Team Robust

86832B

2024 - 2025

International School of Beijing

Contents

1. Introduction **5**
- 1.1. Notebook Layout 5
- 1.2. Digital Notebook Rationale 5

2. The Team **6**
- 2.1. Definition of Roles 7

3. Innovation Award Submission Form **8**

4. Goals **9**
- 4.1. Robust's team goal 9
- 4.2. Individual goals 9
  - 4.2.1. Eddie 9
  - 4.2.2. Emily 9
  - 4.2.3. Sophie 9
  - 4.2.4. Patrick 10
  - 4.2.5. Jason 10
  - 4.2.6. Aiden 10

5. The Design Process **11**

6. High Stakes Game Analysis **13**
- 6.1. Game Summary 13
- 6.2. Design Challenges 13
- 6.3. Design Goal 13

7. Define and Inquire - Independent Inquiries **14**
- 7.1. **Independent Inquiry (Analyzing Precedents)** 14
  - 7.1.1. Design 1: Cage 14
  - 7.1.2. Design 2: Lady Brown 14
  - 7.1.3. Design 3: Elevating intake 15
  - 7.1.4. Design 4: Fish Mechanism 15
  - 7.1.5. Design 5: Four bar lift 16
  - 7.1.6. Solution Selected 16
- 7.2. **Independent Inquiry - Programming** 16
  - 7.2.1. Language: Blocks 16
  - 7.2.2. Python 17
  - 7.2.3. JavaScript 17
  - 7.2.4. Pros C++ 17

8. Development Logs **19**
- 8.1. Pre-Season (Summer Building) 19
- 8.2. Early Season (August — APAC Tournament) 20
  - 8.2.1. Drivetrain Build 20

9. Define and Inquire **22**
- 9.1. Developing the Success Criteria 22
  - 9.1.1. Point-Scoring Priorities 22
  - 9.1.2. Final Success Criteria 22
- 9.2. Design Solution Brainstorming 23

9.2.1. Concept 1 — Robotic Arm ... 23
9.2.2. Concept 2 — Forklift ... 23
9.2.3. Concept 3 — Traditional Intake and Belt ... 24
9.3. Decision Matrix ... 24
9.4. Best Solution and Plan ... 26

10. Develop and Plan, Create and Improve ... 27
10.1. Forkarm Build ... 27
10.2. Intake Design ... 28
10.3. Belt Design ... 29
10.4. Testing Phase ... 32

11. Reflect and Share - Pivot ... 34
11.1. Pivot to Concept 3 — Being Conventional ... 34

12. Develop and Plan, Create and Improve (Again) ... 35
12.1. New Design Ideas ... 35

13. Mid-Season Development Logs ... 36
13.1. Post-APAC — ACAMIS Tournament ... 36

14. Reflect and Share - Post-APAC Reflections ... 37
14.1. Individual Goal Reflection ... 38

15. Create and Improve - Rebuild ... 40
15.1. New Drivetrain ... 40
15.2. Belt Design v2 ... 42
15.3. Intake Design v2 ... 43
15.4. Clamp Design v2 ... 44
15.4.1. Mini Independent Inquiry (Jason) ... 44
15.5. Pneumatics ... 45

16. Reflect and Share - Post-ACAMIS Reflections ... 46
16.1. Individual Goal Reflection ... 47

17. Define and Inquire - Further Independent inquiries ... 49
17.1. ACAMIS North Robots ... 49
17.1.1. BIBA 20068A: ... 49
17.2. General Independent Inquiry ... 49

18. Develop and Plan - Post-ACAMIS ... 50
18.1. New Intake ... 50
18.2. New belt ... 51
18.3. New hooks for belt ... 51
18.4. Climb ... 52
18.5. New clamp ... 53
18.6. Lady Brown ... 53
18.7. Doinker ... 54

19. Create and Improve - Post-ACAMIS ... 55
19.1. Belt ... 55
19.2. Intake ... 55
19.3. Clamp ... 55

19.4. Lady Brown 56
19.4.1. Lady Brown Pneumatics Holder 57
19.5. Doinker 59
20. Develop and Plan, Create and Improve - One week sprint 60
20.1. The Issue 60
20.2. Rethinking Everything 60
20.3. Ideological Change 63
20.4. Building Logs 64
21. Reflect and Share- Post-TIS Reflection 66
21.1. Lady Brown 66
21.2. Belt 67
21.3. Doinker 67
21.4. Intake ramp 67
21.5. Gameplay and Strategy 68
21.6. Individual Goal Reflection 68
22. Software 71
22.1. Introduction 71
22.2. **Define and Inquire - Learning Journey** 71
22.3. **Develop and Plan, Create and Improve - Development Journey** 71
22.3.1. September 71
22.3.2. October 72
22.3.3. November 74
22.3.4. December 74
22.3.5. January 75
22.3.6. February 76
22.4. **Reflect and Share - Programmer Self-Reflection** 78
22.5. List of Current Features 78
22.6. List of Planned Structures 78
22.7. Folder Structure 79
22.8. Brain Ports Map 80
22.9. Operations Control 81
23. APPENDIX 1: Complete Program 83
24. APPENDIX 2: Bimonthly Notes and Goals 96
24.1. August 22 96
24.2. August 29 96
24.3. September 12 96
24.4. September 26 96
24.5. October 10 96
24.6. October 24 96
24.7. November 21 96
24.8. December 5 96
24.9. December 19 96
24.10. January 9 96
24.11. January 23 96

24.12. January 30 97
24.13. February 6 97
24.14. February 13 97
24.15. February 20 97
24.16. February 24 97
25. APPENDIX 3: Daily logs 98

1. Introduction

1.1. Notebook Layout

Team 86832B's Engineering Notebook will follow the structure of ISB's Design Process (refer to "The Design Process"). For team contribution and project management records, flip to the final pages for detailed logs. This includes hours of contribution, meeting notes, goals, accomplishments, and personal reflections. Key moments in which large decisions were made, or immense changes were implemented will still be in the main section of the Engineering Notebook.

Team 86832B's Engineering Notebook is designed for an easy read-through in chronological order, with less relevant documentation and evidence at the end for reference if needed. This is to ensure clarity, useability, and completeness for the reader. The design cycle has also been repeated multiple times throughout the season. Thus, different iterations of our robot are shown in this notebook through eras: Pre-Season, Early Season, and Mid-Season.

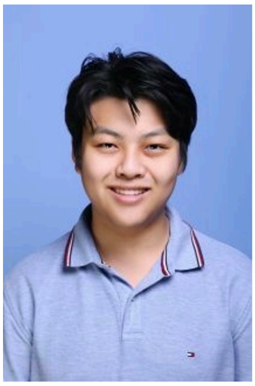
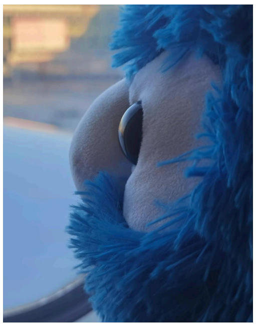
With that being said, enjoy!

— The Editing Team

1.2. Digital Notebook Rationale

For Robust, a digital notebook written via Typst was the ultimate choice. This is due to its automatic formatting and typesetting features not found on more traditional softwares like Microsoft Word or Google Documents. Robust initially used Microsoft Word, but pivoted to Typst mid-season as Microsoft Word could no longer support our growing notebook's needs. Some prevalent issues with Word included increased instances of lag, formatting inconsistencies, and a lack of control over stylistic features. Typst also allows for more control over the styling of the document and allows us to maintain a uniform and consistent style throughout the notebook. Compared to its competitor LaTeX, Typst is more modern, offers a simpler syntax and newer interface; compared to traditional handwriting, Typst allows for more liberty and ease in editing, allowing multiple team members to contribute at once, and also allows us to contribute from anywhere.

2. The Team

	Name: **Eddie Xu** Grade: 11 Years in Robotics: 2 Role: Team Captain, Lead Designer, Driver, Builder, E.N. writer Intro: GO LISTEN TO BORTKIEWICZ PIANO CONCERTO NO 1 I am singlehandedly responsible for the “Great Spacer Shortage” at the International School of Beijing’s Robotics department. Spacers for life.
	Name: **William Pan** Grade: 11 Years in Robotics: 2 Role: Lead Engineer, Builder
	Name: **Jason Yang** Grade: 11 Years in Robotics: New Role: Builder, Designer, E.N. writer Intro:
	Name: **Patrick** Grade: 10 Years in Robotics: 0.5 Role: Programmer, Builder, E.N. Writer, Unofficial Graphic Designer Intro: Jack of all trades, master of some

	Name: **Aiden Kim** Grade: 9 Years in Robotics: 1 (Vex IQ) Role: Builder, Designer
	Name: **Emily Zhou** Grade: 9 Years in Robotics: 1 (Vex IQ) Role: Builder, E.N. writer, Designer
	Name: **Sophie Wang** Grade: 9 Years in Robotics: 1 (Vex IQ) Role: E.N. Writer, Builder

2.1. Definition of Roles

Builder: Someone who physically attaches and assembles components onto the robot. They also test prototypes and diagnose issues with designs, relaying the feedback to designers.
Designer: Someone who plans the robot's design prior to assembly. They contribute to ideation, assist builders during the construction process, and may do 3D modeling/CAD work when planning.
E.N. Writer: Someone who contributes to the team's engineering notebook.
Programmer: Someone who writes code and autonomous routines for the team.
Driver: The person who operates the robot during matches and for the Skills challenge.
Lead Engineer: The person who facilitates the entire assembly process and leads the builders.
Lead Designer: The person who facilitates the entire design process and manages the Fusion repository. They factor all designs during the ideation phase and do the majority of the CAD work.
Team Captain: The person who leads the team, distributes tasks, and coordinates with teachers.

3. Innovation Award Submission Form

86832B

Brief description of the novel aspect of the team's design:

The novel aspect of this team's design is our approach to the engineering notebook and the entire design process. Specifically, our consistent reflective process throughout the season, hoping to learn from our mistakes and grow as individuals. We hope that the highlighted innovation sectors demonstrate how our thought process and approach to learning have had an effect our design and final robot.

We achieve this by reflecting heavily on what we have learned, what we have improved on, and how we have done in terms of our personal goals after every major tournament. Using this, we are able to grow exponentially as a team, and ultimately end up in a far better place than we could've ever expected. Thus, we believe that this reflection process itself is worthy of consideration for the Innovation Award.

Identify the page numbers and/or the section(s) where documentation of the development of this aspect can be found:

Section 4, Section 11, Section 14, Section 16, Section 21 and section 22.4.

Explain why your submission is unique from other approaches to the problem it solves or task it performs:

While many would approach V5 with the sole intention of performing well during competitions, we take a unique approach in which we emphasize more on the process and the journey together. We hope that with the constant reflective process and a growth mindset, we are able to both develop new skills and excel during tournaments. This innovative approach to learning highlights **VEX's mission**, allowing us to be the "learners of today" and the "problem solving leaders of tomorrow."

4. Goals

August 22, 2024

4.1. Robust's team goal

As a team, our focus this year is on learning and improving. The majority of the members are completely or relatively new to the V5 program, and we have decided to take this year as an opportunity to accumulate knowledge and experience together. We hope to achieve this by consistently reflecting on our achievements and mistakes — striving to improve our technical and collaboration skills.

4.2. Individual goals

4.2.1. Eddie

After competing in Over Under last year, I'd like to take some of the experience I had and expand upon it. Specifically, I want to be more involved in the design process, creating prototypes, modeling robot concepts, and making unique components that I can be proud of. In this aspect, my main goal is to familiarize myself with Fusion 360, the modeling software our school provides for us. The skills developed by doing so won't just be useful for VEX — it will be helpful for any engineering or design field, which is what I'm thinking of going into. On the side, I'd also like to pass any new knowledge I learn down to the newer members of our robotics team, so that I can foster and nurture new talent for the years to come.

This year is also my first year as the main driver for our team. I hope to take my experience with the many video games I play and the game strategy that I love and apply it to High Stakes. Last year, I was the backup driver for my team, but did not get many opportunities to hone my skills. So, I'm looking forward to both improve my technical skill with the controller, but also think more critically when on the field. I hope to achieve this by analyzing my good and bad plays after matches, and learning and improving upon them for a cohesive winning strategy.

4.2.2. Emily

As a freshman coming into robotics with prior Vex IQ experience, I hope to learn a lot this year. As the main builder and designer of my IQ team, what I learned from a season was to plan before you build. Although this might take a lot of time, it is crucial in order to not encounter as many problems in the building and assembly process. Although I was on the sideline for all the matches, what I've learned is to give time to your drivers to adapt to not only the robot but also the game environment. The major lessons that I learned were to design before you build, divide the work among the builders, and not finish the robot at the last minute, as that reduces the driver's time to practice. I hope to integrate it into the HS club, as it was something that my team 86832F learned very late into the season, and to learn a lot more about V5 throughout this season.

4.2.3. Sophie

I have 1 year of experience in VEX IQ, with different robot structures compared to VEX V5. However, there are some transferrable skills that both forms of robotics share. During VEX IQ, the assembling process is much easier, as it does not require any additional tools (e.g., screwdrivers, ranches). Therefore, we often did not plan in advance before building. An important lesson learned from last year's experiences was to plan out the robot as a whole before distributing each section to different team members. Issues arose during the assembly process of the different components (intake, drive train, body) as each did not consider the connection points to the other. Great changes had to be made before the tournament and minimal time was given to practice driving. As the driver of last year VEX IQ's robot, it is necessary to have adequate time for driving practice to reduce avoidable human errors during the match. The major lesson learned after last year was to schedule the building process logically and effectively to ensure progress and success in matches. As a beginner in VEX V5 robotics this year, I hope to learn more about robotics and gain experience through participation in matches.

4.2.4. Patrick

As a first year in both robotics and programming in C++, my main goal for this year is not to excel, but to improve. I do not want to spend this time creating the most astonishing, optimal code, but rather experiment, seeing what I can do with the API, learning what experienced programmers do, improve my C++ skills, etc. I also believe this is a great opportunity to test my programming skills in an applicable scenario, as I have honestly never previously put substantial effort into one single project, and I'm hoping this season may be my first, so ambitiously speaking, robotics is a combination of several milestones for me. Ultimately, I want this to be a learning experience for me, in preparation for what lies ahead, robotics or software in general.

4.2.5. Jason

My journey with Vex Robotics started last year when I was introduced to this club by friend and fellow club member Bowen. Although I wasn't an official member as the season was pretty much over by then, they allowed me to use the spare parts to tinker around which I had tremendous fun doing. This year I hope to actually contribute and build a robot in a game environment. Through this process I would also learn to more about the techniques and building process of a bot, building my own skills and experiences. I would also like to experience the competitiveness of the game, learning from other teams and exploring ideas I would have never thought of. As I learned last year while attempting to build a bot blind, there are a lot of moving parts that all work with each other and solutions to problems that I could not have figured out without the help from the other members. Thus, by doing Vex Robotics I dont just want to learn with experience but also through collaboration with other people.

4.2.6. Aiden

As a freshman with 1 year of experience in Vex IQ, my main goal was to learn as much information about V5 as possible. Some main techniques I wanted to learn this year were designing and building most efficiently. I tried to achieve this goal by observing other team members with more experience at the start of the year so I could learn about how to use the materials and design correctly. Since I wasn't showing much dedication and concentration last year at Vex IQ, this year, I tried to participate in Vex V5 whenever I had time and tried to work along with my team members either building or designing.

5. The Design Process

The International School of Beijing (ISB) has a design process in which 86832B has determined is suitable and applicable to the VEX Robotics Competition.

Define & Inquire

- Clearly and concisely describe the design problem and design commitment.
- Analyze a wide range of precedents to inspire and inform solutions to the design problem.
- Define specific and thorough success criteria to meet the user's needs.
- Describe all details about the end user relevant to the design problem.

Develop & Plan

- Effectively apply design thinking practices to generate a wide range of divergent ideas.
- Develop selected ideas into distinct design concepts that vary in style and type and clearly justify decisions.
- Develop a highly detailed plan for creating the preferred design.

Create & Improve

- Create a high-quality design that meets success criteria and responds to the stated problem.
- Apply extensive feedback to iterate and highly refine the design. Show evidence of this feedback and the team's response.
- Demonstrate strong growth in technical and design skills.

Reflect & Share

- Document the design process in detail, sharing the product and learning with others.
- In detail, use the success criteria to explain the strengths and weaknesses of the process and product.
- Reflect specifically and meaningfully on what further improvements could be implemented to develop the product.
- Reflection must be consistent throughout all three stages of the design process. Share, Critique, and Impact must also be present during the entire design process.

The ISB Design Process is intended to be employed in a perpetual loop, encouraging students to continuously ideate, create, test, and reflect, improving the product every time.

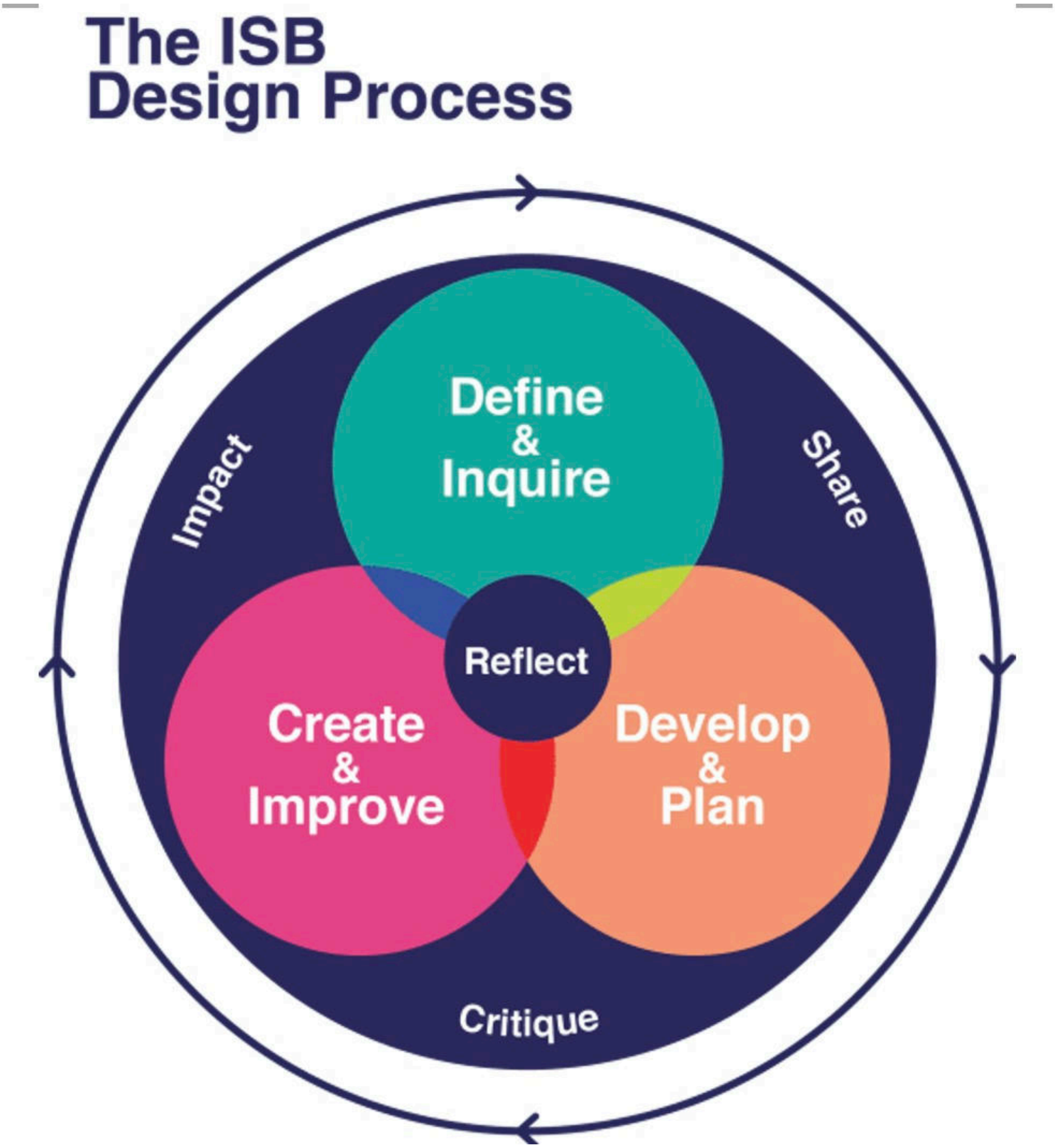


Figure 1: *The ISB Design Process*

6. High Stakes Game Analysis

6.1. Game Summary

High Stakes is a game played on an open field with plastic rings as the primary game scoring object, with a total of 48 rings consisting of 24 blue and 24 red. These rings can be scored on four types of stakes: mobile, alliance, wall, and high. Mobile goals are a point of attention as their points can effectively double or subtract from the final count if placed in either positive or negative corners. Top rings are also critical for success as they are valued as +3 points rather than +1 for regular rings. Rings may also be removed at any time from any stake. Points may also be earned during the autonomous period and during the endgame elevation period.

6.2. Design Challenges

Since the stakes are of varying heights, the first main design challenge is being able to score on all types of stakes. This means being able to score on mobile goals, alliance, and wall stakes. Being able to reach all vertical parts of the field will be key to success. An open field with no game-defining corners means that speed and control is critical. Having the ability to play strong defense, control a corner, and get around the field efficiently and fast pace is essential. Contact with other robots due to the open field is inevitable. Robots must be strong enough to withstand collisions and robust enough to endure the length of an entire match. A strong offensive game is also paramount to controlling the field's corners and scoring well.

6.3. Design Goal

To succeed in High Stakes, we hope to tackle what the game demands in an efficient, innovative, and creative manner. This means creating strong success criteria and keeping them in mind throughout the entire design, build, and competition process.

7. Define and Inquire - Independent Inquiries

7.1. Independent Inquiry (Analyzing Precedents)

7.1.1. Design 1: Cage

`[Sophie]`

Scoring: Wall stakes, alliance stakes, mobile stakes, level 1 elevation

Pros:

- Versatile scoring methods.
- Components: conveyor belt, extendable arm, "elevation", clamp
- High agility
- Able to cross middle

Cons:

- Difficult to "de-score" rings directly out as they require greater motor strength, which is currently unavailable in our school.
- Difficult to build and program because you must redirect it into the cage
- Requires custom parts
- Slower and less efficient than other methods
- Requires much more extensive planning because of the 6-bar lift
- Not robust and can be broken if other robots collide with the mechanism

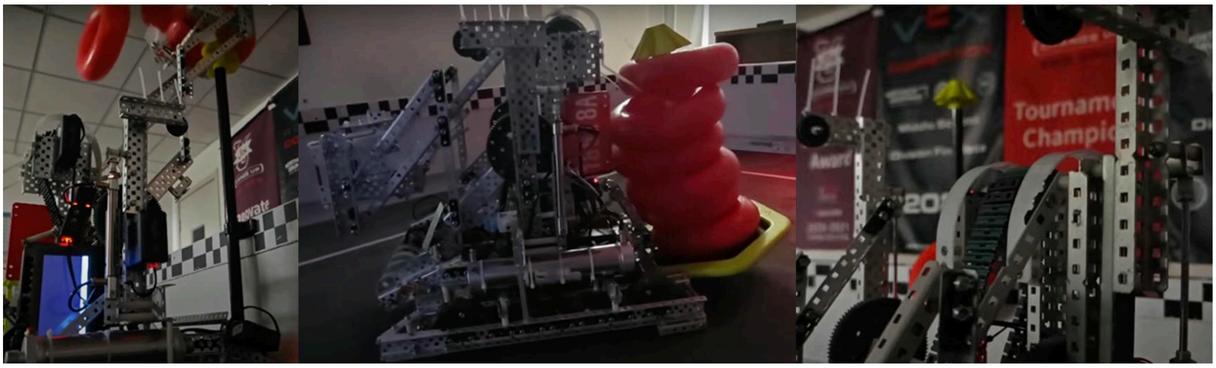


Figure 2: *18518A's Design (https://youtu.be/kPbByWzR5LU?si=Mjclji_PJWQnC5f2)*

7.1.2. Design 2: Lady Brown

`[Emily]`

Scoring: Wall stakes, mobile stakes, alliance stakes, level 1 elevation

Pros:

- High agility
- Efficiently scores wall stakes
- Strong hold onto mobile stakes when scoring (prevents opposing teams from taking)
- Small size
- Sturdy structure: Every component has protection around it, as it forms a rectangular prism-shaped structure.
- Able to climb level 1

Cons:

- As we were building, certain difficulties we encountered: Getting the conveyor belt to precisely intake the ring onto the mobile stake, which incorporates both the conveyor belt and clamp system for scoring. Making the lady brown mechanism to score wall stakes also required a lot of testing to aim precisely onto the stake and effective methods to attach the mechanism to the robot base. Another issue we encountered was holding the ring with enough friction and extending the arm long, so it reaches the height of the wall stake.

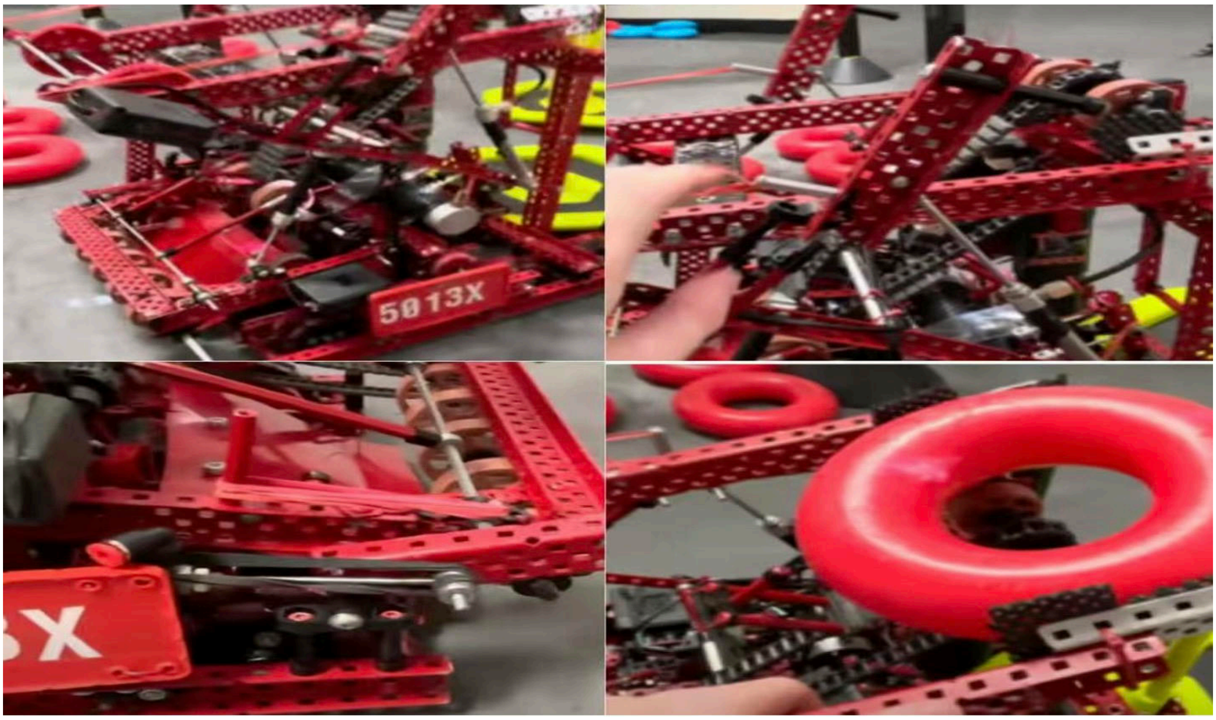


Figure 3: *5013X's Design (https://www.youtube.com/watch?v=aThosKiQQ0Y)*

7.1.3. Design 3: Elevating intake

`[Sophie]`

Pros:

- The conveyor belt could be elevated to score the ring directly onto alliance stakes.
- Have one less mechanism, as the conveyor belt is being elevated to score wall/alliance stakes.
- Various scoring methods: The robot could be used to score mobile, alliance, and wall stakes directly from the conveyor belt. There is potential to achieve level 1 elevation.

