fails to shoot second good strategy, decent speed
98709A	9364E 24816V	Win	10-7	Left, 2 goal cycle auton works perfectly decent speed, good strategy

date: 4/16-17/21

Megan M 4/17/21

4/17/21

TN STATE CHAMPIONSHIP ANALYSIS CONTINUED

Autonomous Routine Success Rates:

- Left, two goal cycle : used 5 times
  - fully worked: 60% (3 times)
  - partially worked: 40% (2 times)
  - completely failed: 0%
- Right, two goal cycle : used 5 times
  - fully worked : 60% (3 times)
  - partially worked : 40% (2 times)
  - completely failed: 0%
- Right, one goal cycle : used 2 times
  - fully worked : 100%
  - partially worked : 0%
  - completely failed : 0%
- I did not use Left, one goal cycle with or without the center goal, or the home row routine, or my no auton. option.

TN State Championship Elimination Match Results

4/17/21
M.M.

Alliance Partner: 97934X

Opponents	Win / Loss	Score	Analysis
5999J 57249A	Win	24-5	R16 #3-1 Right, one goal cycle works perfectly good speed and strategy good teamwork
24816T 9364B	Loss	6-11	QF 2-1 Right, one goal cycle works perfectly poor strategy, good speed

- I won the Excellence award and Robot Skills Champion, which double-qualifies me for the VRC HS World Championship!

WHAT I LEARNED:

- Odometry helps deal with field differentiation, but it doesn't fix it - several teams with odometry still struggled with autonomous skills, just like those without odometry.

project designed by: witnessed by:

date: 4/16-17/21

Megan T 4/17/21

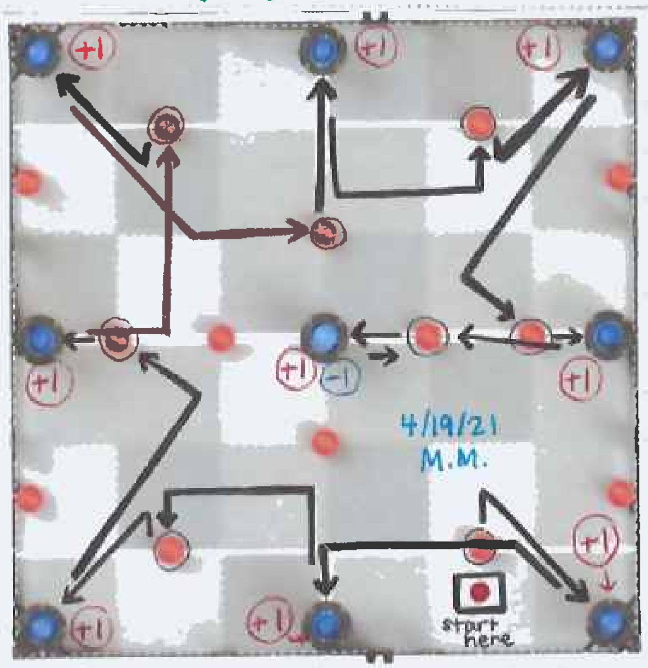
4/19/21

PLAN TO IMPROVE FOR WORLDS

- Improvements to make:
  - ☐ Higher driver skills score: PRACTICE!!!
  - ☐ Higher scoring autonomous skills score: WORK ON CONSISTENCY AND SPEED!!!
  - ☐ Work on completing this path for 106 points in autonomous skills.

- I'm opting for the LRS option over LRT for Worlds.

Goal for LRS Worlds autonomous skills path



TIMELINE UNTIL WORLDS

Calendar Key:
- days with time to work on the robot
- competition

Programming
Autonomous Skills
4/20 - 5/19

April

S	M	T	W	T	F	S
	today			1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

May

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Driving Practice
Driver Skills runs
4/20 - 5/19

April

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

May

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

4/19/21 M.M.

DEADLINES:
- April 29 - Live Remote Skills Tournament
- May 20-22 - VEX Robotics HS World Championship

project | designed by: | witnessed by:

date: 4/19/21

Megan M 4/19/21

4/20/21

WORKING ON AUTONOMOUS SKILLS

GOAL: Tune the autonomous skills routine for more consistency as well as increase the speed in order to work toward scoring 106 points for LRS Worlds. **GOAL COMPLETED 4/29/21**

