Robotics 2020-2021
DFHS
7432C

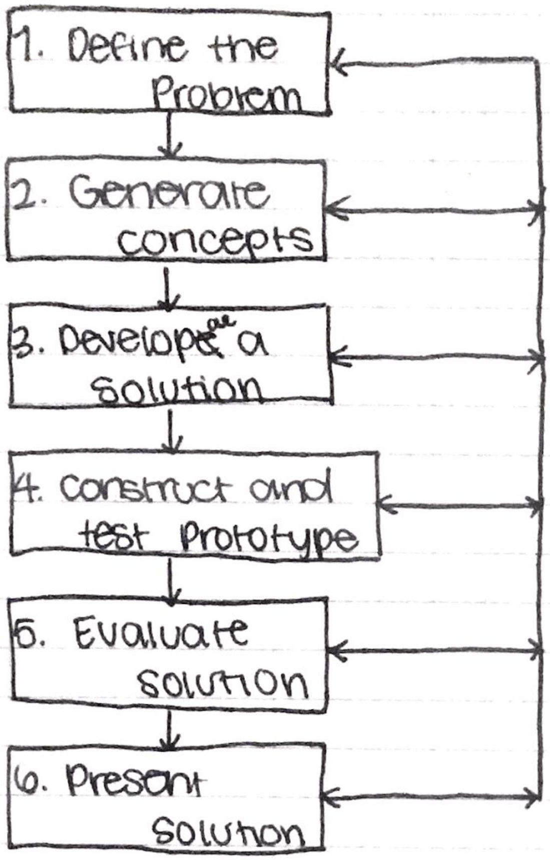
PLTW | Engineering

TABLE OF CONTENTS

TABLE OF CONTENTS

DATE	Page Title	PAGE #

DATE	PAGE TITLE:	PG #
8/30/20	The Design Process	1



Engineering can be defined as the application of practical and scientific knowledge to the solving of a problem through the use of a methodical process. The methodical Process engineers use to solve problems is the Design Process (pictured to the left). Although there is no ~~wide spread~~ universally accepted single version of the design process, our team uses this six step version. As we build our robot we will refer to this process, and as we come across problems, or things we need to improve, we will go back to Previous steps to ameliorate our original design.

Define the problem – includes problem being identified, validated, and justified. Design requirements are determined, and design Brief is created.
↳ most important step of the design process

Generate concepts - researching past and current solutions, and brainstorming Possible new solutions. Selecting approach using a Decision Matrix.

Develop a Solution – includes creating detailed sketches, technical drawings and CAD software models.

Construct and Test Prototype - building a testable robot. Includes testing it for performance, usability, durability. Data will be collected.

Evaluate a Solution - includes evaluating solution's effectiveness and making plans for improvement of the design

Present Solution - our solution (robot) will be presented to judges and will be used in competition

NAME:	CONTINUED ON PG #
August Horton	

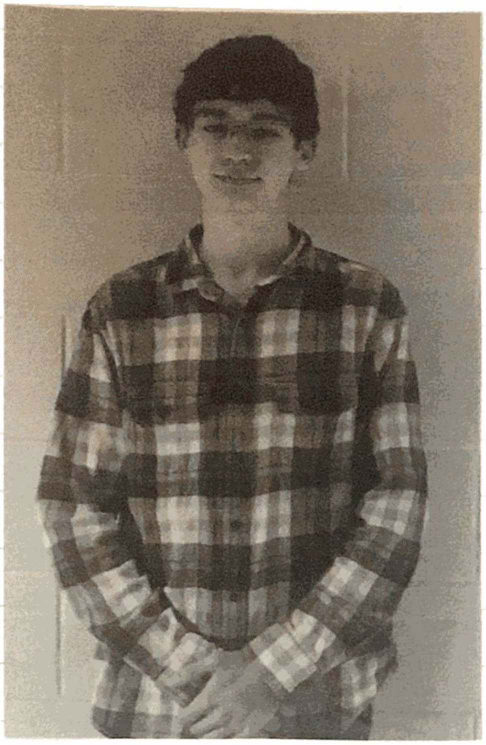
Continued from Page

Design Brief

Design Team: ae

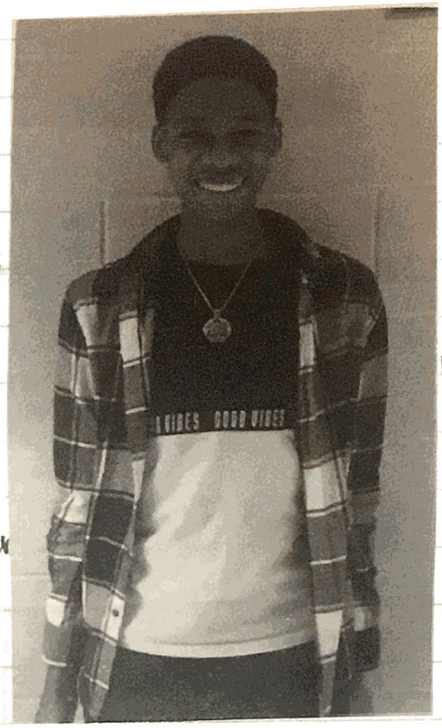
ae

ae



Eric Mitchell:
grade: 12
Age: 17
role on team: team leader, Programmer, CAD
others: part of robotics for 4+ years, PLTW experience: IED, POE, part of STEM, computer programming classes

ae



Trenton Brooks:
grade: 10
age: 15
role on team: builder, designer
others: part of robotics 3 years, PLTW IED, middle school PLTW

ae

ae

Eli Bryson:
grade: 11 age: 16
role on team: Builder, designer, ~~3 years of robotics~~ ae
others: part of robotics for 3 years, PLTW IED, STEM, computer programming classes

ae

August Horton:
grade: 10
age: 15
role on team:
Engineering notebook
others: part of robotics 3 years, PLTW IED, middle school PLTW, STEM

ae

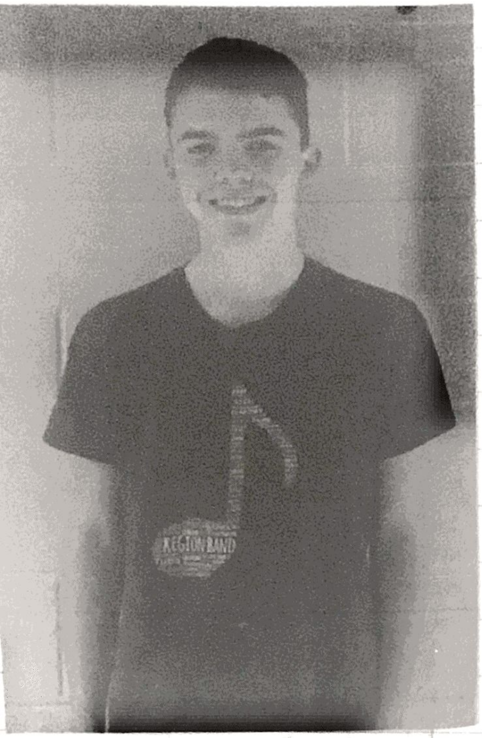
Continued to Page

DATE 9/12/20

PAGE TITLE: Design Brief

Continued from Page

Design team(continued):