Cons:

- Low efficiency: Time is required to rotate the conveyor belt.
- High precision required: The ring would need to be directed precisely onto the stake top to score.
- Multiple canisters are required for clamp, doinker

7.1.4. Design 4: Fish Mechanism

`[Emily]`

Scoring: Mogo, Wall Stakes, alliance stake, level 1 elevation

Pros:

- Unique design
- Can score 2 rings at once

- Very fast and efficient on every scoring object
- 100% accuracy
- Able to do level 1 elevation
- Still able to cross under (very important to William)

Cons:

- High precision required: every mechanism runs smoothly with the next
- Time limitation
- Difficulty level is very high

7.1.5. Design 5: Four bar lift

`[Aiden]`

Scoring: mobile stake, wall stake, alliance stake

Pro's:

- Steady build with 4 bars
- High accuracy on scoring wall stakes
- Ability to minimize the structure(fold)
- Mid accuracy on scoring alliance stakes
- Can be easily made

Con's

- Heavy mechanism compared to others
- Unable to go down the rhombus
- Slow scoring system
- Lowers the robot's agility, which can lead to more malfunctions
- Lower efficiency compared with other mechanism
- Unable to do level 1 elevation

7.1.6. Solution Selected

[Sophie]

After considering multiple robot structures, our team believes that the lady brown mechanism can score both mobile and wall stakes most efficiently. The robot can score on the mobile and alliance stakes directly from the ramp and wall stakes by utilizing the additional lady brown mechanism. The structure of the robot would need to be redesigned to incorporate the various mechanisms. Therefore, we redesigned the robot starting from the base to effectively employ all space available.

7.2. Independent Inquiry - Programming

Evaluation of Different Programming Languages

7.2.1. Language: Blocks

This language is what I personally consider to be one of more rudimentary nature, snapping together blocks to form a pseudocode-esque program that performs actions.

Pros

- Simple to use
- Straightforward to read and interpret
- Plentiful documentation and active development community
- Personal prior experience
- IDE (Vexcode V5) Contains built-in assistive features

Cons

- Quite rudimentary, lacks more advanced functions
- Drag and drop – inefficient
- Inefficient in terms of code writing and execution
- IDE has known saving and formatting issues

Final Evaluation

I personally believe there is nothing wrong with this language, but for me, I known already know more complicated languages, along with the fact block based hinders efficiency majorly, I may use this for quick testing programs, but for the actual main program, I think not.

7.2.2. Python

This language is one that I am personally most familiar with, being the first real programming language I learned and having years of experience with it, I am most comfortable with this language.

Pros

- Familiar language
- Experienced in said language
- Plentiful documentation and guides
- Straightforward syntax

Cons

- Lacking feature-wise in some regards
- Slow compilation
- IDE (Vexcode V5) has known saving and formatting issues

Final Evaluation

This was a strong contender when deciding on which language to use, but in the end, I decided against it due to the features it lacked as well as my desire to learn and grown in the vex program, and choosing a familiar language would devoid me of that.

7.2.3. JavaScript

This is a language I only recently learned supported VEX V5 while perusing the VEX Forums, it appears relatively new and immature.

Final Evaluation

No Pros-Cons evaluation will be done for this language as I cannot find a logical reason to use this language, the VEX Library for JavaScript is quite new and thus has no community, it also lacks greatly in functionality support and proper documentation.

7.2.4. Pros C++

I was made aware of this C++ Library by the head programmer of the club – Leon. It is a library developed by the Purdue Sigbots VURC team, and it greatly adds functionality on top of the official VEX C++ Library.

Pros

- Most versatile in terms of functionality
- Detailed documentation
- Thriving development community
- Addon for Visual Studio Code, my IDE of choice
- Fast compilation and efficent execution

Cons

- Most complicated of all languages
- Language I'm unfamiliar with
- Most recent version still under development

Final Evaluation

This library is what I ultimately decided on using, partly due to personal decision and partly due to Leon's recommendation, he insists that it is the most complete and decent in terms of VEX development libraries. I also decided this language because I wanted to challenge myself and gain out of the robotics experience, and an unfamiliar library in an unfamiliar language felt like an apt challenge.

8. Development Logs

8.1. Pre-Season (Summer Building)

July 20-August 18 2024

`[William]`

I spent time working with George, Samuel, Ryan, and our graduated senior Jun during the summer break. The summer robot was powered by stacked, "motor block", which is using gears to connect 3 motors for a side of the drive train as the power source, we think it takes less space in the front.

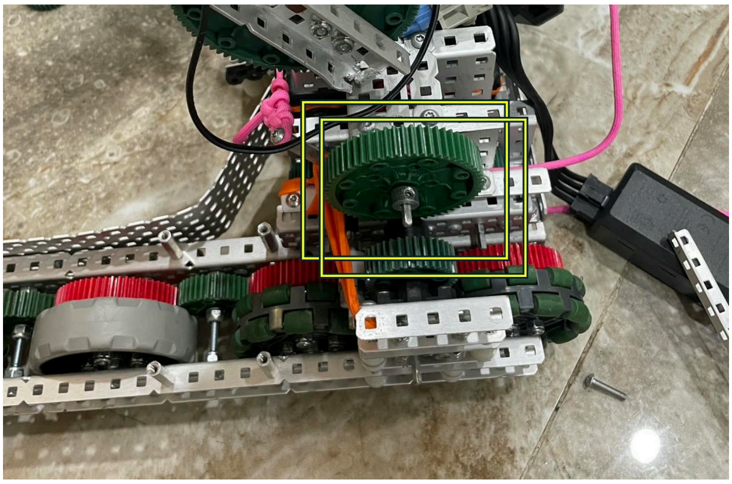


Figure 4: *"Stacked Motor Block"*

That summer drive train finally went to Team D's robot, But I also learned from it. Due to my old experiences' considerations, I decided to make the drive train slow but robust. So that makes the "boomer" robot, the motors are stacked at the top of the wheels and uses a gear to transform the power.

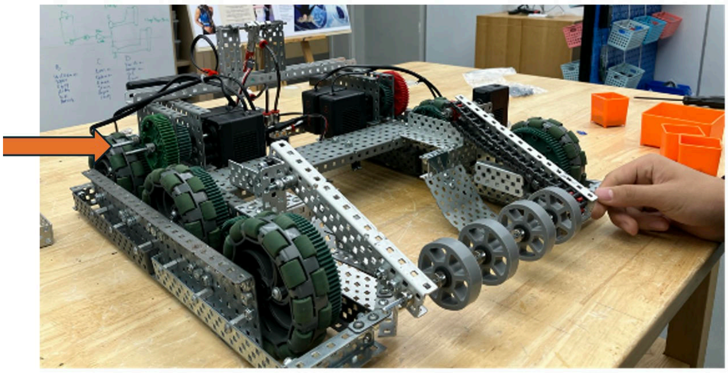


Figure 5: *Drivetrain*

At this moment, I must admit this design is a bit not good. There is quite a lot of friction made by the gears, so we spent extra time fixing them. Not to mention that later the season, the gears started to slip and malfunctions. I had to add a C-channel to stabilize the connection. Later in the new robot we abandoned this idea. So yes, it wasn't fully robust on the power system part.

The front Steel C-channel is made because I don't want the wheels get collide with other robots. So, it is there. The advantage for that wasn't avoid collision (because the wheels are good) but lower the center of mass. I was very aware of center of mass because… Last year I witnessed some robots got tipped over in a middle of match. So for our final work in this robot, you can observe that the upper parts are made of aluminum.

8.2. Early Season (August — APAC Tournament)

8.2.1. Drivetrain Build

September 26th, 2024

`[William]`

Orientation did take a lot of time, but it is necessary in the drive train

Figure 6: *The drive train “sides” (wheel-and-gear assembly)*

With the change of collision during the match, we have decided to add c channels to ensure the function of our drive train.

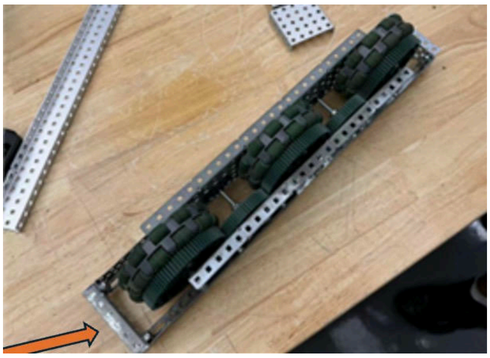


Figure 7: *One of the sides with its protection channels*

Since the L - channels were over the limit. There were 2 solutions that we (Emily and Sophie) came up with.

1. The first idea is to retract the L-channels, that is the simplest solution. But we were concerned this would block the wheels from operating normally.
2. The second way to fix the issue is to replace the L-channel with another shorter C-channel. C-channel’s front bar is more durable than the L-channels.

September 26th, 2024

`[Sophie]`

During this session, we added aluminum pieces to the bottom of the motors to ensure its stability. The weight of the motors was bending the extended part we made previously, as only two nails are keeping the extended piece onto the base. Adding an aluminum piece underneath would eliminate this problem and secure the motors from bending down. After testing if the solution worked on one side, we updated the changes to the other side and connected the two sides together.

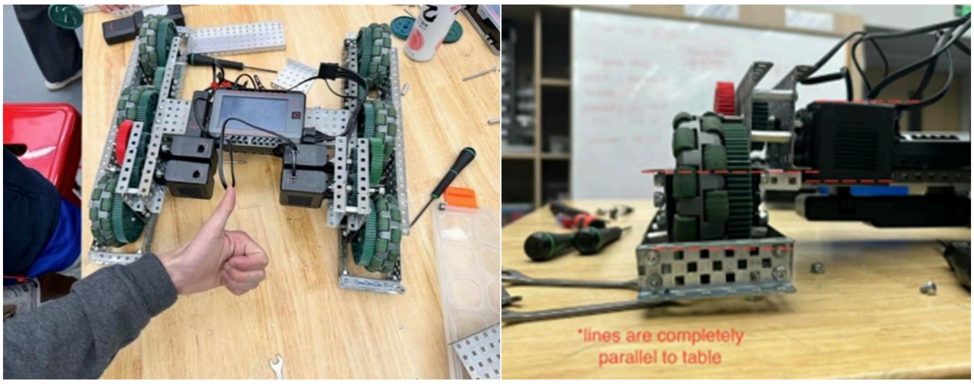


Figure 8: *Assembled drivetrain & slanted motors*

After connecting the drivetrain to the code, we observed an issue that the motors are shaking to an unusual extent when the drivetrain is turning directly on the spot.

9. Define and Inquire

9.1. Developing the Success Criteria

<u>August 22nd, 2024</u>

`[Eddie]`

By calculating which points are worth the most per ring, we were able to figure out a list of priorities for our design. In short, top ring is king.

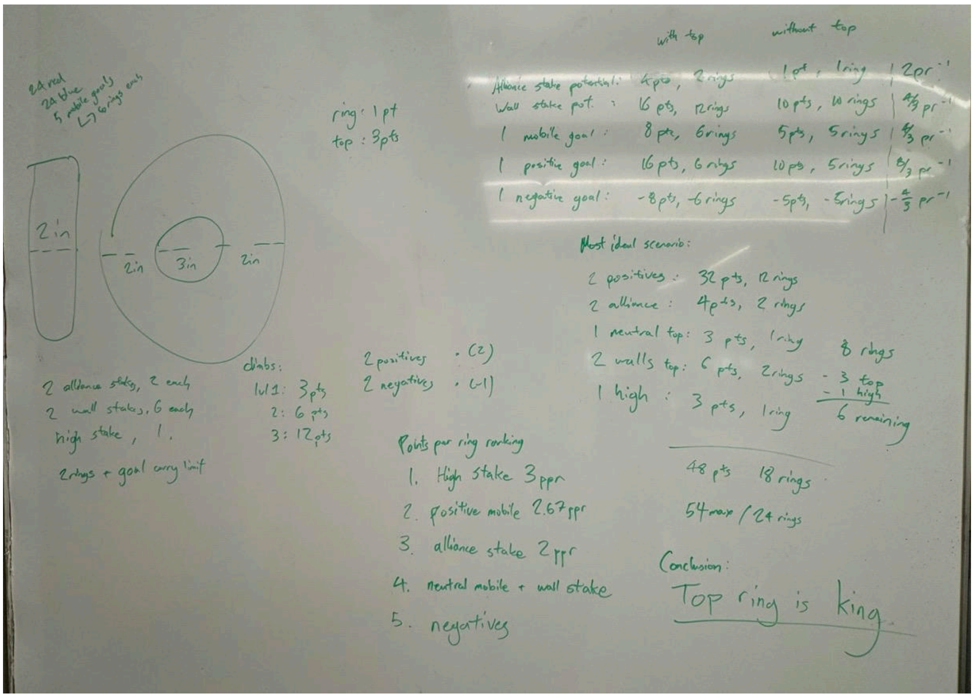


Figure 9: *Eddie's calculations of all scoring methods and their point per ring efficiency*

9.1.1. Point-Scoring Priorities

1. Top rings
2. Mobile goals and corners of the field
3. Alliance and wall stakes (to get top ring advantage)
4. Automation period
5. Endgame period

From interpreting the game and taking on our priorities, we have decided that our robot must:

- Be able to score on all types of stakes
- Be fast enough to get around the field quickly and take control of corners
- Be durable enough to withstand collisions and collide with others
- Be able to consistently score without mechanical errors
- Have a strong autonomous code that scores well

By considering these major factors, we have come up with the following success criteria to base our designs from, as well as to analyze which option suits our needs best.

9.1.2. Final Success Criteria

1. The robot must be able to score on all types of stakes.
2. The robot must be strong enough to withstand collisions and collide with others.
3. The robot must be consistent in its scoring mechanism.

4. The robot should have a strong autonomous code that scores well.
5. The robot should have a user-friendly interface for the driver.

9.2. Design Solution Brainstorming

9.2.1. Concept 1 — Robotic Arm

`[Sophie]`

Our team determined the goal and functions of our robot during the competition. We agreed as a team to build a robot that reduces the score of the opposing team. We will achieve this by taking the rings out of the scoring system other teams had already put in. In order to achieve this, our team planned on making a mechanic-hand-like structure to take the rings out at an angle. Xander has created a sketch of the specific structure:

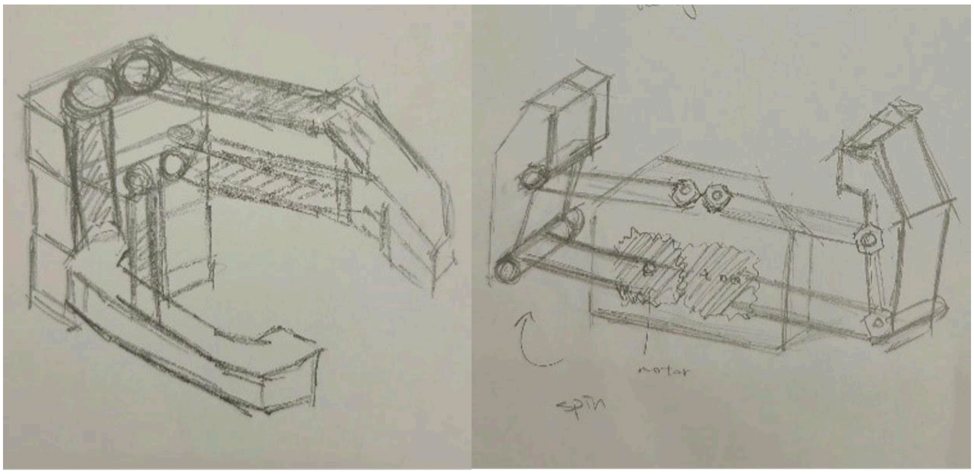


Figure 10: *Xander's sketches of the robotic arm*

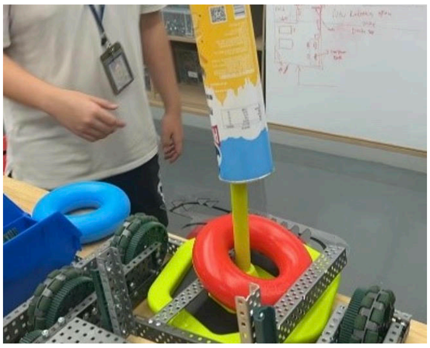


Figure 11: *Pringles can on mobile stake*

The robot arm eliminates the possibility of using Figure 5 as a design. This design does not take out the rings at a certain angle which increases the difficulty and strength required to execute the task. It is better off we could take the rings out at an angle, which reduces the friction between the ring and the spear (of the stake) and eventually improve efficiency.

9.2.2. Concept 2 — Forklift

<u>September 12th, 2024</u>

`[William]`

Eddie found a combination of robot arm and forklift. Indeed, the robot arm is not efficient because it only takes out one donut each time, making it waste time going back and forth. Xander then further developed the idea from a video1 founded by Eddie.

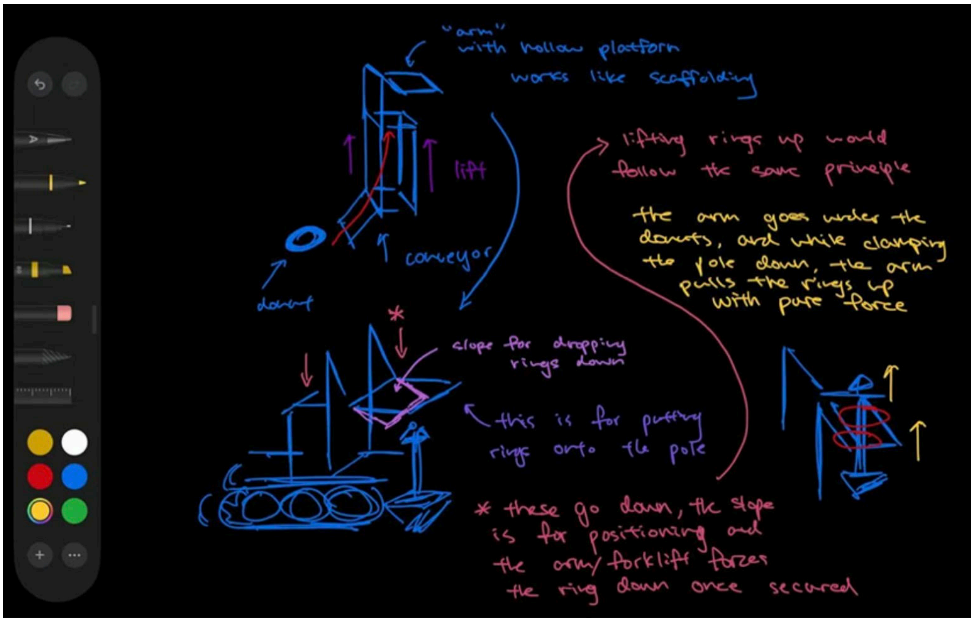


Figure 12: *Xander's brief idea for the Fork-arm*

9.2.3. Concept 3 — Traditional Intake and Belt

September 12th, 2024

`[Emily]`

I connected the forklift design to what my team did for Vex IQ during the last robotics season (Full Volume). A suggestion and idea for building an arm is something like what is inserted below. We can create an input system that collects the rings from the floor. It will go to a "container" where it can be extended out to put the ring on the mobile stake.

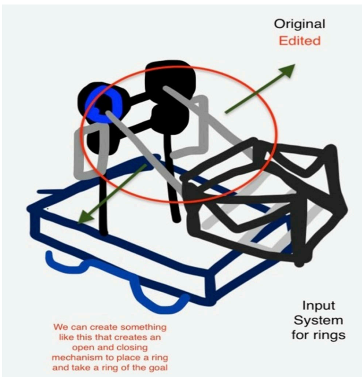


Figure 13: *Emily's sketch of the belt-intake design inspired by VEX-IQ*

`[Eddie]`

While this design, consisting of a traditional intake and a belt which throws rings on a clamp is effective and commonly seen, we believe that it is a bit "too conventional". It'd be more impressive to pull off something more creative, so we're leaning more towards the other ideas, like the forklift. However, this is definitely a solid option in case we need consistency and something safe to lean back on.

9.3. Decision Matrix

September 26th, 2024

	Concept 1 - Robotic Arm	**Concept 2a - Fork-arm**	**Concept 2b - Forklift**	**Concept 3 - Regular belt intake**
Motor Power usage	44W (4+0/3+2)	33W~44W	38.5W (2+3)	22W (2+0/1+2)
Complexity (1-10)	9	8	8	5
Reliability (1-10)	5	7	4	8
Controlling ease	Very Difficult	Moderate	Difficult	Easy
Accuracy aid (1-10)	2 ~ 5, depends on program	6	4	9
Program difficulty (1-10)	10	8	7	5
Cool factor (1-10)	10	8	6	2
Able to insert donuts?	Yes	Yes	No	Yes
De-score donuts?	Yes	Yes	Yes	No
Unique?	Yes	Yes	Yes	No

Seeing how the fork-arm/forklift offers the best compromise if we want to both be unique and de-score donuts, we decided to run with that. Our vision is to be the greatest ally for any alliance and restrict opposing alliances' progress by removing their rings.

9.4. Best Solution and Plan

`[Eddie]`

To compensate for the heavy wattage usage in the arms/lift, we have decided that we will run a 4-motor drive to save as much wattage for Concept 2 as possible.

`[Sophie]`

The structure of the forkarm allows us to extend the arm higher to take off the rings scored by the other team. The objective of this forkarm structure is to ascend/descend the ring without it rotating along with the movement of the arm/motor. As shown in the picture below, the piece requires one motor attached to two gears (on each side of the arm pieces) for the arms to extend. For the side labeled “1”, there are two pieces (one takes up 1/3 of the and another takes up 2/3 of the total length). The side labeled “2” is connected directly to the motor, which transfers the momentum for the arm to extend. There are three pieces that make up this side, each taking 1/3 of the total length. Considering the function of the structure, the total length of the pieces added up in one side should be equal to the other side. One of the concerns that should be brought into consideration is the weight of the pieces that assemble the forkarm, as there are only two motors to extend the forkarm. Along with the piece to which the motor is connected, it might be possible to attach a design that could help the robot climb to the first level of the ladder to score extra points.

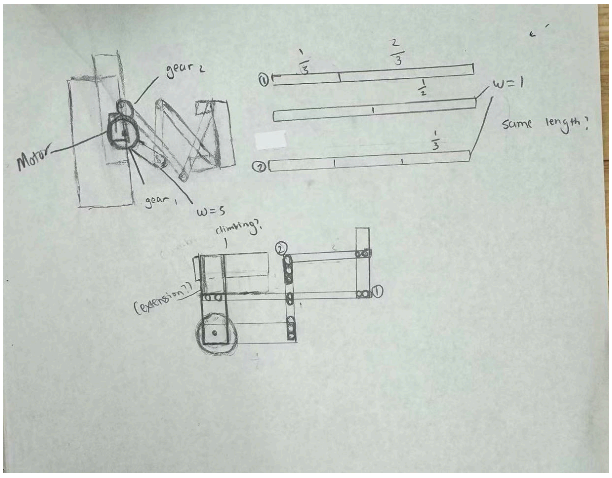


Figure 14: *Sketch of Concept 2 — the “Forkarm”*

10. Develop and Plan, Create and Improve

10.1. Forkarm Build

October 14th, 2024

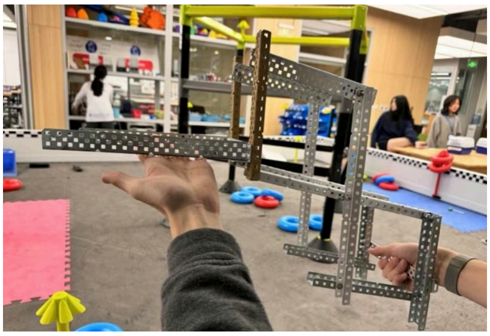


Figure 15: *Basic prototype of the forkarm by Eddie*

We haven't completed the main structure and there is still a lot to change. Other than that, I am aware that if all of us are focusing on this, we won't have enough time to make the adapter for it to attach on the drive train.

We also found the arm probably won't be able to reach the bottom donut if it lies on ground. This is not a big deal as our main goal is still to de-score. Other than that, Emily and Sophie also suggested a long time ago that we also need to be capable of scoring. Currently, the donuts tend to get pushed but not enter the arm, and if we want to get them, we need to push it to a corner. Although it doesn't affect the performance when interacting with the stakes, this issue is still a hot potato. Adding a may help with this, if it can start to spin once we contact the donut.

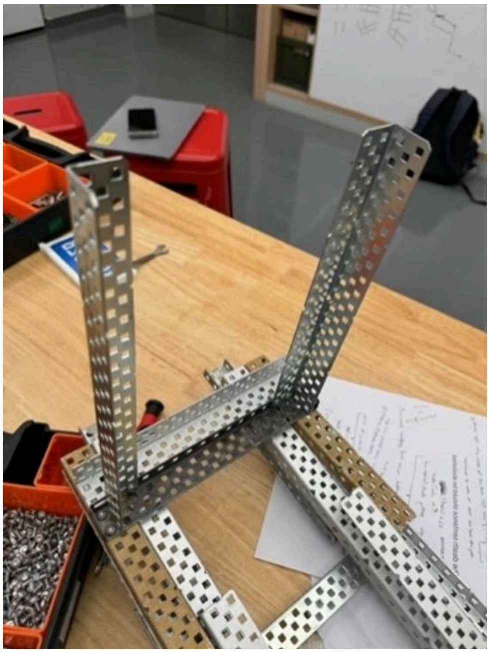


Figure 16: *Recent forkarm developments*

We realized that the arm wasn't stable enough. Eddie resolved this problem by changing c-channels to a l-channels. This significantly helped the problem of instability.

The shafts do not provide enough power to rotate the arm upwards, and it makes the clicking sound, meaning there is too much force. So, I (William) then think that we should have a longer shaft join in and provides power against. We now need to face these issues:

1. The adapter to the robot is wobbly.
2. The conveyor belt to intake the donuts.
3. The movement of the arm is not only vertical. When it lifts, it also contains horizontal movements, making our use of the mechanism harder. I told Patrick to make an autopilot when it is lifting, so the horizontal position stayed the same.

Our arm was high enough to score / de-score the wall stake. I know this is a very unique point of our robot. This is because in order to score on the wall stake, the robot must have an extension since the wall stake is higher than the build limit.

Figure 17: *Testing phase of new arm*

The problem now is that it is too slippery, not enough friction is present to the donut in relative to the donut holder. When we tried to de-score, it slipped, and I wasn't able to handle the horizontal movement caused by our design's inherent flaw.

10.2. Intake Design

For our intake system, we opted for a 15-unit aluminum C-channel. Although this exceeds the robot's size limit, we can flip the intake up onto the ramp before matches to stay compliant. A longer intake allows for better positioning in front of the robot, making it easier to roll in rings as we move forward. This design reduces the difficulty of guiding rings into the system since the rollers are located ahead of the drive train.

Using aluminum keeps the intake lightweight and simplifies the flipping mechanism. The intake needs to adjust its height slightly when rolling in rings, so we installed the motor on the outer side to avoid blocking its full range of motion. A small motor suffices for this task, as it doesn't demand much energy.