- Unfortunately, I have not had as much time to work on robotics as I had anticipated, but I did work some.
- I tweaked the inches passed into the trigonometry functions and the angles to which the robot corrects at each goal.
  - Because Worlds is Live Remote, I can now program with higher precision because field differentiation is no longer a factor.
- In order to increase the speed of the routine, I lowered the percentages of acceleration and deceleration on certain motions. I found that the movement from the second corner to the ball in front of the fourth goal (see diagram on previous page; also on the same move from the fourth corner goal to the ball in front of the eighth goal) is the mostly likely place for the robot to miss the ball.
  - the trigonometry I use here is different than moving from other goals and relies on the robot turning exactly 90° in order for the distance to the ball to be correct (see pg. 86)

This is probably the cause of the inconsistency I may change the trigonometry here to be closer to the other goals

Date	Total Score	Rows Owned
4/25/21	118	8
4/26/21	103	6
	121	8
4/27/21	104	6
	104	6
	105	6
	116	8
4/28/21	120	8
	116	8
	120	8
4/29/21	117	8
	119	8

4/29/21 M.M.

DRIVER SKILLS PRACTICE SCORES

witnessed by:

date: 4/20-29/21

Megan M 4/29/21

4/29/21

LIVE REMOTE SKILLS ANALYSIS

MY GOALS FOR THE LIVE REMOTE SKILLS-ONLY TOURNAMENT:

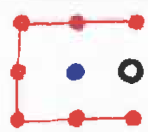
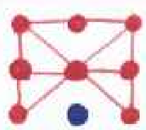
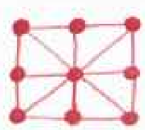
- [x] Become comfortable with the process of LRS for Worlds.
- [x] Score at least 120 points in Driver Skills.
- [x] Score more than 73 points in Autonomous Skills (my current highest score at a competition, see pg. 95)

[Completed 4/29/21]

10703Z Stats:

- Total skills score of 198 points
- Ranked 4th out of 17 teams

Live Remote Skills Results 4/29/21 M.M.

Driver / Auton	Score	Analysis
Auton run #1	75	 score (not score and descore). scored 7 red balls, descored 2 blue balls. missed picking up one red ball, ran out of time at eighth goal.
Driver run #1	105	 scored 12 red balls, descored 9 blue balls. got hung up picking up red balls. within a second of scoring last goal.
Driver run #2	123	 scored 13 red balls, descored 14 blue balls. good speed. highest score ever up to this point!

- My total score increased by 8 points (190 from the now finalized robot skills World rankings) to 198.
- I opted not to use my last driver attempt or the last two programming attempts, as I doubt I would be able to improve either score.

project designed by: witnessed by:

date: 4/29/21

Megan T. 4/29/21

5/1/21

INCREASING AUTONOMOUS SKILLS SCORE

GOAL: Continue working on autonomous skills so that it consistently scores 106 points for LRS Worlds.

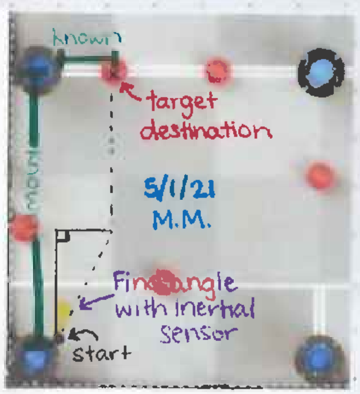
GOAL COMPLETED 5/21/21

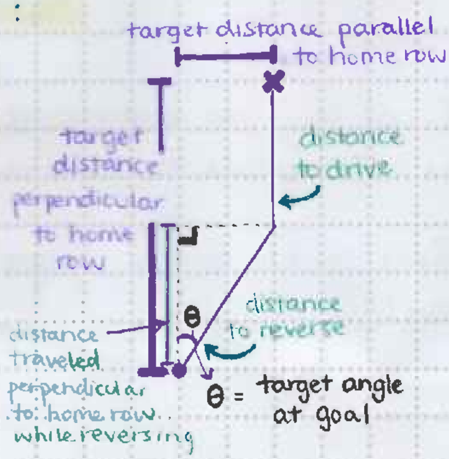
- For the LRS World Championship, I plan to use the "score" option on the Brain Screen. "Both" is scoring and descoring, and is a backup option.
- I removed the "descore" only option from 4/13/21 (see pg. 92).

5/1/21 M.M.

Here is the LRS WC brain screen with "score" and "ready" selected.