Lance williams:

grade: 10

age: 15

role on team: builder, designer, Programmer

others: part of robotics 4 years, PLTW IED, middle school PLTW, STEM Program, computer programming classes

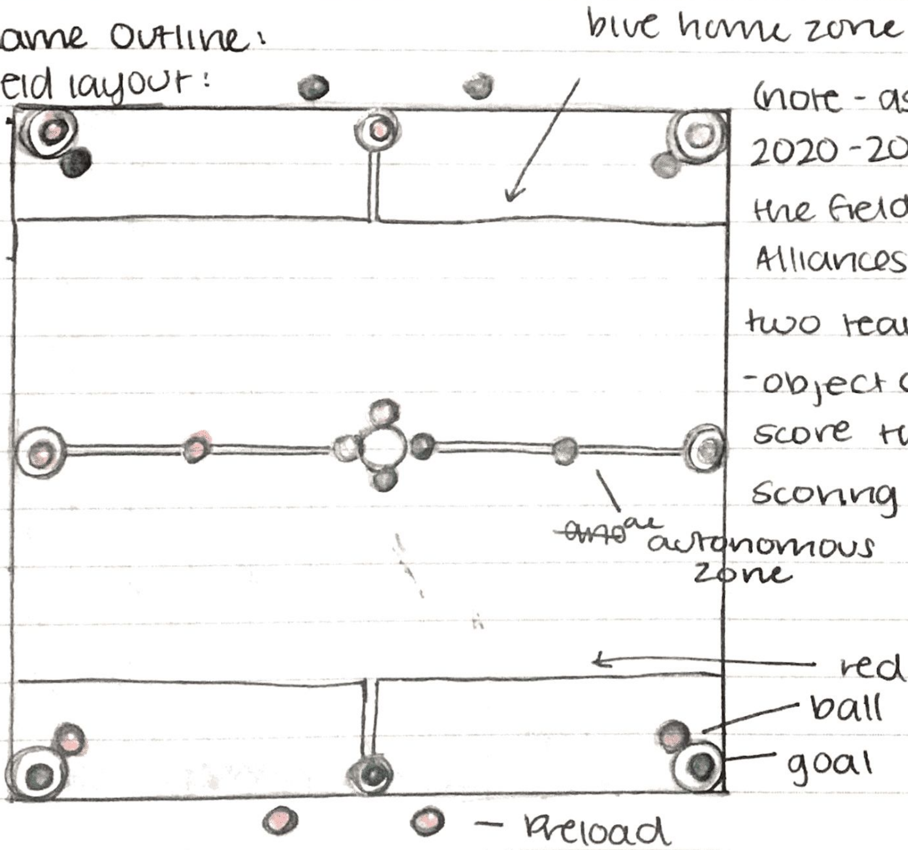
ae

ae

ae

Game Outline:

Field layout:



(note - aspects of field are not to scale)

2020-2021 game: matches are played on the field pictured to the left. There are two Alliances - one red, one blue - composed of two teams each compete in each match.

- object of game is to attain a higher score than the opposing alliance by scoring balls and connecting rows

ae

Continued to Page

NAME: August Horton

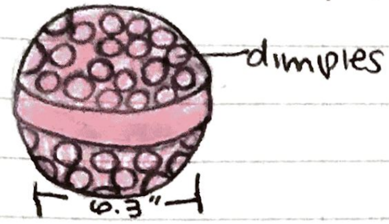
CONTINUED ON PG #4

Continued from Page

Game Outline (continued):

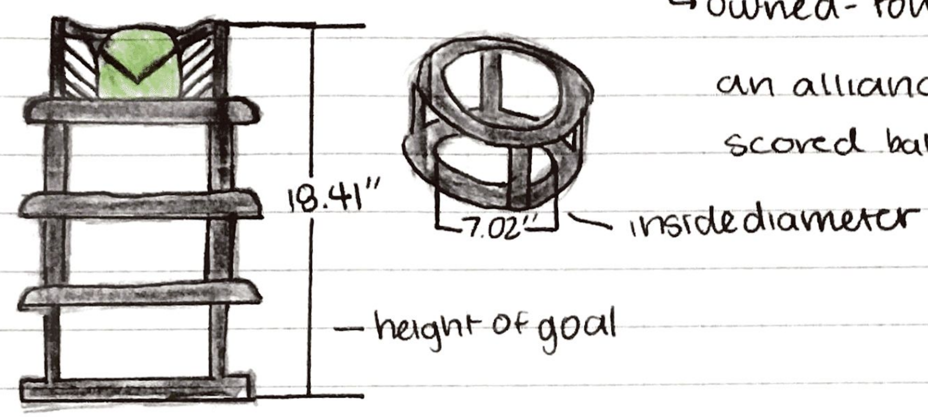
Field components:

Ball - hollow, dimpled - 32: 16 red balls, 16 blue balls, can be scored in goals
↳ if three balls of same color are in 3 goals in one line → connected row



Goal - cylinders where balls can be scored. consists of 4 retaining rings and 4 PVC pipes. Outer edge of ring = outer edge of goal, upper edge of top ring is the upper edge of the goal

↳ owned - tower is considered owned when an alliances colored ball is the highest scored ball in the goal



Home zones (Pictured in field pic) - one for Blue alliance and one for red - where robots start the match and defines the alliance's home row - defined by outer edge of field and tape

↳ Rows - diagonal, vertical, or horizontal line/ connection between 3 goals

Problem statement:

attain the highest score by scoring balls in goals and connecting rows of goals

ae

Continued to Page

Continued from Page

ae

criteria and constraints:

- 15 second autonomous, 1 minute 45 second driver control period
- 18x18x18 in starting robot size
- no easily detatchable / loose parts
- only use 9 or less motors
- no blocking/trapping - robots or balls

ae

Robot components:

- ball intake
- ball storage
- chasis
- ball ejector/shooter
- ball descoring
- hood to prevent balls from shooting upword - top of robot

offensive ~~strategive~~ ae strategies:

- beneficial ball removal
  - ↳ ex: removing bottom ball to provide more space for our ball, or knocking out the opposing alliance's ball to help our alliance win that goal
- ball placement in goals
  - ↳ focusing on this part of the game could be more beneficial because it would help our alliance score as many balls as possible
- maneuverability:
  - ↳ focusing on the speed and aerodynamics of the robot could help to get ~~up~~ ae us from one side to the other side of the field - this could help our team because it would help to insure that we win rows - rather than just goals individually
- using our preload in the tower closest to us
  - ↳ doing this would give the driver and programmar a ~~change~~ ae sense of normalcy - in autonomous or driver period - and would help us to start off the game strong and not w/ a bunch of slip ups

