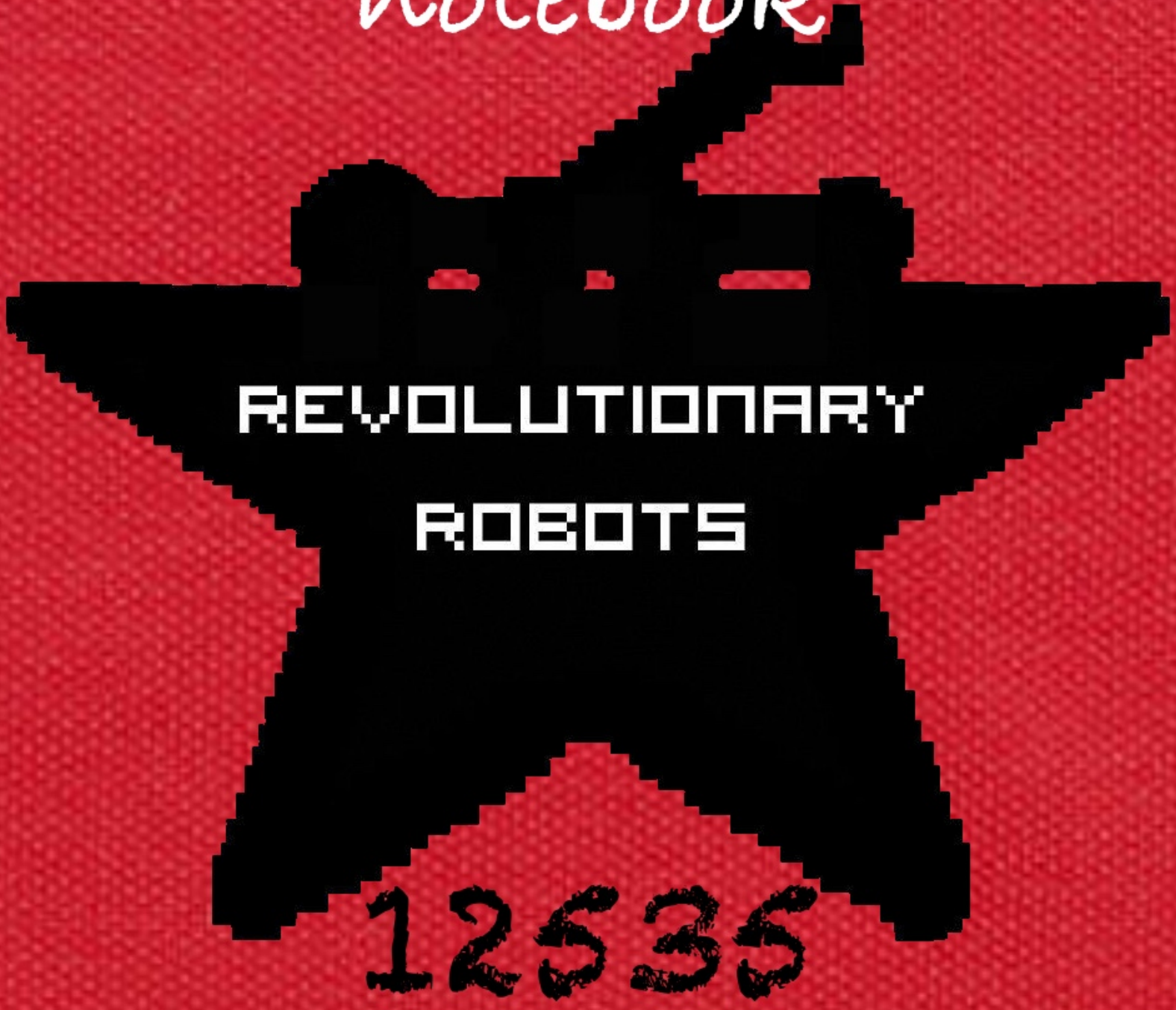
Engineering notebook



The Revolutionary Robots Team #12535

Team Summary

★ **Engineering Section:** The engineering section of our notebook is meant to take you on a journey of our learning experiences throughout the season over the robot design process. It shows our goals, progress on those goals, and through pictures show a visual version of our robot progress. We worked this year on our decision making processes and using engineering concepts

See Tab A for Engineering Notes Pages 20, 20 A-D re tipping point calculations
See Tab B for Engineering Notes Page 55 re speed of linear slide

★ **Computer-Aided Design:** We created the Computer-Aided Design section to show the evolution of our team's CAD process. You can see our overall process of designing the robot in CAD and highlight specific parts of the robot such as the wheelbase and linear slide. We also have drawings of components of the robot, such as our linear actuator as well as all the 3D printed parts on our robot.

See Tab C for CAD Note Pages 8-10 re wheelbase
See Tab D for CAD Note Pages 32-36 re linear slide

★ **Programming Section:** The Programming Section of our notebook shows the evolution of the programing and our advancement with sensors. This year we have started to use the gyro sensor for autonomous for consistency which is needed. We use a touch sensor on the actuator to stop it from hyperextending.

See Tab E for Programming Note Page 9 re gyro

★ **Outreach Section:** The Outreach Section is divided into four categories of Outreach we worked on this season. Community Engagement show the work we have done to share *FIRST* and STEM with our community, but also just giving back. Mentoring and Elevation demonstrates the work we have put in helping FLL and Jr. FLL teams and encouraging their journeys. Industry Connections exhibits our work to share what we do with others in STEM industry while also learning from them. FTC Collaboration shows the ways in which we worked with other FTC teams.

See Tab F for Outreach Page 1 re our Outreach with the biggest reach - Tinkerfest
See Tab G for Outreach Page 6 re the connection we have with our FLL mentorees

★ **Goals, Accomplishments, and Advancements:** As a second year team we have accomplished many things. Last year we finished thirteenth in the qualification matches at state last year as a rookie team. Since then, we have improved nearly every aspect of our team including robot assembly and our notebook. We plan to continue adding knowledge to our team to continue the Revolution into the future.

See Tab H for Engineering Note Page 17 for a picture showing how far we have come
See Tab I for Business Page Page 4 for our Sustainability Plan

TABLE OF CONTENTS

Electrical

Computer-Aided Design

Programming

Team Education

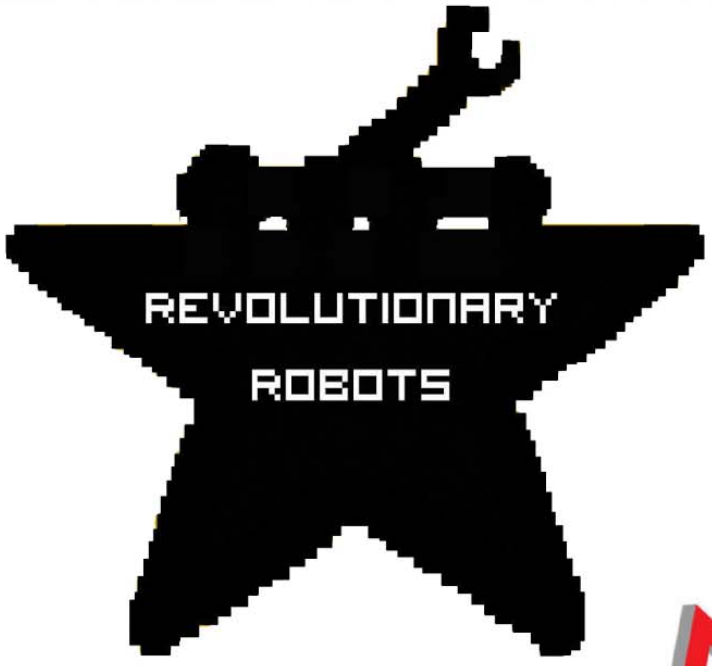
Checklists

Reference Materials

Game Manual Part 1
Game Manual Part 2
Engineering Notebook Guidelines
Wiring Guide
Expansion Hub Guide

Revolutionary Robots
12535

Team Information
2018-2019



Moore Community STEM Club

FIRST®

Team Leads

We encourage team members to work some in each area in order to gain a more balanced *FIRST* robotics experience. However, we have leaders for different areas of the program. The leaders are responsible for making sure all the work in their area is accomplished in a timely manner.

Team Member	Area of Focus
Drew Busch Jackson Elliott	**Parts and build**- Responsible for the ordering of all parts, the transportation of all parts to all events, as well as supervising the build of the robot.
Olivia Martin Ivory Moore Jackson Busch	**Notes, outreach, and flyers-** In charge of inspecting all notes that are taken, organizing the details of all outreach events, and designing pamphlets that are distributed to tell others more about our team and promote the First program at all team events.
Mark Hazen Andy Miller	**Programming-** Responsible for both the TeleOp part of the match as well as programming the fine-tuned autonomous.
Drew Buch Mark Hazen	**Design and CAD-** In charge of designing all parts that need to be 3D printed in the CAD program Inventor Professional 2018.
Ivory Moore Oliva Martin Josiah Shannon Jackson Busch	**Strategy-** Responsible for strategizing with other teams and finding alliance partners that would be the most beneficial to our team.

Team Bios

Drew Busch

Drew Busch is fourteen years old and lives in Oklahoma City, Oklahoma. This will be his second year in the *FIRST* Robotics organization. He is the general for the Revolutionary Robots this year. He hopes that being a part of the *FIRST* organization will give him enough robot experience for future jobs in the future as a mechanical engineer.

"GET BACK TO WORK"

Team Bios

Mark Hazen

Mark is thirteen years old. He is in the 8th grade. Mark is the lead programmer on the team. He is also on the Robot Design group. Mark participated in FLL for two years as part of the Electrons (Team #15846). He decided to join an FTC team to learn more about programming and building robots. This is now his second year. Mark mentored the Electrons last year. Mark wants to be a roboticist and an astronaut.

"You cannot use Microsoft Edge and keep your sanity."

Team Bios

Ivory Moore

Ivory Moore is sixteen years old and the oldest member on the team. This will be her seventh year in the FIRST program. This is her second year in FTC, the other five years of her experience come from the FLL program, where she participated for four years and mentored for four years. She mentors the Revolutionary Robots sister team the Electrons. Her dream after high school is to go into the fields of computer science and engineering at OU.

"Numbers, numbers, math math math."- Gaige, *Borderlands 2*

Team Bios

Jackson Busch

Jackson Busch is thirteen years old and lives in Oklahoma City, Oklahoma. This will be his second year in the *FIRST* program. Jackson has been interested in robotics for most of his life and has more recently become interested in an engineering career. Jackson's dream is to attend the University of Oklahoma after he graduates high school and possibly join OU's engineering program.

"Yay Team!"

Team Bios

Jackson Elliott

Jackson Elliott otherwise known as “Flapjack” is thirteen years old and lives in Oklahoma City, Oklahoma. He loves anything automotive. He enjoys building, designing, and drawing. He attends Highland West Junior High. When he grows up he would like to be an aeronautical engineer or in the automotive industry.

“I love cars. Too much? Naw.”

Team Bios

Olivia Martin

Olivia is fourteen years old and she is in the 8th grade. She is on the robot construction team and is also on the outreach team and strategy. This is her fourth year participating in the FIRST program. Olivia would like to make a career as either an engineer or an animator.

"I think we can solve any problem if we work together."

Team Bios

Josiah Shannon

Josiah Shannon is fourteen years old and lives in Oklahoma City Oklahoma. This will be his first year in the *FIRST* program. He wishes to learn to programme and pursue a career in science and technology.

"Fools live by their own mistakes, but wise men live by the mistakes of others."- author unknown.

Team Bios

Andy Miller

Andy Miller is a fifteen-year-old 9th grader from Oklahoma City, Oklahoma. Because of his love for building and his desire to learn to programme, he would like to pursue a career in electronics with the United States Air Force.

"Gimme a sec… I'm still thinking."

Coach and Mentor Bios

Chris Hazen
Chris Hazen is the head coach of the Revolutionary Robots. He has a bachelor's degree in mathematics and a minor in computer science/programming. He works at Dell as a manager of the enterprise storage support. (Those are the big storage boxes that hold all the information, like a cloud.) He has coached FLL for two years and this will be his third year coaching FTC. He was a founding father of the Moore Community STEM Club. He also brought the *FIRST* robotics to Dell to start the Dell Scrimmage Day. "If it hurts, don't do it."

Kristi Hazen
Kristi Hazen is the second coach of the Revolutionary Robots. She has a J.D. from the University of Oklahoma. This is her fourth year of being involved in *FIRST*. She has mentored and assisted with the Moore Community STEM Club's FLL team for two years. This is the second year working with the Revolutionary Robots. She is one of the founding members of the Moore Community STEM Club. "Actually, we should add..."

Marty Martin
Marty is an Apple computer repair technician but his self-proclaimed vocation is Bohemian Tinkerer. He got his training at Apple HQ in Cupertino, California in 2008. Marty has been a hardware repair technician for The Delcom Group since 2011 and his current client is the University of Oklahoma. He has enjoyed working with FIRST for the past three years. This is his second year involved as a mentor. Marty also runs a technology blog called Doctor Geek/Nerd. "Restart. Reset. Restore."

Brandon McCabe
Brandon is a Structural and Payloads Design Engineer at Boeing and this is his first time mentoring with FIRST. He has a bachelor's degree in Mechanical Engineering from the University of Oklahoma, considers himself an amateur vexillologist and is a self-proclaimed professional rec-league softball player. "It's not right until you've done it six times."

FIRST Core Values and Our Team

- **Discovery**: We explore new skills and ideas.
    - **(For examples of our discoveries see Coach and Mentors Bios Pg. 10)**

- **Innovation**: We use creativity and persistence to solve problems.
    - **(For examples of innovation see the Engineering Pg. 34)**

- **Impact**: We apply what we learn to improve our world.
    - **(For examples of how we impact our world see Outreach Notes Pg. 1)**

- **Inclusion**: We respect each other and embrace our differences.
    - **(For examples of inclusion see Team Education Pg. 1-12)**

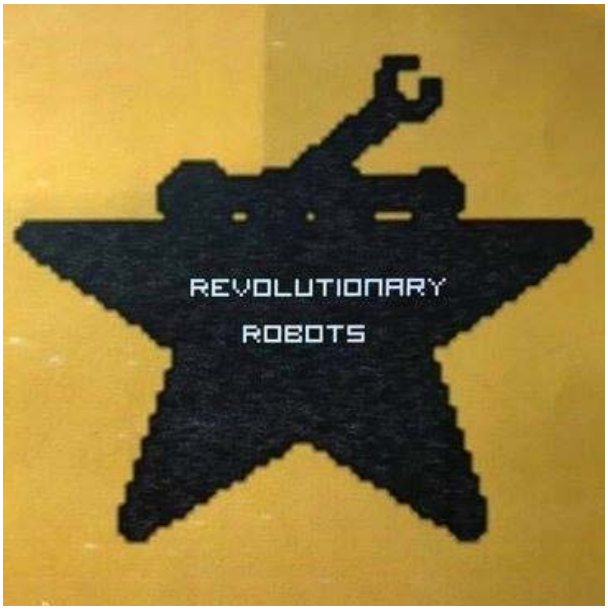
- **Teamwork**: We are stronger when we work together
    - **(For example, see Engineering Notes Pg. 6)**

- **Fun**: We enjoy and celebrate what we do!
    - **(For examples of fun see Engineering Notes Pg. 17 & 37)**

The *FIRST* core values are a large part of what our team is. We recite them before and after each meeting. We do our best to incorporate both Gracious Professionalism and Coopertition™ in our meetings at all times.

Team History

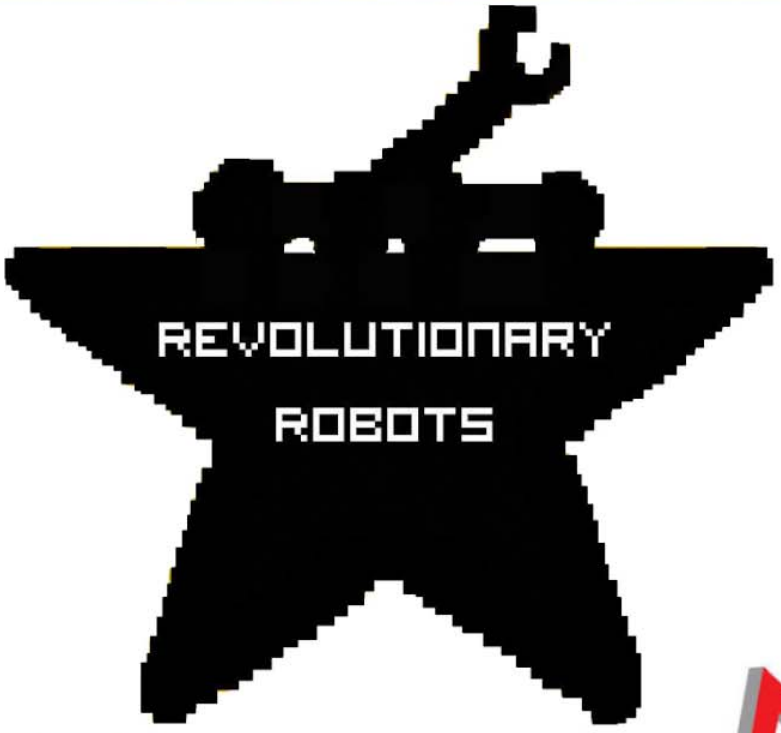
REGISTRATION	2016 Recipient of a Jump Start Rookie Grant from FIRST
ROOKIE YEAR	2017-2018
TEAM DEMOGRAPHICS	8 members 2 girls, 6 guys 1 7th Grader, 3 8th Graders, 3 Freshman, 1 Junior Students from home school, online public, private and public schools
MENTORS AND COACHES	● Coach: Chris Hazen- Enterprise Storage Support Manager ● Coach: Kristi Hazen- Attorney ● Mentor: Marty Martin- Apple Computer Repair Technician/ Bohemian Tinkerer ● Mentor: Brandon McCabe- Structural and Payloads Design Engineer
SPONSORS	● Moore Community Stem Club ● Dell EMC ● Dell Planet ● Boeing
SOCIAL MEDIA	● Facebook: Revolutionary Robots ● Twitter: @FTC12535 ● YouTube: Revolutionary Robots: FTC 12535
2017-2018 RELIC RECOVERY AWARDS	<u>Chickasaw Nation Qualifier</u> ● Motivate Award Finalist <u>Newcastle Qualifier</u> ● Final Alliance ● Control Award Winner ● Motivate Award Finalist
2018-2019 ROVER RUCKUS AWARDS	<u>Gordon Cooper Qualifier</u> ● Inspire Award Finalist ● Think Award Finalist ● Rockwell Collins Innovate Award Winner ● Motivate Award Finalist ● Control Award Winner ● Winning Alliance <u>Newcastle Qualifier</u> ● Design Award Finalist ● Think Award Finalist ● Winning Alliance

	Oklahoma State Championship • Control Award Finalist • Think Award Finalist • Winning Alliance
Some Fun Facts!	
ROBOT NAMES	2017/2018- Hamilton (Hammy) 2018/2019- General Washington (Washi)
TEAM COLORS	Blue, Cream, and Gold
LOGO	

Team Goals		
Goals	**Actions Taken to Achieve Goal**	**Accomplishments**
Qualify for State	We have built and altered both our robot and notebook while following the FIRST guidelines.	We have achieved our goal of getting to State! This means we now have a new goal or earning a spot to the Worlds Competition.
Win an Inspire Award	We tried to focus on what we perceived our weakness to be so we worked even harder on our designs, outreach and teamwork.	We were finalists for an Inspire Award at this years Gordon Cooper Qualifier, so we believe we are on the right track. We have won or been a finalist for many other awards, so we are working our way up to Inspire.
Educate team on java	We have some members who have gone over online education material over java by Will Edds, and FRC mentor who has occasionally helped us out. We use android studio to program in Java.	We have one team member who is a very efficient programmer with an excellent understanding of the language. He is helping others to understand it.
Share our knowledge about STEAM and FIRST	We have introduced some younger kids to FIRST and given their guardians information at Tinkerfest, Dell Day, and Haunt Old Town.	Many expressed interest and asked for more information.
Sustain a supportive team environment	We encourage exploration of new ideas and solutions and communicate amongst one another encouraging to build upon old ideas to make something new.	We currently support each other.
Earn a Spot to Worlds Competition	We have focused on the areas we felt like needed improvement the most after our qualifiers, such as robot reliability and	We are currently working very hard on this one!

Revolutionary Robots 12535

Business 2018-2019



Moore Community STEM Club

FIRST®

FINANCIAL STATEMENT

Income for the 2018/2019 Fiscal Year			
Date	Payor	For	Amount In
5/31/2018	Dell	Employee Donation and Match	60.00
7/2/2018	Dell	Cause Card Donation	150.00
7/2/2018	Dell	Employee Donation and Match	60.00
8/8/2018	Dell	Employee Donation and Match	30.00
9/5/2018	Dell	Employee Donation and Match	60.00
9/9/2018	Tabitha Shannon	FTC Registration	200.00
9/16/2018	Mickey Elliot	FTC Registration	200.00
9/16/2018	Aaron Busch	FTC Registration	400.00
9/16/2018	Marty Martin	FTC Registration	200.00
9/16/2018	Ledeana Miller	FTC Registration	200.00
10/19/2018	Mickey Elliot	Shirt Purchase	30.00
10/21/2018	Marty Martin	Shirt Purchase	30.00
10/26/2018	Dell	Cause Card Donation	150.00
10/26/2018	Dell	Employee Donation and Match	60.00
11/15/2018	Dell	Employee Donation and Match	25.00
2/19/2019	Parent Fundraiser	Donation	414.32
2/19/2019	Dell	Cause Card Donations	300.00
PENDING	Parent Fundraiser	Donation	70.00
PENDING	Boeing	Employee Volunteer Grant	100.00
		Total Income	2739.32

Expenses for the 2018/2019 Fiscal Year			
Date	Payee	For	Amount Out
9-6-18	AndyMark, Inc	Field Setup	512.22
9-9-18	AndyMark, Inc	Robot parts	163.71
9-16-18	Pitsco	Robot parts	427.69
9-16-18	Servocity	Robot parts	67.97
9-16-18	RevRobotics	Robot parts	45.25
10-1-18	Servocity	Robot parts	89.62
10-1-18	Home Depot	Tool Boxes	49.84
10-22-18	Servocity	Robot parts	28.69
10-30-18	RevRobotics	Robot parts	53.60
11-14-18	RevRobotics	Robot parts	107.83
11-14-18	AndyMark, Inc	Robot parts	395.98
11-15-18	Servocity	Robot parts	86.05
11-27-18	WalMart	Cart	32.63
11-27-18	Servocity	Robot parts	30.61
12-20-18	SWOSU	State registration	250.00
1-16-19	Servocity	Robot parts	41.08
1-17-19	AndyMark, Inc	Robot parts	152.89
1-22-19	Metal SuperMarket	Aluminum for wheelbase	53.15
1-28-19	RevRobotics	Robot parts	32.94
1-28-19	Servocity	Robot parts	65.27
2-14-19	Servocity	Robot parts	29.98
		Total Expenses	2717.00

Projected Expenses for Remainder of Fiscal Year 2018-2019	
Worlds registration	$2000.00
Robot Supplies	$400.00
Misc. Supplies	$100.00
T-Shirts and Swag	$200.00
Total Projected Expenses	$2,700.00

Projected Income for Remainder of Fiscal Year 2018-2019	
Cause Cards	$150.00
Dell Employees and Match	$80.00
Various Fundraising Efforts	$2,470
Total Projected Incomes	$2,700.00

To cover the expenses of going to Worlds, if we are fortunate enough to earn a spot, we are planning to fundraise for the remaining $2,470 for registration and robot parts. Travel expenses have not been included because team members and their families agreed to pay their own travel expenses.

We are currently working on a tier system to encourage future sponsors to donate so much for their logos/names on our shirts, banners, etc. We have identified potential sponsors and are working with parents to schedule time to meet with them to pitch our plans.

We also plan to work on other fundraisers and have been brainstorming ideas, such as selling our T-shirts, raffles, or working a Sonic fundraisers.

Sustainability Plan

Here are our strategies to maintain our revolution into the future!

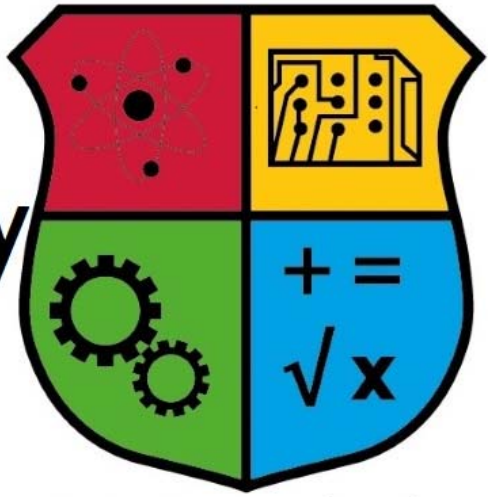
Strategy	How it works
Team Education	This year we implemented a team education system. This allowed us to not only educate all members of our team on key components of our team. These powerpoints that were created our found in our **Team Education Section**, and are also in our team drive. They are easy to access for current and future members.
Recruiting New Members	The Moore Community STEM Club has four teams they sponsor and they are associated with the Capitol Hill FRC Team. Having multiple teams from many levels, 2 Jr. FLL, 1 FLL, and our FTC team allow for us to have a steady influx of members. We won't be having any of the FLL students join us next year, as they will be using their last year of FLL as a crash course into Java programming. It was also a decision as a club to wait for the students to be 8th graders before joining our FTC team. See the **Outreach Pages 5-9** for more on how we worked with our younger teams.
Recruiting New Mentors	Some of our members' parents work in the technology field and spend time at our meetings providing assistance when they can. We were able to get our Boeing mentor when one of our Jr. FLL coaches got us into contact with Brandon McCabe, who provided us with great engineering help **See Engineering Notes Pages 20-20C.**
Fundraising	The team gets its finances primarily through our sponsors and registration fees. The sponsors are our most steady income and maintaining these sponsors is a part of our sustainability. However, we understand that we need to continue adding layers to our sponsorship and fundraising to maintain financial stability. See the financials above.

SWOT

Strengths	Weakness
● We have industry connections through mentors and coaches. ● Steady sponsorships and income ● The ability to educate our members on every component of our team.	● Some team members gravitate to the field that they are most experienced in instead of working to acquire new skills. ● It can be harder to recruit members and it is easy to feel isolated. ● We are not built into a previously established institution.
Opportunities	**Threats**
● Mentoring FLL and Jr. FLL ● Opportunity to gain new team members from diverse backgrounds ● Scholarship and internship opportunity through FIRST and mentors	● Losing a team member or coach ● Losing sponsorships ● Loss of build space/equipment

Our Sponsors Are:

Moore Community STEM Club



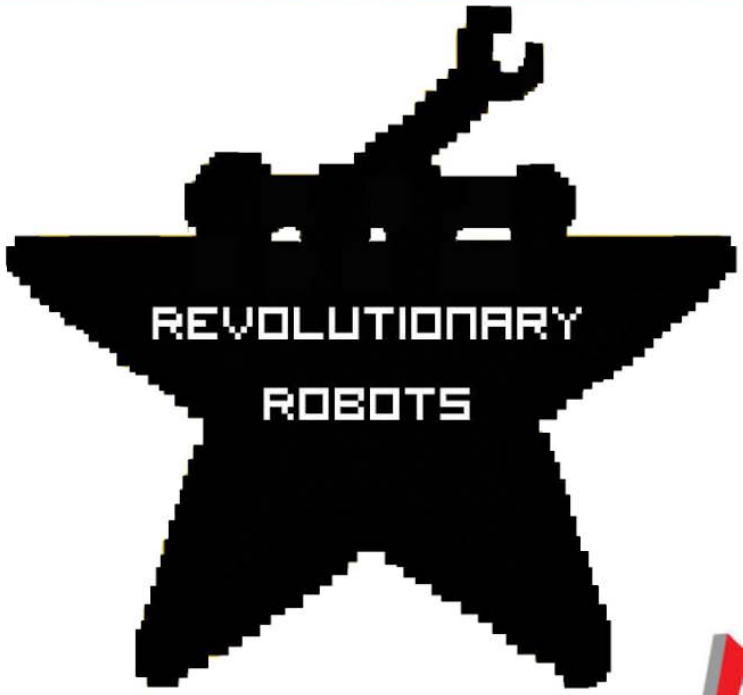
DELL EMC

Planet

BOEING

Revolutionary Robots 12535

Outreach 2018-2019



Moore Community STEM Club

FIRST®

Community Engagement

Tinkerfest

Goals for Outreach	Outcome
Talk to thousands of people about getting their kids involved in the *FIRST* community	On September 29, 2018, The Science Museum in Oklahoma City, Oklahoma held an event called Tinkerfest. The members of the Revolutionary Robots set up a booth and got to tell others about *FIRST* and the team. 10,000 people were there, several families were interested in *FIRST* and were able to learn more about not only FTC but all the programs *FIRST* offers. People of all ages were also very interested in driving around General Washington, our robot. Many kids got a chance to drive the robot around and soon after decided that they want to work with robots. The event was a great learning opportunity and a great one to get more kids interested in the *FIRST* program and science.

Community Engagement

Haunt Old Town

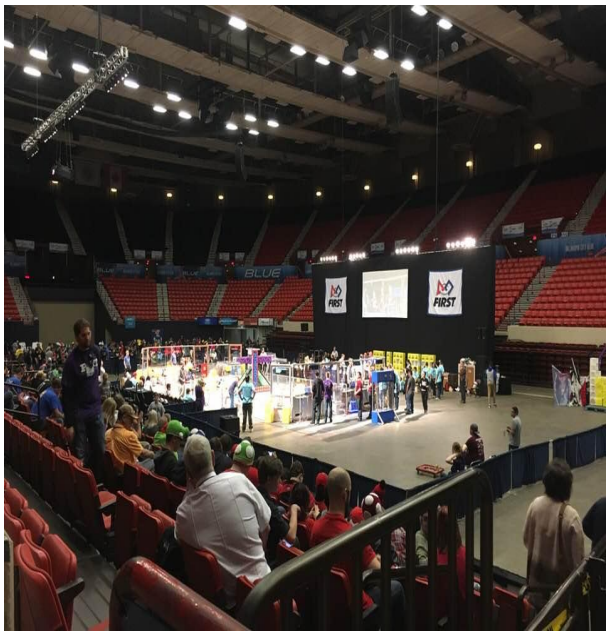
Goals for Outreach	Outcome
Share the word about *FIRST* by handing out candy in a trunk or treat community event	The team volunteered at the annual community event called Haunt Old Town. We gave away sixty-nine pounds of candy from inside a giant cardboard robot head made from cardboard, tin foil, tape, film, and a pull string bag. During this event, roughly 10,000 were there and we were able to talk to several thousand people about the first programs Jr. FLL, FLL, FTC, and FRC and how many age groups can be involved in *FIRST* It seemed as if we interested many children and their parents to possibly join the *FIRST* program.

Community Engagement

Volunteering at the Regional Food Bank

Community Engagement

Volunteering at FRC Regional Championship



Mentoring and Elevating

Dell FLL Scrimmage Day

Goals for Outreach	Outcome
Talk to people about getting involved with FTC by showing off our robot at an FLL scrimmage	The Revolutionary Robots were able to attend the Dell FLL scrimmage where our sister FLL team, the Electrons, competed in a scrimmage against other local FLL teams. While we were there we were able to work on our robot as well share the word about *FIRST.* tech challenge and other levels of *FIRST*. The Electrons competed very well at the scrimmage including finishing 2nd and coming close to scoring the most amount of points possible. This event lasted from 9:00 am to 4:00 pm and was on October 20th.

Mentoring and Elevating

Inviting FLL team members our events

Goals for Outreach	Outcome
Encourage interest in FTC	We encouraged any of our sister FLL team members to come see us at our events. We even had team members who hang with us at our first qualifying event this season.

Mentoring and Elevating

Mount St. Mary's FLL Qualifier

Goals for Outreach	Outcome
Volunteer for the Mount St. Mary's FLL Qualifier and represent the Revolutionary Robots along with *FIRST* Tech Challenge	On Saturday, November 17th the team volunteered at the Mount St. Mary's FLL qualifier. The team spent the day queuing teams, reset fields, and one member got to ref matches throughout the day. Our coaches also volunteered, they got to judge project presentations and core values for the FLL teams. Our sister FLL and Jr. FLL teams were at this qualifier and performed presentations for judges in their specific sections. It was a busy day for everyone involved, and any of our members can tell you they were having fun helping and scoring teams. We could also tell you most of us were missing our voices at the end of the day.

Mentoring and Elevating

Open House for FLL and Jr. FLL Teams

Goals for Outreach	Outcome
Speak with younger members of the club to ensure that they stick with their interest in the FIRST program.	The team held a potluck in order to show our FLL and Jr. FLL sister teams, Electrons, Mini-Electrons, Mini-Protons, what FTC is like. We showed both the parent and kids our notebook, robot, and programming while talking about our experience in FTC.

Mentoring and Elevating

Mentoring the Electrons

Goal	Outcome
Speak with the Electrons and assist with their secret missions	Olivia and Ivory went over to speak with the Electrons. The FLL team was working on the secret missions that FIRST had released. They had been working on building a box robot and ran into issues of it strafing and wobbling as it ran. This was due to their wires running on the ground, and them not having the tires they wanted. We were able to help them with their wires and pinning them up, and providing insight into programming.

Industry Connections

Tour of Dell/EMC Enterprise Lab

Goals for Outreach	Outcome
Learning more about our sponsor	We took a tour of Dell to gain a better understanding our our sponsor and what they do.

Industry Connections

Recruiting a Boeing mentor

Goals for Outreach	Outcome
Gaining more insight into a Engineering career	We recruited a Boeing engineer as a mentor this year, Brandon McCabe. Brandon has been great about providing us more insight into the engineering we do, as well as insight into engineering as a career.

Industry Connections

Sooner Rover Team

Goals for Outreach	Outcome
Learning more about what they do and the Engineering programs at OU	The Sooner Rover Team showed us what they do, how they do it, and *why* they do it. The robot was far larger than ours, but we were able to ask them how their actuator worked. They told us, and we told them about our idea for our actuator. They watched the Rover Ruckus challenge video, and as one of their members was a *FIRST* member and they gave us all their opinions on our ideas.

FTC Collaboration

Challenge Reveal

Goals for Outreach	Outcome
Work with other teams on strategy for the season	On Saturday, September 8, 2018, the team met at Mount St. Mary's for the FTC challenge reveal. We talked with other teams and viewed this year's field. The team also began to develop priorities and possible robot design. We discussed different strategies with other teams. The programming team decided to build a discord server to discuss and to help each other learn about Android Studio. The team discussed how the last year's robot design could influence this year's design. We also picked up a new team member at this event: Hello, Jackson Elliott!!

FTC Collaboration

Strategy Session

Goals for Outreach	Outcome
Have conversations with other teams about strategy for the season	Teams met at Mount St. Mary's for a strategy session on Saturday, September 15th. All the teams discussed possible strategies and shared prototype designs for the robot. The Rockettes, Makin' Stuff Move, Atomic Shock, the RoboCats, the Hornet Heroes, and the Senoritas of Awesomeness were all at the strategy session. The teams discussed ideal electronics, parts, sensor tips, and programming. The strategy session lasted from 9:00 A.M. to 4:00 P.M. Along with everything else in this time a prototype chassis was built along with a lift and an intake system. It was here that we learned about Onshape, which has helped us develop our CAD abilities. Shout out to Rockettes!!

FTC Collaboration

Practicing with the Atomic Gears

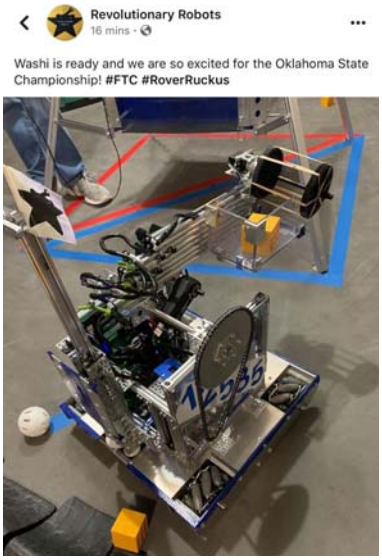
Goals for Outreach	Outcome
Practice driving and working with other teams	We worked out some bugs and shared programming tips. The wheelbase had screws on the inside between the wheel and the wall. It took us a solid fifteen minutes to get all the screws back in place with loc-tite, but it happened and we were able to get plenty of drive practice with Washi.
Spend time driving and working with the Gears	The robot had gotten a quick tune-up and we realized that we had left both of our phones, our battery, and our controllers. The Atomic Gears were kind enough to lend us some extra materials they had so that we could drive.

FTC Collaboration

Practicing With The Lightsabers

Goal	Outcome
Practice Driving with the Lightsabers	We practiced driving against the Lightsabers and got to know the team a little.
Compare and share ideas.	The Lightsabers showed us their notebook, how it is organized, and gave us ideas for improving our presentation. We shared our ideas with how we fund a community team through industry partnerships.

Social Media!



Revolutionary Robots: FTC 12535

YouTube

Scheduled Future Events

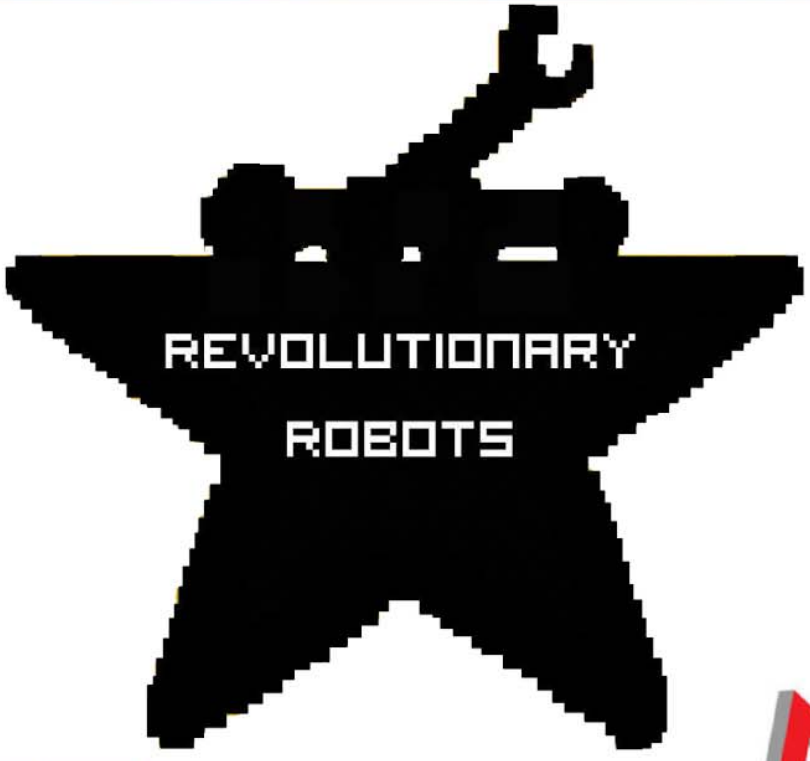
Event	Date
FRC State Competition	3/9/19
Food Drive	3/30/19
Volunteer at Food Bank	4/6/19
FAA	6/20/19

Planning Page for Future Events

Event Ideas to Work On	Status of Planning
Inviting the Jr. FLL team to share their World's presentation	Have spoken with their coach and will schedule with is as soon as they have something ready to share
Brick Fest	It is too late for us to sign up for a booth this year. Need to look into this in early January of 2020. Will leave on this page as a reminder.
Trash pickup/ Adopt-a-Street	Moore does adopt a park. Contact info is here: https://www.cityofmoore.com/departments/parks-recreation/parks-news-and-updates Oklahoma City's program info is found here: https://www.okc.gov/departments/public-works/resident-community-resources/adopt-a-city-street-program Norman's info is found here: http://www.normanok.gov/content/adopt-a-street
Ronald McDonald House	https://rmhc-okc.org/get-involved/volunteer/
Baptist Children's Home or other foster care S.T.E.A.M. activity	Have not followed up on this yet.
OKCPD	Team parent who is a police officer has suggested working with the police force to create a community engagement project. We need to follow up on the this to create a plan.
Plant Trees	We killed a lot of trees making the notebook.

Revolutionary Robots 12535

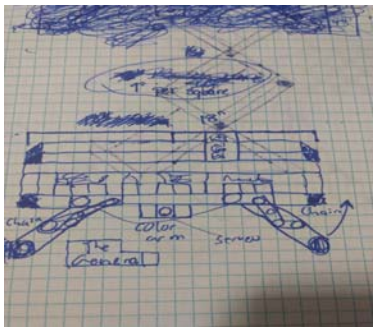
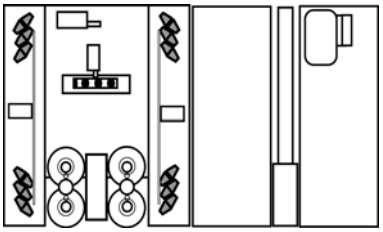
Engineering notes 2018-2019



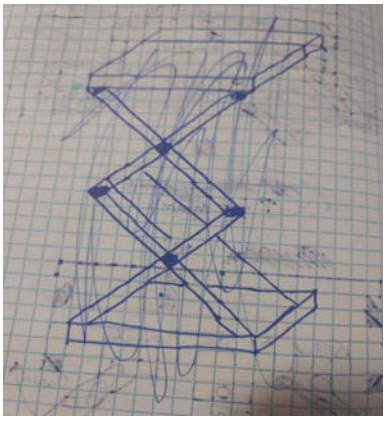
Moore Community STEM Club

FIRST

Sunday, September 9, 2018, 1-5 pm
Attending members: Drew, Jackson, Mark, Jackson, Ivory, Olivia, Andy, and Josiah

Goals For This Meeting	Progress on Goals	Photos
Design Ideas	A spider bot with movable legs and a conveyor belt to lift scoring elements. Hook underneath the conveyor, which folds up in three pieces to stay within the size limit. The design at the bottom was a rectangle design that would have an intake system on one side, and a hook/lifting system on the other, to balance out the weight and for conserving space. All the wiring and hardware would be beneath the platforms. The wheels seen there are mecanum. This prompted a discussion for buying mecanum wheels. The answer was a resounding yes, as it would give us the mobility we are looking for.	  
Robot Priorities	With the discussion of simplicity and point values, we decided having a hook and lifting mechanism would be the best thing for us to focus on with our robot. Another big point was being able to easily maneuver scoring elements in the center lander easily without having to constantly turn the robot around. This would save us time in the game and allow for more point earning during the tele-op period and autonomous.	

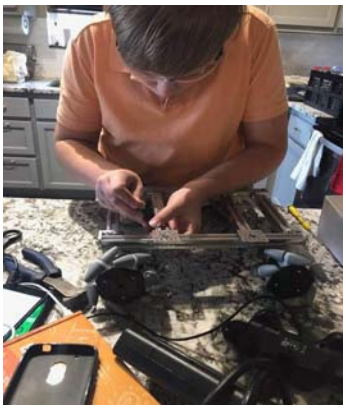
Con’t - September 9, 2018

Design Process	The design chosen was that of a lowrider car/spider bot combination and we began the building of the chassis. The intake system is undecided, but we would have a scissor lift for the platform the hook was on, and spider legs that were four inches long and would lift up off the ground to give us the clearance we needed.	

Mount St. Mary's Strategy Session
Saturday, September 15th, 2018, 9 am-4 pm
Attending Members: Ivory, Olivia, Mark, Andy, Josiah, Jackson E.

Goals For This Meeting	Progress on Goals	Photos
Finish Chassis Build	A challenge was put in place that anyone who could get a robot built together and moving would get to drive around on the Mount St. Mary field and participate in a scrimmage. We did not get a robot together in time, but the robot was put together by the end of the day. The mecanum wheels were a struggle to put on, as they were put together incorrectly the first time. We had to take one apart and find the error in order to attach the wheel. The robot did get a chance to drive, but the wheels did pop off. More progress needs to be made in the next meeting.	
Prototype for lifting mechanism and intake system	The lifting mechanism was rethought. We did not have the materials to make a scissor lift, and after some discussion with teams at the strat session, decided to work with a four-bar lift. This would lift up the platform that had our hook on it up to the ring on the lander. It would run off one motor and a 2:1 gear ratio for both speed and enough torque to get the robot off the ground and hold it in place. The intake system used gears and medical tubing. The gears would be attached to one or two motors that would spin. Upon approaching scoring elements the spinning tubing would suck them in.	 

