



Western University of Canada  
Faculty of Engineering



Western  
UNIVERSITY • CANADA

MSE 2202 – Introduction to Mechatronic Design

## **Team Project – Tesseract Robot**

by

Andrew Randell, Fan Kang, Lovdeep Singh, Maral Mohagheghi

Product Development File

Group 9, Prof. Naish

Date Submitted: April 11, 2018

# Table of Contents

## Contents

Specification and Planning .....	4
Problem Statement.....	5
Meeting Minutes #1.....	5
Product Design Requirements.....	6
Meeting Minutes #2.....	6
Discussion.....	6
To be completed:.....	7
Design Requirements/ Specifications: .....	8
Program Plan .....	11
Work Plan .....	11
Conceptual Design.....	12
Meeting Minutes #3.....	13
Conceptual Design Development .....	14
Pyramid Manipulation .....	14
Pyramid Storage .....	15
Meeting Minutes #4.....	16
Cube Manipulation .....	17
Cube Storage and Release.....	19
Cube Location.....	19
Drive Concepts .....	20
Concept Evaluation .....	23
Cube Manipulation .....	23
Cube Storage and Release .....	23
Cube Location.....	23
Pyramid Manipulation .....	24
Pyramid Storage .....	24
Pyramid Location .....	24
Drive Design .....	24
Lift Design.....	24
Electronic Design.....	25
Product Design .....	27

Project Generation.....	28
Meeting Minutes #5.....	28
Cube Manipulation and Storage.....	30
Cube Location.....	31
Drive Design .....	36
Full Product Design.....	38
Full Prototype Design.....	38
Engineering Analysis.....	48
Drive .....	48
Lift and Claw Arm.....	49
Pyramid Intake.....	49
Detailed Documentation .....	50
Product BOM.....	50
Product Technical Drawings .....	52
Prototype BOM .....	72
Prototype Technical Drawings.....	74
Final Code.....	77
MSE-bot.h.....	77
MSE-bot.cpp .....	81
Main.ino .....	96
Product Evaluation .....	100
Cost of External Parts sourced for Prototype .....	101
Sourcing of External Parts .....	102

# Specification and Planning

# Problem Statement

Create an autonomous system that locates and retrieves a tesseract and places it under the pyramid that is emitting the correct signal.

## Meeting Minutes #1

**Date:** February 15, 2018

### **Agenda:**

- Define/ Understand Problem Statement
- Break down the problem into smaller tasks

### **Discussion:**

#### **1. Problem Statement**

- a. create an autonomous system that locates and retrieves a tesseract and places it under the pyramid that is emitting the correct signal.

#### **2. Step by Step breakdown**

- Locate the cube
- Retrieve the cube
- Locate the correct pyramid
- Pick up pyramid
- Place pyramid over cube

#### **3. Concerns:**

- The robot's motion might not be able to effectively move over the conduits
  - Friction calculations required to select appropriate motors and wheels
  - Unsteady motion over the conduits may result in the loss of the location of the pyramid or tesseract when locating them
- Cube might be pushed over the wall during cube retrieval

### **To be completed:**

1. Brainstorm possible strategies for locating the tesseract
2. Compile several solutions for tesseract and pyramid retrieval
3. Research possible sensors that could be used in the products

# Product Design Requirements

## Meeting Minutes #2

Date: February 22, 2018

### Agenda:

- Analyze field setup and discover specific problems that we feel we must overcome in our robot design.
- Define design requirements
- Begin sketching implementations of ideas and analyzing the plausibility and effectiveness of these designs.
- Generate a list of questions to be answered both internally and externally.

### Discussion

#### Analyzing Field Setup

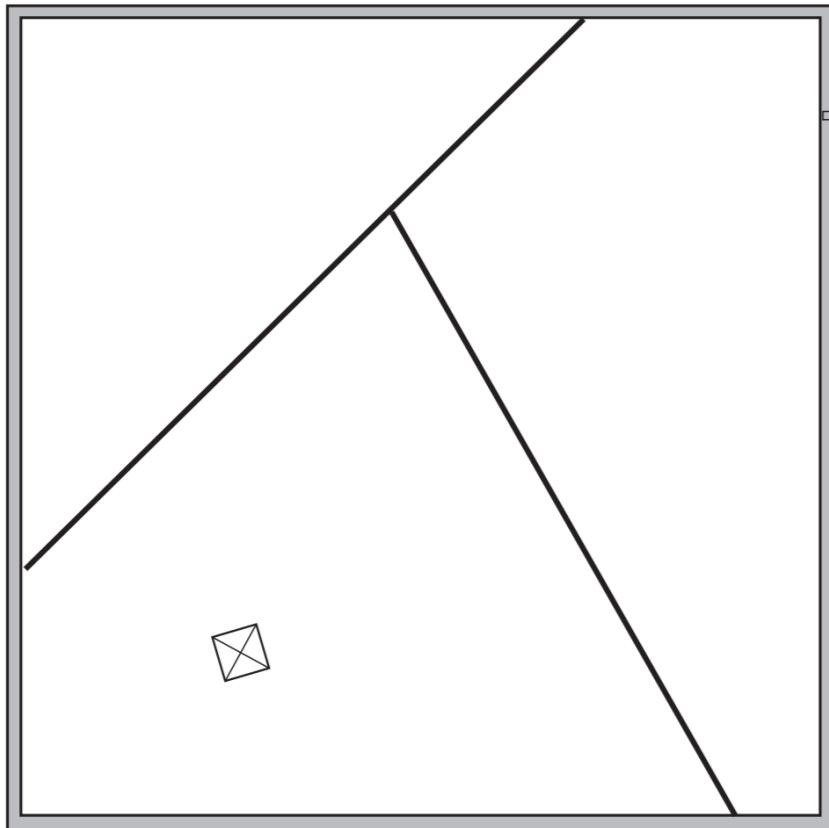


Figure 1: Field Setup

- Encoders, ultrasonic range finders and other sensors that give the robot a sense of placement will be vital for an autonomous period within the open field.

## Questions

Question	Approach
How precise is pyramid hole size compared to cube?	Eugen
Angle of pyramid	Eugen
How much of our bot can be laser cut / 3d printed (within budget?)	Eugen
How do we use suction cups with arduino?	Google
Friction coefficients with lexan?	Google
Hall effect sensors	Google
Photosensitive (IR) sensors	Google
Ultrasonic sensors	Google
Can we use outsourced motors	Naish
What are the specs on the VEX sensors Prof. Naish has for us	Owl

Figure 2: Questions Table

### To be completed:

- Analyze the implementation of various designs and begin discussing best choices for first prototype.
- Discuss financial budget and parts that are essential for building such designs.

## Design Requirements/ Specifications:

Considering the Problem Statement a list of consumer needs was compiled. These consumer needs then lead to a set of engineering requirements that were used to set up a Quality Function Deployment (QFD).

The main goal of the robot being designed is to reliably and consistently locate and place the tesseract under the correct pyramid. To do this each of the tasks the problem is broken down into must be completed effectively. More specific requirements were then generated for each part of the task after the task was broken down into multiple steps. These task specific requirements were used to evaluate the validity of the concepts generated during the concept generation process. The following sub-requirements were generated:

1. being able to navigate the course without being hindered by the conduits
  - a. Move straight forwards or backwards
  - b. Do not slip over the conduits
  - c. Be able to turn over the conduits
2. Locating and retrieving the tesseract
  - a. Consistently locate the tesseract no matter the location
    - i. modular
  - b. Accurately locates the tesseract
  - c. Does not push the tesseract somewhere unreachable during the locating process
  - d. Does not lose/ drop the tesseract while navigating the course after retrieval
3. Locating the correct pyramid and retrieving it
  - a. Consistently reads the signal from each pyramid
  - b. Consistently selects the correct pyramid
  - c. Does not lose the signal after locating the pyramid
  - d. Effectively pick up the pyramid
  - e. Do not drop the pyramid

Row Number	Demanded Quality (a.k.a. "Customer Requirements" or "Whats")	Weight / Importance	Relative Weight
1	Fast	4	13.33
2	Reliable	5	16.67
3	Inexpensive	2	6.67
4	Easy to maintain	2	6.67
5	Easy to set up	3	10.00
6	Aesthetically pleasing	1	3.33
7	Fuel efficient	2	6.67
8	Clean operations	5	16.67
9	Reasonably sized	4	13.33
10	Incorporate latest technology	2	6.67

Figure 3: Customer Requirements QFD

Relationship Between Requirements:													
9 - Strong    3 - Moderate    1 - Weak													
Row Number	Max Relationship Value in Row	Column Number	1	2	3	4	5	6	7	8	9	10	
		Max Relationship Value in Column	9	9	9	9	9	9	9	9	9	9	
		Requirement Weight	120	150	60	96.67	90	30	76.67	200	150	76.67	
		Relative Weight	11.43	14.29	5.71	9.21	8.57	2.86	7.30	19.05	14.29	7.30	
		Difficulty (0=Easy to Accomplish, 10=Extremely Difficult)											
		Minimize (▼), Maximize (▲), or Target (x)	▼	▲	▼	▲	▼	▲	▼	x	▼	▲	
		Target or Limit Value	3	100	1000								
		Quality Characteristics (a.k.a. "Functional Requirements" or "Hows")	Completes task within x minutes	Completes task x percent of time	Final product costs x dollars	Needs to be maintained after x uses	Product begins operations after x inputs	x percent of surveyed individuals indicate that product is aesthetically pleasing	Product consumes x watts of power	Product does not drop any debris during operations	Product is within 25x50x80 cubic cm areas	Product incorporates latest technology in x	
Row Number	Max Relationship Value in Row	Relative Weight	Demanded Quality (a.k.a. "Customer Requirements" or "Whats")	Completes task within x minutes	Completes task x percent of time	Final product costs x dollars	Needs to be maintained after x uses	Product begins operations after x inputs	x percent of surveyed individuals indicate that product is aesthetically pleasing	Product consumes x watts of power	Product does not drop any debris during operations	Product is within 25x50x80 cubic cm areas	Product incorporates latest technology in x
1	9	13.33	Fast	9									
2	9	16.67	Reliable		9	1					3		
3	9	6.67	Inexpensive		9							1	
4	9	6.67	Easy to maintain			9							
5	9	10.00	Easy to set up				9				3		
6	9	3.33	Aesthetically pleasing					9				3	
7	9	6.67	Fuel efficient			3			9				
8	9	16.67	Clean operations							1	9		
9	9	13.33	Reasonably sized								9		
10	9	6.67	Incorporate latest technology									9	

Figure 4: Functional Requirements QFD

Row Number	Quality Characteristics (a.k.a. "Functional Requirements" or "Hows")	Correlations: Positive (+) or Negative (-)									
		Column Number	1	2	3	4	5	6	7	8	9
	Completes task within x minutes										
4	Needs to be maintained after x uses	-	-								
5	Product begins operations after x inputs										
6	x percent of surveyed individuals indicate that product is aesthetically pleasing		-								
7	Product consumes x watts of power	-			-						
8	Product does not drop any debris during operations	-	+	-	-						
9	Product is within 25x50x80 cubic cm		-	+			+	+	+		
10	Product incorporates latest technology in x areas	+	+	-	+		+	-	+	-	

