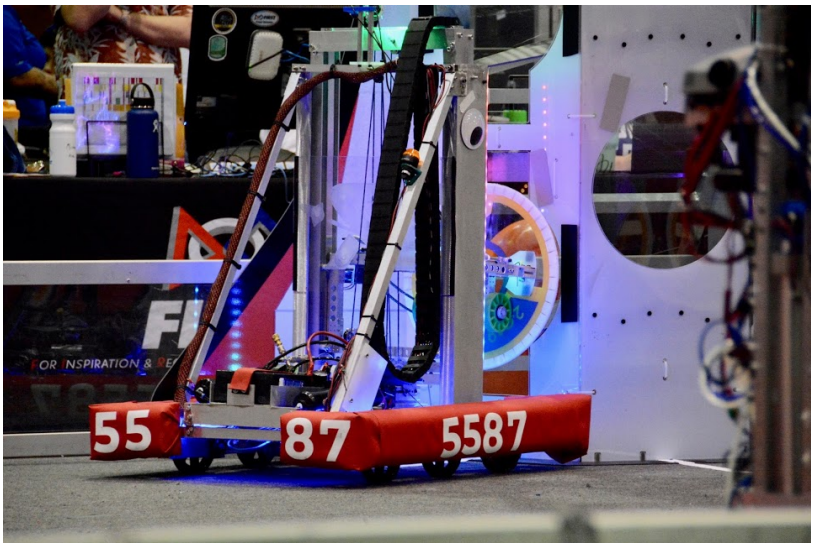
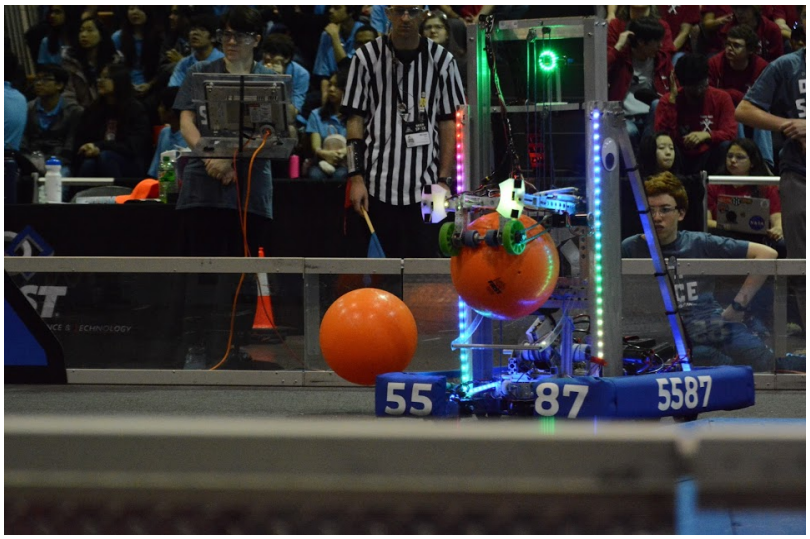
TITAN ROBOTICS FRC 5587
2019 ENGINEERING NOTEBOOK





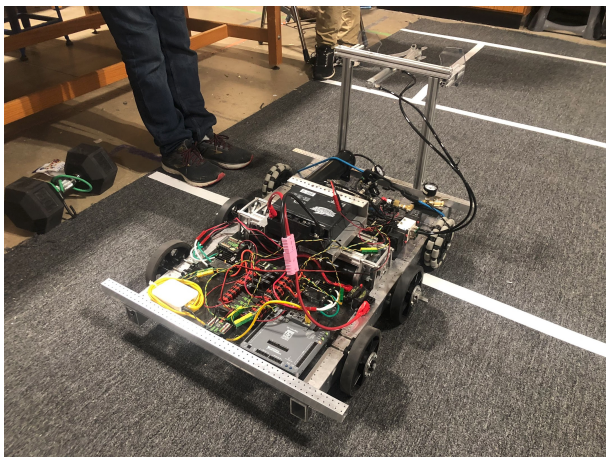
GUILLOUTINA

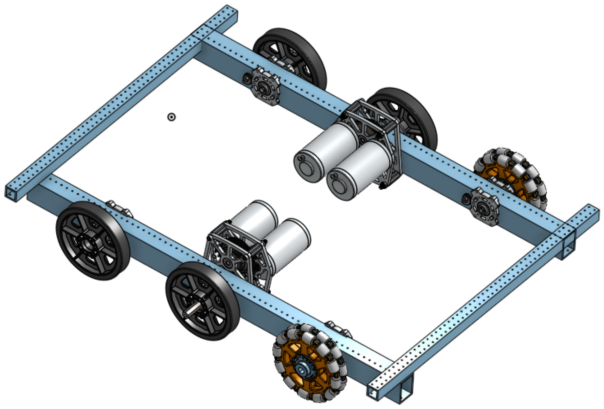
Priorities:

- HAB Level 3 due to 12 point score
- 15 second cycle times
- Focus on hatches as it is needed for cargo scoring

Drivetrain:

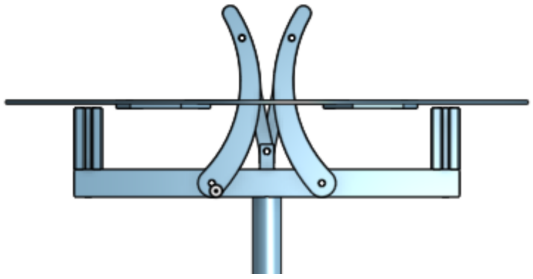
- 6 Wheel drive with front omnis
- Internally routed chain to protect it in impacts

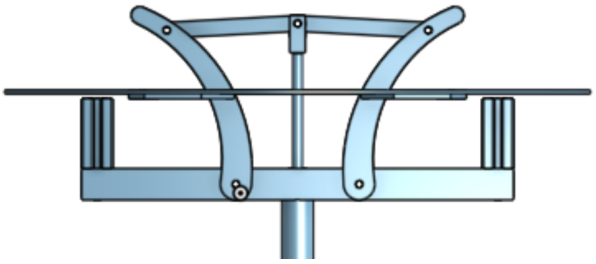




Hatch Intake:

- Two fingers that provide outward pressure and pull in the hatch.
- Single piston actuation as low as 5psi
- Good pickup angles due to long finger
- 2 Limit switches for detecting hatch



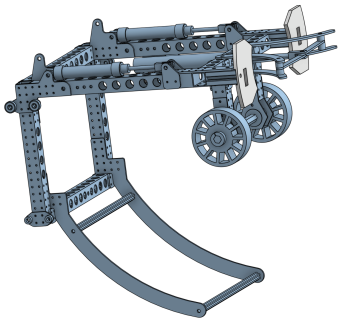


Programming:

- Camera with vision code running on a Raspberry Pi for alignment
- Camera running on roboRIO for driver control

Ball Intake:

- Built during unbag time
- Vertical rollers for ball pickup
- Mounted below hatch intake



Elevator:

- C-Channel cascading elevator.
- 2 775Pros for lifting in under 2 seconds.
- Bottom limit switch to set zero for accurate setpoints.
- Current limiting to prevent burning out motors.

Level 2 Lift:

- Pistons push up back of robot to complete level 2 climb
- Angle iron protects pistons from impact
-

Electronics:

- Hinged battery mount and pneumatics board
- Elevator and Drivetrain LEDs
- Connection to Arduino over I2C and Raspberry Pi over ethernet (TCP).

Day 1 - 1/5

- Climbing to HAB Level 3 identified as top priority
    - 12 pts is a lot in a game maxing out at 130ish pts (equal to 6 hatches or 4 cargo)
    - Can almost guarantee ranking point each match as another robot only has to drive onto HAB Level 1
- Fast, small, agile bot necessary to be efficient cycler due to small spaces for maneuvering and number of game pieces to be moved
    - Try to get to 15 sec hatch cycle time (would allow 6-7 game pieces to be scored + 1 in sandstorm)
- We wouldn't be able to get Rocket RP by ourselves
    - Need to be able to contribute to RP
- Hatches vs. Cargo
    - We decided to focus on hatches as they are needed to score any points from cargo
        - Currently going to pick them up from player station
    - Low level cargo functionality could be added later in season if time
- Low level vs. All heights
    - This was the main point of contention today
    - Low level hatches & cargo only is extremely simple, allowing us to spend adequate time on figuring out climbing, which will be difficult
        - Could execute extremely well, but at higher levels of play, might not be as valuable
    - All heights of hatches would require an additional mechanism, but would give us more value
        - Robot would likely be bigger, potentially making climbing more difficult
    - **Decision:** Once intake design is chosen, make static hatch mechanism, mount on old drivetrain, and give to programmers and drivers. This way, we have a failsafe as we'll have a robot capable of playing the game very early on. During this time, the elevator and drivetrain/climber will be developed in parallel. On 2/1, we'll look at the status of the elevator and static hatch mechanism and mount whichever is better on the drivetrain/climber.
- Sandstorm
    - We're good at simple, consistent autos, so we'll start sandstorm with a a simple auto such as driving off HAB level 2 and delivering a hatch. Then we'll switch to using cameras to try to score another game piece or move into position to be ready to cycle as soon as teleop begins.
- Initial Design Concepts
    - Drivetrain - probably single speed, rectangular, tank drive, at least 6 in. wheels, possibly pneumatic wheels, likely 4 wheels per side to make climbing easier
    - Climber - something to push drivetrain up, roller arms to pull further on platform, 8 wheel drivetrain, this mechanism is the most unsure design
    - Lifting - elevator or 4 bar

- Intake - pneumatic hook, velcro, magnets

Week 1

1/7

- Elevator
  - Prototyped 4 bar lift and cascading elevator
  - Thought about design and viability of each option, taking into consideration the fact that speed and climbing are our main focus
  - Decided on 3 or 4 stage cascading elevator with rope and pulley system for upper stages, and driven with chain and sprocket. Final design will feature nested stages to increase strength of elevator and maintain a solid center of gravity
- Drivetrain/Lift
  - Considered angled metal and two lifts on either side like feet.
  - Decided on single scissor lift where front arm slides back to push robot forward and falls out of slot so that the robot leans forward onto the platform.
  - Basic CAD to determine geometries.q
- Hatch Mechanism
  - Prototyped the finger clamp mechanism and the velcro with push away pistons.
  - Noticed that the push away pistons were ineffective, because the hatch would continue to be stuck to the mechanism, despite the pistons being actuated, and would then hinge on the remaining contact points, finger clamp mechanism would be harder to control, as the driver would have to position the finger inside of the hatch and move the elevator down to remove it.
  - Decided to continue prototyping into tuesday, testing a design in which the velcro pulls away from the hatch, as opposed to pushing the hatch away from the velcro.

1/8

<u>Goals:</u>

- Prototype 3rd design for the egg slicer - in progress
- Have Kernflake operational - complete
- Determine geometry and materials for elevator - in progress
- Finalize drivetrain - complete

- Egg Slicer
  - Prototyped velcro pull away mechanism and outward pressure mechanism
  - Noticed that Velcro continued to hinge on the other piece of velcro, suggesting that a one-bar design would be beneficial, or at least a mechanism with support on both sides of the Velcro, Outward pressure design must be modified so that the backboard is off of the hinge plane, to provide pressure on the hatch when the mechanism is applied.
  - Decided to continue on to Wednesday, prototype one Velcro bar mechanism with fixed bars on either side, planning to start on final mechanism thursday if not Wednesday
- Drivetrain/Lift
  - Finalized dimensions of drivetrain (26” x 34”)

    - Two 6” colsons, one 6” omni per side
    - Chain in tube WCD
    - Scissor lifts fold under drivetrain
- Elevator
    - Thought about dimensions of elevator, no finalized dimensions
    - Considered C-channel to reduce space taken by elevator
    - Brainstormed possible methods to design elevator based on past strategies