Sunday, September 16th, 2018, 1-5pm
Attending members: Jackson.B, Drew, Mark, Olivia, Jackson.E, Ivory, Andy, and Josiah

Goals For Meeting	Progress on Goals	Photos
Get a moving robot	The base had to get a small redesign. The plate across the chassis was not as stable as we wished, so we temporarily removed the plate and put in some bars across the center for stability. The four-bar lift idea was also re-designed. We had some leftover linear slide pieces from our relic arm last year and thought it might be a good way to lift up our hook. It would be on a single spool, retracting the slide when it spun one way and expanding it when it spun the other way. This would be run on a motor and lift the robot off the ground. Coding just needs to be done to get the robot moving.	 

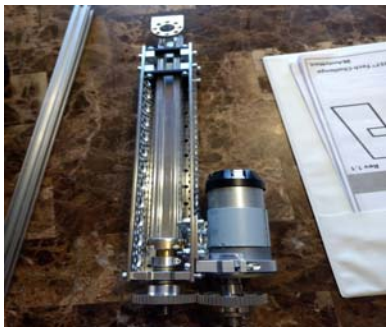
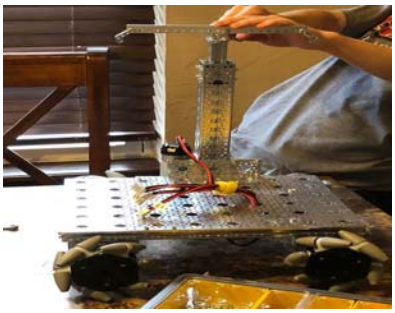
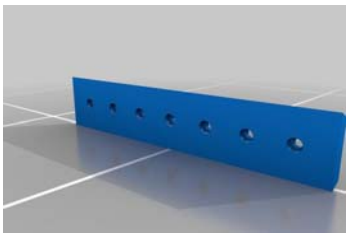
Sunday, September 23rd, 2018, 1-5pm
Attending members: Jackson B, Drew, Andy, Olivia, Ivory, Mark Jackson M, and Josiah

Goals For Meeting	Progress on Goals	Photos
Finish Chassis Design	After the stability added in the last meeting, we attached the base plate into place and felt much better about the stability. The REV Hub was attached, as well as the CADed battery holder from the year before. Designing for the linear slide attachment place began, but was not yet finished	
Put together the lift	The linear slides were put together, and it was discovered we only needed two of our five linear slides to reach the holding on the lander. We discovered you could not extend and retract on one spool and string set, so we strung up both sides with our kevlar string and put together both of our spools from the year before to set in place. They would attach to a single motor on a small platform so the spool would not drag against the base plate.	

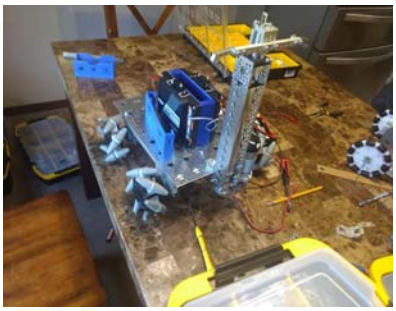
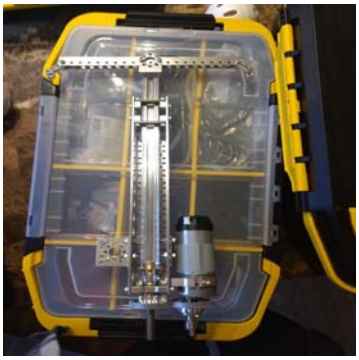
Sunday, September 30th, 2018 1-5pm
Attending members: Jackson B, Drew, Ivory, Mark, Olivia, and Andy

Goals For This Meeting	Progress on Goals	Photos
Tinkerfest Recap	While at Tinkerfest we did some *tinkering* with the robot. The supports we wanted to do for the linear slide were going to involve drilling new holes into the slide themselves, and without the proper tools to do so at the event, we simply removed the slide and let people drive around Washington as he was.	
Linear movement Decision for Slide or Actuator	As we began to talk about how to drill the holes necessary and getting the linear slide back onto the robot. A new idea was proposed. A member had seen a kit for a linear actuator we could put together. It included a worm gear system that could lift a robot that was 50 lbs six inches of the ground in six seconds. This was using a motor we were not allowed to use. A team vote was put into session and we decided to get the kit. It was confirmed to work, as we were unsure of how the slide was actually going to be able to hold the robot in place and be able to hold something within our weight constraint.	
Intake System Decision (What ideas do we have?)	The engineering and build team had a session of throwing intake system ideas back and forth. They looked through videos and found photos of intake systems they believed would work for our challenge this year. All three ideas were put up against a decision matrix **included below**. The highest scoring point idea was the item the team chose to focus on, as it had exactly what we wanted it to do.	

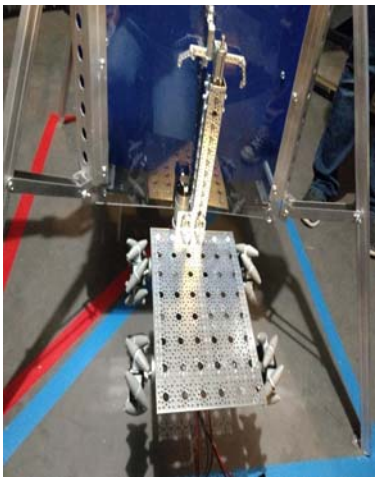
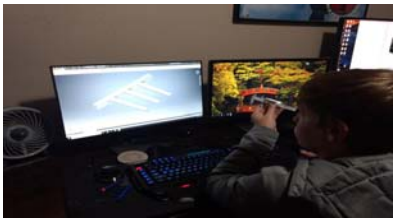
Sunday, October 7th, 2018 1-5pm
Attending Members: Drew, Jackson E, Andy, Josiah, Olivia, and Mark

Goals For Meeting	Progress on Goals	Pictures
Linear Actuator (Build and attach)	The pieces for the actuator came in! The team spent time putting it together. Initial testing with the Rev Hex motor showed it is a bit slower than what the website said, as they were using a Neverest motor, but still works for what we need it to do! We had to put a motor hub at the end of the gears to keep them in place, the gear collar provided did not fit onto the motor we are using. no other problems ran into. A bar and angled pieces were added onto the top of the actuator to make our hook for lifting.	 
Mineral Intake	The original prototype was taken apart as it was too big for what we wanted and could pick up three minerals. We also wanted to use foam darts instead of the original idea of medical tubing.	
CAD	A “REV sandwich” holder for a battery and two REV hubs began printing. This design was done by the Bridge Creek Robots. The blue piece attaches to the base of the robot. The holder then slides on and keeps the hubs in place.	 

Monday, October 8th, 2018 6-8pm
Attending members: Jackson B, Jackson E, Drew, Olivia, Ivory, Josiah, and Mark

Goals For Meeting	Progress on Goals	Photos
Base of Robot	The plate we had on the robot was dipping in the middle of it, so we temporarily removed it to add rails for stability. The two short rails we wanted to use had one that was shorter than the other, and a discussion ensued. It was a decision of buying new rails and hoping they came in before the next meeting or switch up the base to use REV rails. The new base is almost completely done, and made of REV rail. The plate and actuator were put back on.	
Linear Actuator	The hook was finished, and the linear actuator was attached to the new base. The original idea was to attach it to the side. With the new size of the robot, there isn't enough space between the wheels to get the actuator in place without the gears catching. Attaching it on the front has more space and would be better for holding the weight of the robot.	

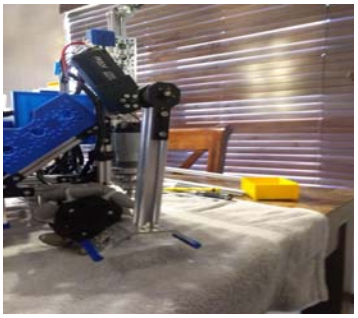
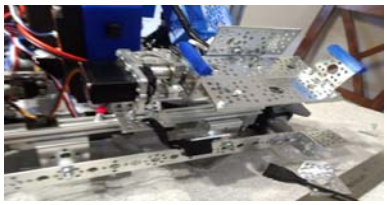
Sunday, October 14th, 2018 1-5pm
Attending members: Jackson B, Jackson E, Drew, Olivia, Mark, Ivory, Josiah, Andy

Goals For Meeting	Progress on Goals	Photos
Linear Actuator testing	The actuator itself made grinding noises as we began moving it up and down. The gears at the bottom of the actuator were misaligned and grinding against each other as they spun. While testing we also noticed the REV hub began blinking orange. Looking around we discovered our switch wires had completely disconnected from the connector, and we now have to order a new switch.	
CAD of the intake system	More progress was made on the CAD of the intake system. The project Drew is working on there is the mechanism that will attach to the spinning gears and hold the foam darts.	

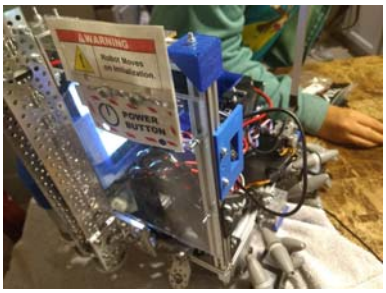
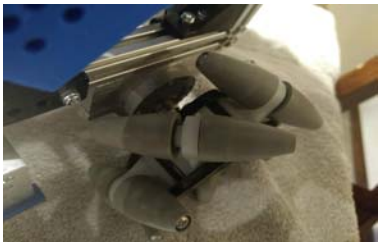
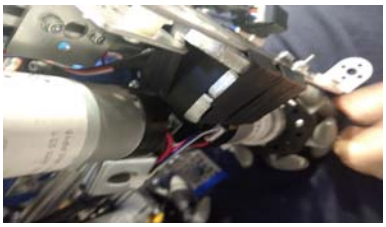
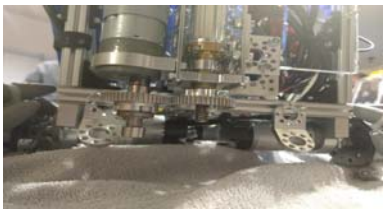
Friday, October 19, 2018
Attending members Drew, Jackson B, Jackson E, Ivory, Mark, Andy, Josiah, and Olivia

Goals For Meetings	Progress on Goals	Photo
Removing plating	The base plate was removed from the robot. This gave us the opportunity to attach anything we needed to directly to the rails. It also gave us a couple inches of space on the back to place our intake system within the 18” size constraint.	
Showcase!	At this Dell FLL Scrimmage, we were supporting our sister team, the Electrons. The Mt. St. Mary’s teams were also there at the scrimmage and we both got to showcase our robots for the room of FLL kids as well as their parents and the event volunteers. The best part of it was showing off the actuator and people getting to see Washington lift himself off the ground!	
Rails to support Actuator	While testing the actuator in the showcase we realized that it was not as stable as we wanted it to be We built up more rev rails to attach it to. They also will serve as a place to attach our intake system once it is built.	

Sunday, October 28, 2018 1-5pm
Attending members Drew, Jackson B, Jackson E, Ivory, Mark, Andy, and Olivia

Goals For Meeting	Progress on Goals	Photos
Attach color arm	The color arm uses the same bar from last year. A second one was added to the center of the back side. These will read two of the sample minerals at once, and the robot will react accordingly to what it sees.	
Attach “Golf Club” arm	This golf club arm is a temporary attachment. With time running out before the qualifier we wanted something that would be able to get the minerals out of the crater. This spins in a full 360 degrees and putts minerals out of the crater onto the field.	
Add bumpers on all sides	We had the issue of the minerals getting caught underneath the robot. This could cause us issues of controlling more than two minerals in the game. These bumpers provide a way to keep the minerals out from the underneath and causing issues as well as a way to push minerals around outside of the crater.	
Add a place for the marker to sit and be placed into the depot	The marker itself is 3D CADed and printed, except for the rubber duck at the top. The cage on the side holds the marker in place and is attached to a servo. The servo will tilt the plate to drop the marker into the depot during the autonomous.	 

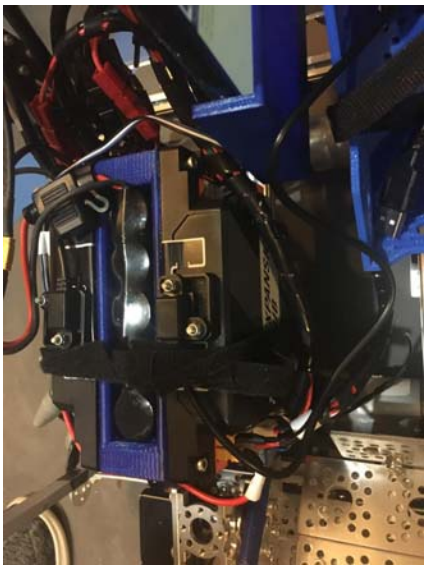
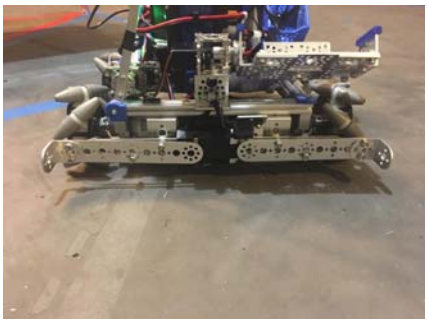
Sunday, November 4th, 2018 1-5pm
Attending Members: Drew, Jackson B, Jackson E, Mark, Ivory, Olivia, Andy, and Josiah.

Goals For Meeting	Progress on Goals	Photos
Power switch and labels	The new power switch to the robot. All labels were also added.	
Wheels falling off needs loc-tite	The set screws for the wheels became loose and the wheels would fall off. We spent some time removing the wheels and adding loc-tite to the set screws to keep the wheels in place.	
Wire management	The wires began to be organized. The have proper labels to show what wire set is for what part, and it began to be attached to the Lexan on the robot as well as velcroing it into place.	
Tipping Issue	We had trouble driving over the crater without tipping over so we attached weights with electrical tape and zip-ties to the rear bumper of our robot. The robot is now able to reach a 45-degree angle and not tip over.	
Motor/ Gear Height	We had a problem driving over the crater with our linear actuator hanging down that low, so we removed it to attach it two inches taller in order to give us more clearance over the crater	

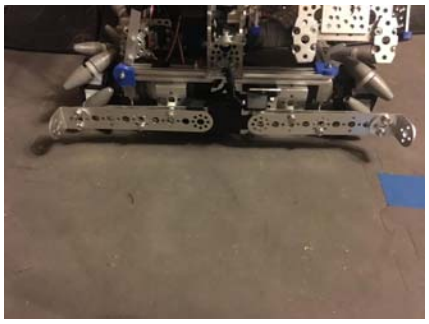
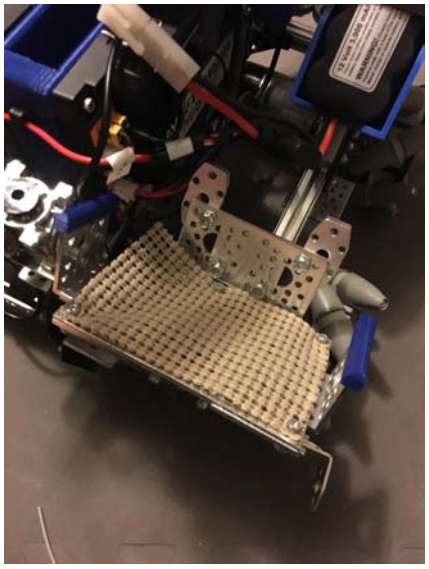
Con’t - November 4, 2018

Bumper Effectiveness	An initial test showed that both the gold and silver minerals could still get under the robot. The bumper on the back was dropped down. This will keep the minerals from coming under the robot from the back side, and widening it keeps the wheels from climbing over the minerals while moving backward. It needs to be widened out to keep the mecanum wheels from catching the minerals and scooping them under the robot.	
Battery Falling Out	With the robot tipping over, we noticed the REV battery we have in the hub holder is not secured in place as it is falling out. The idea is to velcro across the holder to keep in place, not yet implemented or tested.	

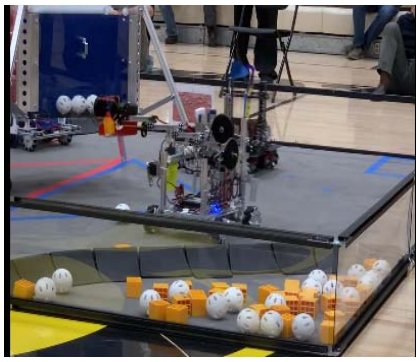
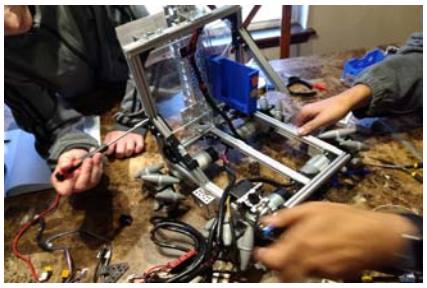
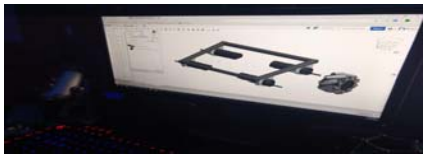
Monday, November 5th, 2018: 6-8pm
Attending Members: Drew, Jackson B, Jackson E, Olivia, Mark, Andy, Josiah, Ivory

Goals for Meeting	**Progress on Goals**	**Photos**
Secure rev battery	When we would drive over the crater to pick up minerals, our robot would tip then return to normal, but our rev battery would fall out so we strapped it in with several pieces of velcro. Now we have no problems.	
Lower and extend rear bumper	We still had problems with getting minerals stuck under the robot and getting in the way of our wheels, so we lowered the bumper almost a full inch as well as extending the bumper to cover the wheels	
Attach team number	The team number needs to be visible from two different sides, so we attached it to the top of our linear actuator bracing.	

Friday, November 9th, 2018 6-11: 30 pm
Attending Members: Ivory, Olivia, Mark, Jackson E., Josiah, Andy

Goals For Meeting	**Progress on Goals**	**Photos**
Bumpers for minerals	The bumpers on the back of the robot got widened out to block the back wheels. This allows us to not have to worry about any minerals getting caught on the back mecanum wheels and getting sucked under the robot.	
Marker Dumper	We noticed the marker would wiggle and jolt around as our robot drove during the autonomous. To fix this issue, we raised the walls along the side to keep it from falling out and put some cushioning there to give the marker a bit more grip.	
Robot Inspection	We passed!	

Sunday, November 18th, 2018 1-5pm
Attending members: Ivory, Olivia, Jackson B, Jackson E, Mark, Andy, Drew

Goals For Meeting	Progress on Goals	Photos
Intake System Ideas	After the Gordon Cooper Technology qualifier, we began to talk about how we wanted to update Washington for Newcastle. We took notice that the idea we wanted for our intake system may not work the way we wanted it to, as teams who had the same idea struggled with using the arm they built without almost knocking over their bot. We took time to do research on other intake systems that had been used before in previous years as well as ones that had been designed with the Rover Ruckus challenge in mind. A design by team #11301, the Mustang Gear Gang caught our eye. We already had most of the components we needed to start putting it together and got to see how well it performed and worked.	
Re-Evaluate Design	With the new intake system in mind, we began talking about how our robot's design will change. The innards of the robot will get a whole redesign, with the idea of making plenty of space for our linear slide mechanism to move around in mind.	
On-Shape CADing	New members began working on how to learn the new On-Shape CADing software we want to use to CAD our robot. Other members also began to the process of putting together the base in On-Shape and downloading the pieces needed to put together the linear actuator	 

After the Meeting, our robot looked like this……

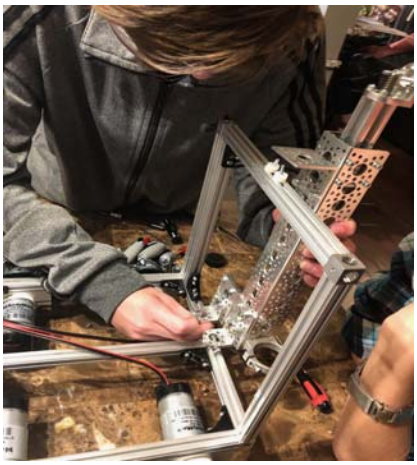
Monday, November 19, 2018 6-8pm

Goals For Meeting	Progress on Goals	Photos
Start reassembling the robot	The team reassembled a part of the base and attached the motors and wheels.	
On-shape CADing	Olivia imported parts of the linear actuator to be assembled in On-Shape, because the assembled Linear Actuator would not import into On-Shape.	For more pictures of our CAD design, see our Section devoted to Computer Aided Design

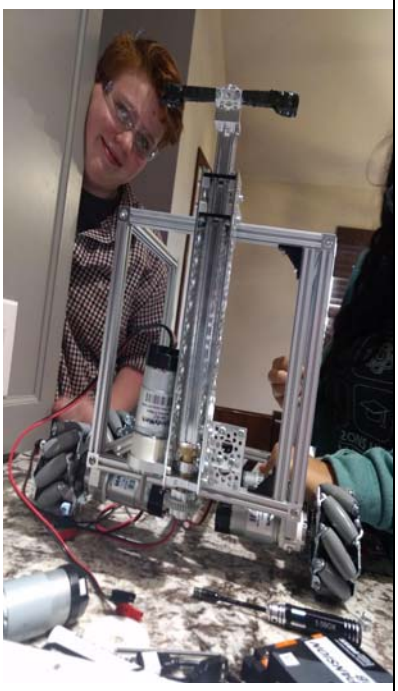
Tuesday, November 20, 2018 6-8pm

Goals For Meeting	Progress on Goals	Photos
Continue redesign and assembly	Jackson(Flapjack) and added the 15mm corner pieces to the robot to attach the rails to. Olivia disassembled the frame of the old actuator so that we could use it for the new frame. We need to reassemble or position the gears on the linear actuator still.	
Discuss structure and design with an engineering mentor	We informed him on our new base we were designing with CAD with OnShape. We also discussed how to wire our robot, because of it. He gave us an algorithm to find the center of gravity, so our robot doesn't tip.	
On-Shape CADing	Today Drew assembled the linear actuator in On-Shape so that we could put it on the robot. Mark started to assemble the intake in On-shape.	

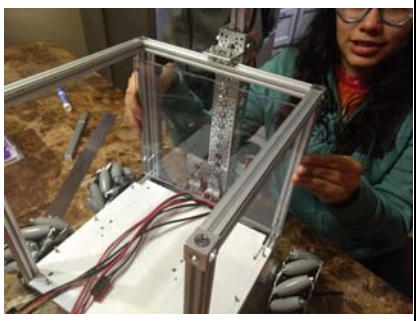
Monday, November 26, 2018 6-8pm

Goals For Meeting	Progress on Goals	Photos
Continue rebuild -Actuator attachment	The next big step in the rebuild of Washington was to attach the linear actuator. The only issue we ran into was the t-screws at the end of attachment. They were too short to go through the attachment places. We changed our screws to the hex screws, they were longer and what we had used the first time attaching it. We also began looking into changing gears so that the linear actuator gear and the motor gear don't lose connection.	
Tipping and Math!	Our mentor, Brandon McCabe who is a structural engineer at Boeing, gave us an equation to assist with finding the center of gravity on our robot. With the intake system in mind, he provided us with the knowledge to see if our robot would tip over at any point. All pieces of the robot were weighed, including the pieces of the intake system that we have, and were put into the equation $\Sigma m=0$. Here m=force, or in our case weight. There is also a variable we had to add, which was d. d=distance, and was used especially during the fact of when our intake system reaches into crater and lifts over the robot. In theory, unless we somehow add three pounds to the end of our robot, it should not tip over during the games!	Math work is on the next page!
On-Shape CADing	Drew organized files of our parts and Mark worked on arranging the linear slide to fit in the 18-inch sizing-cube.	

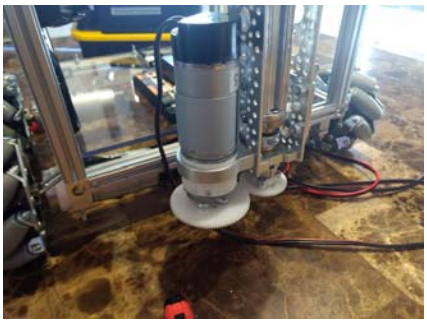
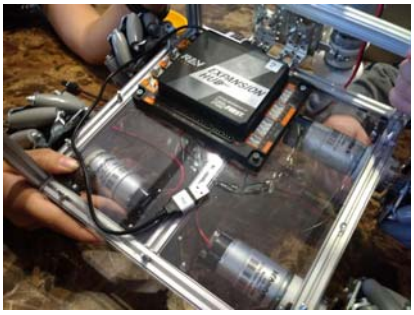
Sunday, December 2, 2018 1-6pm
Attending members: Olivia, Josiah, Ivory, Jackson B, Jackson E, Drew and Mark

Goals For Meeting	Progress on Goals	Photos
Finish attaching the linear actuator -Move it forward -Attach final gear	The actuator needs to be adjusted slightly in its placement. With the OnShape CAD of the robot and the current placement of our actuator, we would not be able to fit our plexiglass. It was moved forward a bit, and the plexiglass was able to be put in place without an issue. The gears purchased for the actuator have not yet been attached or tested. They would change the gear ratio from a 1:1 to a .6:1 ratio. Changing the actuator motor to a NeveRest 20 motor, from a NeveRest 60 motors. We believe that this decrease in gears will increase our speed considerably. We are pretty confident the linear actuator design will be compatible with the lower geared motor. We are hoping that this will give us the extra "vroom" we need.	
Assemble linear slide	Flapjack began the assembly of the linear slide, it was not yet attached to the robot. The motor was attached to the slide, a new spool is going to designed and printed for this design, and finalizing for placement on it needs to be done. The motor is currently an HD-Hex motor, but with the positioning, we want the motor is sticking out of the frame. The idea is to switch over to the REV Core Hex motor in the next meeting.	

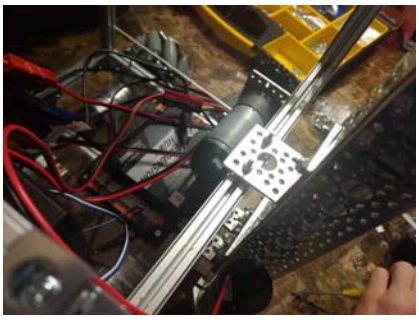
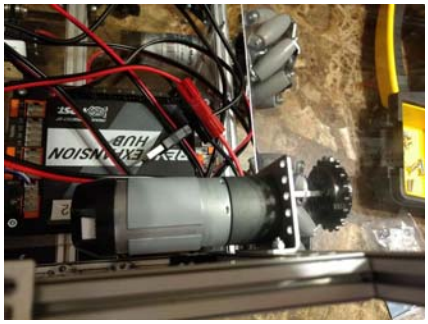
Monday, December 3rd, 2018 6-8pm
Attending Members: Oliva, Josiah, Andy, Jackson E, Drew, Ivory, and Mark

Goals For Meeting	Progress on Goals	Photos
Adjust and Tighten the Wheels	Noticed there was camber (sideways tilt) on wheels, discovered a screw was missing that caused camber and added a screw to fix the problem. Motors are no longer angled.	
Attach more Acrylic	We ran into measurement problems with the acrylic. The side panels had to cut down on the top right corner 16mm by 90mm to fit within the rev rails. We marked the corner that needed to be cut.	
Gears for Actuator	New gears were put onto the actuator today. We are a bit worried that with how thin the gears are, and if they will come apart if the actuator is accidentally moved too far, needs to be tested. It is also thought that we need a spacer between the actuator and the motor driving it, to keep the pieces from scraping.	

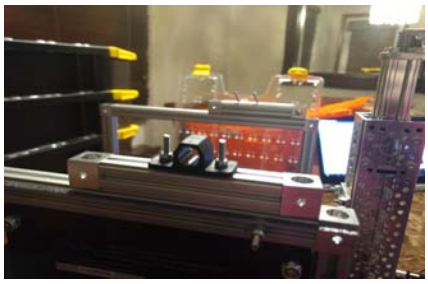
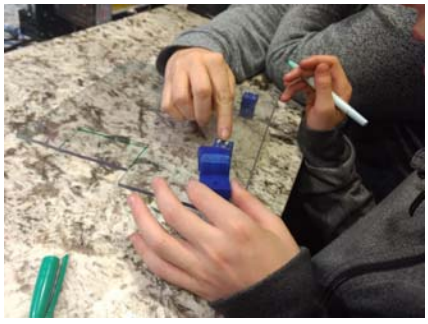
Sunday, December 9th, 2018 1-4pm
Attending Members: Ivory, Jackson E, Jackson B, Drew, Olivia, Josiah, and Mark.

Goals For Meeting	Progress on Goals	Photos
Gearing up actuator	Today we created a wedge to keep the motor stable, so the linear actuator gears will stay lined up. We had been having a problem with the gears slipping a little.	
Attaching Expansion Hubs	We attached the Hubs to the acrylic bottom plate and the right acrylic side plate so not to mess with the linear slide on both sides. Both hubs were attached, one at the bottom and the other on the side.	
Acrylic side plate additions	We attached the acrylic sides to the outsides of the robot to move the Hub inward so not to mess with the linear slide mobility. Holes were drilled in the side panels for the hubs.	

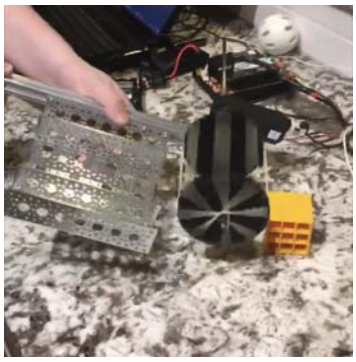
Sunday. December 16th, 2018 1-5pm
Attending Members: Olivia, Ivory, Drew, Jackson B, Jackson E, Mark, Andy, Josiah

Goals For Meeting	Progress on Goals	Photos
Add crossbar to the front of the robot	We put the crossbar on the robot. We did this for the sidebars and the stability of the LA. The piece has two screws that go into the sidebars, and when twisting one way you would tighten one screw but not the other. Instead, corner braces were put to attach it.	
Attach Actuator	We attached the LA to the crossbar. So the LA would not come off the robot when lift. We fixed the big gear on the robot. The cause of this action is one screw was too long and when we tightened the screw it would get stuck on the motor. We changed that screw to a shorter screw and added nuts on the ends of all the screws.	
Build servo block for marker dispenser	The servo on the CAD was a REV servo and had to be changed because it would not connect to the little circular metal piece seen in the photo. We switched to the servo shown and attached to a REV servo block to be attached to the robot.	
Attach Motor for the beginning of the linear slide	We attached the Hex Motor to the frame. We had to drill a hole in the plexiglass in order to have the shaft go through. This motor flips the arm that scores to the lander.	

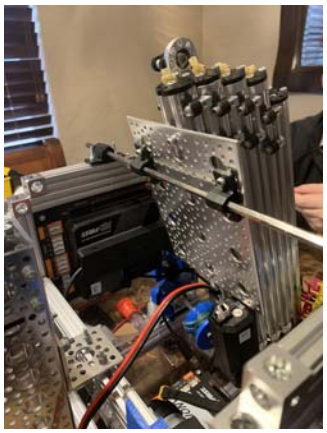
Monday, December 17th, 2018 6-8pm
Attending Members: Jackson E, Andy, Ivory, Drew, Mark, and Josiah.

Goals For Meeting	Progress on Goals	Photos
Attach 3-inch rails for the linear slide bar.	The three-inch rails hold a bar that will be supporting the linear slide. No issues putting them on. The rails themselves were cut to size by FlapJack's mom. (THANK YOU!)	
Tighten wheel screws.	The wheels and base were wobbly when driving around the robot, and this could cause a possibility of it not driving straight or the wheels falling off. These were tightened up and the wheels are much more stable.	
Add a bar to hold marker on the marker dispenser servo	The bar is to hold the new marker being designed. It will wrap around this bar, and when the servo spins it to a downward facing position the marker will slip of into the alliance depot.	
Drill holes for the flag holder	The flag holder come in two pieces and needed to be put onto the robot now rather than later. The plate it is meant to be on was removed and holes were marked to be drilled.	

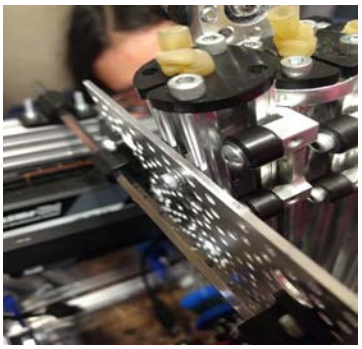
Monday, December 18th - 19th
Attending Members: Jackson E, Mark, Drew

Goals For Meeting	Progress on Goals	Photos
Creating a prototype for the intake	We made multiple intakes for the linear slide. One is a rubber band intake. One has lego wheels. The last is one with a bottle brush. There are multiple so we can switch out between matches and switch if one breaks.	 
Attaching chain	We cut the chain for the sprockets. We attached it to the two sprockets, so we can lift out the slide.1	

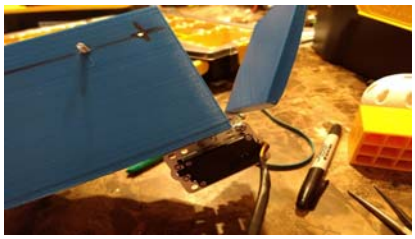
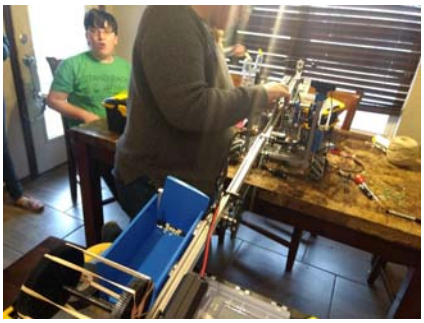
Wednesday, December 26th
Attending Members: Jackson E, Mark, Drew

Goals For Meeting	Progress on Goals	Photos
Attach the intake box	We drilled holes to attach the mineral box to the linear slide. We put spacers between the box so it wouldn't hit the slide.	
Adding a plate to the slide	We were worried about the little plastic pillow that attaches the axle to the slide braking, so we attached a plate to two of them to spread out the weight.	

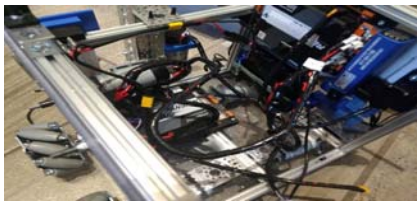
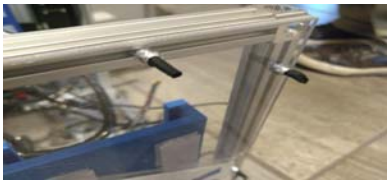
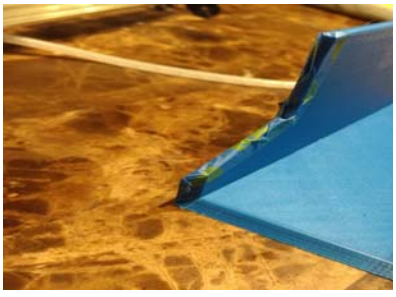
Saturday, December 29th
Attending Members: Jackson E, Mark, Andy

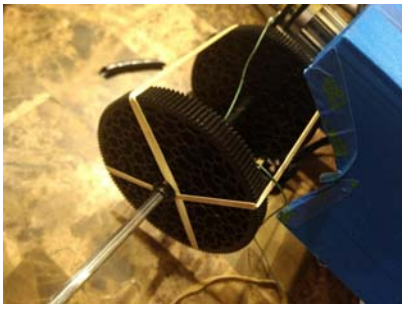
Goals For Meeting	Progress on Goals	Photos
Stabilize the slide	We attached washers to the slide so it won't wobble as much. The slide was not as stable as we wanted it to be, the washers stabilized the bottom two slides, but the other two still wobble. Tightening the screws should help with this problem.	
Stabilize motors	We had problems with the motors moving so we put plates underneath so they would not move as much. We also attached zip ties to keep the motors even more stable.	
Attach camera	The camera needed to be attached for our autonomous. But we needed a way to attach it. First, we tried Velcro but it wasn't strong enough. Next, we tried zip ties but it slid over. Finally, Mark 3D printed a piece to attach to the rev rail.	

Sunday, December 30th, 2018 1-5PM
Attending Members: Mark, Drew, Olivia, Ivory, Jackson B, Jackson E, Josiah

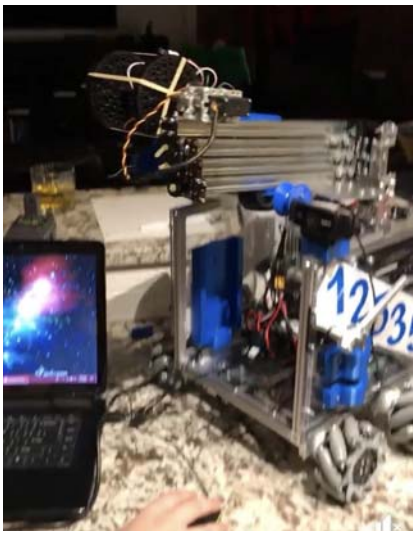
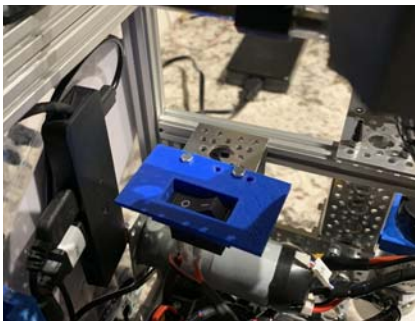
Goals For Meeting	Progress on Goals	Photos
Attach the back of the bucket to the rest of the bucket	We used a servo block to attach the back of the bucket to a beam which allows the minerals to slide out the back of the box. We had to disassemble the servo block in order to get the screw in the drilled slot in the bucket.	
Reattach Camera	We successfully reattached the camera after changing the position of the screw that attached the camera to the mount.	
Linear Slide Work	The linear slide was getting caught on the pulley so we inserted a screw to stop it from sliding. We did wire management by wrapping and attach clips to stabilize it.	

Monday, December 31st, 2018 1-7pm
Attending Members: Mark, Olivia, Jackson E, Ivory

Goals For Meeting	Progress on Goals	Photos
Wire Management	We spent a large portion of the meeting covering the wires in tape and plastic tubing to prevent ESD and protect the wires from getting caught in anything.	
Safety Tape	The screwheads sticking out of the robot were also wrapped in tape to protect our teammates and be in compliance with the safety rules.	
Backdoor for the mineral box	The backdoor for the mineral box was warped and too thick when it was printed. A new one was printed.	
Marker dropper	The team marker we have designed has a loop that will hand on the bar. The stoppers shown in the photo were added to prevent the marker from going back too far and getting stuck on the bot.	
Slide weight	The linear slide was way too heavy for our sprocket and chain system and almost broke the shaft, so to fix it we took out one stage of our slide as well as changed the Core Hex motor to a continuous rotation servo. This reduced the weight of the slide.	
Move intake box forward	In order to keep the minerals in the box we had to move the box forward and cut a 2x2 piece out of the box. It allowed us to put the intake closer to the box for better ability to get the minerals in place	

Intake	Instead of having four rubber bands on the intake, we decided to move down to two. It allows for a better grip and suck of the minerals.	
Driving	With the competition being so close we still have not been able to practice driving the robot. It is crucial that the drivers spend time practicing so that we are ready for the competition.	

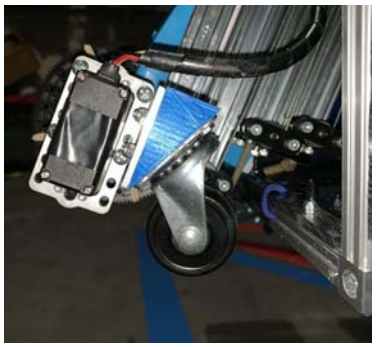
Tuesday, January 1, 2019
Attending members: Mark

Goals For Meeting	Progress on Goals	Photos
Tested motor/gears on linear slide	I hooked up the REV Expansion Hub the the Hub Interface on the computer. This helped make sure that the torque was enough to raise the linear slide.	
Moved switch	We were afraid that the wires that ran to the switch might get bumped, so we moved to the inside of the robot. It also helped the wires from the switch reach the hub.	

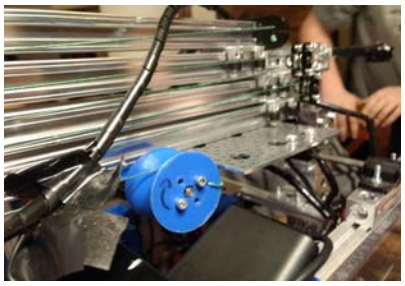
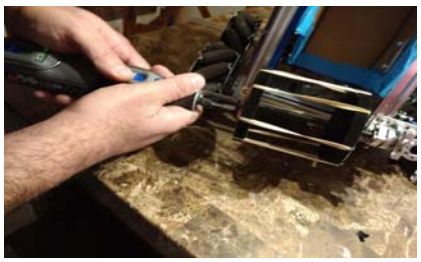
Wednesday, January 2, 2019
Attending members: Drew, Jackson E, Mark

Goals For Meeting	Progress on Goals	Photos
Wire Management on Arm	We wrapped the servo wire on the end of the arm to the Hub with wire wrap. At the end of each section of arm we put wire clamps to secure the servo wire.	
Add second Rev rail	We added a second Rev rail to stabilize the motor that rotates the arm. Also we 3-D printed a motor mount to help it as well.	
Wire management	We wrapped all the wires in wire wrap and attempted to put the wires out of the way and zip-tie and clip the wires down so they couldn't move around.	

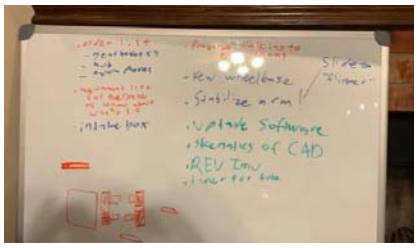
Thursday, January 3, 2019
Attending members: Drew, Jackson E, Mark

Goals For Meeting	Progress on Goals	Photos
Added castor wheel	To make our linear slide work better while collecting minerals, we added a castor wheel to the servo plate. With the castor wheel, our arm rolls along the ground instead of dragging against the minerals	
Added guard for linear actuator gears	When we would drive up against the crater, our linear actuator gears would smash up against the crater, which we thought would cause problems in the future. So we made a guard that attached to the rev rail in the front. We then attached two L brackets to come up and around the two gears on the Linear actuator.	
Made temporary box for linear slide	Our 3D printed box broke and while we were printing another one we made one out of cardboard by tracing the four sides of the box, then cutting the cardboard with scissors. The box worked as well as our regular one	
Changed camera mount	We had to change the camera mount so the camera is facing the three sampling minerals and not the ones in the crater. Now the mount is at a incline instead of straight.	

Friday, January 4, 2019
Attending members: Drew, Mark, Jackson B, Olivia, Jackson E, Ivory, Andy, Josiah

Goals For Meeting	Progress on Goals	Photos
Linear Slide	The screws of the slide fell out, causing a major issue with wobble. The screws were replaced to rebalance the slide. The surgical tubing in the top slide had fallen through, and before we could do official testing the tubing had to be pulled back through.	
Bar Axle	We cut the bar off for it was to long. It would pass the 18'' mark. This bar was for the spinning mechanism at the end of our linear slide.	
Mineral Guard	In autonomous we had problems with the minerals not moving enough, so we added on a bumper to make mineral movement more consistent.	

Sunday, January 6th, 2019 3-5
Attending Members: Andy, Jackson E, Drew, Jackson B, Olivia, Mark, Ivory

Goals for Meeting	Progress on Goals	Photos
Recap competition	The team gathered in the living room of our base of operations and went over all the thing that had happened the day before at the Newcastle qualifier. While the robot few technical and connectivity issues, all in all we had a good day.	
New robot fixes	It was decided that we wanted a more sturdy wheel base to prevent camber, or the wheels tilting. The linear slide would also need to be either changed or rebuilt with loctite to make it more sturdy and usable.	