Continued to Page

defensive strategies:

Preventing our robot from tipping / helping our alliance get back up from a tip/fall

↳ it would be very beneficial to base the concepts of our robot on stable and maneverable designs. If we have an alliance tipped over, we need to be prepared to help them back up or to be able to support the team with just one robot

making sure to protect our goals towards the end of the match

↳ although we want to avoid blocking/possesion - we need to keep in mind that the balls we place are not in their final positions and we will need to protect them - especially at the end of the match

making sure our balls are scored correctly

↳ if the balls are not in the goal vertically - it will not be scored correctly and our team will have a difficult time scoring more balls in that goal/ in those goals

ae

al

al

Continued to Page

DATE 9/20/20

PAGE TITLE: Brainstorming- Chasis

Robot component - chassis

Chassis criteria

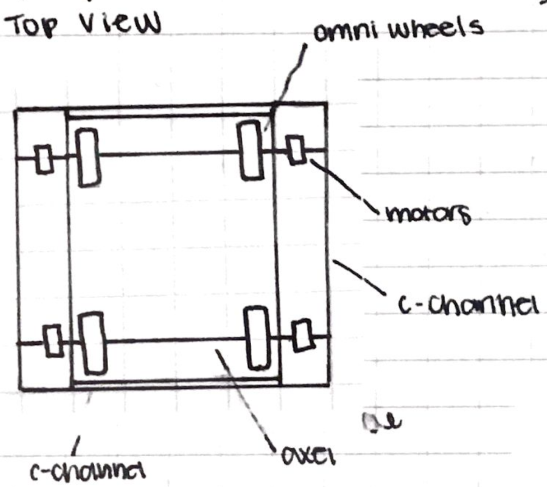
- must be less than 18" x 18" - so that arm mechanism and cube pickup can go outside of the chassis
- Lightweight - in order for our robot to get around quickly we need a lightweight chassis design that has minimal friction
- must be able to work with other robot components
- Must have enough power to last multiple rounds if needed

Chassis ideas

- Heavy square chassis
- H-frame chassis
- Enclosed H-frame chassis
- Holonomic chassis

Heavy square chassis

Top view

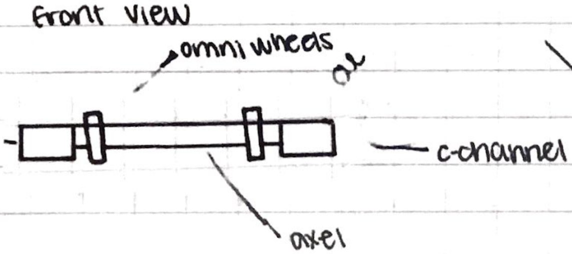


This chassis is low and heavy. It provides a sturdy base and is hard to move, because it is extremely low to the ground.

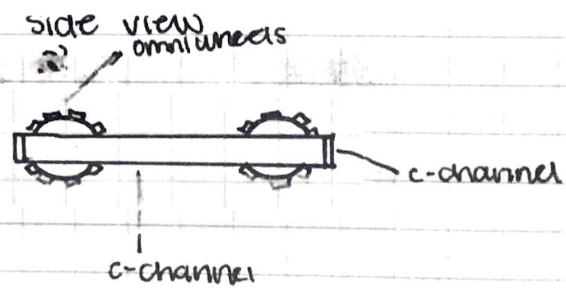
pros: hard to push, hard to flip.

cons: little manuverability, slow, very bulky.

Front view



Side view



Continued to page

NAME: August Horton

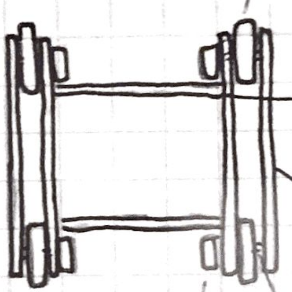
CONTINUED ON PG # 8

H-frame chassis

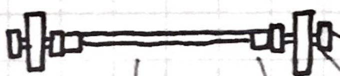
Pros: light, relatively maneuverable, good traction
Cons: not amazing maneuverability, can be flipped

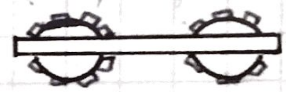
The H-frame chassis is designed to allow the driver to turn easily without too much trouble. It is also designed to not be pushed easily because its center of mass is in the center. It allows for two intakes and outakes to be on the robot

Top view



Front view





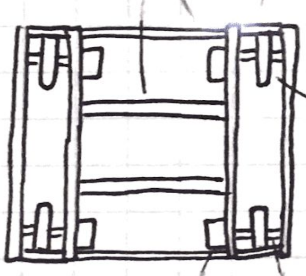
side view

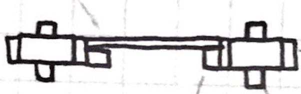
Enclosed H-frame chassis

Pros: sturdy, doesnt flip easily, good traction, relatively maneuverable
Cons: bulky, slow turning, not the most maneuverabl can be slow

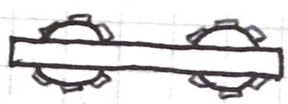
The enclosed H-frame chassis is designe to minimize robot flippage. similarly to the normal H-frame design it has space for both intakes and outakes. With the robots outside wheel protectors the robot will firmly stay in its ~~plaq~~ place.

Top view





Front view



Side view

DATE 10/5/20

PAGE TITLE: Chasis - decision matrix

PG # 9

ae

criteria → / ↓ Designs	Maneuverability 5	Lightweight 4	Not Easily Flippable 3	Space for inputs and outputs 4	overall design scores
Heavy Square Chassis	1	1	3	3	8
H-frame Chassis	4	4	2	4	14
Enclosed H-frame Chassis	2	2	3	3	10

Heavy Square Chassis

Maneuverability - 1; we gave the heavy square chassis a score of 1 on maneuverability because in order to turn it would need to stop in one place and in a game that would take too much time.

Lightweight - 1; This chassis is ~~build to~~ ae built to be a sturdy and heavy base, it is not lightweight in any way.

Not easily flippable - 3; we gave the heavy square chassis a 3 on not easily flippable because it was built for that purpose.

Space for inputs and outputs - 3; Although this design is large, it does not give ~~a~~ ae excess space for inputs and ~~oup~~ ae outputs that are needed in a game. If you look at the sketch there are c-channels surrounding all parts of the omniwheels.

Total score - 8; the lowest scoring chassis design

H-frame chassis

Maneuverability - 4; we gave the H-frame chassis a score of 4 for maneuverability because of its size ~~are~~ ae and weight. while the chassis has to stop momentarily, because of it's weight it does not take as long to turn as the Heavy square chassis or Enclosed H-frame chassis.