1/9

Goals:

- Finalize elevator dimensions - complete
- Improve inatake #3 - complete
- Choose intake mechanism - complete (intake #3 with modifications for cargo)
- Drivetrain CAD - in progress
- Design lift mechanism - in progress
- Kernflake mounted with camera & raspberry pi with vision code - complete
- Set up new batteries - in progress
- Put closet back together - complete

- Elevator
    - Elevator design finalized, switched from cascading to continuous for speed in retracting elevator and simplicity
    - Nested aluminum channel selected for each stage, with square bar in the middle of the assembly with a bearing and spacer system as rollers
    - Began to document necessary parts and CAD the subsystem
- Egg Slicer
    - Modified the Eggo slicer to have a raised platform from hinge plane, worked better and Eggo Slicer grabs tightly. Redesigned retracting velcro design (velcro sandwich) to increase efficiency, worked, but not as well as hoped. Decided to continue with Eggo Slicer and maybe try to incorporate a ball mechanism into open jaws.

Drivetrain/Lift

- Decided to have roller drop off lead screw to begin pulling legs up
- Worked on drivetrain CAD

1/10

Goals:

- Start on final intake design - in progress
- Finalize elevator materials list - next meeting complete
- Continue elevator CAD - in progress
- Finish drivetrain CAD - next meeting complete
- Lift materials - in progress
- Mount spoil board - next meeting complete

- Build mount for Kernflake intake- Not started

Egg Slicer
- Started assembly with 4” stroke piston
- Had to wait for router pieces to slowly accumulate and assemble

- Elevator
  - Worked on CAD
  - Finalized material list
  - Fleshed out design (including pulley system)

1/11 & 1/12

Goals
- Decide on elevator cord - complete
- Finish elevator parts list - complete
- More elevator CAD - in progress
- Design 3d printed pulley - complete
- Finish egg slicer cad - complete
- Finish egg slicer assembly - in progress
- Router completely functioning - complete
- Finish intake mount - complete
- Finish drivetrain CAD - complete
- Lift CAD - not started
- Finish lift parts list - complete
- Ultrasonic sensor functioning
- Get kernflake running with updated firmware
- Build hab - complete

Drivetrain/Lift
- Drivetrain CAD finished
- Parts have been ordered to manufacture next week
- Thinking of modifying ironing board legs for scissor lift legs

Egg Slicer
- Finished CAD for Egg Slicer final
- Continued assembly

Elevator
- Finished parts list
- Designed and tested 3D printed pulley
- Started designing attachment for the egg slicer (flippy thing)

- Brainstormed designs and created CAD for 2 of them

1/15

Goals

- Finish routing intake brackets - in progress
- Cut metal for drivetrain - complete
- Assemble drive gearboxes - complete
- Finish building intake
- CAD electrical board - in progress
- CAD lift
- Turret for programming

Drivetrain

- Cut drivetrain metal
- Gearboxes assembled except motors need to be added

Egg Slicer

- Continued assembly
- Almost finished, aside from 2 piston hinge brackets
- Decided to try to use a 1/2” stroke Bimba piston after 4” stroke piston was causing severe hatch warping
- Started CAD with 1/2” piston modification

Router

- Continued routing brackets for intake
- Realized we were cutting 5052 aluminum on Saturday, switched to 6061 today
- At first cut much better, then began having issues with gumming and chip clearing today
- Will try raising spindle RPM tomorrow

Electrical Board

- Importing electrical components onto files
- Planning out placement of electrical components
  - Decided to have the PDP in the middle of the electrical board in order to have a clear connection between the motor controllers and the PDP. Also, having the PDP in the middle allows us to have easy accessibility
  - Decided that the motor controllers should go in a upside down U formation in order to have a clear understanding of how the CAN bus should be runned
- Decided to have two extra Talons SRX onto the electrical board for emergency use
  - A total of nine motor controllers are needed

Code

- Added the drivetrain from the 2018 Withhold repository to start testing with Kernflake

1/16

Goals

- Put motors on gearboxes - in progress
- Finish routing intake brackets- complete
- CAD electrical board - in progress
- Finish intake V1 with ½" piston- complete
- Make flippy thing arm - complete

Egg slicer

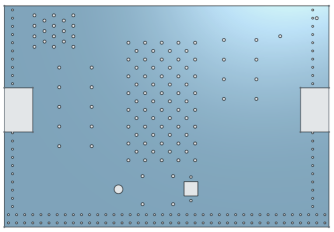
- Finished assembly and tested (works well)

Elevator/Flippy thing

- Redesigned brackets for inside 3" c channel
    - Routed new ones
- Made prototype arm

Electrical Board

- Dimensioning holes for each electrical components
- Converting holes onto the CAD model for the electrical board



Code

- Added the turret subsystem to start experimenting with vision code on Kernflake
- Added test for serial communication between the Raspberry Pi and RoboRIO
- Created the vision repository to house all of our vision code
- Wrote basic detection script for retroreflective tape and green LED ring

1/17

Goals

- Put motors on gearboxes - in progress
- Test ½" stroke intake - complete
- Did drivetrain CAM - complete
- Add pistons to flippy thing - in progress
- Started adding motors to gearboxes-in progress
    - Pinion gears need to be pushed down more
- CAD Electrical Board - in progress

Elevator/ Flippy thing

- Added 4" stroke piston to flippy thing arm
    - Worked but not well

Electrical board

- Deciding the battery placement
  - Battery was placed towards the center facing a vertical position in order to be easily accessible
- CAD Battery Model
- Imported the CAD battery model onto the electrical CAD model

1/18

Goals

- Drill out corner gusset holes to 3/16” - complete - complete
- Cut eight 3” long 1x1 square tube pieces - complete - complete
- Add hex hubs to wheels - complete- complete
- Cut holes in drivetrain - complete
- Need to find another bearing for the other one-in progress
- Finish putting motors on gearboxes- complete
- Make intake V2 - complete
- Finish flippy thing prototype - complete
- Finish CAD Electrical Board - in progress