Figure 5: Functional Requirements Roof QFD

Row Number	Quality Characteristics (a.k.a. "Functional Requirements" or "Hows")	Minimize (▼), Maximize (▲), or Target (x)	Target or Limit Value	Max Relationship Value	Requirement Weight	Relative Weight (Relative Importance)
1	Completes task within x minutes	▼	3	9	120.00	11.43%
2	Completes task x percent of the time	▲	100	9	150.00	14.29%
3	Final product costs x dollars	▼	1000	9	60.00	5.71%
4	Needs to be maintained after x uses	▲	10	9	96.67	9.21%
5	Product begins operations after x inputs	▼	2	9	90.00	8.57%
6	x percent of surveyed individuals indicate that product is aesthetically pleasing	▲	75	9	30.00	2.86%
7	Product consumes x watts of power	▼	10	9	76.67	7.30%
8	Product does not drop any debris during operations	x	0	9	200.00	19.05%
9	Product is within 25x50x80 cubic cm	▼	25x50x80	9	150.00	14.29%
10	Product incorporates latest technology in x areas	▲	2	9	76.67	7.30%

Figure 6: QFD Summary

# Program Plan

## Work Plan

<b>Week</b>	<b>Deliverable</b>
February 15	Stuff to do for reading week, general direction discussion
February 22	Prelim ideas, CAD of arm concepts, research
March 1	First design review ready, tasks assigned for prototype
March 8	Work on prototype build, most parts sourced
March 15	Work on prototype build and coding
March 22	Second design review ready, prototype complete
March 29	Final parts laser cut or otherwise created
April 5	Robot complete
April 11	Report complete

Figure 7: Work Plan

# Conceptual Design

# Meeting Minutes #3

Date: February 24, 2018

## Agenda:

- Discuss strategies for locating the tesseract
- Discuss designs for cube and pyramid retrieval
- Discuss sensors and actuators that could possibly be used in the design

## Discussion:

### 1. Strategies for locating the tesseract

- Move to the center of the field and spin slowly and scan all four walls looking for the cube
  - Hall effect sensors have quite a short range so if this strategy is to be used, sensors with a longer range should be used.
- Go around the entire field and scan the entire length of the wall until the cube is located
  - Hall effect sensors could be used
  - Must be mind full of not knocking the cube off the wall just before locating it as we got around the wall
- Design a retrieval mechanism that picks up anything that is on the wall
  - No need to find the cube before retrieving it
  - We would just need to know if we have the cube
    - A limit switch or hall effect sensors could be used

### 2. Possible Pyramid retrieval designs

- Each member brainstormed and sketched several ideas individually
- The ideas were looked at by all the members and the duplicates were removed
- Sketches or cad drawings of the pyramid pick up ideas are attached below

### 3. Sensors:

- A document is created for the purposes of compiling a list of possible sensors that can be used for the product
  - This document contains the name of the product, locations where the product can be sourced from, the price of the products and a link to the spec sheet for the product
  - This file will be regularly updated as possible sensors are further researched and the group strategies for completing the task continues to evolve.

## To be completed:

1. Finalize the pyramid pick up concepts
2. Complete the required concept selection processes for the pyramid pick up (i.e. Decision matrix ...)

## Conceptual Design Development

To generate design concepts, each team member developed sketches and models for each particular task or challenge.

### Pyramid Manipulation

#### Flap Intake Mechanism Design Concept (CAD)

The flap design with a 30-degree angle ramp based on belts and rollers in factory production lines is developed to verify if the pyramid and cube can both be manipulated with one intake mechanism, or if separate claw-type mechanisms are needed for cube manipulation.

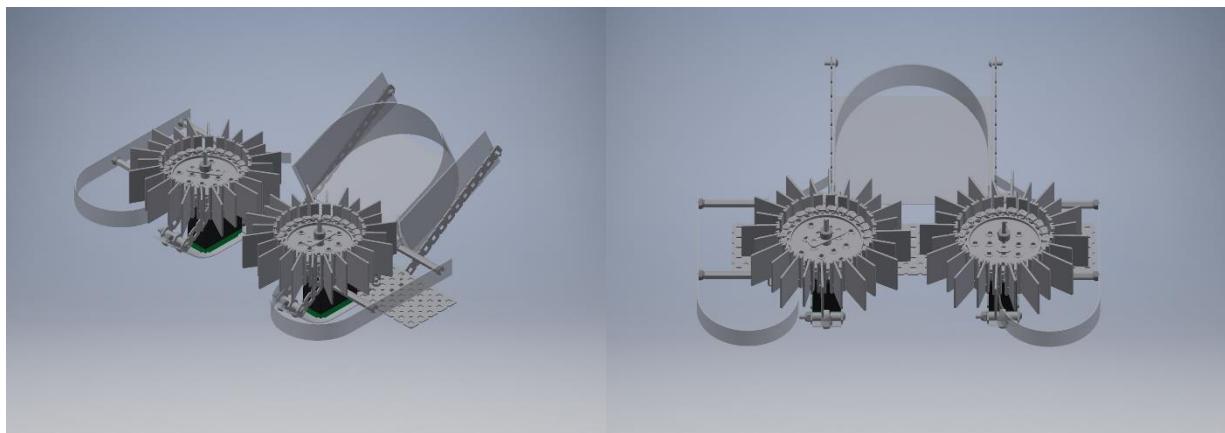


Figure 8: Flap Intake CAD Model

#### Roller Intake Design Concept

The pyramid roller intake aims to make direct contact with the upper walls of the pyramid, spin opposite to each other to use friction to push the pyramid into a ramp holder. Measures would need to be made to acquire the right type of material (possibly rubber) for the rollers.

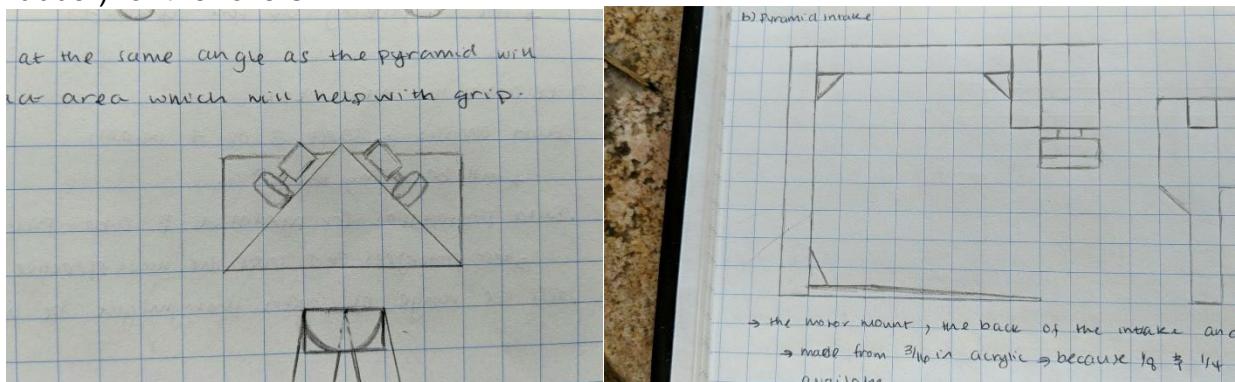


Figure 9: Roller Intake Sketches

## Vertical Suction Cups

Using a single servo, two arm links rotate to close the vertical claw onto the pyramid, at which point the suction cups push onto the pyramid and lift it. To increase the clamping force, a gear train with a larger output gear can be pursued.

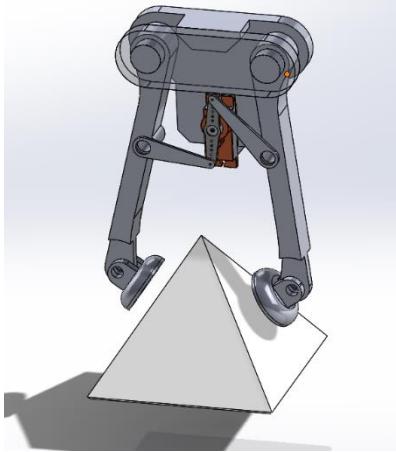


Figure 10: Vertical Suction Mechanism CAD Model

## Pyramid Storage

It was decided that having the ability to store and move the pyramid would allow for more flexibility and reliability in design. Without storage, the robot would need to have retrieved the cube and transfer it underneath the pyramid immediately after the pyramid is retrieved, which would both limit design, programming, and marketability of the product. As such, each design is able to contain and maneuver the field through the use of an elevated ramp or suction cups.

## Pyramid Location

As a design requirement is to identify the correct pyramid transmitting a code via IR, an IR sensor is necessary. In addition to searching for the pyramid through IR, an ultrasonic sensor may be used to verify the position of the pyramid.

# Meeting Minutes #4

Date: March 2, 2018

## Agenda:

- Discuss Designs for Cube retrieval
- Discuss possible chassis designs

## Discussion:

### 1. Designs for cube retrieval

- Each member brainstormed cube retrieval methods individually
- The concepts generated by the members were examined and the duplicates were removed.
- A drawing or cad model of the concepts generated are available on the next page.
- The group originally considered modifying the pyramid retrieval mechanism to pick up the tesseract as well to simplify our design but this would require the pyramid intake to be able to become much smaller than it needs to be when intaking the pyramid. This design would also require the intake to reach above a wall.
  - Due to the above challenges, it was observed that although using one mechanism would decrease the number of components needed, it would increase the complexity of the intake design and lower its reliability.
- The remaining designs are as follows
  - Magnetic Intake: **Figure 11**: Magnetic Intake CAD Model
  - Arm Claw: **Figure 12**: Arm Claw Sketches
  - Rail Claw: **Figure 13**: Rail Claw Sketches
- **Note** – Depending on the design selected for cube retrieval our strategy from locating the cube might change
  - As described in meeting minutes #2 from February 24, 2018

### 2. Chassis Design

- A four-wheel chassis design was selected over a two or three-wheel design to gain higher stability
  - This will allow the product to navigate the conduits in a more consistent and stable manner
    - A three-wheel design with a passive front wheel will be less consistent so the path of the robot would be harder to predict as it moves over the conduits
  - **Concerns**: The 4-wheel design might make it harder for the robot to turn in tight corners
- One motor controlling each side would ensure that the two wheels on one side of the robot are moving in sync and are not fighting against one another at any point. This will make the product easier to program.
- A chain can be used to drive the two wheels on each side using one motor.

- The chain allows us to adjust the distance between the wheels to reach a balance between the consistency of the drive and the space required for the cube and pyramid mechanisms to be mounted on.
- Calculations are required to find effective and efficient motors to use.

**To be completed:**

1. Research possible motors and wheels that can be used for the drive base
2. Complete the required concept selection processes for the cube retrieval mechanism (i.e. Decision matrix ...)

## Cube Manipulation

### Magnetic Intake Design Concept

This design involves using a rotating mechanism with various flaps attached to it. A magnet would be attached to each of these flaps. As a result the mechanism can pick up the pyramid anywhere on the wall as the robot goes around the wall. This design does not require us to know where the tesseract is. With this design we would only need to know when the tesseract is retrieved by the robot. A secondary mechanism must then be used to remove the cube from the magnet.