The roller configuration features outer rollers matching the ring's diameter, while the inner rollers fit the diameter of the ring's hole. This ensures efficient intake and smooth operation.

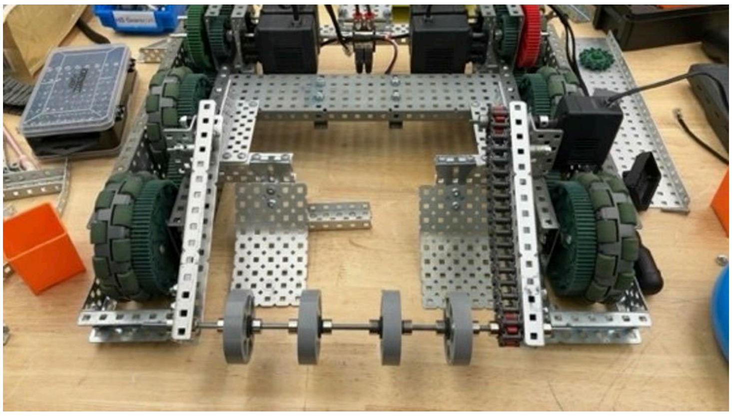


Figure 18: *Intake Roller*

<u>Nov 7, 2024: Complete robot intake</u>
`[Sophie]`
As discussed in our last session, we installed a regular green motor onto the conveyor belt. This choice enhances the rotating speed and efficiency of bringing the rings up the ramp. We opted against using smaller motors because we still had four motors available, having only utilized four for the drive train.

<u>Nov 12, 2024: Change in intake design</u>
`[Sophie]`
We switched to a smaller conveyor belt because it was challenging to attach the ramp with a frame to the supporting aluminum C-channel. The smaller belt helps save space, which had been a significant concern. This simplified design also reduces the overall weight and complexity of the robot, which is crucial given our limited build time before the competition.

10.3. Belt Design

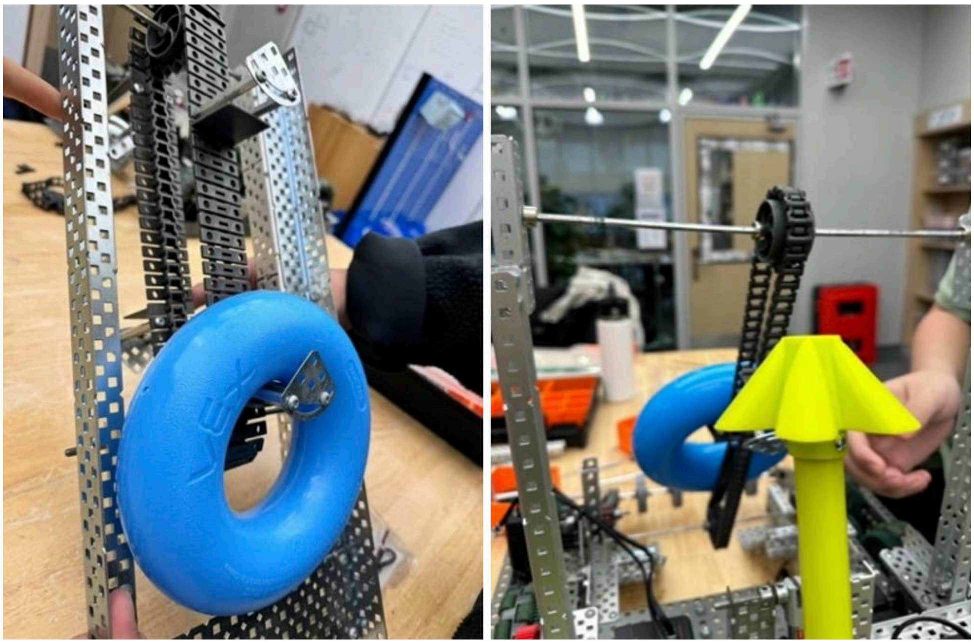


Figure 19: *Belt*

We also installed the conveyor belt (partly) onto the ramp. We encountered the issue of the protrusions conflicting with the bottom aluminum piece that held the two steel pieces frame together. Therefore, we had to reposition the shafts and the steel L-channels. After the tournament, we may revert to using the ramp, as it would enable us to attach an arm for scoring wall stakes and removing rings scored by the opposing team.

<u>Nov 14, 2024: Fix conveyor belt issues</u>

`[Sophie]`

One of the major issues from the previous session was that the protrusions on the conveyor belt couldn't "swing" the ring onto the mobile stake. For the first hour of the session, out team attached two aluminum flat pieces to the supporting pieces. As the ring would end up swinging straight out or going down onto the robot because of the lack of momentum, the two aluminum pieces would help assist the ring to be swung onto the stake.

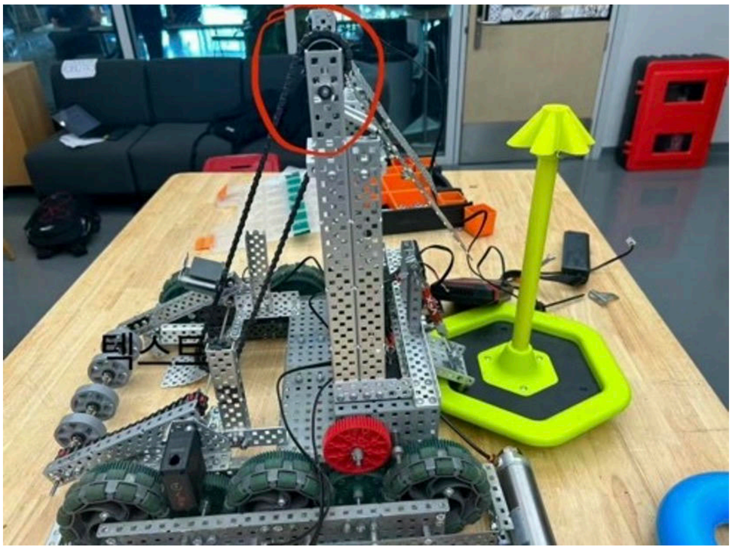


Figure 20: *Belt Issue Diagram*

However, this design was quickly proven wrong because the protrusions are bringing the ring down while the two pieces are resisting it from doing it, which causes a conflict between the two objects. It also did not help resolve the issue since there is no downward force in order for the ring to go into the stake. The next design our team thought of was to add another gear below the existing one. It is thought that this could create a downward force for the ring to go into the stake. Since the rings would simply swing out of the system because of the high momentum or drop onto the robot because of the low momentum, this design could potentially fix this issue.

The next design our team thought of was to add another gear below the existing one. It is thought that this could create a downward force for the ring to go into the stake. Since the rings would simply swing out of the system because of the high momentum or drop onto the robot because of the low momentum, this design could potentially fix this issue.

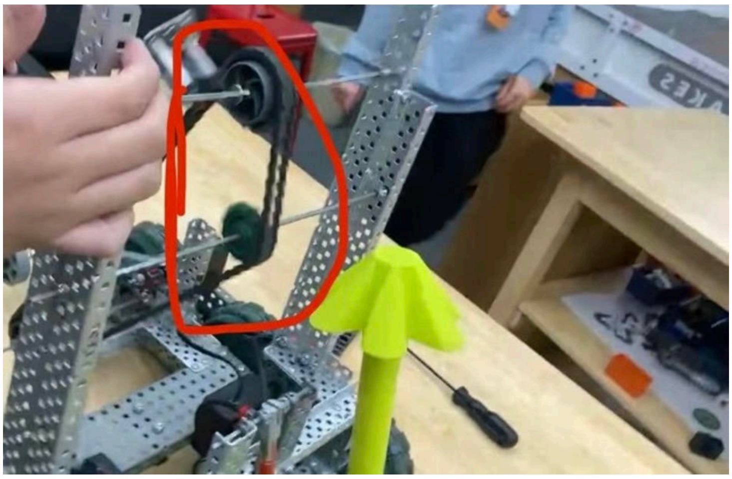


Figure 21: *N/A*

Nov 18, 2024: Attach Brain

`[Sophie]`

We attached the brain to the sides of the supporting aluminum c-channels. Since there was no other place for attachment on the drive train, the sides were the only place available.

Our team also switched the two small gears into one bigger gear. This could reduce the extra energy wasted by using several gears for one functionality.

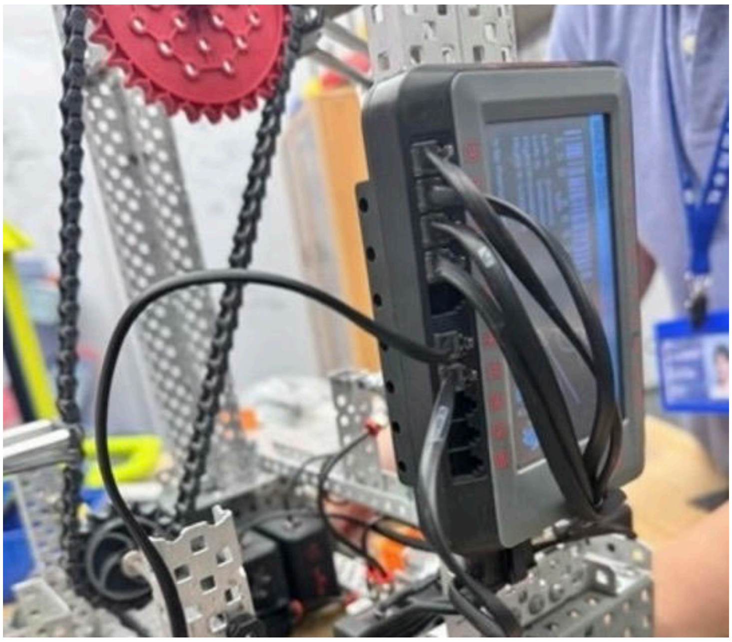


Figure 22: *Attached Vex Brain*

Nov 19, 2024: Finish Conveyor Belt

`[Patrick]`

We uncovered the issue with the pneumatics module today, the ADI pin plugged into the B TriPort had functional voltage and ground, but the signal pin was bent and not plugged into the brain. The fix was around 5 minutes. We still need to resolve with the extension cable continuously detaching during testing, perhaps tape will suffice. We spent the remaining session time tuning and adjusting the belt, eventually ending up with a semi-consistent belt-intake combo.

`[Sophie]`

We tested the different speeds for the conveyor belt so that it could swing the ring onto the stake. The whole intake system can now successfully swing the ring onto the mobile stake without additional human help. The robot is now functional as several rings can be scored to the mobile stake when rings are taken into the system. However, once one of the rings did not fully get scored into the stake (hanging onto the top of the stake), the whole intake system would get stuck after the next ring was swung onto the stake but missed because the previous ring was still hanging. The ring that was missed would drop back onto the robot because of the continuous rotation of the conveyor belt. This would then blockade the movement of the conveyor belt as the other components on the robot would block the ring from following the rotation of the conveyor belt, which would then stop the motion of the conveyor belt because of the protrusions that hang onto the ring. This issue could become fatal in a competition as the whole intake system would get stuck once the situation occurred and couldn't be resolved without human assistance. One practical solution is to make the protrusions bidirectional so that the conveyor belt can reverse its rotation direction once a ring is stuck in the system.

It should also be taken into account that the robot's base is very close the ground. The steel bars of the robot is currently scrapping the floor in the field (which is soft). The robot's weight could be causing this issue since the floor surface is soft, but it could be a potential problem when running longer distances.

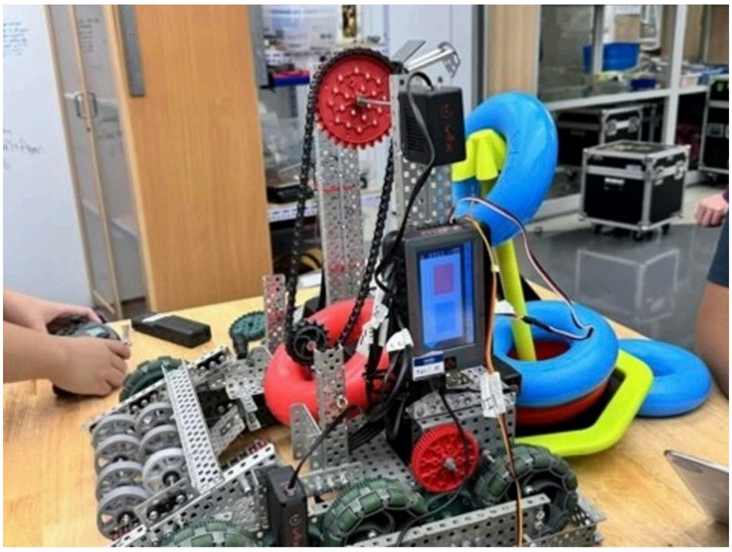


Figure 23: *Robot Function Testing*

10.4. Testing Phase

`[Sophie]`

After testing with the rings, we recognized that our system can effectively remove rings from the scoring area, but it requires additional support to prevent the rings from being pushed forward when transferring them to the pallet. We considered a few solutions: one option is to push the rings against a wall to use that force to guide them toward the robot, or we could add a conveyor belt underneath the pallet for easier intake. We also identified some concerns during testing. The robot's arm needs space to fully extend, meaning it must move forward while extending to remove rings from the pole and then back up when retracting.

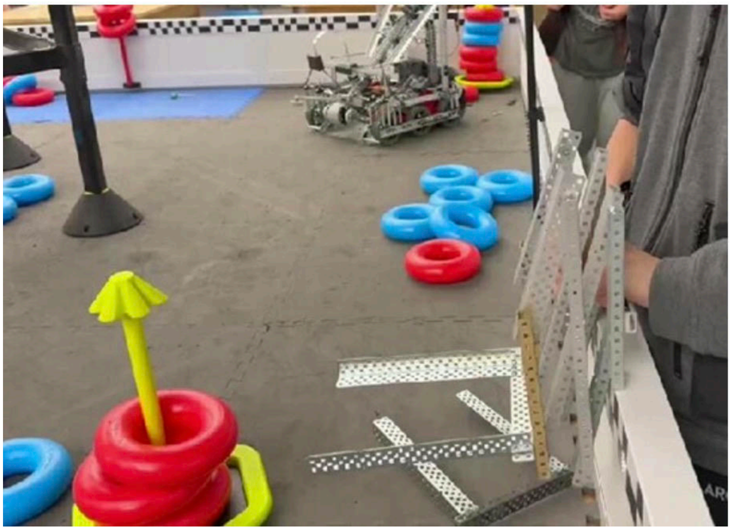


Figure 24: *Image of completed fork arm on field*

`[Patrick]`

Since there was not much for me to program, I decided to continue experimentation on the pitchfork gears. By the end of the "session", I had slightly modified the fork in a state where it could loosely extend.

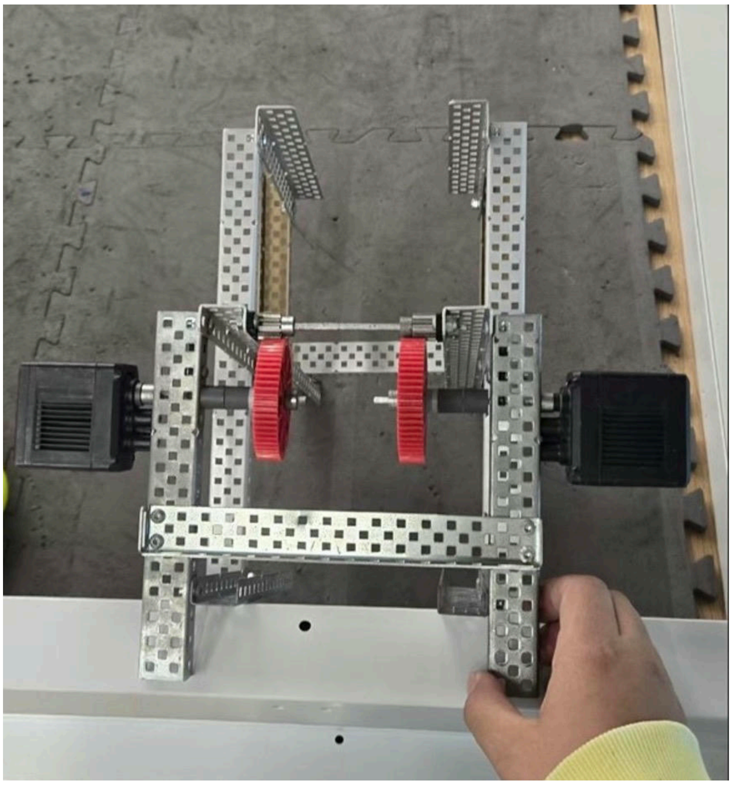


Figure 25: *Modifications made to the arm*

11. Reflect and Share - Pivot

11.1. Pivot to Concept 3 — Being Conventional

November 2nd, 2024

`[Eddie]`

For the last two months, we've made steady progress for our robot. This in large part can be attributed to our amazing team. I'd argue that we have the most dedicated group of individuals working tirelessly on our robot. The numbers also back us up: out of all the teams at ISB, we rank the highest in terms of average number of hours in contribution — 32.29, to be exact (as of the time of writing).

But, despite the large amount of contribution, we still faced many struggles, just like every VEX team devising a concept for their robot. First off, we spent a good majority of October building our "Forkarm".

But, after conducting many rigorous tests and thinking carefully about where we want to direct our project, we realized that our design was inherently flawed and that we needed a new solution. Some limitations of the forkarm include not being able to pick up rings from the floor, not being able to remove rings efficiently, not being able to score on mobile goals, etc. So, with the looming beast that is APAC just around the corner, we decided to scrap the entire month's work for a new, radical idea: to be just like everyone else.

Our new design will consist of an intake, a belt-conveyor system, a clamp, and a beautiful, majestic arm that would slam-dunk any bagel onto a wall stake with ease. The problem? We only have two weeks to complete this. Some rapid prototyping and a large time commitment must be made if we want to pull this off by the time APAC rolls around.

12. Develop and Plan, Create and Improve (Again)

Nov 2, 2024: Change in Robot intake/arm design

[Sophie]

After moving the black spacer to the outside of the steel piece, the height was still lower than the requirement. We couldn't add anything else to the arm anymore because it would create excessive stress on the motors for the extension of the arms. One accomplishment is that we can successfully pick up the two rings on top of one from the ground and put them into the mobile stake. Since the height requirement wasn't met and the design of the arm was inefficient relative to the teams we would compete with, our team decided to change the intake design.

12.1. New Design Ideas

Three New Design Ideas:

1. I (Jason) proposed a forklift design, which lifts rings vertically up/down to intake and puts them into the stakes. To take out rings, we put a "Pringles chip can" shaped object over the stake so that we do not need extra force or take it out at an angle. As discussed, this design would be inefficient but would be practical to make within the time before APAC robotics.
2. The design from: https://www.youtube.com/watch?v=kPbByWzR5LU - 18518A | Vex High Stake reveal. It is similar to our original arm design. The design from the video separates the intake and the arm. They intake the rings on the ground onto a ramp that could either transfer rings to the back, where they clipped the mobile goal onto their robot or let it drop into the arm to score the neutral/alliance wall stakes. This design would be the most efficient both in scoring and taking the rings of the opposing team. However, one constraint is there is a high possibility that we will not finish the robot before APAC. As the inputs and arms are separated, it is not possible to have a minimal viable product for the game if we do not finish any of the components.
3. I (Sophie) proposed to make the arm intake rotational so that when the arm takes the two rings from the ground, it is parallel to the ground. When the arm is extended to score the neutral wall stake, we rotate it approximately 30° so that it is tilted backward, which prevents rings from falling out and is able to put the rings into the stake at an angle. Putting the rings at an angle would decrease the downward force required to push the rings into the stake. Also, being one of the issues we recognized, the rings are extremely easy to fall out of the intake. The space is exactly the size of the ring. Even when we were driving the robot without any unexpected forces, the rings would occasionally fall out of our intake. If we were on the game court, contacts from other robots would make it easier for the rings to fall out. However, as mentioned earlier, we could not add any additional force to our robot arm as making the intake rotational would require us to add 2 more motors.

Nov 4, 2024: Conveyor belt protrusions

[Sophie]

We changed the design of the conveyor belt protrusions. As our original design of the protrusions is not strong enough to bring the rings up, we borrowed the design from the video (https://www.youtube.com/watch?v=kPbByWzR5LU). We screwed spacers onto the conveyor belt with the same width of the ring. As shown in the picture below, this design would withstand the weight of the ring better. We used a total of 5 protrusions on a conveyor belt with 71 units, with intervals of 8.

13. Mid-Season Development Logs

13.1. Post-APAC — ACAMIS Tournament

Nov 22-23, 2024 (APAC)

`[Sophie]`

Changes made to the robot during the matches:

Add: Bent aluminum piece in the back. This addition would prevent rings from falling into the robot body once it has not been successfully scored into the mobile stake. During the matches, it was highly likely for the rings to miss the mobile stake and often times it gets stuck into the robot body. This would prevent the conveyor belt from rotating and block the intake from operating. It would then take more time to release the stake and reverse the rotation direction of the conveyor belt. The final solution to this issue was to add the aluminum sheets. This addition turned out to be very effective as there was no chance for the rings to enter the inside of the robot (except by shoving it in from directly above).

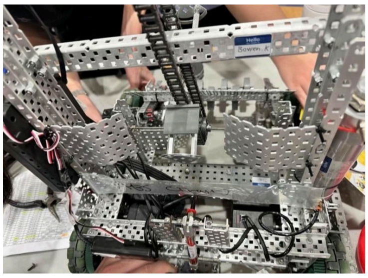


Figure 26: *Added Aluminum Plates*

Add: Steel 5*25 sheet. This sheet helps ensure that the rings are accurately picked up onto the protrusions on the conveyor belt. It is currently a temporary design as it is fixed onto the intake by zip ties.

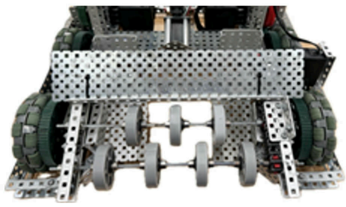


Figure 27: *Added Roller Intake Cover*

Change: Position of conveyor belt. There was a major issue that the rings could not be consistently scored onto the mobile goal. Our team attempted to solve the issue by changing the positions of the conveyor belt and the clamp. Since the rings went further than the "spear" target, moving it one unit forward successfully fixed the issue

Figure 28: *Close-up of belt*

14. Reflect and Share - Post-APAC Reflections

`[Sophie]`

After the APAC learning experience, we decided to make several changes to the robot.

1. **Agility of robot**
   a) Base: The current robot base was constructed based on last year's robot base (Over-under), which required stability and anticollision features. However, this does not adapt well to this year's game, as relatively fewer collisions would occur among robots and the robots would need to cover the field at a faster pace.
   b) Unnecessary steel bars: Since fewer collisions would occur, we should minimize the use of steel bars, to reduce the weight of the robot. Most robots that were considered successful had high agility, and they used mostly aluminum bars.
      i. Steel bars used on the base could be changed to aluminum.
      ii. The main support bars could be replaced by aluminum with standoffs (to prevent interior deformation).
   c) Size: Most robots observed are much smaller than the size limit, which improves agility.
2. **Structure of robot**
   a) Lady Brown: Most robots that score both the wall stakes and the mobile stakes intake the ring when it is delivered through the conveyor belt. The robot requires less precision when scoring wall stakes and can run at a faster speed.

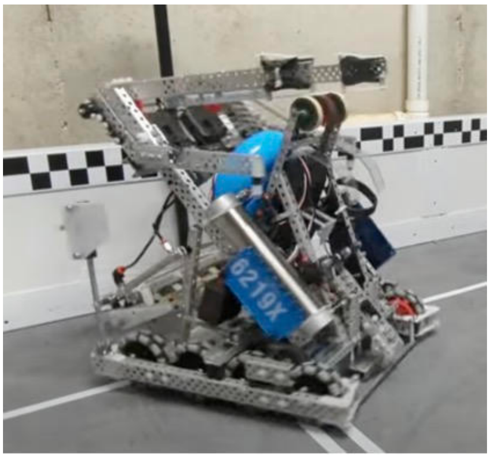


Figure 29: *(Credits: https://www.youtube.com/watch?v=aqhSpaBPNkw)*

   b) Cage: (https://www.youtube.com/watch?v=kPbByWzR5LU)
      i. Can score alliance stakes just directly from the conveyor belt.
      ii. The ring is transferred to the mounting system by the conveyor.
      iii. Has the ability to take rings out from wall stakes.

14.1. Individual Goal Reflection

`[Emily]`

I have learned a lot since the beginning of this season, and going to a competition has allowed me to understand the game better. From our reflections, you can see that we still have a lot to improve on. One thing that I would like to emphasize continuing into the season is designing before we build as not planning ahead of time has led to a lot of wasted time and trials. Although there might seem like a lot of time left in the season, I think that I have to reflect back and realize that creating a good robot takes a lot of time. Since we are a less experienced team I hope to see continuous growth throughout the season.

`[Aiden]`

Throughout this season I have learned a lot about the designing and building of V5. The materials and overall game are definitely harder than IQ, but I have definitely learned a lot.

`[Eddie]`

Most of the time I've been spending lately has been focused around building the physical bot. So, a lot of effort was spent on the fork-arm which was scratched, but also on the intake. I learned new techniques and theory such as the importance of proper tension in a chain, and ramp angles for good intakes on robots. However, this is not fully related to my goal because we haven't been to a competition, so I haven't been able to drive in a competitive environment.

But, I did begin looking into how to CAD VEX robots in Fusion. I began the installation of a VEX repository with all the parts from doing research online, and set up a Fusion team that all my teammates can access. Currently, I'm focusing on learning the basics — creating sketches, extruding objects, and making practical designs. I believe that learning these core skills will help me once I move to modeling our competition robot in the future.

`[Patrick]`

A hectic few weeks, but I have truly learned a lot, C++ was surprisingly not as grueling as I expected, and Leon was of great help when needed. Since the team's been busy slaving away at the robot, I've taken this opportunity to program some novel features that I just thought were cool. The GUI on the screen is my personal favorite, though it did take an ungodly amount of time, it also admittedly should not have been my first object, so I think time management is also something I have to work on. Another feature that is actually semi-useful this time is a drive mode switcher that allows Eddie to switch between different driving configurations. Overall, I think my time up until now has been decently utilized, I've created a functioning program, as well as used the excess time to explore functions that pique my personal interest. One thing I believe I need to improve on is sorting out priorities, though these aforementioned features are cool to have, in hindsight I should have looked more into (for lack of a better word) actually useful functions. That being said, I think I want to set myself a few mini-goals for ACAMIS.

- start on the autonomous code (APAC did not have auton)
- Learn more about functional aspects of robot programming

`[Sophie]`

In retrospect, I have gained a much more thorough understanding of VEX V5 robotics since the beginning of the season. I became more familiar with the methods of robot compositions during weekly build sessions. As there were many more aspects to consider in the assembly process of V5 robots, I understood the importance of planning before building. At the start of the season, I was inefficient when building and did not consider many physical factors (e.g., friction, gear combinations) as they were not as significant in VEX IQ. During the APAC and TIS tournaments, I had the opportunity to observe different designs of robots and learn from the other teams. Specifically, I realized that the formatting and way or recording the team's engineering process was done incorrectly. After talking with judges, they offered us advice for improvements and also looked at the notebooks of other teams who received higher grades. Overall, I was able to learn a lot from the tournaments and practical operations with the robot throughout the season.