USING A DIFFERENT TRIANGLE FOR THE TRIGONOMETRY FROM THE HOME ROW RIGHT CORNER GOAL TO THE BALL ON THE AUTONOMOUS LINE:





- Whereas the previous triangle used here (see pg. 86) determined the path based on the angle at the goal, this triangle determines the angle at the goal needed to follow the specified path.

distance traveled perpendicular to home row while reversing = target distance perpendicular to home row / 2

This number can be adjusted

how much of the distance between the start and the destination to reverse vs drive forward

$$\text{distance to reverse} = \sqrt{\left(\begin{array}{c}\text{target distance}\\\text{from goal parallel}\\\text{to home row}\end{array}\right)^2 + \left(\begin{array}{c}\text{distance traveled perpendicular}\\\text{to home row while reversing}\end{array}\right)^2}$$

I used the Pythagorean theorem to find the distance to reverse

project

designed by:

witnessed by:

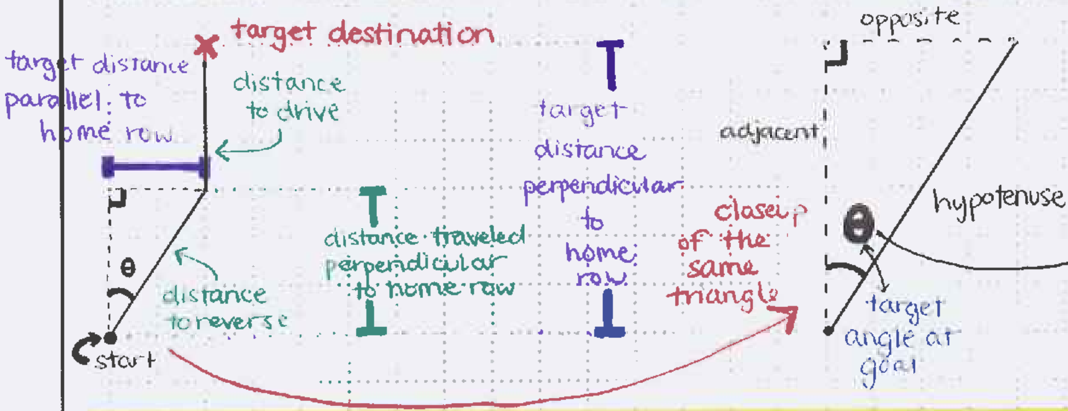
date: 5/1/21

Megan T 5/1/21

5/1/21

INCREASING AUTONOMOUS SKILLS SCORE CONT.

USING A DIFFERENT TRIANGLE FOR TRIGONOMETRY CONTINUED:



$\tan\theta = \frac{opp.}{adj.}$

$$\boxed{\theta = \tan^{-1}\left(\frac{opp}{adj}\right)}$$

angle at goal needed for trig. distances to be correct

$$\boxed{\text{distance to drive} = \text{target distance perpendicular to home row} - \text{distance traveled perpendicular to home row while reversing}}$$

$$\boxed{\theta \text{ or target angle at goal} = \tan^{-1}\left(\frac{\text{target distance parallel to home row}}{\text{distance traveled perpendicular to home row while reversing}}\right) \times \left(\frac{180}{\pi}\right)}$$

[atan is the function for tan⁻¹ in the code]

This converts radians to degrees

- From this trigonometry, I now have the distances for the robot to reverse and drive, as well as the angle needed for these distances.

Now, I need to have the robot correct to the target angle found above with the trigonometry:

```
double angleAtGoal = Inertial.yaw();
if(angleAtGoal < targetAngleAtGoal) {
  DriveMotorLeftBack.spin(directionType::fwd,75,velocityUnits::rpm);
  DriveMotorRightBack.spin(directionType::rev,75,velocityUnits::rpm);
  waitUntil(Inertial.yaw() >= targetAngleAtGoal);
}
else if(angleAtGoal > targetAngleAtGoal) {
  DriveMotorLeftBack.spin(directionType::rev,75,velocityUnits::rpm);
  DriveMotorRightBack.spin(directionType::fwd,75,velocityUnits::rpm);
  waitUntil(Inertial.yaw() <= targetAngleAtGoal);
}
drivetrainStop();
```

Depending on the orientation of the robot (i.e., which direction it is facing), you may have to take the absolute value and/or subtract 180° to get the value you need here.

Only using the back wheels to rotate around the goal

5/1/21 M.M.

project: designed by: witnessed by:

date: 5/1/21

Megan T 5/1/21

5/1/21

FINAL AUTONOMOUS TWEAKS / DRIVER PRACTICE SCORES

GOAL: Finish preparing for LRS Worlds, both in programming and in practicing driving. **GOAL COMPLETED 5/21/21**

- If the robot misses a red ball, I have it programmed to descore the blue ball(s) in that goal in order to break the blue row, EXCEPT on the last side goal before moving to the center goal.
  - ↳ There's no time to dispose the blue ball and it could interfere with scoring the center.
- For driver skills, I've been practicing with automatic sorting set to OFF. This way I choose where to dispose the blue balls so as to leave the red balls in their initial starting positions.

