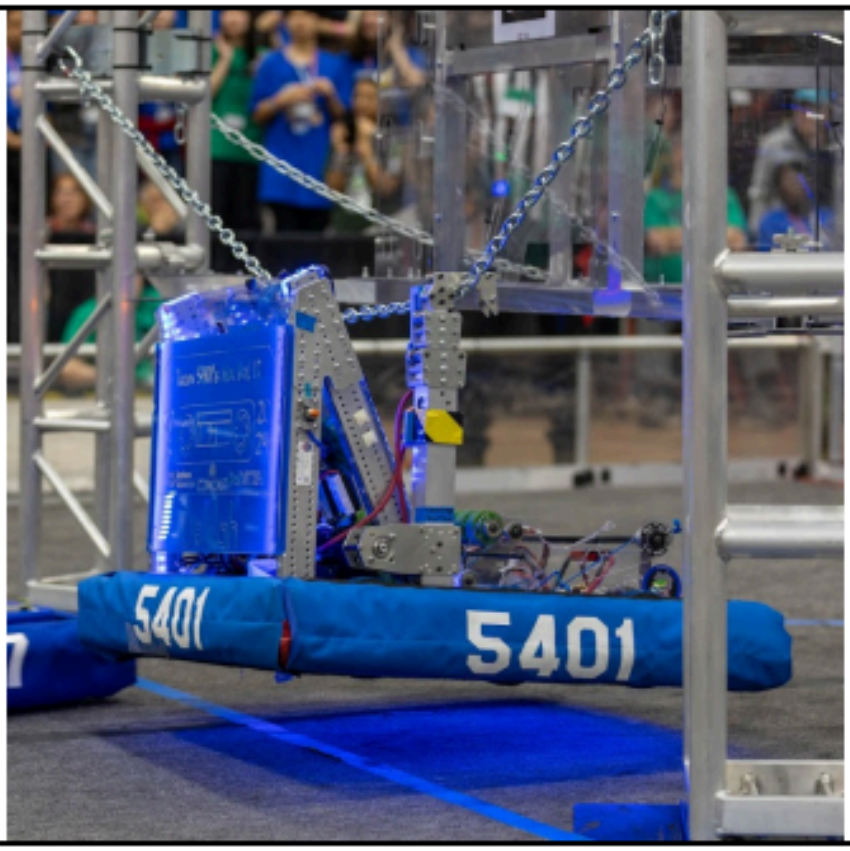


Team 5401

Technical Summary

2023-2024 | Fightin' Robotic Owls | CRESCENDO

Table of Contents

1. Introduction

Team 5401, the Bensalem High School Fightin' Robotic Owls, enthusiastically returns to the 2024 season of the FIRST Robotics Competition, CRESCENDO^{SM}, presented by Haas, after overcoming pandemic-related hurdles. Despite facing an initial membership crisis, our team rallied through an extensive recruitment campaign, welcoming aboard a significant influx of new members, representing 3/5ths of our roster. Through diligent mentorship and training efforts, including the integration of Delilah, named after the iconic song by Plain White T's, we've cultivated a dynamic and motivated rookie class.

This season, our team is infused with a renewed sense of energy and determination, blending the fresh perspectives of our newcomers with the seasoned expertise of our veterans. With unity at our core, we are poised to tackle the challenges of CRESCENDO^{SM} with creativity, teamwork, and innovation. Together, the Fightin' Robotic Owls are ready to spread our wings and soar to new heights, embodying the spirit of FIRST Robotics Competition as we strive for excellence in all our endeavors.

2. Kickoff Weekend

On Saturday, January 6th, Team 5401 gathered in Bensalem High School to watch the CRESCENDO^{SM} game release video. We expect team members to take their own notes on the video and begin brainstorming ideas individually. We immediately dive into the released Game Manual and discuss the rules and possibilities for our robot. Saturday featured a few breakout rooms, where groups of around 10 students began strategy discussions. The main debates this year were ground vs station pickup and speaker vs amp scoring. The meeting is a short 4 hours, but marks the beginning of our busy season. Between Saturday night and Sunday morning, students continued their discussion online and read over the game manual again. By Sunday morning, our strategy for the season was decided.