15. Create and Improve - Rebuild

15.1. New Drivetrain

`[Eddie]`

When planning our new drivetrain, we had to consider a few things with our new design philosophy in mind:

- The speed of the drivetrain
- The width of the drivetrain
- The length of the drivetrain
- The weight of the drivetrain
- The number of motors used

We settled with the following requirements for the new drivetrain after considering our success criteria and the materials we had on hand:

- Speed: fast
- Width: medium length (30 wide and enough to just fit one ring)
- Length: short (27-long C channels as this was the longest, we had)
- Weight: light to reduce strain on motor and increase speed
- 6 motor drive to generate enough power

We also did not want to stray too far from our original design to increase redundancies in design and to minimize the number of new concepts we need. This meant sticking with omni wheels and a tank-drive. But we decided to utilize friction wheels in our drivetrain to prevent the possibility of us being pushed on the side by other bots (one of the biggest weaknesses of omni wheel tank drives).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Drive Gear		Driven Gear	100 RPM Cart.	2.75" Wheel	3.25" Wheel	4" Wheel	200 RPM Cart.	2.75" Wheel	3.25" Wheel	4" Wheel	600 RPM Cart.	2.75" Wheel	3.25" Wheel	4" Wheel
2	1		1	100	1.2 Ft/s	1.42 Ft/s	1.75 Ft/s	200	2.4 Ft/s	2.84 Ft/s	3.49 Ft/s	600	7.2 Ft/s	8.51 Ft/s	10.47 Ft/s
3	1	:	2	50	0.6 Ft/s	0.71 Ft/s	0.87 Ft/s	100	1.2 Ft/s	1.42 Ft/s	1.75 Ft/s	300	3.6 Ft/s	4.25 Ft/s	5.24 Ft/s
4	1	:	3	33.333	0.4 Ft/s	0.47 Ft/s	0.58 Ft/s	66.667	0.8 Ft/s	0.95 Ft/s	1.16 Ft/s	200	2.4 Ft/s	2.84 Ft/s	3.49 Ft/s
5	1	:	4	25	0.3 Ft/s	0.35 Ft/s	0.44 Ft/s	50	0.6 Ft/s	0.71 Ft/s	0.87 Ft/s	150	1.8 Ft/s	2.13 Ft/s	2.62 Ft/s
6	1	:	5	20	0.24 Ft/s	0.28 Ft/s	0.35 Ft/s	40	0.48 Ft/s	0.57 Ft/s	0.7 Ft/s	120	1.44 Ft/s	1.7 Ft/s	2.09 Ft/s
7	1	:	6	16.66666667	0.2 Ft/s	0.24 Ft/s	0.29 Ft/s	33.33333333	0.4 Ft/s	0.47 Ft/s	0.58 Ft/s	100	1.2 Ft/s	1.42 Ft/s	1.75 Ft/s
8	1	:	7	14.28571429	0.17 Ft/s	0.2 Ft/s	0.25 Ft/s	28.57142857	0.34 Ft/s	0.41 Ft/s	0.5 Ft/s	85.71428571	1.03 Ft/s	1.22 Ft/s	1.5 Ft/s
9	2	:	1	200	2.4 Ft/s	2.84 Ft/s	3.49 Ft/s	400	4.8 Ft/s	5.67 Ft/s	6.98 Ft/s	1200	14.4 Ft/s	17.02 Ft/s	20.94 Ft/s
10	2	:	3	66.667	0.8 Ft/s	0.95 Ft/s	1.16 Ft/s	133.333	1.6 Ft/s	1.89 Ft/s	2.33 Ft/s	400	4.8 Ft/s	5.67 Ft/s	6.98 Ft/s
11	2	:	4	50	0.6 Ft/s	0.71 Ft/s	0.87 Ft/s	100	1.2 Ft/s	1.42 Ft/s	1.75 Ft/s	300	3.6 Ft/s	4.25 Ft/s	5.24 Ft/s
12	2	:	5	40	0.48 Ft/s	0.57 Ft/s	0.7 Ft/s	80	0.96 Ft/s	1.13 Ft/s	1.4 Ft/s	240	2.88 Ft/s	3.4 Ft/s	4.19 Ft/s
13	2	:	6	33.33333333	0.4 Ft/s	0.47 Ft/s	0.58 Ft/s	66.66666667	0.8 Ft/s	0.95 Ft/s	1.16 Ft/s	200	2.4 Ft/s	2.84 Ft/s	3.49 Ft/s
14	2	:	7	28.571	0.34 Ft/s	0.41 Ft/s	0.5 Ft/s	57.14	0.69 Ft/s	0.81 Ft/s	1.0 Ft/s	171.43	2.06 Ft/s	2.43 Ft/s	2.99 Ft/s
15	3	:	1	300	3.6 Ft/s	4.25 Ft/s	5.24 Ft/s	600	7.2 Ft/s	8.51 Ft/s	10.47 Ft/s	1800	21.6 Ft/s	25.53 Ft/s	31.42 Ft/s
16	3	:	2	150	1.8 Ft/s	2.13 Ft/s	2.62 Ft/s	300	3.6 Ft/s	4.25 Ft/s	5.24 Ft/s	900	10.8 Ft/s	12.76 Ft/s	15.71 Ft/s
17	3	:	4	75	0.9 Ft/s	1.06 Ft/s	1.31 Ft/s	150	1.8 Ft/s	2.13 Ft/s	2.62 Ft/s	450	5.4 Ft/s	6.38 Ft/s	7.85 Ft/s
18	3	:	5	60	0.72 Ft/s	0.85 Ft/s	1.05 Ft/s	120	1.44 Ft/s	1.7 Ft/s	2.09 Ft/s	360	4.32 Ft/s	5.11 Ft/s	6.28 Ft/s
19	3	:	6	50	0.6 Ft/s	0.71 Ft/s	0.87 Ft/s	100	1.2 Ft/s	1.42 Ft/s	1.75 Ft/s	300	3.6 Ft/s	4.25 Ft/s	5.24 Ft/s
20	3	:	7	42.857	0.51 Ft/s	0.61 Ft/s	0.75 Ft/s	85.714	1.03 Ft/s	1.22 Ft/s	1.5 Ft/s	257.1428571	3.09 Ft/s	3.65 Ft/s	4.49 Ft/s
21	4	:	1	400	4.8 Ft/s	5.67 Ft/s	6.98 Ft/s	800	9.6 Ft/s	11.34 Ft/s	13.96 Ft/s	2400	28.8 Ft/s	34.03 Ft/s	41.89 Ft/s
22	4	:	2	200	2.4 Ft/s	2.84 Ft/s	3.49 Ft/s	400	4.8 Ft/s	5.67 Ft/s	6.98 Ft/s	1200	14.4 Ft/s	17.02 Ft/s	20.94 Ft/s
23	4	:	3	133.3333333	1.6 Ft/s	1.89 Ft/s	2.33 Ft/s	266.6666667	3.2 Ft/s	3.78 Ft/s	4.65 Ft/s	800	9.6 Ft/s	11.34 Ft/s	13.96 Ft/s
24	4	:	5	80	0.96 Ft/s	1.13 Ft/s	1.4 Ft/s	160	1.92 Ft/s	2.27 Ft/s	2.79 Ft/s	480	5.76 Ft/s	6.81 Ft/s	8.38 Ft/s
25	4	:	6	66.66666667	0.8 Ft/s	0.95 Ft/s	1.16 Ft/s	133.3333333	1.6 Ft/s	1.89 Ft/s	2.33 Ft/s	400	4.8 Ft/s	5.67 Ft/s	6.98 Ft/s
26	4	:	7	57.143	0.69 Ft/s	0.81 Ft/s	1.0 Ft/s	114.286	1.37 Ft/s	1.62 Ft/s	1.99 Ft/s	342.86	4.11 Ft/s	4.86 Ft/s	5.98 Ft/s
27	5	:	1	500	6.0 Ft/s	7.09 Ft/s	8.73 Ft/s	1000	12.0 Ft/s	14.18 Ft/s	17.45 Ft/s	3000	36.0 Ft/s	42.54 Ft/s	52.36 Ft/s

Figure 30: *A resource shared to me by Samuel outlining every possible gear ratio and wheel size combination and their respective speed output. For context, red is generally too slow, blue is generally too fast, and green is in the "goldilocks zone" (just right).*

Using this spreadsheet, I concluded that the most ideal gear and wheel combination for our robot would be a 3:4 ratio (36T and 48T) with 2.75-inch wheels. This would result in an output RPM of 450 and a speed of 5.4Ft/s, which is on the faster end of drivetrains but still perfectly reasonable.

I then began my initial prototyping using Fusion 360 to CAD our robot. As someone with no prior experience with 3D modeling for VEX, I had to consult online resources and download a VEX library to begin my modeling. But, after a couple of nights I had a design ready:

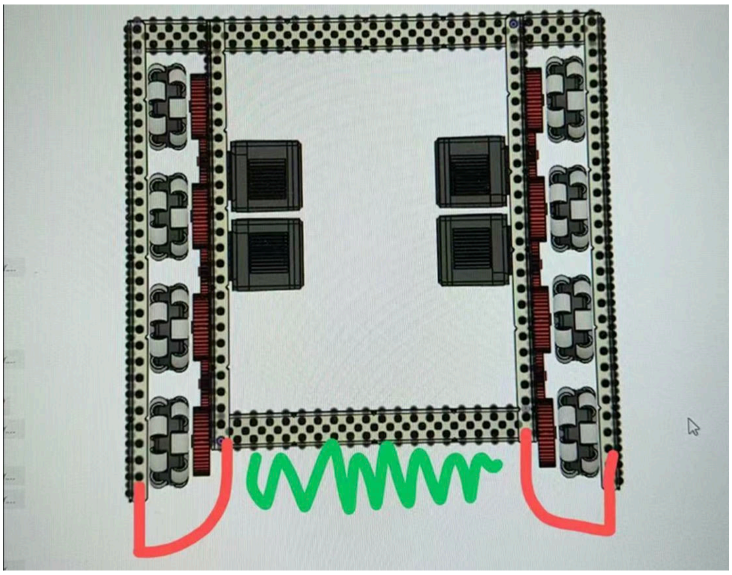


Figure 31: *My model in Fusion 360. Red arcs are laser cut pieces that we intend on adding and the green scribbles are where the intake rollers will be located.*

I also utilized spacers and a long, 30-long C channel to brace the drivetrain from the back. Here are some photos from the initial phases of building:

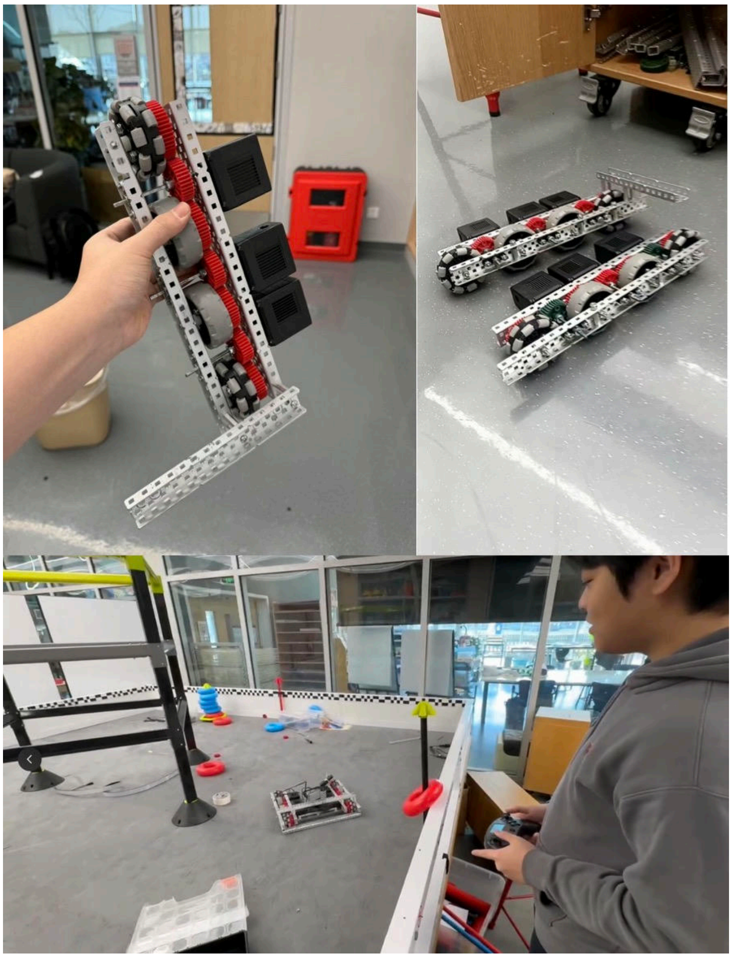


Figure 32: *N/A*

After tuning and resolving some initial friction issues, we had a drivetrain that was fast, easily maneuverable, and fulfilled all of our success criteria. After constructing our new base, we had to focus our attention to the other critical components, such as the intake, belt, and clamp.

15.2. Belt Design v2

`[William]`

We recognize the need for improvement and ongoing innovation. I(William) designed a versatile belt system capable of wall staking, flipping over mobile stakes, and standard scoring—truly a "jack of all trades." However, its performance fell short in all areas.

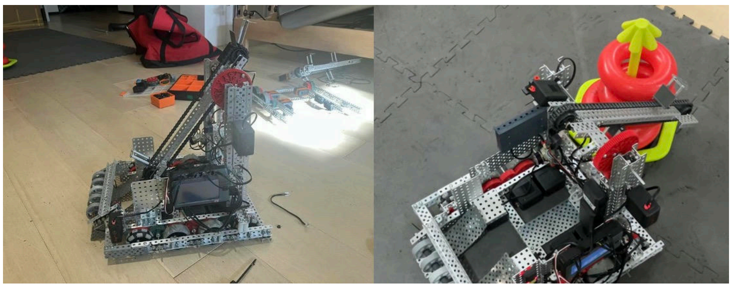


Figure 33: *New Belt Design*

When flipping the belt forward and back, it wobbles at the end of the movement. I checked my build, it was concise, then later I found that it was the very small gap between the motor insert and the shaft caused a little wobble that should be negligible, but the tiny wiggle got zoomed up because the "lever arm" (力臂) is too long. So, it is a fundamental problem that there is no way to fix.

To enhance functionality, team members decided to separate these tasks: wall staking will be handled by Lady Brown, a proven wall stakes mechanism, while the flip-over function will be assigned to "Knife." Although this requires additional time, the team is committed to making it work.

`[Emily]`
For the intake mechanism, we designed a system that efficiently gathers game rings while ensuring reliable performance during matches. The intake consists of a series of strategically placed rollers mounted at the front of the robot, allowing it to collect the objects as it drives forward. We utilized the 600 RPM blue motor to get the belt working, from the 5.5 watts old one. The increased performance made us faster process the donuts, which pairs with our fast drive train.

15.3. Intake Design v2

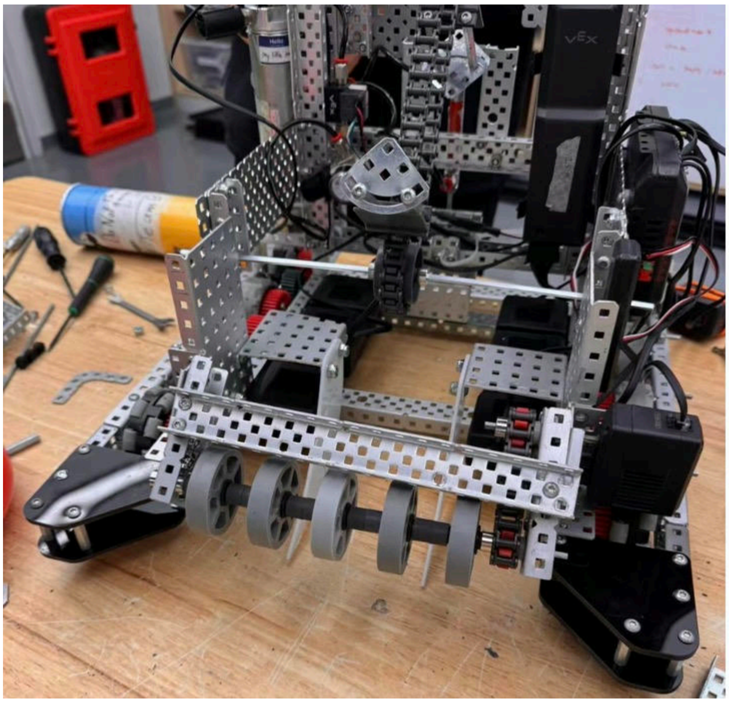


Figure 34: *New Intake*

`[William]`
I made the intake. Because it was built in my home (build over winter) so I got limited materials to build with. I think we should go easy with the intake, therefore I made it like this. The joint on the left is made of screw, and right is just a shaft. If there is not a lot of parts, I think it should malfunction less.

Pros:

- High efficiency of collecting the rings
- High accuracy on collecting the rings
- Ability to transmit the rings to conveyor belt
- Minimized use of materials on the intake

Cons

- Cannot bear heavy/strong pressures (Low quality material)
- The intake bars are not stable (Not enough spaces to screw)

15.4. Clamp Design v2

15.4.1. Mini Independent Inquiry (Jason)

`[Jason]`

The design of the clamp has changed many times throughout the season. But there were two leading schools of thought when it came down to the clamp. The two types of designs both relied on pneumatics but approached the problem in two different directions (literally). One design calls for a pneumatic piston to push down perpendicularly to the stake and secure the stake by pressing down on one of its corners. The other design pushes a flap down against the edge of the stake, clamping the stake against the main body of the robot.

Design 1 (Perpendicular piston)

Pros

- Simplistic
- Space saving

Cons

- Requires more precision
- Requires "funnels" to hold the stake to the correct orientation

Design 2 (Flap)

Pros

- Can be less precise
- Holds on to the orientation

Cons

- Requires more space
- More moving parts, harder to build
- Cannot pick up stakes with already donuts

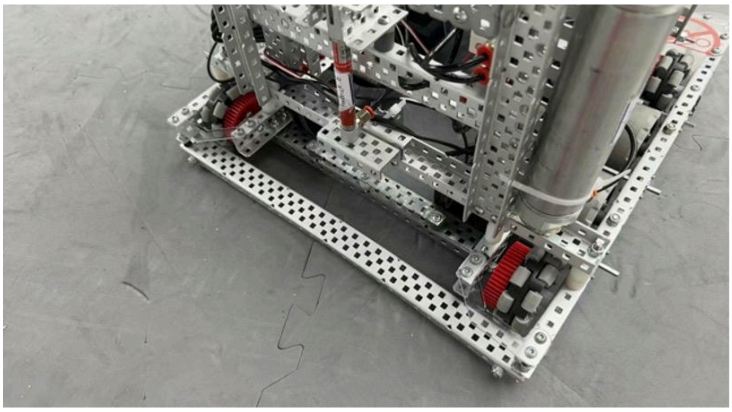


Figure 35: *New Clamp*

In the end, our team went with the perpendicular design as it is much more simplistic and when funnels are added, the downsides are negligible.

15.5. Pneumatics

`[William]`

This pneumatics system is in fact easier than we thought. We use only piston, so the only thing I need to sort out is the “solenoid”, or I would call it electric valve. Without the valve, the piston would not have been controlled, so it’s gonna connect to a 3-port wire and goes to the brain. The valve system looks like this.

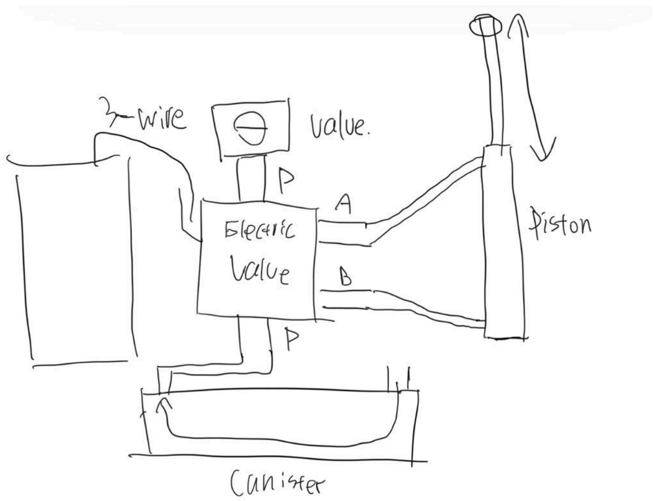


Figure 36: *Pneumatics System Diagram*

Maybe a little abstract but yet it should tell all the necessary parts. The new electric valve contains a “pathway” (P in the picture) that you need something to block it at the end of the path (where the manual valve locates), otherwise it would’ve malfunction and eject air out from the P. That is basically the only difference than the legacy one, so I quickly got started with it. Then you have A and B to connect to the piston since the piston also got 2 places to plug. Once that is done, get another P connect to the canister. That is pretty much it.

16. Reflect and Share - Post-ACAMIS Reflections

[Patrick]

Another tournament, this time official, after sadly being eliminated in the semi-finals and happily winning the Innovation Award for our documentation, the team has come to yet another unanimous decision to rebuild our robot, this time keeping the drivetrain. After observing our bot's in-round performance and other robots built by our talented opponents (though some of dubious student-centered nature), we have come to some further conclusions regarding our bot moving forward.

- Belt

The belt we have created is largely functional, but an error not found on our old belt now exists in which a donut in the process of being conveyed may randomly fly out, causing slow-downs and occasionally jams. As the programmer of the team, I cannot be certain as to the reason of this shortcoming, but my guess lies on the tension or speed of the belt, thus, we have come to a decision to remake it. (Update: After researching other robots created by other teams, I've recommended to the team we add two flap like plastic components that may help with this issue.)

- Intake

Though functional, our current intake has a glaring inadequacy – its shape. Besides being slanted, it is also shorter than our old intake, causing minor hiccups during intaking, needing extra time to fully roll in the ring.

- Notebook

Although our notebook is decent, a shortcoming I've realized is that when we wrote notebook entries, we would often neglect the date, leading to many entries being devoid of a timestamp. Realizing this, going forward we as a team have pledged to be more diligent in our date-keeping.

- Programming

Not a physical or programmatical shortcoming, but rather one of process inadequacy, but I personally have realized that I do not have enough time to program a decent autonomous routine for the robot both for matches and for skills, thus, I have come to an agreement with the team in which more time will be allotted for me to focus on this sector of the bot's program.

[Emily]

Belt: There is not a big issue with the functionality of the belt for this design. But the overall issue with evolving our robot with the designs is the constrains of the belt being connected to 2 side c-channels. For the next design, we would like to make the belt self-standing (meaning that it does not rely on the sides of our robot) to allow less obstruction to the rest of our robot. We found this to be a big issue in adding a lady brown because, with the current situation the lady brown cannot extend high enough for us to reach the wall stakes.

Intake: There weren't any large issues with our intake, but the overall length of the structure was shorter than what we wanted, causing us to work with what we had. For our future intake, we want the c-channels that we plan to use to be longer and hopefully have a cutout poly-carb piece to protect the intake in case of any contact with other objects.

Structure: The overall structure of our robot is not the greatest in expanding/adding other functionalities to the robot. The back of the robot where the belt is mounted is something that we plan on getting rid of, as it creates nonfunctional space.

16.1. Individual Goal Reflection

`[Emily]`

Going back to time and design before you build, we have realized that these are necessary steps. Although we struggled on some aspects the overall results of this competition were good and a great learning curve for us.

`[Aiden]`

I have learned so much in this past month with building and designing, the aspect that is repeatedly said in our team is to "design before you build". This is such an important takeaway and having good time management.

`[Jason]`

I was not able to make it to APAC so this is my first tournament. It was a fun one, I was very happy with how the bot turned out though there is much room for improvement. Through this process I got a much better understanding of the game. I learned roughly how a tournament is run and learned to make quick improvements to the bot in case of an error. But in the tournament our bot ran very reliably something I am immensely proud of, this being my first official bot and all. In the tournament, looking around different teams and understanding the different approaches being taken by each team gave me a lot of inspiration and I hope i can implement them into the next design.

`[Eddie]`

I'm very delighted with the progress I'm making as a designer. After watching countless tutorials and reading through pages of forums, I modeled and built my first drive train! And, it seems to align quite well with our needs, being fast, durable, and taking up a smaller footprint.

One thing I learned, is that considering the physical constraints are EXTREMELY important when designing and modeling, as it doesn't matter how beautiful and innovative a design is if we don't have that part. So, I had to make a few iterations of our drivetrain after going to our robotics space and looking at what parts we had. Unfortunately, we lacked 30-long C-channels so I had to make do with slightly shorter ones.

Another thing I learned is that some things cannot be accounted for when CADing. For instance — friction. Friction is crucial when building a drive base, as it's what differentiates a good drivetrain from an excellent one. We had issues when assembling the drivetrain I modeled as the left side had unreasonably more friction than the right. This resulted in a robot constantly spinning in circles. Turns out the issue had to do with screw joints being too tight. We realized that when building drivetrains, it's best to give everything just a bit of wiggle room to avoid friction and overheating.

`[Patrick]`

Since my last reflection, in compliance with my goal, I've looked into what experiences programmers usually do, and I've discovered a lot of things that would greatly alleviate some of the problems I'm having now. One of the main ones is a PID (Proportional-Integral-Derivative) controller. From my understanding, this controller is used to help alleviate the error when the robot is moving around, which would be of great benefit to me since I've been experiencing a lot of inconsistencies in terms of the movement of the robot during autonomous coding, I want to try exploring these more advanced functions, but from what I understand it requires a lot of complicated math that as an admittedly not very bright person, could take a while. Another feature I've seen around is color rejection, essentially attaching a color sensor to chuck out rings of the opposing color, I believe I can take a crack at this, as it doesn't seem to be overly complicated.

Finally, as aforementioned, I believe one of the main points of improvement for us as a team would be time management, we often leave a lot of things to last minute, and I believe if we want our team to be truly successful, that is something we have to master. To finish off this reflection, I want to set myself some more goals:

- Attempt to learn and implement more complicated features (PID, color sensing, etc.)
- Improve on personal (and team) time management

17. Define and Inquire - Further Independent inquiries

17.1. ACAMIS North Robots

17.1.1. BIBA 20068A:

Something that we liked about their robot was the pneumatic for the lady brown, this allowed the rings to be pushed into the desired position to have the best results to score a wall stake.

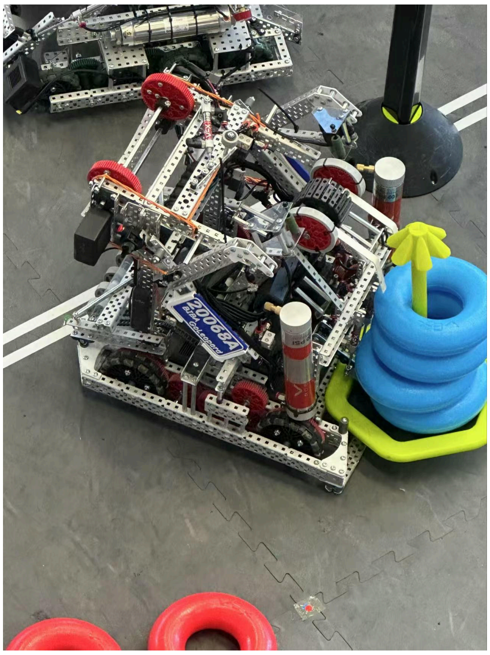


Figure 37: *Team 20068A's Robot*

January 22, 2025

17.2. General Independent Inquiry

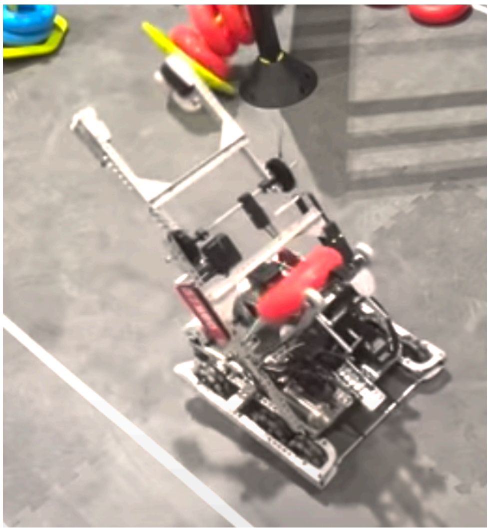


Figure 38: Lady Brown team:

18. Develop and Plan - Post-ACAMIS

18.1. New Intake

Feb 3rd, 2025

`[Patrick]`

Returning from ACAMIS, a feature I wanted to include in the new iteration of our robot was the ability to raise and lower the intake. I want the ability to do this because I believe it could benefit both the autonomous and driver control phases. For autonomous, it could be used to obtain only the top ring in a stack of two, allowing for more potential rings during the routine, and for driver control, it could be used to sort through a messy pile / potentially alleviate jams.