Sunday, January 13th, 2019 1-5
Attending Members: Josiah, Andy, Drew, Olivia, Ivory, Jackson E, Jackson B, Mark

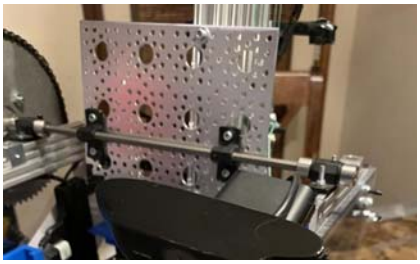
Goals For Meeting	Progress On Goals	Photos
Prototype Rev Slide	We used the instructions from the REV Website to create a prototype for a possible alternative to our linear slide. In the end, we decided the alternative was not an improvement to our existing design, if we make some minor changes such as, replacing screws and using loctite.	
Aluminum sheet metal sizing	We calculated the area of the plates for the wheel base we designed was about 4 ft^2 . The closest size plate we could find was a 2x2 ft plate, being about $63 from MetalsDepot before shipping..	Math and diagram on next page
Tensile Strength for slide bar	We looked up the tensile strength of stainless steel compared to carbonized steel. We currently use stainless steel for our shaft, but we are switching to carbonized steel in order to reduce the flex in our shaft. ***Also we learned that our shaft could not exceed 980 degrees celsius, or it would become very brittle. We will have to be careful!***	Math on next page

Monday, January 14th, 2018 6-8
Attending members: Andy, Mark, Jackson B, Jackson E, Ivory, Josiah, Olivia

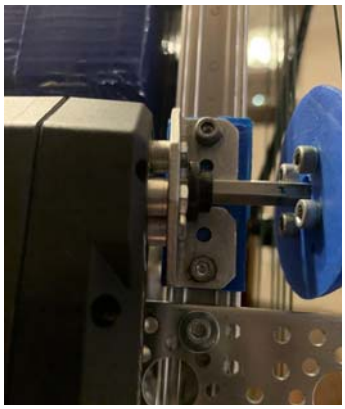
Goals For Meeting	Progress on Goals	Photo
Linear Slide	The linear slide was wobbly and not as stable as it could be. We believe this is because we had some screws that keep loosening up as we used it. To solve this, we loc-tited and tightened each screw that was loose.	
Config tutorial	Mark made a presentation on how to configure the robot. Not everyone on the team knew how to do it, and we felt like we should all know how. We have decided to continue these presentation to make sure knowledge is shared between team members..	Find presentation in the **Team Education Section**

Sunday, January 20th, 2019 1-5

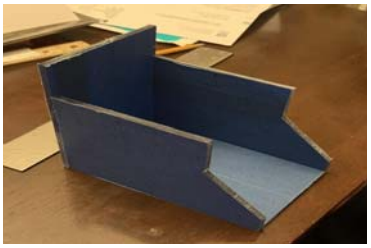
Attending members: Andy, Mark, Jackson B, Jackson E, Ivory, Josiah, Olivia

Goals For Meeting	Progress on Goals	Photos
Linear slide axle	Today, we replaced the aluminium axle on the robot with a carbonized steel axle because the aluminium axle was flexing. It does seem to be more stable. We need to test after reassembling the rest of the parts.	
Change out sprockets	We replaced the sprockets attached to the linear slide to increase the gear ratio so that it would move slower so we can have more control over our linear slide when it is in motion and keep the weight from overcoming the torque. This should also help the electrical issues we were having at competition that we believe were being caused by the impact to the frame when the slide would hit the bottom.	
Finalized linear slide reassembly	When slide was reassembled, it is much more stable now but it seemed to be catching and not returning to its fully retracted position. We realized that one of the four screws on the end pieces was rubbing the end in the level above. We decided that those screws were redundant and removed them.	

Monday, January 21th, 2019 6-8
Attending Members: Olivia, Drew, Jackson E, Josiah, Mark, Andy

Goals For Meeting	Progress on Goals	Photos
Stabilize intake motor	Before we put the liner slide on Washy, we had to add another screw to the Core Hex Motor to stabilize it. So after that we attached the linear slide to the mount.	
Wheel base	Flapjack and Mark took a field trip to the Metal Supermarket to get the metal plate that will be cut for the base. We got an aluminum sheet of 5052 and aluminum angle. We can now send it out to be cut!	
Linear actuator	The only change we made to the linear actuator was moving the a collar down a few millimeters. It appeared this had loosened up through use at the competition. We should add this to the list of things to check between rounds at competition.	

Wednesday, January 23, 2019
Attending Members, Mark and Jackson E

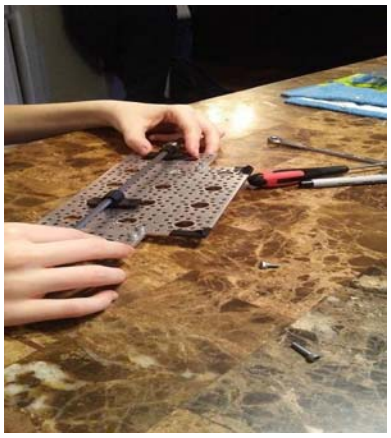
Goals For Meeting	Progress on Goals	Photos
Attaching New Camera Mount	Prior to the Newcastle qualifier, we added a last minute camera mount to help aim the camera at the minerals in the autonomous period, We need a more stable solution, so we CADed a better design. It is angled to see the minerals on the field and designed to work with T-screws.	
Mineral Box	We got the mineral box cut and soon realized that the dimensions were just a little off. We changed the dimensions and back cutting board.	

Saturday, January 26, 2019
Attending Member, Mark

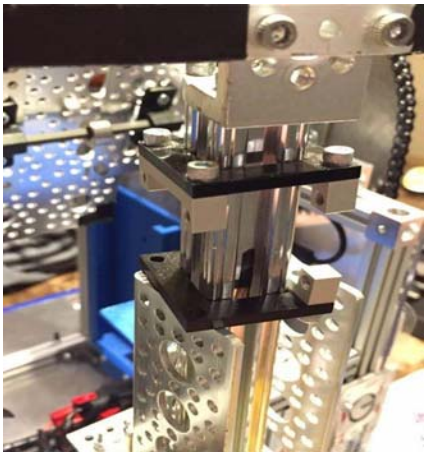
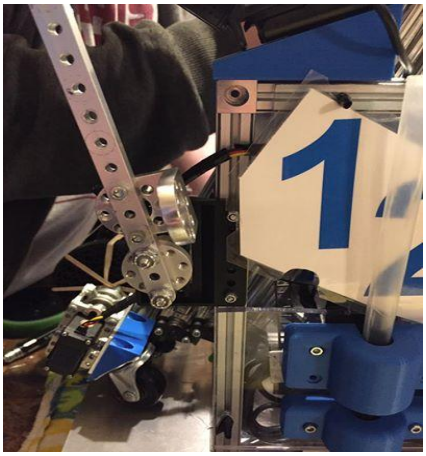
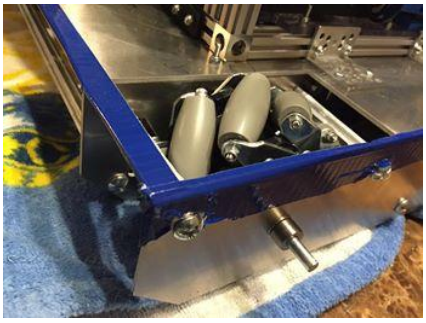
Goals For Meeting	Progress on Goals	Photos
Creating Brackets	We got angle iron when we went to the metal supermarket so we could make our own brackets. We cut two types of angle brackets. One type is for the mineral box and the other is for the top plate.	
Assembling the wheel base	While assembling the wheelbase we ran into a couple problems.The axles did not go into the bearings smoothly. We had to take a mallet and hammer the bearings on. We did not have a plan for wire management, so we cut a hole in the plate for wires. Our base was a little too small but we got it to fit together.	  
Assembling the mineral box	We decided to make the intake box out of acrylic for a more permanent solution than our cardboard box. This also allows us to be able to see through the box when grabbing minerals. After cutting the acrylic, we put it together with custom brackets based off of our CAD design.	

Sunday January 27, 2019

Attending members, Andy, Mark, Drew, Jackson B, Jackson E, Ivory, Josiah, Olivia

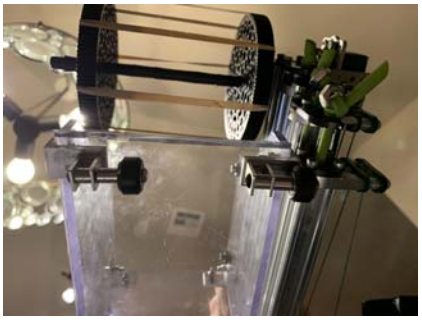
Goals For Meeting	Progress on Goals	Photos
Attach wheel base	We attached the new wheel base to the robot core. This design seems to fix the problem we were having with the wheel cambor. We realized that the way our motor were attached previously all of the sideways torque n the wheels whas being carried by some small REV screws on the REV rails.	
Fix bar support holding linear slide	Today, Ivory and Olivia built a small bar to help stabilize the linear slide axle. Even though we changed the axle to a more durable metal, we were still experiencing enough flex to move our sprockets allowing there to be slack in our chain. To solve this problem, we moved the pillows closer to the sides and not in the middle. If this does not work we will be looking into buying a one-inch pipe to replace the axle.	
Tighten Linear Actuator	The linear actuator need to be moved to allow it to be properly placed with the new wheel base surrounding the gears. This allowed us remove the guard previously in place.	

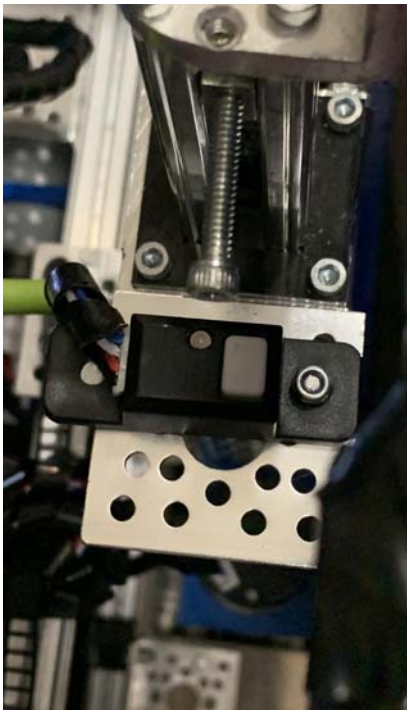
Monday, January 28th, 2019 6-8pm
Attending Members: Ivory, Drew, Mark, Josiah, Jackson E, Andy, Josiah, Olivia

Goals For Meeting	Progress on Goals	Photos
Replace screws on actuator	The bottom black plate, or X-Rail slide plate came loose from the silver bracket, as we lost screws while on the field. We had to remove the entire X-Rail on the actuator. We then assembled everything, but while we were putting a screw in the head broke off. We had to get rid of the screw by drilling it out.	
Move player marker	With the change in the wheelbase, we became concerned that the marker would land on the wheelbase and not on the field. To prevent this, we moved the servo from inside the rev rail to outside. Now we can drop the marker without it touching the base.	
Tape sharp edges	To prevent the sharp edges on our base plate from harming us or the field, we taped the edges with duct tape.	

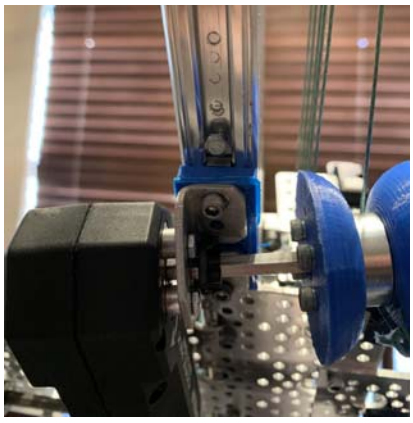
Strengthen Large Sprocket	We have been working to strengthen up our entire chain and sprocket assembly, to prevent any flexing that would allow slack in the chain. Today, we still observed some flexing when we would reach the apex of our rotation, so we put a wheel conversion plate on both sides of our large sprocket, not just one side which has drastically decreased the flex of the sprocket.	

Wednesday, January 30, 2019
Attending Members: Jackson Elliott, Mark Hazen

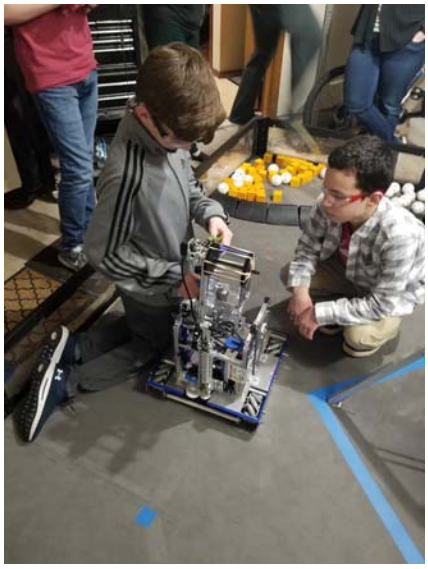
Goals For Meeting	Progress on Goals	Photos
New Roller Wheels	The caster wheel worked great with the old box, but not with the new box so we attached some of the small V-wheels to the box and they work much better than the previous one.	
Tighten Chain	To fit within the 18x18 sizing cube, we had to move the arm. So we had to readjust the chain to fit the sprockets.	
Putting Copper Filament in the Couplers	We put copper filament in the couplers to keep the set screws from working their way out. While running the robot, the screws loosened up quickly.	

Stop For Actuator	We wanted a stop switch for the downwards movement of the actuator, so we won't strip the gears when retracting.	

Sunday, February 3, 2019
Attending members: Ivory, Drew, Mark, Josiah, Jackson E, Andy, Josiah, Olivia

Goals for the Meeting	**Progress on Goals**	**Photos**
Loctite Motor Spacers	Today we applied loctite to the spacers on the motor mount to make sure the wheels don't come apart easily. We have already had three screws come off in two minutes while driving around, and the loctite should keep the screws in place. Testing required.	
Add touch sensor on linear slide	To prevent our linear slide from sliding too far and putting excess stress on our motor. We added a touch sensor attached to a bracket that when the slide would reach its extension limit it would slide into another bracket causing the sensor to be activated, thus making the motor slide stop extending.	
Add stop on linear slide for motor	When our linear slide was extended to its max, the motor would keep pulling and forcing the motor down the slide. This made lifting it difficult due to weight so we drilled a hole in the slide right next to the motor, we then put a screw in the hole so now the motor cannot move from its desired location.	

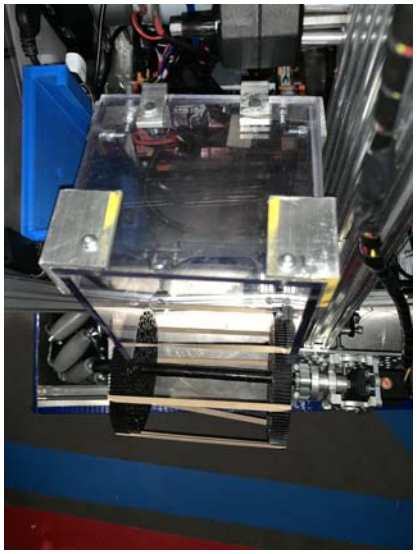
Monday February 4, 2019
Attending Members:

Goals for the Meeting	Progress on Goals	Photos
Removed the wheels from the minerals box	We originally had small wheels on the bottom side of the bucket to keep the edges of the box from harming the field, but the wheels had two problems. First, the wheels did not come in contact with the ground because the brackets attaching them were too large and hit the ground before the wheels. Second, the wheels raised the bucket too far off the ground for our intake system to work, so we replaced the wheels with duct tape along the front edge of the bucket. The duct tape now keeps the field from being harmed when we use the linear arm.	
Prepared the robot for practicing tomorrow (2/5/19)	We were used this time to identify what screws were loosening to loctite them and test out our robot's functions.	

Tuesday February 5, 2019
Attending Members: Mark Hazen and Jackson Busch

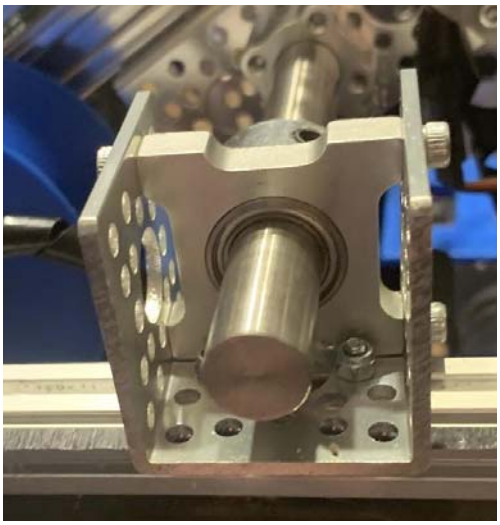
Goals for the Meeting	**Progress on Goals**	**Photos**
Practice at Gordon Cooper with Atomic Gears	While practicing the screws and/or the set screws would fall out of the wheel base. The set screws were from the coupler and the screws were attaching the motor mounts. We need to keep this on or radar and come up with a better solution to keep these couplers tightened for state.	

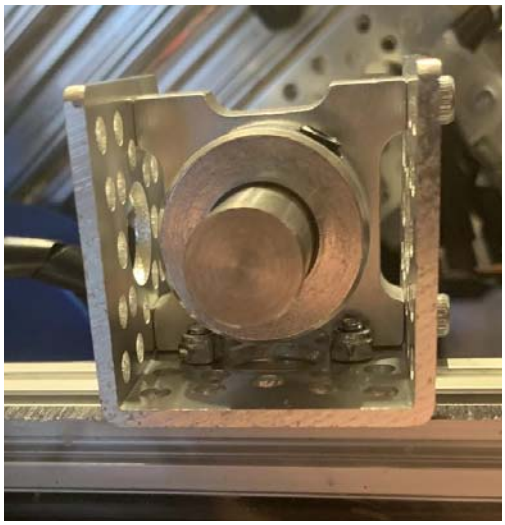
Thursday February 7, 2019
Attending Members: Mark Hazen and Jackson Busch

Goals for the Meeting	**Progress on Goals**	**Photos**
Make Bigger Brackets	The box was catching on the lander while practicing, so we made wider brackets to keep it from catching and used button head screws. The holes might have to be countersunk for the screws if it continues to catch on the lander.	

Friday February 8, 2019
Attending Members: Mark Hazen

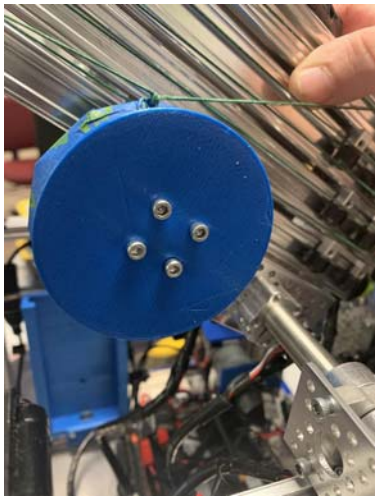
Goals for the Meeting	**Progress on Goals**	**Photos**
Attach New Shaft	We got a new shaft so the big sprocket remains stable while running. The 5mm carbonized steel hex axle did not have enough tensile strength to overcome the torque created by lifting the linear slide while extended. The diameter of the new shaft is ½ inch and it is connected with pillow blocks that are on channels which connect to the REV Rails.	





Sunday, February 10, 2019

Attending Members: Drew Busch, Jackson Busch, Jackson Elliott, Mark Hazen, Andy Miller, Josiah Shannon

Goals for the Meeting	**Progress on Goals**	**Photos**
Put on Bigger Spool	We wanted a bigger spool to make the arm go out faster, but the spool was hitting parts of the slide so we had to go back to the original.	
Remove Touch Sensor for Linear Slide	The touch sensor was hurting us more than helping us because it shortened the slide so it was harder to reach and we felt like we weren't having any issues with over extending.	

Monday February 11, 2019
Attending Members:

Goals for the Meeting	Progress on Goals	Photos
Driver and Coach Tests	We did tests so our best drivers would be driving at state for the best opportunity. The scores were weighted based on the extra time commitment.	Our chart for scores is below.

Test with Position	Round 1	Round 2	Round 3	Average	Extra Credit	Total
Mark Hazen 1st	105	76	100	93.66666667	44.75	138.4166667
Andy Miller 1st	20	30	75	41.66666667	5.5	47.16666667
Josiah Shannon 1st	65	25	75	55	4.5	59.5
Drew Busch 2nd	35	35	95	55	12.5	67.5
Jackson Busch 2nd	105	110	100	105	9	114
Jackson Elliot 2nd	105	76	100	93.66666667	12.5	106.1666667
Mark Hazen 2nd	0	0	0	0	44.75	44.75
Andy Miller 2nd	65	70	65	66.66666667	5.5	72.16666667
Josiah Shannon 2nd	20	30	75	41.66666667	4.5	46.16666667
Drew Busch Coach	105	110	100	105	12.5	117.5
Jackson Busch Coach	35	35	95	55	9	64
Jackson Elliott Coach	102	95	100	99	12.5	111.5
Mark Hazen Coach	0	0	0	0	44.75	44.75
Andy Miller Coach	65	25	75	55	5.5	60.5
Josiah Shannon Coach	65	70	65	66.66666667	4.5	71.16666667

Tuesday, February 12th, 2018
Attending Members: All

Goals	Progress	Photos
How fast does our slide move?	Baylor, a member of the Atomic Gears, shared with us a google sheets guide that contains the algorithms used to determine speed of linear. We discovered our new, larger spool moves the about 9.3 inches/second, as opposed 3.9 inches/second. We will test the larger spool out to see how it performs.	

Linear Mechanism

	Free Speed (RPM)	Stall Torque (N*m)	Stall Current (Amp)	Free Current (Amp)
Rev Hex Motor	150	2.80	11.50	0.40

# Motors per Gearbox	Gearbox Efficiency	Travel Distance (in)	Applied Load (lbs)	Pulley Diameter (in)
1	80%	36	1	1

Driving Gear	Driven Gear		Elevator Linear Speed	Arm Time to move Travel Distance
1	2	No Load:	3.9 in/s	9.17 sec
1	1	Loaded:	3.9 in/s	9.29 sec
1	1			
1	1			
2.00 : 1	<-- Overall Ratio		Current Draw per Motor (loaded)	Stall Load
			0.51 amps	79.20 lbs

Speed of slide with original spool

Linear Mechanism

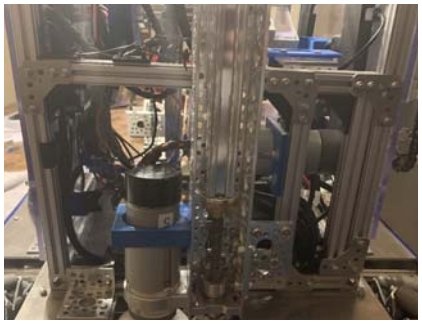
	Free Speed (RPM)	Stall Torque (N*m)	Stall Current (Amp)	Free Current (Amp)
Rev Hex Motor	150	2.80	11.50	0.40

# Motors per Gearbox	Gearbox Efficiency	Travel Distance (in)	Applied Load (lbs)	Pulley Diameter (in)
1	80%	36	1	2.362

Driving Gear	Driven Gear		Elevator Linear Speed	Arm Time to move Travel Distance
1	2	No Load:	9.3 in/s	3.88 sec
1	1	Loaded:	9.0 in/s	4.00 sec
1	1			
1	1			
2.00 : 1	<-- Overall Ratio		Current Draw per Motor (loaded)	Stall Load
			0.66 amps	33.53 lbs

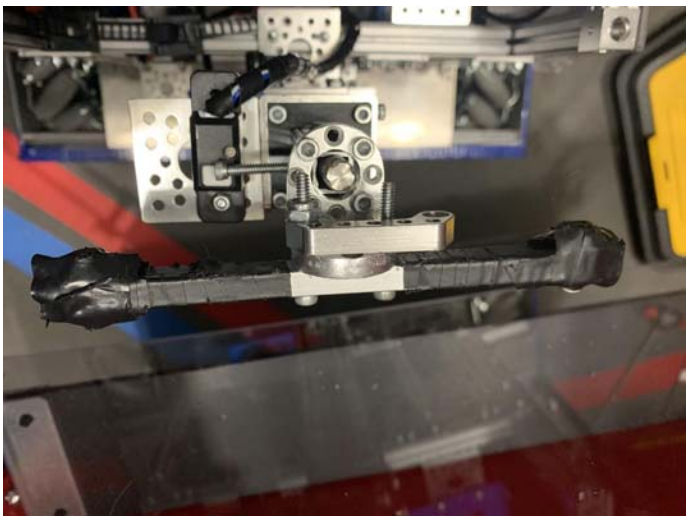
Speed of slide with new spool

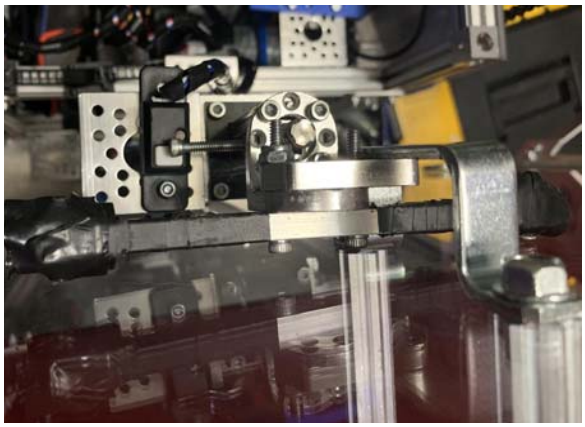
Friday February 15, 2019
Attending Members: Mark Hazen

Goals for the Meeting	**Progress on Goals**	**Photos**
Tighten Actuator Bar	I needed to tighten the crossbar between the two walls because it caused the actuator to wiggle. I added more supports on the front using L brackets and corner brackets.	

Saturday February 16, 2019
Attending Members:

Goals for the Meeting	Progress on Goals	Photos
Add a Spacer on Actuator	After our new wheelbase our robot was too low to the ground because of the weight, so I added another bar to space the actuator to keep the robot level. Then it was really hard to get out of the hook in autonomous so I started to design a claw like hook to not have to turn off the lander. While building we thought of a simple idea of adding a plate to the front of the actuator and shortening the current hook. This worked for the lander and our height.	






12535

Sunday February 17, 2019
Attending Members: Drew B., Jackson B., Jackson E., Mark H., Olivia M., Ivory M., Andy M., Josiah S.

Goals for the Meeting	Progress on Goals	Photos
Change Servo	The intake servo was a HiTec Continuous Servo and we wanted something faster but with the same or better torque. After some researching we found that the REV Smart Servo has higher speeds and torque. Using the SRS Programmer we changed the REV servo into a continuous servo from a normal servo. We had to order a 25 tooth spline hub because the HiTec servo had a 24 tooth spline. We previously used a REV Core Hex Motor for the intake and it worked well but it was very heavy.	

New Servo Specs

SPECIFICATIONS

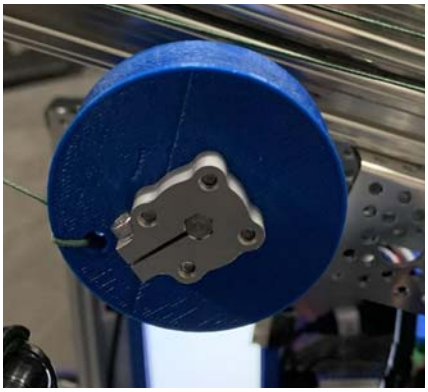
- Size: 40.2mm x 20.0mm x 38.0mm
- Weight: 2.05 oz.
- Speed: 0.14 s/60° (at 6V) - Note this speed is slightly higher for current inventory than previously stated (.14s/60° vs the .13/60°)
- Stall Torque: 13.5 kg-cm / 187.8 oz-in (at 6V)
- Voltage Rating: 4.8V – 7.4V, 6V nominal
- Input Pulse Range: 500µs – 2500µs
- Default Angular Range: 180°
- Maximum programmable range in angular mode - 280°
- Gear Material: Metal
- Spline type – 25T
- Spline Internal Thread Depth: 6mm
  - Do not exceed this depth as it may damage the servo

Old Servo Specs

Dimensions	1.59" x 0.77"x 1.44" (40.6 x 19.8 x 36.6mm)
Product Weight	1.47oz (41.7g)
Output Shaft Style	24 tooth (C1) spline
Voltage Range	4.8V - 6.0V
No-Load Speed (4.8V)	44 rpm
No-Load Speed (6.0V)	52 rpm
Stall Torque (4.8V)	38.8 oz-in (2.8 kg-cm)
Stall Torque (6.0V)	42 oz-in (3.1 kg-cm)
Pulse Amplitude	3-5V
Operating Temperature	-20°C to +60°C
Current Drain - idle (4.8V)	3mA
Current Drain - idle (6.0V)	3mA
Continuous Rotation Modifiable	Completed (stock)
Direction w/ Increasing PWM Signal	Clockwise
Motor Type	Brushed
Potentiometer Drive	No Potentiometer
Feedback Style	This servo does not have position feedback
Output Shaft Support	Dual Ball Bearing MR106
Gear Type	Straight Cut Spur
Gear Material	Nylon
Wire Length	7" (178mm)
Wire Gauge	25AWG

Make sure the software is updated	We verified that the hub software the SDK software and the phone software was up to date. Hub software-1.8.2 SDK software- 4.0 Phone software-4.3	

Monday, February 18th, 2019
Attending: All

Goals for the Meeting	Progress on Goals	Photos
Attach new spool	After driving with the 2.362 in spool we felt like we lost the torque needed to fully extend our linear arm, but the original 1in hex spool lacked in speed so we designed a new spool to split the difference of speed and torque. (See math below and in table on Engineering Notes Page 57).	

Linear Mechanism

	Free Speed (RPM)	Stall Torque (N*m)	Stall Current (Amp)	Free Current (Amp)
Rev Hex Motor	150	2.80	11.50	0.40

# Motors per Gearbox	Gearbox Efficiency	Travel Distance (in)	Applied Load (lbs)	Pulley Diameter (in)
1	80%	36	1	1.587

Driving Gear	Driven Gear		Elevator Linear Speed	Arm Time to move Travel Distance
1	2	No Load:	6.2 in/s	5.78 sec
1	1	Loaded:	6.1 in/s	5.90 sec
1	1			
1	1			
2.00 : 1	<-- Overall Ratio		Current Draw per Motor (loaded)	Stall Load
			0.58 amps	49.90 lbs

Speed with this spool

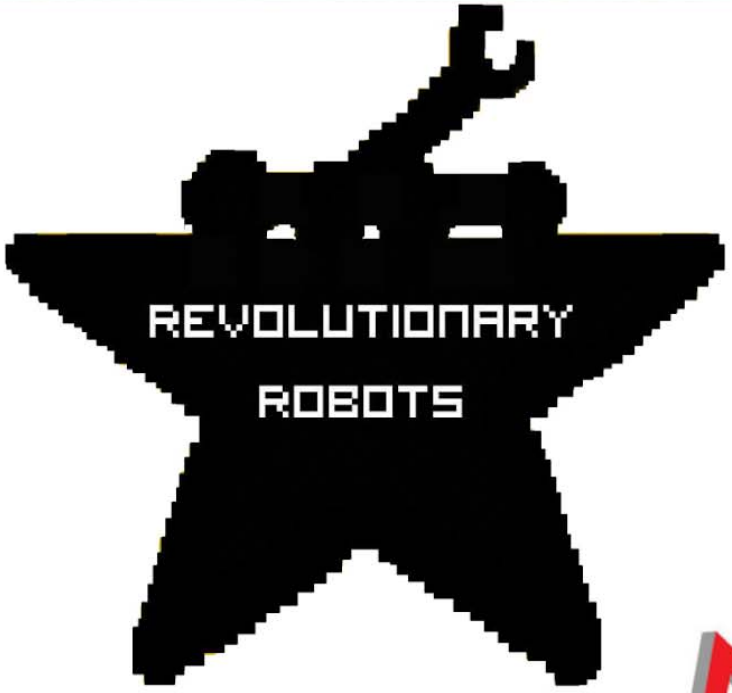
February 21, 2019
Attending team members: Drew, Jackson, Jackson, Mark, Olivia, Andy, Ivory, Josiah

Pieces Of The Robot That People Are Proud Of	
Team Member	What and why
Drew Busch:	I am proud of the linear slide, because it has evolved since last qualifier where we barely used it to a valuable piece of the robot.
Jackson Busch	I like the ½ in stainless steel bar on the rotation mechanism for the linear slide. If it bends we have bigger problems.
Jackson Elliott	The linear actuator is a part I am glad we found. With it we can easily lift our robot off the ground, because of the lead screw.
Mark Hazen	I am so happy that we were able to create a wheelbase, but also that the Atomic Gears helped. Because of it, our motors and wheels won't flex because of strain.
Olivia Martin	Our mineral bucket has evolved a lot since the first CAD design. It is simple, but it works.
Andy Miller	I am confident in the wheelbase, because of how sturdy and consistent it is.
Josiah Shannon	The linear actuator is a powerhouse of a mechanism. It is good at hanging which has a priority since the beginning of the season.

Goals for the Meeting	**Progress on Goals**
Clean Up Robot	We blasted the dust off of the robot and took care of the edges of the screws by cutting them. We then ran a self inspection so that we knew we would be fine during inspection.

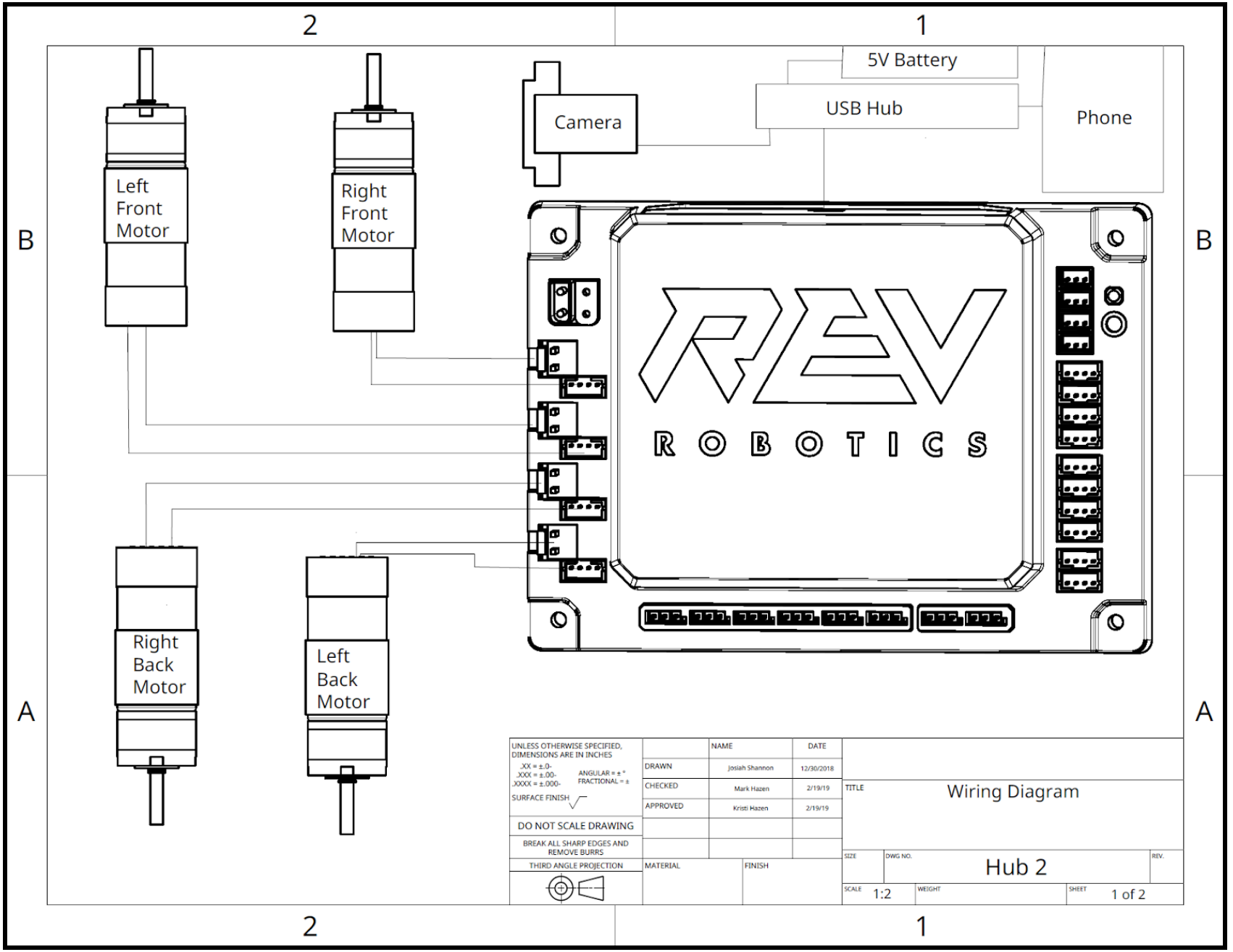
Revolutionary Robots 12535

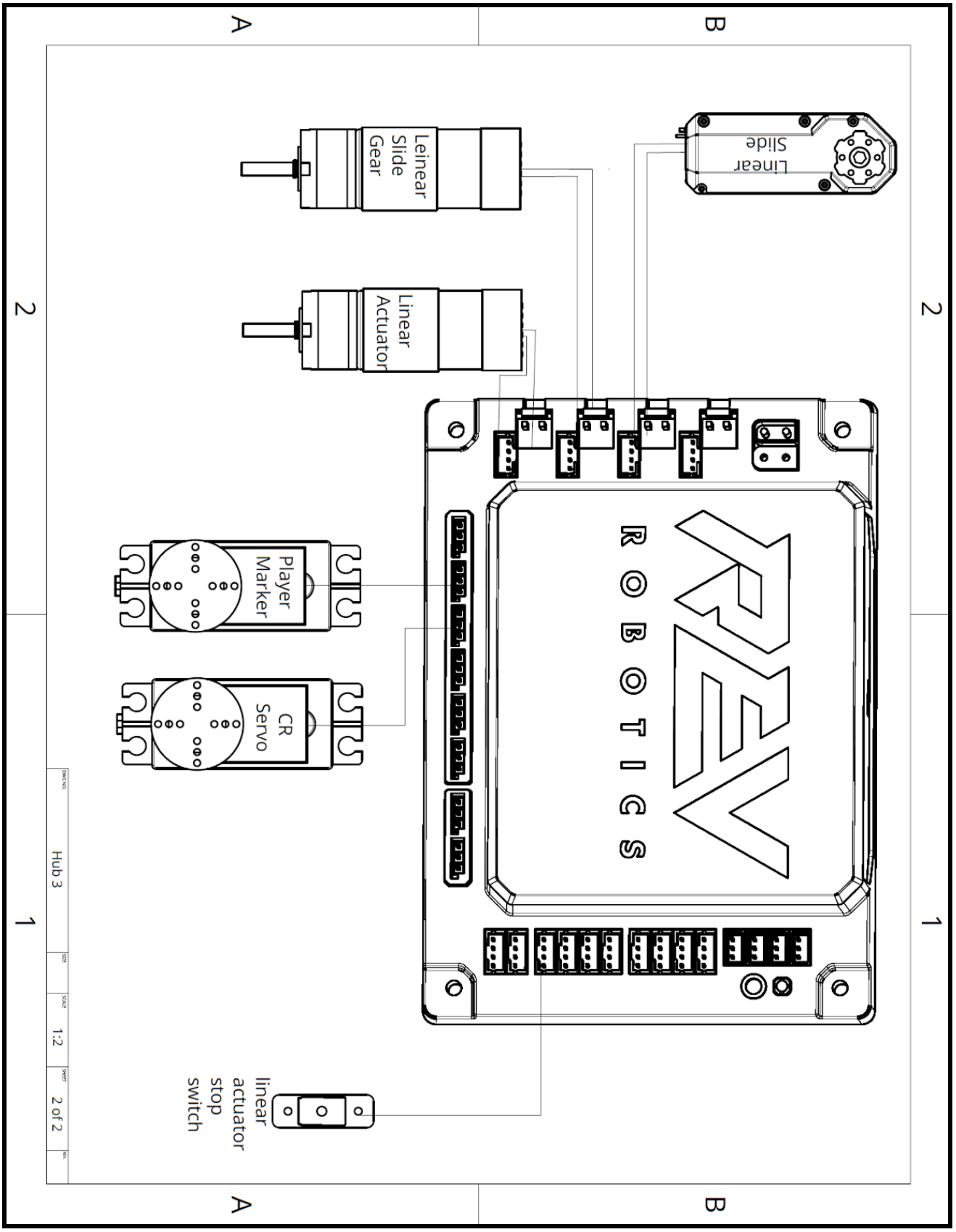
Electrical 2018-2019



Moore Community STEM Club

FIRST


2
1
B
A
5V Battery
USB Hub
Phone
Camera
Left Front Motor
Right Front Motor
Right Back Motor
Left Back Motor
REV ROBOTICS
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES
.XX = ±.0-
.XXX = ±.00-
.XXXX = ±.000-
ANGULAR = ± °
FRACTIONAL = ±
SURFACE FINISH
DO NOT SCALE DRAWING
BREAK ALL SHARP EDGES AND REMOVE BURRS
THIRD ANGLE PROJECTION
NAME
DATE
DRAWN
Josiah Shannon
12/30/2018
CHECKED
Mark Hazen
2/19/19
APPROVED
Kristi Hazen
2/19/19
MATERIAL
FINISH
TITLE
Wiring Diagram
SIZE
DWG NO.
Hub 2
REV.
SCALE
1:2
WEIGHT
SHEET
1 of 2


Linear Slide
Leinear Slide Gear
Linear Actuator
Player Marker
CR Servo
linear actuator stop switch
REV ROBOTICS
Hub 3
1:2
2 of 2

-Robot Configuration-			
Hub	**Port**	**Type**	**Name**
Hub 2	Motor Port 0	NeveRest 20 Gearmotor	rightBack
Hub 2	Motor Port 1	NeveRest 20 Gearmotor	leftBack
Hub 2	Motor Port 2	NeveRest 20 Gearmotor	rightFront
Hub 2	Motor Port 3	NeveRest 20 Gearmotor	leftBack
Hub 2	I2C Bus 0	REV Expansion Hub IMU	imu
Hub 3	Motor Port 0	REV HD 40 Gearmotor	intakeArm
Hub 3	Motor Port 2	NeveRest 20 Gearmotor	linearActuator
Hub 3	Motor Port 3	REV Core Hex Motor	pulley
Hub 3	Servo Port 2	Servo	playerMarker
Hub 3	Servo Port 3	Continuous Rotation Servo	intake
Hub 3	Digital Port 1	Digital Device	actuatorTouch
Hub 3	I2C Bus 0	REV Expansion Hub IMU	imu 1
Webcam	USB Hub	Webcam	Webcam 1

-Electrical Components-		
Hubs		
REV Expansion Hub	Used to control robot	

REV ROBOTICS

EXPANSION HUB PRODUCT BRIEF

The REV Robotics Expansion Hub is a low-cost education device that can communicate with any computer (Commonly an Android Phone or the REV Robotics Control Hub) to provide the interfaces required for building robots and other mechatronics. The Expansion Hub was purposed built to stand up to the rigors of the classroom and competition field. It features a mature firmware designed for basic and advanced use cases with the ability to be field upgraded in the future.

The REV Robotics Expansion hub is an approved device for use in the ***FIRST*** Tech Challenge and ***FIRST*** Global.

FEATURES

- **Physical Dimensions**
  - 143mm X 130mm X 29.5 mm
  - Mounting holes on a 16mm spacing
- **Processor**
  - Texas Instruments ARM® Cortex®-M4
- **Input Voltage**
  - Dual XT30 Connectors
  - 12VDC nominal (8V MIN-15V MAX)
- **Integrated Motor Controllers**
  - 4X - 20amps Max H-bridges
  - Integrated PID controls
    - Motor Position
    - Motor Speed
    - Motor Current
- **Integrated Encoders**
  - 4X 3.3V Quadrature Encoder Inputs
    - 2x in hardware
    - 2x in software
- **Servo Drivers**
  - 6 channels – 5 Amps Max
    - 2 Amps per channel (shared)
    - Overcurrent protected (PTC)
- **Aux output**
  - 2x – 5VDC power ports
  - 2 Amps (shared)
- **Inertial measurement unit (IMU)**
  - Bosch 9 axis IMU with Sensor Fusion
  - Internally wired to I2C (port 0)
- **Digital IO**
  - 8X – user programmable in/out
  - 3.3VDC
- **Analog Inputs**
  - 4X – 12 bit Analog inputs
  - 3.3VDC nominal (5V tolerant)
- **USB**
  - 2.0 Speed
  - Supports charging of USB host devices
  - Firmware upgradeable
- **I2C**
  - 4X – 3.3V 100KHz Independent buses
- **RS485**
  - 2 connector 3.3V signaling @400Kbaud
  - Used for linking up to 12 Expansion hubs
- **UART**
  - 2X UART Debugging ports
  - Future additional functionality
- **Safety features**
  - Reverse polarity protection
  - Polarized & latching connectors
  - Failsafe mode at communication loss
- **Mode Button** – User programmable
- **LED** – User programmable status RGB LED

Copyright © 2017 REV Robotics, LLC

Anker USB Hub	Splitting Camera and Expansion Hub	

System Requirements

Windows 8 / 7 / Vista / XP, Mac OS X 10.2 (and above)

Mac OS X Lion 10.7.4 users should upgrade to Mountain Lion 10.8.2 or later to avoid unstable connections.

Compatibility:

2.4 GHz wireless devices, MIDI devices and some USB 3.0 devices may not be supported. Try using the host port or a USB 2.0 connection.

Power Usage:

For a stable connection, avoid connecting high power-consumption devices, such as external hard drives. The hub will sync but not charge tablets and other devices that require a higher power input.