The next day, we met at the high school from 9-4, with the goal of answering the question How? We broke into groups and discussed every aspect of design one at a time: infeed, shooter, climbing, and more. Our team gathered after every breakout session to review, critique, and expand upon each group's design. Mentors were able to provide constructive criticism to ideas and to play "devil's advocate" to help students clearly deliver their ideas and ensure the viability of these plans. By the end of this weekend, our outline for the season was nearly complete. We immediately started our build season schedule the next day and got to work!

3. Robot Design and Control

3.1 Drive Train

3.1.1 Initial Design and Prototyping

After some deliberation about Swerve Vs. West Coast Drive (WCD), we decided to use WCD as it would not negatively affect our cycle times when compared to swerve because our shooter and infeed would face opposite sides. From our Kickoff discussion, we determined that we did not want to be able to drive over the notes, to avoid the potential risk of them getting caught under the robot. We found that a set of six 6” wheels with a ½” bumper height worked the best to not let pieces under the robot while still allowing us to climb on the lowest point of the chain without the bumpers touching the ground. We decided to keep the Neo motors from last year as they still worked well in this competition and our team was experienced in their electrical components. Additionally, we updated our frame perimeter from last year to 28” x 30”, allowing the opportunity for three robots to climb on one chain, especially if it is spotlit. Everything is held in-place by ⅛” walled 1”x2” aluminum box-tube frame rails riveted with custom made L-Gussets.

3.1.2 Final Design

Our West Coast Drive drivetrain is propelled by 4 Neo Brushless Motors attached to Gearboxes as well as a final frame perimeter of 28” x 30”.

3.1.3 Control System

This uses Spark maxes that are directly connected to the motors as well as the encoders connected to the sparks as well. We are able to run a standard tank drive and use those controls.

3.2 Infeed

3.2.1 Initial Design and Prototyping

During kickoff weekend, and the immediate week following kickoff, we got straight into prototyping infeed mechanisms. Our focus this year was to efficiently intake notes off the ground from over the bumper. To do this, we started to prototype with wood and spare shafts and AndyMark compliant wheels, in sizes varying from 1” - 3”. We used two pieces of wood, with holes cut out for bearings, and tested what space between shafts with wheels, and what size wheels compressed the note the best after picking it up from the ground. We also tested what was the best height off of the ground the infeed needed to be to efficiently infeed the note. We tested the infeed prototype with drills and ½ inch hex shaft adaptors. In the beginning of the prototyping process, we were considering a belt over the bumper infeed, but

made the decision that this year, our team was best suited to efficiently make a wheel based infeed.

3.2.2 Final Design

Our infeed design has been mounted to the robot via two polycarbonate brackets. These brackets then connect to a set of polycarbonate plates, which make up the outer sides of our infeed. A motor is mounted to the left bracket spinning two 68 tooth gears that then rotate the infeed in and out of the robot. This bracket is mounted directly to our frame rails in a placement that keeps our infeed within the 12” maximum extension outside of frame parameter. A secondary motor is mounted to the interior of the left infeed polycarbonate plate that rotates a pulley that runs a 110 tooth belt, rotating the top hex shaft of wheels. This shaft then runs a second pulley that runs a 70 tooth belt to a separate hex shaft with another pulley connected to the 70 tooth belt and a 32 tooth gear. This 32 tooth gear interlocks with another 32 tooth gear to rotate the bottom shaft of compliant wheels. These gears are used to then insure the two sets of compliant and mecanum wheels rotate opposite directions in order to pull a game piece in. The two hex shafts are distanced perfectly apart so that the compliant wheels compress the note by a half an inch in order to pull the game piece in. A curved portion of polycarb pans across the two polycarbonate plates that make up the infeed to hold a limit switch at the desired distance for the note to be pulled in.

3.2.3 Control System

This system uses two NEO-1650 motors, and two spark maxes that are directly connected to the motors, one motor for rotation and one for intake; with encoders also connected to the sparks. The infeed also has a limit switch directly in the center to control and stop the note after it has been pulled into the intake.