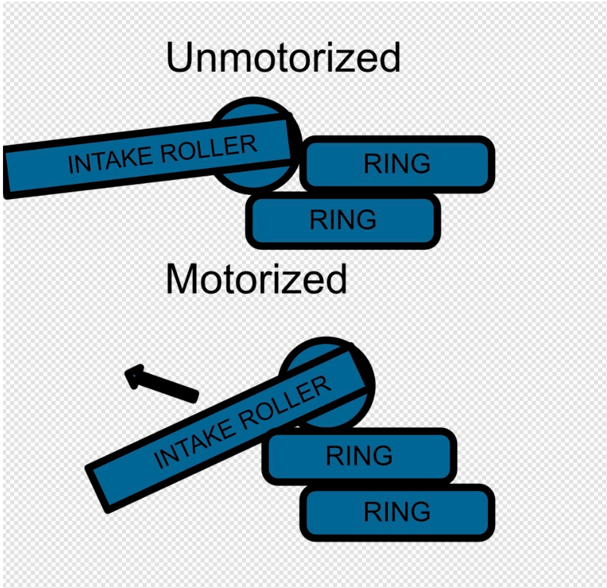


Figure 39: *Potential Idea for Motorized Intake*

`[Eddie]`

By connecting the intake and the belt, we can use an 11 wat motor to allow it to be quicker and more accurate. It also saves 5.5 watts and saves less power. Although it uses the same motor, the gear ratio is different and thus the chain is slower and the intake is faster.

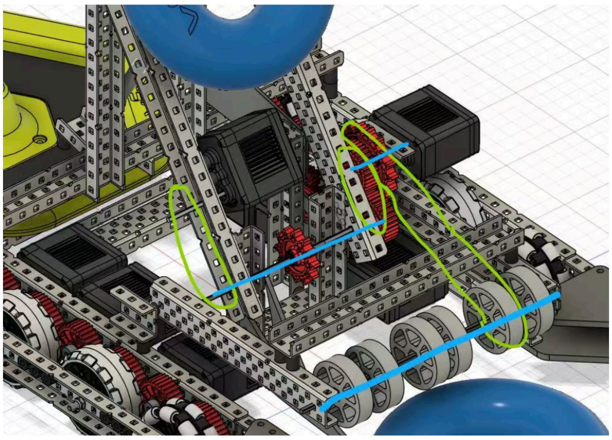


Figure 40: *CAD Intake*

18.2. New belt

Feb 5th, 2025

`[Emily]`

- Change the overall structure of the belt, as it was originally connected to the sides of the robot and now is free standing. This will allow fewer limitations in the overall building process.

`[Sophie]`

After ACAMIS, we realized that the belt had too much friction an the angle was not the best. The friction causes wear on the gears and a decrease in speed. It also creates overheating because of the torque and how the motor is using a lot of of energy.

`[Aiden]`

When we started planning on making the belt, we decided to make the slop steeper so it could score the mobile goal easier. One improvement we made in the new belt was CADing the design before building, which was mostly done by our team lead Eddie.

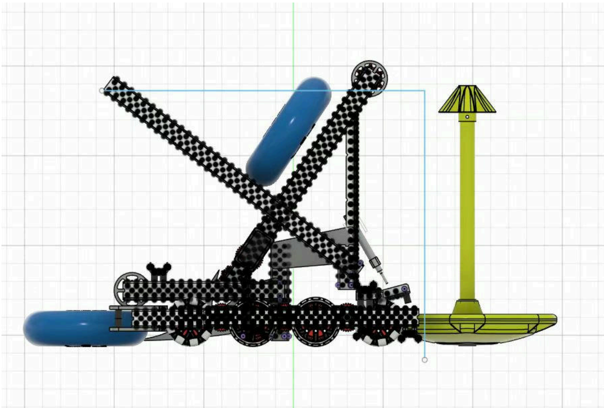


Figure 41: *CAD Robot Side view*

18.3. New hooks for belt

`[Emily]`

- Originally, we used a pivot gusset as the hook for our belt, which worked pretty well. But it was too wide to fit through the c channels of the support of our belt, which caused us to hand cut prototypes.
- We eventually ended with these triangle hooks which allowed the belt to have a greater accuracy in scoring the ring onto the mobile goal.

Figure 42: *Conveyor Belt Protrusions*

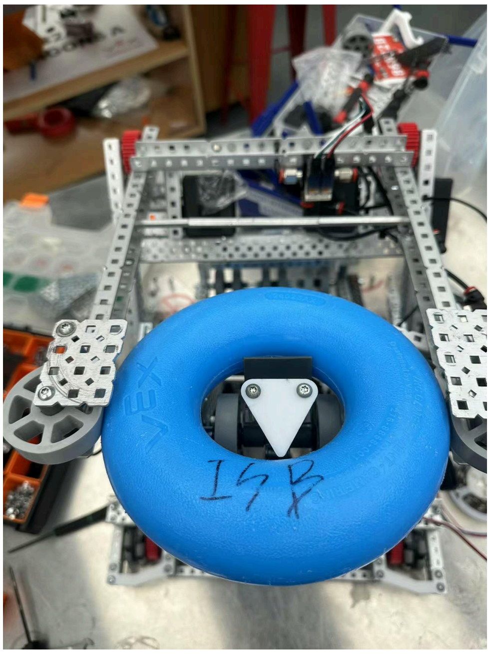


Figure 43: *Lady Brown top view*

18.4. Climb

Feb 4th, 2025

`[Patrick]`

After (admittedly) browsing YouTube for an excessive amount of time, I've devised a potential solution for creating a climb for our robot, if time allows, I think it is worth attempting. In our previous attempts to devise a plan for climb, we've been stopped by the lack of torque provided from the amount of spare wattage left after completing the essential functions (intake, belt, etc.), the solution I've realized is some form of gearing that can be activated via a pneumatics piston, which would push down and attach a gear train onto the existing drivetrain gears, essentially allowing us to tap into the torque of the six motors for climb.

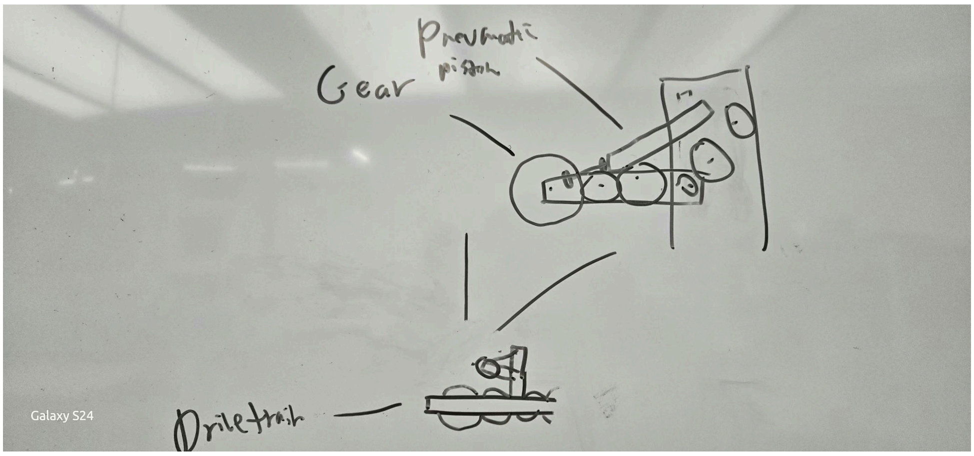


Figure 44: *Crude diagram of my idea*

Moving up from that chain, I envision the gears leading to a wheel with a roll of string, which is then attached to another roller at the top of a C-Channel mounted on a guiding rail.

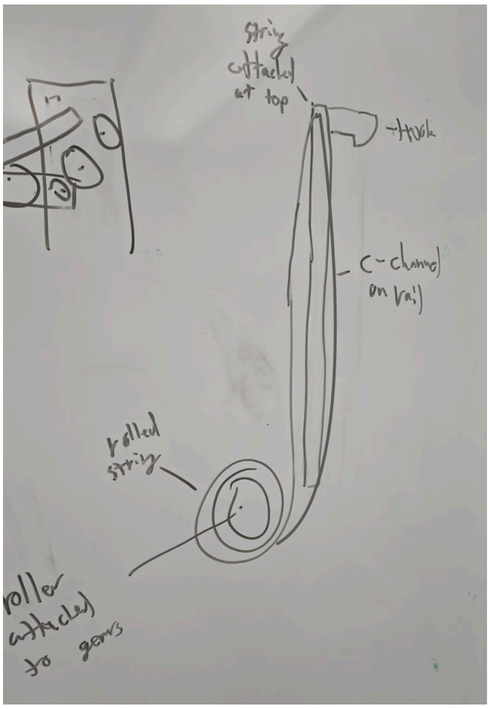


Figure 45: *Crude diagram of my idea pt. 2*

Feb 6th, 2025

`[Patrick]`

After much deliberation, I've decided to shelve this concept unless we have sufficient time to further develop this idea. Conventionally, the drivetrain would be designed with climb in mind, and taking on this feature to an existing drivetrain would be difficult; additionally, I was unable to visualize how the string roller mechanism would work out, and I concluded that it was not worth spending more time on unless we had some to spare.

18.5. New clamp

`[Sophie]`

- After rebuilding the robot, the clamp design was also revised. We looked back on the two options of perpendicular piston and hinge piston and decided to change the design to the hinge clamp due to the perpendicular piston seeing damage during ACAMIS.
- The new design consisted of a screw joint connected to a C channel which then allows the C channel to pivot. The piston is then attached to the C channel to control the movement.
- Although the original plan was to only have one piston, it did not turn out that way in the end.

18.6. Lady Brown

`[Emily]`

- Everything below on our robot was to be compatible with our lady brown as that was a big issue with our old robot.
- For 1st lb design we used 2 short c-channels and friction pads to be able to hold the rings but overall it didn't work so well, so we plan on using poly carb pieces and either flex wheels and friction pads to remake/design the lady brown

18.7. Doinker

`[Emily]`

The doinker was an important part of our ACAMIS robot, it allowed the robot to extract the rings out of the negative and positive corners. We used a piston to activate the doinker which overall worked pretty well, but we might have to keep in mind the extension limit.

`[Patrick]`

In my personal opinion, I believe the doinker/knife is fine as it is, but it could be improved by tacking additional functionality onto it. For one, adding a plate on top of it could give it the ability to sweep, and for two, if we add a small pertusion at the end of the could allow it to grab mobile goals (for the auton winpoint).

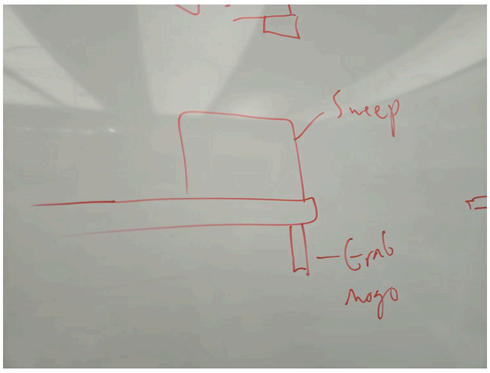


Figure 46: *Diagram of doinker ideas*

19. Create and Improve - Post-ACAMIS

19.1. Belt

Feb 8th, 2025

[Emily]

Friction Problems - The tension between our was most likely caused by excessive tension that can affect how the belts engage with the system. Too much tension can lead to increased friction and wear, while too little can cause slippage.

[Aiden]

Friction issues - The friction issue slows down the speed and efficiency of the belt system, adding on the frictions also causes the motors attached to belt overheat quickly compared to other motors. One of our solutions to this issue was lowering the tension of the belt by loosening the chain.

19.2. Intake

Feb 9th, 2025

[Aiden]

We built the intake system mainly with 2 C-channels connected to the belt system, and we connected those two C-channels with another C-channel on the top. Then we added a shaft between the two C-channels that had flex wheels and spacers on it. For the stability of the shaft, we also added barring, the barring added some friction on the shaft but it was not a big issue. To run the intake system we decided to share one motor with the belt. We connected the shaft that was on the belt system to the intake system and added gears on both sides of the intake with shafts to use chains and connect them. This solution supported us in reducing the usage of motors on the intake/belt system so we could use the motor on other mechanisms. Additionally, we added a pneumatic tank on the top of the intake, the pneumatic tank added counter weight that made it easier to take donuts in.

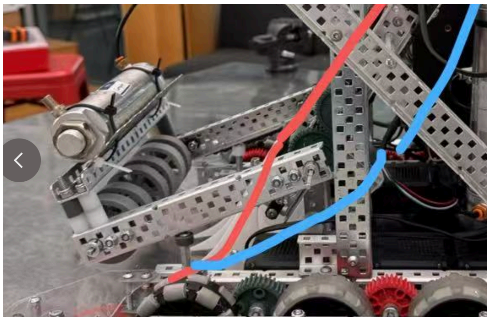


Figure 47: *The intake system connected with belt system*

19.3. Clamp

Feb 10th 2025

- While building the clamp and testing it, the piston quickly gave out after only a few trials or when the weight of the stake became too much. This meant that one piston was not enough to power the clamp effectively. this meant the design needed another piston in order for the clamp to be satisfactory.
- After this, we also decided to buy some pneumatic block thing which could help us reduce the piston from a double action (control both up and down) into a single action (only pushes down). this saves a lot of air in the pneumatics but also loses the ability to retract the piston. to

counteract this, a rubber band was placed in the clamp which allowed the clamp to retract after being extended.
- On top of that, the length of the C channel and the standoffs went through many going from 4 by 2 to 3 by 2.

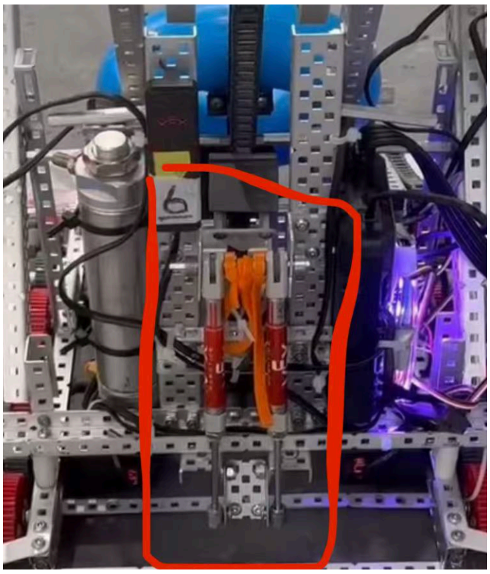


Figure 48: *Clamp design V1*

19.4. Lady Brown

Feb 14th, 2025

`[Emily]`

-We originally used 3 pieces of c channels, with 2 being horizontal and 1 being vertical. With friction pads attached to it. This doesn't allow stability and accuracy when holding a ring. -We then changed the front part to be 2 pieces of 3x4, l-shaped 3x2, and flex wheels as a holder for the rings. This allowed the ring to be able to be held in place, but not front enough to place it in the right position to be able to score a wall stake.

`[Sophie]`

One issue we encountered when building the lady brown was finding a viable solution to mount the ring onto the lady brown mechanism. Since the ring would rotate down along with the protrusions on the conveyor belt, there is a specific point where the ring position is parallel with the ground. If the reception point is not precise, the lady brown would block the regular rotation of the conveyor belt and hinder the intake process. We tested several methods to grasp the ring as it rotates upward on the conveyor belt. However, there was not enough friction between the two channels to grasp the ring. Another issue was that the ring could not reach the tip of the lady brown arm to reach the height of the wall stakes goal. As the ring can only be rotated onto a position on the lady brown that is not inwards enough, we tried installing a piston to push the ring further onto the hold for greater grasp.

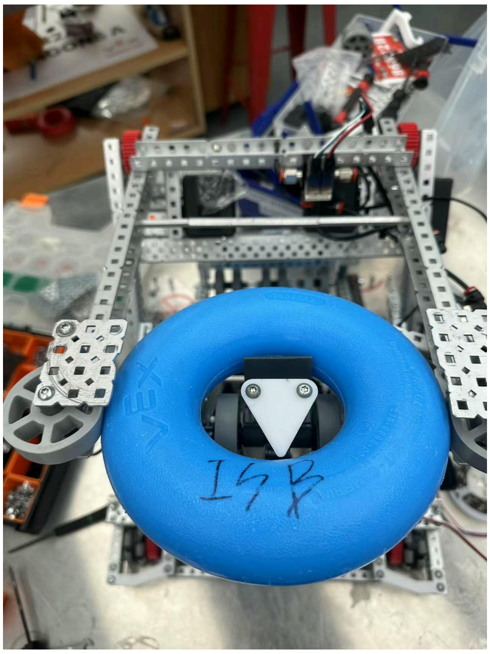


Figure 49: *Insert Caption*

19.4.1. Lady Brown Pneumatics Holder

Feb 15th, 2025

`[Patrick]`

Realizing the issue present in our lady brown, we decided to create a retracting and extending pneumatics component to help push the donut in place.

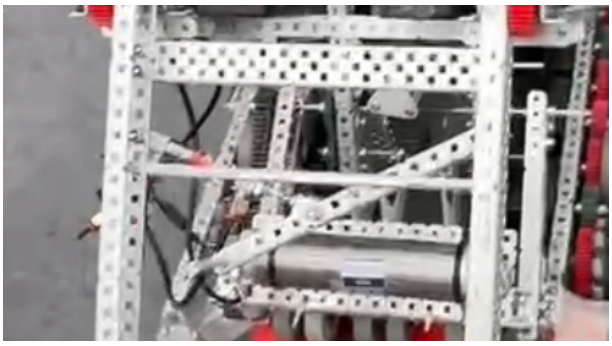


Figure 50: *William's first iteration of the lady brown pneumatics holder*

Pictured above is the first iteration of William's pneumatics component, it served its intended purpose quite well, but it had a flaw: it would collide with the wall stake bar, preventing the lady brown from going down further, based on this, we decided to strip it and rethink the lady brown.

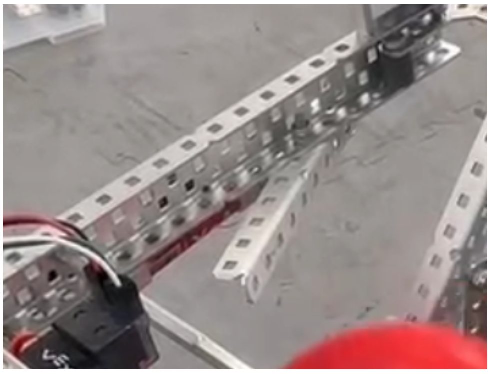


Figure 51: *Patrick's (unfinished) lady brown pneumatics holder*

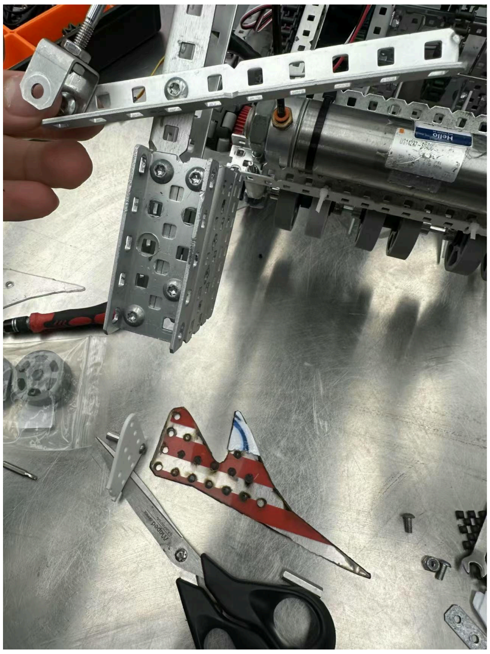


Figure 52: *Patrick's lady brown pneumatics holder*

After much consideration, I devised the pictured design, with a first class lever with the pneumatics piston on the outside, alleviating the aforementioned issue.

Around this time, Jason experimented with using one piston to clamp down on the ring from both sides (pictured below). He got far with this idea, but ultimately scrapped it due to 1. space constraints, 2. Overengineering. lol, 3. the travel provided by this mechanism wasn't sufficient in testing.

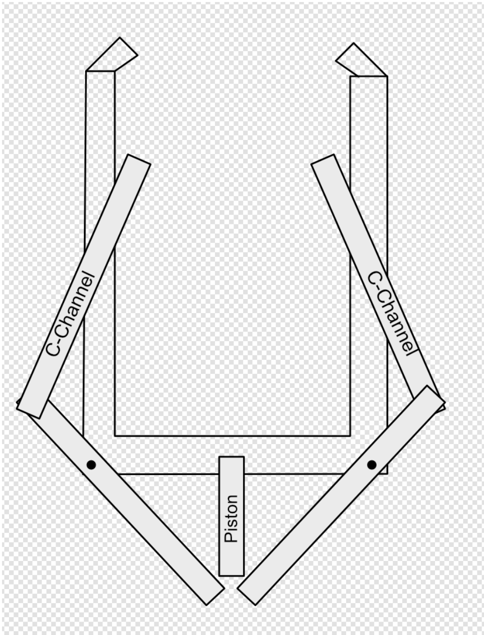


Figure 53: *Diagram of Jason's Pneumatic Module*

We ultimately reverted back to my design, and experimented with adding a plate on the channel.

19.5. Doinker

Feb 18th, 2025

`[Aiden]`

The doinker was detached after ACAMIS, and we have thought about remaking a doinker. For the doinker we hope to add a "paddle" in the front of it to allow one extraction of the rings. This not only creates efficiency but has a higher accuracy than a standoff a mushroom. Additionally, we want to decrease the consumption of pumped air since the doinker is sharing a tank with another mechanism.

20. Develop and Plan, Create and Improve - One week sprint

20.1. The Issue

February 16th, 2025

[Patrick]

After encountering a near catastrophic issue with our current robot (the angle of the intake was beyond repair), despite our best efforts, nothing could alleviate this issue. we made a major decision - rebuild the robot, better - this is an ambitious task, but we soon realized the choice was between doing nothing and have a non-functioning robot, or try to rebuild. We decided on the latter.

20.2. Rethinking Everything

February 16th, 2025

[Eddie]

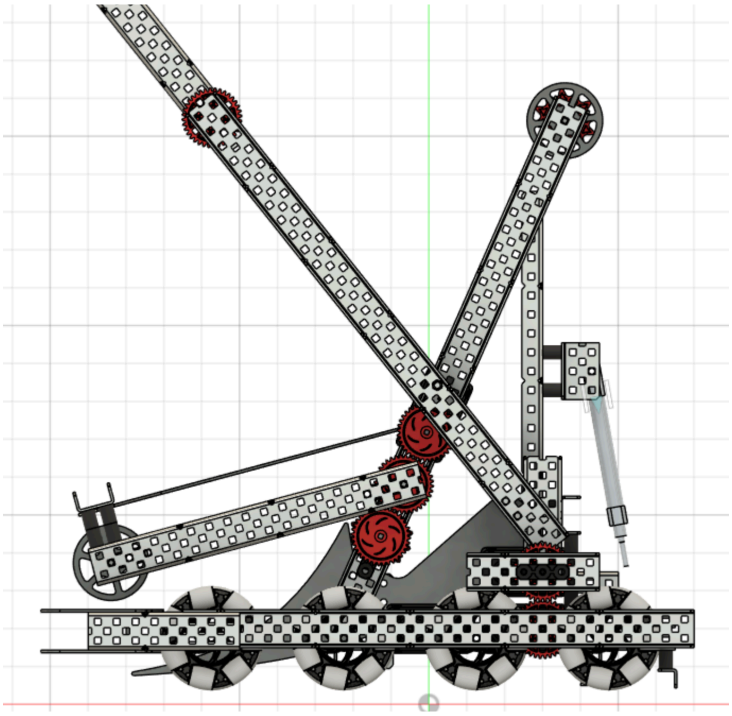


Figure 54: *Overall Changes*

When developing the new design for our robot in Fusion, the main change that needed to be made was to rotate the belt so that it is steeper and further away from the front of the robot. This makes for more space for the donut to gradually run up the ramp, something that was previously impossible with the geometry of the robot. However, making a change in the belt structure requires alterations in almost every other component of the robot. So, to minimize the effort needed to redesign everything, I tried to keep as many existing parts and designs as possible to make the transition as smooth as possible.

Doing this required some clever thinking: because of the nature of the vex holes being half an inch away from each other, many positions for the belt would be nearly impossible to construct without being a hassle to build or being structurally unstable. Thus, the solution to this was to use one long custom plastic piece to act as both structural support for the belt and lady brown but also act as the ramp for the donuts. This also minimizes our plastic waste while cleverly making the most out of the limited space we have.

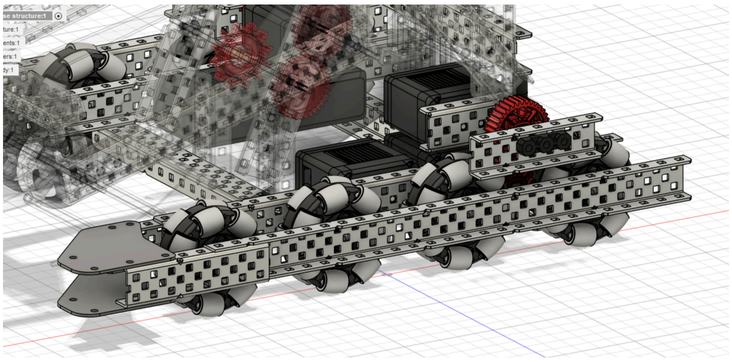


Figure 55: *Drivetrain Changes*

One of the major inconveniences of changing the belt angle was that the front motors of the drivebase blocked the beam of the belt and rose above the ramp. This requires us to move the motor backwards. The reason why this is not ideal is because altering the drivetrain takes lots of time and effort, and in general stacking motors generates more friction than not doing so. However, considering that the front space of the robot would be vital for our conveyor system, we were willing to bite the bullet and make this major change.