Lightweight - 4; This chassis is very light which is one of its highest perks. our team

Continued to page

NAME: August Horton

CONTINUED ON PG # 10

H-Frame chassis (cont.)

Not easily flippable - 2; our team gave the H-frame chassis a score of 2 for not easily flippable because it has no front omniwheel protection, therefore in a situation where the robot was in a tight space with another robot, trying to place, score, or pick up a cube it would be the first to fall over. Another factor is its weight. This chassis is relatively lightweight so it could be flipped over if bumped by another robot.

Space for inputs and outputs - 4; The team gave this chassis a score of 4 out of 4 for space for inputs and outputs because it does not have c-channels on two sides. This provides space for the inputs and outputs to be on the inside of the chassis without going over the limit of 18"x18".

Total score - 14; second highest scoring chassis design

Enclosed H-frame chassis

Maneuverability - 2; our team gave the ~~H-frame chassis~~ Enclosed H-frame chassis a 2 for maneuverability because of its weight. The chassis would have to completely stop for a considerably long period of time to turn, and then continue moving. In this ~~gamt~~ game where the robot is to score as many cubes as possible, stopping to turn is completely ~~unacceptable~~ unacceptable.

Lightweight - 2; our team gave the Enclosed H-frame Chassis a 2 for lightweight because it is very heavy, but not as heavy as the heavy square chassis, because the heavy square chassis is larger.

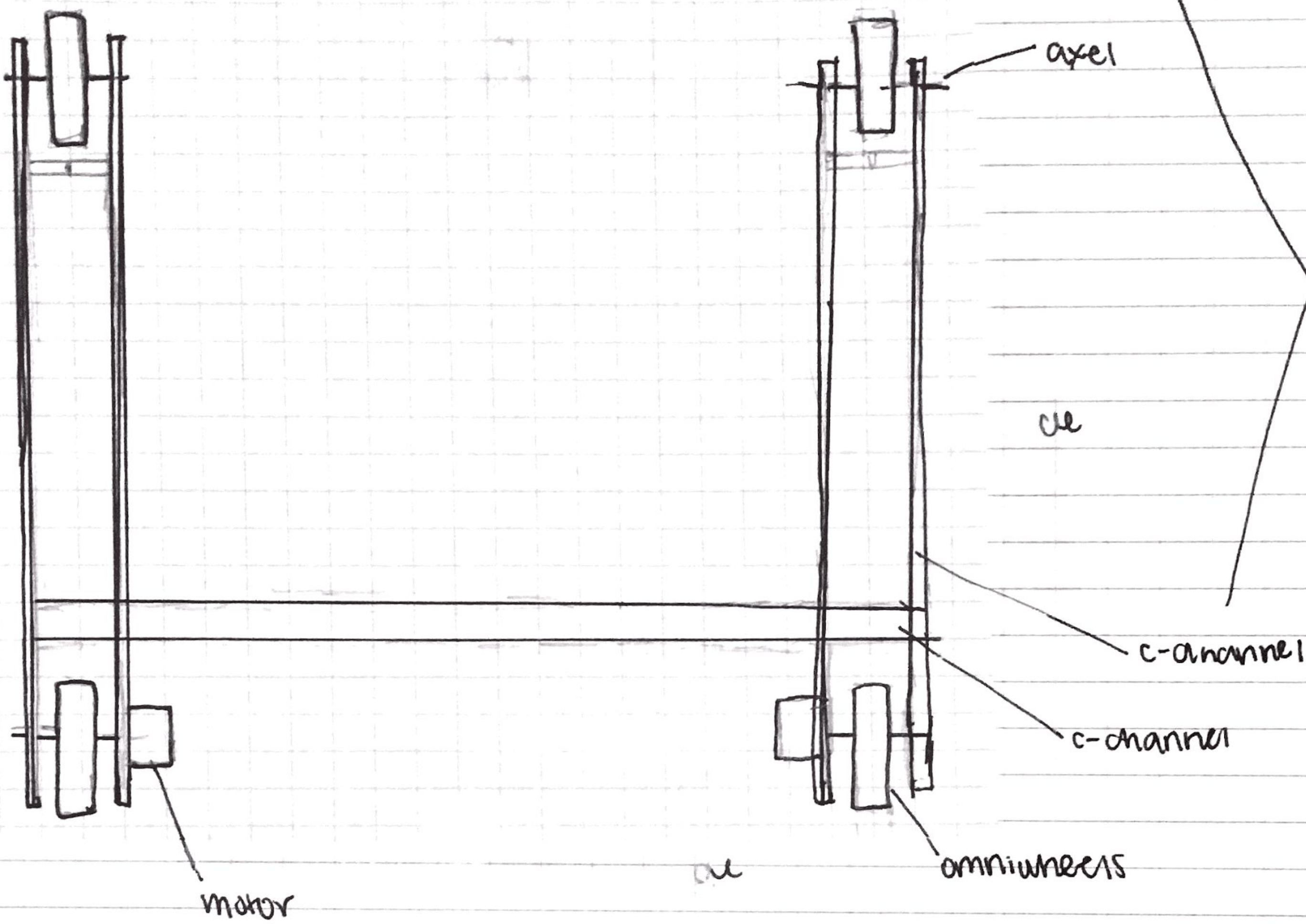
Not easily flippable - 3; We gave this chassis a 3 for not easily flippable because it has all around omniwheel protection, (reason for the name enclosed) therefore it could withstan hits by other robots, and could withstan wall hits.

Space for inputs and outputs - 3; Although this design is similar to the H-frame chassis, it has c-channels on all of its sides to prevent flippage, therefore it does not have as much space for inputs and outputs. If our team were to use this design we would most likely go outside of 18"x18"