Drivetrain

- Cut practice drive rail
- At first holes were slanted because of the uneven pressure applied when clamping in the tube jig
- After changing the clamps, routed straight
- Need to set permanent zero for tube jig to decrease machining time
- Hex hubs put into wheels
- Bumper standoffs cut
- Corner gussets drilled out
- Pinion gears completely pushed down on motors
- One gearbox completely done

Elevator/ flippy thing

- Replaced 4” piston with 2” piston for arm
  - Less difficult to pull arm up

1/19

Goals

- Refine flippy thing design
- Attach egg slicer to kernflake
- Make hatch pick up and drop off field elements

- Route elevator brackets
- Route drive train
- Finalize motor, gearbox, and ropes for elevator
- Finish CAD Electrical Board- Complete

Drivetrain
- Metal routed
- Bellypans redesigned

Egg slicer
- Attached to Kernflake
- Tested on the rocket hatch dock with shocking success

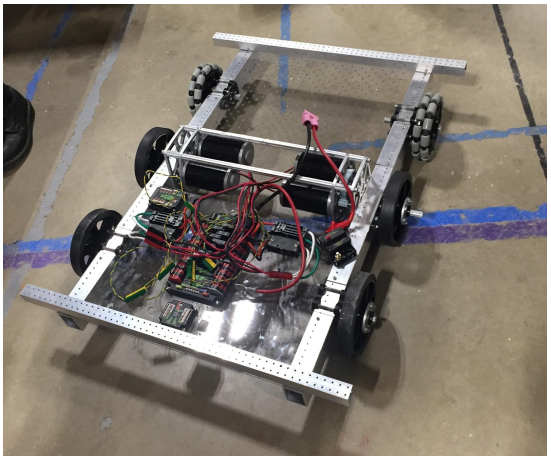
Elevator
- Made gearbox (BaneBots with RedLines)
- Cut 5” and 3” c channel to length
- Updated pulley CAD

Code
- Added a TCP server (what other teams recommended on Discord and Chief Delphi, and what 254 used in Stronghold)

Electrical Board
- Finished CAD electrical board model
- Preparing to route the CAD model for the electrical board

1/21

Drivetrain
- Bellypans routed
- Initially when chains were assembled and put in tubes, chains were rubbing on bottom and couldn’t spin freely
- Eventually cut slots in the bottom where the sprockets were so they wouldn’t rub
- That helped chain spin freely, especially after being tensioned
- Wheels were put on
    - Middle wheel is about an 1/8in too high due to bearing blocks moving outside wheels slightly down
    - Will test and see if middle wheel needs to adjusted



        - Might add tread to middle wheels to make them touch the ground
        - Might switch front omni wheel if too spinny
    - After putting on bellypan, robot needs more cross bracing as the lexan is relatively thin
        - Churros added between gearboxes to improve stability
        - Will add corner brackets
        - Lift and elevator will also provide cross bracing

Elevator

- Cut square tube to length
- Began CAD for elevator mounting system
- Printed new pulleys
- Began making back spacers for elevator
- Started CAD for holes on stages

Code

- Added the ArcadeDrive command to the project (ported from last year's code)

Electrical Board

- Transfer electrical components unto the electrical board
  - PDP, 40 and 30 amp Breakers, VRM, Four Talon SRX and Main Breaker
- Connected the Talon SRX motor controllers into the PDP
- Powerpoled drivetrain motor controllers white and green wires
  - Connect power pole connection to the drive drain motors power poles
- Powerpoled Can Bus and connect them with each other

1/22

Elevator/Flippy Thing

- Tested flippy thing redesign
    - Design did not work because it locked and could not be pulled back up
- Decided it might not be beneficial to spend as much time/weight on the flippy thing and should move towards a simpler design

Code

- Added the code to test the hatch prototype

Electrical

- Placed NI roboRIO and access point on the electrical board
- Finish wiring the electrical board
  - Connect the NI roboRIO, VRM and PCM to the PDP
  - Connect the access point to the VRM

1/23

Goals

- Battery mount - in progress
- Robot driving - not started

- Begin lift construction - in progress

Lift
- Determined legs should be 30” with the pivot point in the center
- Cut legs

Elevator
- Did weight analysis to decide if we could use existing elevator design or if we needed to remove the 5” c channel due to weight constraints
    - The original elevator design is ~35 lbs, without 5” c channel it’s ~17 lbs
    - Due to the drivetrain and lift weight being only 50lbs together, decided that the 35lbs elevator is probably better due to the fact we want the elevator to be sturdy

Code
- Wrote the auto-centering code to work with the spinning turret and vision code
- Removed the serial test script, as the TCP server works well and the serial communication stream was not tested as extensively as the TCP server and is theoretically more prone to issues due to serial port coming out at either end

Battery mount
- The CAD model of the battery did not work because the battery was blocking the wires coming out of the PDP.
- Decided to redesign the battery mount
    - Design a battery mount that's able to hold the battery from three sides and hold the fourth side with a bungee cord
    - Used parts from an old cut piece of drivetrain metal in order to hold the battery
    - Tested battery mount
        - Battery mount failed because the spacing between the battery and the screw was too tight.
- Decided to redesign the battery mount again

1/24

Goals
- No goals due to informal meeting due to science fair

Elevator
- Finished CADing holes
- Machined holes on innermost stage
- Made 15:1 gearbox with encoder and two 775 motors
- Tested and verified that the gearbox works

Lift
- No work due to science fair

Code
- Wrote code for use with the new drivetrain rather than Kernflake (more akin to what was written for last year’s robot)
- Added camera streaming

Battery Mount

- Looked back at steamworks battery mount and made some design changes
- Remove unattached rivets, nuts, and bolts
- Placed slide nuts and bolt in order to place the battery mount on a 80/20