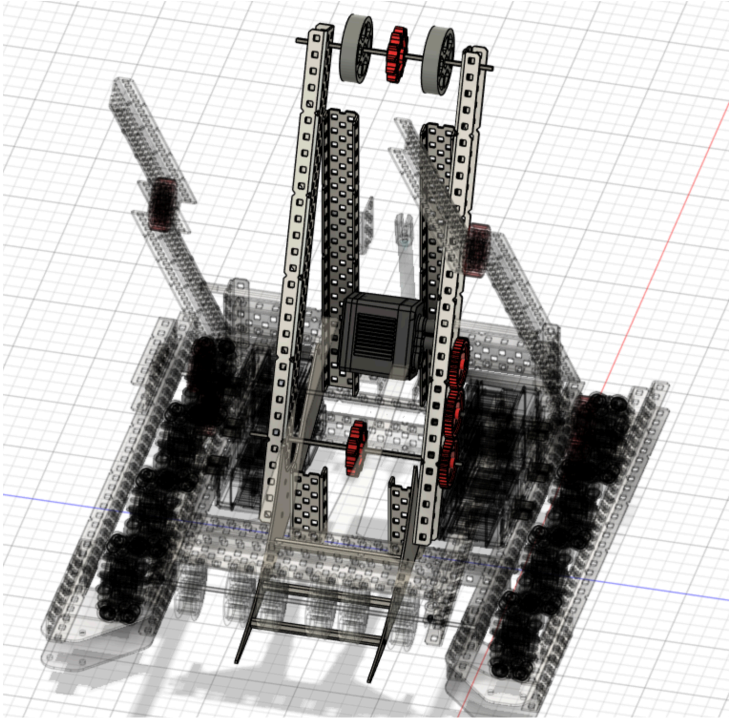


Figure 56: *Belt Angle Changes*

Learning from the mistakes we made when building our previous belt, I tried to reduce the friction induced on the system by keeping the number of shafts we have to a minimum. This step was definitely crucial, as prior to the rebuild the donut would slow down drastically near the ends of the belt due to the friction. There was also audible squeaking of components - a telltale sign of high friction and poor construction. The new design uses only two shafts for belt and intake parts, and one additional driving shaft to power them all, keeping the motor tucked inside the belt structure.

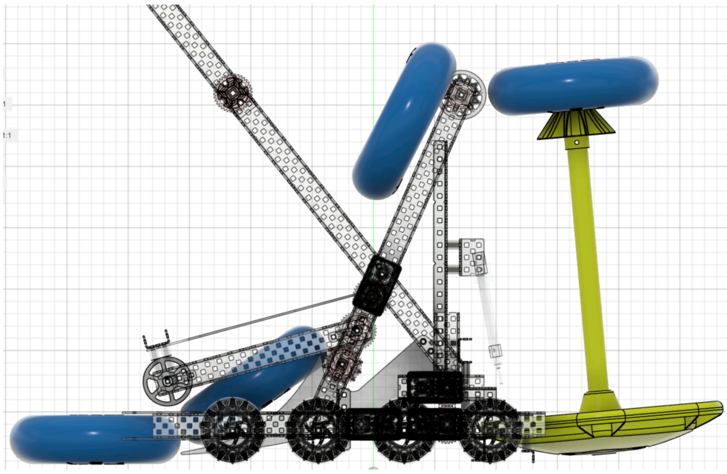


Figure 57: *Belt Position Changes*

The previous robot also had minor issues regarding the scoring of rings onto the mobile stake, due to the position of the mobile goal being a little too close to the end of the belt. The belt was subsequently designed to be a little closer to the front of the robot to alleviate this issue.

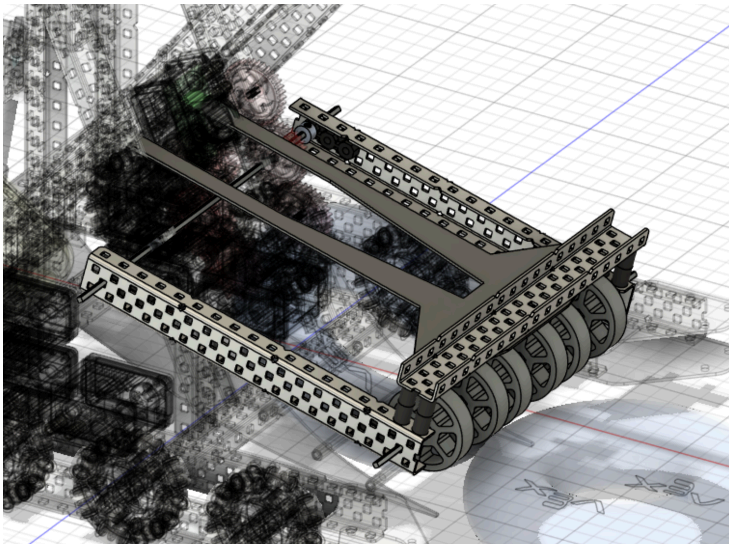
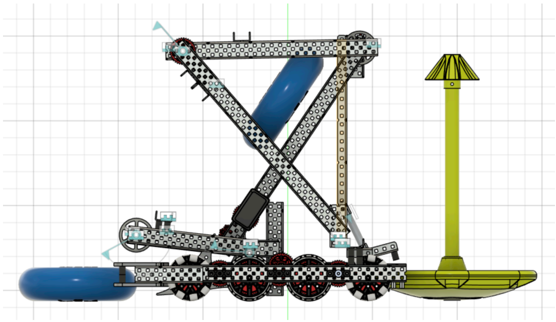
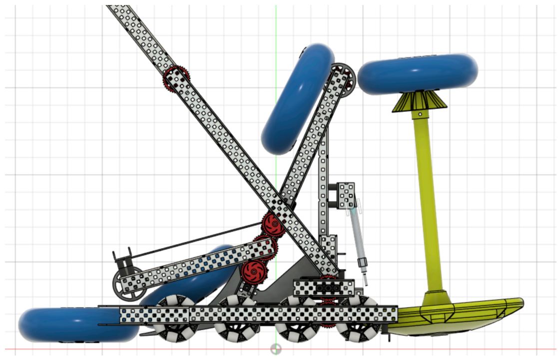


Figure 58: *Intake Roller Changes*

The intake rollers were also slightly modified, as the bar had to be extended to be 18 units long, compared to the previous 15. A piece of custom plastic cover was also designed with the suggestion of Patrick. The intention is for this piece to force the donut down, fixing a recurring issue we had of the hook hitting the donut during a narrow timeframe which would hit the donut and make it fly out. As the shaft and intake wheels will be easy to swap between components, this part will be relatively quick to build.

Comparison between old robot's side profile and new:

Old	New
	

20.3. Ideological Change

February 18th, 2025

`[Patrick]`

After gaining inspiration from one of the judges at ACAMIS North regionals, we fully took to adapting the scrum-building technique (delegating tasks so that each person can work on a different part simultaneously), this turned out to be of great benefit to us, while Eddie worked on changing the drivetrain, me (Patrick) and Jason worked on finishing the pneumatics component on the lady brown, while Sophie and Emily worked on rebuilding the intake and clamp according to the CAD Eddie had reworked.

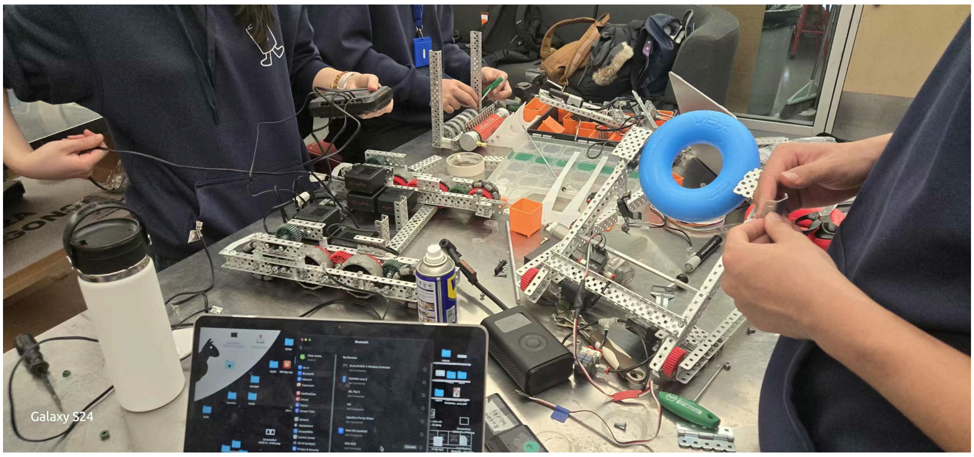


Figure 59: *Picture of us building in scrum*

pictured left to right: Eddie (out of frame), Emily, Aiden, Jason, Patrick (behind camera)

20.4. Building Logs

February 17th, 2025

[Patrick]

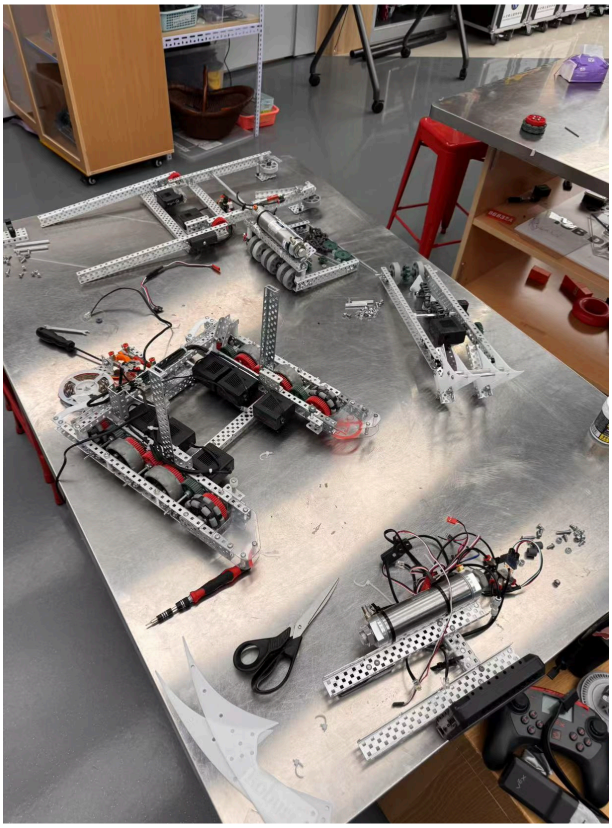


Figure 60: *Disassembly of the robot*

After realizing our plan, we quickly took to taking apart the necessary sections of the robot, detaching the lady brown, intake, clamp, belt and drivetrain.
During the day, Eddie used his free time to rearrange the motors on the drivetrain to allot for the additional space in front in line with Eddie's new CAD.

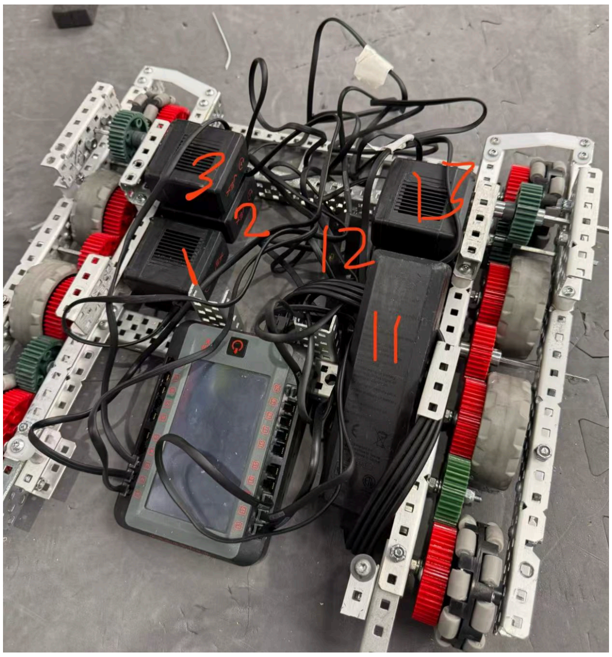


Figure 61: *Re-assembled Drivetrain*

<u>February 18th, 2025</u>
Being an official session, this time allowed for the majority of the team to be present at once, so we took advantage of this to attempt to implement the scrum building method, this ended up yielding great results, and allowed us to complete the necessary modifications to the lady brown, intake and drivetrain.
<u>February 20th, 2025</u>

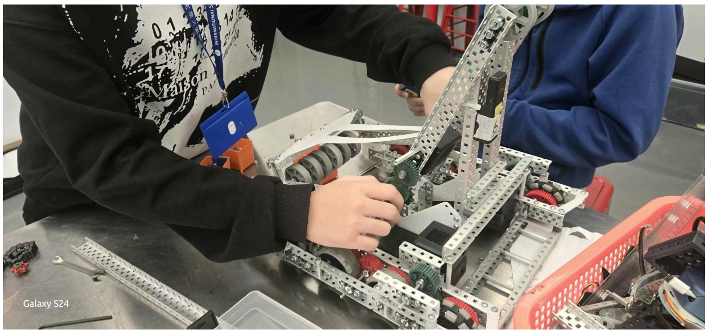


Figure 62: *Re-assembly of completed individual components*

Due to the fact that this notebook is being printed prior to the completion of our final session, I will simply outline the plans we have for this upcoming session. /

- Assemble knife/doinker
- Attempt to assemble passive hang
- Assemble all separately completed components
- Complete color sorting functionality

The fruits of these plans can probably be viewed during the competition.

21. Reflect and Share- Post-TIS Reflection

Feb. 25, 2025

21.1. Lady Brown

[Emily]

- Although we finished making the lady brown before TIS, during the competition we found a lot of different problems. This caused us to take it off our robot and reconstruct its front part. Initially, we used flex wheels as the ring holder, but we changed it to a few L-channels and zip ties, allowing a greater surface area to hold the ring.

[Sophie]

- During the tournaments, we changed the hold of the lady brown mechanism to improve its functionality. We were unable to rotate the ring fully onto the hold because of its positioning. In our previous design, the ring could be partially rotated onto the hold but requires the incorporation of a piston to push the ring further into the hold to reach the minimum height for wall stakes. As mentioned, the hold was changed to zip ties and steel L-shape sheets instead of the flex wheels. It has demonstrated a better performance in scoring the wall stakes and the fixation of rings when mounting the lady brown. Another design has been utilized to replace the piston on the lady brown to preserve more air in the canister for the clamp and doinker. The replacement of the piston consists of zip ties and rubber bands that are extendable according to the expansion of the lady brown. As shown in Figure 63, a rotational small L-channel is attached to the lady brown arm, connected to an elastic zip tie structure, which the other end is fixed onto the robot body. When the lady brown extends, tension increases on the zip tie structure, causing the small L-channel to rotate counterclockwise and push the ring further into the holds. The new design is similar to an automated structure, as the ring gets fully pushed into the holds and is prevented from falling out as the lady brown arm is raised.

[Jason]

In the tournament our team found ourselves with a lot of downtime. This paired with the fact we didnt have a functioning lady brown yet gave us an excuse to start working on it. We didnt have a lot of resources to use and was generally confined with what we had brought but with the help from the other teams (stealing) we managed to scrape together enough material to revise the lady brown. I noticed the other teams usually had a material on top and bottom keeping the donut in place while the sides provide the friction to keep it from slipping. we tried a similar approach but due to the limited material we had to resort to zip ties for the up and down hold and we didnt have enough time to be precise, keeping the donut in place was done by a pushed powered by rubberbands. There were many problems we ran into while trying to make the intake such as it interfering with the ramp, and the zip ties getting stuck on the shaft. These problems were quickly solved using more zip ties and cutting down material to not impact the ramp, after all priority still went to the ramp.

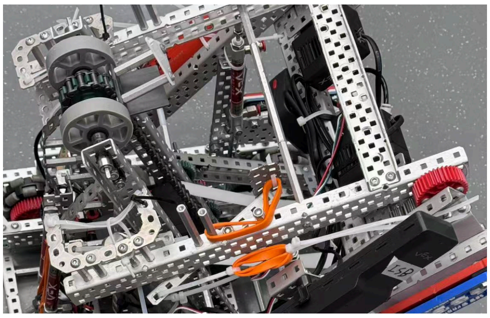


Figure 63: Close-up view Lady Brown

21.2. Belt

[Emily]

- The belt is still creating a certain amount of friction causing it to move a little slower than planned. The accuracy is a big part, and that was the main problem that was encountered. A lot of rings during the matches didn't all go on the mobile goal, creating a delay in our strategy. This also led to a major problem with skills, as we had to rescore certain mobile stake rings.

21.3. Doinker

[Emily]

- The doinker was overall not built the greatest, as it was flimsy and too short. The flimsy part came with the overall design and build. Which we fixed by adding more screws and changing part of where the pneumatic connected.

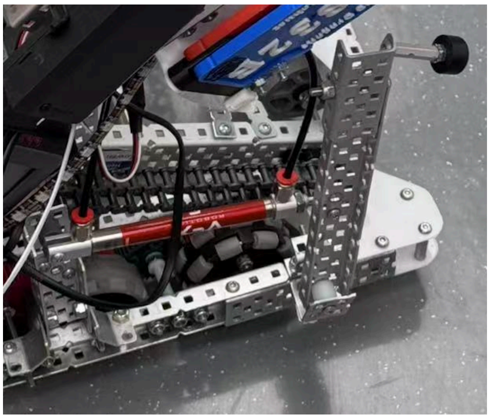


Figure 64: Close-up view doinker

21.4. Intake ramp

[Aiden]

- During the TIS competition, our intake ramp broke while programming the autonomous. Unfortunately, the tip of the intake ramp got stuck on the tape used to guide the autonomous directions. Even though the tip broke, the ramp still worked well but it would be best to make a new intake ramp so we can attach it instead of a broken ramp.

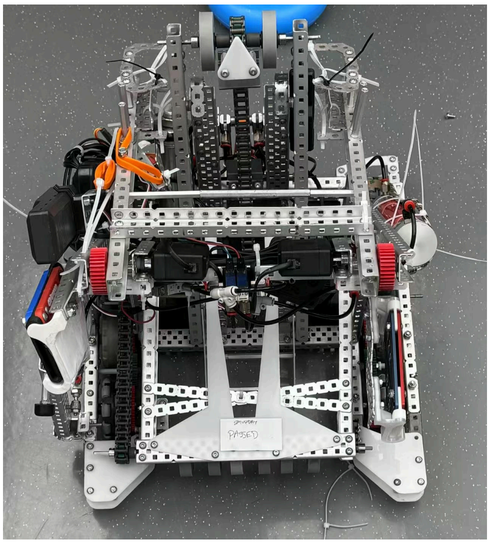


Figure 65: Post-TIS Robot Front View

21.5. Gameplay and Strategy

`[Eddie]`

- As the driver for the team, one of the biggest lessons I learned during the TIS tournament was the importance of good communication with alliance partners, as well as the need for sufficient practice before tournaments. On our first day of qualification matches, we went 0-4, losing all of our matches. I did some game analysis and noticed that some matches were not necessarily lost due to a lack of robot functionality, more of a lack of communication among alliance partners and poor strategy.
- Thus, going into ACAMIS Nationals, I hope to make the following changes:
  - Do preliminary scouting of alliance partners — talk to them before matches and discuss strategy, making sure everyone is on the same page.
  - During the matches, be very open to communication. If they prefer speaking Chinese, then speak in Chinese and have teammates who are proficient in the language on the field.
  - Develop potential short-hand names for certain plays. For instance, during the elimination rounds with Shanghai American School Pudong, our alliance had quick phrases to mean important things, such as "switch" to swap goals, and "pressure" to stop opponents from scoring wall stakes.
- With these changes in mind, hopefully our qualification and elimination matches go better during ACAMIS. I look forward to leaving my best on the field and playing a worthy game.

21.6. Individual Goal Reflection

`[Emily]`

TIS was the second to last competition of this season, and I have learned a lot. All the things that I found so important in my IQ season were also used in V5. This season, the overall growth of our team was vast. We learned to plan before we build, use time effectively, split the work, and give time to our driver to use. Although at first I didn't think that this was going to be that important, and the end it was our final verdict of this season and overall we've grown so much as a team and all individually have learned so much.

`[Aiden]`

I have learned so much about robotics and certain techniques of building and design techniques this season. I have learned so much from my teammates and gained so much experience. This year I dedicated a lot of time to my team and learned that time efficiency and cadding before building is critical.

[Jason]
First tournament out of Beijing. Very fun. Saw so many other designs that I would have never even imagined, Including inspiration for the lady brown which we hastily put together with zip ties and spacers. Although never used in-game even though it was functional and ready in the last game, it showed great promise. The building process of the bot was also very interesting as we completely built up the bot from nothing in less than a day thanks to our team working on separate pieces before putting everything together allowing us to have a better bot. The only down side was the software couldnt match the hardware and Patrick had to rush an auton (poor guy). nevertheless, with the game experience in this tournament the text tournament will be better since we know our bot better. One aspect I think we can work on is making the lady brown more consistent and easier to use. Taking inspiration from BIBA bot, using Pneumatic pistons to position the lady brown would be a great way to do it I think. Overall, I learned a lot from my fellow members and the other teams we went against.

[Eddie]
This kind of overlaps with my post-TIS reflection, as I believe that good communication and having a clear head while driving will allows us to excel during ACAMIS. But, I'd like to spend some time thinking back to my goal of improving my CAD skills.

Throughout the season, I went from having virtually zero knowledge of Fusion to designing our entire robot with it. I'm extremely proud of that achievement. I also dived into figuring out the properties of different types of plastics that we could use for our robot. I ended up settling with POM because it does not create any toxic fumes when laser cutting if handled properly, unlike polycarbonate plastic. I learned how to turn my 3D designs into .dxf files for laser cutters to read and process. I also spent some time working on decorative aspects of our robot, such as the license plate and controller modifications that get 3D printed. These are all great outlets for me to hone my modeling skills.

One of my biggest prides with our latest robot at TIS and ACAMIS is that I organized all the parts of the robot into component folders, making the design much more accessible to my teammates and easy to modify. Prior to that, I had a long dropdown menu of literally every single part of the robot. Now, because of the clean organization into components of the bot, I'm able to show my fellow designers and builders through the Fusion file without them getting lost in a sea of digital VEX parts. This also makes my life much easier when making changes too. I realized, sometimes, it's the small things such as naming folders and parts that can really make a big difference. Also, the design and assembly of our new robot was MUCH faster compared to the previous months because of all the experience we had. I was able to map out all the dimensions necessary quite easily because I had a clear image of what I wanted in my head. Keeping functioning components and only replacing inefficient ones also made the process much smoother. All in all, I'm super happy about the progress I've made over the course of the last 3-4 months, and I believe that these skills which I've developed will last me a long way, from future VEX events to potentially my career. :)

[Sophie]
During the TIS tournament, I saw very divergent designs of how teams approached scoring on the

mobile stakes and wall stakes. The most common designs were the ones with lady browns to mount the ring and store on wall stakes. One distinct design I saw was the four-bar lift that could elevate their ramp and rotate the ring directly onto the wall stake. During the tournament, there were also hero bots that functioned effectively in scoring and moving the mobile stake to intended areas. I learned that the nuance differences between robots with ladybrown mechanisms could affect their performance significantly. Top performing teams in the tournament had effective designs that we could learn from to incorporate into our robot. In conclusion, the tournaments assisted greatly in my learning process and there were many takeaways.

`[Patrick]`

Perhaps the final reflection I will be writing for this season, I just want to briefly look back on what I have done thus far in this season. One of the aspects we definitely did not improve significantly on is time management. As aforementioned, we had to completely rebuild our robot in a week, and as a result of that, the programming autonomous was left to the very last minute. Looking back, this has been an prevalent issue throughout this season, and I think starting off next season time management and foresight should be one of our biggest points of focus. Besides this, I have began looking into PID and whatnot, and have decided to use this summer to learn functions like odometry and controllers. I believe doing so could alleviate a lot of the major issues in programming we had this year. Overviewing my entire journey here this year, I think it was a period of great personal growth, and I have now have a clear direction for what improvements need to be made next year.

22. Software

`[Patrick]`

22.1. Introduction

This section is dedicated to documenting the process of my learning of the Pros Library, in addition to the later development of the final program of the robot.

22.2. Define and Inquire - Learning Journey

Before any code could be written, I first needed to understand the API, I had minimal prior experience with C++, knowing only the basic syntax and such, so this proved to be quite the challenge. Luckily, the Purdue Sigbots team has written very comprehensive documentation for their library.

https://pros.cs.purdue.edu/v5/api/cpp/index.html

This website contains all callable functions within the Pros API, to learn, I briefly read through all the subsections and revisited the website when I needed to know more about a specific function.

https://www.vexforum.com/

Another website I found hugely helpful is the VEX Forums, this forum consists of countless active members of the VEX community eager to assist in any problems you may have, I personally have gained much knowledge and assistance from this site.

In the past few months, I have fully comprehended how to create motor groups, create different control schemes/drive modes, control ADI (VEX Tri-Port) components, create autonomous routines, among others.

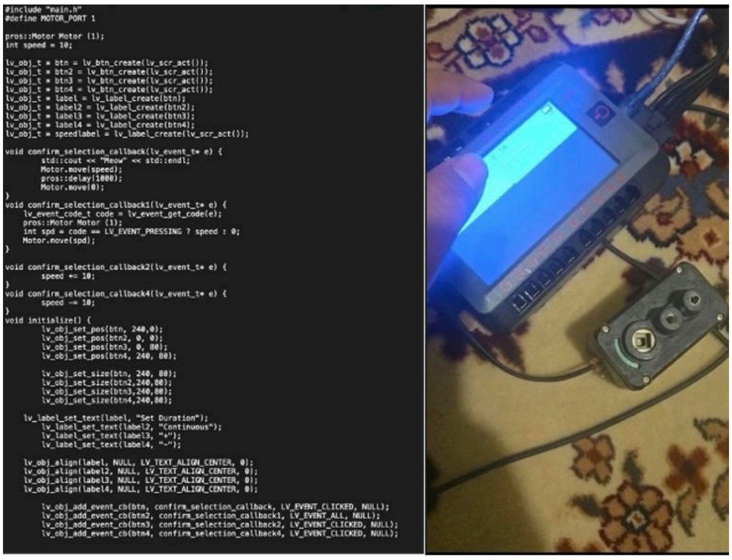
22.3. Develop and Plan, Create and Improve - Development Journey

22.3.1. September

The first month was spent mostly experimenting with the API and learning from Leon; I started my first program in Python then promptly switched to Pros upon evaluation of the different languages. A short detour was taken after observing Leon's program, which displayed a hyper-customized menu on the V5 Brain screen, upon inquiry of how he did it, he introduced me to LVGL, a screen library built into Pros.

Figure 66: *Image drawn on screen*

By the end of the session I had successfully displayed an image on the brain screen. For the remainder of the month, the bulk of my time was focused creating simple programs for the build team for testing, and HDI (Human Device Interaction), where I eventually completing this test program.



```
#include "main.h"
#define MOTOR_PORT 1

pros::Motor Motor (1);
int speed = 10;

lv_obj_t * btn = lv_btn_create(lv_scr_act());
lv_obj_t * btn2 = lv_btn_create(lv_scr_act());
lv_obj_t * btn3 = lv_btn_create(lv_scr_act());
lv_obj_t * btn4 = lv_btn_create(lv_scr_act());
lv_obj_t * label = lv_label_create(btn);
lv_obj_t * label2 = lv_label_create(btn2);
lv_obj_t * label3 = lv_label_create(btn3);
lv_obj_t * label4 = lv_label_create(btn4);
lv_obj_t * speedlabel = lv_label_create(lv_scr_act());

void confirm_selection_callback(lv_event_t* e) {
        std::cout << "Meow" << std::endl;
        Motor.move(speed);
        pros::delay(1000);
        Motor.move(0);
}
void confirm_selection_callback1(lv_event_t* e) {
    lv_event_code_t code = lv_event_get_code(e);
    pros::Motor Motor (1);
    int spd = code == LV_EVENT_PRESSING ? speed : 0;
    Motor.move(spd);
}

void confirm_selection_callback2(lv_event_t* e) {
        speed += 10;
}
void confirm_selection_callback4(lv_event_t* e) {
        speed -= 10;
}
void initialize() {
        lv_obj_set_pos(btn, 240,0);
        lv_obj_set_pos(btn2, 0, 0);
        lv_obj_set_pos(btn3, 0, 80);
        lv_obj_set_pos(btn4, 240, 80);

        lv_obj_set_size(btn, 240, 80);
        lv_obj_set_size(btn2,240,80);
        lv_obj_set_size(btn3,240,80);
        lv_obj_set_size(btn4,240,80);

    lv_label_set_text(label, "Set Duration");
        lv_label_set_text(label2, "Continuous");
        lv_label_set_text(label3, "+");
        lv_label_set_text(label4, "-");

    lv_obj_align(label, NULL, LV_TEXT_ALIGN_CENTER, 0);
    lv_obj_align(label2, NULL, LV_TEXT_ALIGN_CENTER, 0);
    lv_obj_align(label3, NULL, LV_TEXT_ALIGN_CENTER, 0);
    lv_obj_align(label4, NULL, LV_TEXT_ALIGN_CENTER, 0);

        lv_obj_add_event_cb(btn, confirm_selection_callback, LV_EVENT_CLICKED, NULL);
        lv_obj_add_event_cb(btn2, confirm_selection_callback1, LV_EVENT_ALL, NULL);
        lv_obj_add_event_cb(btn3, confirm_selection_callback2, LV_EVENT_CLICKED, NULL);
        lv_obj_add_event_cb(btn4, confirm_selection_callback4, LV_EVENT_CLICKED, NULL);
```

Figure 67: *Motor control program*

This program is capable of spinning a motor either in a continuous motion or at a set interval, with ability to increase and decrease the motor speed, all controlled by buttons rendered on screen using the LVGL Library. This was a huge capability milestone for me, marking the point in which I had become somewhat comfortable with the library and aspects such as motor control and controller mapping.