3.3 Shooter

3.3.1 Initial Design and Prototyping

The prototyping process for the shooter was fairly straightforward, seeing as we were expecting to have a shooter towards the back of the robot. We were testing the compression needed on the note to produce the furthest distance as well as how many wheels we wanted in the shooter. Originally, we thought that we were going to use flywheels in the shooter, but we determined that green compliant wheels from AndyMark worked the best for it. We used one piece of wood with pulleys and bearings and two NEO-1650’s; there were four compliant wheels that were set up horizontally on the wood with pulleys and belts connecting them. We also added a scrap piece of polycarbonate to the wood to reduce the friction. We then tested them using drills with hex shaft adaptor bits, testing different angles and compressions for shooting in front of the speaker. From that testing, we determined that a 65 degree angle was

the best for our robot for shooting into the speaker. We also tested if we could shoot into the amp with this design. During Kickoff we determined as a team that we wanted to prioritize speaker shooting, so we kept the 65 degree angle even though it did not work well with the amp. While our shooter did not work for the amp, we also tried running our infeed in reverse into the amp.

3.3.2 Final Design

Our shooter is designed around shooting at a calculated 65 degree angle, having four upright 2”x1” tubes angled to lay flush with the shooter itself. We then created custom gussets to mount the shooter to these uprights as well as the front uprights to a crossbar and the rear uprights to the frame rail. There is a set of two flywheels contained within either side of the shooter, each flywheel composed of two 4” green compliant wheels. The lowest wheels are each connected to a motor mounted to the underside of the shooter. These wheels then have a pulley located above the top plate of the shooter which runs a belt to the next two wheels, which then spin at the same time and speed as the first two wheels.

3.3.3 Control System

This system uses two NEO-1650 motors and two Spark Maxes directly connected to the motors, with encoders also connected to the sparks.

3.4 Climber

3.4.1 Initial Design and Prototyping

Initially, coming out of kickoff we wanted to design and build our own climber design, and what we came up with was a chain and pulley climber. We used an old drive base that we had to mount the wooden prototype on. It consisted of two large sprockets and a large length of chain with a hook attached to it. After testing, we determined that this design, while successful, did not mesh well with the other mechanisms on the robot, especially for the intake and the shooter. We made the decision to reference the AndyMark “Climber in a Box” and make modifications to it to better suit our robot.

3.4.2 Final Design

Our climber consists of two 2-stage climbers, one on either side of the robot. The base stage stands at 14.5” tall connected to the crossbar of the drivebase using gussets. The following two stages stand at 13” tall each, each having a custom bracket to keep it within the same axis as the prior tube. Originally each tube was 1” longer, but we decided to cut it down to ensure our robot fully comes off the ground in endgame. These tubes extend using constant force springs, and are kept in a retracted position and retracted using a motor mounted to a spool that winds and unwinds rope, tied to the top of the climber. Our 90 degree hook design