1/25

Goals

- Drill holes into C-Channel
- Rivet rollers into square tube and C-Channel
- CAD lift - in progress
- Drive robot - complete!
- Prep lift motor - complete
- Plan out lift components - in progress

Lift

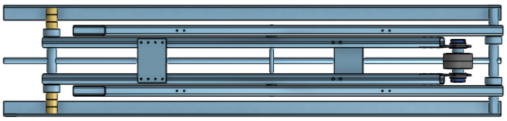
- CAD started
- Initial gearbox (50:1) mounted on minicim

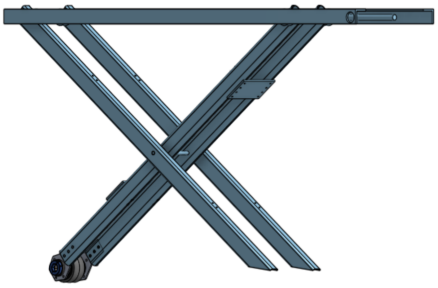
Drivetrain

- Very fast, turns around back wheels currently
- In initial practice, was placing low hatches with a 10-15 sec cycle time
- Black wheels will wear over time, will need to get some extras to switch out and can also switch middle and back wheels to reduce wear due to the “reverse drop center”

1/26

Lift





- Finished CAD
- Had to adjust leg lengths and pivot point so the lift could fold flat and come up keeping the robot relatively flat
- Fleshed out rider and hinge design
- Might end up having rider fall out at the end so legs lift away from HAB instead of towards HAB
- Changed motor gearing

    - 34” long 16 tpi lead screw in 10 sec means minimum RPM needs to be about 5000
    - Free speed of a minicim is about 5000 RPM, so no gearbox will be used, with a 1:1 external reduction to attach to lead screw
- Began manufacturing
    - All metal has been cut
    - Holes have started being drilled
    - Rider and roller wheel assembled

Drivetrain:

- Difficult to control to do high speed and turning with omni wheels in front:
    - Tried switching omnis to high grip but resulted in too slow turning.
    - Added code for manual power limiting to 80%
    - Shaft collar fell of wheel wall driving and had to replace it with a new one due to a striped shaft collar.
    - Attempted omnis in front of drivetrain, but exaggerated turning led to replacing them in the front again.

Intake:

- 1/16 arms bent from repeated use picking up hatches
- Switched to 1/8” lexan which works just as well
- Will need to fabricate more in case they break in competition

Code

- Refactored the vision code to make it faster and more readable and changed the green color parameters used by it

1/28

Goals:

Elevator Stages: Connected one stage
Lift Assembly: All holes and cuts made
Full Flippy Man CAD: Complete
Pneumatics Board: CAD Complete, no Assembly
75% Flippy Man completion: Not Started

1/29

- Meeting cancelled due to weather

1/30

Goals:

- Assemble Lift: All holes drilled
- Finish Elevator Stages: Drilled side bearing holes
- Route Elevator Brackets: Not Started

Lift

- Fixed some holes that were drilled wrong
- Shifted people temporarily to help finish elevator

Code

- Wrote lift code
- Wrote the elevator subsystem and command

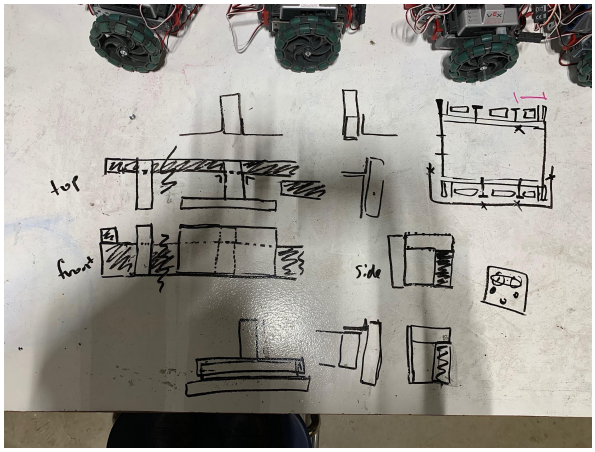
1/31-2/1

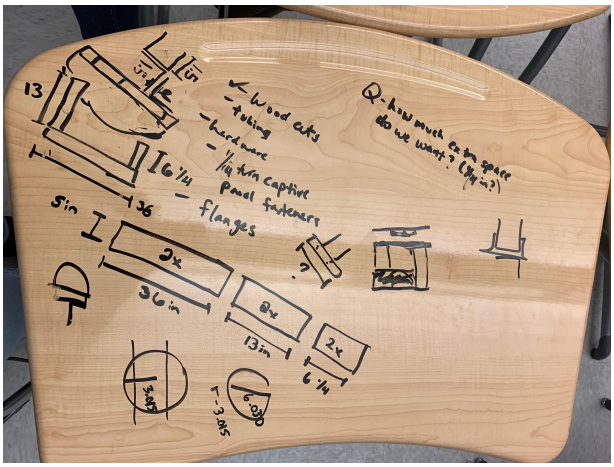
Goals:

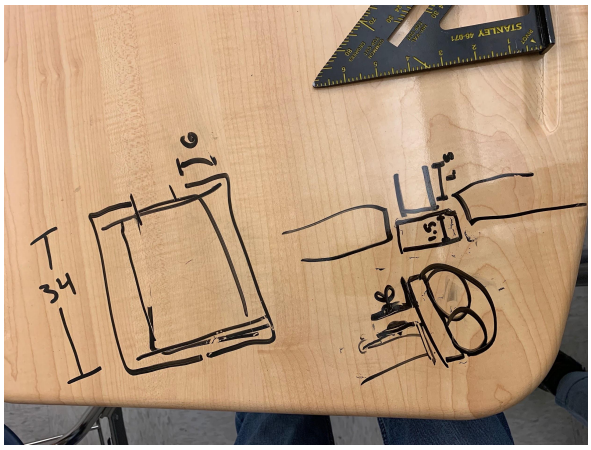
- Finish elevator: complete
- Mount elevator: complete
- Run elevator
- Finish lift: complete
- Mount lift: in progress
- Start bumpers: Brainstormed and designed
    - Intenvory materials Done
    - Mounting solution Designed
    - Dimensions Most dimensions measured
- Route flippy boi brackets: complete
- Assemble flippy boi: complete
- Finish pneumatics board:
- Fix auto-centering code (low pass filter?):