Date	Total Score	Rows Owned
5/1/21	119	8
	109	6
	106	6
5/3/21	106	6
	121	8
5/4/21	106	6
	106	6
	120	8
	119	8
5/5/21	121	8
	101	5
	105	6
5/6/21	106	6
	109	6
	101	6
	122	8
5/7/21	117	8
	109	6
	109	6

5/21/21 M.M.

Date	Total Score	Rows Owned
5/16/21	107	6
	121	8
	121	8
	119	8
5/17/21	112	8
	107	6
	119	8
	119	8
	121	8
5/18/21	121	8
	121	8
	119	8
5/19/21	115	8
5/20/21	119	8
	120	8
	121	8
5/21/21	122	8
	122	8

5/21/21 M.M.

project designed by: witnessed by:

date: 5/1-21/21

Megan M 5/21/21

5/21/21

LRS WORLD CHAMPIONSHIP ANALYSIS

MY GOALS FOR LRS WORLDS:

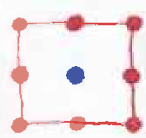
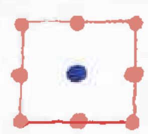
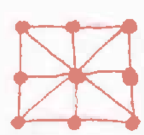
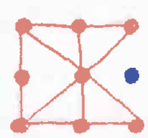
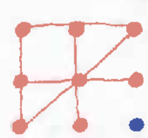
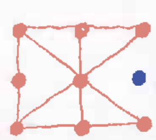
- [x] Score at least 120 points in driver skills. [Completed 5/22/21]
- [x] Score 106 points in autonomous skills.

Venus division (second highest skills division)

107037Z Stats:

- Total score of **213** points
- Ranked 6th out of 46 teams in my division

Live Remote World Championship Skills Results 5/22/21 M.M.

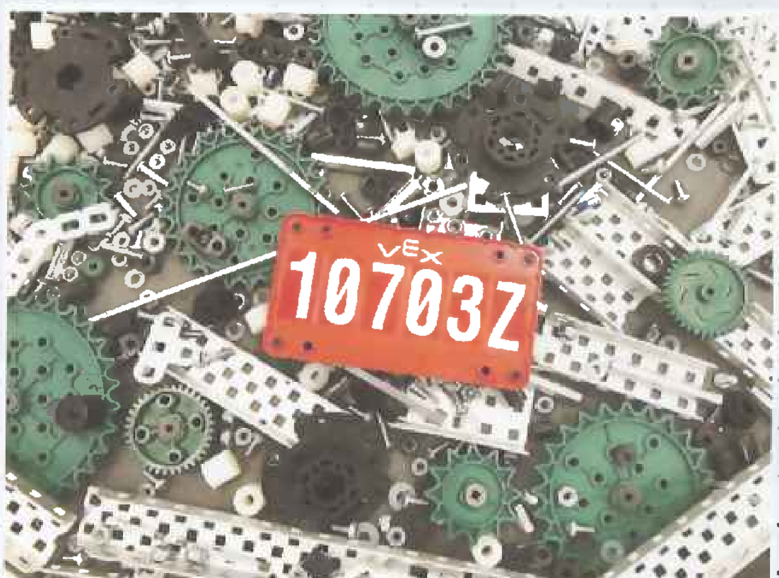
Driver / Auton	Score	Analysis
Auton. run #1	80	 worked perfectly up until the center goal. missed the red ball, which got in the way of descoring the center.
Auton. run #2	81	 same as Auton run #1, but managed to descore 1 blue ball from the center goal for 1 point.
Auton run #3	$\boxed{106}$	 worked perfectly all the way around! almost missed the red ball to be scored in the fourth goal
Driver run #1	$\boxed{107}$	 So close to scoring that last goal. scored 10 red balls, descored 13 blue balls
Driver run #2	100	 my macros failed once or twice. took ~~two~~ too long on the center goal. red ball got jammed in corner
Driver run #3	106	 accidentally disposed the red ball to be scored in the final goal. had to re-cycle the first corner.

- I won the Excellence award for the Venus division!

5/29/21

DISASSEMBLING THE ROBOT

5/29/21 M.M.



5/29/21 M.M.

project

designed by:

witnessed by:

date: 5/29/21

Megan M 5/29/21