NAME: August Horton

CONTINUED ON PG #

Top view

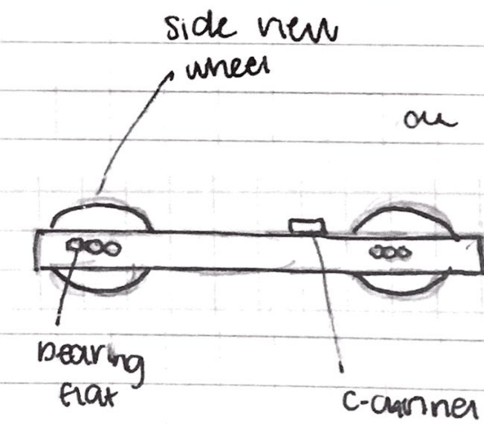


au

au

au

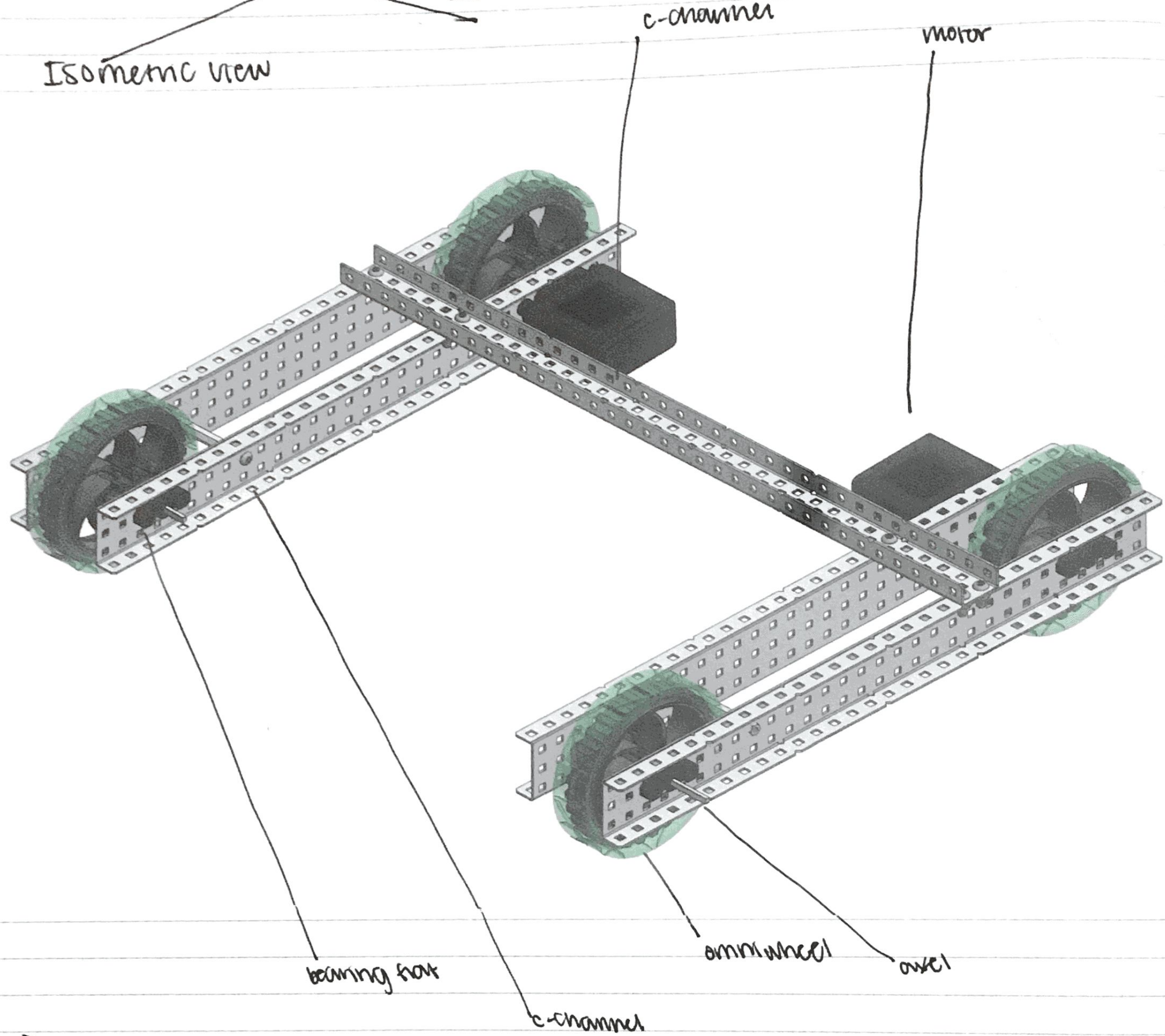
side view



au

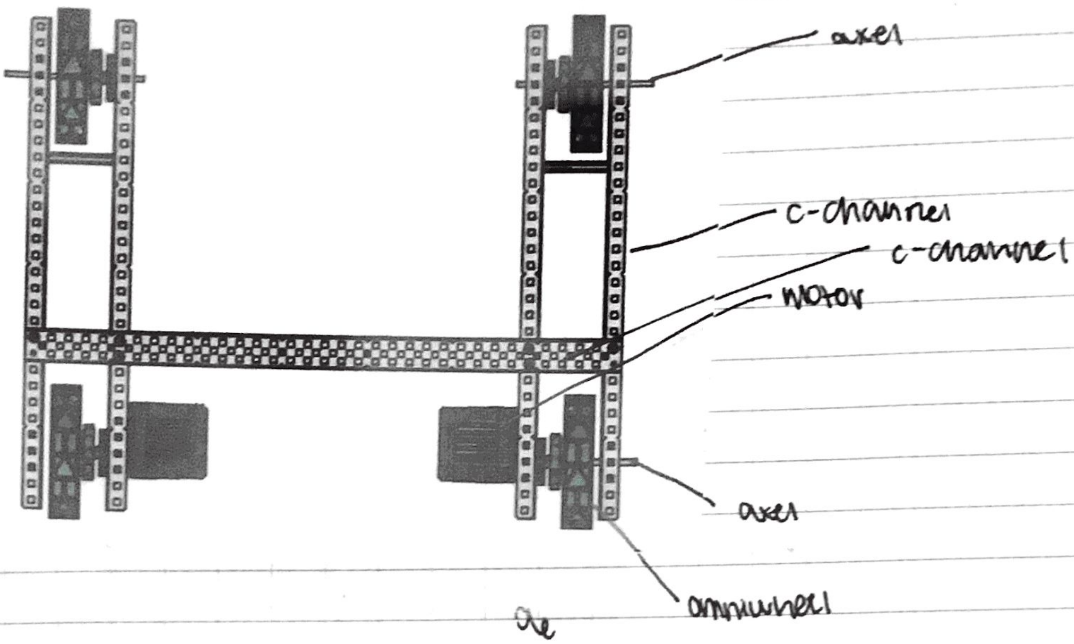
au

Isometric view



CONTINUED ON PG # 13

Top view

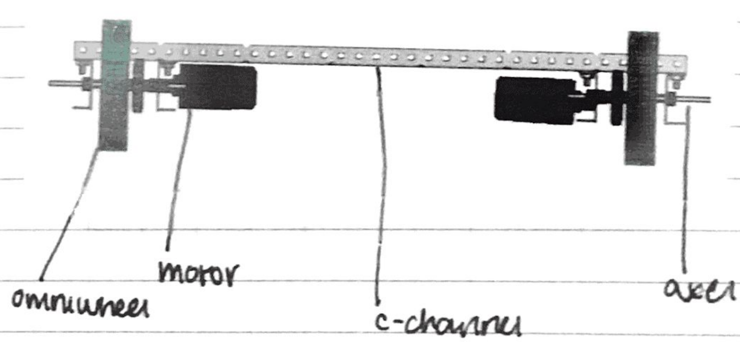


al

al

Front view

al



al

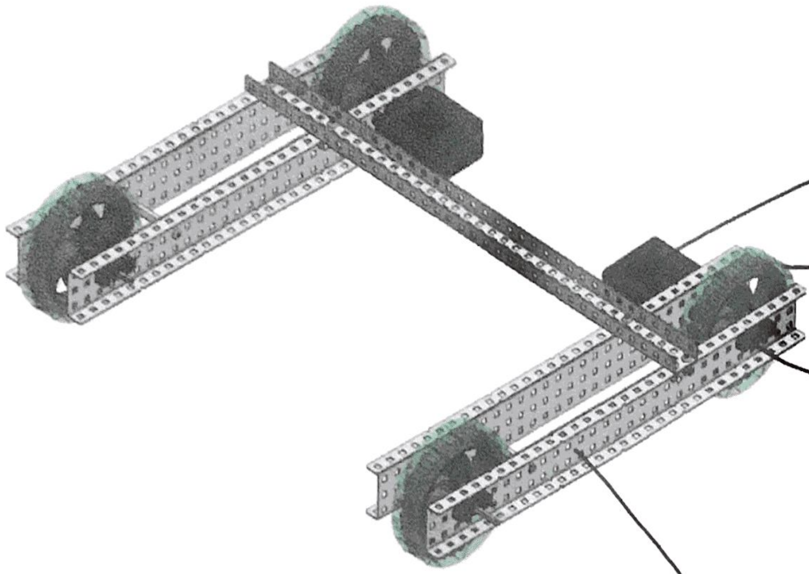
al

CONTINUED ON PG # 14

DATE 10/20/20

PAGE TITLE: Chasis CAD drawings

Isometric



motor

omniwheel

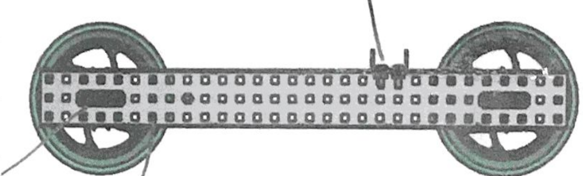
bearing flat

c-channel

al

al

c-channel



bearing flat

al

c-channel

omniwheel

al

NAME: August Horton

CONTINUED ON PG #

DATE 10/28/20 | PAGE TITLE: Chasis materials and goals | PG # 15

materials list

- c-channels
  - 2 35 hole c-channels: to support middle of the robot and create a space to build on
  - 2 23 hole c-channels: sides/outer shell of the robot, can be ~~useds~~ a used for building space e.g. towers
  - 4 8 hole c-channels: outer wheel support, outside shell/frame of chassis
  - 4 11 hole c-channels: inner wheel support, connection/support for the frame of the chassis
  - 1 cut up c-channel: used for wheel securers, helps with support
- wheels
  - 4 omniwheels
- motors
  - 4 v5 motors: used/attatched to each wheel (we need four motors as opposed to 2 because each wheel is at a different angle)
- shaft collars, standoffs, and spacers
  - 8 shaft collars: 2 used per omniwheel. ~~to~~ keeps the omniwheel in place
  - 16 spacers: 4 used per omniwheel. Used so that the omniwheel stays in the center between the two c-channels
  - 8 standoffs: used on the wheel securer so that it is raised to the top portion of the omniwheel
- bearing flats