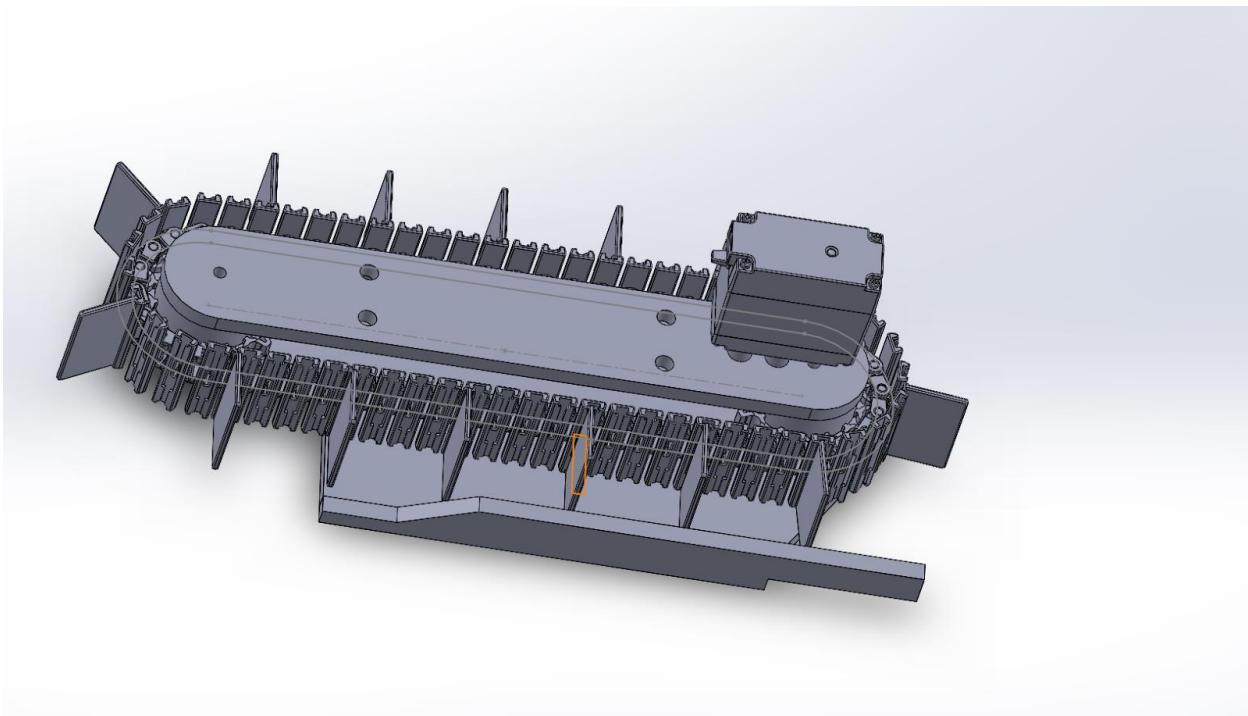


Figure 11: Magnetic Intake CAD Model

## Arm Claw

The arm would be mounted on the robot at the same height as the wall, and can pivot to adjust for minor height deflections. The claw design is much more reliable than the intake for picking up and securing the tesseract, and the arm can rest over the wall while the robot drives along the perimeter of the arena.

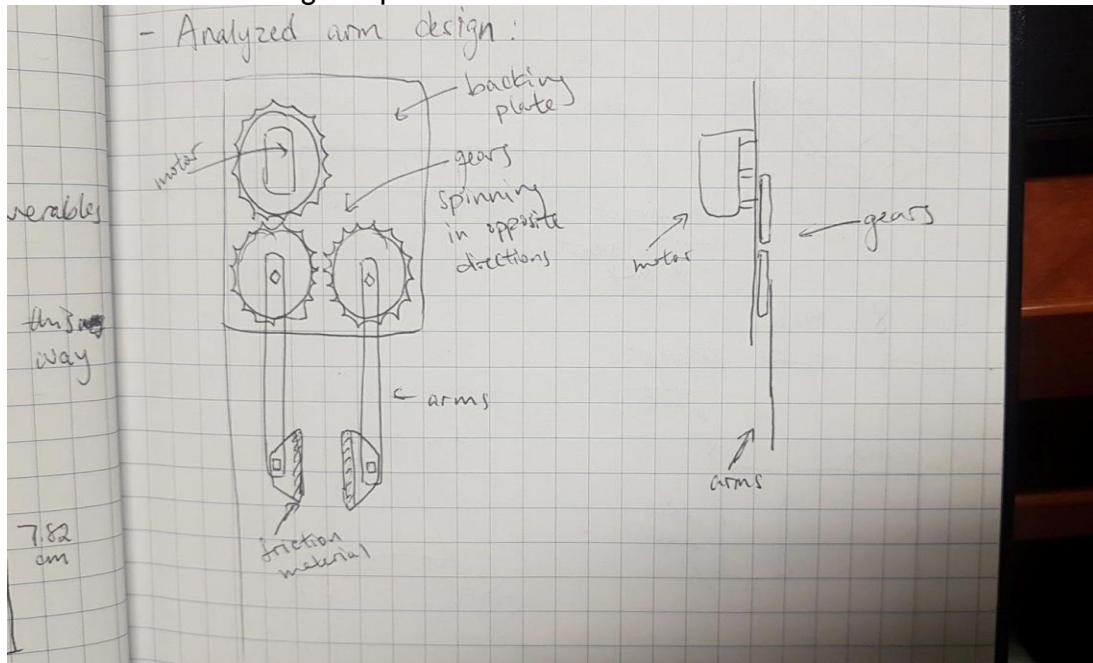


Figure 12: Arm Claw Sketches

## Rail Claw:

Utilizes the same claw design as the arm claw, but retracts and extends the claw by mounting it at the end of a rack and pinion. The tesseract would be retrieved and controlled by retracting the rack and dropping it in a funnel on the robot. This eliminated the problems of powering a motor to hold horizontal position and the tesseract bouncing randomly, increasing the reliability of the entire procedure.

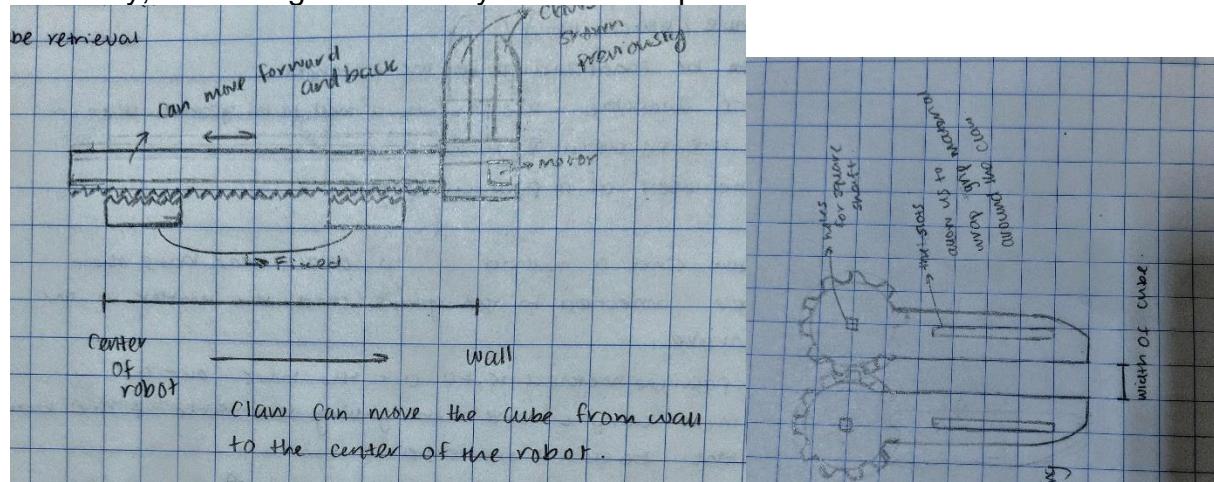


Figure 13: Rail Claw Sketches

## Cube Storage and Release

To store the cube as the robot begins to search for the pyramid, all three retrieval mechanisms are able to keep the cube inside the mechanism. To release the cube, however, the magnetic intake requires a stopper or a motorized mechanism to remove the cube from the magnets. A ramp or a funnel can be used to direct the cube to the bottom of the robot, close to the pyramid mechanism (**Figure 14: Ramp Sketches**).

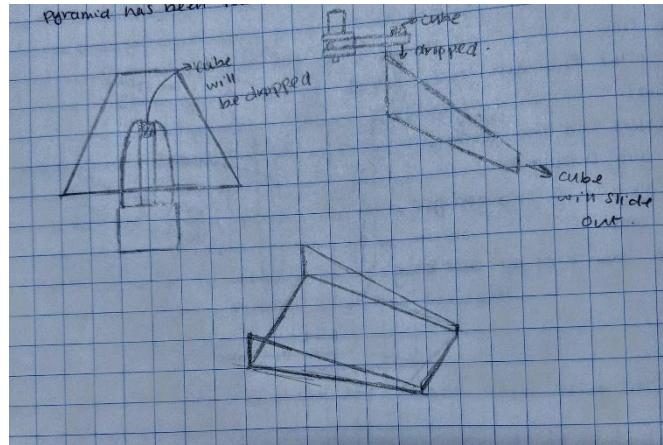


Figure 14: Ramp Sketches

## Cube Location

Since the tesseract contains a magnet, a hall effect sensor can be used to detect the magnetic field created by the tesseract. In addition, a magnetometer or a compass can be used to take advantage of the magnet inside the cube. Both of these sensors, however, could pickup magnetic fields from other ferrous parts of the robot and so it must be ensured that the cube manipulation mechanism is built as nonferrous as possible. As the magnetic intake mechanism would inherently be problematic, the intake could constantly be rotating along the wall, and when the cube is retrieved, a limit switch or ultrasonic range finder could detect possession.

**Note** - The concepts modeled below were modeled to provide a clear visual basis for the ideas presented. This ensured that all members fully understood each concept. The use of VEX parts allowed the group to complete these models quickly which resulted in more efficient concept generation sessions. The models below are not a representation of the final product.

## Drive Concepts

### Chassis Heights

**Low Chassis:** Pros: Easier to implement. Gives you more space to build. Cons: Would need more wheels and perhaps more than 2 motors. May not go over the conduits.

**High Chassis:** Pros: Can go over the conduits with ease. Will at most require four wheels although more are possible. Cons: Loses some build space. More materials are required to elevate chassis.

### Mecanum Drive Design Concept (CAD)

The four-wheel drive train proposed will be elevated >1 inch through the use of 4-inch wheels to be able to drive over the conduits in the power plant. A holonomic drive is preferred for the ability to drive in all directions without turning to best locate the cube, and will be experimented with to support this decision.

