Table of Contents

1. Introduction

Team 5401, the Bensalem High School Fightin' Robotic Owls, is thrilled to be participating in 2018 FIRST PowerUp. Last year's Steamworks season was amazing and successful for our team. We finished 8th place at the end of Qualifications at Seneca District Competition, 3rd at the MAR Regional Championships, and qualified for the 2017 FIRST Championship in St. Louis, where we got to Quarterfinals in the Carson Division. Building off of the success and knowledge gained from last year we were, and still are, excited for this season. We have welcomed new members and into our team and welcomed back many veterans to FIRST. Our more hands on student involvement, led by our expanded Student and Mentor Leadership positions, demonstrates our goal of teaching STEM and business values, inspiring younger generations, and preparing students for life after high school. This year, Team 5401 presents, Ivy, which can place Powercubes on the Scale, Switch and portal, as well as climb with two extra robots onboard. Ivy is named after Ivy Valentine, from the Soulcalibur arcade game which came out in the mid-90s. This is our first year our graduating class of seniors consist of females which is why we decided to focus on female names for the robot. Additionally, since this is our fourth year as a team and 4 in roman numerals, is IV or Ivy.

2. Kickoff and Game Analysis

2.1 Kickoff Weekend

Our kickoff weekend began by sending a small group of about 10 students to our regional Kick-Off location, Hatboro-Horsham High School. The rest of the team reported to Bensalem High School where they watched the kickoff video and immediately began reading the rules and game manual. By the end of Day One, we had determined our overall strategy for playing the game, and students were sent home with homework to read the game manual and ensure they have a full understanding of the year's game. Day Two was spent entirely on robot design, where we discussed potential robot systems. At the end of Day Two, we had concluded what type of robot systems we wanted and a general direction of design.

2.2 Score Analysis

2.2.1 Autonomous Strategy

The maximum number of points that can be achieved during the 15 second Autonomous period, per action is:

- Cross Auto Line (Auto-Run)
  - +5 pt/robot - 15 pts total
- Owning the Switch
  - +2 pt for activation, +2 pt/sec of ownership - 32 pts total
- Owning the Scale
  - +2 pt for activation, +2 pt/sec of ownership - 32 pts total
- All three robots on an alliance complete an Auto-Run and the alliance owns their switch
  - +1 Ranking Point

The theoretical maximum score for an alliance is 79 points and one Ranking Point. This is earned if the alliance owns the switch and scale for the entire 15 second autonomous period and all robots on the alliance complete an Auto-Run.

Conclusion:

The main focus in the Autonomous period should be mastering the Auto-Run and owning the Switch. Being able to own the Switch regardless of your position on the field and being able to cross the baseline is essential. We determined this because this is one of the two times in FIRST Power Up that we can score a Ranking Point. After mastering Auto-Run and owning the Switch regardless of position, we will focus on getting cubes onto the Scale to score the double-valued Autonomous Scale Ownership points.

2.2.2 Teleoperation Strategy

The maximum number of points that can be achieved during the 135 second Autonomous period, per action is:

- Owning the Switch
  - +1 pt for activation, +1 pt/sec of ownership - 136 pts total without PowerUps
  - With Boost Power Up Max Score - 146 pts
- Owning the Switch
  - +1 pt for activation, +1 pt/sec of ownership - 136 pts total without PowerUps
  - With Boost Power Up Max Score - 146 pts
- Power Cube in Vault
  - +5 pts/cube - 45 pts total
- Parked on Platform
  - +5 pts/robot - 15 pts total
- Successful Climb
  - +30 pts/robot - 90 pts total
- All three robots on an alliance successfully climb
  - +1 Ranking Point

The theoretical maximum score for an alliance is 407 points and one Ranking Point. This is earned if the alliance owns the switch and scale for the entire 135 second teleoperation period, nine Power Cubes are placed into the Vault, and all three robots have a successful climb.

Conclusion:

While Power Up is very complex and a set strategy is hard to determine, we believe that the main focuses should be on maintaining ownership of the Scale throughout the match and acquiring all three climbs during the end game. The Scale is an extremely vital game element and will likely be the most contested method of scoring on the field. Maintaining ownership of the scale will ensure that the only way for the opposing team to win would be through ownership of both switches, Power Ups, and or a better end game. As far and end game goes, having all three robots successfully climb is not only essential for the 90 points it gives, but also gives a Ranking Point, which should always be the top priority of the game as Ranking Points are what determines overall rank at the competition.

2.3 Strategic Conclusion

Based off of the maximum scoring values we decided on a practical strategy for our team. Where the highest priorities are: having a successful autonomous that earns us a Ranking Point, in the end game being able to successfully climb, taking a cube and owning it, and putting cubes on the Scale for ownership points. These strategic conclusions will ultimately drive the design of our robot and dictate the list of tasks our robot will need to complete.