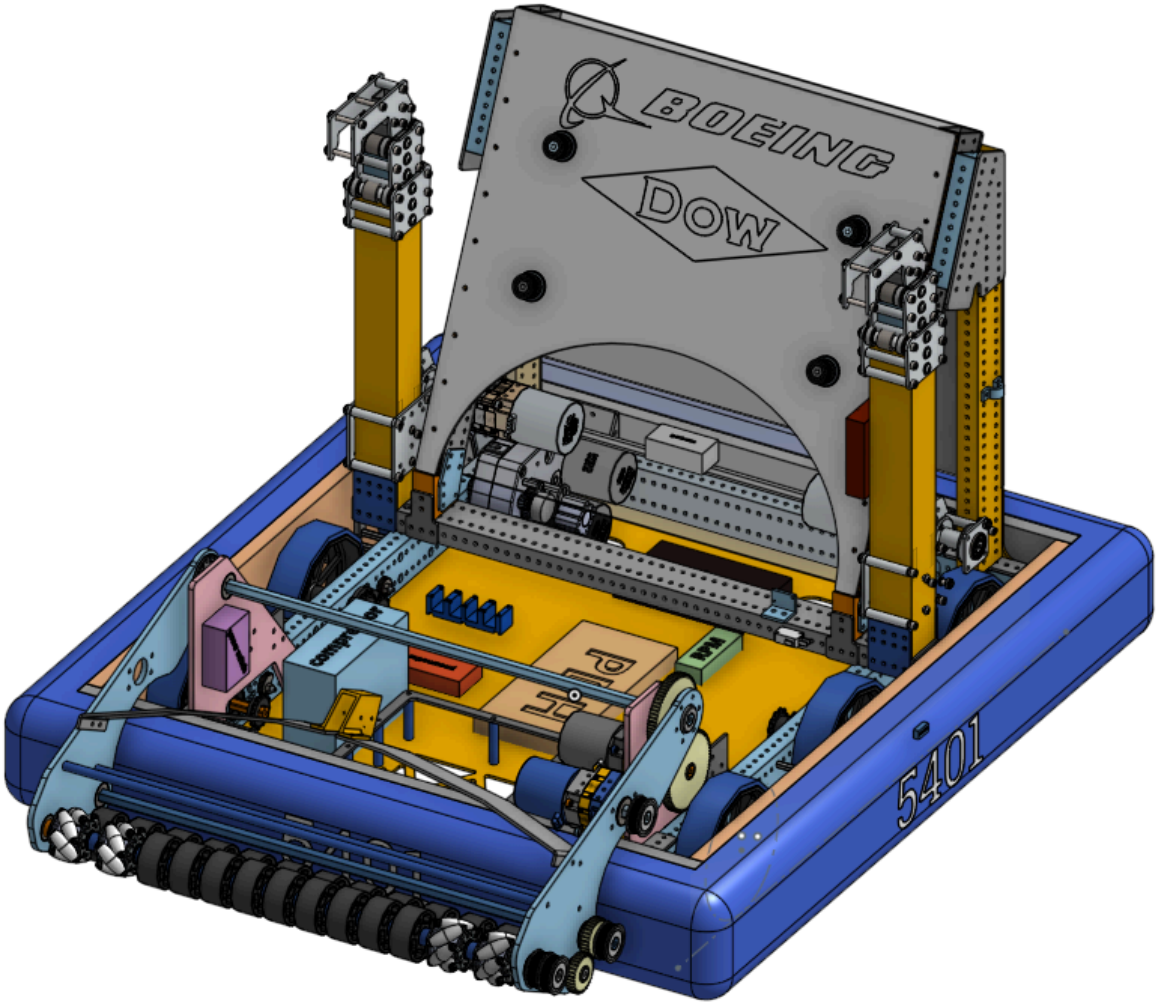
was chosen to maximize our ability to climb the chain, giving us both the least amount of wasted space on the hook as well as pulling us higher off the ground due to the bottom of the hook being closer to the ground. Each climber has a limit switch mounted to the first stage on the inner side of the robot, which is switched once 3-D printed pillars attached to the hook get to the desired height for resting position of the climbers.