22.3.2. October

At the start of the month, I created three modes of driving, one in which all movement is controlled by the left joystick (named joystick drive), one in which each joystick controls the acceleration of the respective chain of motors (named tank drive), and one in which the left joystick controls the acceleration and the right joystick controls the turn angle (named arcade drive). Please refer to below diagram for clearer explanation.

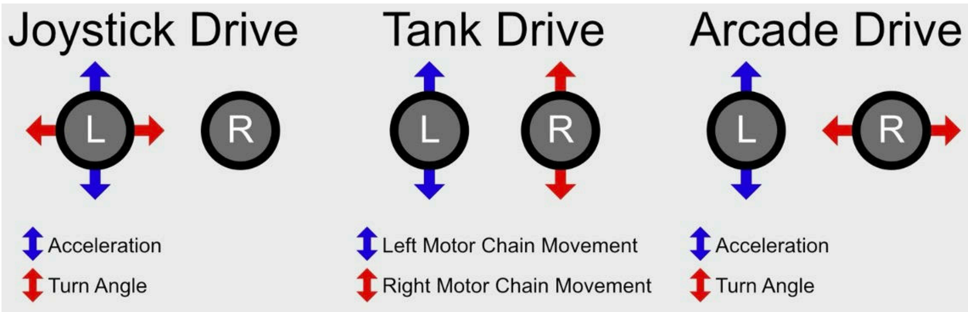
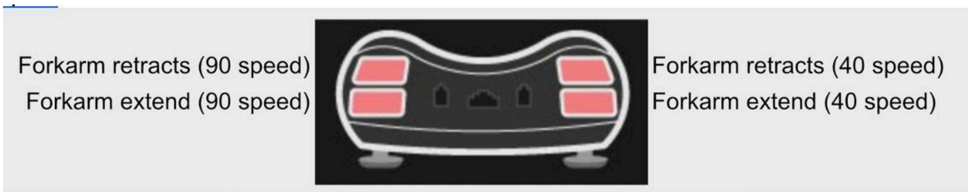


Figure 68: *Joystick control modes*

The bulk of the remainder of the month was spent writing then updating programs used to operate the Fork arm mechanism and drivetrain actively being constructed by the building team, adjusting functionality as the design is updated and remapping controls as requested by the driver. The initial program simply retracts/extends the arm at a set speed, which later evolved to using the shoulder buttons at speeds labeled in the diagram below, the speed of these buttons could be compounded, meaning if both L1 and R1 buttons are pressed, the Forkarm would extend at 130 speed.




```
// Arm control code
robot::armspeed = 0;
if (robot::master.get_digital(pros::E_CONTROLLER_DIGITAL_R1)){
    robot::armspeed += 90;
}
if (robot::master.get_digital(pros::E_CONTROLLER_DIGITAL_L1)) {
    robot::armspeed -= 90;
}
if (robot::master.get_digital(pros::E_CONTROLLER_DIGITAL_R2)) {
    robot::armspeed += 40;
}
if (robot::master.get_digital(pros::E_CONTROLLER_DIGITAL_L2)) {
    robot::armspeed -= 40;
}
robot::Arm.move_velocity(robot::armspeed);
```

Figure 69: *Arm control scheme*

On the aesthetic aspect, I replaced the current LLEMU (Legacy LCD Emulator) Screen used to display acceleration data with a customized menu more aesthetically pleasing.

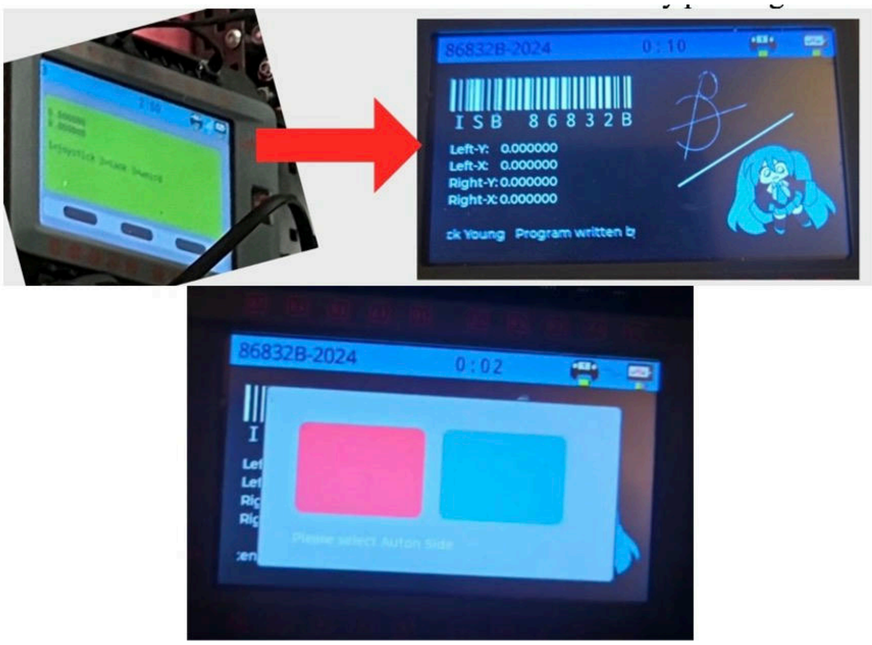


Figure 70: *Arm control scheme*

I also created a popup window prompting the user to choose the autonomous side, which currently has no use. The final change was an internal file hierarchy restructuring, I moved all screen related code into a separate file named "screen.cpp".

The final amendments made in this month included another long detour (ADHD moment) after learning about the legality of decorative components, namely LED strips, and spending many sessions attempting to append an LED strip to our bot, eventually succeeding.

A problem we have noticed since the initial prototype is that as the arm is extended or contorted according to its vertical movement, its horizontal is different, to counter this, the driver would need to manually account for this horizontal difference and slowly push forward the bot, William proposed that we automatically account for this and automatically correct in software. A problem with this is depending on the position of the arm (above or below midpoint), the motion needed to counteract the contraction is different. I began conceptualizing possible methods of correction and realized that it was possible to obtain the current motor position of the forklift motors, and using said number (1560) I can loosely define a divider line.

22.3.3. November

As this point in time, the team completely pivoted and began creating a new design after realizing the shortcomings of our previous design, as a result, a portion of my current program was cut and archived in preparation for the new design. As the team constructed various components of our now more conventional bot, I created code to operate the belt and intake, Initially a simple amendment in which the component would run if a button is held, before changing to a toggle/switch scheme (diagramed below).

Figure 71: *Current Control Scheme*

I also spent a considerable amount of time learning how to interact with the ADI components (VEX Tri-Port, used by components like Cortex modules and pneumatics) on the V5 Brain using the Pros Library, eventually creating a functioning pneumatics function that put our mobile goal component into use.

Towards the end of the month, we attended the APAC Robotics Invitational, which gave me ample opportunity to observe my program in action, and in reflection, I believe my program to be overall quite reliable, albeit lacking a few features observed in higher level programs

22.3.4. December

Not much was completed program-wise in this month, ISB hosted a scrimmage towards the beginning, and during this time I was able to create a rudimentary program capable of scoring two rings during the autonomous period.

```
pros::delay(200);
robot::MotorLeft.move_velocity(180);
robot::MotorRight.move_velocity(180);
pros::delay(800);
robot::MotorRight.move_velocity(0);
robot::MotorLeft.move_velocity(0);
pros::delay(100);
robot::Joshua.move_velocity(180);
pros::delay(1000);
robot::Joshua.move_velocity(0);
robot::MotorRight.move_velocity(0);
robot::MotorLeft.move_velocity(180);
pros::delay(1650);
robot::MotorLeft.move_velocity(0);
pros::delay(10);
robot::MotorLeft.move_velocity(180);
robot::MotorRight.move_velocity(180);
robot::TakeMeToHeaven.move_velocity(200);
robot::Joshua.move_velocity(180);
pros::delay(600);
robot::MotorLeft.move_velocity(0);
robot::MotorRight.move_velocity(0);
pros::delay(400);
robot::TakeMeToHeaven.move_velocity(0);
pros::delay(2000);
robot::Joshua.move_velocity(0);
```

Figure 72: *Current Autonomous Period Code*

All actions in this autonomous are written manually, which I am not proud of, but at the time of scrimmages, there was not ample time to do anything of more advanced nature. The scrimmage also made me realized the urgent need of sensors on our robot, as I essentially needed to code autonomous for a robot that was virtually blind and deaf. For the remainder of the month, little to no changes were made to the program, and I assisted more in the building process.

22.3.5. January

During the break, many teams decided to completely recreate their robot, including us, meaning I had to once again rewrite code. I ported all the code over to the best of my ability, preserving all features. The build team briefly added rotation functionality to the belt portion of the bot, and I created control code for said component, but it was later removed due to the wattage limit, thus my program followed.

I did further finetuning of the three aforementioned control schemes, adapting it not only to the new bot but also to the needs of our driver - Eddie, tweaking rotation sensitivity, etc.

We then proceeded to create a "knife" - essentially a channel that unfolds to be able to clear out the corners on the field. Our "knife" was powered by a piston, which gave me an opportunity to further familiarize myself with the ADI interface.

I also added indicators on the controller screen, showing the current state of both the intake and belt.

Figure 73: *Controller Screen*

A change in control scheme of the belt and intake specifically happened numerous times, I will not list them all, but as of current, they are as below.

Figure 74: *Current Control Scheme*

The intake control scheme is a but more convoluted, if the intake is off, depending on which button is pressed, it will spin in that direction. While spinning, if the other shoulder button is pressed, the direction will change, but if the same button is pressed twice, the intake will turn off.

22.3.6. February

Admittedly a quite uneventful month, I had committed myself more to design and building in an effort to expand my skill set, but when the program needed amendments, I of course did so.
Eddie was complaining that the driving program would decelerate at a rapid pace, causing the robot to tilt at a quite scary angle.

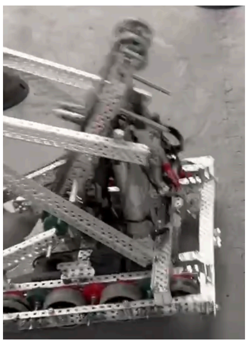


Figure 75: *Robot performing a sick but scary wheelie*

I consulted Leon on this matter, since he had somehow prevented this kind of tilting in his driving program, and from him I learned how to control the rate of change in velocity, allowing for a smoother driving experience.

```
static int lset = 0;
static int rset = 0;
static long long ltime = 0;
constexpr float slew_rate = 700;
void weird() {
        // get controller analog
    float lefty = (float) robot::master.get_analog(channel: pros::E_CONTROLLER_ANALOG_LEFT_Y) / 1
    float rightx = (float) robot::master.get_analog(channel: pros::E_CONTROLLER_ANALOG_RIGHT_X) /
    float dt = (pros::micros() - ltime) / 1e6f;
    ltime = pros::micros();
    if (lefty<robot::LIMIT_TOP && robot::LIMIT_BOTTOMHAHA<lefty) {
    robot::MotorLeft.brake(); rset = 0;
    robot::MotorRight.brake(); lset = 0;
    } else if (lefty>robot::LIMIT_TOP or lefty<robot::LIMIT_BOTTOMHAHA){
    float slew_range = slew_rate * dt;

    //this section was admittadly written with help from Leon, essentially, this section limits t
    rset = std::clamp(val: -lefty-rightx, lo: rset - slew_range, hi: rset + slew_range);
    lset = std::clamp(val: -lefty + rightx, lo: lset - slew_range, hi: lset + slew_range);
    robot::MotorLeft.move_velocity(velocity: lset);
    robot::MotorRight.move_velocity(velocity: rset);
    }
```

Figure 76: *Program excerpt*

The slew range here essentially calculates the maximum change in velocity the output should be able to have without tipping, then the `std::clamp` function ensures that the output given to the motor group is always limited to within this range.

On a less brain-pain inducing note, I had also written the code for our lady brown as well as a small pneumatics clamp that assists in pushing the ring in place. This sector of the program is quite simple, just using two shoulder buttons to drive the motor group, though Eddie did request a macro to help efficiently position the lady brown to where it needs to be.

```
// Lady Brown Load Position
void JiaBrownPositioning(){
    if (robot::master.get_digital_new_press(button: pros::E_CONTROLLER_DIGITAL_RIGHT)){
        isJiaBrownPositioning = true;
    }
    if (isJiaBrownPositioning) {
        float kp = 1.5;
        float error = -300 - robot::JiaBrown.get_position();
        robot::JiaBrown.move_velocity(velocity: kp * error);
        printf("Error is %f\n", error);
        printf("Position is %f\n", robot::JiaBrown.get_position());

        if (fabs(error) < 10) {
            isJiaBrownPositioning = false;
            robot::JiaBrown.brake();
        }
    }
}
```

Figure 77: *Program excerpt*

I initially implemented a simple solution, but a problem with it was that the variance between one run to the next was wildly different, and thus Leon helped me again by teaching me PIDs without the ID. Essentially, the destination value is set as the error, say I want to get to 300, and my motor starts at 0, if I move the motor by 10, then the error would become 290, so on and so forth. kp here is known as the proportional gain, which needs to be carefully tuned, a high kp will cause the program to respond swiftly, but may cause overshooting and result in oscillation, where as a low kp would be more accurate, but it would be unbearably slow. These issues could be lessened with the ID portion of the PID, but I still evidently have a lot of learning to do.

22.4. Reflect and Share - Programmer Self-Reflection

February 15th, 2025

Hi.

It's me, the programmer for Robust.

It's been quite an eventful few months, and this being my first real year in VEX Robotics, I felt the need to reflect on how I've written my program and things to improve on in the future. Truth be told, I'm quite proud of what I've managed to clobber together this season. Considering that I haven't touched C++ since the online class I was forced to take (and didn't listen to) in 5th grade and I haven't ever used PROS before, I'd deem my performance pretty okay. Of course, I would be improper if I don't credit Leon, ISB Robotics' lead programmer, though (like all programmers) lacking in expressiveness, when I had a question, he had an answer, and he was of great assistance throughout my journey. Now, moving on to some points of improvement; most apparently, my development did not follow DI,DP,CI,RS very strictly, and I think this was due to a combination of my personal lack of experience, and me waiting on the build team to finish the robot before programming. I've since realized that this mindset is quite wrong, and it often leads to things like programming everything in the last 4 hours before a competition (not good). I've also realized that as a programmer, I should be making more demands, not like "buy me bubble tea" demands, but "please put a rotation sensor here" demands, since I learned from experience that coding a blind and deaf bot is not very fun. Most importantly, I should also be less scared of math, towards the beginning of the year, Leon's frequent mentions of PIDs, Odometry, Pure Pursuit, etc. scared me a lot, but I've soon realized that if I want our robot to do cool things, I'd first have to learn the uncool math. To summarize, I think as a first year participant, I did decent, but I should not remain in this comfort zone, I need to improve if I want our team to improve, and especially since the only knowledgeable programmer is graduating, we need to try hard to fill that position.

22.5. List of Current Features

This list is all the features currently present in the program.

- Clamp Operation
- Belt & Intake Operation
- "Knife" Operation
- LED Strip Control
- Two Point Autonomous
- Drive mode Switching
- Custom Screen Display
- Lady Brown Functionality
- Color Rejection (In progress)

22.6. List of Planned Structures

- Autonomous for all four configuration

22.7. Folder Structure

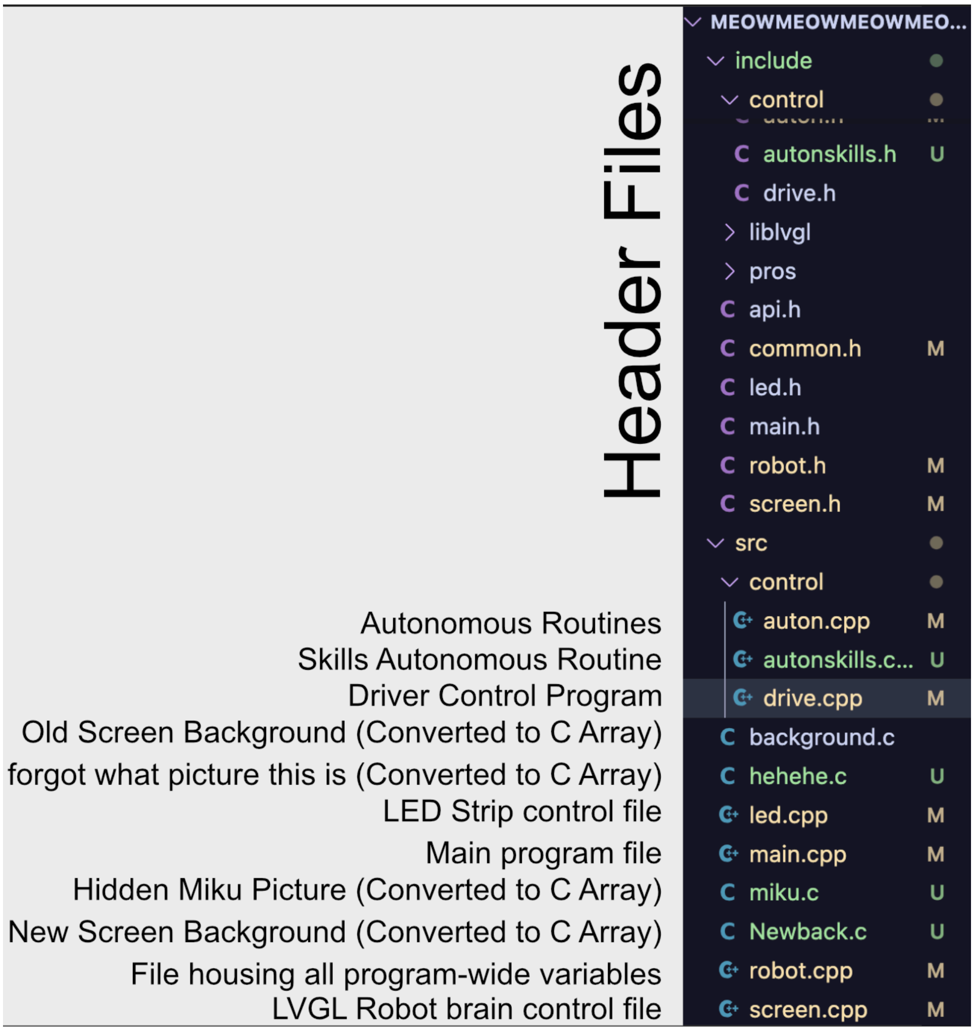


Figure 78: *Folder structure of current program*

This is the current folder structure of the program used, per guidance of Leon, all files are separated and categorized by functionality, and functions are all sorted into their own files for convenient calling and locating.

22.8. Brain Ports Map

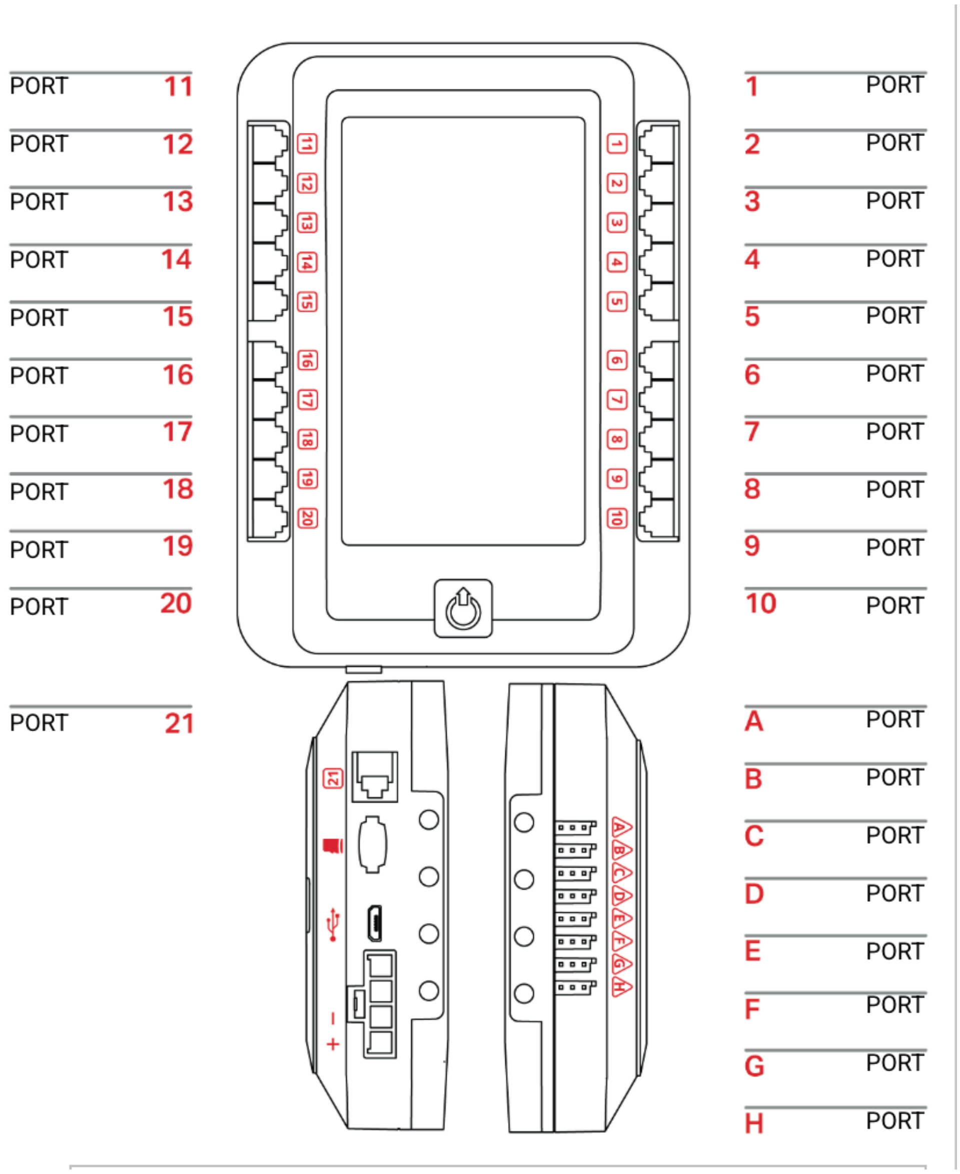


Figure 79: *Brain Ports Map*

This is a map of all the used ports on our brain.

22.9. Operations Control

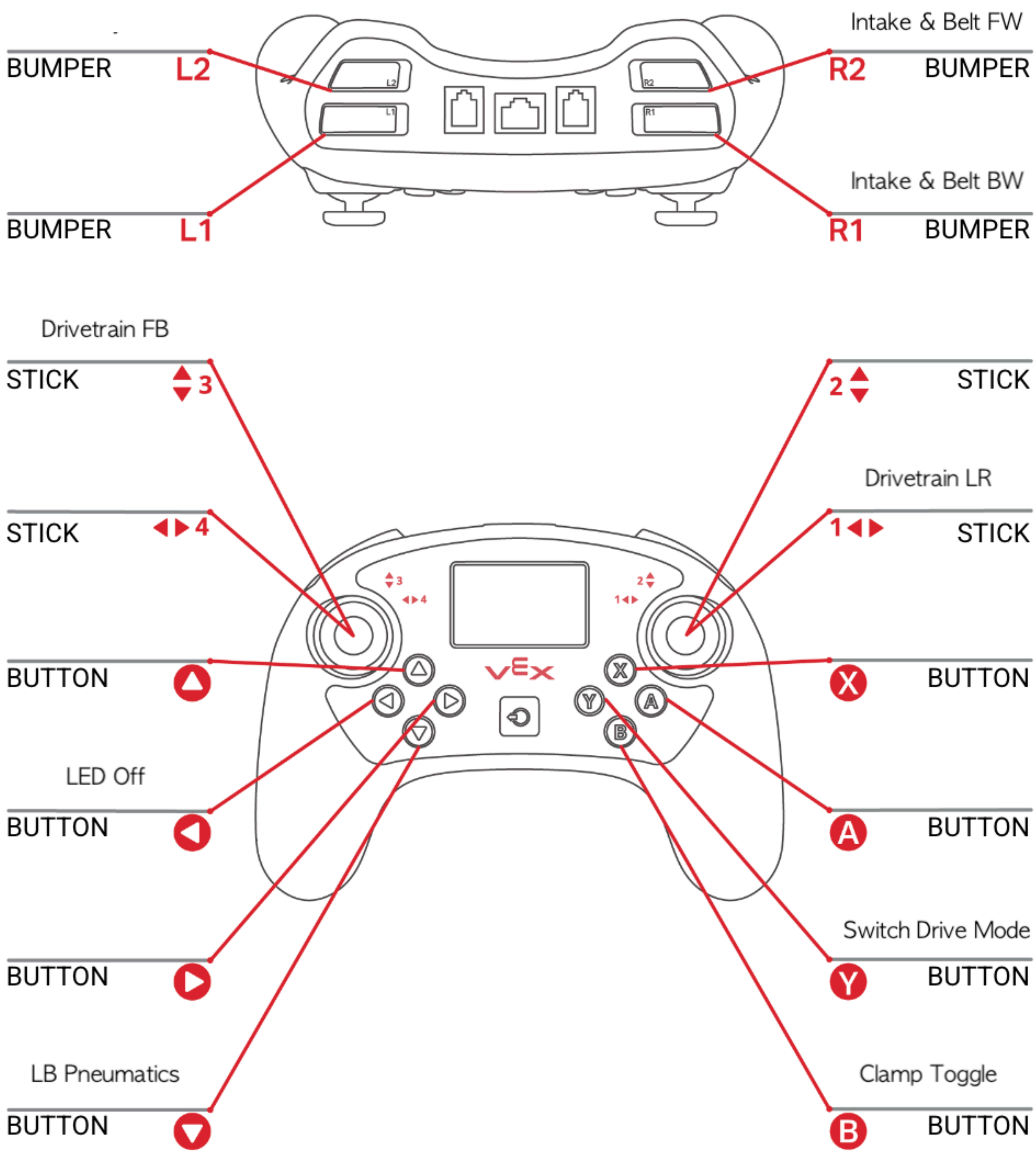


Figure 80: *Final Control Scheme*

Note: all important program files (excluding header files) are listed in Appendix 1

Note: This is the end of team 86832B’s engineering notebook. All following pages are weekly meeting notes, goals, and daily logs recorded throughout the past few months of building. There may also be other miscellaneous documentation. Thank you for your time!