1. Drive
2. Hold cubes and intake cubes from floor
3. Start with cube, put cube on Switch
4. Have a successful climb

3. Robot Design and Control

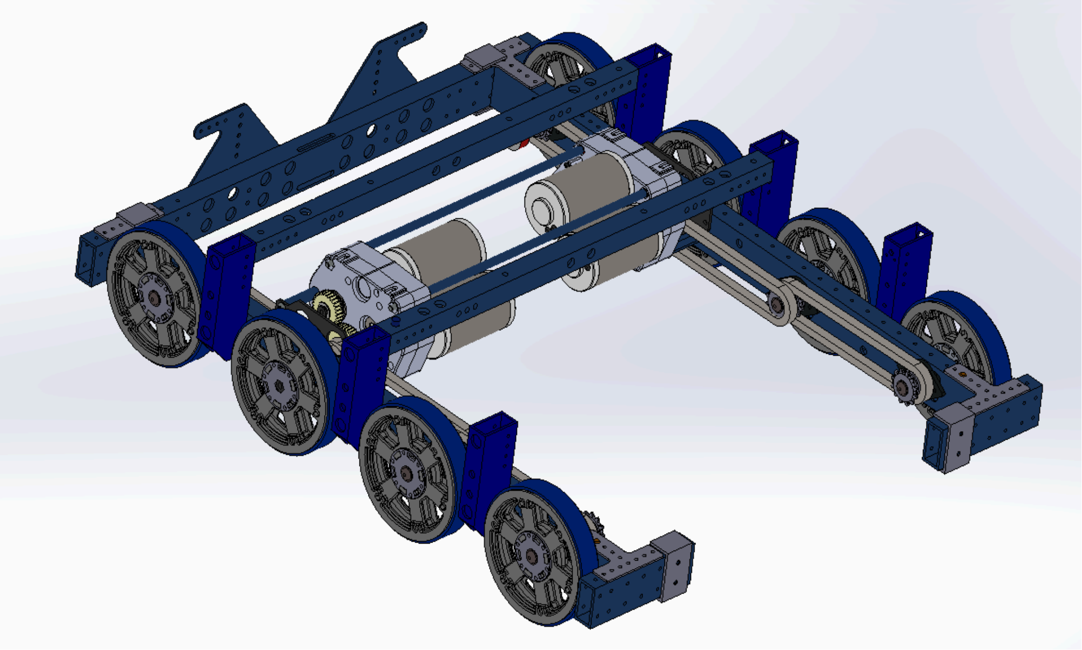
3.1 Drive Train

3.1.1 Initial Design and Prototyping

We started off build season knowing that we wanted to have a West Coast Drive (WCD) system, with either 6 or 10 wheels. After looking at the field and looking at the scale platform, we realized that we needed to use a 8 wheels. 8 wheels worked perfectly and didn't get stuck while driving over the scale ramp.