  - 8 bearing flats: used to secure both the wheels and the axles to the 8 and 11 hole c-channels on either side of the omniwheels
- axels
  - 4 axles: used to attatch the omniwheels to the robot
- hex nuts and screws
  - 58 hex nuts: used to secure screws on many spots on the robot
  - ~~tot~~ 69 ~~t~~ screws: used to secure many things on the robot
- Angle gussets
  - 8 angle gussets: used to attatch the angled wheel frame to the rest of the chassis frame

building day 1 meeting goals

Today we need to completely build the frame of the chassis, or have it mostly completed. If possible we would like to have the entire chassis finished by the end of the day or the beginning of tomorrow, this way we can be on track with the gantt chart as we begin building our conveyor arm in the next few days. We will begin coding as soon as we finish building the chassis. Right now Eric is working on the set up of our computer, so that he is ready to code when we need him to.

Continued to page

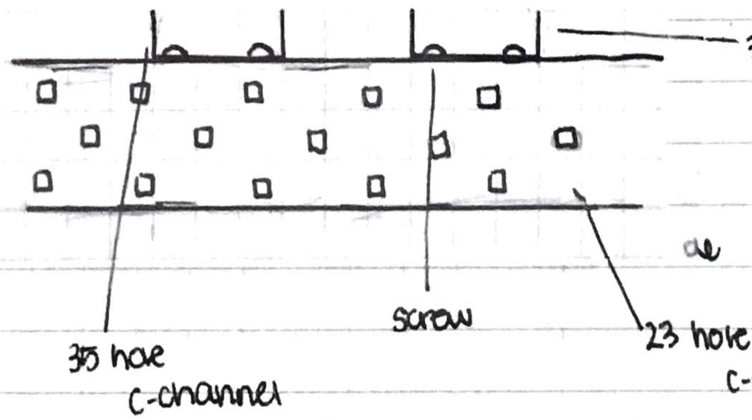
NAME: August Horton

CONTINUED ON PG #

DATE 10/28/20 | PAGE TITLE: Chasis framework | PG # 16

chassis framework - connection bars

sketch of side view-35 c-channel to 23 hole c-channel

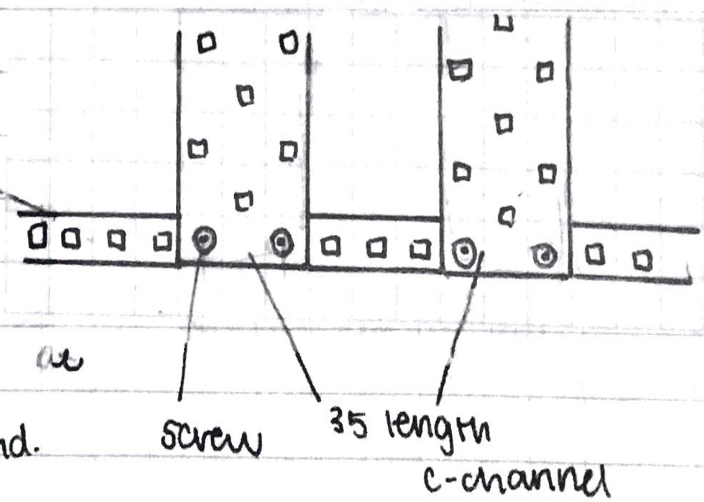


the 2, 35 hole c-channel connection bars give space for the arm and lift mechanisms. we chose to use two c-channels as opposed to 1 because two c-channels gives the lift mechanism more stabilit and is easier to build "towers" on to which is c-channel in some of our lift ideas. To connect the two 35 hole c-channels to the 23 hole c-channel we used 4 screws and 4 ~~nexs~~ hex nuts which were screwed onto the 7th and 8th holes as well as the 12th and 13th holes of the 23 length c-channels on either side (starting from the back wheels).