Monoprice USB Hub	Splitting the Controllers for Driving	

- 4 USB 2.0 ports for easy expansion
- Supports a data transfer rate of up to 480 Mbps
- 40 Times faster than USB 1.1 products
- Plug-n-Play, hot swappable
- Backward compatible with USB 1.1 specifications
- Supports Windows 98, 98SE, 2000, ME, XP, Vista and Mac OS X or later version
- Overcurrent Protection
- Self Powered
- USB cable tucks away when not in use

-Electrical Components-

Motors

Component Type	Where Used	Picture
NeveRest Orbital 20 Gearmotor	Wheels Linear Actuator	

Theoretical Performance Specifications:

- Gearbox Reduction: 19.2:1
- Voltage: 12 Volt DC
- No Load Free Speed, at gearbox output shaft: 340 RPM
- Force Needed to Back-Drive: 6.4 oz-in
- Gearbox Output Power: 14 W
- Stall Torque: 175 oz-in
- Stall Current: 11.5 A
- Output counts per revolution of Output Shaft (cpr): 537.6
- Output pulse per revolution of encoder shaft (ppr): 134.4

REV 40 HD Gearmotor	Linear Slide Rotation	

40:1 SPUR GEARBOX OPTION

- Weight: 350g (including motor)
- Output Shaft: 5mm hex
- Output Shaft Length: 40mm
- Mounting Holes: 10 - M3 tapped - use a 5mm length or shorter bolt
- Free Speed: 150 rpm (15.7 rad/s)
- Stall Torque: 594.7 oz-in (4.2 Nm)

REV Core Hex Motor	Pulley	

SPECIFICATIONS

- Output Shaft: 5mm Female Hex
- Weight: 7 oz
- Free Speed: 125 RPM
- Stall Torque: 3.2 N-m
- Stall Current: 4.4 A
- Gear Ratio: 72:1
- Encoder Counts per Revolution
  - At the motor - 4 counts/revolution
  - At the output - 288 counts/revolution

-Electrical Components-

Servos

Component Type	Where Used	Picture
HiTech HS-485HB	Placing Player Marker	

Dimensions	1.57" x 0.78" x 1.49" (39.88 x 19.81 x 37.85mm)
Product Weight	1.59oz (45g)
Output Shaft Style	24 tooth (C1) spline
Voltage Range	4.8V - 6.0V
No-Load Speed (4.8V)	0.22sec/60°
No-Load Speed (6.0V)	0.18sec/60°
Stall Torque (4.8V)	66.6 oz/in. (4.8kg.cm)
Stall Torque (6.0V)	83.3 oz/in. (6.0kg.cm)
Max PWM Signal Range (Standard)	553-2425µsec
Travel per µs (out of box)	.102°/µsec
Max Travel (out of box)	190.5°
Pulse Amplitude	3-5V
Operating Temperature	-20°C to +60°C
Current Drain - idle (4.8V)	8mA
Current Drain - idle (6.0V)	8.8mA
Current Drain - no-load (4.8V)	150mA
Current Drain - no-load (6V)	180mA
Continuous Rotation Modifiable	Yes
Direction w/ Increasing PWM Signal	Clockwise
Deadband Width	8µs
Motor Type	3 Pole Ferrite
Potentiometer Drive	Indirect Drive
Feedback Style	5KΩ Potentiometer
Output Shaft Support	Top Ball Bearing, Bottom Bushing
Gear Type	Straight Cut Spur
Gear Material	Karbonite
Wire Length	11.81" (300mm)
Wire Gauge	25AWG

REV Smart Servo	Intake	

SPECIFICATIONS

- Size: 40.2mm x 20.0mm x 38.0mm
- Weight: 2.05 oz.
- Speed: 0.14 s/60° (at 6V) - Note this speed is slightly higher for current inventory than previously stated (.14s/60° vs the .13/60°)
- Stall Torque: 13.5 kg-cm / 187.8 oz-in (at 6V)
- Voltage Rating: 4.8V – 7.4V, 6V nominal
- Input Pulse Range: 500µs – 2500µs
- Default Angular Range: 180°
- Maximum programmable range in angular mode - 280°
- Gear Material: Metal
- Spline type – 25T
- Spline Internal Thread Depth: 6mm
  - Do not exceed this depth as it may damage the servo

-Electrical Components-

Sensors

Component Type	Where Used	Picture
REV Touch Sensor	Linear Actuator Stop	

CONFIGURATION

The REV Touch Sensor is a digital sensor that is wired to the N+1 channel on our standard 4-pin JST PH connector. In other words, the REV Expansion Hub digital ports are grouped with two input/output channels per port: 0-1, 2-3, 4-5, 6-7. In general we refer to each channel as N and N+1. When pressed, the touch sensor pulls the digital channel low from its default high state.

Logitech HD Pro Webcam C922	Tensor Flow	

Device Type
Web camera

Dimensions (WxDxH)
3.7 in x 2.8 in x 1.7 in

Weight
5.71 oz

Connectivity Technology
Wired

Camera
Color

Frame Rate (max)
30 frames per second

Video Modes
720p, 1080p

Focus Adjustment
Automatic

Manufacturer Warranty
1 year warranty

-Electrical Components-

Batteries

Component Type	Where Used	Picture
REV Battery	Robot Power	

SPECIFICATION

- Voltage: 12V
- Capacity: 3000mAh
- Weight: 567g
- Maximum Dimensions: 113.5mm x 90.5mm x 23mm
- Wire Gauge: 16 AWG
- Connector: XT30
- Replaceable Fuse: 20A ATC
- Maximum Discharge Rate: 10C
  - While the battery cells are rated at a 10C (30A) discharge, the in-line fuse limits this to 20A.
- Wire Length (excluding XT30): 150mm

BEST PRACTICES

All rechargeable batteries have a finite lifespan. Factors that affect lifespan include the number of discharge/charge cycles and the average loading of the battery. The following best practices can help maximize the lifespan of your battery:

- Charge rate
  - Minimum: 1.5A
  - Maximum: 3.0A
  - Recommended: 1.8A or 2.0A
- Do not overcharge
  - Disconnect the battery from the charger once it indicates a full charge.
  - Typical charge time does not exceed 2 hours.
  - Do not charge a battery that hasn't been discharged significantly.
    - For example, running the robot under minimal load for a few minutes will not significantly discharge the battery.
- Minimum no-load voltage: 9.0V
  - Discharging the battery past 9.0V can reduce the lifespan of the battery and can permanently damage the cells.
  - Periodic dips below 9.0V when under load is expected and OK.
    - For example, don't forget to unplug your battery after you are finished running the robot and don't run your robot until it completely stops responding!
- Temperature
  - Let the battery cool before and after charging.
  - The battery may feel warm after heavy loading or after charging. This is normal.

Tetrix Battery	For Backup	

Specifications

- Weight: 579.6 g per battery
- Size: 115 mm x 45 mm x 50 mm
- Other Products Required: Battery charger, sold separately

When a new battery pack is connected to a load, it will show a full charge for only 10-15 minutes. If this occurs, remove the battery and allow it to cool for 15 minutes and then charge.

Use only the TETRIX battery charger set on the 0.9 amp setting. It is recommended to always charge your battery pack on the 0.9 amp setting to ensure safety. The charger should be switched off if the battery temperature rises above 55 degrees Celsius. To avoid voltage depression, do not discharge the battery pack below 0.8 volt per cell. NiMH cells do not accept full charge and discharge if the temperature is above 50 degrees or below 0 degrees Celsius. NiMH cells perform best at 0-40 degrees Celsius.

NiMH cells can hold a full charge for a month if stored at normal temperature. To maximize battery pack life, condition the battery once a month. This is done by completely discharging the battery and then charging completely.

If the battery is not used for long periods of time, first charge the battery fully and store in a cool, dry place. Recharge again once a month. Always inspect your battery pack before charging. If there is any visible deformity, damage, or leaking fluid, do not attempt to use it and dispose of it properly.

Note: It is recommended that the battery pack be replaced after two years of use.

5V Battery	For Camera	

-Electrical Components-

Phones

Component Type	Where Used	Picture
Moto g4	Robot Controller and Driver Station	

operating system

Android™ 6.0.1, Marshmallow

system architecture/processor

Motorola Mobile Computing System, including an up to 1.5 GHz Qualcomm® Snapdragon™ 617 octa-core processor with 550 MHz Adreno 405 GPU

memory (RAM)

2 GB

storage (ROM)

16 GB / 32 GB internal, up to 128 GB microSD Card support**

dimensions

Height: 153 mm
Width: 76.6 mm
Depth: 7.9 mm to 9.8 mm

weight

155g

display

Corning® Gorilla® Glass 3
5.5"
1080p Full HD (1920 x 1080)
401 ppi

battery

All-day battery† (3000 mAh)
TurboPower™ for up to 6 hours of power in 15 minutes of charging†

water protection

Water repellent coating for added water resistance‡

networks

4G LTE (Cat 4)
CDMA / EVDO Rev A
UMTS / HSPA+
GSM / EDGE

bands (by model)

Moto G—XT1625
CDMA (850, 1900 MHz)
GSM / GPRS / EDGE (850, 900, 1800, 1900 MHz)
UMTS / HSPA+ (850, 900, 1700, 1900, 2100 MHz)
4G LTE (B1, 2, 3, 4, 5, 7, 8, 12, 13, 17, 25, 26, 41)
Band coverage varies by model, country, and carrier.

rear camera

13 MP
ƒ / 2.0 aperture
Color balancing dual LED flash
Professional Mode
Quick Capture
Best Shot
Tap (anywhere) to capture
4X digital zoom
Burst mode
Auto HDR
Panorama
Drag to focus & exposure
Video Stabilization
1080p HD video (30 fps)
Slow Motion video

front camera

5 MP
ƒ / 2.2 aperture
Wide-Angle lens
Display flash

connectivity

Micro USB, 3.5 mm headset jack

Bluetooth® technology

Bluetooth version 4.2 LE

Wi-Fi

802.11 a/b/g/n (2.4 GHz + 5 GHz)

speakers/microphones

Front-ported loud speaker
2-Mics

video capture

1080p (30 fps)

location services

A-GPS

sensors

Acceleromter
Gyroscope
Ambient Light

base color

Black or white

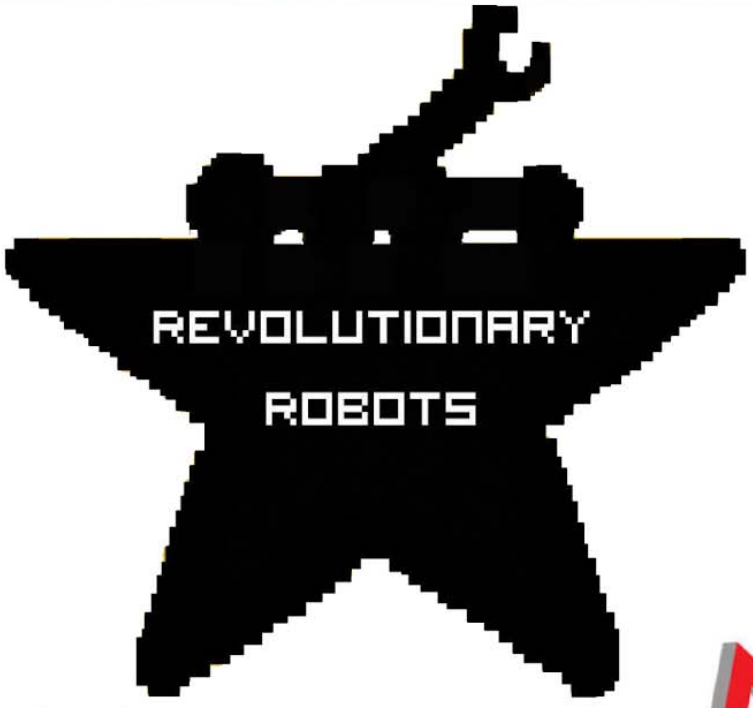
moto maker

Choose from a variety of materials, colors, accents and more.

-Electrical Components-		
Controllers		
Component Type	Where Used	Picture
Logitech Gamepad F310	Driver One and Driver Two	
XBox 360 Wired	Backup Driver	

Revolutionary Robots
12535

Computer-Aided Design
2018-2019

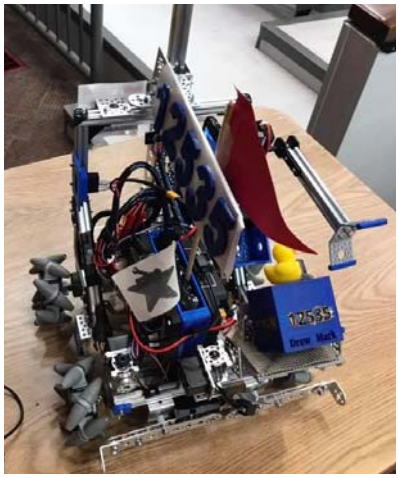


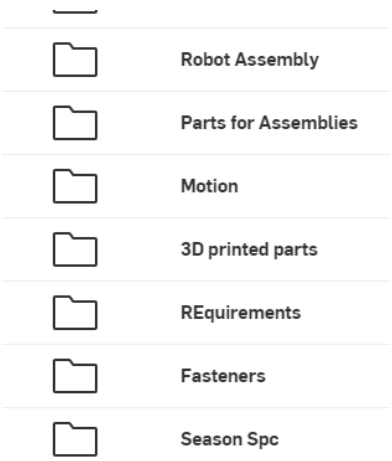
Moore Community STEM Club

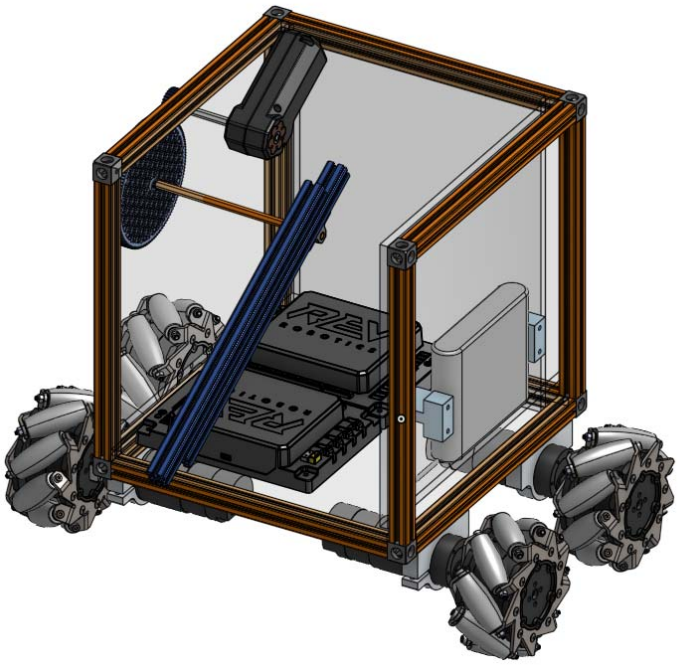
FIRST®

-Overall Design-

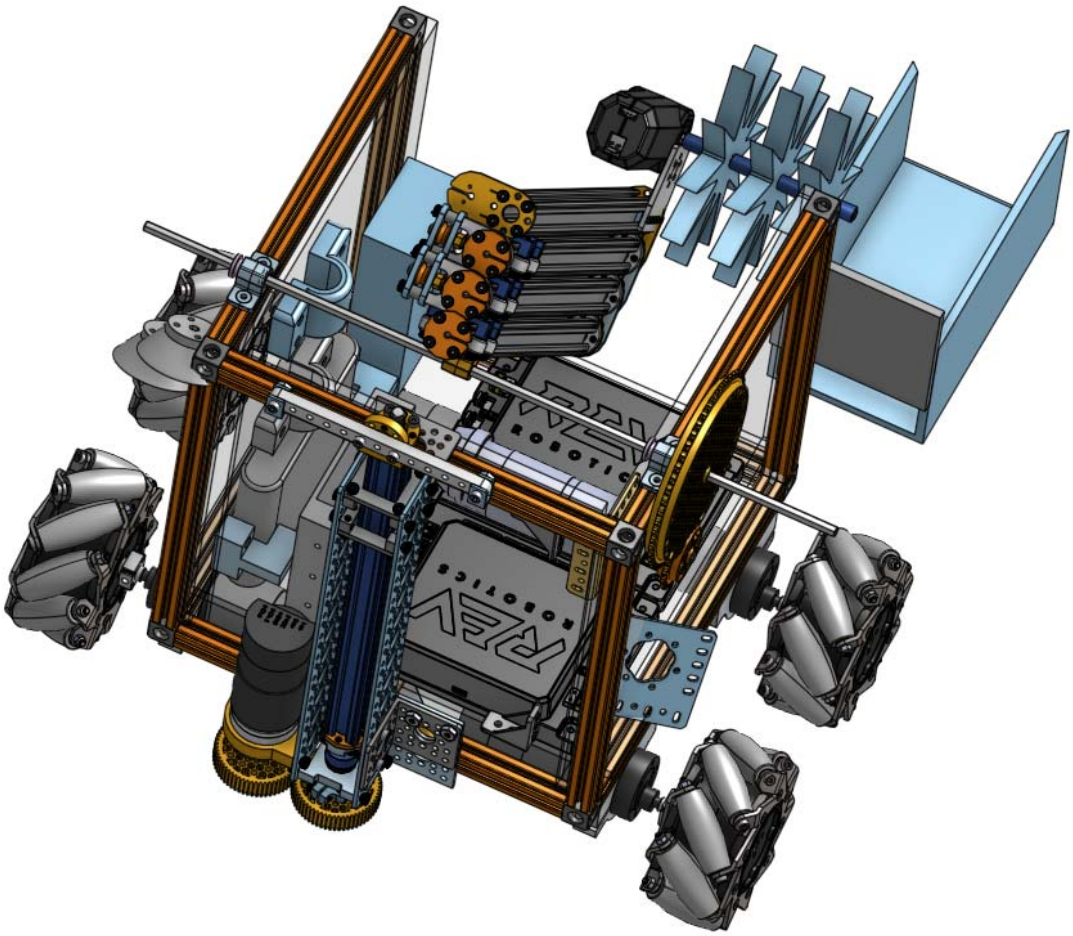
18/19 November, 2018	The Start of CADing	After the Gordon Cooper qualifier we decided to CAD our robot design instead of tinkering with parts. We started importing parts to assemble the bot in On-Shape. Today we had a base design with a placeholder arm.



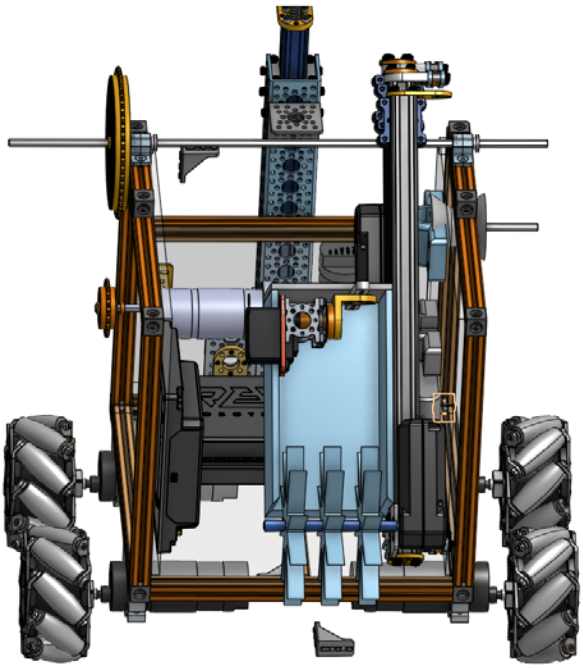


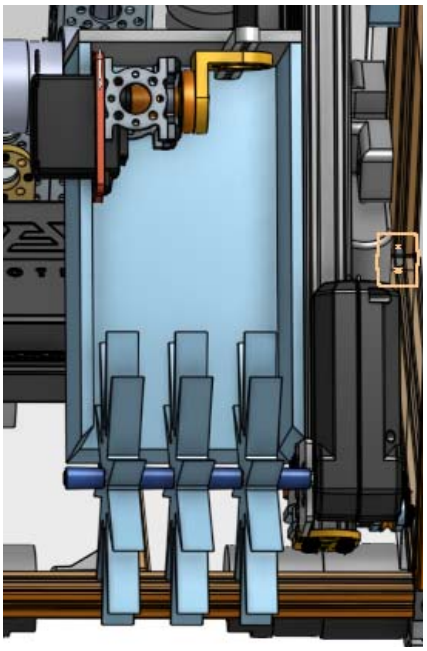


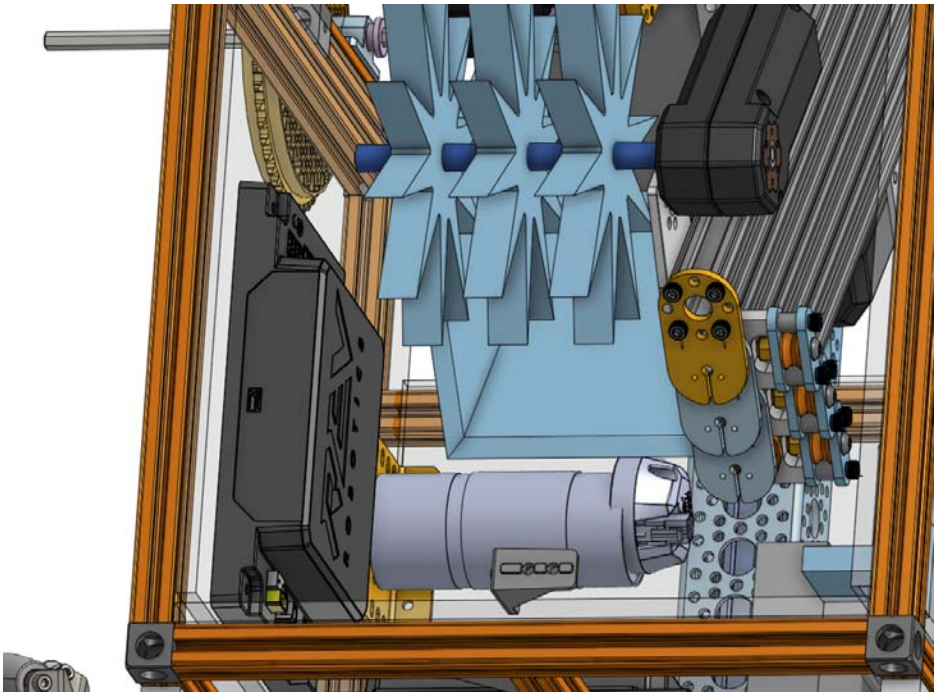
2 December, 2018	Linear Actuator and Linear Slide on base	We have the linear actuator which we use to hang and the linear slide which we use to score. This is the basic idea of our robot.



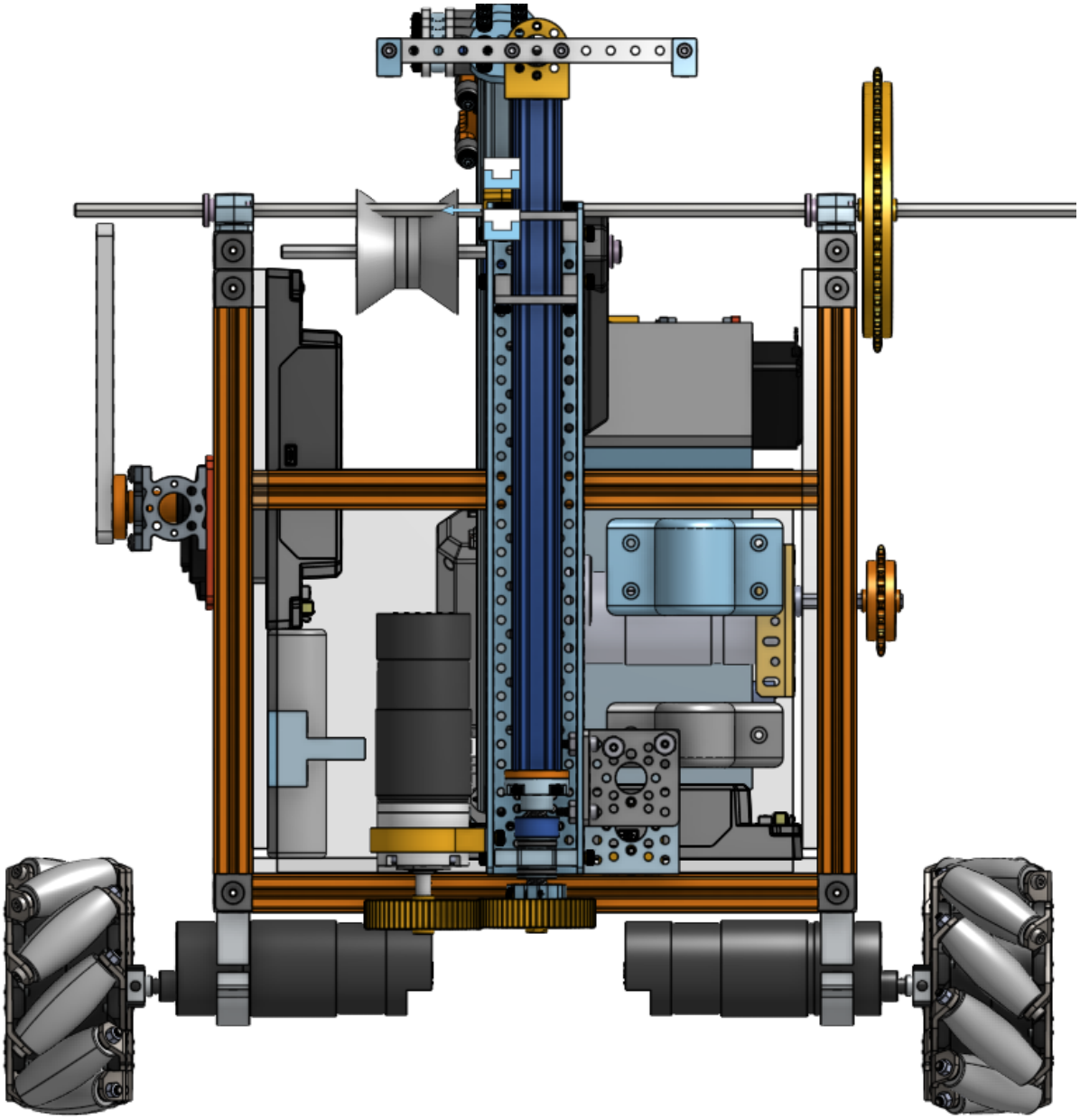
7 December 2018	Moving down the Linear Actuator bar, updating the Mineral Box, and Moving the Rev Hub	We moved the linear actuator bar, because the linear slide hit the bar. We were afraid of the linear slide from hitting the expansion hub on the floor. We have the mineral box to dump the minerals



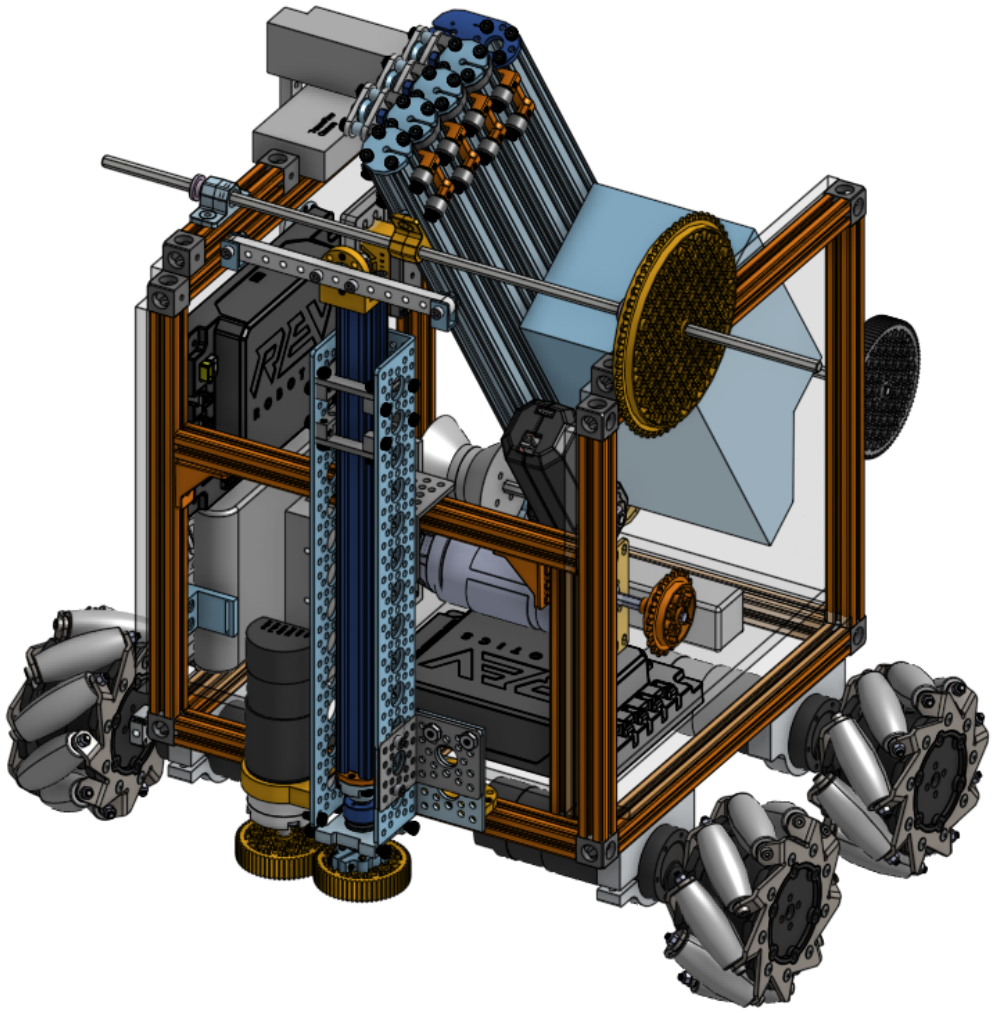




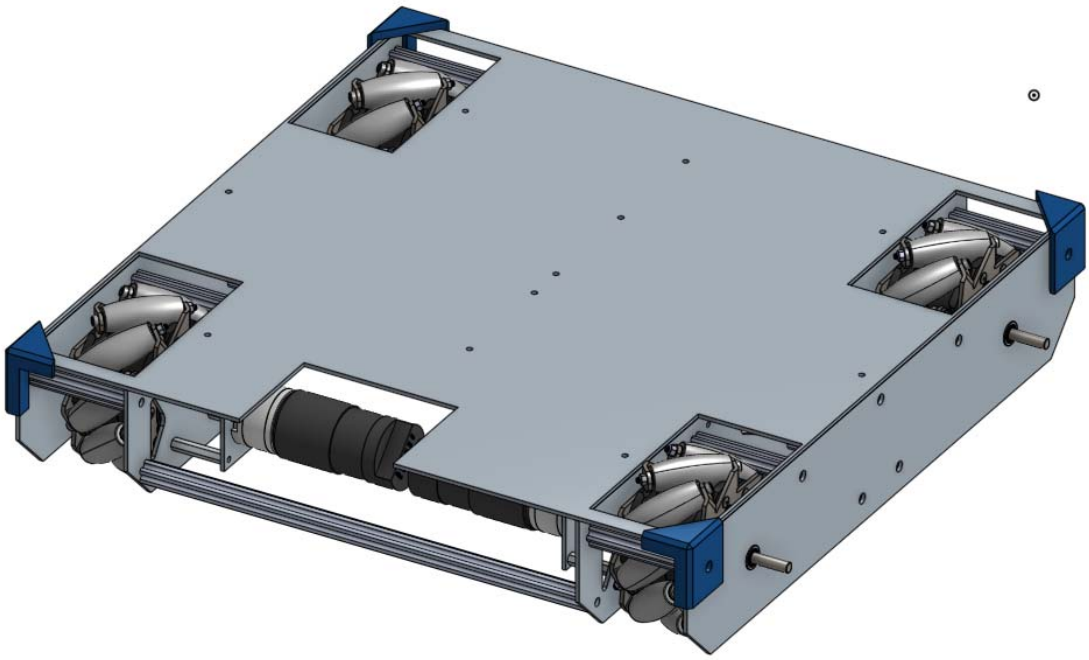
8 December 2018	Moving Expansion Hub and Flag Holder	The hub was still going to be hit by the linear slide, so we moved it to the right side to keep it safe. The flag holder was were the hub needed to be, so we moved it to the front.

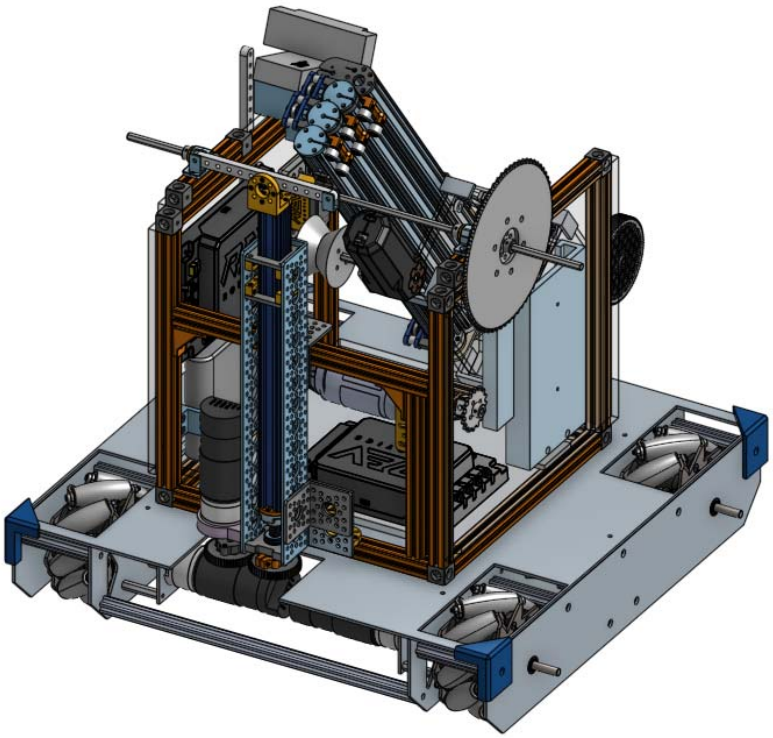


4 January 2019	Updated and Finished Robot Design for Newcastle	This is our first robot we CADed completely. We attempted to start on drawings, but the pictures would not draw correctly.

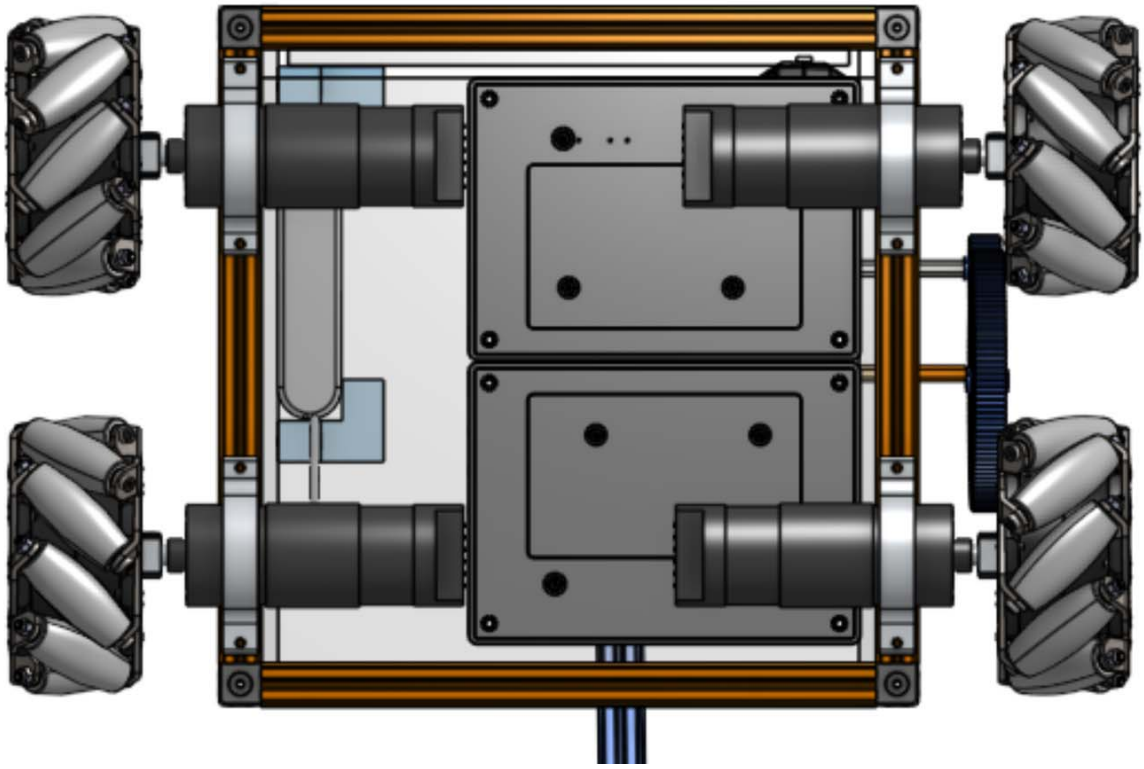


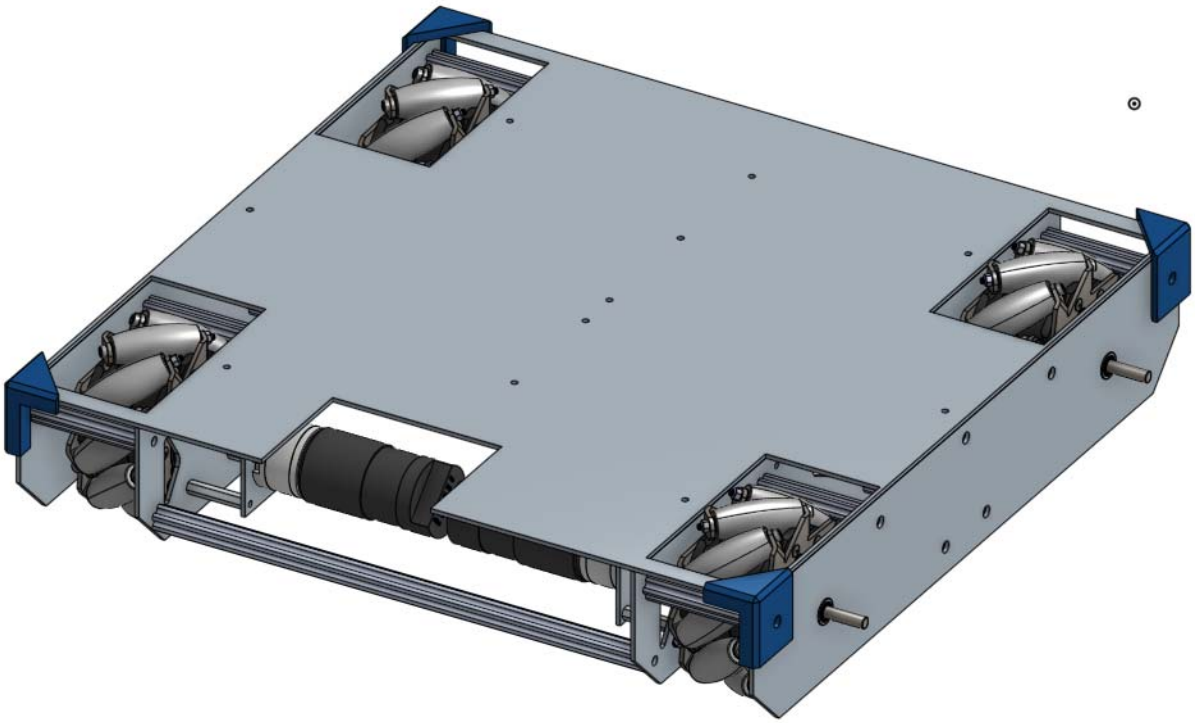
5 January, 2019 - 24 January 2019	Starting a Wheel Base for Washi	We were having problems with our motors bending so we created a base that contained the wheels for this competition and later ones too. We have corner brackets to keep the sides together that are 3D printed.

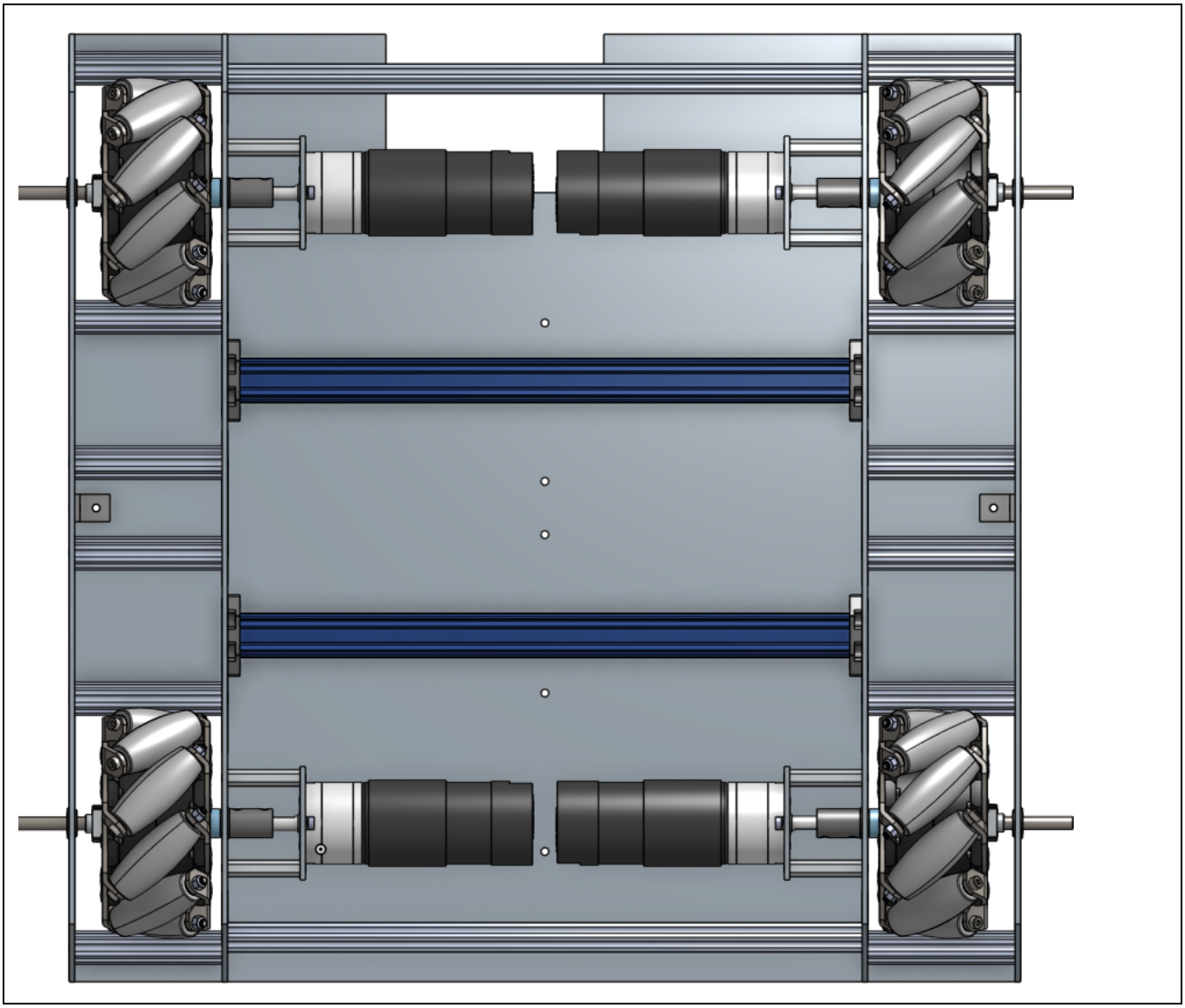


2 February 2019	Attaching updated Washi to the base.	We updated Washi to have a metal sprockets, clear mineral box, and correct small details.
		