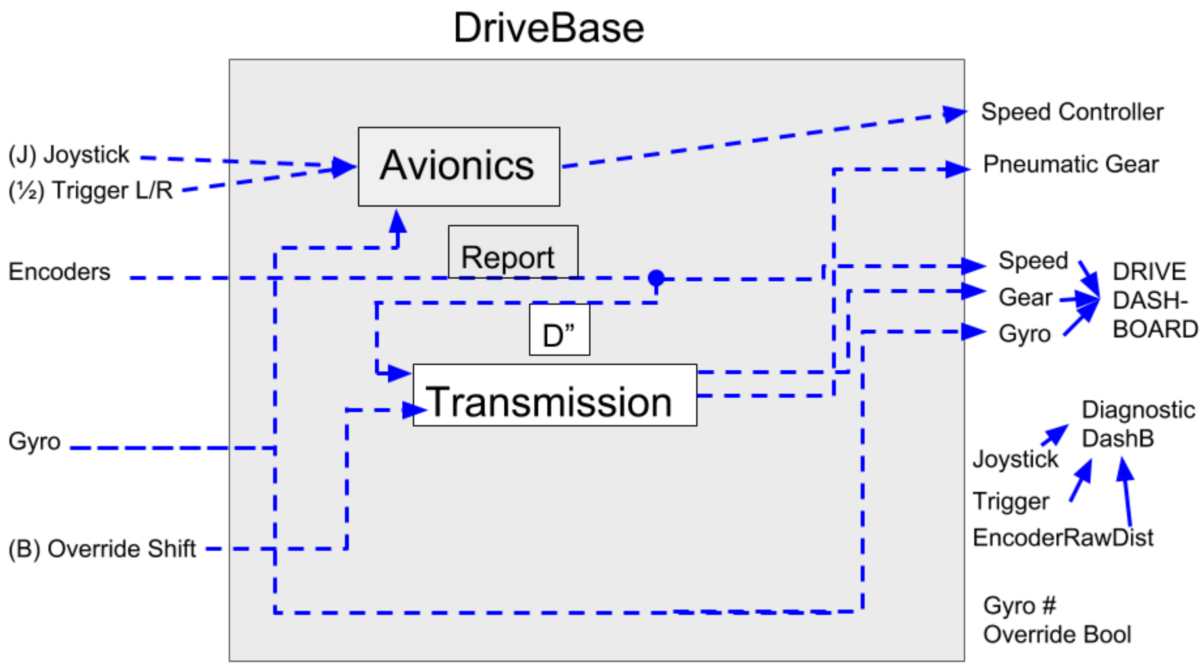
3.1.2 Final Design

Our robot is a West Coast Drive with 8 wheels. The wheels are driven by two 2 CIM motors. The CIMs are Ball Shifting Gearbox (217-3195). The low gear is a 33.33:1 ratio, and yields a speed of 4.50 ft per second. The high gear is a 9.07:1 ratio, and yields a speed of 16.56 per second. All the wheels are 6" VexPro Traction wheels equipped with Blue-Nitrile Roughtop Tread. Our robot's frame perimeter is 32.5 by 27.5 inches. The drivetrain uses #35 chain and 12 tooth sprockets with four 20.25 inch chains and two 20.625 inch chains. The frame rails are .1 thickness 1 x 2 aluminum box tube and riveted together using VexPro T-gussets and a 0.090" lexan plate, with clearance slots for the gearboxes.



3.1.3 Control System

This uses two Victor SP’s as well as a Gyro and Encoders. One notable thing is that we configured the Speed Controllers as PID controllers through the use of FRC libraries allowing us create an Auto-Drive and Auto-Turn function using PID. In teleop, the driver uses Grand Theft Auto style tank controls. The gear transmission is manual.



3.2 Infeeder

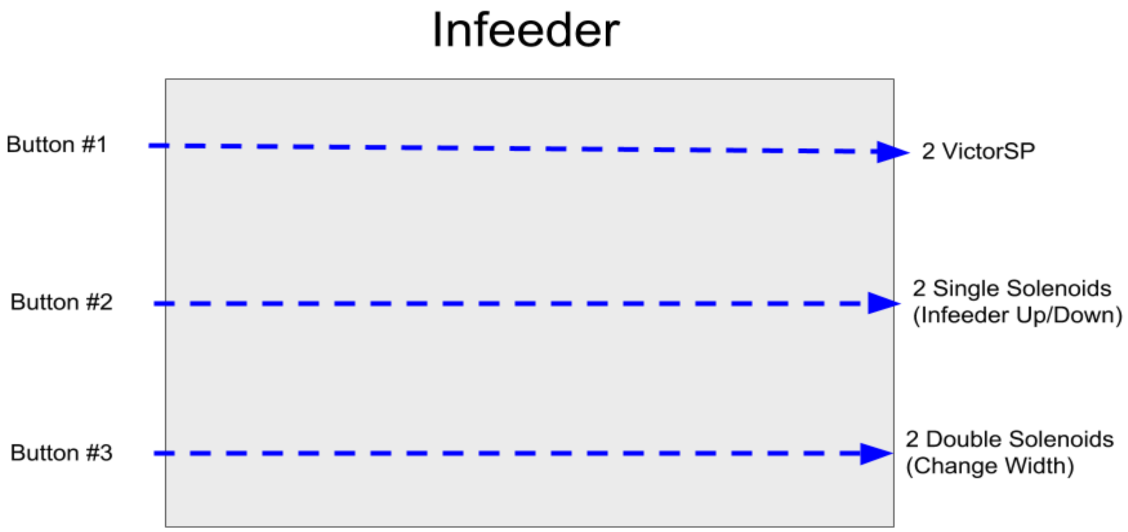
3.2.1 Initial Design and Prototyping

We started prototyping the infeed the first week of build season. We tried to use top and bottom rollers first. We tried to make an infeed like that work but it didn’t work well. Then we decided to prototype a side roller infeed and it was successful. The space in between the wheels were 10.375 inches and 12.375 inches and it worked out great. We used 4 wheels that can squish a little to be able to manipulate the cube how we wanted to. In conclusion, a side roller infeed using 4 squishy wheels is a great idea.

3.2.2 Final Design

3.2.3 Control System

The Infeeder will deploy through pneumatic actuators on a button. The Infeeder activate its motors and take in Power Cubes through a button as well. The infeeder can take in a Power Cube with 11 inch width and a Power Cube with 13 inch width.