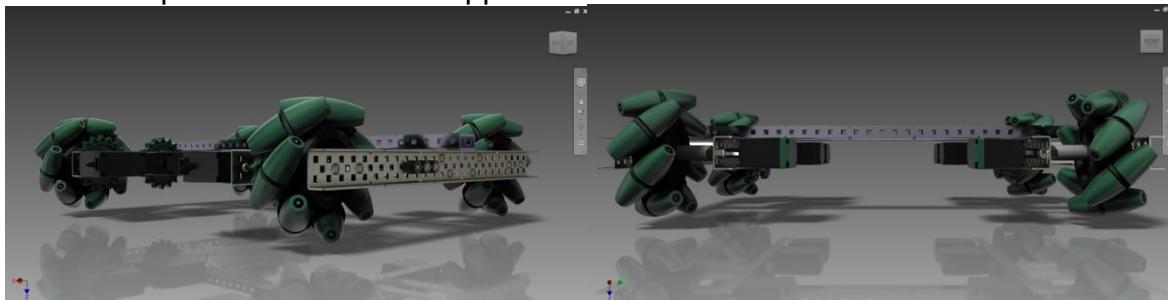
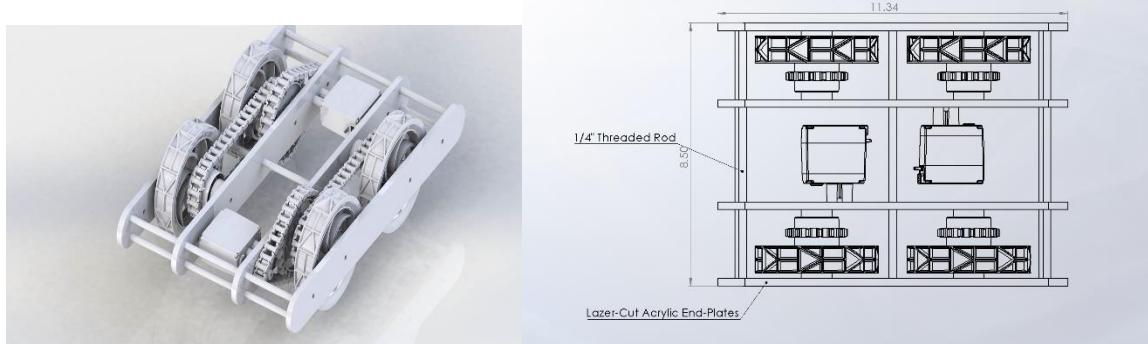


Figure 15: Mecanum Drive CAD Model

### Tank Drive Design Concept (CAD)



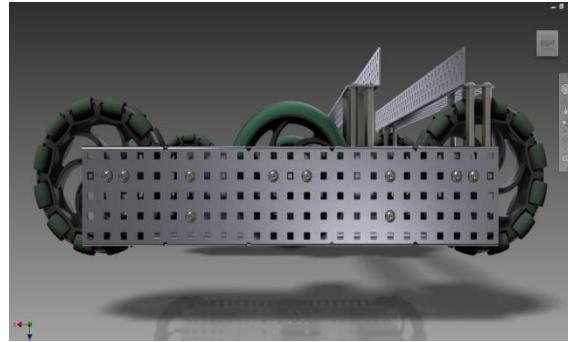


Figure 16: Tank Drive CAD Model

This chained tank drive allows the use of only two motors, with power distributed to both wheels on either side. This is a much simpler design that would make programming more reliable.

### X-Drive Holonomic

A 4 motor drive with 4 wheels angled at 45 degrees allows for maneuvering forward and backwards, and left and right.

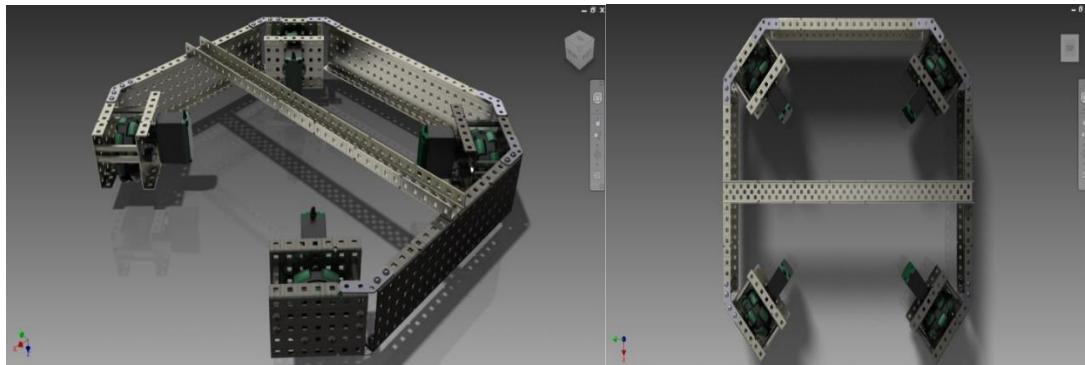


Figure 17: X-Drive CAD Model

### H-Holonomic Drive

An additional motor powering a perpendicular wheel allows the robot to maneuver left and right. May compromise space.

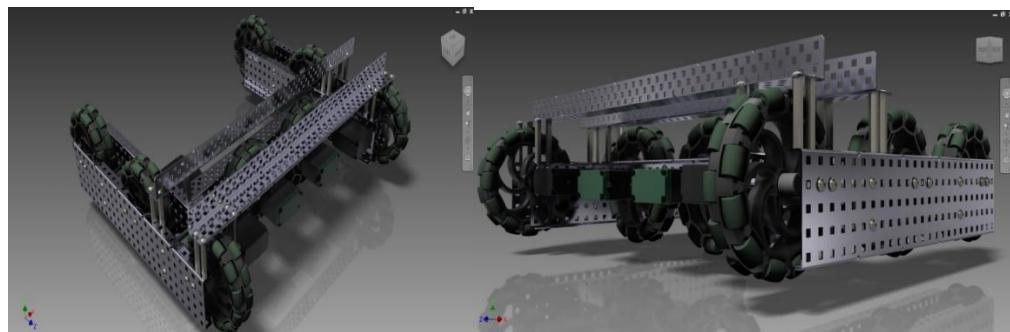


Figure 18: H-Holonomic Drive CAD Model

## Lift Concepts

**Lift Design Concept:** If a need for a lift is found, it is suggested the lift be a linear lift to best maneuver the tesseracts and Pyramids autonomously, or a simple non-linear lift design for reliability.

### Simple Lift

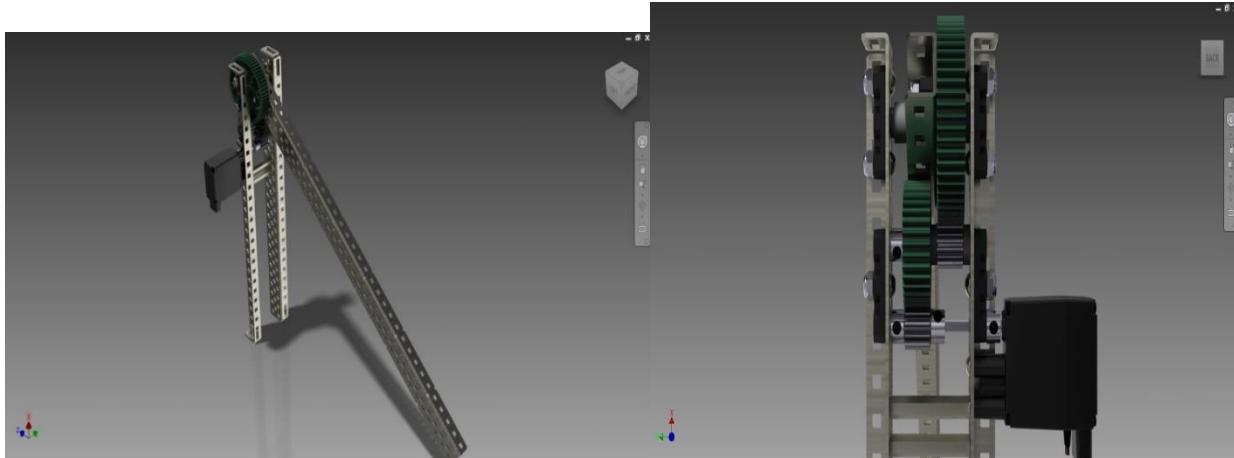


Figure 19: Simple Lift CAD Model

### Chain Bar

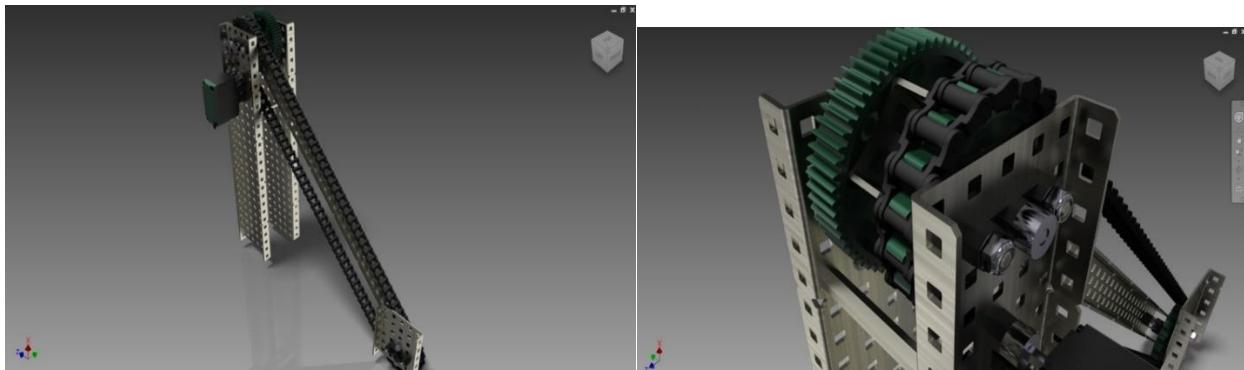


Figure 20: Chain Bar CAD Model

### 4-Bar Linkage

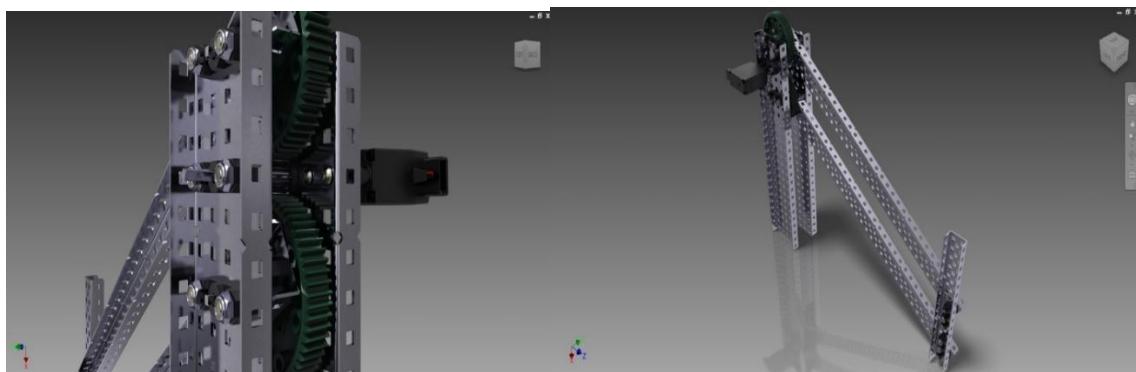


Figure 21: 4-Bar Linkage CAD Model

## Concept Evaluation

After concepts were fully discussed and defined, they were objectively compared through decision matrices to select the best concept. Some design decisions were made during discussion or selected based on other concept selections.

### Cube Manipulation

Decision Matrix - Tesseract Retrieval Mechanisms								Options			
	Reliability	Size	Complexity	Speed	Cost	Total	Relative Weight	Intake	Arm Claw	Rail Claw	Mag. Intake
Reliability	1	2	4	4	4	15	0.44444444	-1	0	1	1
Size	0.5	1	2	2	2	7.5	0.22222222	-1	0	0	-1
Complexity	0.25	0.5	1	1	1	3.75	0.11111111	-1	0	-1	-1
Speed	0.25	0.5	1	1	1	3.75	0.11111111	0	0	0	0
Cost	0.25	0.5	1	1	1	3.75	0.11111111	0	0	0	-1
<b>Total</b>						<b>33.75</b>	<b>1</b>	<b>-0.7777778</b>	<b>0</b>	<b>0.33333333</b>	<b>0</b>

Figure 22: Cube Manipulation Decision Matrix

Out of the three designs, it was decided through a decision matrix that the rail claw would be best for cube manipulation due to its ability to reliably retrieve the cube at large heights and lengths from the robot and bring it to a position underneath the robot. The Arm claw would require the cube to drop into an indeterminable position and the magnetic intake would result in a lack of sensor data, although possibly a quicker method.