3.4.3 Control System

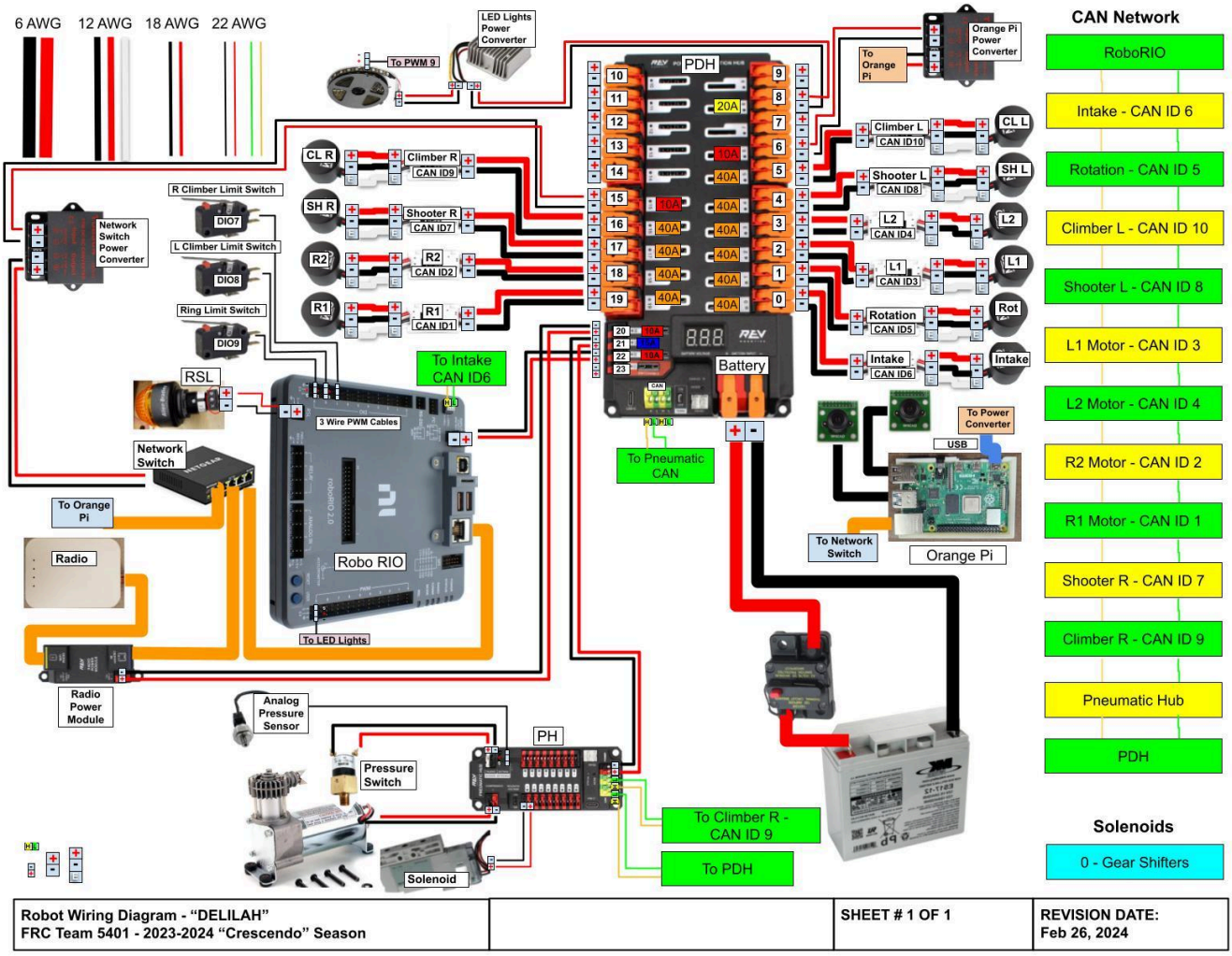
This system has two NEO-1650 motors with Spark Maxes connected directly to the motors along with the encoders. Each arm also has a limit switch attached to the top of the 2'' box tube to stop the climber from overwinding.

3.5 Technical Drawings

3.5.1 CAD Drawing



3.5.2 Wiring Diagram



Robot Wiring Diagram - "DELILAH" **FRC Team 5401 - 2023-2024 "Crescendo" Season**		**SHEET # 1 OF 1**	**REVISION DATE:** **Feb 26, 2024**

4. Software and Controls

4.1 FRC Architecture and Object Oriented Programming

Team 5401, like many other teams, primarily utilizes the Java programming language. Java provides us the ability to match objects found in the real world, with objects directly embedded into our code.
Code is split into Subsystems and Commands. Subsystems mimic the physical parts of the robot, while commands execute actions that the physical robot can perform. The Robot and RobotContainer classes serve to loop the code of our robot utilizing the Scheduler class built into the FRC Architecture. With these loops, we are able to map a button to a given command which can be executed in Tele-Op and the same commands can be used in autonomous periods of matches as well. This command system has allowed us to do on the fly path generation bound to a single button on the driver controller. For example driving right over the station to pick up the note.

4.2 Collaborative Work Environment

The software team for Team 5401 have a weekly status meeting to discuss achievements from the previous week and goals for the upcoming week. We run a modified version of SCRUM with daily standup meetings asking for blockers & questions. Meetings are often organized to discuss the general logic implementation of each individual Subsystem. Outside of meetings, programmers may work individually or in a team depending on the project currently being worked on. We practice paired programming where the ‘rider’ and ‘driver’ alternate throughout the meeting.
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. Slack has the GitHub integration enabled and allows automatic messages to be sent when Pull Requests (PRs) are open and are merged.
- GitHub: This application is used to store the 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 which 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. If the experimental code works, the code can then be merged into working code. This year, we’ve adapted a “Staging” Branch for any new in and in progress testing, keeping our production branch, or main branch, free from error.