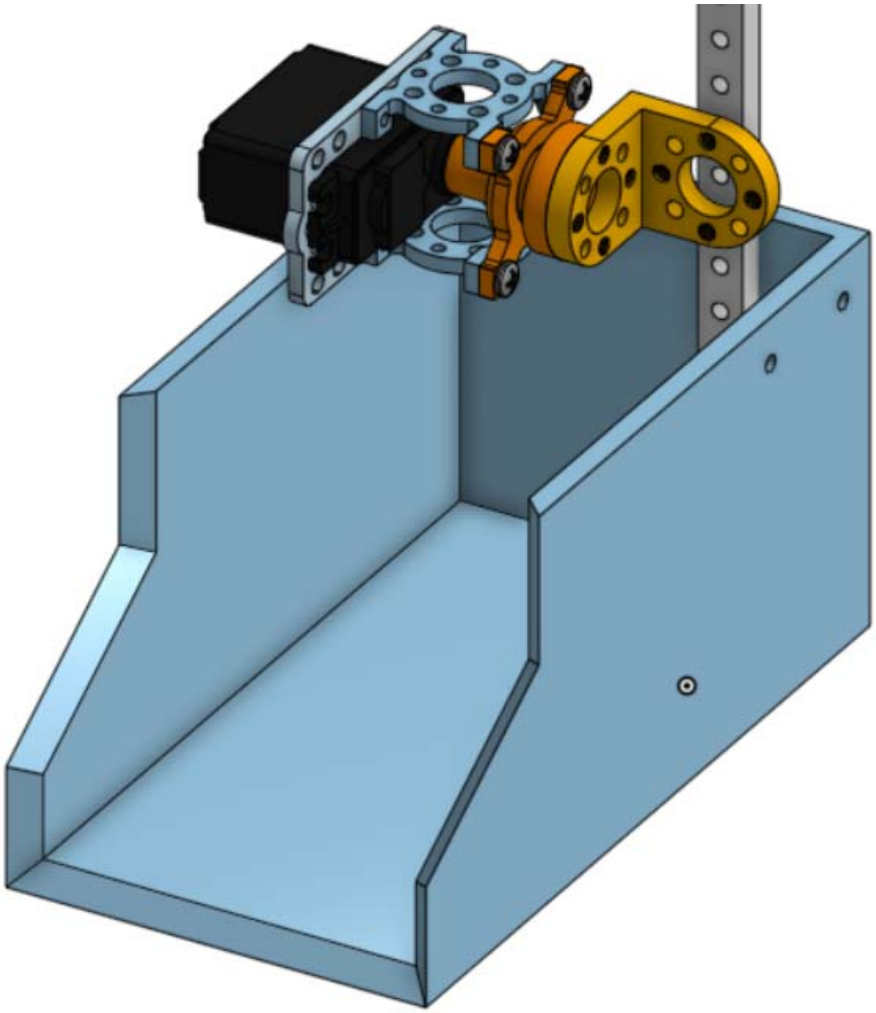
-Wheel Base-		
September 9- November 10	Original wheel base	The base consisted of four plastic mecanum wheels attached to four Neverest 60:1 gear motors.
We have no CAD of this wheel base, but there is a picture at Engineering Notes Page 7.		
November 11- January 5	Motor mount to rev rail base	This base consisted of four motor mounts to the rev rails. This base differed from the first in that we found that the plastic mecanum wheels did not perform well when we began to add more and more weight so we purchased HD mecanum wheels that are rubber.



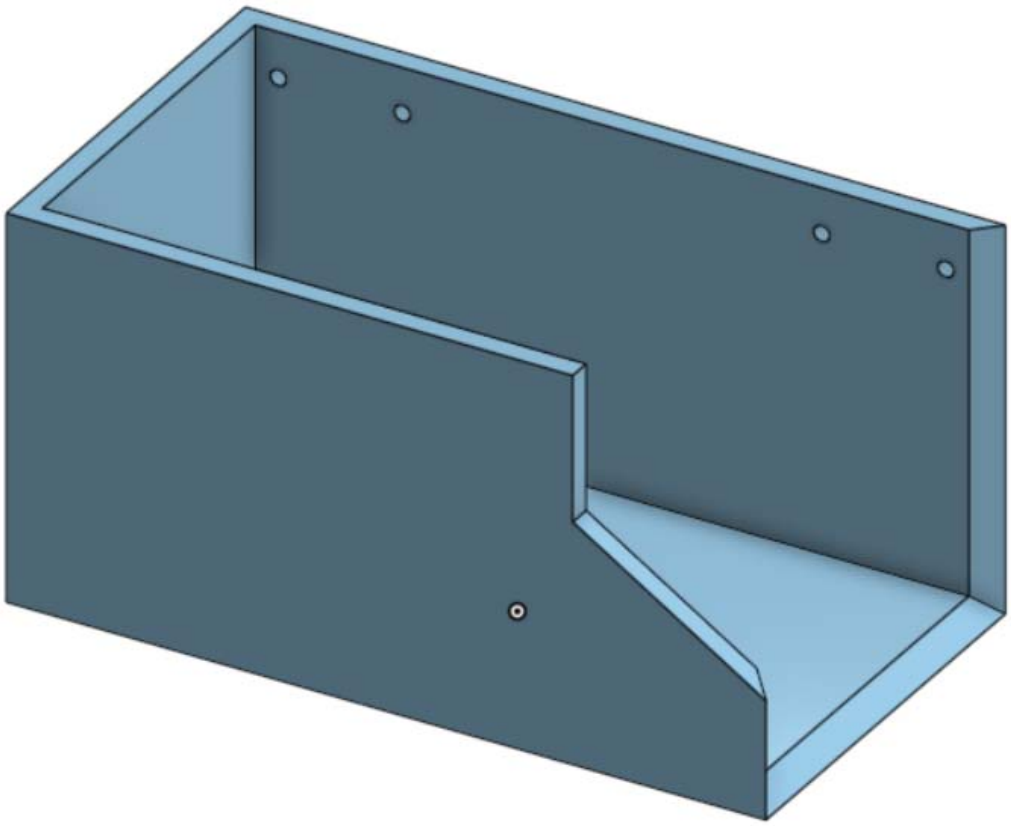
January 6- February 22	Aluminum sheet metal base	The motors were bending and as time went on our camber increased. Mark started to plan out his prototype in CAD to stabilize the wheels. In the end we use three brackets on each side to attach the top plate to keep the plate on. The X-Rail on the bottom supports the top plate as well. While designing Mark kept in mind the long term use. He made it a 17in by 17 in so we can mount a little bit on the sides but the robot can be bigger.
		



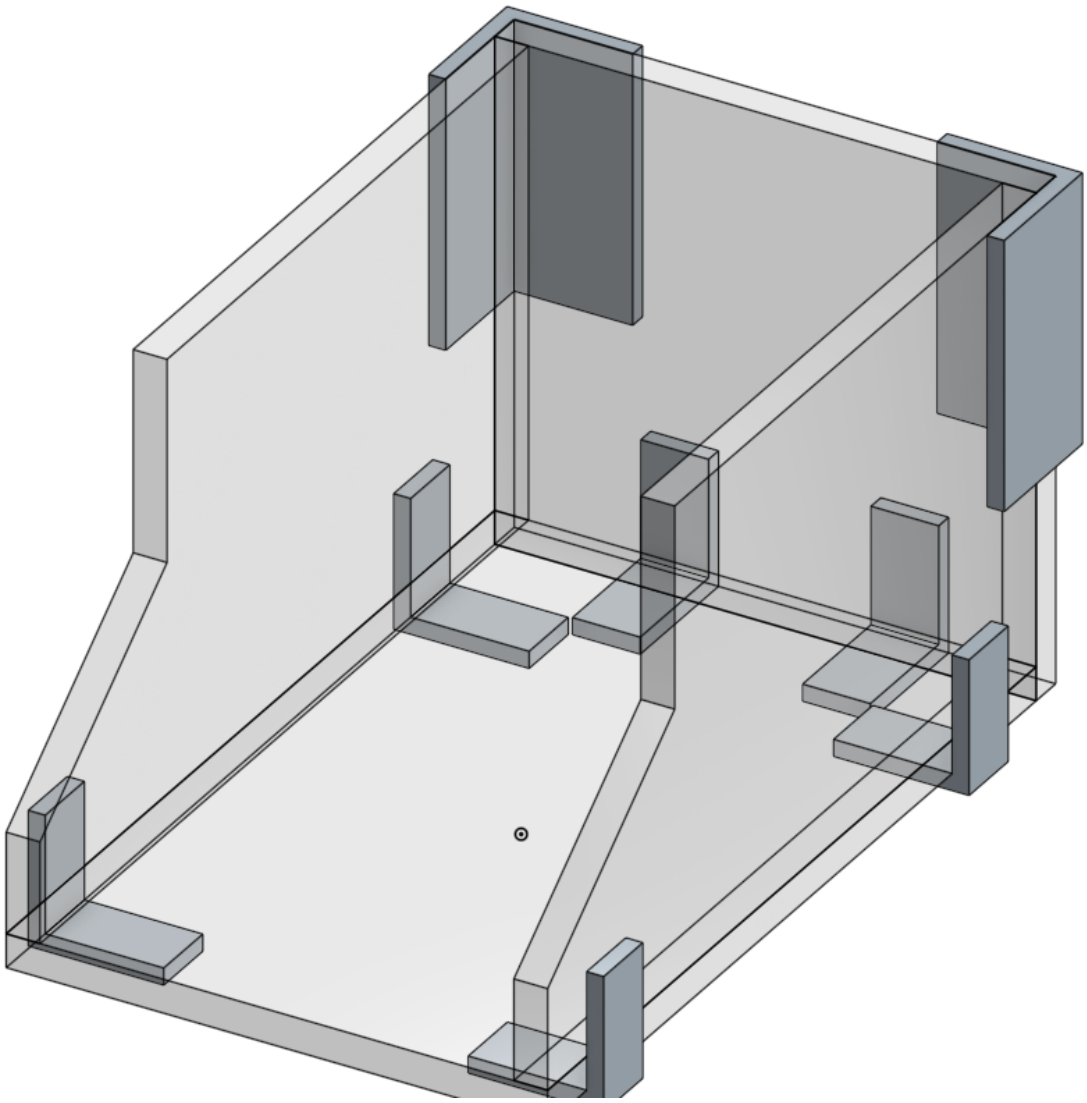
-Mineral Box-		
November 11-November 27	3D printed box with a servo	We planned on using a 3D printed box with a servo on the back to swing open a “door” that would let the minerals slide out the back

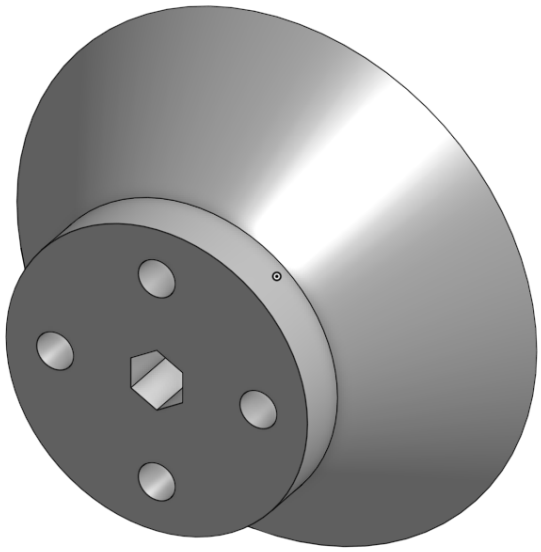
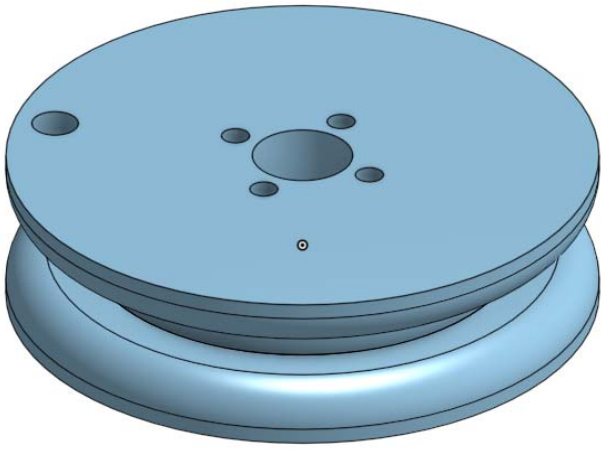


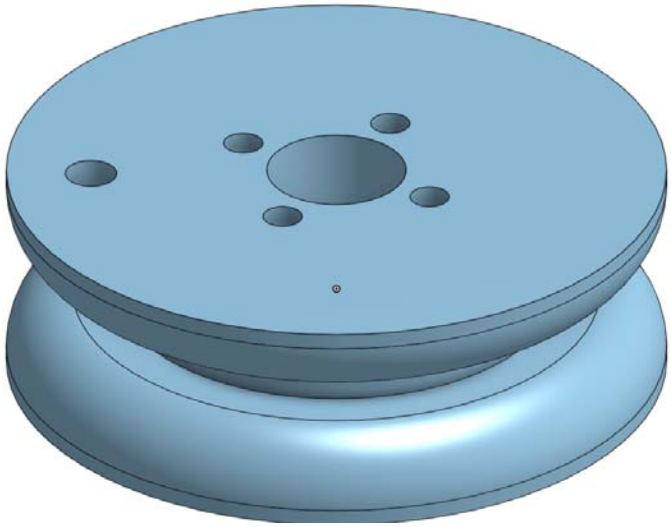
November 28- December 28	3D printed box	After discussions as a team, we decided that the servo would be increased weight without much benefit, and instead just let the minerals slide out the back of the box.



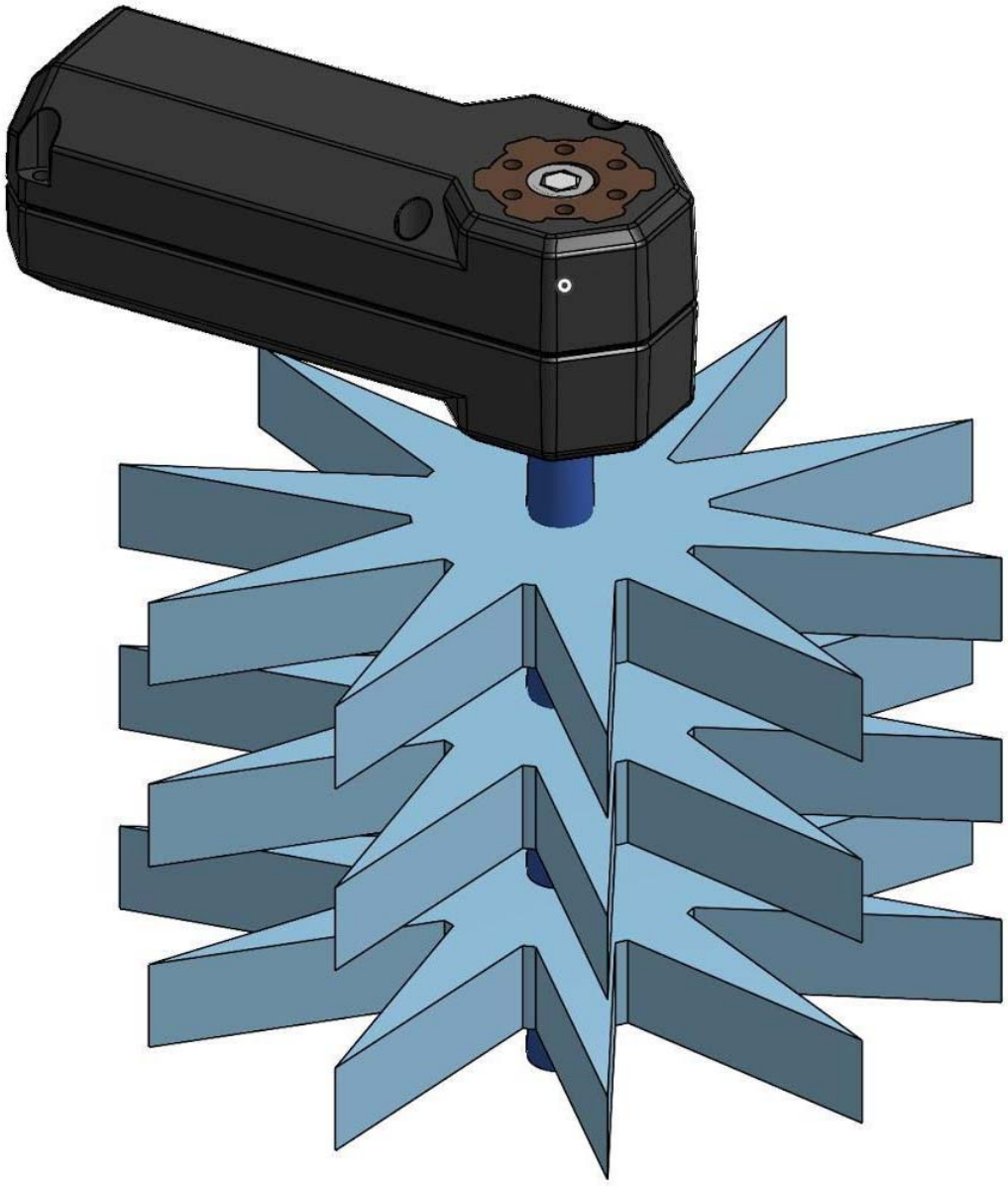
January 6- February 22	Acrylic box	The 3D printed box was too brittle and kept breaking when we screwed into it, so we decided on acrylic for the newer version of the box. The acrylic lets our drive team see in the box to see the minerals. The pieces of acrylic are held together with aluminum brackets



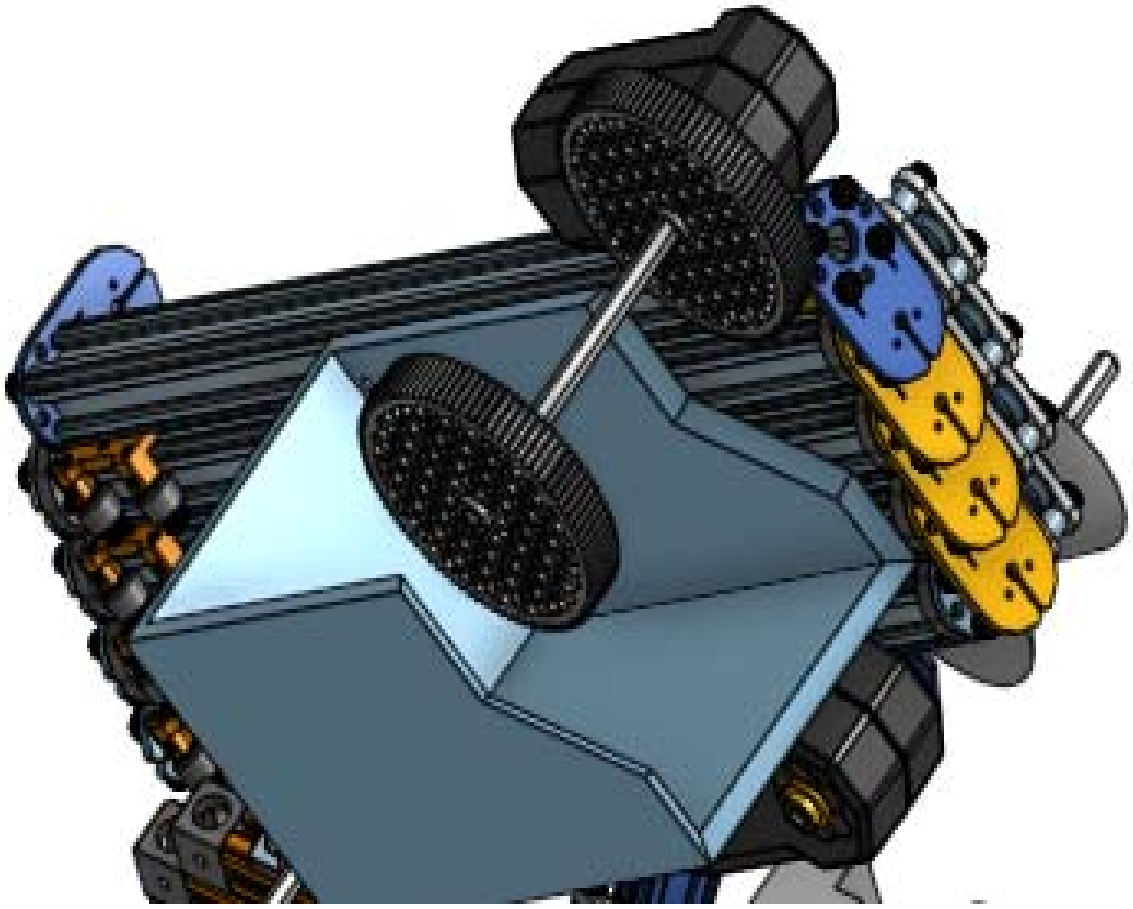
-Pulley-		
November 11- January 10	Hex spool	We CADed this spool to use for our linear arm with a hex hole pattern for the core hex motor
		
January 10- February 16	3in spool	We decided that the hex spool was too small and we could increase our efficiency by increasing the pulley diameter
		

February 16-February 22	2in spool	The 3in spool was too big and kept getting caught in the linear arm's flywheels so we designed the 2in to fix the problem while still increasing our speed.
		

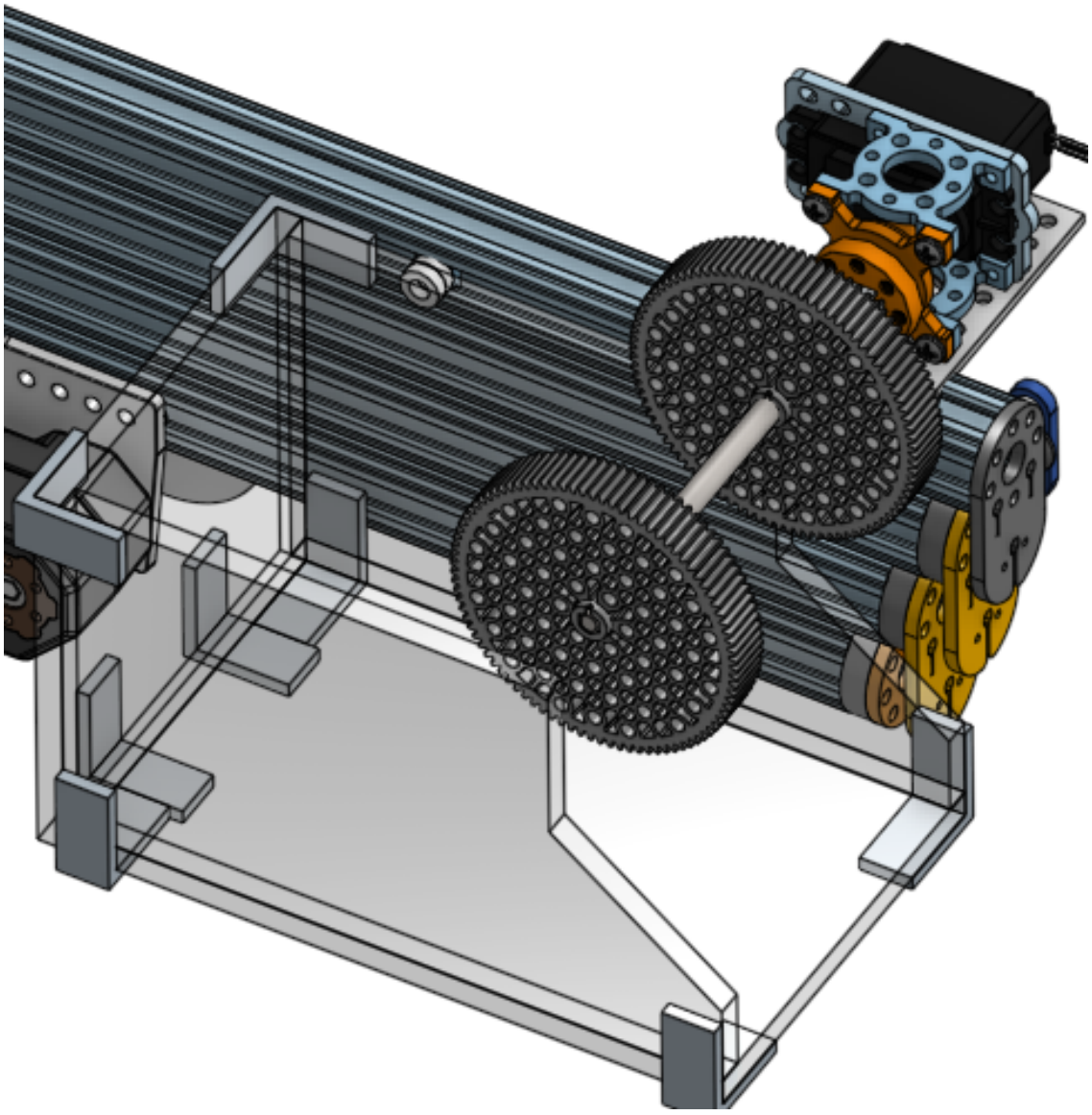
-Intake System-		
November 11-December 18	Entraption star intake	We did some research and found rev entraption stars on AndyMark, but to conserve budget money we decided to 3D print them instead, but the stars ended up being too thin and flimsy and couldn't grab onto minerals.

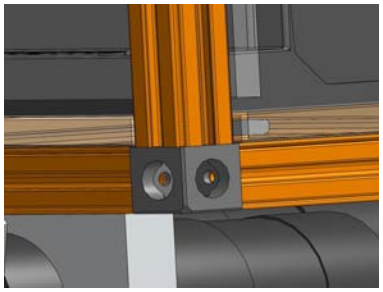
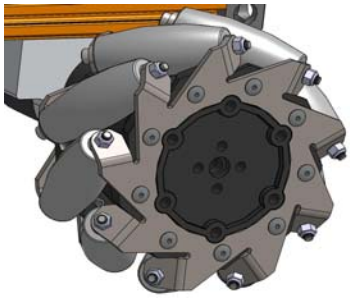
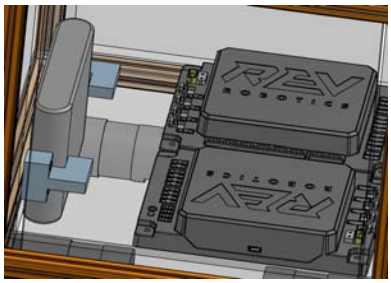
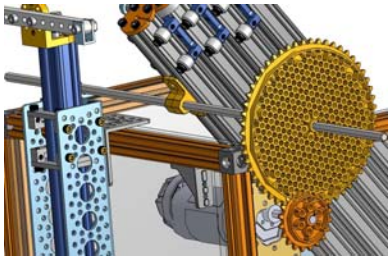


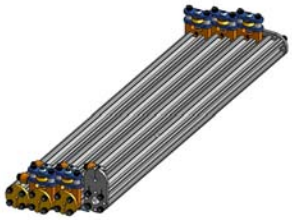
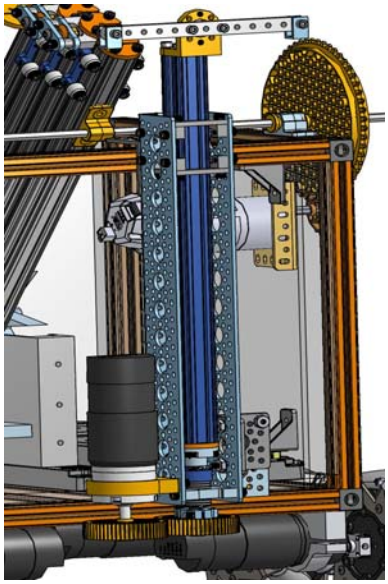
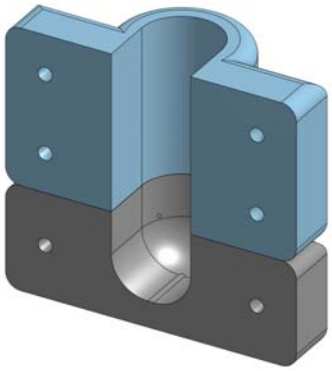
December 19- December 30	Core hex intake	To get our intake system spinning fast enough to pick up the minerals we used a core hex motor attached to a shaft and gears with rubber bands to pick up the minerals.

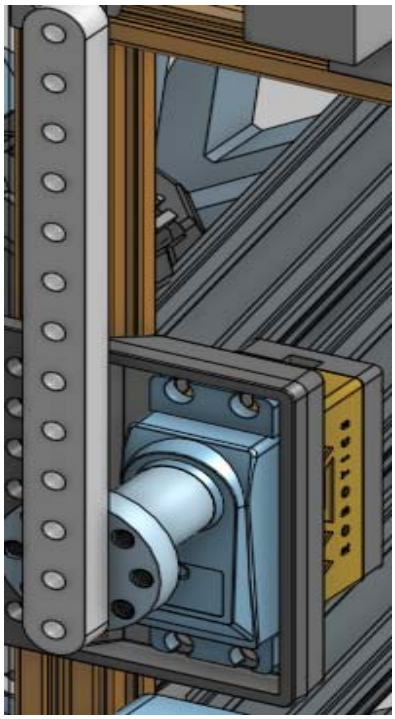
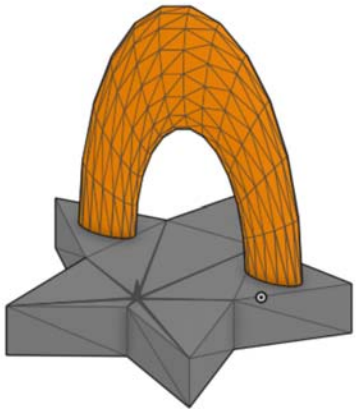
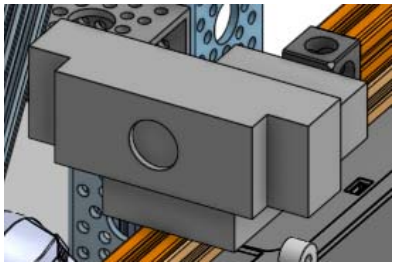


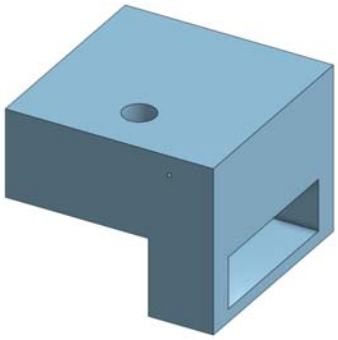
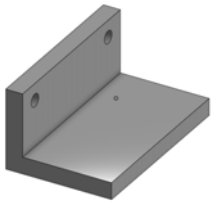
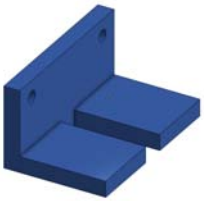
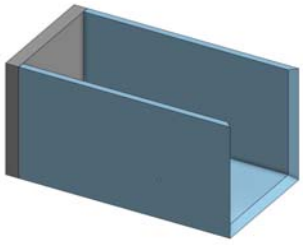
December 30-February 22	CR servo intake	Our arm was much too heavy to rotate up and score in the lander, so we used a CR servo instead of a motor. which took a lot of weight off the end of the slide. This allows us to rotate the arm up to the lander. The gears and rubber bands are the same as the previous system.

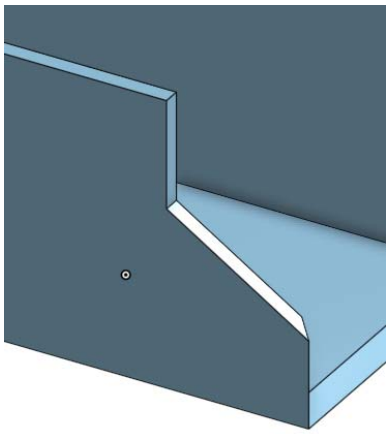
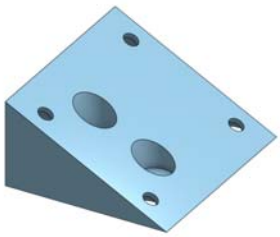
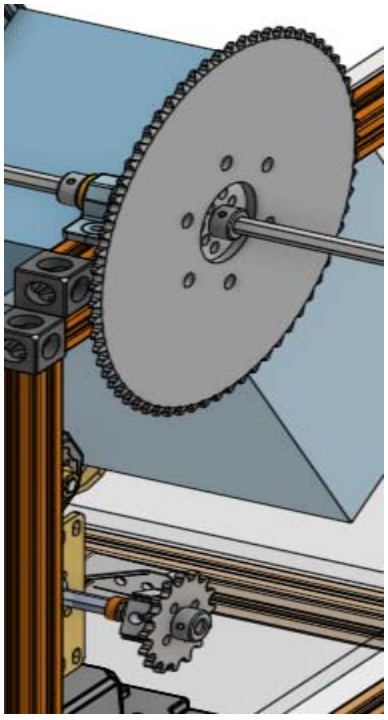


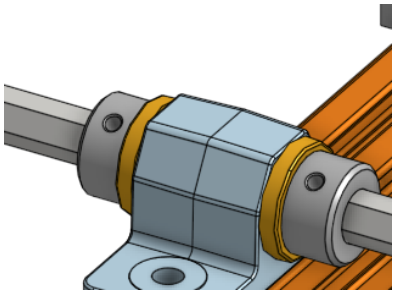
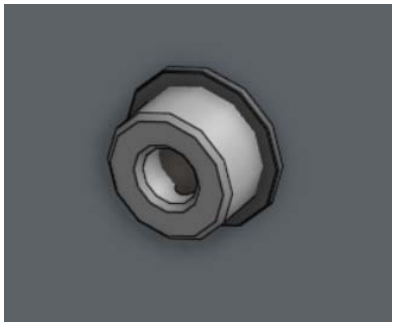
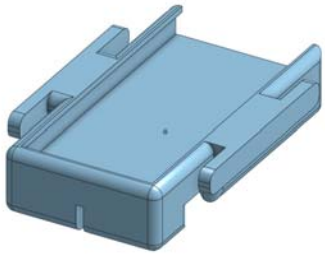
-Other CAD Notes-		
Goals for Design Session	Progress	Photo
Assemble x-rail and square corner base	Mark worked on assembling the base on November 19. We finished this and now we have a complete 10 x 10 x-rail cube	
Group mecanum wheels	We figured out that you must highlight all the parts and click the group button on the toolbar. This allows you to move the whole wheel without moving individual parts of the wheel	
Attach battery, hubs, phone mount to the base	We made sure there was an allowance for cables to and from the hubs and battery during the design. We made our own battery holder with the acrylic and “hooks”. We also kept the idea of easy access in mind while placing the design.	
Attach sprocket and shaft system that rotates the linear slide	We have a small to big ratio to have a lot of torque for the weight of the arm.	

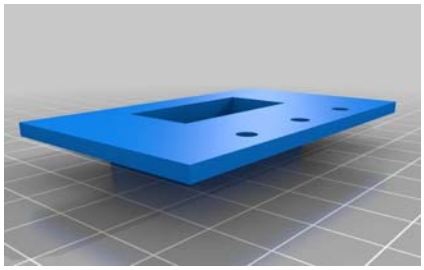
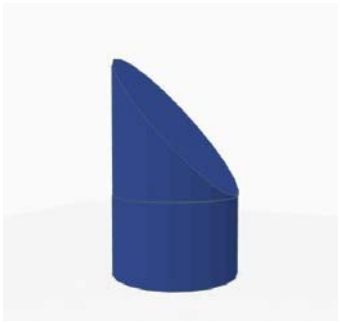
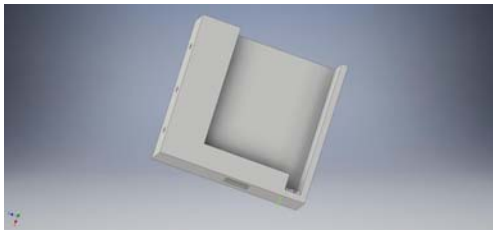
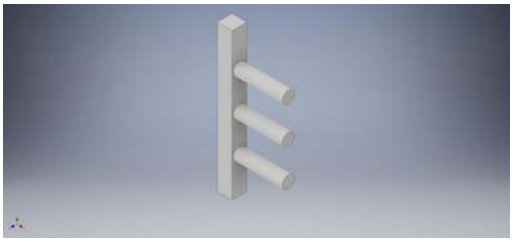
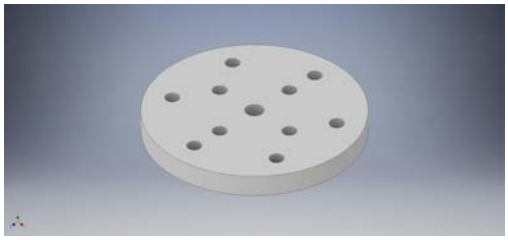
Assemble linear slide	We were able to get a preassembled CAD drawing off Servocity with three stages. We had to add a fourth it is now attached to the intake system and the shaft and gear system.	
Assemble linear actuator	Drew worked on assembling the linear actuator 11/20 pm he got it almost totally assembled except for the motor in the motor mount. This would allow for the linear actuator to be added to the assembly	
Flag Holder	While drawing we made it excessively big because we've had problems with a flag being too big for the holder. We can tape the top if need be.	

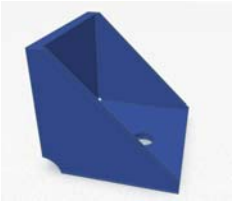
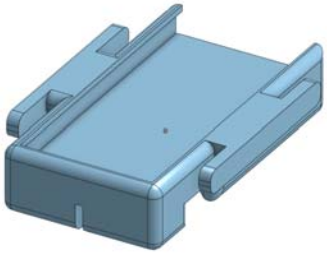
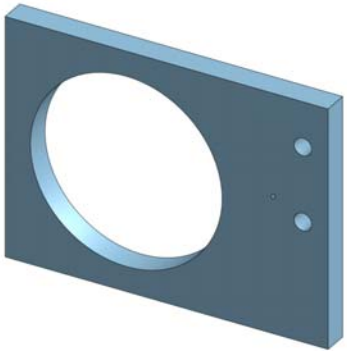
Player Marker Delivery System	We used a rev smart servo with mini channel attached to put our marker on the channel. We will rotate the servo and our marker will slide off	
CAD new marker	We wanted a marker with a hole and have a star so we made a rainbow on a star and will attach our all famous duck to it.	
CAD items for TensorFlow (camera, USB hub, battery pack)	Drew CADed the camera, battery pack, and USB hub in Inventor and imported them then assembled them in the G2 document	

Camera Mount	The camera wasn't secure so we made a block to stabilize it. It mounts on the REV Rail with a screw that can be inserted into the camera. The cut out is so the T-screws can have a nut to secure the mount.	
Phone Holder	We made a new phone holder so the phone can mount straight to the polycarbonate.	
Wire Cradles	The cradles hold the wire in place so it does not fall out mid game. There are two different versions depending on whether we have a straight wire or the wire is at a ninety-degree angle.	 
Mineral Box	We made the parts to hold only two minerals. Both of the parts will be connected by a servo. The box is how we will score.	

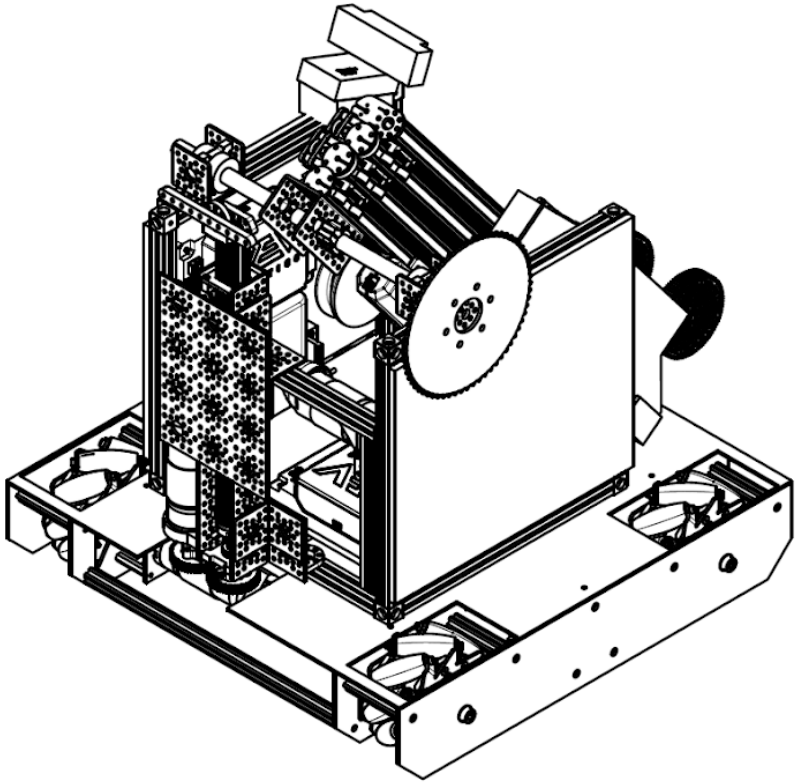
Cutting a corner off	We wanted to make the mineral box closer to the intake so we cut off part of the corner of the box.	
Making a mount for the caster wheel	The wheel and the Actobotics plate have different patterns so we made an adapter. It is also slanted to help angle the wheel.4	
Change sprockets	We imported new sprockets from andymark that are metal and give more torque to lift our arm.	

Add collars	To keep our pillow blocks from sliding, we added collars with rev screws.	
Add Collars to Wheelbase	To keep our axles from sliding, we put collars on the outside of the base and now the axles slide a lot less.	
New Driver Station Holder.	Cables were getting tangled while driving so we created a holder so the coach won't have as hard of a time holding everything.	
Research ways to put chain and rubber bands on the robot.	Our robot has chain on the sprockets and rubber bands on the intake so we needed to find ways to put those in the CAD, we did some work, working toward this goal, but we are still lacking knowledge of how to put them in the CAD.	

-Other CAD Designs-	
CAD Item Used on Bot/Designed By Team	**Photo of Item**
This is a power switch holder found on Thingiverse made by olstweartrobots. It is used to hold and attach the power switch in an easy access and visible place.	
This is a design by Mark that is not being used anymore after the new base.	
This is the old phone holder designed by Drew	
This is a component for the intake attachment that holds nerf darts on the ends of the rods. This was made by Drew.	
This is a wheel intake adapter plate to help spin piece above on a servo that was made by Drew.	

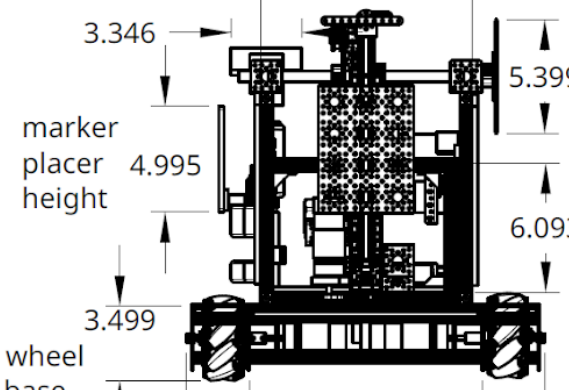
This is a bracket cap designed by Drew to fit on the Actobots brackets.	
These corner caps created by Mark help make our robot safer.	
The corners were to protect people from sharp edges and keep the edge from flexing which were created by Mark.	
The driver station holder is used to help keep cords out of the way while driving. Created by Mark.	
The motor mount is to stabilize the intake arm motor which is by Mark.	