3.3 Arm / Wrist

3.3.1 Initial Design and Prototyping

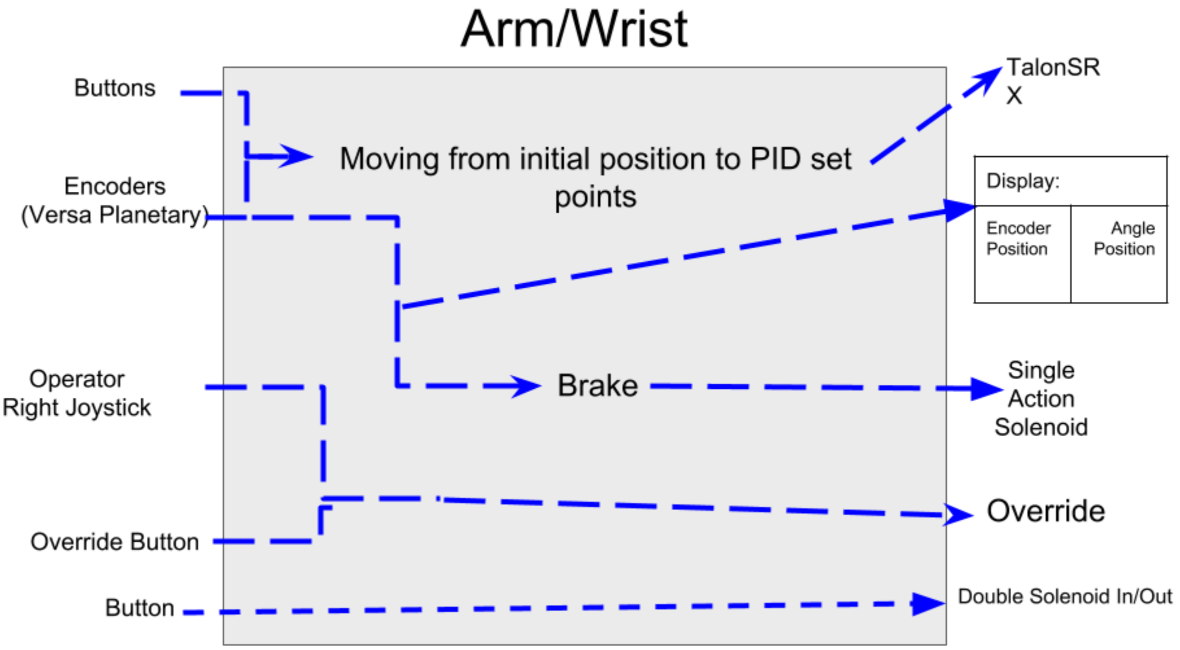
The initial design of the arm was just a small arm that goes up and down on a 3 tier elevator. Then after working with it on CAD, we realized that it wouldn't work and we had to come up with another idea for the arm. The next idea we had was to have a hinge point at the top and another shoulder rotates on that hinge.

3.3.2 Final Design

The arm system is controlled by multiple pneumatics. It can reach above the scale and it can put a cube on top of another cube if the scale is full with cubes. The arm can go to the height of the portal exchange zone and can also get cubes from the pyramid of cubes in the power cube zone. There is a brake in the arm to stop it from moving when it is in the desired position.

3.3.3 Control System

This control system takes advantage of pneumatics and a Talon SRX speed controller. The Talon SRX allow the system to use PID to move to certain heights that are predetermined. A manual override is included as well if the PID fails during a match. The pneumatic actuators are automated and acts like a "wrist". A pneumatic brake is also automated to hold the arm in place when the Talon SRX is not active.