- GitHub Projects: A Kanban board and issue tracking system which exists outside of a single repository. This allows for us to have a single board to create “issues”, or to-do items, for tracking across multiple code bases/projects.
- GitHub Actions/Pages: GHA is a cloud-based CI/CD pipeline runner available to all public and private repositories. We use GitHub pages to host our team wiki site which has documentation from subteams compiled into a single space.
- 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

4.3.1 Identifying the Problem & Research

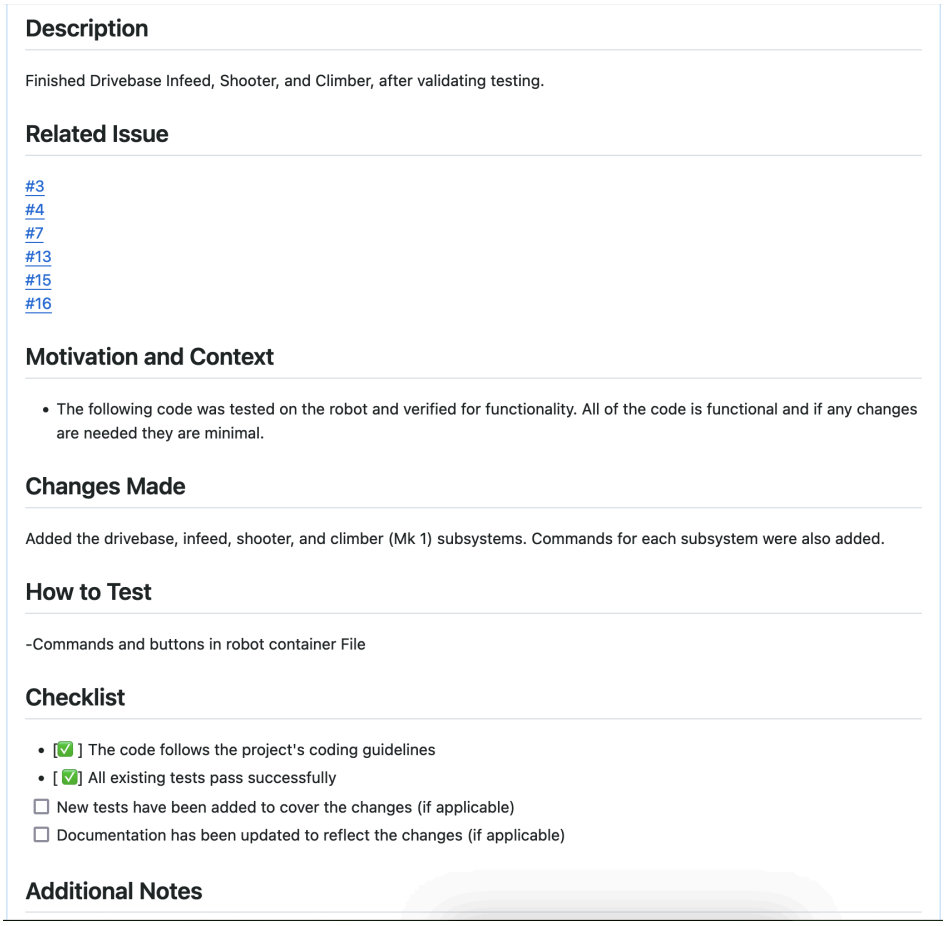
Projects are either self-identified from the programming team or given to us by another sub-team; such as scouting or business. Once a problem is found, research is done to see if another FRC team or product has solved it. If the search is not fruitful, additional research is done into tooling & frameworks needed to solve the problem, and many forms of training are offered to the programmer.

4.3.2 Designing/Drafting the Solution

After research is completed, High-Fidelity mockups are made in Figma for UIs and a class diagram is made for backend logic. The software team then converts the mock ups and class diagrams into GitHub Project issues to assign to programmers. Certain issues are handled individually and others are paired depending on difficulty. Since the programming team sits together in the same room lots of collaboration happens regardless of pairing. Coding branches are made for specific purposes on a case-by-case basis. For example, testing Shuffleboard, Vision, and Autonomous for the first time typically requires its own branch before it can be integrated into the finished product. This also allows us to have multiple programmers work on different aspects of the project at the same time.

4.3.3 Testing and Revising

When an issue is completed a Pull Request(PR) is opened which has a premade template the programmer must fill out like the example below.



Description

Finished Drivebase Infeed, Shooter, and Climber, after validating testing.