### Cube Storage and Release

The claw's servo closed position will be set greater than perpendicular to the cube and along with the use of a gripping material, the claw will be able to securely store the cube. A ramp at the center of the robot was chosen to best support the rail claw, which would utilize a rack and pinion rail to bring the claw and cube into the center and drop it through the funnel created by the ramp.

### Cube Location

Since the cube retrieval mechanism will not use any magnets, a magnetometer will be used to detect the location of the cube. A compass magnetometer has the ability to detect magnetic fields in all three directions (x, y, z) which will give a much more accurate reading than a hall effect sensor, which only provides a reading within inches to the magnet, after initial tests.

### Tesseract Sensor Tests

Distance Detected (cm)	10	5	3	2	1
Magnetometer	No	Yes	Yes	Yes	Yes
Hall Effect	No	No	No	Yes	Yes

Figure 23: Tesseract Sensor Tests

## Pyramid Manipulation

Decision Matrix - Pyramid Retrieval Mechanisms								Options		
	Reliability	Size	Complexity	Speed	Cost	Total	Relative Weight	Claw	Roller Intake	Arm Intake
Reliability	1	3	4	3	4	15	0.43902439	-1	0	1
Size	0.33333333	1	2	1	2	6.33333333	0.18536585	1	0	1
Complexity	0.25	0.5	1	0.5	1	3.25	0.09512195	1	0	0
Speed	0.33333333	1	2	1	2	6.33333333	0.18536585	1	0	-1
Cost	0.25	0.5	1	0.5	1	3.25	0.09512195	1	0	0
Total						34.1666667	1	0.12195122	0	0.43902439

Figure 24: Pyramid Manipulation Decision Matrix

The decision matrix resulted in the Arm Intake to be selected. The Arm Intake has a reliable method of containing and releasing the pyramid with a large normal force. The claw may face issues with spontaneously dropping the pyramid if the suction cups were not strong enough and the roller intake's large nature would be unpredictable and would most likely require a wall to push up against.

## Pyramid Storage

The pyramid would be stored in a forklift type ramp which has an indented cutout at the bottom to allow for the cube to be placed underneath the pyramid. When this is done, both the pyramid and cube could be released onto the field, with the challenge completed.

## Pyramid Location

It was decided to use the IR sensor to detect the pyramid as determining which pyramid was the correct one is essential.

## Drive Design

While several drive designs were developed, after the initial status report meeting, it was determined that holonomic drives would lead to several problems during the autonomous run, including the robot moving left and right without a good way of detecting this motion. As such, the tank drive using 4-inch wheels was selected, as this would be high enough to move across conduits and be a reliable drive to program. Two distance sensors on the side of the robot will be used to follow the wall at a distance to search for the cube, while a distance sensor at the front will be used to detect when a turn is to be made.

## Lift Design

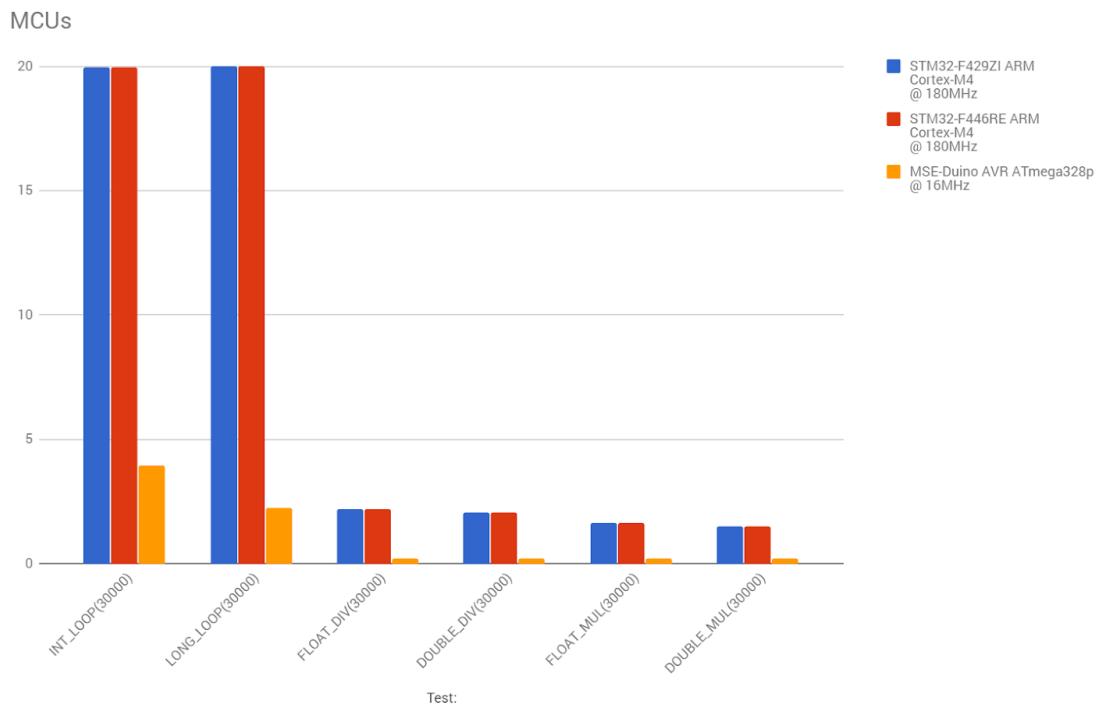
As the Arm Intake mechanism was pursued, a lift is required to elevate the pyramid above the cube and maneuver over the conduits without the intake hitting them. A horizontal four-bar design was selected due to its strength torque versatility.

## Electronic Design

While the MSE-Duino provides functionality due to the two microcontrollers, other ARM and AVR based microcontrollers were also considered for higher speed of sensor computation and more I/O.

	STM32-F429ZI	STM32-F446RE	MSE-Duino	Arduino Mega		
Test:	ARM Cortex-M4 @ 180MHz	ARM Cortex-M4 @ 180MHz	AVR ATmega328p @ 16MHz	AVR ATmega2560 @ 8MHz		Arm Improvement over AVR
INT_LOOP(30000)	19.95	19.95	3.98		MIPS	<b>501.26%</b>
LONG_LOOP(30000)	19.99	19.99	2.24		MIPS	<b>892.41%</b>
FLOAT_DIV(30000)	2.22	2.22	0.21		MFLOPS	<b>1057.14%</b>
DOUBLE_DIV(30000)	2.05	2.05	0.21		MFLOPS	<b>976.19%</b>
FLOAT_MUL(30000)	1.63	1.63	0.21		MFLOPS	<b>776.19%</b>
DOUBLE_MUL(30000)	1.51	1.51	0.21		MFLOPS	<b>719.05%</b>
CoreMark		602.44		4.25	Points	<b>14175.06%</b>
Calculate PI as fast as possible	5000	5000	63000	80000	milliseconds	<b>1260.00%</b>
Specs:						
Architecture	32-bit ARM	32-bit ARM	8-bit AVR	8-bit AVR		
CPU Freq	180	180	16	8	MHz	
RAM	256	128	2	8	KB	
FLASH	2048	512	32	256	KB	
I/O Pins	144	64	28	64	Pins	
UART / USART Buses	8	6	1	4	Buses	
I2C Buses	3	3	1	1	Buses	
SPI Buses	6	4	2	5	Buses	
ADC	12-bit, 24-ch	12-bit, 24-ch	10-bit	10-bit	Bits	
Interrupts	All GPIO	All GPIO	2 ?			

Figure 25: Microcontroller Considerations



**Figure 26: Microcontroller Considerations Graph**

# Product Design

# Project Generation

## Meeting Minutes #5

Date: March 9, 2018

### Agenda:

- Select the materials to be used in the product
- Decide on methods to replicate the product as closely as possible using the materials available for the product

### Discussion:

#### 1. Material Selection

- The materials used on the product should be non-ferrous
  - The tesseract has a magnet inside so it would stick to magnetic materials
  - This can disturb the process
    - If the cube sticks to a part of the robot the cube might not be able to be retrieved and placed under the pyramid correctly
  - Aluminum, carbon fibre, Lexand, acrylic and wood are some non-ferrous options
    - Aluminium components can be used for portions of the robot that are under a lot of loading and need to be relatively strong and non-brittle
      - Much lighter than steel and nonferrous
    - Acrylic is a good material to use for parts of the robot that are not under much loading
      - Acrylic can be laser cut easily so manufacturing different parts from it is very simple.
      - Light weight
      - Acrylic is brittle so it cannot be bent into different shapes
      - Glue with small connectors (also cut from acrylic) could be used to make 3 dimensional designs from the laser cut piece
      - **Concerns** – since acrylic is brittle the parts made from it would be easily damaged if they fall or are loaded in a non-favorable angle. Gluing the pieces will also result in mechanisms with low structural integrity as glue might not hold the pieces together securely enough
    - Lexand components would be stronger than those made out of acrylic and they can be bent to make the 3-dimensional mechanisms on the product.
      - Although they cannot be laser cut, they can be machined using plastic specific tools and machines.
      - Lexand is a good alternative for the cube and pyramid retrieval mechanisms.

#### 2. Prototype Design

- The 4 inch VEX wheels available in the lab will be used on prototype instead of the 5" elastic wheels selected for the product.
  - These wheels are harder than the wheels selected for the product so they may have some traction problems when going over the conduits
    - Extra grip might need to be added to make the drive base more closely replicate the product
- VEX 393 motors with integrated encoders are used instead of the Nemo 23 motors selected for the drive base.
  - These motors are not as strong as the motors selected for the product and have different internal gearing so they will have to be geared differently
- Use laser cut acrylic pieces for the cube pick up and the pyramid intake
  - Must note that this will be less durable than the actual mechanisms since acrylic is more brittle than Lexan and aluminium.

**To be completed:**

1. Complete the CAD for the intake and cube pick up parts that need to be laser cut for the prototype
2. Complete the prototype
3. Finalize the Product materials and designs.

## Cube Manipulation and Storage

### Product

While the initial design was able to manipulate the tesseract promisingly, it would need to drop the cube from a high distance, resulting in bounce and roll. This would mean the robot would need to search for the cube again. As such, the rail design concept that was selected was pursued, that stores the cube within the claw, and when the pyramid is retrieved, drops the cube through the ramp.

In the final product all the parts for the rail design would be made from  $\frac{1}{8}$  inch aluminum to ensure that the parts are non-ferrous but durable and strong. The claws would be made out of acrylic since they are not under any great amount of loading. If manufacturing processes allow the claws could be made from 1 layer of  $\frac{3}{8}$  inch acrylic each (See **Figure 27**: Cube Retrieval Design), If testing proves  $\frac{3}{8}$  inch acrylic to be too brittle aluminium can be used instead. The ramp is also not under a great amount of loading however it does have corners that need to be bent upwards. Instead of using 3 pieces of acrylic and gluing them to one another a piece of  $\frac{1}{8}$  inch aluminum can be cut and bent to form the ramp.