3.4 Climber and Stabilizer

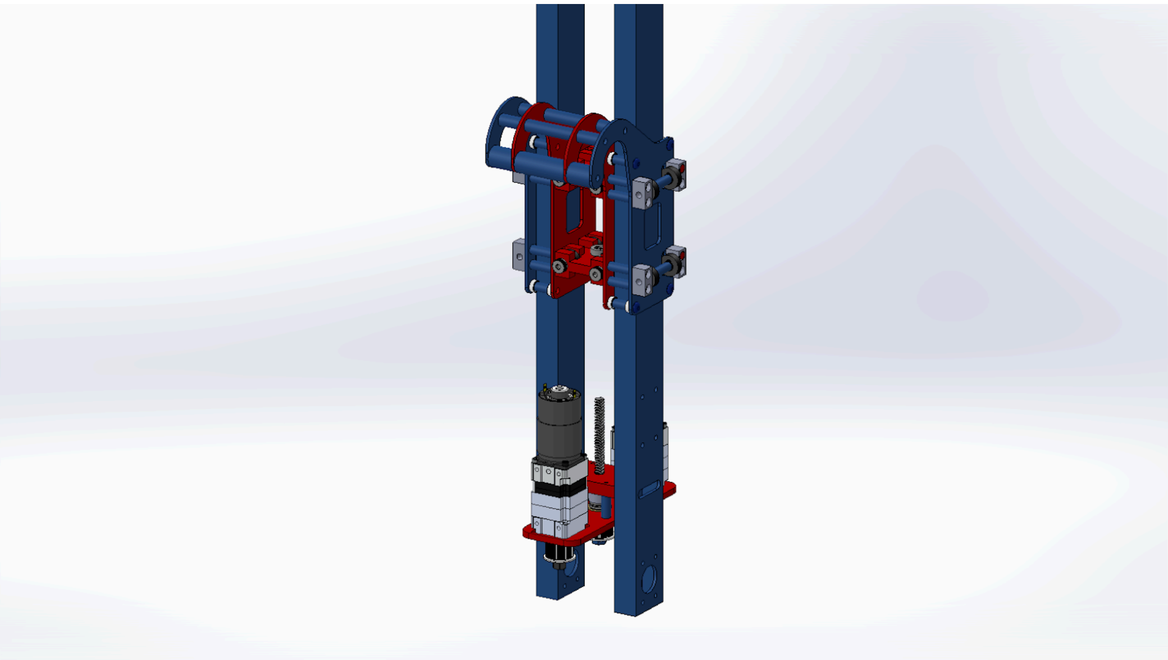
3.4.1 Initial Design and Prototyping

In the beginning of the build season, the climber was just two hooks that went over the rung on the scale and then we would just pull ourselves up. Through many design studies, we found out that a screw driver type climber was the best way to go.

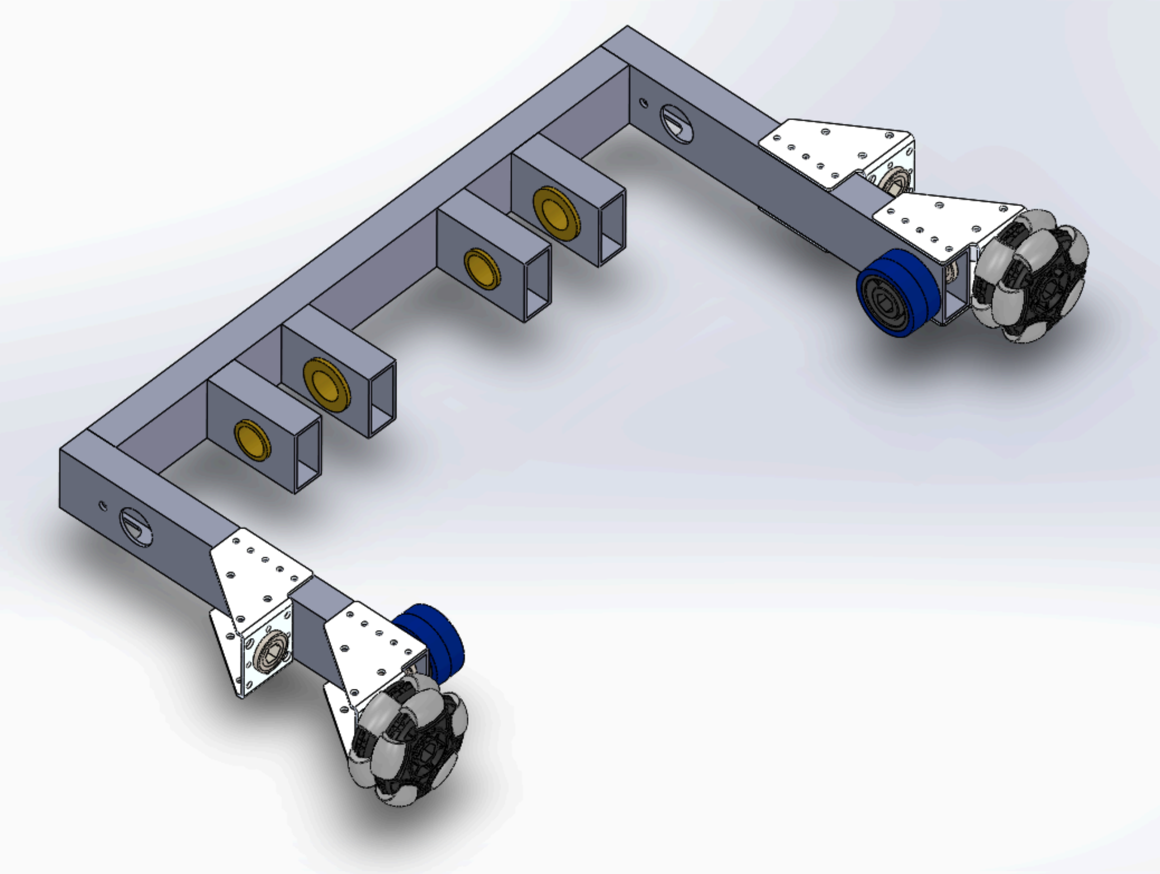
In the beginning, we knew that we needed to have something that made it easier to climb up the side of the scale. We can up with the idea of having a stabilizer that would help keep the robot from tilting as it was climbing up the scale.