Related Issue

#3
#4
#7
#13
#15
#16

Motivation and Context

- The following code was tested on the robot and verified for functionality. All of the code is functional and if any changes are needed they are minimal.

Changes Made

Added the drivebase, infeed, shooter, and climber (Mk 1) subsystems. Commands for each subsystem were also added.

How to Test

-Commands and buttons in robot container File

Checklist

- [✅] The code follows the project's coding guidelines
- [✅] All existing tests pass successfully
- ☐ New tests have been added to cover the changes (if applicable)
- ☐ Documentation has been updated to reflect the changes (if applicable)

Additional Notes

Additionally PRs are initially created to a “staging” branch which is deployed and tested to the robot. After multiple successful test runs of the sub-system/feature, another PR is opened from the “staging” branch to “master”.

4.4 Teaching New Programmers

Every year, several rookie programmers join the team with little to no experience in writing code for the robot. This year, our subteam increased by 250%, going from 4 members to 14! The Programming Lead assumed the responsibility of teaching any and all incoming members with help of the programming mentor where needed.
Unlike last year, a lesson plan was developed for the purpose of teaching incoming (and returning) members. The lesson plans varied from a wide-array of disciplines, with rookies becoming well versed in basic Java, FRC Control Systems, and fundamental software (like VSCode & Github) before being sorted into the “clique” of their choosing. Programmers were able to choose between learning more advanced Java with Robot Code or exciting Mobile App Architecture with Android. This year we have also looked into using game Engines like Godot and Unreal to make real time robot sims for our cad team.

4.5 Human Machine Interface

The Team Utilizes two Xbox controllers for ease of learning, this choice was made for ease of learning, as many people have used an Xbox controller or similar. This year we have decided to automate as much as possible, even for the Tele-Op Period. Buttons are bound to Auto Commands for not only the Operator but Driver too.

The Operator has controls for the Infeed, Shooter, and Climber. The Operator has set points for moving the Infeed, and set Velocity for the shooter so the Operator does not need to play it by ear. The Operator has nearly full manual control over both arms of the climber, only losing control once the limit switch for the climber is triggered, causing the climber to stop. The Driver has a few commands, the main one being the drive controls. However, the driver can also switch gears and change the lights. However, the new control for the Driver is the ability to generate a path right to the station from anywhere on the field, which reduces the room for human error inside of the robot. The Driver also has the ability to align with the speaker for quick and near error free shooting.

4.6 Scouting App

The “FROScoutingApp” was developed using Kotlin. The app is optimized to run on an Amazon Fire 7 Tablet running Android 5.1 (API 22). This application can be used to scout any game of any year, as it is entirely customizable, allowing for users to set up and create the layout and scoring guidelines they find to be the most crucial that year. A file containing the layout of inputs can be exported from the app and imported on other devices. The data from scouting is outputted into a json file using a custom naming convention. There is then a desktop scouting app that is able to pull the JSON files from the tablets and synchronize the data in excel, which is then exported to tableau for even better analysis from our scouting team.

4.7 Vision Processing and Camera Streaming

The vision subsystem in programming has gained multiple significant changes this year. We have incorporated the use of programs such as PhotonVision to help working on AprilTag recognition this year. This will allow the robot to have a faster and more precise reaction to the various changes on the field that may occur during this dynamic competition. We have also played with the idea of using the camera for pose estimation to allow for even better on the fly path generation. We implemented this new vision system by using an 8gb Orange Pi 5v, and an ArduCam for our camera.

4.8 Autonomous Features

Our Autonomous has taken a major leap from last year, with even further improvements in the works. In the past we have used a very crude and rudimentary system for driving, simply checking the encoder value to see if we are there yet. This year as a team we have decided to step up not only our 15 second Auto period but the entire match. This includes a set of new sensors, including limit switches and even some A.I for vision processing.

The primary digital sensor we use on the robot is limit switches. We use this for automatically stopping our infeed once a note is detected, and stopping the climber once it has reached full contraction. This allows the Operator to handle less and less input, allowing them to focus on the next action before the current one is done.