Washi 3.5

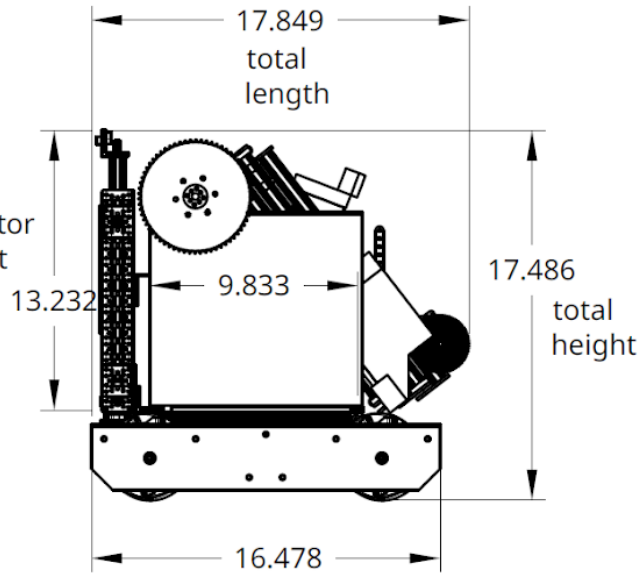


DWG NO.	SIZE	SCALE	SHEET	REV.
1	A	1:5	1 of 3	

Front view



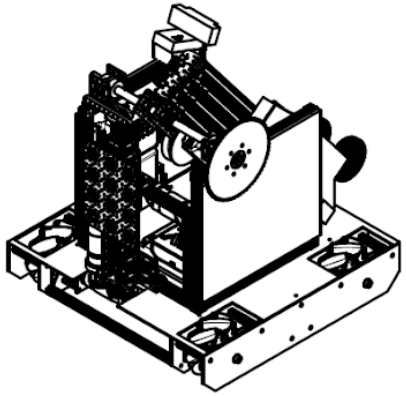
right view

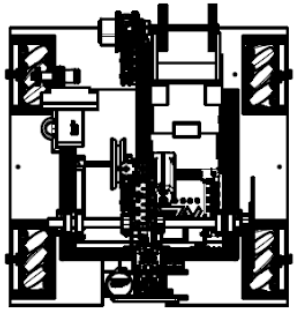


UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- .XXXX = ±.000- ANGULAR = ± ° FRACTIONAL = ± SURFACE FINISH		NAME	DATE
	DRAWN	DREW BUSCH	01/29/2019
	CHECKED		
	APPROVED	---	---
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

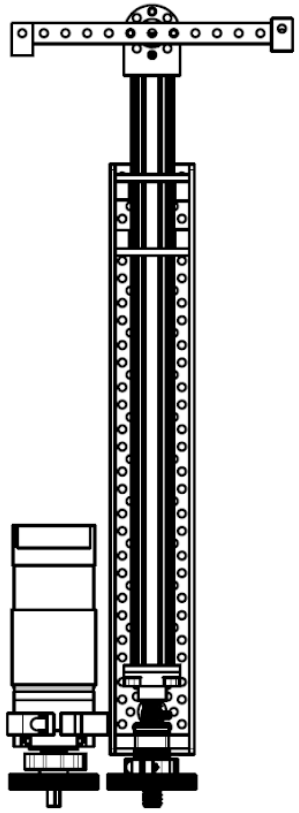
TITLE	Overall Design	
SIZE	DWG NO. 1	REV.
SCALE 1:8	WEIGHT	SHEET 2 of 3

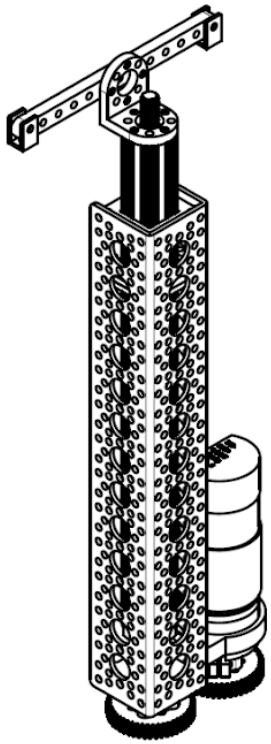
CAD Page 29





DWG NO.	SIZE	SCALE	SHEET	REV.
1		1:10	3 of 3	



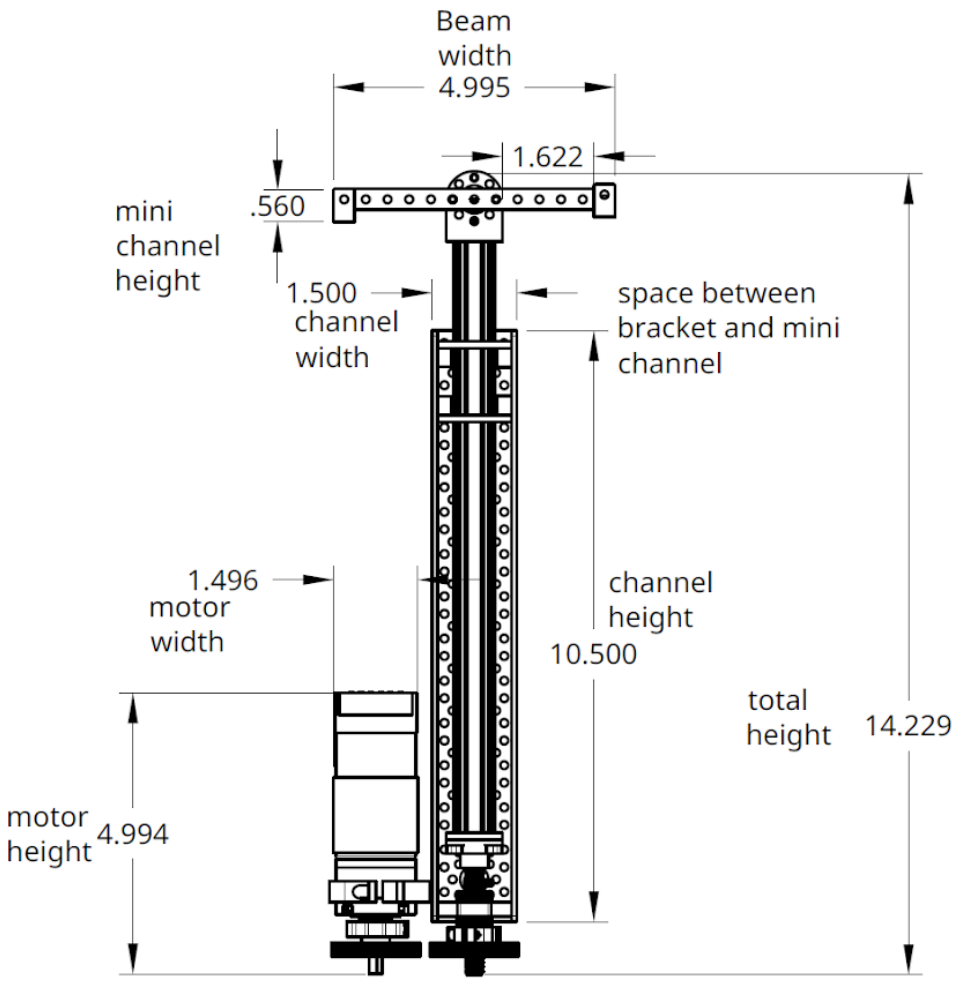


UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- ANGULAR = ± ° .XXXX = ±.000- FRACTIONAL = ± SURFACE FINISH		NAME	DATE
	DRAWN	DREW BUSCH	02/08/2019
	CHECKED	Mark Hazen	2/19/19
	APPROVED	Kristi Hazen	2/19/19
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

TITLE: Linear Actuator

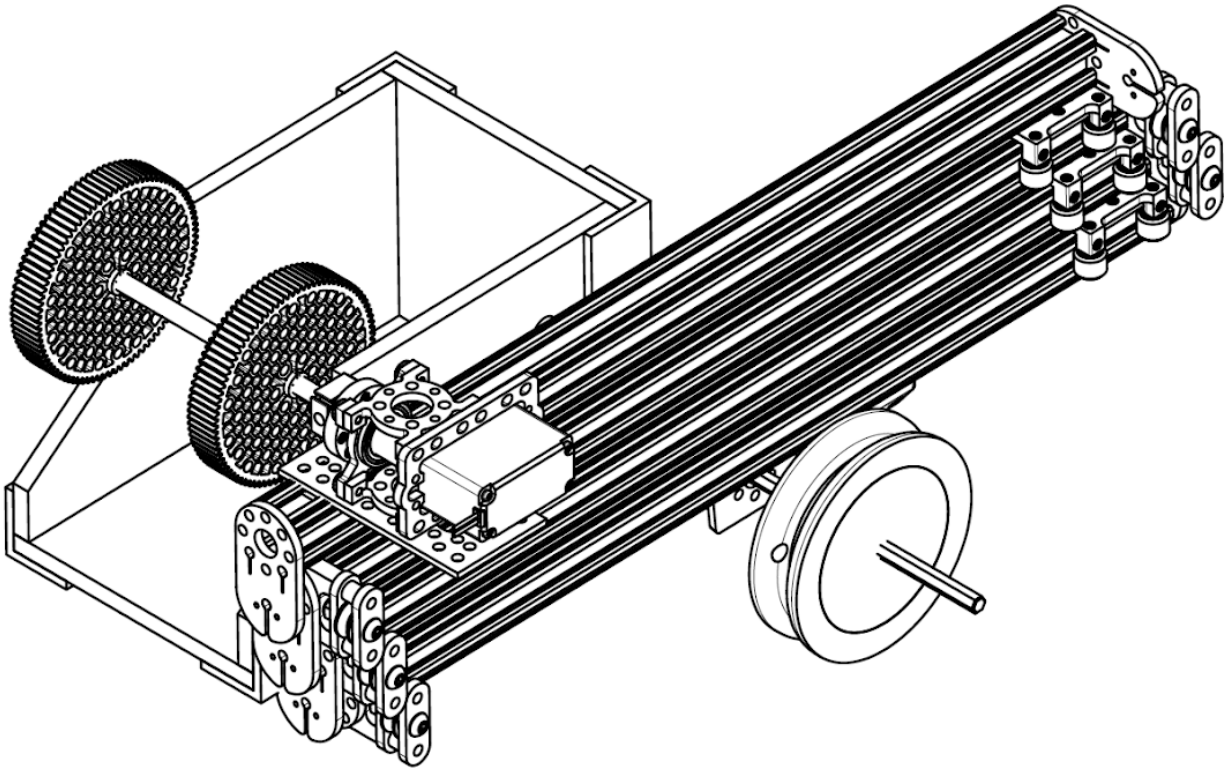
SIZE | DWG NO. 2 | REV.

SCALE 1:3 | WEIGHT | SHEET 1 of 2



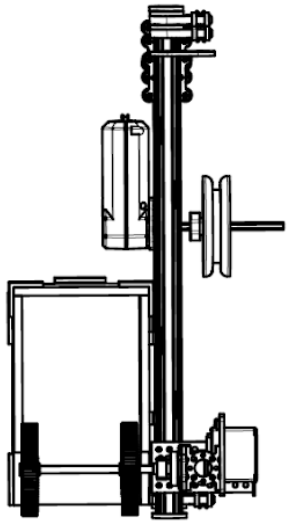
DWG NO.	SIZE	SCALE	SHEET
2		1:3	2 of 2

Linear Slide

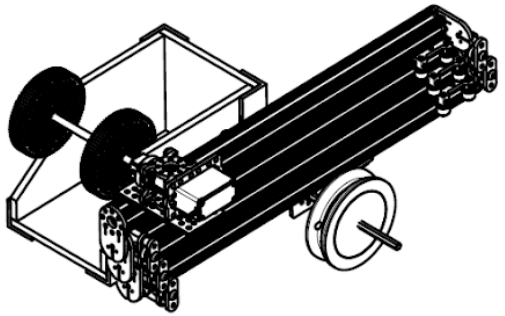


DWG NO.	SIZE	SCALE	SHEET	REV.
3	A	1:2	1 of 5	-

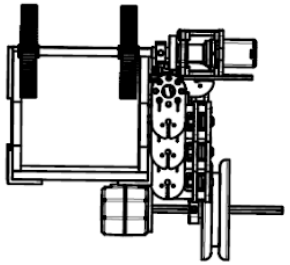
Top



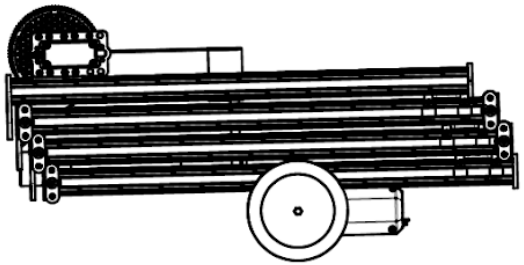
Isometric



Front



Right



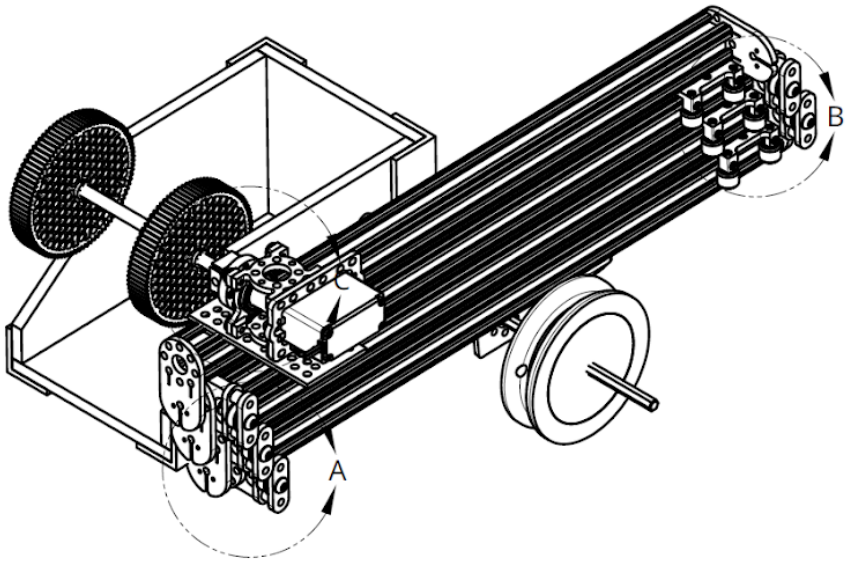
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- ANGULAR = ± ° .XXXX = ±.000- FRACTIONAL = ± SURFACE FINISH		NAME	DATE
	DRAWN	DREW BUSCH	02/04/2019
	CHECKED	Mark Hazen	9/19/19
	APPROVED	Kristi Hazen	2/19/19
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

TITLE: Linear Slide

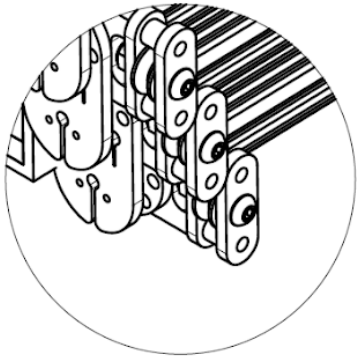
SIZE | DWG NO. 3 | REV.

SCALE 1:5 | WEIGHT | SHEET 2 of 5

Isometric view



front bearing system

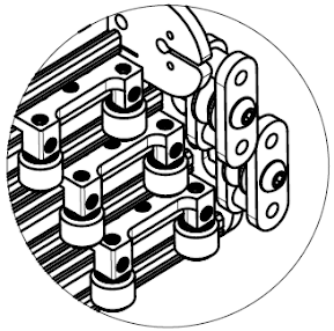


DETAIL A
SCALE 2:3

servo mount and servo block

DETAIL C
SCALE 2:3

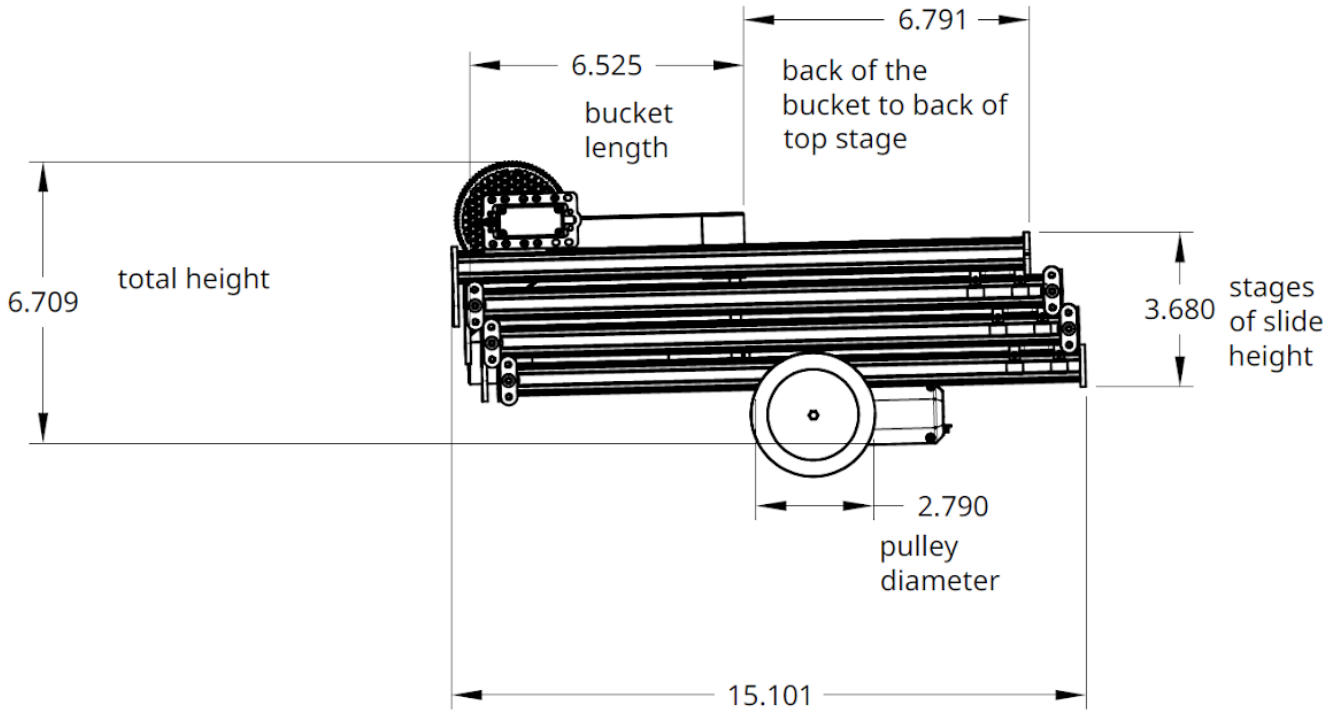
rear bearing and slide rollers system



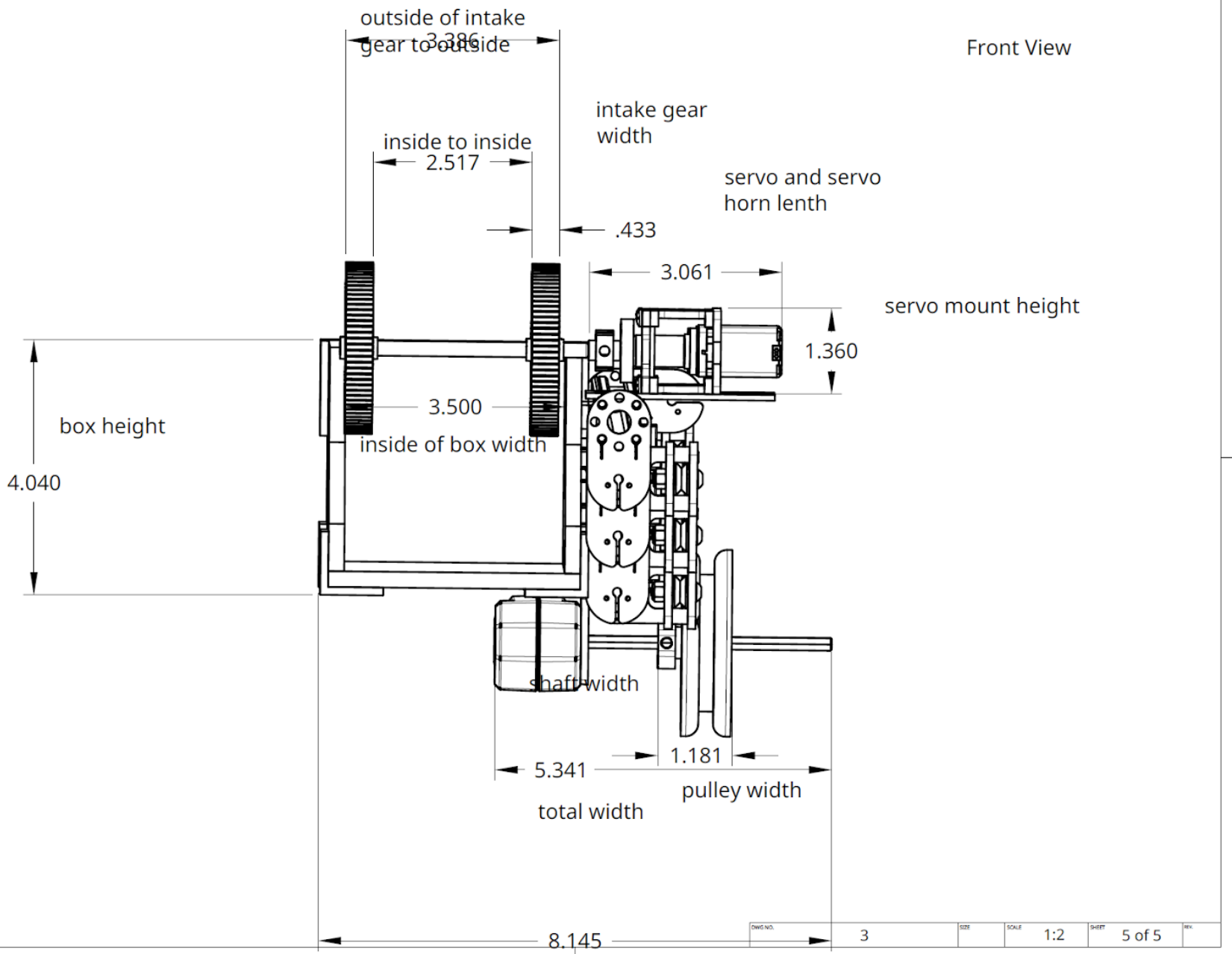
DETAIL B
SCALE 2:3

DWG NO.	SIZE	SCALE	SHEET	REV.
3		1:3	3 of 5	

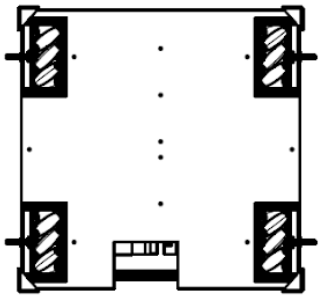
Right View



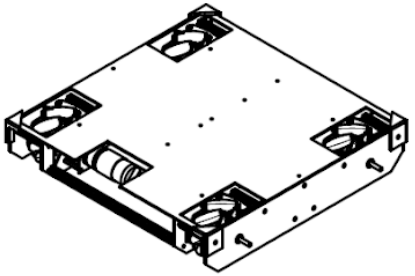
DWG NO.	SIZE	SCALE	SHEET	REV
3		1:4	4 of 5	


Front View
outside of intake gear to outside
3.386
intake gear width
inside to inside
2.517
servo and servo horn lenth
.433
3.061
servo mount height
1.360
box height
4.040
3.500
inside of box width
shaft width
1.181
5.341
pulley width
total width
8.145
DWG NO.
3
SIZE
SCALE
1:2
SHEET
5 of 5
REV.

top view

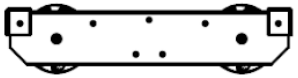


isometric view



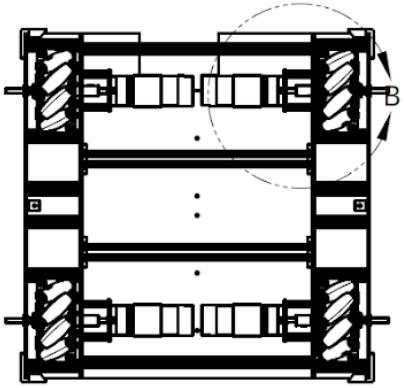
front view

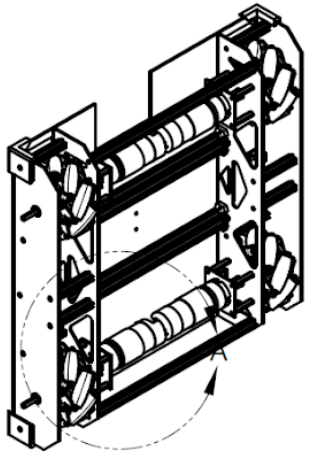
side view

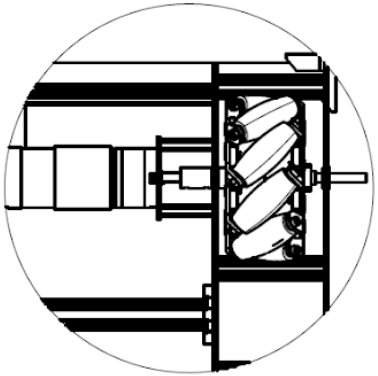


UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- .XXXX = ±.000- ANGULAR = ± ° FRACTIONAL = ± SURFACE FINISH		NAME	DATE
	DRAWN	Mark Hazen	01/29/2019
	CHECKED	Drew Busch	2/18/19
	APPROVED	Kristi Hazen	2/19/19
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

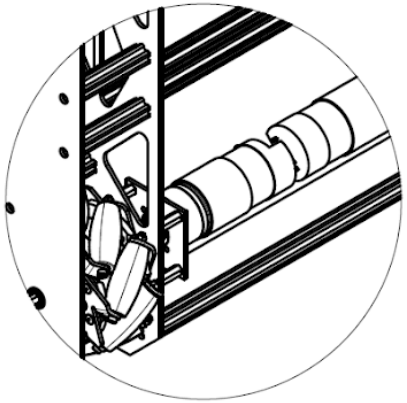
TITLE	Wheel Base	
SIZE	DWG NO. 4	REV.
SCALE 1:10	WEIGHT	SHEET 1 of 3





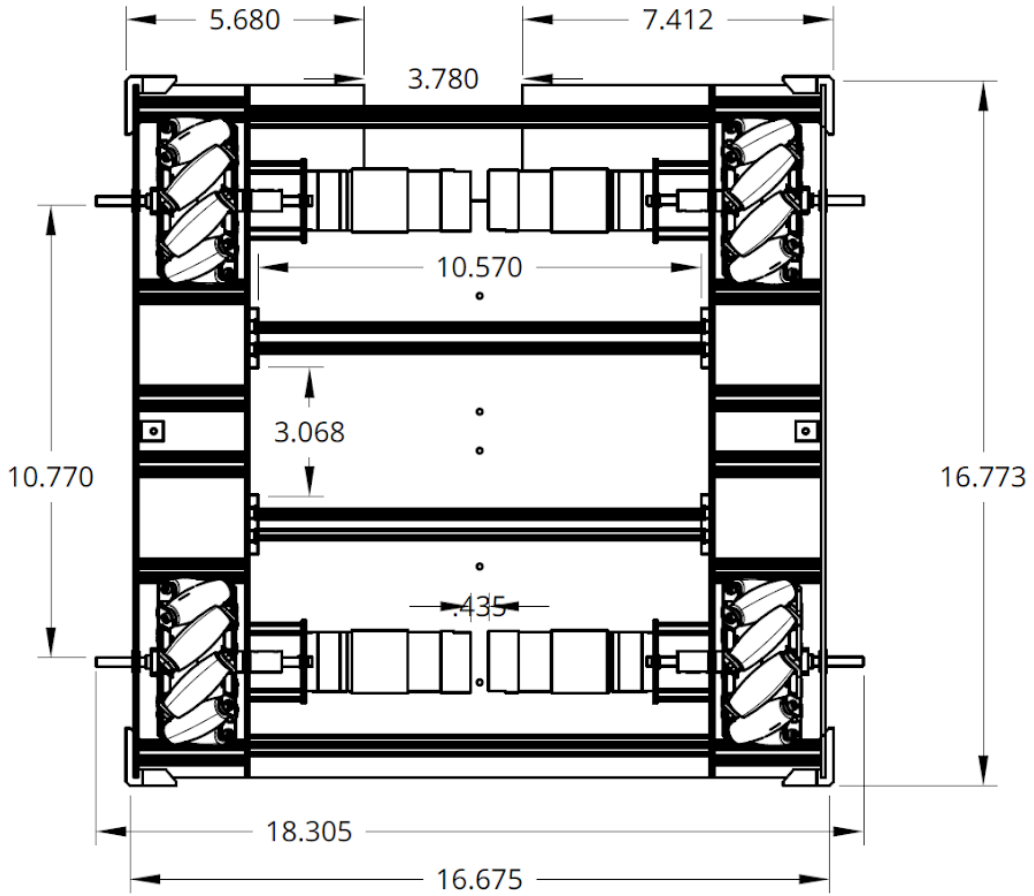


DETAIL B
SCALE 1:4

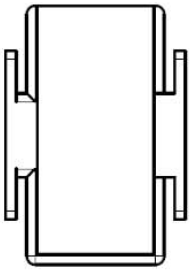


DETAIL A
SCALE 1:4

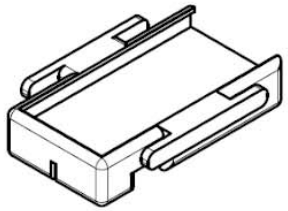
DWG NO.	SIZE	SCALE	SHEET	REV.
4		1:8	2 of 3	



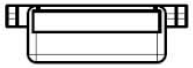
DWG NO.	SIZE	SCALE	SHEET	REV.
4		1:4	3 of 3	



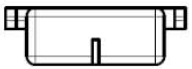
Top



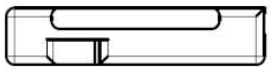
Isometric



Front



Back



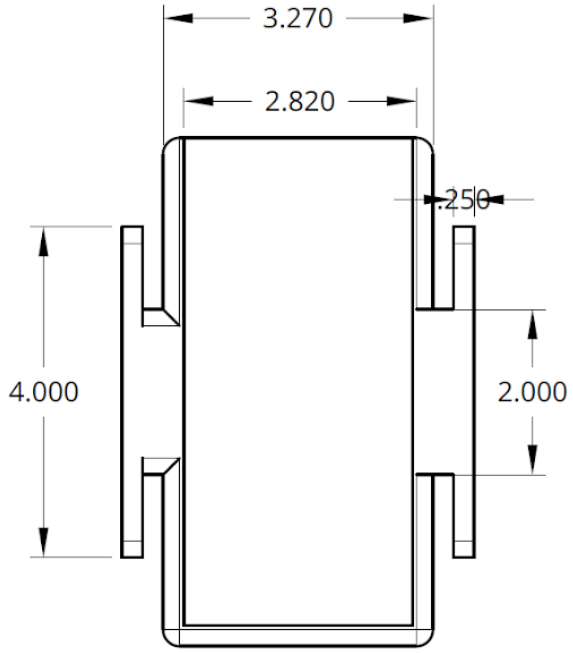
Side

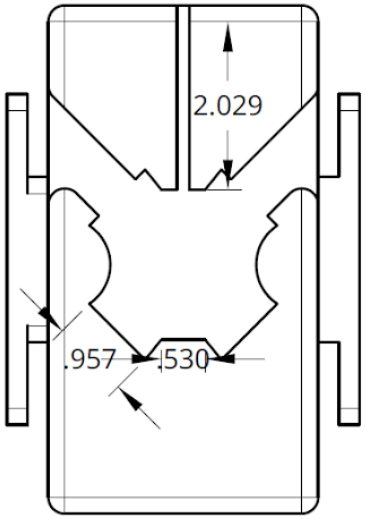
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- .XXXX = ±.000- ANGULAR = ± ° FRACTIONAL = ± SURFACE FINISH √		NAME	DATE
	DRAWN	DREW BUSCH	02/19/2019
	CHECKED	Mark Hazen	2/19/19
	APPROVED	Kristi Hazen	2/19/19
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

TITLE: Diver Station Holder

SIZE	DWG NO.	REV.
	5	

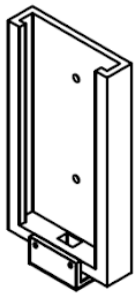
SCALE	WEIGHT	SHEET
1:4		1 of 2



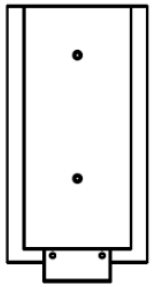


DWG NO.	SIZE	SCALE	SHEET
5		1:2	2 of 2

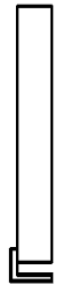
Top



Isometric



Front



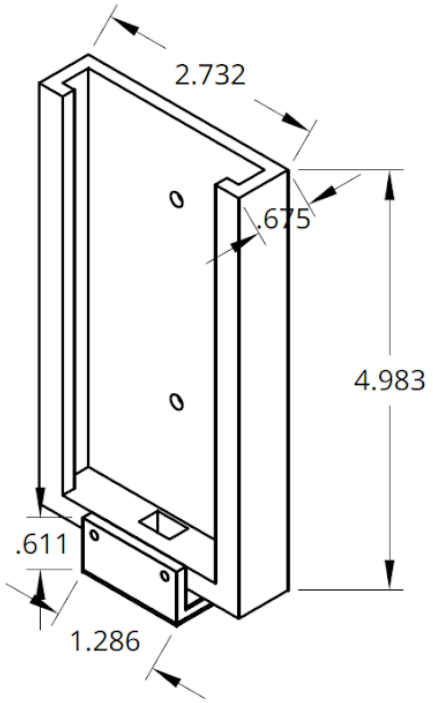
Side

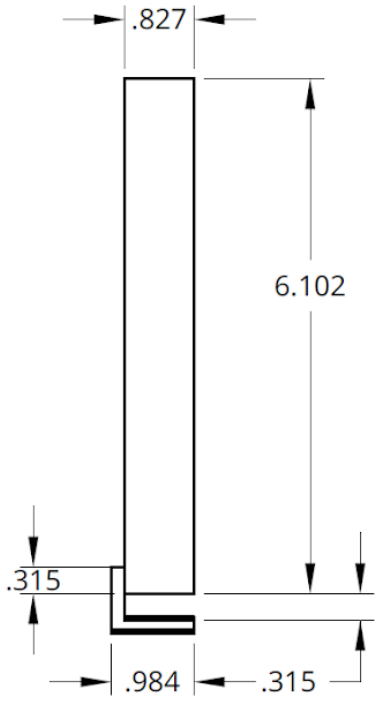
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- .XXXX = ±.000- ANGULAR = ± ° FRACTIONAL = ± SURFACE FINISH √		NAME	DATE
	DRAWN	DREW BUSCH	02/19/2019
	CHECKED	Mark Hazen	2/19/19
	APPROVED	Kristi Hazen	2/19/19
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

TITLE: Phone Holder

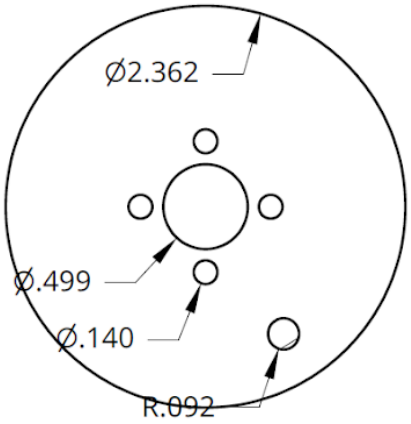
SIZE | DWG NO. 6 | REV.

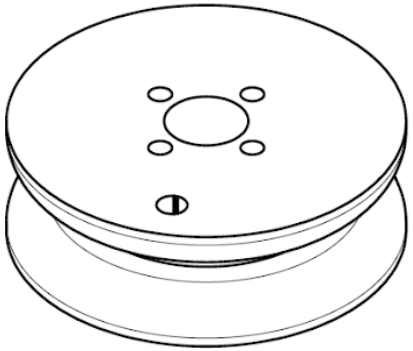
SCALE 1:4 | WEIGHT | SHEET 1 of 2

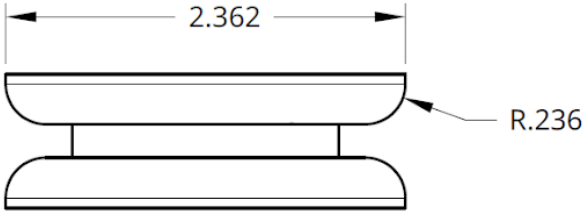


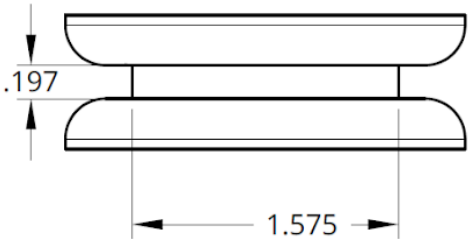


DWG NO.	SIZE	SCALE	SHEET	REV.
6		1:2	2 of 2	



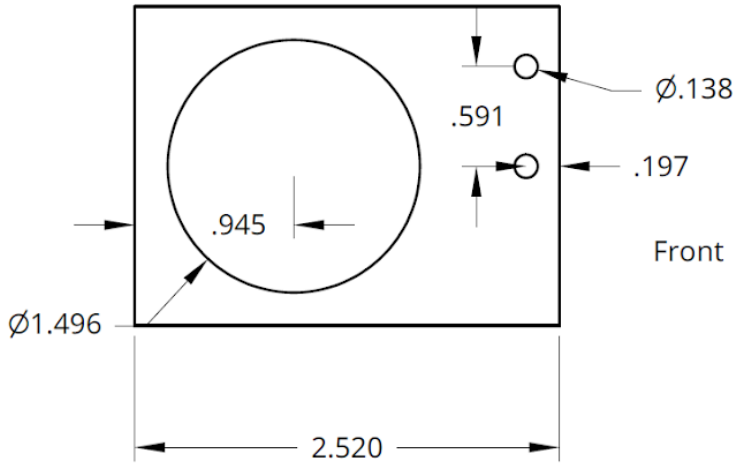


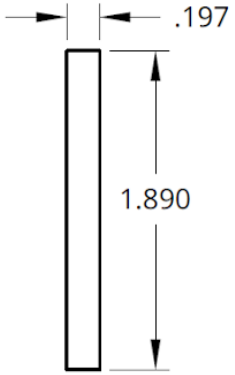


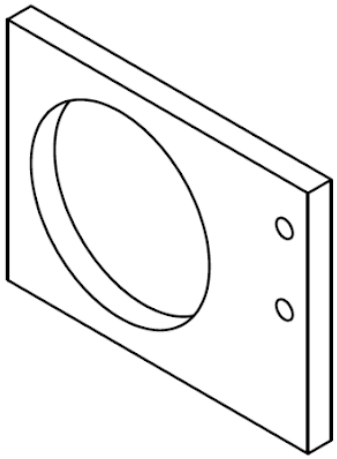


UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- .XXXX = ±.000- ANGULAR = ± ° FRACTIONAL = ± SURFACE FINISH		NAME	DATE
	DRAWN	DREW BUSCH	02/19/2019
	CHECKED	Mark Hazen	2/19/19
	APPROVED	Kristi Hazen	2/19/19
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

TITLE	Linear Slide Spool	
SIZE	DWG NO. 7	REV.
SCALE 1:1	WEIGHT	SHEET 1 of 1







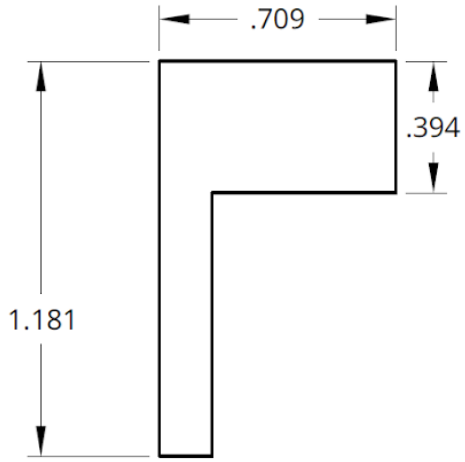
Isometric

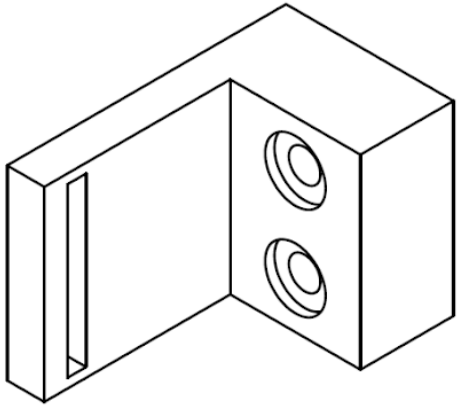
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- .XXXX = ±.000- ANGULAR = ± ° FRACTIONAL = ± SURFACE FINISH		NAME / DATE
	DRAWN	Mark Hazen 02/19/2019
	CHECKED	Drew Busch 2/19/19
	APPROVED	Kristi Hazen 2/19/19
DO NOT SCALE DRAWING		
BREAK ALL SHARP EDGES AND REMOVE BURRS		
THIRD ANGLE PROJECTION	MATERIAL	FINISH

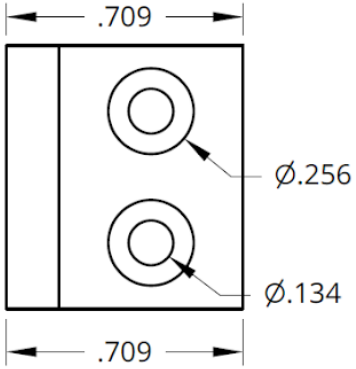
TITLE: Linear Slide Motor Mount

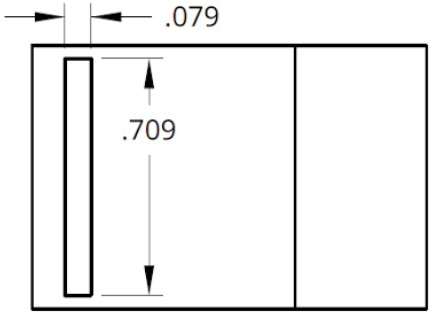
SIZE | DWG NO. 8 | REV.