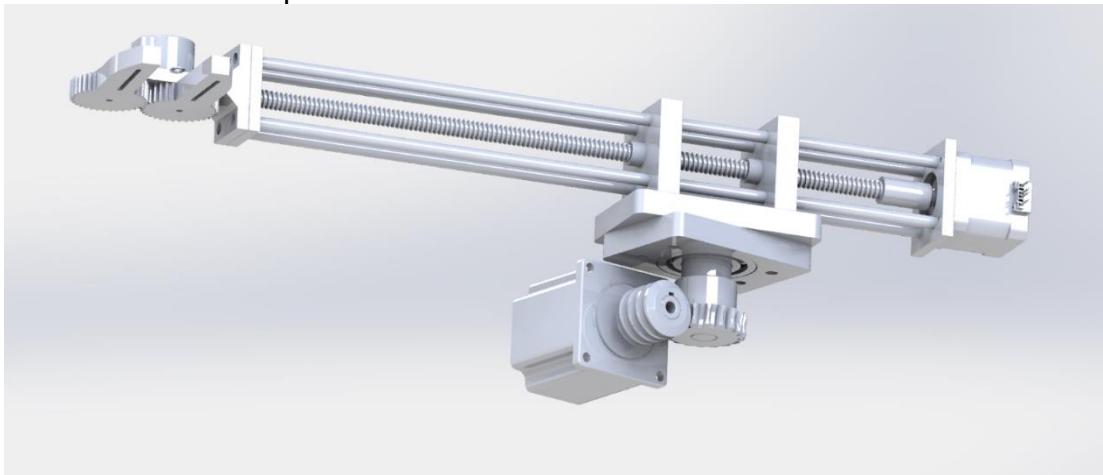


Figure 27: Cube Retrieval Design

### Prototype

Initially, an arm claw prototype was built with VEX parts to refine the claw concept.



Figure 28: Cube VEX Prototype

The claw prototype served as a proof-of-concept for the claw to be a feasible solution for the product, and allowed us to pursue this design. The VEX parts, however, are ferrous and our custom solution was pursued.

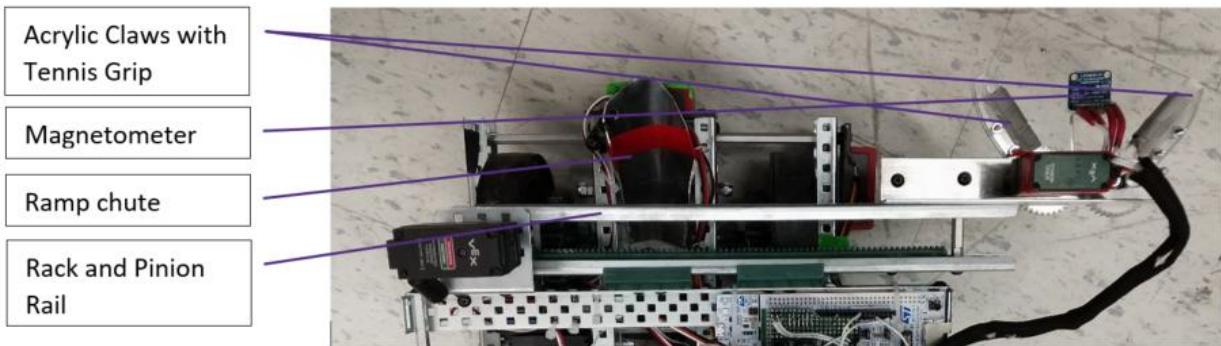


Figure 29: Claw Acrylic Prototype

## Cube Location

The magnetometer that was selected provided better results than the hall effect sensor in the product. It was also used to prototype the programming logic to locate the tesseract. However, it was noticed that it still would only locate the cube within 5 cm. This meant the cube could not be located by the robot just turning and scanning the field from one location.

To ensure that the cube was retrieved every time, The robot would have to find a wall and track the wall as the magnetometer scans the top of the wall for the cube. This process would take longer than a randomized search in some cases but it is more consistent which is our most most important design requirement. The magnetometer will be mounted directly above the claws of the cube retrieval mechanism to ensure that the cube is located just as the claws are approaching it.

## Pyramid Manipulation / Storage and Elevation

### Product

The final product will consist of 2 motors, running in the opposite direction at all times. The motors will be inwards when the pyramid is being retrieved and running outwards after the pyramid is placed on top of the cube. The motors and wheels connected to them will be at the same angle as the pyramid to ensure that maximum contact surface area occurs the wheels and the pyramid.

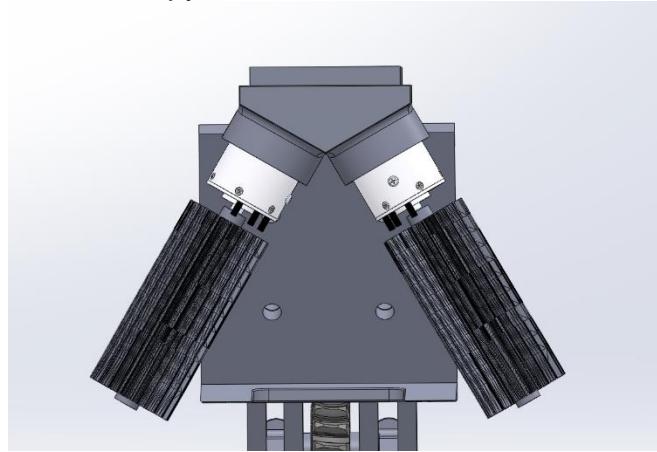


Figure 30: Front View of Pyramid Design

Tests conducted using various prototypes showed that a 10 degree incline for the bottom of the forklift is the most effective angle for intaking the pyramid. These test were conducted by putting the intake at 7 different angles and 10 trials were conducted per angle. As the result the intake is placed at a 10 degree incline.

The bottom of the forklift must be less than 1/32 in thick at the tip to ensure that the bottom of the pyramid does not get caught on it. 1.32 in thick carbon fibre will be used for this bottom piece. Carbon fibre is durable and despite being very thin in this case it will not break easily.

The carbon fibre will be connected to an aluminium piece at the bottom if this piece is too flexible to ensure that the pyramid would not slide off of the intake when being pulled up. The aluminum piece would only be half of the size of the bottom piece to ensure that the tip of the piece is thin enough.

The entire intake system will be lifted to lift the pyramid after the pyramid is retrieved. This is done through the use of a arm with 2 joints attached to a motor. The motors used will be discussed under the engineering analysis section. This arm will lift the pyramid and the intake so that the pyramid can be placed over the cube. This lift system will also keep the intake up when navigating the field to ensure that the intake does not get caught on the conduits.



Figure 31: Pyramid Retrieval Product Render

## Prototype

The roller design was prototyped with VEX parts. Different gripping mechanisms were tested that included VEX intake wheels and LEGO wheels.

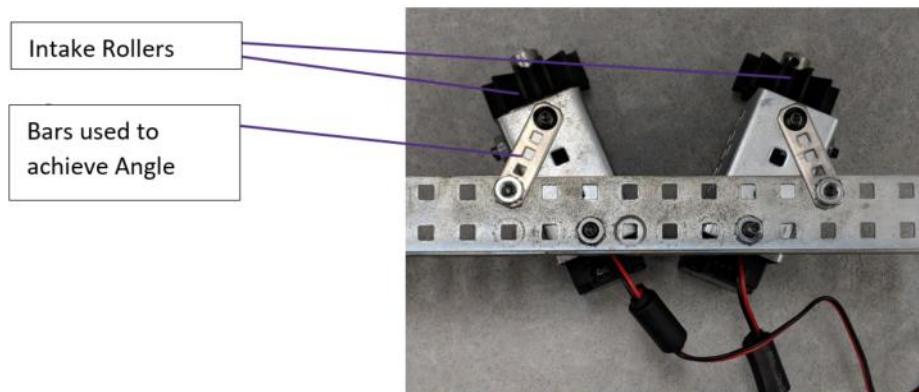


Figure 32: Pyramid Retrieval Prototype Top View

Testing this prototype allowed us to understand that the ramp needed to be thinner than 1/16in to reliably push the pyramid up the ramp. In addition, doubling the intake wheels on each side was found to be a better solution as it increased the contact surface area.

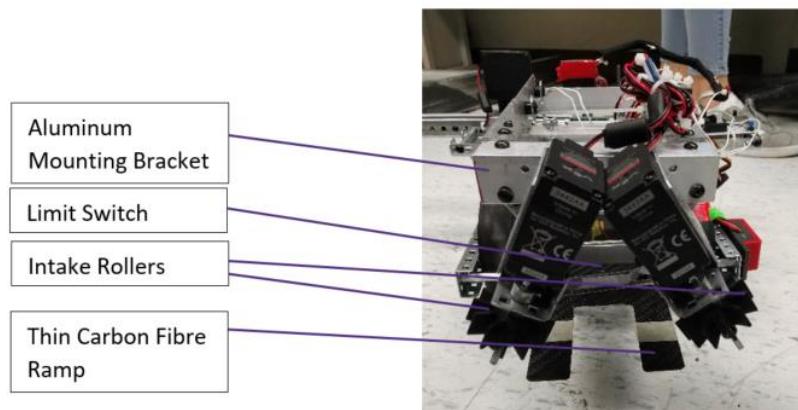


Figure 33: Pyramid Retrieval Prototype Front View

The selected design allows the pyramid to remain on top of the ramp, held in by the rollers. However, the carbon fibre sheet was not strong enough to hold the pyramid on its own when elevated, and so, 1/16in aluminum was mounted halfway underneath the carbon fibre sheet to maintain rigidity. The Aluminum piece mounted also allowed us to give the tip of the carbon fibre a downward angle which eased the intaking processes as the pyramid would not get caught on the tip of the bottom of the forklift.