3.4.2 Final Design

The climber uses a screw drive to climb. At the end of the match, the arm of the robot goes all the way up and the climber goes as high as it can on the arm. Then the screw drive system starts to turn and the climber descends onto the rung and the screw drive pulls us up above the 12 inch mark and gains us the 30 points.



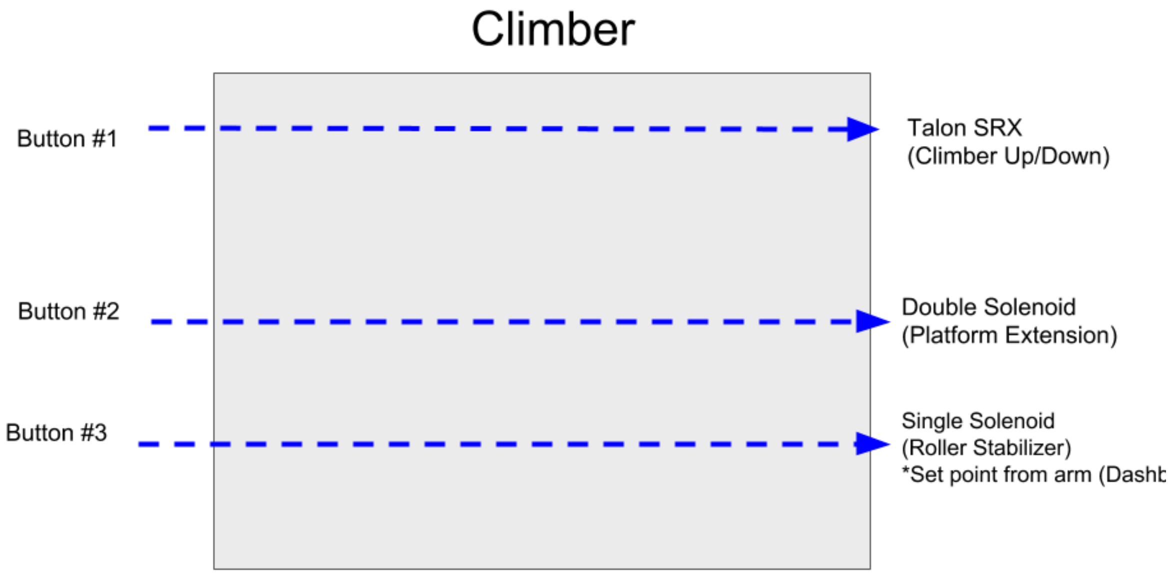
The stabilizer has 2 Omni-Wheels and has 2 smaller wheels in between the two Omni-Wheels. The Omni-Wheel rolls up the side of the little extrusion on the scale and the small wheels on the inside roll up the front of the extrusion of the scale. Two pneumatics keep the stabilizer up and inside the frame perimeter the whole match. Then at the end of the match, the two pneumatics actuate and push the stabilizer down. The robot can smoothly roll up the scale and not tilt while climbing.



3.4.3 Control System

The climber moves uses input from the D-pad to move. Up pulls the robot up and down lowers the robot.

NOTE: The same diagram is used for climber and platforms



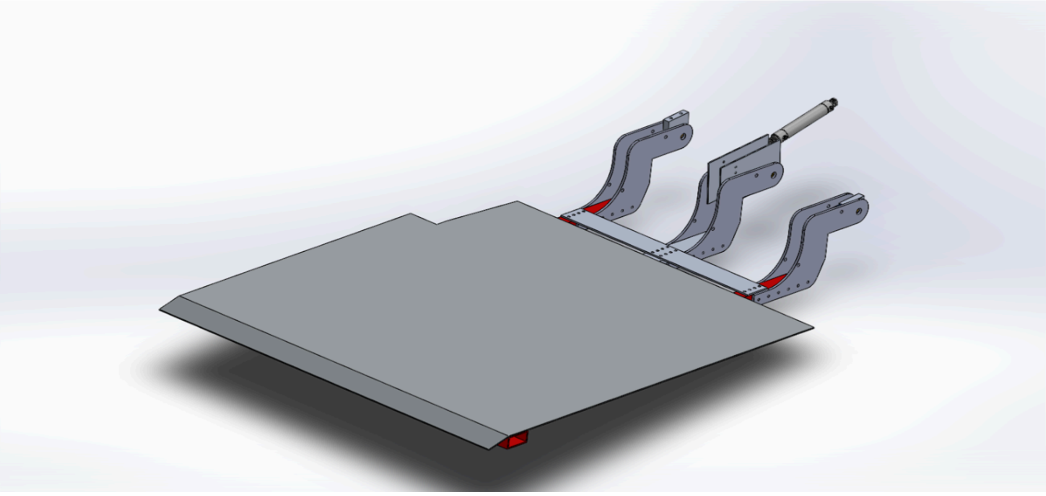
3.5 Endgame Platforms

3.5.1 Initial Design and Prototyping

The thought of having end-game platforms came to the team when we realized just how small of a space the climbing bar was going to be. Since we didn't want to rely on getting the levitation ability, we came up with an idea to have platforms come off our robot during end-game so that when we climb, we can lift a robot on either side of us. The thought behind the end-game mechanism was that it would make us a great choice during alliance selection. The full climb would give us a ranking point each match as well.