SCALE 1:1 | WEIGHT | SHEET 1 of 1





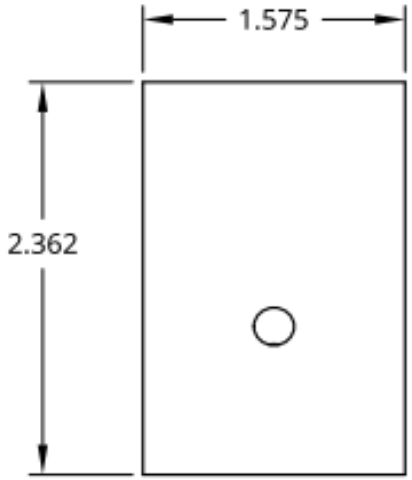


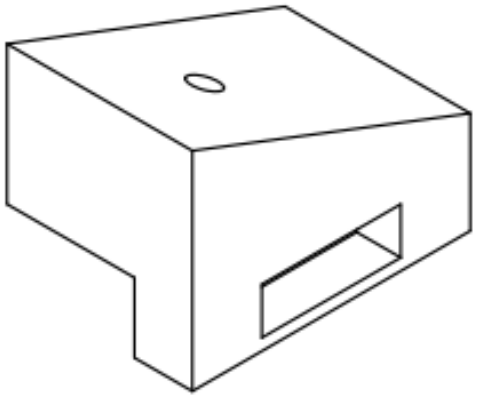


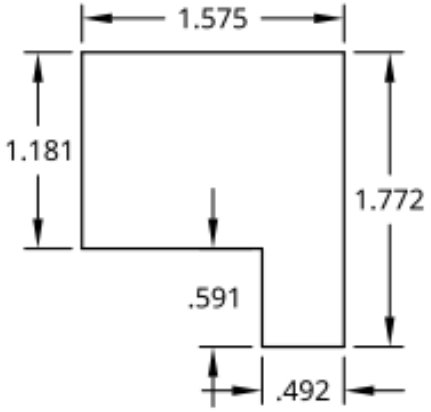
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- .XXXX = ±.000- ANGULAR = ± ° FRACTIONAL = ± SURFACE FINISH		NAME	DATE
	DRAWN	DREW BUSCH	02/19/2019
	CHECKED	Mark Hazen	2/19/19
	APPROVED	Kristi Hazen	2/19/19
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

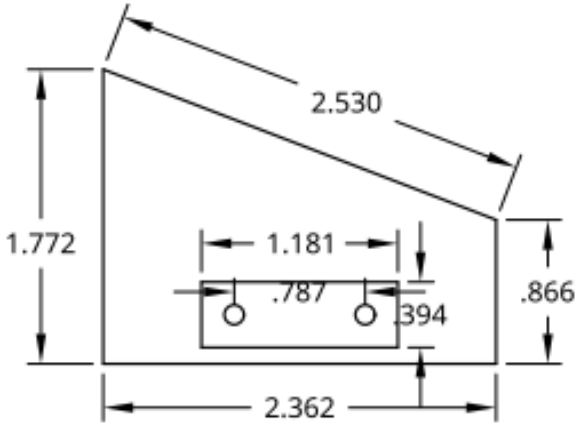
TITLE	Battery Clip	
SIZE	DWG NO. 9	REV.
SCALE 2:1	WEIGHT	SHEET 1 of 1

Camera Mount



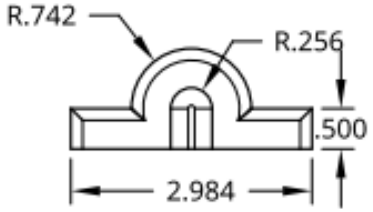


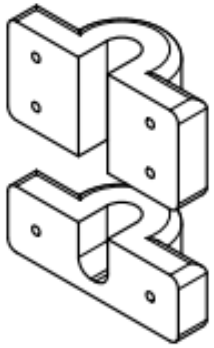


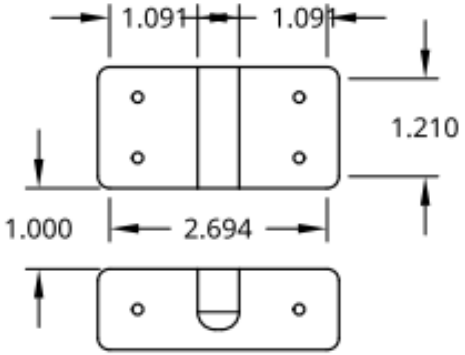


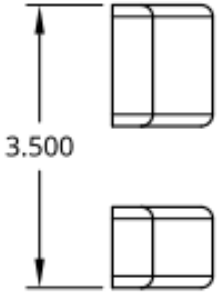
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- .XXXX = ±.000- ANGULAR = ±° FRACTIONAL = ±		NAME	DATE
SURFACE FINISH √	DRAWN	DREW BUSCH	02/15/2019
	CHECKED		
	APPROVED	---	---
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

TITLE	---- ---- ----	
SIZE A	DWG NO. ----	REV. -
SCALE 1:1	WEIGHT	SHEET 1 of 1









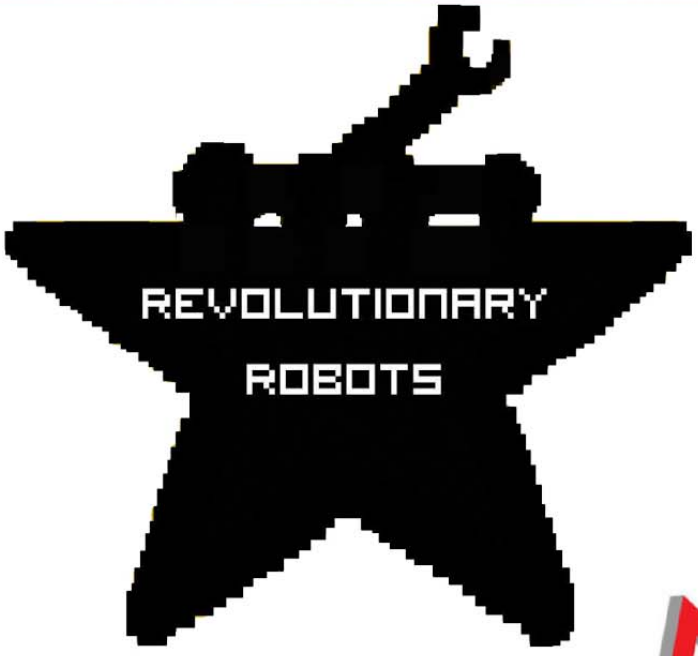
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES .XX = ±.0- .XXX = ±.00- .XXXX = ±.000- ANGULAR = ±° FRACTIONAL = ±		NAME	DATE
	DRAWN	DREW BUSCH	02/15/2019
	CHECKED		
SURFACE FINISH √	APPROVED	---	---
DO NOT SCALE DRAWING			
BREAK ALL SHARP EDGES AND REMOVE BURRS			
THIRD ANGLE PROJECTION	MATERIAL	FINISH	

TITLE: ----

SIZE: A	DWG NO. ----	REV. -
SCALE: 1:2	WEIGHT	SHEET: 1 of 1

Revolutionary Robots 12535

Programming 2018-2019



Moore Community STEM Club

FIRST®

Driver One

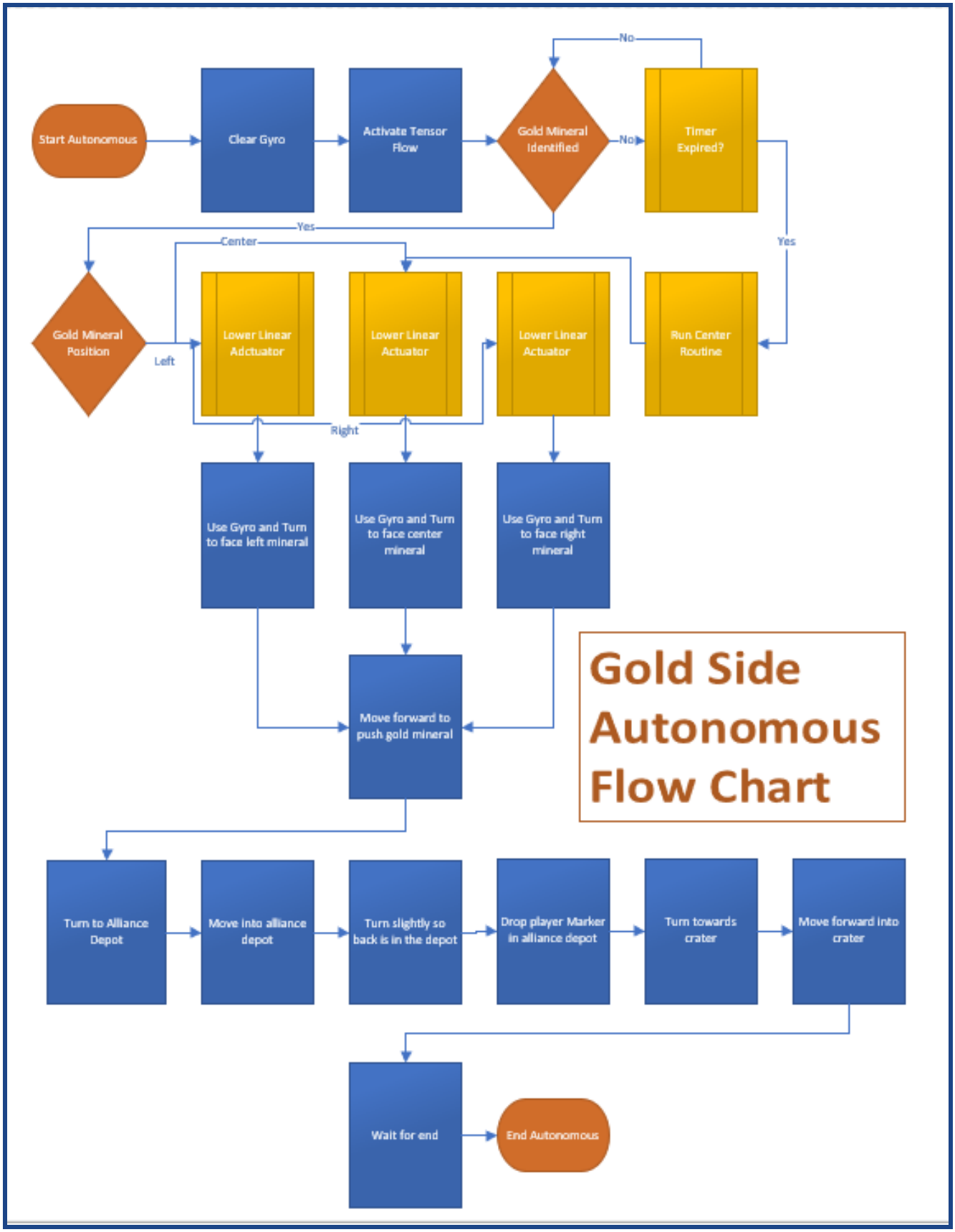
	NAME	DATE
DRAWN	Drew Busch	2/09/19
CHECKED	Mark Hazen	2/18/19
APPROVED	Kristi Hazen	2/18/19

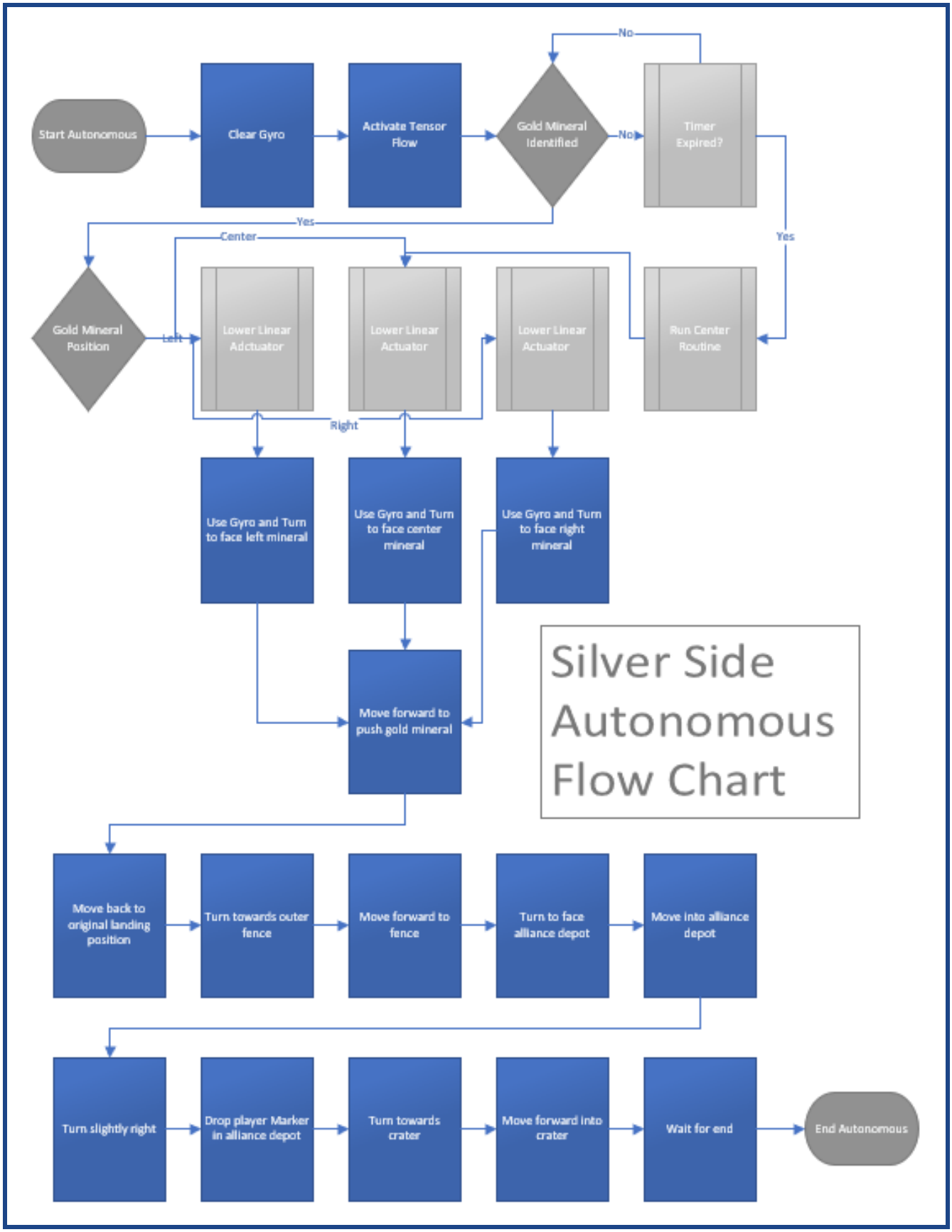
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES
.XX = ±.0-
.XXX = ±.00-
.XXXX = ±.000-
ANGULAR = ± °
FRACTIONAL = ±
SURFACE FINISH
DO NOT SCALE DRAWING
BREAK ALL SHARP EDGES AND REMOVE BURRS
THIRD ANGLE PROJECTION
MATERIAL: Plastic with Wires
FINISH

TITLE: Controller Layout
SIZE | DWG NO. | REV.
SCALE | WEIGHT | SHEET 1 of 2

Driver Two

2 of 2


Start Autonomous
Clear Gyro
Activate Tensor Flow
Gold Mineral Identified
No
Timer Expired?
No
Yes
Yes
Center
Gold Mineral Position
Left
Lower Linear Adctuator
Lower Linear Actuator
Lower Linear Actuator
Run Center Routine
Right
Use Gyro and Turn to face left mineral
Use Gyro and Turn to face center mineral
Use Gyro and Turn to face right mineral
Move forward to push gold mineral
Gold Side Autonomous Flow Chart
Turn to Alliance Depot
Move into alliance depot
Turn slightly so back is in the depot
Drop player Marker in alliance depot
Turn towards crater
Move forward into crater
Wait for end
End Autonomous


Start Autonomous
Clear Gyro
Activate Tensor Flow
Gold Mineral Identified
No
Timer Expired?
Yes
Yes
Center
Gold Mineral Position
Left
Lower Linear Adctuator
Lower Linear Actuator
Lower Linear Actuator
Run Center Routine
Right
Use Gyro and Turn to face left mineral
Use Gyro and Turn to face center mineral
Use Gyro and Turn to face right mineral
Move forward to push gold mineral
Silver Side Autonomous Flow Chart
Move back to original landing position
Turn towards outer fence
Move forward to fence
Turn to face alliance depot
Move into alliance depot
Turn slightly right
Drop player Marker in alliance depot
Turn towards crater
Move forward into crater
Wait for end
End Autonomous

November 2nd, 2018
Mark Hazen

Goals for Programming	Progress on Goals
Deploying	When we first started the linear actuator wouldn't stop with encoders. We started testing with motors separately to troubleshoot the problem. When using the motors separate the encoders worked so it was a hardware issue. With more digging we found that the encoder wasn't working, so we changed the motor out with another REV Hex 40:1 motor.

November 4th, 2018
Mark Hazen

Goals for Programming	Progress on Goals
Sampling	We first started with RGB, red blue green, values, but they were inconsistent. We switched to hues to be more consistent. Now if the mineral is in 10cm of the sensor it gives a consistent hue.

November 9th, 2018
Mark Hazen

Goals for Programming	Progress on Goals
Finish Autonomous	Decided to go for a 1/3rd chance to sample one mineral because we are having a difficult time moving sideways. We believe this may be because our mecanum wheels are not handling the weight of our robot well. Insufficient time to test/fix this before the competition. We completed dropping our player marker in the alliance depot and parked in the crater. It was very easy to program because we are using encoders

December 9th, 2018
Mark Hazen

Goals for Programming	Progress on Goals
Getting the new bot with wheels to move.	Android Studio wasn't working because of the "Thread Compiler", so we switched to OnBot Java for the night while the children were there. While programing we left out "extends OpMode" so we spent a while troubleshooting. We had the robot running at the end of the night.

December 15th, 2018
Mark Hazen

Goals for Programming	Progress on Goals
Fixing Android Studio	After checking GitHub and the FTC SDK we figured out that it had updated. Now Android Studio works properly.
Setting up "TensorFlow"	We used the sample in the programming samples for "TensorFlow" and it worked very well. We use a camera connected to the phone.

December 30th, 2018
Mark Hazen

Goals for Programming	Progress on Goals
Make a preliminary code	Because we had no moving robot we made a preliminary code to get ready for autonomous.

January 3, 2019
Mark Hazen

Goals for Programming	Progress on Goals
Get the robot to work in TeleOp	While testing we added in toggle to set the linear slide's power to either 0.5 or 1. Then we had a problem with the intake not working, but we figured it out. Continuous rotation servos can only be set to -0.5 to 0.5 not to 1.

Adjusting Autonomous	The encoder on the Linear Actuator was not working. We checked hub three and found that the encoder wire was plugged into the wrong port.

January 4, 2019
Mark Hazen and Andy Miller

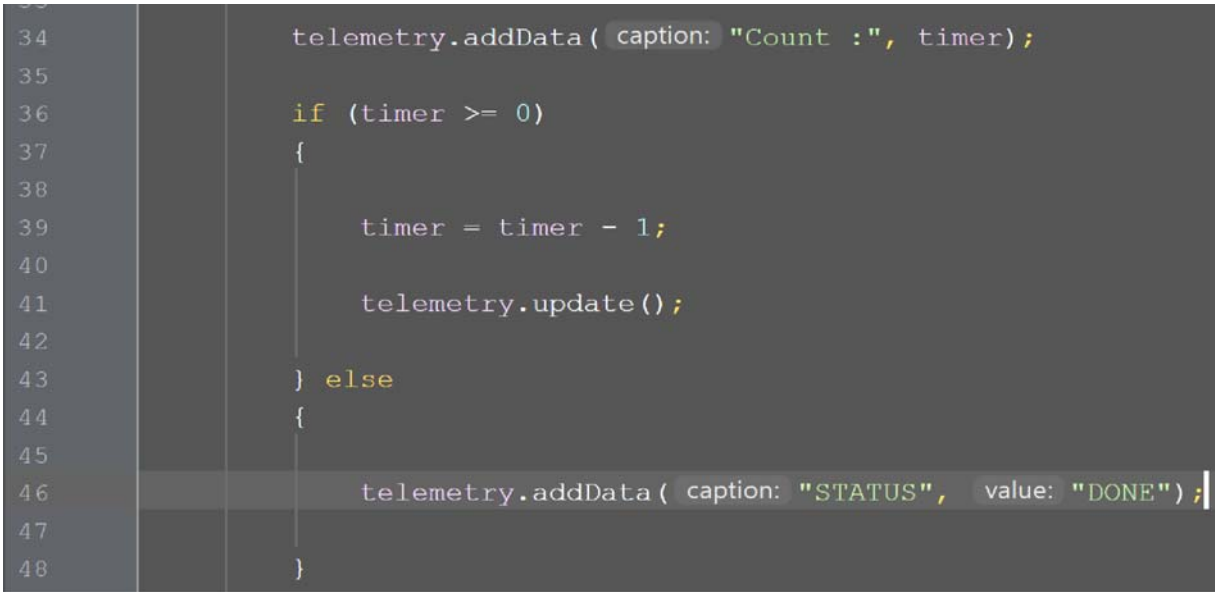
Goals for Programming	Progress on Goals
Finish Autonomous	We completed the Silver Lander side with all three mineral placements. We had a problem with inconsistencies, so we slowed the robot down.
Programming class	We built a mini-bot. We did this for Andy our backup programer so he would get used to OnBotJava. What we did is we programed the robot to move backwards. Andy learned how to move the robot back and forth.
Add driving speed options	Because the robot was fast and not accurate we added in a button to set the robot to half speed instead of full. It sets a variable to either 1 or 2 to divide our speed by.

```
124         if (gamepad1.x)
125         {
126
127             dSpd = 1;
128
129             spdMode = "Full";
130
131         } else if (gamepad1.y)
132         {
133
134             dSpd = 2;
135
136             spdMode = "Half";
137
138         }
```

```
57          leftFront.setPower((-(gamepad1.left_stick_y)+(gamepad1.right_stick_x))/dSpd);
58          rightFront.setPower(((gamepad1.left_stick_y)+(gamepad1.right_stick_x))/dSpd);
59          leftBack.setPower((-(gamepad1.left_stick_y)+(gamepad1.right_stick_x))/dSpd);
60          rightBack.setPower(((gamepad1.left_stick_y)+(gamepad1.right_stick_x))/dSpd);
```

January 27th, 2019
Mark Hazen

Goals for Programming	Progress on Goals
Making a timer for Autonomous default	This delay was aimed to count down for five seconds to make sure Washi always descended even if he doesn't detect anything. All it does is subtract from the timer number by one every time the loop occurs. It still needs implemented into the Autonomous



```
34          telemetry.addData( caption: "Count :", timer);
35
36          if (timer >= 0)
37          {
38
39              timer = timer - 1;
40
41              telemetry.update();
42
43          } else
44          {
45
46              telemetry.addData( caption: "STATUS",  value: "DONE");
47
48          }
```

January 28th, 2019
Mark Hazen

Goals for Programming	Progress on Goals
Adding in Acceleration for TeleOp	I tried squaring the power of the motors, but because the joystick values are floats we could not use the function ^3. I ended up multiplying the values by itself three times, but that would only create a higher speed curve as you move the joystick in a direction instead of having count up to the joystick value. Marty then helped me use a for loop to count up, but then it would pause while counting up instead of continuing to move. We ended up putting it on the backburner for later testing.
	

February 5th, 2019
Mark Hazen

Goals for Programming	Progress on Goals
Touch Sensor Stop	I took the sample program to test the sensor, but it wasn't recognizing a press. After trying another sensor and another wire we figured out that it was our expansion hub. Digital channels did not work for it on Hub 2. I added an and statement to our if loop that controls our linear actuator so it would lower if the left trigger is pressed and the button state is off.

```
80          if (gamepad1.right_trigger != 0)
81          {
82
83              linearActuator.setPower(-1);
84
85          } else if (gamepad1.left_trigger != 0 && actuatorTouch.getState() == true)
86          {
87
88              linearActuator.setPower(1);
89
90          } else
91          {
92
93              linearActuator.setPower(0);
94
95          }
```

February 16, 2019
Mark Hazen

Goals for Programming	Progress on Goals
Work on Gyroscopic Sensor	We wanted to use the gyro sensor to be consistent while getting off the lander. After looking through the sample program for the IMU we found the output but it was in a string. We started to research string to double programs but thought there must be a simpler way. After more looking through google we found a helpful tutorial that allowed us to import the values as a double. Then we used a while loop and compared the gyro value to our wanted value using < and >.

```
410         while (gyroAngle < degree)
411         {
412
413             angles = imu.getAngularOrientation(AxesReference.INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES);
414
415             gyroAngle = angles.firstAngle;
416
417             rightFront.setPower(-spd);
418             leftFront.setPower(-spd);
419             leftBack.setPower(-spd);
420             rightBack.setPower(-spd);
421
422         }
423
424         rightFront.setPower(0);
425         leftFront.setPower(0);
426         leftBack.setPower(0);
427         rightBack.setPower(0);
```

Start Autonomous After Change In Wheelbase	After descending the robot would turn using the gyro sensor for a 180 turn. We consistently knocked off the gold but it would not turn with the gyro sensor again. After a little bit of trouble shooting we realized encoders would work for turning because the floor won't be shaking (if it is than it's bad).

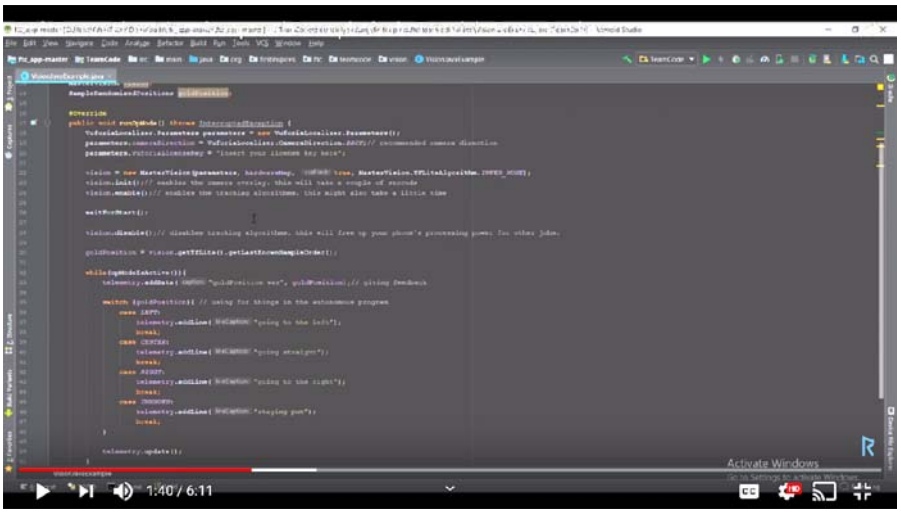
4

Monday February 18, 2019
Mark Hazen

Work on Autonomous	I got the middle gold on the silver side of the lander finished after we had to change the speeds. The robot was too slow at 0.25 power and we were running out of time so we changed the power to 0.3 and tweaked the program for the speed.

Tuesday February 19, 2019
Mark Hazen

Finish Silver/Crater Side for Blue Crater	While programming I found a 90 degree turn with encoders so we used that number and either multiplied or divided it by how much. I used a measuring tape to help with distance. I took the measurement and multiplied it by the increment of the measuring tape.
Finish Gold/Crater Side for Red Crater	The gold took less time than the silver. Because the autonomous is located in one quadrant the robot has to move less than the silver side which is half a field. I used the silver sampling code to line up the robot for sampling.
Add Default for Auto	We wanted a default timer in case the camera does not see three minerals. After trying out a timer a couple of places in the Autonomous I went to the FTC reddit page (r/FTC) and posted for help. I got a reply from a user and it was a video on a simple version of TensorFlow that uses cases, but it did not have camera opportunity from what I saw. The name isFTC TensorFlow Lite - Setup and Example Code Tutorial - Rover Ruckus. I think further testing will help switch to that program. I pulled our coach to talk over the logic of the program and we figured out a way for my original timer to work in our original TensorFlow program. Before the TensorFlow's first if loop I added my own like the test program. In the else I added a telemetry to tell us when the default activates. Pictures on next page.




```
while (opModeIsActive())
{

    if (timer > 0) {

        timer = timer - 1;

        telemetry.addData( caption: "Time", timer);
        telemetry.update();

        if (tfod != null) {
            // getUpdatedRecognitions() will return null if no new information is available since
            // the last time that call was made.
            List<Recognition> updatedRecognitions = tfod.getUpdatedRecognitions();
            if (updatedRecognitions != null) {
                telemetry.addData( caption: "# Object Detected", updatedRecognitions.size());
                if (updatedRecognitions.size() == 3) {
                    int goldMineralX = -1;
                    int silverMineral1X = -1;
                    int silverMineral2X = -1;
                    for (Recognition recognition : updatedRecognitions) {
```

```
272                     } else {
273
274                         telemetry.addData( caption: "Gold Mineral Position", value: "Default");
275
276                         telemetry.update();
277
278                         aC.descend( spd: 1, tic: 6000);
279
280                         aC.gTurnRightG( spd: 0.25, degree: 175);
281
282                         aC.eForward( spd: 0.3, tic: 2600);
283
284                         aC.eBackward( spd: 0.3, tic: 100);
285
286                         aC.eTurnLeft( spd: 0.3, tic: 1200);
287
288                         aC.dropPlayerMarker();
289
290                         Thread.sleep( time: 500);
291
292                         aC.eForward( spd: 0.3, tic: 750);
293
294                         aC.eTurnLeft( spd: 0.3, tic: 125);
295
296                         aC.eForward( spd: 0.3, tic: 3200);
297
298                         Thread.sleep( time: 30000);
299
300                     }
```

File - C:\Users\markh\Desktop\FTC\ftc_app\TeamCode\src\main\java\org\firstinspires\ftc\teamcode\AutoClass.java

```
 1 package org.firstinspires.ftc.teamcode;
 2
 3 import com.qualcomm.hardware.bosch.BNO055IMU;
 4 import com.qualcomm.robotcore.hardware.CRServo;
 5 import com.qualcomm.robotcore.hardware.DcMotor;
 6 import com.qualcomm.robotcore.hardware.DigitalChannel;
 7 import com.qualcomm.robotcore.hardware.Servo;
 8
 9 import org.firstinspires.ftc.robotcore.external.navigation
   .Acceleration;
10 import org.firstinspires.ftc.robotcore.external.navigation
   .AngleUnit;
11 import org.firstinspires.ftc.robotcore.external.navigation
   .AxesOrder;
12 import org.firstinspires.ftc.robotcore.external.navigation
   .AxesReference;
13 import org.firstinspires.ftc.robotcore.external.navigation
   .Orientation;
14
15 public class AutoClass
16 {
17
18     DcMotor leftFront;
19     DcMotor rightFront;
20     DcMotor leftBack;
21     DcMotor rightBack;
22
23     DcMotor linearActuator;
24
25     DcMotor intakeArm;
26     DcMotor pulley;
27     CRServo intake;
28
29     Servo playerMarker;
30
31     DigitalChannel actuatorTouch;
32     BNO055IMU imu;
33
34     Orientation angles;
35     //Acceleration gravity;
36     double gyroAngle = 0;
37
38     public AutoClass (DcMotor lF, DcMotor rF, DcMotor lB,
   DcMotor rB, DcMotor lA, DcMotor iA, DcMotor p, CRServo i,
   Servo pM, DigitalChannel aT, BNO055IMU im, Orientation a
```

```
38 /*, Acceleration g*/)
39     {
40
41         leftFront = lF;
42         rightFront = rF;
43         leftBack = lB;
44         rightBack = rB;
45
46         linearActuator = lA;
47
48         intakeArm = iA;
49         pulley = p;
50         intake = i;
51
52         playerMarker = pM;
53
54         actuatorTouch = aT;
55         imu = im;
56
57         angles = a;
58         //gravity = g;
59
60     }
61
62     void eForward (double spd, int tic)
63     {
64
65         leftFront.setMode(DcMotor.RunMode.
   RUN_USING_ENCODER);
66         rightFront.setMode(DcMotor.RunMode.
   RUN_USING_ENCODER);
67         leftBack.setMode(DcMotor.RunMode.RUN_USING_ENCODER
   );
68         rightBack.setMode(DcMotor.RunMode.
   RUN_USING_ENCODER);
69
70         leftFront.setMode(DcMotor.RunMode.
   STOP_AND_RESET_ENCODER);
71         rightFront.setMode(DcMotor.RunMode.
   STOP_AND_RESET_ENCODER);
72         leftBack.setMode(DcMotor.RunMode.
   STOP_AND_RESET_ENCODER);
73         rightBack.setMode(DcMotor.RunMode.
   STOP_AND_RESET_ENCODER);
74
```

```
75          leftFront.setTargetPosition(tic);
76          rightFront.setTargetPosition(-tic);
77          leftBack.setTargetPosition(tic);
78          rightBack.setTargetPosition(-tic);
79
80          leftFront.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
81          rightFront.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
82          leftBack.setMode(DcMotor.RunMode.RUN_TO_POSITION)
    ;
83          rightBack.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
84
85          leftFront.setPower(spd);
86          rightFront.setPower(spd);
87          leftBack.setPower(spd);
88          rightBack.setPower(spd);
89
90          while (leftFront.isBusy() && rightFront.isBusy()
    && leftBack.isBusy() && rightBack.isBusy())
91          {
92
93
94
95          }
96
97          stopEverything();
98
99          leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
100         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
101         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
102         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
103
104     }
105
106     void eBackward (double spd, int tic)
107     {
108
109         leftFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
```

```
110         rightFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
111         leftBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
112         rightBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
113
114         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
115         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
116         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
117         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
118
119         leftFront.setTargetPosition(-tic);
120         rightFront.setTargetPosition(tic);
121         leftBack.setTargetPosition(-tic);
122         rightBack.setTargetPosition(tic);
123
124         leftFront.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
125         rightFront.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
126         leftBack.setMode(DcMotor.RunMode.RUN_TO_POSITION)
    ;
127         rightBack.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
128
129         leftFront.setPower(spd);
130         rightFront.setPower(spd);
131         leftBack.setPower(spd);
132         rightBack.setPower(spd);
133
134         while (leftFront.isBusy() && rightFront.isBusy()
    && leftBack.isBusy() && rightBack.isBusy())
135         {
136
137
138
139         }
140
141         stopEverything();
142
```

```
143         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
144         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
145         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
146         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
147
148     }
149
150     void eSBackward (double spd, int tic)
151     {
152
153         leftFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
154         rightFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
155         leftBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
156         rightBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
157
158         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
159         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
160         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
161         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
162
163         leftFront.setTargetPosition(-tic);
164         rightFront.setTargetPosition(tic);
165         leftBack.setTargetPosition(-tic);
166         rightBack.setTargetPosition(tic);
167
168         leftFront.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
169         rightFront.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
170         leftBack.setMode(DcMotor.RunMode.RUN_TO_POSITION)
    ;
171         rightBack.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
```

```
172
173         leftFront.setPower(spd);
174         rightFront.setPower(spd);
175         leftBack.setPower(spd);
176         rightBack.setPower(spd);
177
178         while (leftFront.isBusy() && rightFront.isBusy()
    && leftBack.isBusy() && rightBack.isBusy())
179         {
180
181
182
183         }
184
185         stopEverything();
186
187         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
188         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
189         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
190         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
191
192     }
193
194     void eLeft (double spd, int tic)
195     {
196
197         leftFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
198         rightFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
199         leftBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
200         rightBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
201
202         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
203         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
204         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
```

```
205         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
206
207         leftFront.setTargetPosition(tic);
208         rightFront.setTargetPosition(tic);
209         leftBack.setTargetPosition(-tic);
210         rightBack.setTargetPosition(-tic);
211
212         leftFront.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
213         rightFront.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
214         leftBack.setMode(DcMotor.RunMode.RUN_TO_POSITION)
    ;
215         rightBack.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
216
217         leftFront.setPower(spd);
218         rightFront.setPower(spd);
219         leftBack.setPower(spd);
220         rightBack.setPower(spd);
221
222         while (leftFront.isBusy() && rightFront.isBusy()
    && leftBack.isBusy() && rightBack.isBusy())
223         {
224
225
226
227         }
228
229         stopEverything();
230
231         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
232         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
233         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
234         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
235
236     }
237
238     void eRight (double spd, int tic)
239     {
```

```
240
241         leftFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
242         rightFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
243         leftBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
244         rightBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
245
246         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
247         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
248         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
249         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
250
251         leftFront.setTargetPosition(tic);
252         rightFront.setTargetPosition(tic);
253         leftBack.setTargetPosition(-tic);
254         rightBack.setTargetPosition(-tic);
255
256         leftFront.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
257         rightFront.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
258         leftBack.setMode(DcMotor.RunMode.RUN_TO_POSITION)
    ;
259         rightBack.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
260
261         leftFront.setPower(spd);
262         rightFront.setPower(spd);
263         leftBack.setPower(spd);
264         rightBack.setPower(spd);
265
266         while (leftFront.isBusy() && rightFront.isBusy()
    && leftBack.isBusy() && rightBack.isBusy())
267         {
268
269
270
271         }
```

```
272
273         stopEverything();
274
275         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
276         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
277         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
278         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
279
280     }
281
282     void eTurnLeft (double spd, int tic)
283     {
284
285         leftFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
286         rightFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
287         leftBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
288         rightBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
289
290         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
291         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
292         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
293         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
294
295         leftFront.setTargetPosition(-tic);
296         rightFront.setTargetPosition(-tic);
297         leftBack.setTargetPosition(-tic);
298         rightBack.setTargetPosition(-tic);
299
300         leftFront.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
301         rightFront.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
302         leftBack.setMode(DcMotor.RunMode.RUN_TO_POSITION)
```

```
302 ;
303         rightBack.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
304
305         leftFront.setPower(spd);
306         rightFront.setPower(spd);
307         leftBack.setPower(spd);
308         rightBack.setPower(spd);
309
310         while (leftFront.isBusy() && rightFront.isBusy()
    && leftBack.isBusy() && rightBack.isBusy())
311         {
312
313
314
315         }
316
317         stopEverything();
318
319         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
320         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
321         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
322         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
323
324     }
325
326     void eTurnRight (double spd, int tic)
327     {
328
329         leftFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
330         rightFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
331         leftBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
332         rightBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
333
334         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
335         rightFront.setMode(DcMotor.RunMode.
```

```
335 STOP_AND_RESET_ENCODER);
336         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
337         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
338
339         leftFront.setTargetPosition(tic);
340         rightFront.setTargetPosition(tic);
341         leftBack.setTargetPosition(tic);
342         rightBack.setTargetPosition(tic);
343
344         leftFront.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
345         rightFront.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
346         leftBack.setMode(DcMotor.RunMode.RUN_TO_POSITION)
    ;
347         rightBack.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
348
349         leftFront.setPower(spd);
350         rightFront.setPower(spd);
351         leftBack.setPower(spd);
352         rightBack.setPower(spd);
353
354         while (leftFront.isBusy() && rightFront.isBusy()
    && leftBack.isBusy() && rightBack.isBusy())
355         {
356
357
358
359         }
360
361         stopEverything();
362
363         leftFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
364         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
365         leftBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
366         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
367
368     }
```

```
369
370     void eFRight (double spd, int tic)
371     {
372
373         rightFront.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
374         rightBack.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
375
376         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
377         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
378
379         rightFront.setTargetPosition(tic);
380         rightBack.setTargetPosition(tic);
381
382         rightFront.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
383         rightBack.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
384
385         rightFront.setPower(spd);
386         rightBack.setPower(spd);
387
388         while (rightFront.isBusy() && rightBack.isBusy())
389         {
390
391
392
393         }
394
395         stopEverything();
396
397         rightFront.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
398         rightBack.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
399
400     }
401
402     void gTurnRightG (double spd, double degree)
403     {
404
405         while (gyroAngle > -degree)
```

```
406         {
407
408             angles = imu.getAngularOrientation(
    AxesReference.INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES
    );
409
410             gyroAngle = angles.firstAngle;
411
412             rightFront.setPower(spd);
413             leftFront.setPower(spd);
414             leftBack.setPower(spd);
415             rightBack.setPower(spd);
416
417         }
418
419         rightFront.setPower(0);
420         leftFront.setPower(0);
421         leftBack.setPower(0);
422         rightBack.setPower(0);
423
424     }
425
426     void gTurnRightL (double spd, double degree)
427     {
428
429         while (gyroAngle < -degree)
430         {
431
432             angles = imu.getAngularOrientation(
    AxesReference.INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES
    );
433
434             gyroAngle = angles.firstAngle;
435
436             rightFront.setPower(spd);
437             leftFront.setPower(spd);
438             leftBack.setPower(spd);
439             rightBack.setPower(spd);
440
441         }
442
443         rightFront.setPower(0);
444         leftFront.setPower(0);
445         leftBack.setPower(0);
446         rightBack.setPower(0);
```

```
447
448     }
449
450     void gTurnLeftL (double spd, double degree)
451     {
452
453         //angles = imu.getAngularOrientation(
    AxesReference.INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES
    );
454
455         //gyroAngle = angles.firstAngle;
456
457         while (gyroAngle < degree)
458         {
459
460             angles = imu.getAngularOrientation(
    AxesReference.INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES
    );
461
462             gyroAngle = angles.firstAngle;
463
464             rightFront.setPower(-spd);
465             leftFront.setPower(-spd);
466             leftBack.setPower(-spd);
467             rightBack.setPower(-spd);
468
469         }
470
471         rightFront.setPower(0);
472         leftFront.setPower(0);
473         leftBack.setPower(0);
474         rightBack.setPower(0);
475
476     }
477
478     void gTurnLeftG (double spd, double degree)
479     {
480
481         angles = imu.getAngularOrientation(AxesReference.
    INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES);
482
483         gyroAngle = angles.firstAngle;
484
485         while (gyroAngle > degree)
486         {
```

```
487
488             angles = imu.getAngularOrientation(
    AxesReference.INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES
    );
489
490             gyroAngle = angles.firstAngle;
491
492             rightFront.setPower(-spd);
493             leftFront.setPower(-spd);
494             leftBack.setPower(-spd);
495             rightBack.setPower(-spd);
496
497         }
498
499         rightFront.setPower(0);
500         leftFront.setPower(0);
501         leftBack.setPower(0);
502         rightBack.setPower(0);
503
504     }
505
506     void lift (double spd, int tic)
507     {
508
509         linearActuator.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
510
511         linearActuator.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
512
513         linearActuator.setTargetPosition(tic);
514
515         linearActuator.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
516
517         linearActuator.setPower(spd);
518
519         while(linearActuator.isBusy())
520         {
521
522
523
524         }
525
526         stopEverything();
```

```
527
528         linearActuator.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
529
530     }
531
532     void descend (double spd, int tic)
533     {
534
535         linearActuator.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
536
537         linearActuator.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
538
539         linearActuator.setTargetPosition(-tic);
540
541         linearActuator.setMode(DcMotor.RunMode.
    RUN_TO_POSITION);
542
543         linearActuator.setPower(spd);
544
545         while(linearActuator.isBusy())
546         {
547
548
549
550         }
551
552         stopEverything();
553
554         linearActuator.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
555
556     }
557
558     void end ()
559     {
560
561         actuatorTouch.setMode(DigitalChannel.Mode.INPUT);
562
563         while (actuatorTouch.getState() == true)
564         {
565
566             linearActuator.setPower(1);
```

```
567
568             }
569
570         }
571
572         void armUp (double spd, int tic)
573         {
574
575             intakeArm.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
576
577             intakeArm.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
578
579             intakeArm.setTargetPosition(-tic);
580
581             intakeArm.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
582
583             intakeArm.setPower(spd);
584
585             while (intakeArm.isBusy())
586             {
587
588
589
590             }
591
592             stopEverything();
593
594             intakeArm.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
595
596         }
597
598         void armDown (double spd, int tic)
599         {
600
601             intakeArm.setMode(DcMotor.RunMode.
    RUN_USING_ENCODER);
602
603             intakeArm.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
604
605             intakeArm.setTargetPosition(tic);
```

```
606
607         intakeArm.setMode(DcMotor.RunMode.RUN_TO_POSITION
    );
608
609         intakeArm.setPower(spd);
610
611         while (intakeArm.isBusy())
612         {
613
614
615
616         }
617
618         stopEverything();
619
620         intakeArm.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
621
622     }
623
624     void extend (double spd, int tic)
625     {
626
627         pulley.setMode(DcMotor.RunMode.RUN_USING_ENCODER)
    ;
628
629         pulley.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
630
631         pulley.setTargetPosition(-tic);
632
633         pulley.setMode(DcMotor.RunMode.RUN_TO_POSITION);
634
635         pulley.setPower(spd);
636
637         while (pulley.isBusy())
638         {
639
640
641
642         }
643
644         stopEverything();
645
646         pulley.setMode(DcMotor.RunMode.
```

```
646 STOP_AND_RESET_ENCODER);
647
648     }
649
650     void retract (double spd, int tic)
651     {
652
653         pulley.setMode(DcMotor.RunMode.RUN_USING_ENCODER)
    ;
654
655         pulley.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
656
657         pulley.setTargetPosition(tic);
658
659         pulley.setMode(DcMotor.RunMode.RUN_TO_POSITION);
660
661         pulley.setPower(spd);
662
663         while (pulley.isBusy())
664         {
665
666
667
668         }
669
670         stopEverything();
671
672         pulley.setMode(DcMotor.RunMode.
    STOP_AND_RESET_ENCODER);
673
674     }
675
676     void intake (double spd, long sec)  throws
    InterruptedException
677     {
678
679         intake.setPower(spd);
680
681         Thread.sleep(sec);
682
683         intake.setPower(0.5);
684
685     }
686
```

```
687     void dropPlayerMarker ()
688     {
689
690         playerMarker.setPosition(0);
691
692     }
693
694     void stopEverything ()
695     {
696
697
698         leftFront.setPower(0);
699         rightFront.setPower(0);
700         leftBack.setPower(0);
701         rightBack.setPower(0);
702
703         linearActuator.setPower(0);
704
705         intakeArm.setPower(0);
706         pulley.setPower(0);
707         intake.setPower(0);
708
709         playerMarker.setPosition(0.9);
710     }
711
712 }
713
```

File - C:\Users\markh\Desktop\FTC\ftc_app\TeamCode\src\main\java\org\firstinspires\ftc\teamcode\GoldRight.java

```
 1 package org.firstinspires.ftc.teamcode;
 2
 3 import com.qualcomm.hardware.bosch.BNO055IMU;
 4 import com.qualcomm.hardware.bosch.
   JustLoggingAccelerationIntegrator;
 5 import com.qualcomm.robotcore.eventloop.opmode.Autonomous;
 6 import com.qualcomm.robotcore.eventloop.opmode.Disabled;
 7 import com.qualcomm.robotcore.eventloop.opmode.
   LinearOpMode;
 8 import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
 9 import com.qualcomm.robotcore.hardware.CRServo;
10 import com.qualcomm.robotcore.hardware.DcMotor;
11 import com.qualcomm.robotcore.hardware.DigitalChannel;
12 import com.qualcomm.robotcore.hardware.Servo;
13
14 import java.util.List;
15 import org.firstinspires.ftc.robotcore.external.
   ClassFactory;
16 import org.firstinspires.ftc.robotcore.external.hardware.
   camera.WebcamName;
17 import org.firstinspires.ftc.robotcore.external.navigation
   .Orientation;
18 import org.firstinspires.ftc.robotcore.external.navigation
   .VuforiaLocalizer;
19 import org.firstinspires.ftc.robotcore.external.tfod.
   TFObjectDetector;
20 import org.firstinspires.ftc.robotcore.external.tfod.
   Recognition;
21
22 /**
23  * This 2018-2019 OpMode illustrates the basics of using
   the TensorFlow Object Detection API to
24  * determine the position of the gold and silver minerals.
25  *
26  * Use Android Studio to Copy this Class, and Paste it
   into your team's code folder with a new name.
27  * Remove or comment out the @Disabled line to add this
   opmode to the Driver Station OpMode list.
28  *
29  * IMPORTANT: In order to use this OpMode, you need to
   obtain your own Vuforia license key as
30  * is explained below.
31  */
32 @Autonomous(name = "GoldRight", group = "Auto")
33 //@Disabled
```

```
34 public class GoldRight extends LinearOpMode {
35     private static final String TFOD_MODEL_ASSET = "
   RoverRuckus.tflite";
36     private static final String LABEL_GOLD_MINERAL = "Gold
    Mineral";
37     private static final String LABEL_SILVER_MINERAL = "
   Silver Mineral";
38
39     /*
40      * IMPORTANT: You need to obtain your own license key
   to use Vuforia. The string below with which
41      * 'parameters.vuforiaLicenseKey' is initialized is
   for illustration only, and will not function.
42      * A Vuforia 'Development' license key, can be
   obtained free of charge from the Vuforia developer
43      * web site at https://developer.vuforia.com/license-
   manager.
44      *
45      * Vuforia license keys are always 380 characters long
   , and look as if they contain mostly
46      * random data. As an example, here is a example of a
   fragment of a valid key:
47      *      ...
   yIgIzTqZ4mWjk9wd3cZO9T1axEqzuhxoGlfOOI2dRzKS4T0hQ8kT ...
48      * Once you've obtained a license key, copy the string
    from the Vuforia web site
49      * and paste it in to your code on the next line,
   between the double quotes.
50      */
51     private static final String VUFORIA_KEY = "AQUWr4X
   /////AAABme38EPssRkvls9+q/
   BGPYgxKXBXELWHMdkTcCqUqHeyDpyXGWFLCTABgDXEMGe1EmsnDQxmJ7WQ
   069J3YSv+
   kOcfq3g2EnwZr2O3DujsIU1nT0aXgLlAtQU2r7wWAgHvR9ADO5pe/
   q7MzCyhjSTQLCgizGFLgmqfre0A9rjYcXYbYw11R3P7VRHnL3QHn3QH2oF
   VQfMb+dIzmZkfv0cd5qWvdhjovYF8hpZ/
   HT7veIa8ZQ9CIQ0541pxplXVud80z1xWpjFGJPaoQGO+xKWZ8E+
   Zlu7z5umiaV1+
   ChGeJ9pPyIJn0LsnoIHumZoYb4di4tFygMPVmH8ChsTlGJjaPBSCRBFjxz
   BqsXmBZY7eCa6S";
52
53     /**
54      * {@link #vuforia} is the variable we will use to
   store our instance of the Vuforia
55      * localization engine.
```