Top view of connection c-channels

This is just a top view of the part shown above. It is a little easier to understand how it is screwed on at this angle.

↳ this part of the chassis not only provides space for the lift ~~by~~ but it also contributes to its structure. In other words it makes the chassis more aerodymically sound.



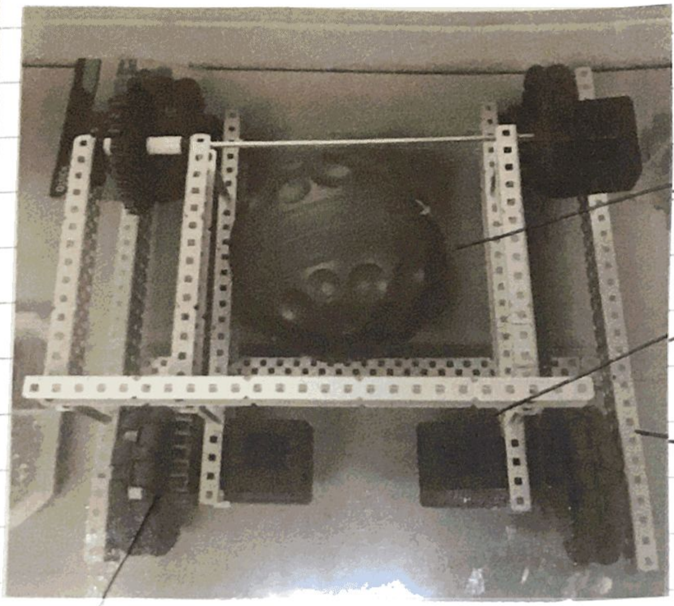
CONTINUED ON PG # 16

DATE 11/9/20

PAGE TITLE: Chasis and beginning stages of intake

Continued from Page

Today we finished up the simple H-frame chasis and attatched motors using the shafts coming from the back omni wheels → running through the chain and sprocket system and into the motor as well as standoffs, bearing flats, screws and washers to attatch these motors.



ae

In addition, we began working on the first stages of the intake, which included 1 - 35 length ae c-channel, two 15 length c-channels, standoffs, shafts, a motor (start of intake wheel), and a sprocket → we plan to create long wheel like mechanisms to create traction so that the ball will role into the holding section of the robot

↳ the spinning of these long wheel mechanisms will also hopefully keep the balls inside the robot while it is moving

our team also plans to create rotating front mechanisms made out of sprockets, chains, and flaps to capture the balls before they hit the second part of the robot → without these, the robot would be less efficient in picking up balls.

Continued to Page

NAME: August Horton

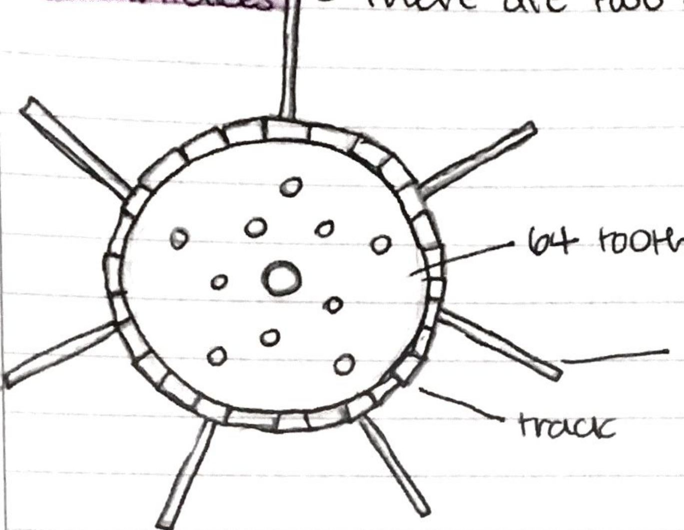
CONTINUED ON PG #

Over the past two weeks our team has made tremendous headway in the building of this robot

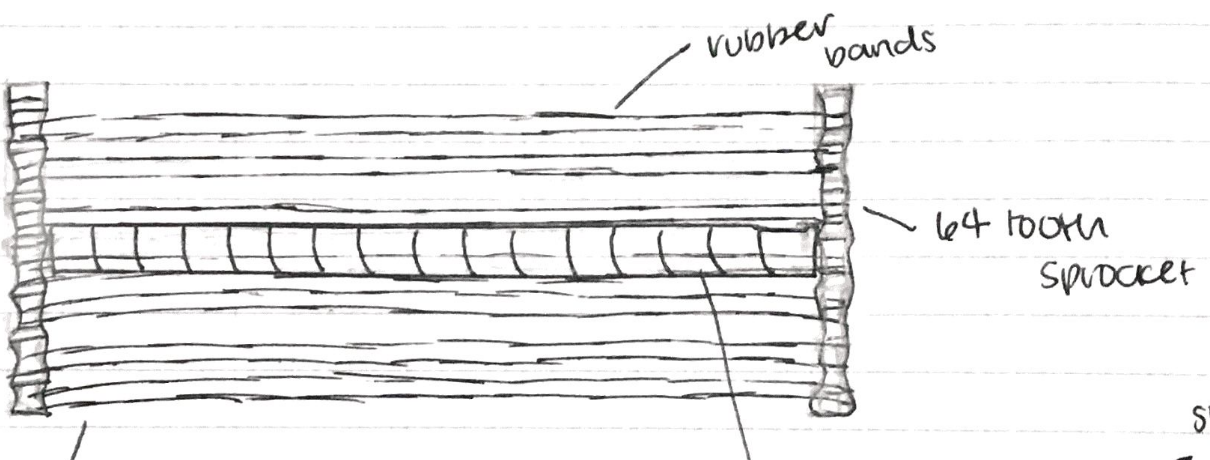
Main areas of work:

- front intakes
- vertical long rollers - with sprocket system and motors
- top rollers for ball ejection

Front intakes - there are two of these on robot



vertical long rollers - there are two of these on the robot



at first we only had rubber bands around some of the divets in the sprockets but we then found it did not provide enough traction