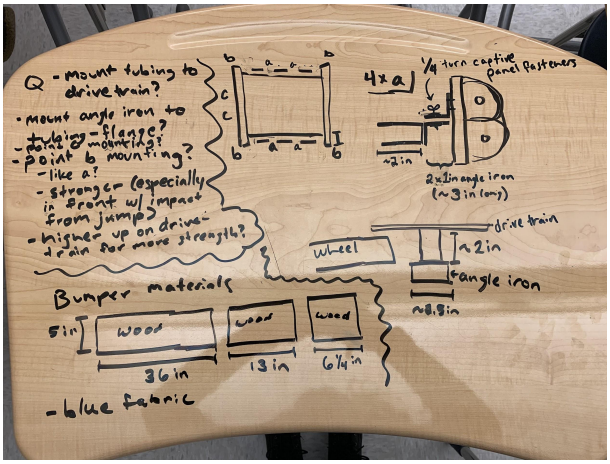
Bumpers

- Brainstormed and designed bumper mounting system
- Will be braced with “T”s between the wheels to maintain 8 inch frame









Lift

- Decided on method to mount to robot
- Drilled holes and cut pieces for mounting
- Finished assembling all other pieces
- Planning on cutting down the lead screw after testing to determine where it should fall out so the legs swing backwards instead of into the HAB
- For second iteration, will be made out of square tube so it can be routed to make manufacturing easier
    - Will be slightly heavier, but more weight lower on the robot isn't a bad idea
    - Legs are steel right now, but pretty thin, so strength shouldn't be significantly affected

Elevator

- Finished attaching rollers to each stage
- Finished crossbars for each stage
- Began mounting elevator
- Began assembling carriage with intake on it
- Attached pulleys to each stage and mounted the motors/gearbox
- Ran pulleys
- Cut outer stage to allow room for intake

2/4

Goals

- Mount elevator roller and motor - complete
- Test elevator - not done
- Finish assembling lift mount - almost complete
- Mount flipply man to elevator - complete
- Finish pneumatics board - in progress
- Rebuild robot cart - complete

Lift

- Cross supports riveted on
- Mounts bolted on
- A few mounting holes need to be adjusted to fit under robot
- Sprocket needs to be put on lead screw

Code

- Tested vision code using distances in x and y to a target and angle, rather than solely an angle for the first time
    - Distances were massively off
    - Math was revised multiple times, but distances still seem off

2/5

Goals

- Test elevator - complete!
    - Run rope - complete
- Mount lift - in progress
- Bumper standoffs - not started
- Finish jet pack (pneumatics board)- in progress

Code

- Shift to use of HashMap to represent setpoints in the code (dummy setpoint values used for now)
- Started overhauling the Pathgen, GyroCompMPRunner, MotionProfileFiller, MotionProfileRunner, and TalonPath classes that interfaced with Pathfinder from the 2018 season, particularly so that they would work across seasons in our Java-Libs repository
- Migrate from use of an enum with associated values to the use of an enum and a HashMap to store elevator setpoints and values

Eggo Slicer

- Routed new backplates out of frosted lexan with indentations to suppress rotational freedom and attached felt pads to the front to stop pivoting and allow the hatch to slide when pulled onto the back plates.

2/6

Code

- Wrote and tested the code for flippy-boi - needed to fix logic so starts down, but otherwise good
- Add Motion Magic control to the Elevator subsystem and ControlElevator command
- Fixed the generic within the Cycler class to drop requirement that elements used to cycle are enum values and use the new Cycle interface
- Finished Pathfinder code migration, resulting in AbstractDrive class to carry drivetrain functionality over across years and to enable Pathfinder code to work
    - The AbstractDrive class only interfaces with CTRE Phoenix-enabled motor controllers right now (as this is what the team typically uses in competition), but it would be nice to write code such that the REV Spark Maxes and PWM controllers would work as well
- Added PID, FPID, and PIDVA data classes to contain PID constants in a more intuitive and clear way than with an array, which has been the solution in the past
- Finalized vision code formula:
- Coordinates of setpoint to use in motion profiling: $(Xi + Xs, Yi + Ys)$
- X and Y distances to vision target used as setpoint: $(\Delta y * tan(\theta h), \Delta h/tan(\theta v + Ca))$, where the Δh = Camera Height - Height of the midpoint on line between the center of both contours.
- Horizontal Angle to target (θh)= $(atan((CFx - CCx)/Fl)$
- Vertical Angle to Target (θv) = $(atan((CFy - CCy)/Fl)$
- When tested with no camera offset angle, the X and Y distances to the target are accurate to within ~5 in.

Eggo Slicer

- Changed out backplate standoffs to be longer and increase hold

2/7

- Had team meeting to determine essential things we need to get done and timeline for the next 12 days

Goals

- Rerun elevator rope - complete
- Tension rope - complete
- Fix bottom pulley - complete
- Figure out if lift needs to be redesigned - in progress
- Make narrower intake brackets - complete
- Install new intake brackets - complete
- Mount roborio and RSL - complete
- Mount LED's - complete

- Fix mounting brackets for jetpack - complete
- Mount jetpack - complete
- Determine setpoints for elevator - not started

Lift
- Decided to start with using 2x1.5 angle iron instead of 1x1 for lift
    - Couldn't cut down to 2x1.5 yet, so first will test with 2x2 and determine if there are other issues
- Mostly disassembled the lift, took off mounting supports
- Tried to route .75" hole in angle iron, but couldn't clamp well enough so there was too much chatter
    - Began drilling holes instead - one is done, one still needs to be done
- Mounted motor to piece of wood for testing setup

Carriage
- Fixed the carriage by putting rivets in the corners of the inside brackets and replaced outside brackets with metal that also attaches Flippy Man to outside of carriage.
- Changed tolerance between bearings to be tighter around the carriage, therefore eliminating wobble.