The Operator also has some Analog sensors. These include the integrated encoder inside the Rev NEO motors that we use. We use these integrated sensors for both the Infeed and Shooter, in quite different ways. We use a PID Controller for the infeed for setting its position, allowing the Operator to push a button and watch as the Infeed pops into position. We use a Similar control Scheme for the Shooter, allowing it to spin up to speed, but we have another feature included. The Infeed will only expel into the Shooter once the Infeed is in position, but also the Shooter needs to be up to speed.

The Driver has some new and exciting Autonomous features. The first of which is Auto-Alignment to an April Tag. This allows the driver to push and hold a button, and watch the robot align itself according to a provided distance and angle. Secondly, our “On-The-Fly” path generation allows the Driver to hit a button, and watch the robot drive right over to the station. These two capables allow the robot to nearly drive itself to and back from the station.

Our primary goal for Delilah this year was to play not only to the Drivers/Operator Strength but the Robot itself. Robots are good at aligning, with a very small margin of error, but they are not capable of planning ahead, and this is where the human comes in. While yes, in theory the robot is capable of operating completely on its own, it is enhanced by our human minds operating the robot. This theory allows for a perfect combination between human and robot-operated control.

5. Manufacturing

The STEM program at Bensalem High School, significantly enhanced by a newly renovated technical education facility in 2017, has seen all manufacturing of Delilah performed in-house by our student-led manufacturing team. This year, alongside our broader manufacturing activities, a few members received training on our CNC router. With only three team members proficient in its use, the CNC router has nevertheless played a key role in our production process. It has been essential for creating specific complex components of Delilah, including the polycarbonate plates for the infeed, all of our gussets and brackets, the shooter plates, the acrylic backplate, and the bellypan. Each of these parts, critical to the robot's functionality, showcases the precision and versatility of the CNC router, despite its limited operator base.

In parallel with the CNC router work, our team has engaged in learning and using other equipment. Two rookies were trained on a modern milling machine with digital measurement and calculation abilities, and we also utilized a set of two lathes. An older model milling machine was revived, offering an experience in traditional machining techniques, emphasizing manual skills and old school methods. This diverse approach to manufacturing, including the selective but impactful use of the CNC router, reflects our team's commitment to developing a wide range of skills and producing high-quality components for Delilah.

To complement these efforts, we've integrated multiple 3D printers into our production pipeline, including a Makerbot Replicator+, an Ender 3D Pro, and a Markforged Onyx Printer, enabling the creation of complex parts from strong, lightweight materials. This blend of 3D printing and precise, though selectively used, CNC routing has equipped our manufacturing sub-team to meet any challenge, ready to manufacture any part needed with innovation and meticulous attention to detail.

6. Practice Field

After Kick-Off Weekend, we started building our field with majority-recycled wooden and cardboard elements to assist prototyping, as well as give the drive team a practice field on which to hone their skills. The assembly of the practice field this year was an incredible opportunity to get as many of our rookie students involved as possible and give them hands-on experience. In the weeks before Kickoff, we had an “Intro to Build Season” day for our new members. They learned basic woodshop and an overview of prototyping and its importance. This helped introduce them to basic engineering concepts to prepare them for going into the season and allow them to help build the field and prototype.

This year, in efforts to expedite the process of field building, we created 3 sections. The first section was tasked with printing out all of the blueprints and organizing them by subassemblies. They then were able to calculate how much of every piece needed so they could send our mentor a bill of any needed materials to buy so we could start construction. The next section was tasked with cutting all the 2x4 pieces of wood to length, all the basic shapes out of ½” Plywood. The last section took all big pieces with special features and CAMed them and ran them on the CNC. These pieces include for example the stage main plates. These three groups working together allowed us to complete our field construction in record time to allow for fast prototyping and drive practice earlier.

7. Conclusion

With our team of approximately 65 members and an expanded student leadership group, Team 5401 takes great pride in the creation of our robot, which embodies a sense of accomplishment for the entire team. From the outset of kickoff weekend, our objectives were clear: develop a robot that aligns with our strategic objectives, engineer it to surpass our quality benchmarks, and compete with it proudly. Through effective course adjustments, Delilah was assembled efficiently, providing each team member with invaluable STEM education and training opportunities. This season has ignited our team with motivation and technological advancements that promise to propel us forward, particularly with our now seasoned group of rookie members. As our achievements crescendo, we trust that Delilah's flair and performance will showcase the growth our team has undergone, leaving you equally impressed with our journey.