File - C:\Users\markh\Desktop\FTC\ftc_app\TeamCode\src\main\java\org\firstinspires\ftc\teamcode\GoldRight.java

```
56      */
57     private VuforiaLocalizer vuforia;
58
59     /**
60      * {@link #tfod} is the variable we will use to store
    our instance of the Tensor Flow Object
61      * Detection engine.
62      */
63     private TFObjectDetector tfod;
64
65     DcMotor leftFront;
66     DcMotor rightFront;
67     DcMotor leftBack;
68     DcMotor rightBack;
69
70     DcMotor linearActuator;
71
72     DcMotor intakeArm;
73     DcMotor pulley;
74     CRServo intake;
75
76     Servo playerMarker;
77
78     DigitalChannel actuatorTouch;
79     BNO055IMU imu;
80
81     Orientation angles;
82     Acceleration gravity;
83
84     AutoClass aC;
85
86     int timer = 100000;
87
88     @Override
89     public void runOpMode()throws InterruptedException
90     {
91
92         BNO055IMU.Parameters parameters = new BNO055IMU.
   Parameters();
93         parameters.angleUnit           = BNO055IMU.
   AngleUnit.DEGREES;
94         parameters.accelUnit           = BNO055IMU.
   AccelUnit.METERS_PERSEC_PERSEC;
95         parameters.calibrationDataFile = "
   BNO055IMUCalibration.json";
```

```
 96          parameters.loggingEnabled      = true;
 97          parameters.loggingTag          = "IMU";
 98          parameters.accelerationIntegrationAlgorithm = new
     JustLoggingAccelerationIntegrator();
 99
100          leftFront = hardwareMap.dcMotor.get("leftFront");
101          rightFront = hardwareMap.dcMotor.get("rightFront"
    );
102          leftBack = hardwareMap.dcMotor.get("leftBack");
103          rightBack = hardwareMap.dcMotor.get("rightBack");
104
105          linearActuator = hardwareMap.dcMotor.get("
    linearActuator");
106
107          intakeArm = hardwareMap.dcMotor.get("intakeArm");
108          pulley = hardwareMap.dcMotor.get("pulley");
109          intake = hardwareMap.crservo.get("intake");
110
111          playerMarker = hardwareMap.servo.get("
    playerMarker");
112
113          actuatorTouch = hardwareMap.get(DigitalChannel.
    class, "actuatorTouch");
114          imu = hardwareMap.get(BNO055IMU.class, "imu");
115          imu.initialize(parameters);
116
117          aC = new AutoClass(leftFront, rightFront,
    leftBack, rightBack, linearActuator, intakeArm, pulley,
    intake, playerMarker, actuatorTouch, imu, angles);
118
119          aC.stopEverything();
120
121          // The TFObjectDetector uses the camera frames
    from the VuforiaLocalizer, so we create that
122          // first.
123          initVuforia();
124
125          if (ClassFactory.getInstance().
    canCreateTFObjectDetector()) {
126              initTfod();
127          } else {
128              telemetry.addData("Sorry!", "This device is
    not compatible with TFOD");
129          }
130
```

```
131         /** Wait for the game to begin */
132         telemetry.addData(">", "Press Play to start
    tracking");
133         telemetry.update();
134
135         waitForStart();
136
137         if (opModeIsActive())
138         {
139             /** Activate Tensor Flow Object Detection. */
140             if (tfod != null) {
141                 tfod.activate();
142             }
143
144             while (opModeIsActive())
145             {
146
147                 if (timer > 0) {
148
149                     timer = timer - 1;
150
151                     telemetry.addData("Time", timer);
152                     telemetry.update();
153
154                     if (tfod != null) {
155                         // getUpdatedRecognitions() will
    return null if no new information is available since
156                         // the last time that call was
    made.
157                         List<Recognition>
    updatedRecognitions = tfod.getUpdatedRecognitions();
158                         if (updatedRecognitions != null)
    {
159                             telemetry.addData("# Object
    Detected", updatedRecognitions.size());
160                             if (updatedRecognitions.size(
    ) == 3) {
161                                 int goldMineralX = -1;
162                                 int silverMineral1X = -1;
163                                 int silverMineral2X = -1;
164                                 for (Recognition
    recognition : updatedRecognitions) {
165                                     if (recognition.
    getLabel().equals(LABEL_GOLD_MINERAL)) {
166                                         goldMineralX = (
```

```
166 int) recognition.getLeft();
167                                     } else if (
    silverMineral1X == -1) {
168                                         silverMineral1X =
     (int) recognition.getLeft();
169                                     } else {
170                                         silverMineral2X =
     (int) recognition.getLeft();
171                                     }
172                                 }
173                                 if (goldMineralX != -1 &&
     silverMineral1X != -1 && silverMineral2X != -1) {
174                                     if (goldMineralX <
    silverMineral1X && goldMineralX < silverMineral2X) {
175                                         telemetry.addData
    ("Gold Mineral Position", "Left");
176
177                                         telemetry.update(
    );
178
179                                         aC.descend(1,
    6000);
180
181                                         aC.gTurnRightG(0.
    25, 145);
182
183                                         aC.eForward(0.3,
    1500);
184
185                                         aC.eTurnRight(0.3
    , 700);
186
187                                         aC.eForward(0.3,
    1350);
188
189                                         aC.eBackward(0.3,
     100);
190
191                                         aC.eTurnLeft(0.3,
     1700);
192
193                                         aC.
    dropPlayerMarker();
194
195                                         Thread.sleep(500)
```

```
195 ;
196
197                                             aC.eTurnRight(0.3
    , 100);
198
199                                             aC.eForward(0.3,
    3200);
200
201                                             Thread.sleep(
    30000);
202
203                                         } else if (
    goldMineralX > silverMineral1X && goldMineralX >
    silverMineral2X) {
204                                             telemetry.addData
    ("Gold Mineral Position", "Right");
205
206                                             telemetry.update(
    );
207
208                                             aC.descend(1,
    6000);
209
210                                             aC.gTurnRightG(0.
    25, 170);
211
212                                             aC.eTurnRight(0.3
    , 350);
213
214                                             aC.eForward(0.3,
    1500);
215
216                                             aC.eTurnLeft(0.3,
     600);
217
218                                             aC.eForward(0.3,
    1250);
219
220                                             aC.eBackward(0.3,
     100);
221
222                                             aC.eTurnLeft(0.3,
     1000);
223
224                                             aC.
```

```
224 dropPlayerMarker();
225
226                                         Thread.sleep(500)
    ;
227
228                                         aC.eTurnRight(0.3
    , 250);
229
230                                         aC.eForward(0.3,
    1000);
231
232                                         aC.eTurnLeft(0.3,
     250);
233
234                                         aC.eForward(0.3,
    3200);
235
236                                         Thread.sleep(
    30000);
237
238                                     } else {
239                                         telemetry.addData
    ("Gold Mineral Position", "Center");
240
241                                         telemetry.update(
    );
242
243                                         aC.descend(1,
    6000);
244
245                                         aC.gTurnRightG(0.
    25, 175);
246
247                                         aC.eForward(0.3,
    2600);
248
249                                         aC.eBackward(0.3,
     100);
250
251                                         aC.eTurnLeft(0.3,
     1200);
252
253                                         aC.
    dropPlayerMarker();
254
```

```
255                                                 Thread.sleep(500)
    ;
256
257                                                 aC.eForward(0.3,
    750);
258
259                                                 aC.eTurnLeft(0.3,
     125);
260
261                                                 aC.eForward(0.3,
    3200);
262
263                                                 Thread.sleep(
    30000);
264
265                                             }
266                                         }
267                                     }
268                                     telemetry.update();
269                                 }
270                             }
271
272                         } else {
273
274                             telemetry.addData("Gold Mineral
    Position", "Default");
275
276                             telemetry.update();
277
278                             aC.descend(1, 6000);
279
280                             aC.gTurnRightG(0.25, 175);
281
282                             aC.eForward(0.3, 2600);
283
284                             aC.eBackward(0.3, 100);
285
286                             aC.eTurnLeft(0.3, 1200);
287
288                             aC.dropPlayerMarker();
289
290                             Thread.sleep(500);
291
292                             aC.eForward(0.3, 750);
293
```

```
294                     aC.eTurnLeft(0.3, 125);
295
296                     aC.eForward(0.3, 3200);
297
298                     Thread.sleep(30000);
299
300                 }
301
302             }
303         }
304
305         if (tfod != null) {
306             tfod.shutdown();
307         }
308     }
309
310     /**
311      * Initialize the Vuforia localization engine.
312      */
313     private void initVuforia() {
314         /*
315          * Configure Vuforia by creating a Parameter object, and passing it to the Vuforia engine.
316          */
317         VuforiaLocalizer.Parameters parameters = new VuforiaLocalizer.Parameters();
318
319         parameters.vuforiaLicenseKey = VUFORIA_KEY;
320         parameters.cameraName = hardwareMap.get(WebcamName.class, "Webcam 1");
321
322         //  Instantiate the Vuforia engine
323         vuforia = ClassFactory.getInstance().createVuforia(parameters);
324
325         // Loading trackables is not necessary for the Tensor Flow Object Detection engine.
326     }
327
328     /**
329      * Initialize the Tensor Flow Object Detection engine.
330      */
331     private void initTfod() {
332         int tfodMonitorViewId = hardwareMap.appContext.
```

```
332 getResources().getIdentifier(
333                 "tfodMonitorViewId", "id", hardwareMap.
    appContext.getPackageName());
334         TFObjectDetector.Parameters tfodParameters = new
    TFObjectDetector.Parameters(tfodMonitorViewId);
335         tfod = ClassFactory.getInstance().
    createTFObjectDetector(tfodParameters, vuforia);
336         tfod.loadModelFromAsset(TFOD_MODEL_ASSET,
    LABEL_GOLD_MINERAL, LABEL_SILVER_MINERAL);
337     }
338 }
```

File - C:\Users\markh\Desktop\FTC\ftc_app\TeamCode\src\main\java\org\firstinspires\ftc\teamcode\TeleOpClass.java

```java
package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.hardware.CRServo;
import com.qualcomm.robotcore.hardware.DcMotor;
import com.qualcomm.robotcore.hardware.DcMotorSimple;
import com.qualcomm.robotcore.hardware.DigitalChannel;
import com.qualcomm.robotcore.hardware.Gamepad;
import com.qualcomm.robotcore.hardware.Servo;

public class TeleOpClass
{

    DcMotor leftFront;
    DcMotor rightFront;
    DcMotor leftBack;
    DcMotor rightBack;

    DcMotor linearActuator;

    DcMotor intakeArm;
    DcMotor pulley;
    CRServo intake;

    Servo playerMarker;

    DigitalChannel actuatorTouch;

    AutoClass aC;

    public TeleOpClass (DcMotor lF, DcMotor rF, DcMotor lB
, DcMotor rB, DcMotor lA, DcMotor iA, DcMotor p, CRServo i
, Servo pM, DigitalChannel dT, AutoClass a)
    {

        leftFront = lF;
        rightFront = rF;
        leftBack = lB;
        rightBack = rB;

        linearActuator = lA;

        intakeArm = iA;
        pulley = p;
        intake = i;

```

```
44          playerMarker = pM;
45
46          actuatorTouch = dT;
47
48          aC = a;
49
50      }
51
52      void TeleOpPackage (Gamepad gamepad1, Gamepad gamepad2
    , double spd, double dSpd, double eSpd)
53      {
54
55          actuatorTouch.setMode(DigitalChannel.Mode.INPUT);
56
57          leftFront.setPower((-(gamepad1.left_stick_y)+(
    gamepad1.right_stick_x))/dSpd);
58          rightFront.setPower(((gamepad1.left_stick_y)+(
    gamepad1.right_stick_x))/dSpd);
59          leftBack.setPower((-(gamepad1.left_stick_y)+(
    gamepad1.right_stick_x))/dSpd);
60          rightBack.setPower(((gamepad1.left_stick_y)+(
    gamepad1.right_stick_x))/dSpd);
61
62          if(gamepad1.dpad_right)
63          {
64
65              leftFront.setTargetPosition(1);
66              rightFront.setTargetPosition(1);
67              leftBack.setTargetPosition(-1);
68              rightBack.setTargetPosition(-1);
69
70          } else if (gamepad1.dpad_left)
71          {
72
73              leftFront.setTargetPosition(-1);
74              rightFront.setTargetPosition(-1);
75              leftBack.setTargetPosition(1);
76              rightBack.setTargetPosition(1);
77
78          }
79
80          if (gamepad1.right_trigger != 0)
81          {
82
83              linearActuator.setPower(-1);
```

```
84
85          } else if (gamepad1.left_trigger != 0 &&
    actuatorTouch.getState() == true)
86          {
87
88              linearActuator.setPower(1);
89
90          } else
91          {
92
93              linearActuator.setPower(0);
94
95          }
96
97          if (gamepad2.right_trigger != 0)
98          {
99
100             intakeArm.setPower(-spd);
101
102         } else if (gamepad2.left_trigger != 0)
103         {
104
105             intakeArm.setPower(eSpd);
106
107         } else
108         {
109
110             intakeArm.setPower(-0.2);
111
112         }
113
114         if (gamepad2.right_bumper)
115         {
116
117             pulley.setPower(-1);
118
119         } else if (gamepad2.left_bumper)
120         {
121
122             pulley.setPower(1);
123
124         } else
125         {
126
127
```

```
128             pulley.setPower(0);
129
130         }
131
132         if (gamepad2.a)
133         {
134
135             intake.setPower(0.5);
136
137         } else if (gamepad2.x)
138         {
139
140             intake.setPower(-0.5);
141
142         } else
143         {
144
145             intake.setPower(0);
146
147         }
148
149     }
150
151     void armSetup ()
152     {
153
154
155
156     }
157
158     void stopEverything ()
159     {
160
161         leftFront.setPower(0);
162         rightFront.setPower(0);
163         leftBack.setPower(0);
164         rightBack.setPower(0);
165
166         linearActuator.setPower(0);
167
168         intakeArm.setPower(0);
169         pulley.setPower(0);
170         intake.setPower(0);
171
172         playerMarker.setPosition(0.65);
```

```
173
174         }
175
176 }
177
```

File - C:\Users\markh\Desktop\FTC\ftc_app\TeamCode\src\main\java\org\firstinspires\ftc\teamcode\Drive.java

```java
package org.firstinspires.ftc.teamcode;

import android.graphics.drawable.GradientDrawable;

import com.qualcomm.hardware.bosch.BNO055IMU;
import com.qualcomm.robotcore.eventloop.opmode.OpMode;
import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
import com.qualcomm.robotcore.hardware.CRServo;
import com.qualcomm.robotcore.hardware.DcMotor;
import com.qualcomm.robotcore.hardware.DcMotorSimple;
import com.qualcomm.robotcore.hardware.DigitalChannel;
import com.qualcomm.robotcore.hardware.Servo;
import com.qualcomm.robotcore.util.ElapsedTime;

import org.firstinspires.ftc.robotcore.external.navigation
.Orientation;

@TeleOp(name = "Rise Up", group = "Right Hand Man")

public class Drive extends OpMode
{

    DcMotor leftFront;
    DcMotor rightFront;
    DcMotor leftBack;
    DcMotor rightBack;

    DcMotor linearActuator;

    DcMotor intakeArm;
    DcMotor pulley;
    CRServo intake;

    Servo playerMarker;

    DigitalChannel actuatorTouch;

    BNO055IMU imu;

    Orientation angles;

    double spd = 0.5;
    double dSpd = 2;
    double eSpd = 0.1;
    String spdMode = "Half";
```

File - C:\Users\markh\Desktop\FTC\ftc_app\TeamCode\src\main\java\org\firstinspires\ftc\teamcode\Drive.java

```
45      String dir = "forward";
46      String actuatorT = "unpressed";
47      private ElapsedTime runtime = new ElapsedTime();
48
49      TeleOpClass tOC;
50      AutoClass aC;
51
52      @Override
53      public void init()
54      {
55
56          leftFront = hardwareMap.dcMotor.get("leftFront");
57          rightFront = hardwareMap.dcMotor.get("rightFront")
    ;
58          leftBack = hardwareMap.dcMotor.get("leftBack");
59          rightBack = hardwareMap.dcMotor.get("rightBack");
60
61          linearActuator = hardwareMap.dcMotor.get("
    linearActuator");
62
63          intakeArm = hardwareMap.dcMotor.get("intakeArm");
64          pulley = hardwareMap.dcMotor.get("pulley");
65          intake = hardwareMap.crservo.get("intake");
66
67          playerMarker = hardwareMap.servo.get("playerMarker
    ");
68
69          actuatorTouch = hardwareMap.get(DigitalChannel.
    class, "actuatorTouch");
70
71          imu = hardwareMap.get(BNO055IMU.class, "imu");
72
73          aC = new AutoClass (leftFront, rightFront,
    leftBack, rightBack, linearActuator, intakeArm, pulley,
    intake, playerMarker, actuatorTouch, imu, angles);
74          tOC = new TeleOpClass(leftFront, rightFront,
    leftBack, rightBack, linearActuator, intakeArm, pulley,
    intake, playerMarker, actuatorTouch, aC);
75
76          tOC.stopEverything();
77
78          leftFront.setZeroPowerBehavior(DcMotor.
    ZeroPowerBehavior.BRAKE);
79          rightFront.setZeroPowerBehavior(DcMotor.
    ZeroPowerBehavior.BRAKE);
```

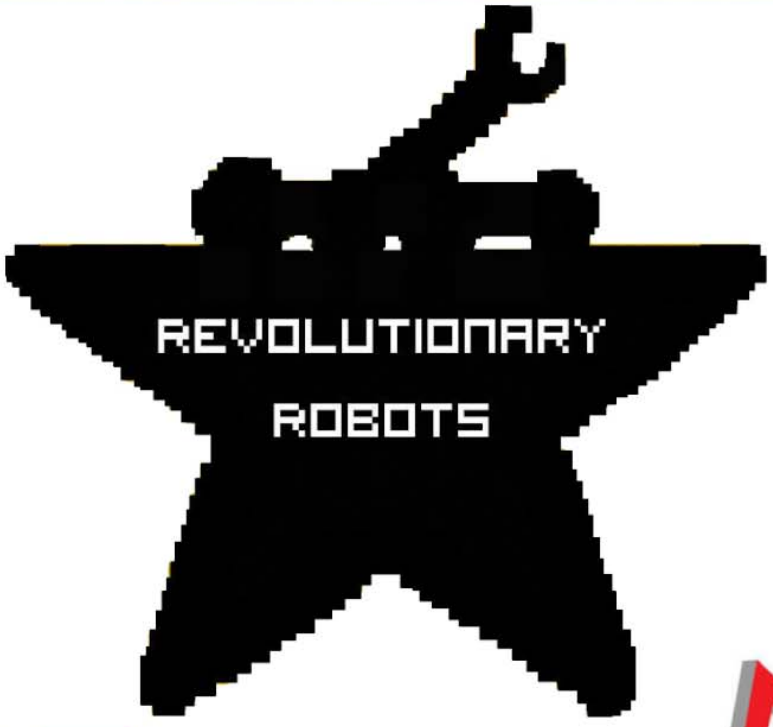
```
 80          leftBack.setZeroPowerBehavior(DcMotor.
    ZeroPowerBehavior.BRAKE);
 81          rightBack.setZeroPowerBehavior(DcMotor.
    ZeroPowerBehavior.BRAKE);
 82
 83          intakeArm.setZeroPowerBehavior(DcMotor.
    ZeroPowerBehavior.BRAKE);
 84
 85      }
 86
 87      @Override
 88      public void start()
 89      {
 90
 91          tOC.stopEverything();
 92
 93          //tOC.armSetup();
 94
 95      }
 96
 97      @Override
 98      public void loop()
 99      {
100
101          actuatorTouch.setMode(DigitalChannel.Mode.OUTPUT)
    ;
102
103          tOC.TeleOpPackage(gamepad1, gamepad2, spd, dSpd,
    eSpd);
104
105          if (gamepad1.a)
106          {
107
108              leftFront.setDirection(DcMotorSimple.
    Direction.FORWARD);
109              rightFront.setDirection(DcMotorSimple.
    Direction.FORWARD);
110              leftBack.setDirection(DcMotorSimple.Direction
    .FORWARD);
111              rightBack.setDirection(DcMotorSimple.
    Direction.FORWARD);
112
113              dir = "forward";
114
115          } else if (gamepad1.b)
```

```
116         {
117
118             leftFront.setDirection(DcMotorSimple.
    Direction.REVERSE);
119             rightFront.setDirection(DcMotorSimple.
    Direction.REVERSE);
120             leftBack.setDirection(DcMotorSimple.Direction
    .REVERSE);
121             rightBack.setDirection(DcMotorSimple.
    Direction.REVERSE);
122
123             dir = "reverse";
124
125         }
126
127         if (gamepad1.x)
128         {
129
130             dSpd = 1;
131
132             spdMode = "Full";
133
134         } else if (gamepad1.y)
135         {
136
137             dSpd = 2;
138
139             spdMode = "Half";
140
141         }
142
143         if (gamepad2.y)
144         {
145
146             spd = 0.75;
147
148         } else if (gamepad2.b)
149         {
150
151             spd = 0.4;
152
153         }
154
155         if (gamepad2.dpad_up)
156         {
```

```
157
158                 eSpd = 0.4;
159
160             } else if (gamepad2.dpad_down)
161             {
162
163                 eSpd = 0.1;
164
165             }
166
167             if (actuatorTouch.getState()== true)
168             {
169
170                 actuatorT = "unpressed";
171
172             } else if (actuatorTouch.getState() == false)
173             {
174
175                 actuatorT = "pressed";
176
177             }
178
179
180             telemetry.addData("Direction", dir);
181             telemetry.addData("Drive Speed", spdMode);
182             telemetry.addData("Arm Speed", spd);
183             telemetry.addData("Extend Speed", eSpd);
184             telemetry.addData("Linear Actuator", actuatorT);
185
186             telemetry.update();
187
188         }
189
190         @Override
191         public void stop()
192         {
193
194             tOC.stopEverything();
195
196         }
197
198 }
199
```

Revolutionary Robots
12535

Team Education
2018-2019



Moore Community STEM Club

FIRST®

STRATEGIZING AT STATE

YOUR DIRECT STRATEGY GUIDE TO THE FTC WORLD COMPETITION

BE HAPPY

OR AT LEAST ACT HAPPY

- WHEN YOU TALK TO SOMEONE NEW YOU NEED TO MAKE A GOOD FIRST IMPRESSION
- DO YOU ENJOY TALKING TO PEOPLE WHO HAVE A BAD ATTITUDE OR ALWAYS SEEM UNHAPPY?

BE PATIENT

- LETS BE HONEST TALKING TO STRANGERS IS NEVER THE MOST FUN THING TO DO
- WHEN THAT PERSON DOES SOMETHING THAT FRUSTRATES YOU IT MAKES THINGS WORSE
- IF YOU STAY PATIENT WITH THEM IT COULD BE CONTAGIOUS AND MAKE THEM MORE PATIENT WITH YOU

COMPLIMENT THE OTHER TEAMS

- COMPLIMENTS CAN GET OTHER TEAMS INTERESTED IN YOU
- IT DOESN'T EVEN NEED TO BE ABOUT THEIR ROBOT
- EVEN COMPLIMENT THEM ON THEIR TEAM NAME

ASK QUESTIONS ABOUT THEIR TEAM

- SOCIALIZING IS A BIG PART OF STRATEGIZING BUT YOU ALSO NEED TO KNOW WHAT THEIR TEAM CAN DO
- IF IT COMES DOWN TO SELECTING AN ALLIANCE YOU NEED TO KNOW WHO IS THE BEST TEAM TO ALIGN WITH

ANSWER ANY QUIESTIONS

- IF YOU ARE NOT THE TEAM CHOOSING YOU NEED TO BE CHOSEN
- IF THE OTHER TEAMS DON'T KNOW WHAT YOU CAN DO THEN THEY WILL NOT CHOOSE YOU

BIG NONOS

1. SARCASM

SARCASM NO MATTER WHAT THE REAL REASON CAN BE TAKEN THE WRONG WAY

2. DON'T IGNORE
3. DON'T CRITICIZE
4. DON'T JUDGE SOMEONE BEFORE YOU MEET THEM

NOTES

A COUPLE IMPORTANT THINGS INCLUDE DOING THIS AT THE RIGHT TIME. WHEN ANOTHER TEAM IS CURRENTLY BUSY DOING SOMETHING COMING UP TO THEM COULD SEEM VERY PUSHY AND RUDE. SO DON'T INTERUPT. MAKE SURE STRATEGY HAPPENS EARLY IN THE COMPETITION (MEANING FRIDAY NOT AFTER COMPETITION ON SATURDAY). RELATIONSHIPS NEED TO BE BUILT BEFORE WE MIGHT NEED THEM TO PICK US INTO THEIR ALLIANCE.

ROBOT CONFIG

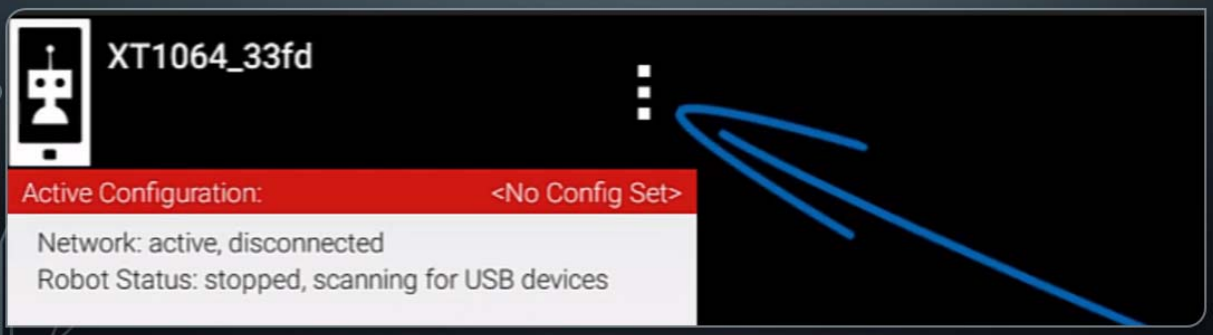
BY: MORP

WHAT ARE THEY?

- Used to set up devices on the robot
- Defines devices ie. Continuous Servos or Servos
- Names the devices ie. Left Wheel
- Uses “Humpback” format ie. leftWheel

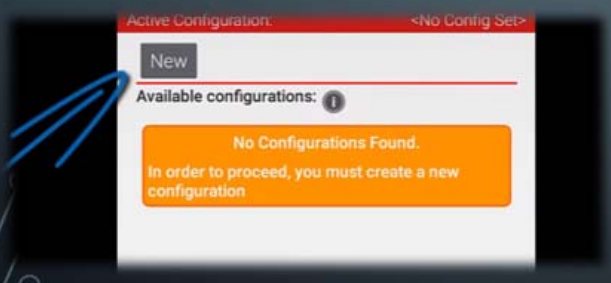
GETTING TO THE CONFIGURATIONS

- Tap on the three dots in the top right
- Tap on Configure Robot

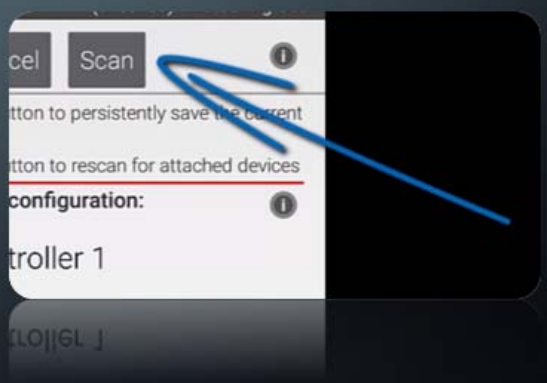


FINDING THE MODULE

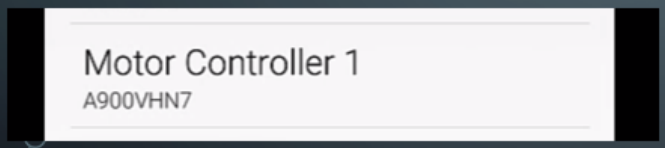
- Click New



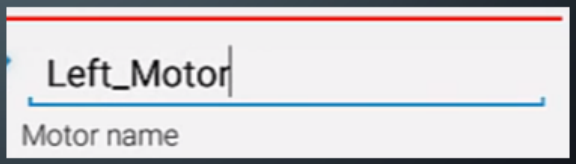
- Tap Scan



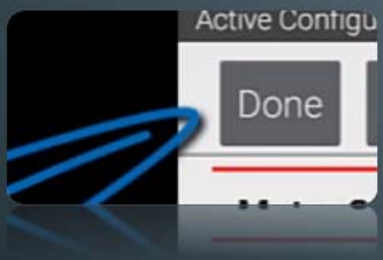
TAP ON YOUR MODULE



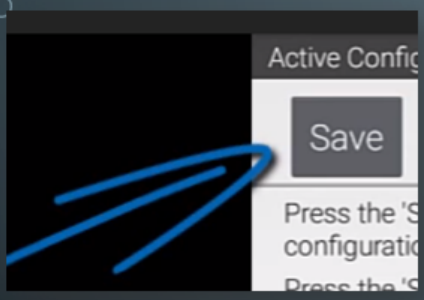
SELECT DEVICE AND NAME PORT

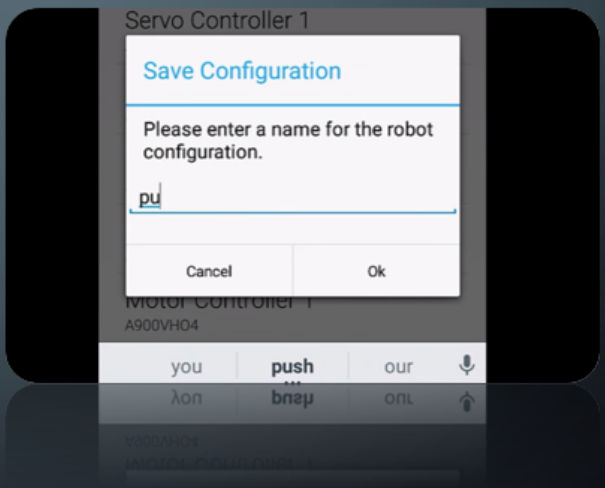


TAP DONE UNTIL ON THE FIRST PAGE
V

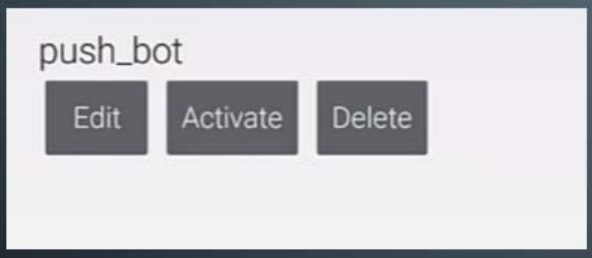


TAP SAVE AND NAME THE CONFIG

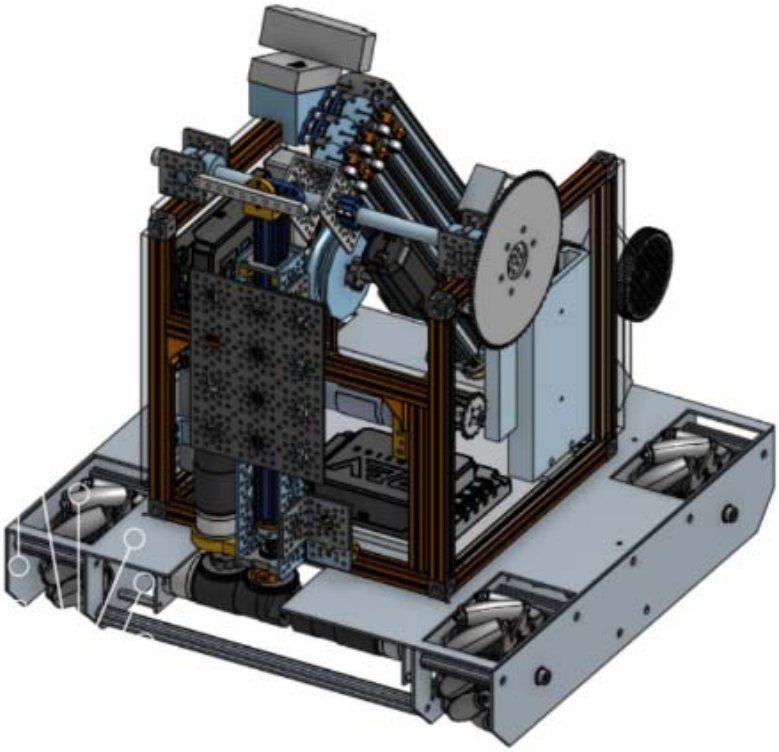




NOW HIT ACTIVATE



- You are now done!



WASHY 3.5

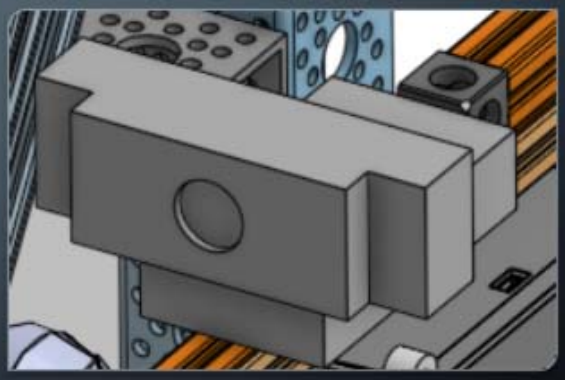
- SIZE: 17.5 IN X 17.5 IN X 16.5 IN
- WEIGHT: 28 LBS.
- IN-GAME: -FULL AUTONOMOUS (80PTS)
- -RAISE IN ENDGAME (50 PTS.)
- - SCORE MINERALS IN LANDER (60 PTS.)

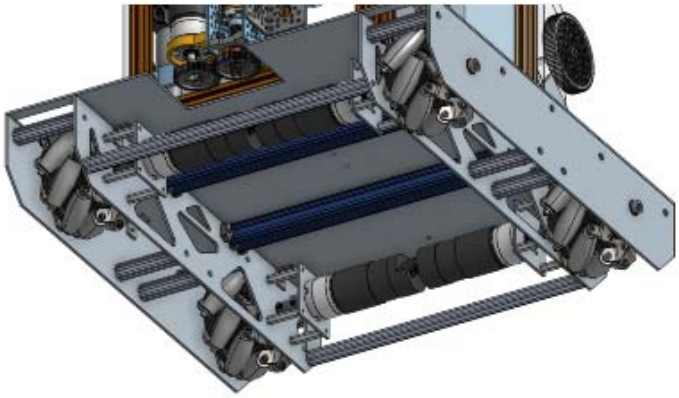
MARKER PLACER

- Our marker placer is composed of a servo attached to a beam. Then the marker slides on the beam. The servo rotates thus dropping our marker in the depot

TENSORFLOW CAMERA

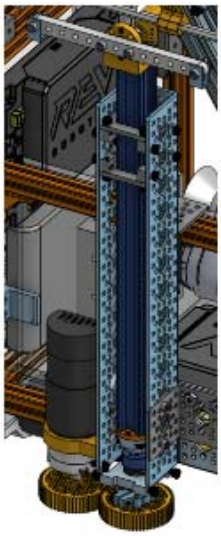
- We use a 1080p webcam attached to a battery-powered USB hub to use TensorFlow that we use in our autonomous





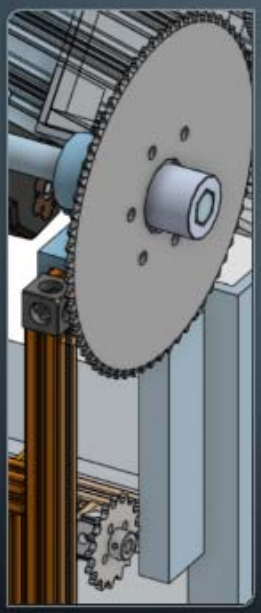
WHEEL BASE

TO DEAL WITH CAMBER (FLEX IN THE MOTOR MOUNTS) WE DESIGNED A BASE THAT WOULD TAKE THE STRESS OFF THE MOTOR. WE USED CAD AND HAD 5052 ALUMINUM CUT. WE USE CHURRO TUBES TO PROVIDE STABILITY AND SERVE AS A GUARD FOR MINERALS. THE MOTORS ARE NEVEREST ORBITAL 20:1 GEAR MOTORS.



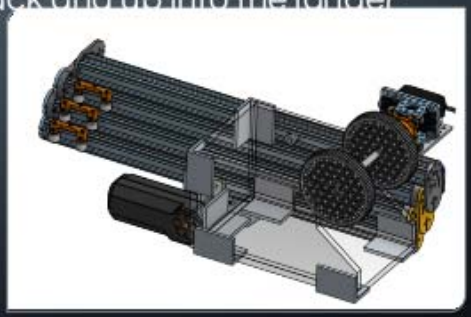
LINEAR ACTUATOR

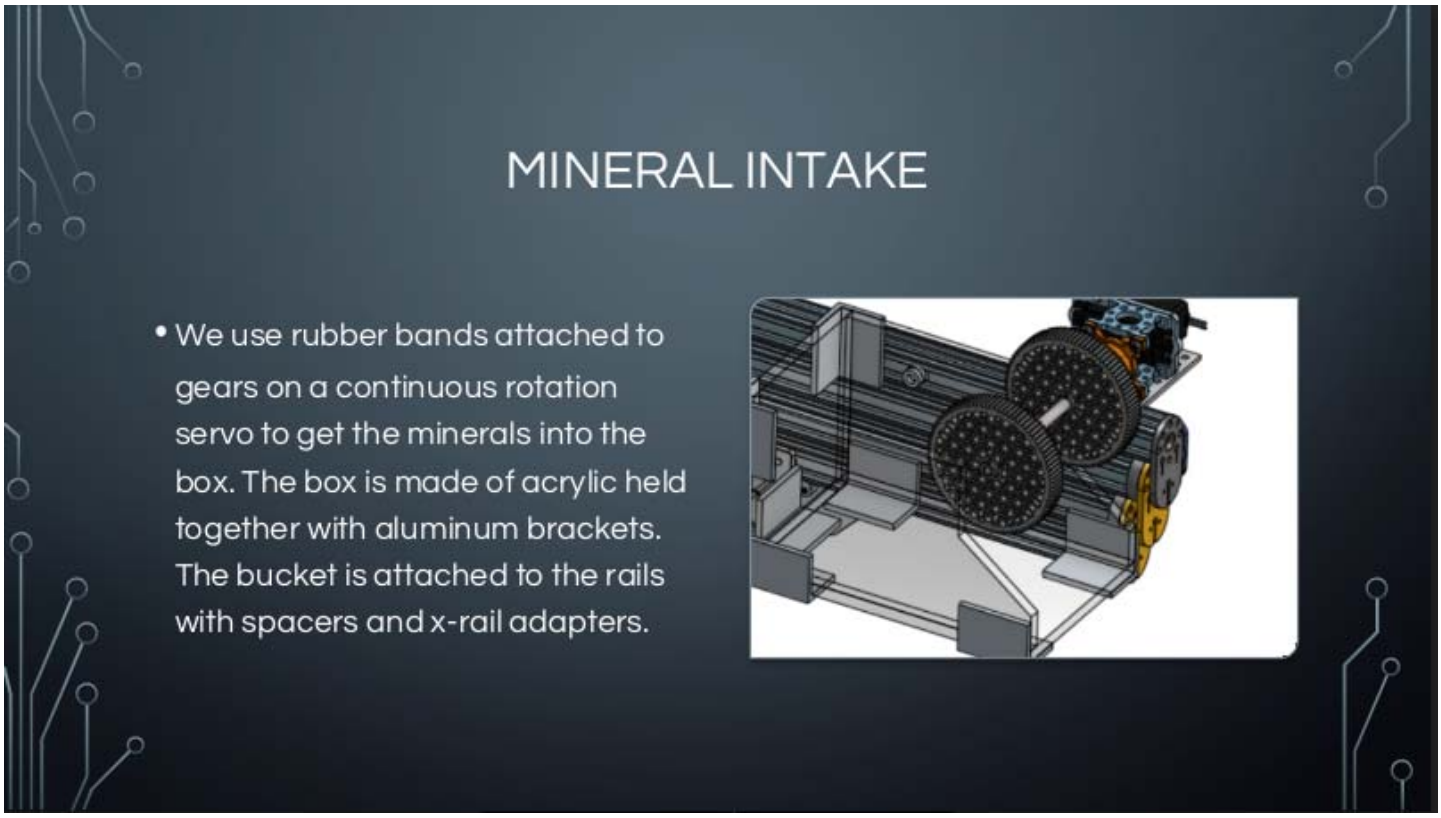
- Our linear actuator allows us to raise and lower ourselves from the lander in 2.5 seconds and is extremely reliable



LINEAR ARM

- Our linear slide is attached to metal sprockets and with a chain we can score minerals in the lander by extending the arm into the crater, using our intake system to grab the minerals then rotate it back and up into the lander



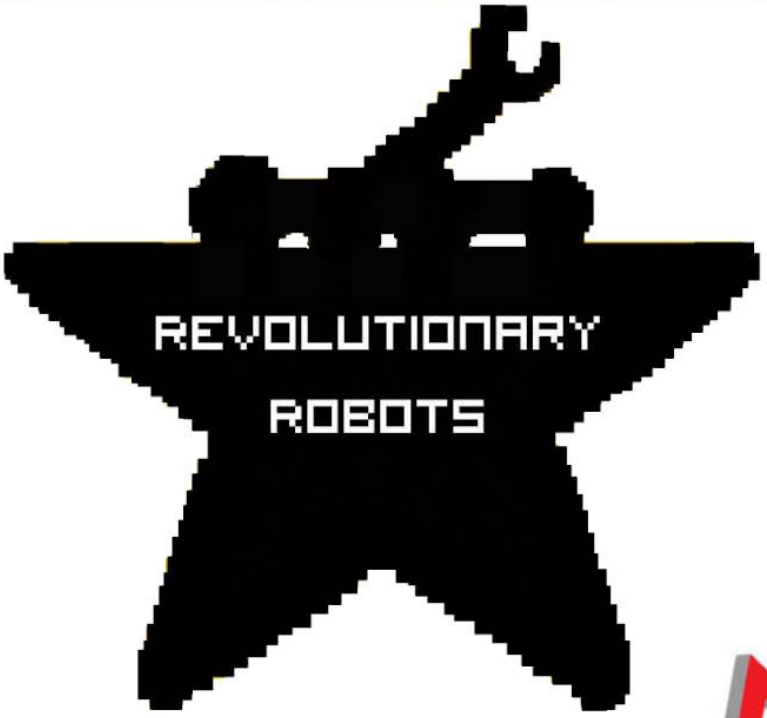


Other Team Education

We encourage our team members to shadow others to learn and understand the program they are using. This way we can promote institutional knowledge to keep our team going after people leave.

Revolutionary Robots 12535

Checklists 2018-2019



Moore Community STEM Club

FIRST®

Between Match Checklist

Linear Slide and Linear Actuator

- ❑ Check screws on linear slide wheels
- ❑ Check string/spool
- ❑ Check rubber bands
- ❑ Check surgical tubing
- ❑ Check right angle with sprockets
- ❑ Check chain
- ❑ Check gears on LA
- ❑ Tighten set screws on the collars on the rod
- ❑ Check the end caps on the slide
- ❑ Check Hub connection points
- ❑ Servo Screw
- ❑ Screws on Sprocket
- ❑ Screw on hubs of Linear Slide

Base and Acrylic Body

- ❑ Tighten all corner pieces
- ❑ Tighten the brackets on the base
- ❑ Check set screws on the wheel nub
- ❑ Check the set screws on couplers

Miscellaneous parts

- ❑ Check TensorFlow camera
- ❑ Clean wheels (not between every round)
- ❑ Tighten wheels

Electrical Parts

- ❑ Check wired connections
- ❑ Check phone connection
- ❑ Check battery level
- ❑ Check the USB hub