Code
- Started migrating prototype code for the two methods of movement in the various routine CommandGroups based on the messages received over a TCP server (Pathfinder and angle-based PID loop) to a class-based format

2/8

Goals
- Bumper mounts on robot - in progress
- Encoder mounts on drivetrain - complete
- Test lift with larger angle iron - complete
- LED code

Code
- Pathfinder-based movement with TCP server messages completed, but untested
- Made Pathfinder-interfacing library code much easier to use by shortening constructors

Lift
- Finished making lift with 2" angle iron
- Tested, but motor didn't have enough torque and gear ratio couldn't be increased as the lift would then take too long

- Decided to switch to powering with rope and pulleys instead

2/9

Goals

- Tune all PID loops - complete
- One set of bumpers made - in progress
- Start designing cargo mechanism - complete

Lift

- Went through many small iterations which gradually improved functioning
- Tried miniCIM with 5:1, 10:1, and 20:1 ratios
- Tried dual input redlines with 12: 1 ratio and 775pros with 25:1 ratio
    - Decided to use 775pros with 25:1
- Added in a “kicker”
    - Rope is routed over it to push wheel end of legs down

Code

- Fixed the FPID constructor to have the F gain first in the library, which the drivetrain code assumed and is more intuitive
- Migrated to use of two 775pros/TalonSRXs with the lift
- Added control for the lift with right joystick on co-driver’s Xbox controller
- Tuned the left half of the drivetrain for PID with F gain
- Fully tuned the elevator PID

2/10

Goals

- Vision code reading right distances - in progress
- Both sets of bumpers made (maybe not covered yet) - in progress

Lift

- Continued lift prototyping
- Decided to move kicker to right in front of the motor instead of to middle of mechanism

Code

- Added setpoints and a conversion factor for ticks to inches and vice versa to the elevator
- Added setpoint control with the buttons on the co-driver’s Xbox controller
- Merged all code from the vision branch to the master branch
- Wrote hatch control code and merged it into the master branch
- Used limit switches on the hatch intake to indicate whether a hatch is currently in the hatch intake, updating the hatch intake appropriately (as an alternative to manual control) and providing a visual indicator on Shuffleboard of the intake state

    - Provided a toggle to switch between a one and two limit switch threshold for this code
- Added skeleton for Manager command to switch between manual control and the autonomous routines (CommandGroups) for moving hatches

2/11

Goals

- Full autonomous hatch movement - in progress

Cargo Mechanism

- Finished CADing the ball mechanism, started testing optimum heights, found that 11.5” was ideal.

Lift

- Decided to make lift out of 80/20
- Would be easier to put together and eliminate some of the wobble present in it now

Code

- Added InstantCommand for setting the Hatch to a position
- Worked on completing the manager class, primarily on the CommandGroups that it actually runs

2/12

Goals

- Assemble 80/20 lift - in progress
- Finish at least one set of bumpers - in progress
- Calibrate camera - not started
- Full autonomous hatch movement - in progress
- Compare angle pid and motion profiling - in progress
- Build level 2 of hab - in progress
- Fix racetrack

Code

- Tuned PID on the left side of drivetrain
- Fixed PCM CAN ID after a bunch of debugging
- Switched to a toggle to switch between manual and setpoint control, in accordance with last year’s code and control scheme
- Disabled the error in AbstractDrive that is thrown whenever there is no gyro or navX plugged in (testing needed to be done without them)

2/13

- Finish assembling 80/20 lift - in progress
- Finish level 2 of hab - complete

- Finish one set of bumpers - in progress
- Full autonomous hatch movement - in progress
- Compare angle pid and motion profiling - in progress
- Limit switches - complete

Lift
- Drilled all holes
- Assembled around pivot point with c channel

Code
- Merged all of the PID tuning code into the master branch
- Added the turn PID controller from the earlier isolated prototype to the main project's code to develop turn PID controller routines in addition to Pathfinder routines

2/14

Goals
- Full autonomous hatch movement - complete
- Compare angle pid and motion profiling - in progress

Testing Log
- When driving, realized we were experiencing brownouts when turning even though the battery was charged
- Checked chains and CIMs and determined it wasn't a mechanical issue
- Checked through electrical and found a drive motor controller had been smoked
- After replacing, problem was fixed

- Started making more field elements

Lift
- Added on motor and cross supports
- Decided to run with rope instead of steel cable as it couldn't spool as well
- Tested with 25:1 on 2 775pros
- Needed to add in more spacers to go up all the way as some brackets were hitting
- Smoked one 775pro while testing, but still got almost all the way up with 123 lbs of weight
- Later we found out that only one motor was running, so with two it should be able to get up all the way with no issues

Code
- Tested the routine code, which worked in its progression, although its accuracy was not checked

- Added current/velocity limiting code to ensure that current wasn't being used to pull the elevator down when it was already down
- Added code to reset the elevator's position when it hits the limit switch newly installed at the bottom of the elevator
- Flipped the directions for the solenoids on the robot
- Negated limit switches after discovering that they were pulled high through testing the limit switch hatch grabbing code
- Split the TCP server for vision processing into a server and processing components, rather than doing them all in one place
    - This means that we could potentially speed up the entirety of the vision code pipeline by multithreading/multiprocessing the Raspberry Pi and robot's processing code
- Refactored the Hatch intake code so that there was a HashMap to make solenoid directions easier to flip in the future and clearer method names