Standoffs on a shaft

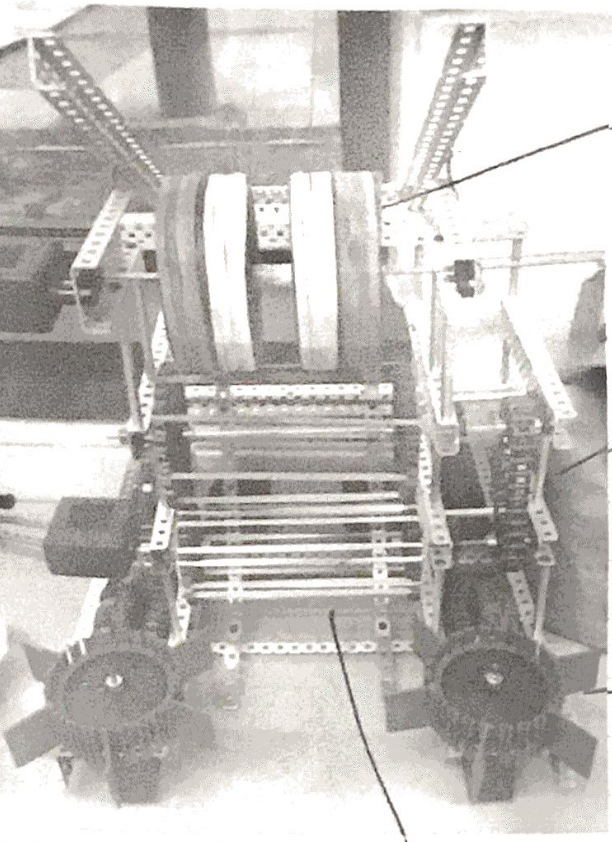
shaft then extends to connect to motor/ chain + sprocket

Continued to Page

DATE 11/26/20

PAGE TITLE: Building of intake

front view of robot up until this point



top rollers for ball ejection

sprocket system + motors

collar

12 tooth sprocket

chain

motor

shaft with roller + sp standoffs on them

c-channel

front intakes

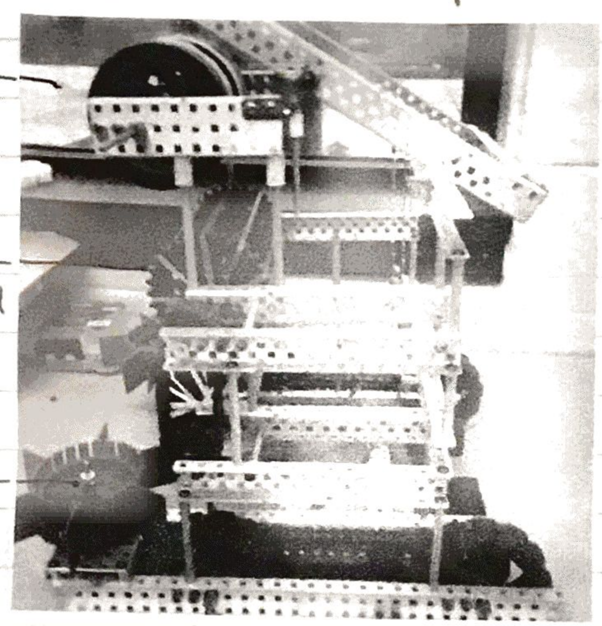
this system helps our team conserve motors (we currently have access to 6) and help the load of our programmer

ae

rollers

side view of robot up to this point

top rollers / made with default wheels and rubber bands



rollers (before we added more bands)

front intakes

ae

Continued to Page

NAME: August Horton

CONTINUED ON PG #

DATE 12/7/20

PAGE TITLE: Code


```
#include "main.h"

/**
 * Runs initialization code. This occurs as soon as the program is started.
 *
 * All other competition modes are blocked by initialize; it is recommended
 * to keep execution time for this mode under a few seconds.
 */
void initialize() {
pros::lcd::initialize();
pros::lcd::set_text(1, "7432C");
}

/**
 * Runs while the robot is in the disabled state of Field Management System or
 * the VEX Competition Switch, following either autonomous or opcontrol. When
 * the robot is enabled, this task will exit.
 */
void disabled() {}

/**
 * Runs after initialize(), and before autonomous when connected to the Field
 * Management System or the VEX Competition Switch. This is intended for
 * competition-specific initialization routines, such as an autonomous selector
 * on the LCD.
 *
 * This task will exit when the robot is enabled and autonomous or opcontrol
 * starts.
 */
void competition_initialize() {}

/**
 * Runs the user autonomous code. This function will be started in its own task
 * with the default priority and stack size whenever the robot is enabled via
 * the Field Management System or the VEX Competition Switch in the autonomous
 * mode. Alternatively, this function may be called in initialize or opcontrol
 * for non-competition testing purposes.
 *
 * If the robot is disabled or communications is lost, the autonomous task
 * will be stopped. Re-enabling the robot will restart the task, not re-start it
 * from where it left off.
 */
void autonomous() {}
```

NAME: August Horton

CONTINUED ON PG # 21

```
/**
 * Runs the operator control code. This function will be started in its own task
 * with the default priority and stack size whenever the robot is enabled via
 * the Field Management System or the VEX Competition Switch in the operator
 * control mode.
 *
 * If no competition control is connected, this function will run immediately
 * following initialize().
 *
 * If the robot is disabled or communications is lost, the
 * operator control task will be stopped. Re-enabling the robot will restart the
 * task, not resume it from where it left off.
 */
void opcontrol() {
pros::Controller master(pros::E_CONTROLLER_MASTER);
pros::Motor chassis_left(1);
pros::Motor chassis_right(2, true);

pros::Motor intake_left(9);
pros::Motor intake_right(10, true);

pros::Motor intake_middle(11);

pros::Motor flywheel(13);

while (true) {
// setup tank control for the chassis
// get input from both joysticks
int stick_left_val = master.get_analog(ANALOG_LEFT_Y);
int stick_right_val = master.get_analog(ANALOG_RIGHT_Y);

// apply values to the respective chassis motors
chassis_left = stick_left_val;
chassis_right = stick_right_val;

// setup lift contols

// setup descore arm
// no motor?

// left triggers linked to back intakes
// SWITCH INTAKES TO TOGGLE
bool l1_pressed = master.get_digital(pros::E_CONTROLLER_DIGITAL_L1);
bool l2_pressed = master.get_digital(pros::E_CONTROLLER_DIGITAL_L2);
```

NAME: August Horton

CONTINUED ON PG # 22

```
if(l1_pressed) {
intake_left = 127;
intake_right = 127;
} else if(l2_pressed) {
intake_left = -127;
intake_right = -127;
} else {
intake_left = 0;
intake_right = 0;
}

// middle intake based on right triggers
bool r1_pressed = master.get_digital(pros::E_CONTROLLER_DIGITAL_R1);
bool r2_pressed = master.get_digital(pros::E_CONTROLLER_DIGITAL_R2);

if(r1_pressed) {
intake_middle = 127;
} else if(r2_pressed) {
intake_middle = -127;
} else {
intake_middle = 0;
}

// flywheel
bool a_pressed = master.get_digital(pros::E_CONTROLLER_DIGITAL_A);

if (a_pressed) {
flywheel = 127;
} else {
flywheel = 0;
}

// delay
pros::delay(20);
} // end while loop
}
```

DATE | PAGE TITLE: | PG #

NAME: | CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE | **PAGE TITLE:** | **PG #**

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE	PAGE TITLE:	PG #

NAME:	CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE | PAGE TITLE: | PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #

DATE

PAGE TITLE:

PG #

NAME:

CONTINUED ON PG #