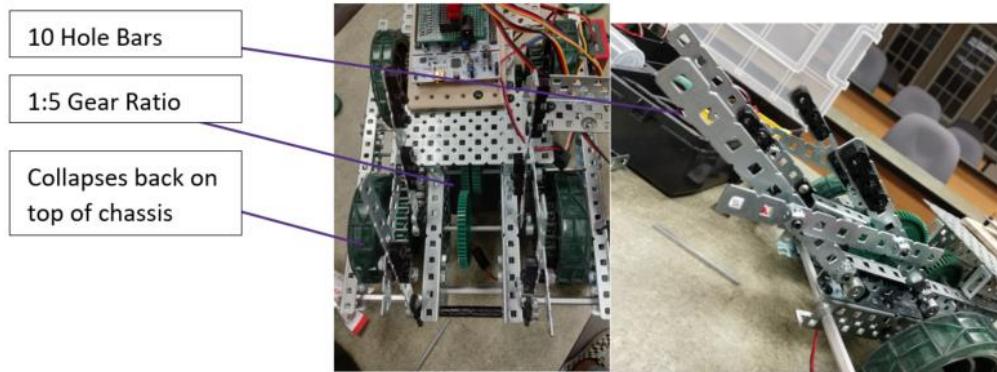


Figure 34: VEX Lift Prototype

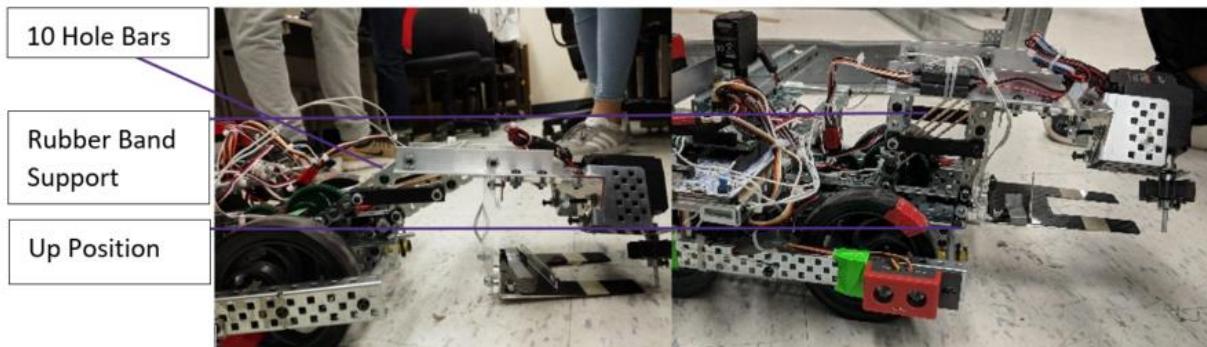
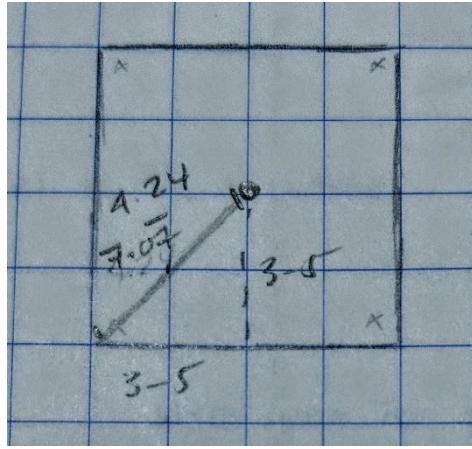


Figure 35: Final Lift Prototype

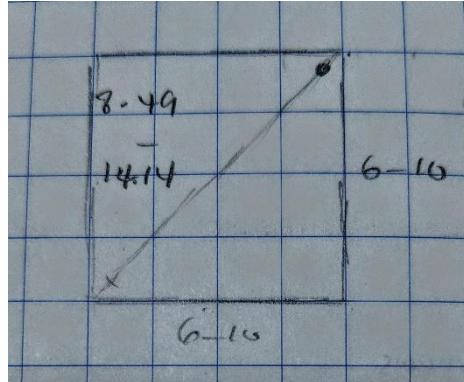
### Pyramid Location

The IR sensors being used have at most a 28 inch range, undervaluing this range just to be safe would give results in a maximum range of approximately 2 ft for the wide (unshielded) IR sensor on the robot. This range is too small for the robot to be able to rotate at one point of the course and pick up the signals from the two pyramids.

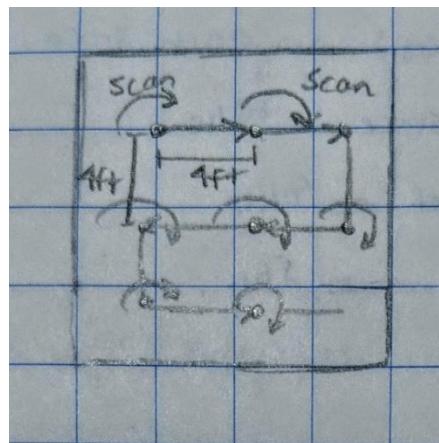
Using pythagorean theorem we see that if the robot is exactly in the middle of the field there is a distance of 4.24 ft to 7.07 ft between the robot and the farthest corner of the field for a field ranging from 6ft to 10ft respectively.



Assuming the robot is at one corner of the field the largest distance increases to a range between 8.49 ft to 14.14 ft.



Since the distance between the pyramid and the robot can be more than 2 ft the robot must navigate through a path and search for the pyramid. This robot must rotate and scan the field every 4 ft. Although a systematic path being taken might not always be the fastest depending on where the pyramids are located, a systematic approach allows us to always, eventually locate the pyramid. A randomized approach might result in the pyramid being missed. The systematic path drawn below is suggested as the path to be taken.



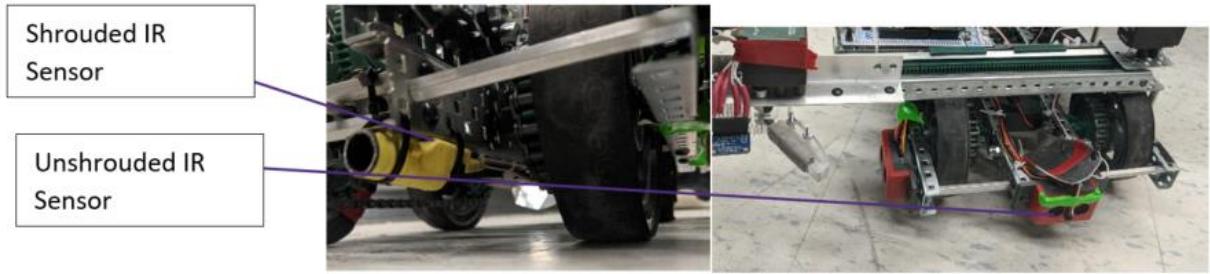


Figure 36: IR Sensor Mounts

## Drive Design

### Product

The base of the robot will be built out of aluminium to ensure that it is strong, light and non-ferrous. 4" elastic wheels from McMaster-Carr are being used as the 4 wheels of the robot. These wheels are very soft and have very good traction so they would not slip over the conduits. The two wheels on each side will be controlled by 1 motor. chain wil be used to connect the two motors. The length of the chain be modified in the wheels need to be farther apart or closer to one another. Finally, the 4 wheel design offers stability for the robot as it navigates over conduits resulting in a smooth navigation.



Figure 37: Drive Product Render

## Prototype

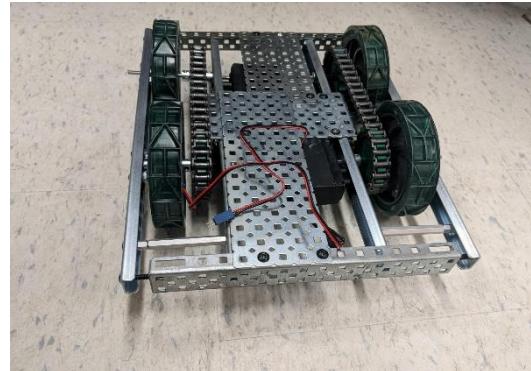


Figure 38: Drive VEX Prototype

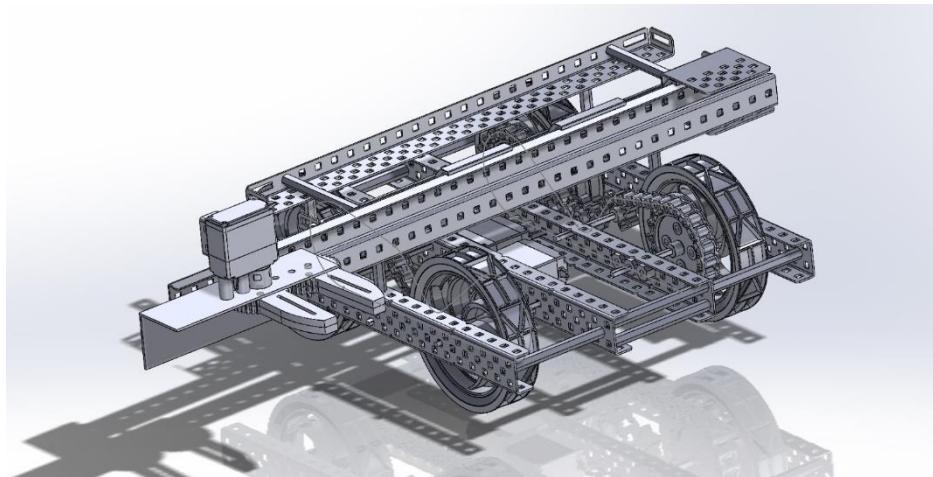


Figure 39: Drive CAD Assembly

The prototype was built using the materials available in the lab. 4" VEX wheels were used. The wheels were quite hard and as the result they slipped over the conduits. Tennis racket grips were added to the wheels to increase their traction so they could replicate the product more closely. VEX 393 motors were used instead of the motors selected for the product. Since these motors were weaker than those on the product additional gearing was required to replicate the product. This addition made the prototype larger than the actual product.

## Full Product Design

As a whole, the Tesseract Robot has the ability to locate, retrieve and store both the tesseract and the pyramid. The rail claw provides degrees of freedom in both X and Y, and has a reliable nonferrous claw system to grip the tesseract. The pyramid intake with two large rollers are designed in the shape of the pyramid to maneuver it with ease. It is coupled with a strong 4-bar lift system to elevate the pyramid to finally place it above the tesseract. The Tesseract Robot is designed with a powerful drive system that is able to conveniently drive its low weight of 6.7kg.

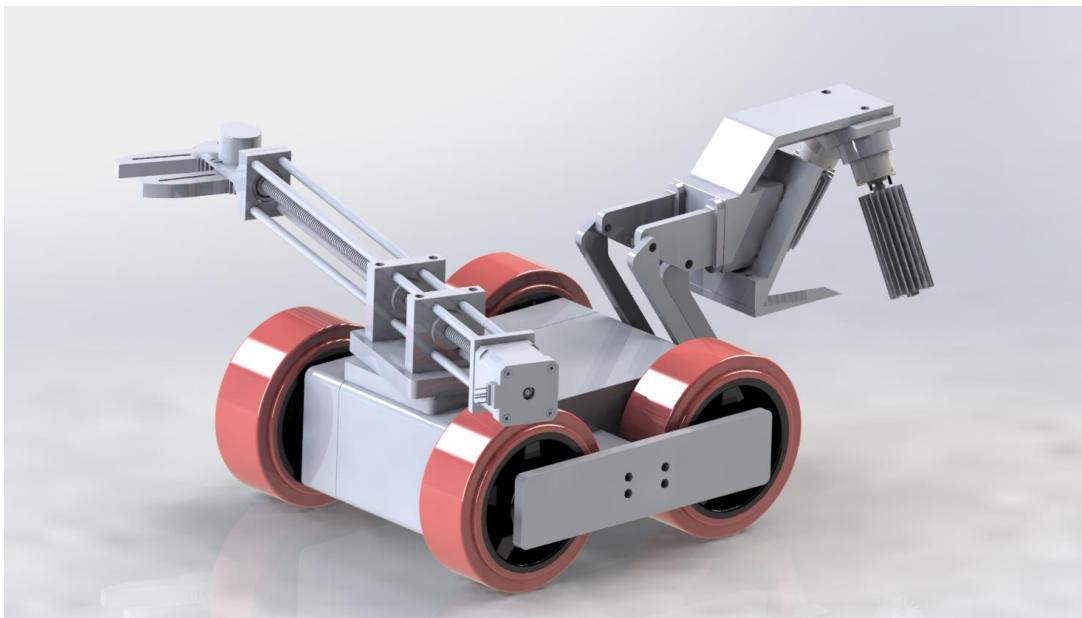


Figure 40: Full Product Render

## Full Prototype Design

The prototype was developed with the product as a model with VEX, laser cut acrylic, aluminum, and carbon fibre parts.