2/15-2/19

Lift

- Put on robot and tested, but didn't have enough power to get all the way up
- Experimented with 3 different sizes of steel cable vs. rope vs. fishing line
- Made the kicker different sizes (4 in and 10 in)
- Experimented with different amounts of wrapping
- Added more supports on the robot
- Cut longer distance for rider to fall out
- Finally, had enough power to get up with 100:1 on 2 775pros, a 10 in kicker, and 3 runs of medium size steel cable
    - However, anchor rod was not long enough to be collared and popped out when testing, twisting the mechanism so it wasn't able to complete the climb
    - Lift also had too much tippiness
- Was taken off the robot before bag - will add longer anchor rod that can span the width of the whole robot, providing more support to combat tippiness and will prevent rod from falling out

2/15

Goals

- Compare angle pid and motion profiling - complete

Code

- Updated the elevator ticks to inch conversion factor with a more accurate one (not sure when there was a change, but retensioning so many times and reworking the pulley system in minor ways might have made it inaccurate)
- Updated the hold voltage for the elevator
- Started the limit switch resetting elevator Command, which had not been started before, and negated that limit switch's value, as only the limit switches on the hatch intake were known to be pulled high before

- Stopped the jerk from the drivetrain's turn PID controller running when the robot is starting up
    - There should definitely be a better way to fix it than by checking to see if it has been signalled for use at least once (i.e. we need to determine why it was trying to run in the first place
- Decided on the use of angle PID, as the motion profiling was dependent on the distance measurement of the vision code, which was not working reliably at all distances (likely due to the camera not being calibrated)

2/16

Code

- Added LimiLimitedHashMap class to libraries to hold a limited amount of entries in a HashMap (extends LinkedHashMap and implements the removeEldestEntry method)
- Added prototype lag compensation for the vision code, but did not enable it
    - Attempts to synchronise times between the Raspberry Pi and the FPGA to get accurate timestamps of when images were taken, which definitely needs to be tested for accuracy
    - The Drivetrain subsystem has LimitedHashMap to hold the times and gyro heading of the robot
    - Finds closest position at any given time to use as a reference (does not approximate it using a spline curve, as other teams - such as 254 - do which could potentially be better if written to perform well)
- Toggled the limit switch number requirement for the hatch intake to require both
- Invert the lift's slave TalonSRX motor controller
- Fix error in Hatch subsystem where the stowed value of the HatchStowedState enum, which was added when the Hatch code was refactored, was added to the stowed HashMap twice and the unstowed state was not added at all, leading to a NullPointerException

2/17

Code

- Enabled the lag compensation code for testing
    - Did not work → the synchronisation of clocks seems to be the issue, as the correct timestamps are not being assigned to the gyro positions and the correct reference frame cannot be found, even when there is an extremely large LimitedHashMap entry limit
    - Not enough time to fix, so it is being disabled for now
- Made the TCP Server connection more persistent on both ends (doesn't crash on disconnect anymore)
    - Both ends now reopen their connection/look for an open server as soon as their connection goes down and don't need to be restarted unless the robot code is deployed while the robot is running (doesn't issue a formal disconnect message?)

2/18

Code

- VISION CODE ACTUALLY WORKS (kind of)
    - The code originally used minimum distance from the closest piece of tape to the center of the frame to determine what tape to use, which was switched to second smallest distance (or smallest, in the case of there only being two pieces on the screen) after discovering that our field was built wrong
    - The second smallest still works with the rocket extremely well, but seems to have issues with the side of the cargo ship unless it is lined up extremely well
- Rather than using full routines, switched the tape centering controls to be on the joystick and having the co-driver's control to be uninterrupted
- Wrapped the values for the gyro to be within -180 degrees and 180 degrees, as this is what the turn PID controller expects

2/19

Drivetrain

- Discovered that there was no grease in the gearboxes
    - We had none in-shop, so this will have to be done during unbag

Code

- Wrote and enabled code for running in Sandstorm, which was really just checking to see whether a list of commands had been enabled before, and enabling them if they had not been

3/1

Code

- Transitioned to the detection of tape pairs and then detection of which tape pair is closest to the center of the frame in the vision code, rather than the unreliable second smallest distance method that was used before
- Added testing suite with images for use with the vision code in the future

3/2

Unbag Part 1

Drivetrain

- Put the grease in the drivetrain

Code

- Added a timer decorator to the vision code repository to add timing benchmark functionality easily and efficiently
- Created a visual horizontal angle detection script that uses pair detection in the vision code repository, to test later

    - Tested and works seemingly just as well as the old one with the rocket, and works significantly better with the cargo ship
- Added a non-visual script to accompany the visual one, which will have to be tested later

Electrical

- Fix LED Board
- Put a Raspberry Pi on the robot
- Replace battery cable
    - Decided not to replace the battery cable because it didn't make a noticable difference for the main breaker to the battery
- Replace broken Limit switch
    - Design changes from soldering wires onto the limit switch to crimping wire to the limit switch in order to replace them quickly if broken in the future

3/3

Unbag Part 2

Code

- Tuned the right half of the drivetrain now that the grease is in

Electrical

- Attach Limit switch to intake
- Route wires through the race trace for the motor for the cargo carriage
- Fix LED board

3/4

Unbag Part 3

- Tested the ball intake, which worked
- Found that the arm that the hatch and ball intakes were mounted to bended too far down to pick up hatches
- Broke metal bars and fingers on the hatch intake
- Discovered that the drivers had no control after autonomous period ended and teleoperated period started in a practice match

Code

- Tuned the left side of the drivetrain to work as reliably as the right side
- Flipped the solenoid directions in the hatch intake
- Disabled the elevator while it is unoperational
- Fixed the control bug by removing the disabling of all control commands at the start of the disabled period
    - This should not be an issue, as the commands should still be re-enabled at the start of teleop, but further testing will be done to figure this out

Electrical

- Replace stripped encoder cable
- Add a SRX Magnetic Encoder to the new motors
- Add ninth motor controller to the can chain
- Connected motor controller powerpole to the motor
- Attach lexan to intake for the Zero set point limit switch