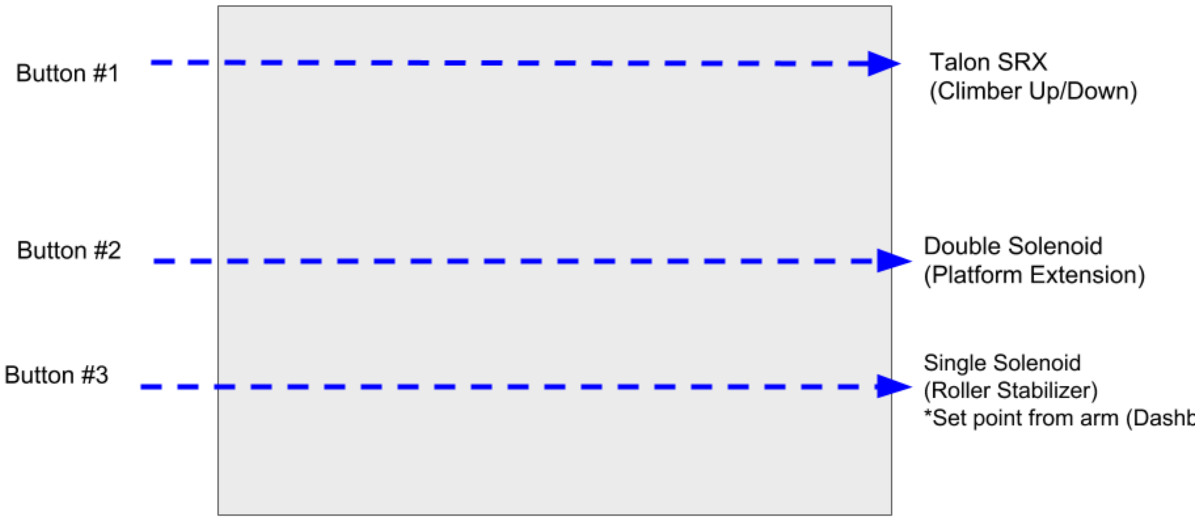
3.5.2 Final Design

The end-game platforms are 30 inch wide plates that rest upon a 2x1 box tube frame and plastic tubings that drops down via Pneumatic actuators. They each have 3 hinge points that connect to the bumper brackets on the drive base. The plates have a slight slope on the far end to let our alliance members drive on. The plates are capable of lifting a bot each so that when we climb we would get a full climb lifting up to 450 pounds.



3.5.3 Control System

This control system allows us to use buttons to manipulate the pneumatic actuators. Give the button a push and down goes our platforms.



Climber

Button #1
Talon SRX
(Climber Up/Down)
Button #2
Double Solenoid
(Platform Extension)
Button #3
Single Solenoid
(Roller Stabilizer)
*Set point from arm (Dashb

4. Software and Controls

4.1 FRC Architecture and Object Oriented Programming

- Like many other teams, team 5401 utilizes Java as the programming language for the robot. Java is most known for its object oriented nature, which allows creations of software objects which mimic physical objects in real life.
- Team 5401 programs their robot through the FRC Architecture. Code is split into Subsystems and Commands. Subsystems mimic the corresponding physical parts on the robot. The Commands execute actions that the physical robot can perform. The “main” block of code runs in a loop and is able to call the Commands through input from a controller. Autonomous code are made in a CommandGroup which combine multiple Commands and require no human input.

4.2 Collaborative Work Environment

The software team for Team 5401 have a weekly meeting to discuss goals for that week. In addition, meetings are organized to discuss how the general logic implementation of each individual Subsystem. Outside of meetings, programmers may work individually or in a team depending on the code or robot part that is being tested.

The following apps are used to communicate and organize the software team:

- Slack: This application is mainly used for communication, such as writing down goals and reminders.
- GitHub: This application is used to store team’s code online as a code repository. As a result, any computer can pull the code from GitHub and edit it. In addition, GitHub offers the ability to “branch” code which creates a copy of the current working code. The existence of the copy allows experimental code to be created and tested without overwriting the current code that works. If the experimental code fails, the branch can be abandoned without modifying the working code. It the experimental code works, the code can be merged into the working code.
- Google Drive is utilized to share other files that are not code. Code documentation and block diagrams are created and kept in Google Slides. Mechanical and software teams share a Google sheet detailing all the PWM, sensor, and pneumatic solenoid channels for the robot. Controller mappings are documented and stored on Google Drive as well.