23. APPENDIX 1: Complete Program

This appendix contains all important files used in the 86832B-2025 program, minor files like the LED strip control file are excluded for length sake. More explanation on specific lines of code can be found in the form of comments (lines greyed out and beginning with "/ /") (please keep in mind that "/ /" is also used to excluded unused code).

Main.cpp

```
####################################################################################
This is Team Robust's 2024-5 Program, Written by Patrick Young using the Purdue
Sigbots developed VEX Pros 4.1.0 Library.
(This is my first experience with C++ as well as the PROs library, please excuse any
spaghetti code :P)
####################################################################################


#include "main.h"
#include "control/drive.h"
#include "control/auton.h"
#include "common.h"
#include <iostream>
#include "control/autonskills.h"

void initialize() {
    //reset pneumatics
    robot::mewmatics.retract();
    robot::doinker.retract();
    //set color sensor brightness to max (pre recommendation of Leon)
    robot::ColorSensor.set_led_pwm(255);
    //rumbles controller to signify program start
    robot::master.rumble("-");
    //set LB brake mode
    robot::JiaBrown.set_brake_mode(pros::E_MOTOR_BRAKE_HOLD);
    //calls screen and LED strip initalize functions
    screen::init();
    led::init();
    //clears controller screen for custom display
    robot::master.clear();
    //display border lol
    pros::delay(100);
    robot::master.print(0,0,"---------------");
    pros::delay(50);
    robot::master.print(2,0,"---------------");
}

void disabled() {
}

void competition_initialize() {
    // initialize();
}

void autonomous() {
```

```
    auton::init();

}

void opcontrol() {
    // screen::controllerinit();

    //run driver control program 😭
    drive::run();
}

// Bro fix all bugs in this code like rn
// Sigma

// "240x480"
```

Robot.cpp

```
##################################################################################
This file contains all globally needed values and variable used
Throughout the code, if anything has “robot::” in front of it,
Its most likely used in more than one place in the program
##################################################################################

#include "robot.h"
#include "pros/adi.hpp"
#include "pros/apix.h"
#include <iostream>
namespace robot {
pros::MotorGroup MotorLeft ({-13, -12,-11});
pros::MotorGroup MotorRight ({1,2,3});
pros::Motor Joshua (8);
pros::MotorGroup JiaBrown ({-16,17});
pros::Motor TakeMeToHeaven (-4);
pros::Controller master (pros::E_CONTROLLER_MASTER);
pros::Optical ColorSensor (7);
pros::adi::Pneumatics mewmatics ('E', false, false);
pros::adi::Pneumatics freeren ('G', false, false);
pros::Rotation Rot (21);
pros::adi::Pneumatics doinker ('A',false,false);
int LIMIT_TOP = -10;
int LIMIT_BOTTOMHAHA = 30;
int drivestate = 3;
int color = 0; //3=red 8=blue
int side = 0; //1=pos 2=neg
int auton = 0;
int r1presscount = 1;
int r2presscount = 1;
}
```

Control/Drive.cpp

```
############################################################################
This file is all the driver control functions, including driving, lady brown
pneumatics, LED strip, and additional features like color sorting.
############################################################################

#include "control/drive.h"
#include "liblvgl/lvgl.h"
#include <iostream>
#include "common.h"

namespace drive {

int multiplier = 180;
int multiplierbuteddie = 160;

//Tank Drive mode
void tank() {
    // get controller analog
    float lefty = (float)
robot::master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y) / 127 * multiplier;
    float righty = (float)
robot::master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y) / 127 * multiplier;

    //our old GUI code used to display the values of the analog sticks, I'm keeping
this here in case I want to switch back

    // lv_label_set_text(screen::leftylabel, std::to_string(lefty).c_str());
    // lv_label_set_text(screen::rightylabel, std::to_string(righty).c_str());

    //the numbers are negative cuz this is code ported from our old bot, and in my
code the drivetrain is technically backwards, and i somehow thought it was a good
idea to just run the motors in reverse instead of reversing the motor definition.
thanks past me.
    robot::MotorLeft.move_velocity(-righty);
    robot::MotorRight.move_velocity(-lefty);

}

//Idk what to name this drive mode, eddie calls it arcade drive (change later)
static int lset = 0;
static int rset = 0;
static long long ltime = 0;
constexpr float slew_rate = 700;
void weird() {
        // get controller analog
    float lefty = (float)
robot::master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y) / 127 * multiplier;
    float rightx = (float)
robot::master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_X) / 127 *
multiplierbuteddie;
```

```
    float dt = (pros::micros() - ltime) / 1e6f;
    ltime = pros::micros();
    if (lefty<robot::LIMIT_TOP && robot::LIMIT_BOTTOMHAHA<lefty) {
    robot::MotorLeft.brake(); rset = 0;
    robot::MotorRight.brake(); lset = 0;
    } else if (lefty>robot::LIMIT_TOP or lefty<robot::LIMIT_BOTTOMHAHA){
    float slew_range = slew_rate * dt;

    //this section was admittadly written with help from Leon, essentially, this
section limits the acceleration/deceleration of the motors to a certain number,
preventing the robot from decelerating too rapidly and tipping.
    rset = std::clamp(-lefty-rightx, rset - slew_range, rset + slew_range);
    lset = std::clamp(-lefty + rightx, lset - slew_range, lset + slew_range);
    robot::MotorLeft.move_velocity(lset);
    robot::MotorRight.move_velocity(rset);
    }
}

//Color Sorting
void ColorSorting(){
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_UP)){
        robot::color = not robot::color;
    }
    if (robot::color == 0){
        if(robot::ColorSensor.get_hue() <= 20){
            robot::Joshua.move_velocity(200);
            pros::delay(350);
            robot::Joshua.move_velocity(0);
    } else if (robot::color == 1){
        if (robot::ColorSensor.get_hue() >= 200){
            robot::Joshua.move_velocity(200);
            pros::delay(350);
            robot::Joshua.move_velocity(0);
    }
}
}
}

//control LED Strip
void ledcontrol(){
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_UP)){
        //these color codes caused me so much headache. if anyone is reading this
someone should make a list and send it to me
        led::ledstrip.set_all(0x00FF0000);
    }
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_LEFT)){
        led::ledstrip.clear_all();
    }
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_DOWN)){
        led::ledstrip.set_all(0x000066FF);
    }
}
```

```
//operate clamp
void mewmatics() {
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_B)) {
    robot::mewmatics.toggle();
}
}

//operate knife
void doinker(){
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_DOWN)) {
    robot::doinker.toggle();
}
}

//operate Lady Brown Pneumatics piston
void WOAHWHAT() {
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_X)) {
        robot::freeren.toggle();
}
}

static bool isJiaBrownPositioning = false;
//operate Lady Brown
void JiaBrowning() {
    if (robot::master.get_digital(pros::E_CONTROLLER_DIGITAL_L1)) {
        robot::JiaBrown.move_velocity(130);
        isJiaBrownPositioning = false;
    } else if (robot::master.get_digital(pros::E_CONTROLLER_DIGITAL_L2)) {
         robot::JiaBrown.move_velocity(-130);
         isJiaBrownPositioning = false;
    } else if (!isJiaBrownPositioning) {
        robot::JiaBrown.brake();
    }
}
// Lady Brown Load Position
void JiaBrownPositioning(){
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_RIGHT)){
        isJiaBrownPositioning = true;
    }
    if (isJiaBrownPositioning) {
        float kp = 1.5;
        float error = -300 - robot::JiaBrown.get_position();
        robot::JiaBrown.move_velocity(kp * error);
        printf("Error is %f\n", error);
        printf("Position is %f\n", robot::JiaBrown.get_position());

        if (fabs(error) < 10) {
            isJiaBrownPositioning = false;
            robot::JiaBrown.brake();
        }
    }
}
// Run Belt
```

```
void GoWhiteBoyGo(){
    if (robot::master.get_digital(pros::E_CONTROLLER_DIGITAL_R1)){
        robot::Joshua.move_velocity(-160);
        // robot::master.set_text(2,0,"Belt: fw");
    } else if (robot::master.get_digital(pros::E_CONTROLLER_DIGITAL_R2)) {
        robot::Joshua.move_velocity(160);
        // robot::master.set_text(2,0,"Intake: bw");
    } else {
        robot::Joshua.move_velocity(0);
        // robot::master.set_text(2,0,"Intake: off");
    }

}

// Run Intake
void AHHHHHHHHHHH(){
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_R1)){
        robot::r2presscount = 0;
        if (robot::r1presscount == 0){
            robot::r1presscount += 1;
            robot::TakeMeToHeaven.move_velocity(200);
            // robot::master.set_text(2,0,"Intake: fw");
        } else if (robot::r1presscount == 1){
            robot::r1presscount = 0;
            // robot::master.set_text(2,0,"Intake: off");
            robot::TakeMeToHeaven.move_velocity(0);
        }
    }
    if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_R2)) {
        robot::r1presscount = 0;
        if (robot::r2presscount == 0){
            robot::r2presscount += 1;
            robot::TakeMeToHeaven.move_velocity(-200);
            // robot::master.set_text(2,0,"Intake: bw");
        } else if (robot::r2presscount == 1){
            robot::r2presscount = 0;
            robot::TakeMeToHeaven.move_velocity(0);
            // robot::master.set_text(2,0,"Intake: off");
        }

    }
}

// Main Drive Loop
void run() {
    while (true) {
        // Mode switch
        if (robot::master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_X)){
            robot::drivestate %= 3;
            robot::drivestate += 1;
            robot::master.set_text(0,0,"Drive mode: " +
std::to_string(robot::drivestate));
        }
```

```
        // Run Functions (I dont think this is the best way to do this but shhh)
        AHHHHHHHHHHH();
        mewmatics();
        JiaBrowning();
        JiaBrownPositioning();
        doinker();
        WOAHWHAT();
        ColorSorting();
        ledcontrol();
        GoWhiteBoyGo();

        //Drive Mode Switch State (Learnt this from Google B)))))
        switch(robot::drivestate) {
            case 1:
                tank();
                break;
            case 2:
                weird();
                break;
            default:
                weird();
        }
        pros::delay(10);
    }
}
}
```

Screen.cpp

```
################################################################################
This file defines all items rendered on the screen, written using the LVGL (Light and
Versatile Graphics Library) included with Pros. It handles side selection for
autonomous as well as color sorting
################################################################################

#include "common.h"
#include "control/auton.h"
#include "liblvgl/lvgl.h"
#include <iostream>


LV_IMG_DECLARE(background);
LV_IMG_DECLARE(Newback);
LV_IMG_DECLARE(miku);
namespace screen {

// defining all objects and such

lv_obj_t *img ;
lv_obj_t *heheimg;
lv_obj_t *leftylabel;
lv_obj_t *leftxlabel;
lv_obj_t *rightxlabel;
lv_obj_t *rightylabel;
lv_obj_t *leftylabellabel;
lv_obj_t *leftxlabellabel;
lv_obj_t *rightxlabellabel;
lv_obj_t *rightylabellabel;
lv_obj_t *author;
lv_obj_t *armlocation;

//auton popup

lv_obj_t *popupwindow;
lv_obj_t *redbtn;
lv_obj_t *bluebtn;
lv_obj_t *skipbtn;
lv_obj_t *hint;
lv_obj_t *side;
lv_obj_t *negbtn;
lv_obj_t *posbtn;
lv_obj_t *hehebutton;
bool imgshown = false;

//Auton side determining logic

void autonside(){
    robot::auton = robot::color + robot::side;
    robot::master.clear();
```

```
    if(robot::auton == 4){
        pros::delay(50);
        robot::master.print(1,0,"Red Positive");
        auton::autonRedPos();
    } else if (robot::auton == 5){
            pros::delay(50);
        robot::master.print(1,0,"Red Negative");
        auton::autonRedNeg();
    } else if (robot::auton == 9){
            pros::delay(50);
        robot::master.print(1,0,"Blue Positive");
        auton::autonBluePos();
    } else if (robot::auton == 10){
            pros::delay(50);
        robot::master.print(1,0,"Blue Negative");
        auton::autonBlueNeg();
    } else{
        pros::delay(50);
        robot::master.print(1,0,"Check Selection");
    }
}

// button return functions

void setred(lv_event_t* e) {
    robot::color = 3;
    lv_obj_set_style_border_color(redbtn, lv_palette_lighten(LV_PALETTE_PINK,
1),NULL);
    lv_obj_set_style_border_color(bluebtn,lv_palette_lighten(LV_PALETTE_NONE,
1),NULL);
    autonside();
}
void setblue(lv_event_t* e) {
    robot::color = 8;
    lv_obj_set_style_border_color(bluebtn, lv_palette_lighten(LV_PALETTE_PINK,
1),NULL);
    lv_obj_set_style_border_color(redbtn,lv_palette_lighten(LV_PALETTE_NONE,
1),NULL);
    autonside();
}
void setpos(lv_event_t* e) {
    robot::side = 1;
    lv_obj_set_style_border_color(posbtn, lv_palette_lighten(LV_PALETTE_PINK,
1),NULL);
    lv_obj_set_style_border_color(negbtn,lv_palette_lighten(LV_PALETTE_NONE,
1),NULL);
    autonside();
}
void setneg(lv_event_t* e) {
    robot::side = 2;
    lv_obj_set_style_border_color(negbtn, lv_palette_lighten(LV_PALETTE_PINK,
1),NULL);
    lv_obj_set_style_border_color(posbtn,lv_palette_lighten(LV_PALETTE_NONE,
```

```
1),NULL);
    autonside();
}
void sethehe(lv_event_t* e) {
    if (imgshown == false) {
        lv_obj_clear_flag(heheimg, LV_OBJ_FLAG_HIDDEN);
    } else {
        lv_obj_add_flag(heheimg, LV_OBJ_FLAG_HIDDEN);
    }
    imgshown = !imgshown;
}
//initiate screen
void init(void) {

    // set background
    img = lv_img_create(lv_scr_act());
  lv_img_set_src(img, &Newback);
  lv_obj_align(img, LV_ALIGN_CENTER, 0,0);

    // hidden miku picture :O
    heheimg = lv_img_create(img);
  lv_img_set_src(heheimg, &miku);
  lv_obj_set_pos(heheimg, 300,0);
    lv_obj_add_flag(heheimg, LV_OBJ_FLAG_HIDDEN);

    // define all buttons and labels
    redbtn = lv_btn_create(img);
    bluebtn = lv_btn_create(img);
    negbtn = lv_btn_create(img);
    posbtn = lv_btn_create(img);
    hehebutton = lv_btn_create(img);
    // this label is just recycled from the old GUI, I'm keeping the names the same
in case I want to go back
    leftxlabel = lv_label_create(hehebutton);

    lv_obj_set_style_bg_color(redbtn, lv_palette_lighten(LV_PALETTE_RED, 1),NULL);
    lv_obj_set_size(redbtn, 48,102);
    lv_obj_set_pos(redbtn, 104.5,14.5);
    lv_obj_set_style_bg_opa(redbtn, LV_OPA_TRANSP, NULL);
    lv_obj_set_style_border_width(redbtn, 5, NULL);
    lv_obj_add_event_cb(redbtn, setred, LV_EVENT_CLICKED, NULL);

    lv_obj_set_style_bg_color(bluebtn, lv_palette_lighten(LV_PALETTE_BLUE, 1),NULL);
    lv_obj_set_size(bluebtn, 48,102);
    lv_obj_set_pos(bluebtn, 104.5,125);
    lv_obj_set_style_bg_opa(bluebtn, LV_OPA_TRANSP, NULL);
    lv_obj_set_style_border_width(bluebtn, 5, NULL);
    lv_obj_add_event_cb(bluebtn, setblue, LV_EVENT_CLICKED, NULL);

    lv_obj_set_style_bg_color(posbtn, lv_palette_lighten(LV_PALETTE_YELLOW, 1),NULL);
    lv_obj_set_size(posbtn, 48,102);
    lv_obj_set_pos(posbtn, 185.5,14.5);
    lv_obj_set_style_bg_opa(posbtn, LV_OPA_TRANSP, NULL);
```

```
lv_obj_set_style_border_width(posbtn, 5, NULL);
    lv_obj_add_event_cb(posbtn, setpos, LV_EVENT_CLICKED, NULL);

    lv_obj_set_style_bg_color(negbtn, lv_palette_lighten(LV_PALETTE_GREEN, 1),NULL);
    lv_obj_set_size(negbtn, 48,102);
    lv_obj_set_pos(negbtn, 185.5,125);
    lv_obj_set_style_bg_opa(negbtn, LV_OPA_TRANSP, NULL);
lv_obj_set_style_border_width(negbtn, 5, NULL);
    lv_obj_add_event_cb(negbtn, setneg, LV_EVENT_CLICKED, NULL);

    lv_obj_set_size(hehebutton, 40,40);
    lv_obj_align(hehebutton, LV_ALIGN_BOTTOM_RIGHT, 0, 0);
    lv_obj_set_style_bg_opa(hehebutton, LV_OPA_TRANSP, NULL);
    // lv_obj_set_style_border_width(hehebutton, 5, NULL);
    lv_obj_add_event_cb(hehebutton, sethehe, LV_EVENT_CLICKED, NULL);

    lv_obj_align(leftxlabel, LV_ALIGN_CENTER, 0, 0);
    lv_label_set_text(leftxlabel, "Miku");
}


// this is the old GUI I dont think I'm switching back

// void autonpopupold(){

//      init();
//      popupwindow = lv_obj_create(lv_scr_act());
//      redbtn = lv_btn_create(popupwindow);
//      bluebtn = lv_btn_create(popupwindow);
//      hint = lv_label_create(popupwindow);
//      skipbtn = lv_btn_create(popupwindow);

//      lv_obj_set_size(skipbtn, 50,20);
//      lv_obj_set_pos(skipbtn, 280,140);
//      lv_obj_add_event_cb(skipbtn, skip, LV_EVENT_CLICKED, NULL);
//      lv_obj_set_style_bg_color(skipbtn, lv_palette_lighten(LV_PALETTE_GREEN,
1),NULL);

//      lv_obj_set_size(popupwindow , 400, 200);
//      lv_obj_align(popupwindow, LV_ALIGN_CENTER,0,0);
//      lv_obj_set_style_bg_color(popupwindow, lv_palette_lighten(LV_PALETTE_GREY,
1),NULL);

//      lv_obj_set_size(redbtn , 140, 100);
//      lv_obj_set_style_bg_color(redbtn, lv_palette_lighten(LV_PALETTE_LIME,
1),NULL);
//   lv_obj_set_pos(redbtn, 20, 20);
//      lv_obj_add_event_cb(redbtn, confirm_left, LV_EVENT_CLICKED, NULL);

//      lv_obj_set_size(bluebtn , 140, 100);
//      lv_obj_set_style_bg_color(bluebtn, lv_palette_lighten(LV_PALETTE_RED,
1),NULL);
//   lv_obj_set_pos(bluebtn, 200, 20);
```

```
//     lv_obj_add_event_cb(bluebtn, confirm_right, LV_EVENT_CLICKED, NULL);

//     lv_obj_set_pos(hint, 20, 140);
//     lv_label_set_text(hint, "Please select Auton Side     Skip ->");

//     while (robot::side != 0 && robot::side != 1 && robot::side != 2) {
//         pros::delay(10);
//     }

// }


//Not used either
void renderside(){
    side = lv_obj_create(lv_scr_act());
    lv_obj_align(side,LV_ALIGN_BOTTOM_MID,0,0);
    lv_obj_set_size(side , 400, 8);
    if (robot::side == 0) {
        led::ledstrip.set_all(0x000000ff);
        lv_obj_set_style_bg_color(side, lv_palette_lighten(LV_PALETTE_BLUE,1),NULL);
        } else if (robot::side == 1) {
        led::ledstrip.set_all(0x00ff0000);
        lv_obj_set_style_bg_color(side, lv_palette_lighten(LV_PALETTE_RED, 1),NULL);

    }  else if (robot::side == 2) {
        led::ledstrip.set_all(0x0000ff00);
        lv_obj_set_style_bg_color(side, lv_palette_lighten(LV_PALETTE_GREEN,
1),NULL);

    }
}
void controllerinit() {
    robot::master.set_text(0,0,"Drive Mode: 1");
  robot::master.set_text(1,0,"Josh Reversed: 0");
  robot::master.set_text(2,0,"In Reversed: 0");
}
}
```

24. APPENDIX 2: Bimonthly Notes and Goals

[Emily]

24.1. August 22

Notes - We worked on developing and inquiring about the vex 2024-2025 high stakes game. We started the drive train.
Goals - Individuals do independent research on designs and start to build the drive train.

24.2. August 29

Notes - We kept on working on the drive train.
Goals - Finish drive train

24.3. September 12

Notes - Start to work on robot arm.
Goals - Start and design robo arm.

24.4. September 26

Notes - Fix robot arms problems and attach onto the base.
Goals - finish robo arm and get a functioning mechanism.

24.5. October 10

Notes - Start building the design for a belt and the intake.
Goals - finish belt and intake by APAC

24.6. October 24

Notes - Keep on building the belt and intake.
Goals, attach the component together.

24.7. November 21

Notes - Fixed a lot to get our robot working, and although it might not be 100%, there is a lot of time left in the competition season.

24.8. December 5

Notes - Rebuild, design, and cad robot.
Goal- fix certain functions of the robot for the ISB scrimmage

24.9. December 19

Notes - keep building the already cadded drive train and design the rest of the robot.
Goals - Finish this new robot by ACAMIS.

24.10. January 9

Notes - remake belt system and design a lady brown
Goals - Create a functioning robot by ACAMIS.

24.11. January 23

Notes - remake the entire robot from the drive train and top **Goals** Independent Inquire new robot designs and cad a new robot

24.12. January 30

Notes - Start building cadded robot, and test and reflect as building each part **Goals** - Try to finish the belt, intake, and clamp by next week

24.13. February 6

Notes - Finish belt and intake, as we have encountered certain problems along the way. **Goals** Finish by next week and start on the lady brown

24.14. February 13

Notes - We ran into friction problems, and the angle of the belt was too steep, causing the whole mechanism not to work **Goals** recad, take apart, and rebuild before TIS

24.15. February 20

Notes Time is limited before TIS, but it is important to have a working intake, belt, and clamp. **Goals** Finish before TIS

24.16. February 24

Notes We ran into friction problems with the drive train which was an easy fix, we have to finalize the lady brown and code. **Goals** Finish robot and code before Nationals.

25. APPENDIX 3: Daily logs

September 24th, 2024

`[William]`

For further development of the robot, we(all) have split into two teams- one is in charge of drive train, the other will work on the arm. We have these members:

Name	In Team	Duty
William	DT (Drive Train)	Troubleshooting / Design
Eddie	ARM	Build
Xander	ARM	Design
Aiden	DT	Build
Sophie	DT	Build/Design
Emily	DT	Build
Jason	ARM	Build / Troubleshoot

Patrick is generally developing the programs for all of them. I heard that he found the C++ API for this program to work. I expect he will soon develop his ideas on coding for VEX.

My team today focused on various details in the motor block building. We found multiple minor issues like unstable shafts, shaft collars that are colliding the gears. Although they do not affect the operations of the motor block, we think it should be and must be better. Otherwise, the reason for designing that good-friction drive train sides is a bit unreasonable.

September 26th, 2024

`[William]`

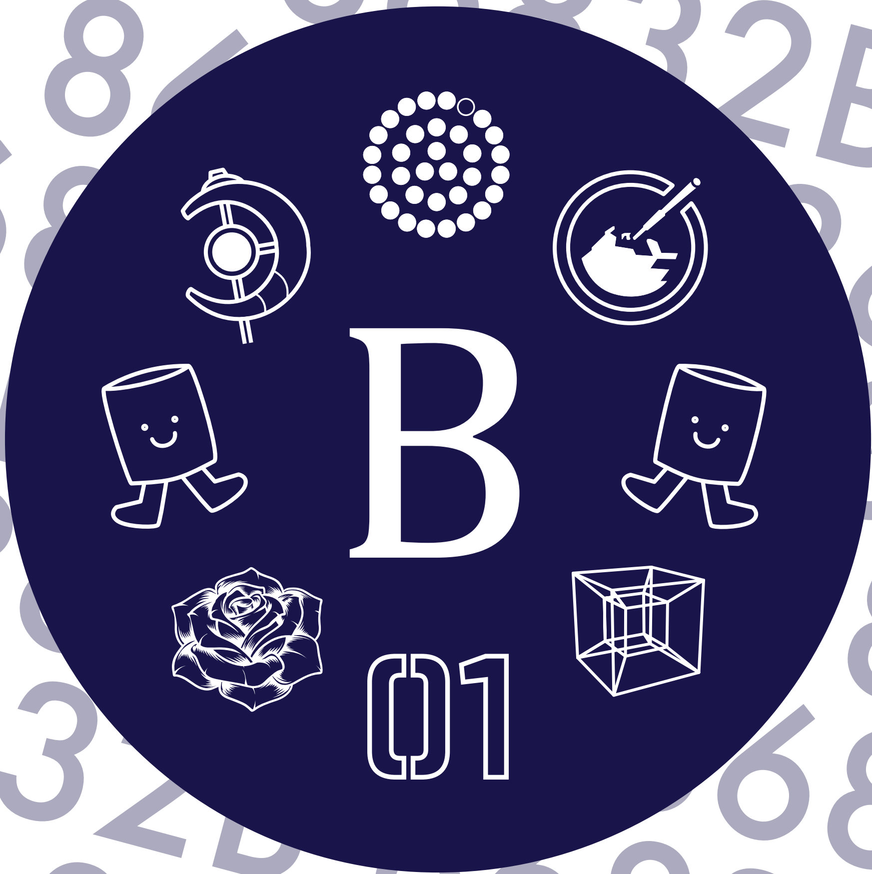
Assembling it wasn't easy. At first, I doubted that the gears would be wobbly. But it turns out that I was overthinking and wasn't trusting my team members enough. Emily, Sophie and Aiden proposed to have an aluminum channel down below to secure the motor block. They proved that my idea was stupid because I suggested that we are doing the channel on the top. Soon, Emily found that this doesn't help because the outer steel C-channel (1*5*1) joints which connect to the drive train sides are still holding everything on the motor block, making the joints burdensome, thus not increasing its robustness. But having the channel installed in the downward position will make another support to the motors, which is much better. We quickly changed our idea and installed it. Although there was a leadership talk happening around (yes, it was quite loud) and we have been distracted a little, it didn't affect our progress.
Next up: drive train test (Must complete or refurbish the code). I am a bit concerned about the 4 blue motors, since I found when running one blue motor it seemed suffered from friction, although we managed to reduce. This should be okay, since when running by only one motor is going to fight the other non-functioning motor's torque.

- I know that Patrick is still figuring out how Vexcode API works, so the code I made for Team D last year can still go up and running, it just needs to be refurbished to fit the new gear system that is slightly different than last year. Kind of ironic that the old team D bot already got destroyed, but I have adapted the code and implemented it to multiple bots, including 24' Team A bot, 24' Team D bot, 24'Spring Fair bot. It feels like...

一码传三代，车走码还在 (One code can serve for more than three generations of bots; Even though the bot got demolished, the code can still work for more bots...)

Robust

ISBeijing



868.32 ฿

Robust