4.3 Implementing the Programming Process

Identifying the Problem

- In the beginning of the build season, the software team has a meeting with the design. The design team gives the software team a brief explanation of each subsystem, or group of mechanical parts that work together, on the robot and its corresponding movements.

Designing/Drafting the Solution

- The software team begins to draw Block Diagrams of each subsystem. The Block Diagrams is very similar to a typical IPO chart: the Block Diagram displays the possible control inputs, the necessary processing, and the actions of the robot as a result.
- The software team then converts the Block Diagrams into pseudocode. Pseudocode is not "real" code and consists of comments strung together in order to provide an outline for the real robot code. Pseudocode also helps work out logical problems.

Writing the Program

- Branches for each Subsystem are made off the main code. Each programmer is assigned a Subsystem to write by converting pseudocode to actual code.
- During this step, programmers will often talk with members of the design team. The goal is to discuss the specific components and movements of each Subsystem.

Testing and Revising

- After the practice robot is built, the software team tests each Subsystem on the practice, one Subsystem at a time. If the Subsystem work, the branch for that Subsystem is then merged into the main code. Discrepancies between the practice robot and competition robot are noted and changes for the competition code are made as well.

4.4 Teaching New Programmers 18

This year, several rookie programmers joined the team with little to no experience in writing code for the robot. The senior members of the software team were responsible to bring the rookie programmers up to speed. They began with teaching Java basics such as variables, loops, logic statements, arrays and objects.

After the Java basics are mastered, the rookie programmers are taught the FRC Architecture. They begin to experiment with the FRC Architecture on last year's robot.

4.5 Human Machine Interface

The team utilizes two Xbox controllers to run our robot this year. Below are controller maps to be used as references for anyone in need of controlling the robot.

4.6 Scouting App

- "FROScoutingApp" was developed using Android Studio. The app is optimized to run on an Amazon Fire 7 Tablet running Android 5.1 (API 22).
- The app can be used to scout any game because the inputs for scouting are not preprogrammed into the app, but rather created in the app itself. A file containing the layout of inputs can be exported from the app and imported on another device.
- The data from scouting is outputted into a text file. The data is separated by commas and each row is one match.

5. Manufacturing

The STEM program at Bensalem High School received a new technical education facility as part of the renovations that took place in 2017.. Thanks to our new space, all manufacturing of Ivy was done in-house in the Technical Education Shop of Bensalem High School, adjacent to our robotics lab, by our student-led manufacturing team consisting of three students. The frame, electronics mount, shoulders, shoulder crossbars, and bumper mounts are constructed of box tube aluminum, while the arms, custom gussets, blocks, spacers, and shafts are made from a wide range of different sizes of aluminum, polycarb, and Delrin. All shafts and other turned parts that were not purchased complete were turned on one of the two shop lathes in the Technical Education area. All drawings given to manufacturing for production can be seen in the Drawings Section at the appendix of this technical summary. Due to our intricate design, new techniques were learned in order to drill small holes in odd shapes and round corners for clearance on the robot. To maximize production time on the 252+ parts on Ivy, two manual mills and two lathes were run at the same time. Ivy was assembled in the robotics lab, in the team's fully functional pit, which is discussed further below.

In order to support the ambitious strategy the team decided on, and to make use of the internal machining capability, a prototype practice robot was built to learn lessons, provide a tool for drive team to practice on, and a test bed for software while the competition robot was built.

6. Practice Field and Pit

6.1 Practice Field

Using the Technical Education Shop of Bensalem High School, wooden field elements were built to assist prototyping of robot assemblies, as well as give the drive team a practice field on which to hone their skills.

(Insert Photos 191, 185)

6.2 Competition Pit

Assembly of [insert robot name here] took place in Team 5401's fully functional competition pit, set up in the robotics lab at Bensalem High School. The team decided that working in the pit would not only give them the best access to tools and equipment, but would provide lessons about pit layout and flow that would be applied to make the pit more functional at competitions.

(Insert photo of Pit)

7. Conclusion

With our team of 36 members, a broader student leadership team, and new student roles on the team such as Deputy Chief Safety Engineer and Non-Technical Leader we have created a robot that brings the team a sense of pride and achievement. From day one of kickoff weekend, our goal was simple: build a machine that achieves our goals, is expertly engineered to exceed our quality standards, and is a robot we are proud to show off at competitions. Although it was tough build season with many setbacks and challenges, we have achieved all of our goals and we look forward to powering up at competitions.