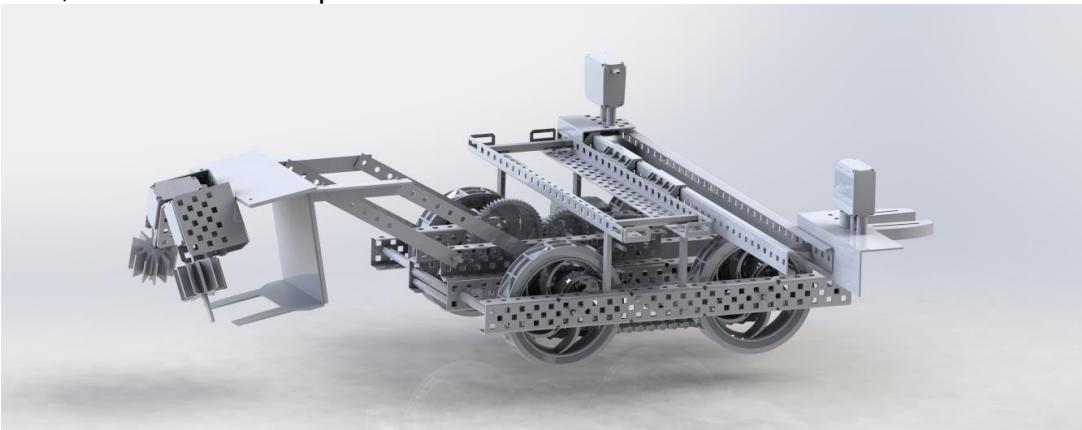


Figure 41: Full Prototype Render

## MSE2202B Prototype Specification Document

- Electronics, Sensors and Actuators
  - Main MCU
    - Nucleo-64, STM32 - F446RE
    - 180 MHz ARM Cortex-M4
    - 512 kb flash, 128 kb RAM
  - Sensors
    - Magnetic
      - ST LSM303
        - 1x Located on claw for cube detection
      - NXP MAG3110
        - 1x Located on rear chassis for cube detection for accurate pyramid placement
    - Distance
      - VEX Ultrasonic Rangefinder
        - 2x located on left side for parallel wall following
        - 1x located on front side for turning
    - IR
      - TSOP32338
        - 1x located on rear for Pyramid detection
        - 1x located on rear for Pyramid detection (with focusing shroud)
    - DC Motors, All controlled via VEX 29 Motor Controller
      - VEX DC motor
        - 1x located on claw linear slice
        - 3x located on pyramid lift
      - Chinese 12V DC motor with gearbox
        - 1x for left drive
        - 1x for right drive
    - Servo Motors
      - VEX Servo
        - 1x located on claw
    - Wireless Communications
      - ESP8266 WiFi Module
        - 1x for user to issue start and stop commands via a phone or computer
    - Misc.
      - Limit Switches
        - 2x located on claw linear slide
          - Connected as interrupts to GPIO ports
        - 2x located on pyramid lift
          - Connected as interrupts to GPIO ports

Figure 42: Prototype Specification Document

## Electronic Design

### Discussion

A third-party microcontroller was used due to increased I/O and computational capabilities when compared to an ATmega238p. STMicroelectronics Nucleo development boards were selected for use on the prototype. Nucleo development boards are an Arduino alternative with superior capabilities when compared to AVR microcontrollers. Two appropriate boards were obtained for this project: a Nucleo-144 with an STM32-F429ZI MCU, and a Nucleo-64 with an STM32-F446RE MCU. STM32 is a line of industrial grade ARM Cortex-M based microcontrollers offered by STMicroelectronics. A summary of the specifications and a STM32 block diagram for these boards is listed below in Figure 1.

The main benefits for using this alternative microcontroller include:

- greater number of GPIO ports
- greater number of IRQ ports
- greater computational capabilities
- easier interfacing with I2C peripherals (does not require logic level shifter)
- multiple hardware UART buses
- theoretical higher polling rate for sensors

A Pinmux is provided below in Figure 2 for the Nucleo-64 which outlines how every peripheral is connected to the STM32 controller. Two Block Diagrams of the electronic components are listed below. Figure 3A is the initial planned peripheral layout. Figure 3B is the peripheral layout which was used on the prototype.

A Daughter board was developed to provide easy VEX plug-and-play capability to the Nucleo Controller Boards. See the schematic below. The pins on the final revision are separated into low and high-power tolerances. High power pins provide a GPIO pin with pwm capability for motors. These high-power pins are routed through a voltage regulator which provides high current directly from the battery. The low power pins provide GPIO communications for various basic sensors and interrupts. 5V is provided from the host board. Additional pins included UART 7, UART 8, and I2C1 for various peripherals. All pins and sockets are 0.1" pitch which complies with the Arduino UNO header standard, and the VEX sensor connectors.

A Three-axis magnetometer was solely used to detect the tesseract while on the wall. This sensor interfaces with the controller through an I2C bus. The firmware required to interface with this sensor sends initialization, resolution, and read

commands over the I2C1 bus to the sensor, then listens on the bus for the response. Three 16-bit values are returned from the sensor; one for each axis. The software simply calculates the Euler magnitude of the magnetic field via the three readings. If this magnitude is greater than a threshold, then the robot will intake the tesseract.

The two infrared sensors are connected to hardware UART buses. This allows the STM32 to receive UART Rx interrupts to identify pyramids, however due to a lack of software development time, this was not able to be implemented. The prototype in its demonstration state was polling the UART 7 and UART 8 buses in order to detect the pyramid. Using the hardware serial buses available on the STM32 eliminates the need to use a cumbersome software serial library to interface with the infrared sensors.

Limit Switches on both intake mechanisms are connected through interrupts to the STM32.

The Nucleo-64 was planned to be used to drive the prototype. 48 hours before the showcase was commenced, the Nucleo-64 board was rendered unusable due to neglect and was replaced with the Nucleo-144.

## **NUCLEO-144**

STM32F429ZIT6 in LQFP144 package  
ARM®32-bit Cortex®-M4 CPU with FPU  
180 MHz max CPU frequency  
VDD from 1.8 V to 3.6 V  
2048 KB Flash  
256+4 KB SRAM, including 64 KB of CCM (core coupled memory) data RAM  
GPIOs (114) with external interrupt capability  
16-stream DMA controller with FIFOs and burst support  
12-bit ADCs with 24 channels (3)  
12-bit DAC channels (2)  
USART/UART (4)  
I2C (3)  
SPI (6)  
Advanced-control Timer (2)  
General Purpose Timers (10)  
Watchdog Timers (2)  
CAN 2.0B active (2)  
SAI  
SDIO  
Random Generator (TRNG for HW entropy)  
USB 2.0 OTG HS  
USB 2.0 OTG FS  
Camera interface

Ethernet  
LCD-TFT

## **NUCLEO-64**

STM32F446RET6 in LQFP64 package

ARM®32-bit Cortex®-M4 CPU with FPU

Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory

180 MHz max CPU frequency

VDD from 1.7 V to 3.6 V

512 KB Flash

128 KB SRAM System

4 KB SRAM Backup

Timers General Purpose (10)

Timers Advanced-Control (2)

Timers Basic (2)

SPI (4)

I2S (2)

USART (4)

UART (2)

USB OTG Full Speed and High Speed

CAN (2)

SAI (2)

SPDIF-Rx (1)

HDMI-CEC (1)

Quad SPI (1)

Camera Interface

GPIO (50) with external interrupt capability

12-bit ADC (3) with 16 channels

12-bit DAC with 2 channels

Figure 3. STM32F446xC/E block diagram

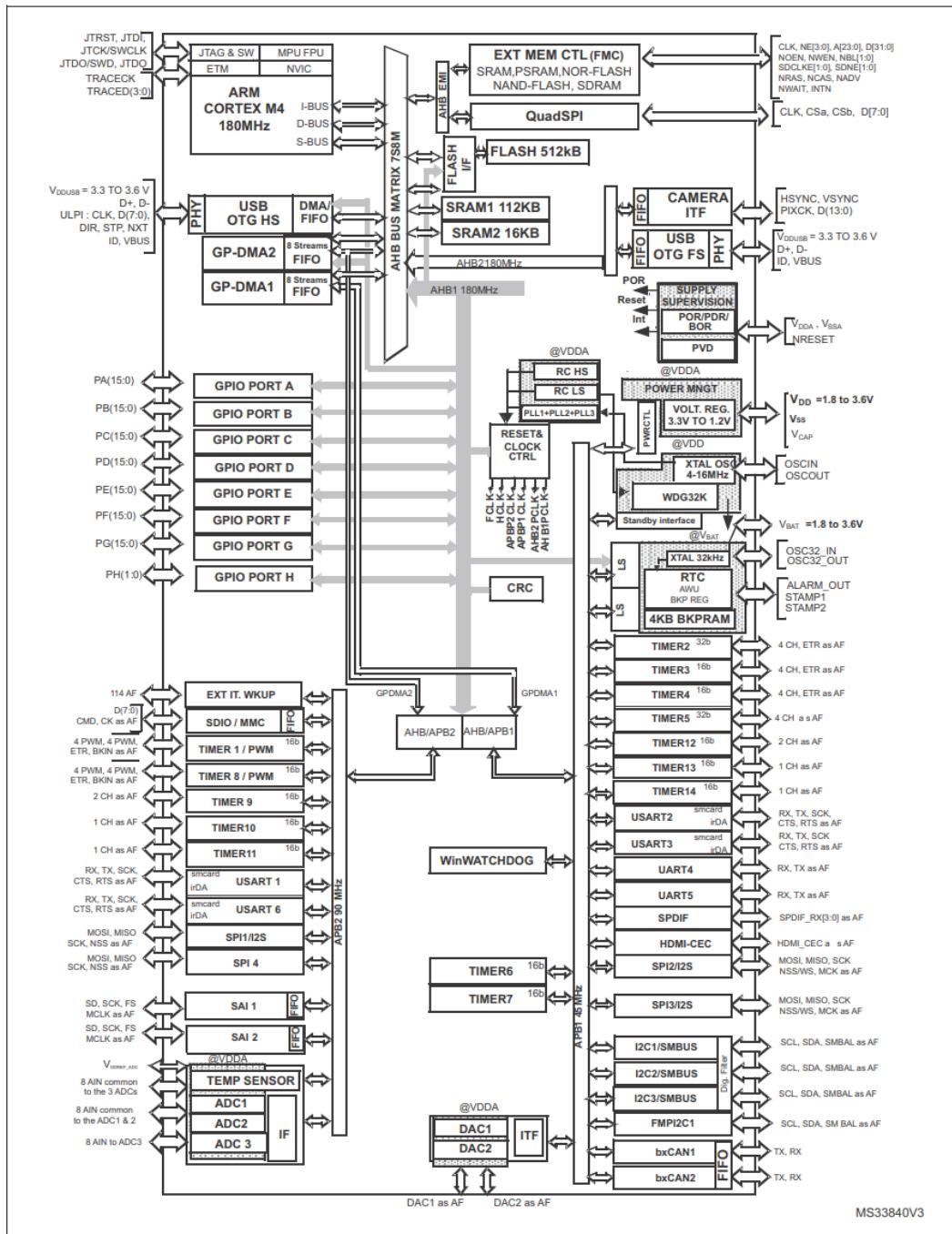


Figure 43: STM Block Diagram

## Daughter Board

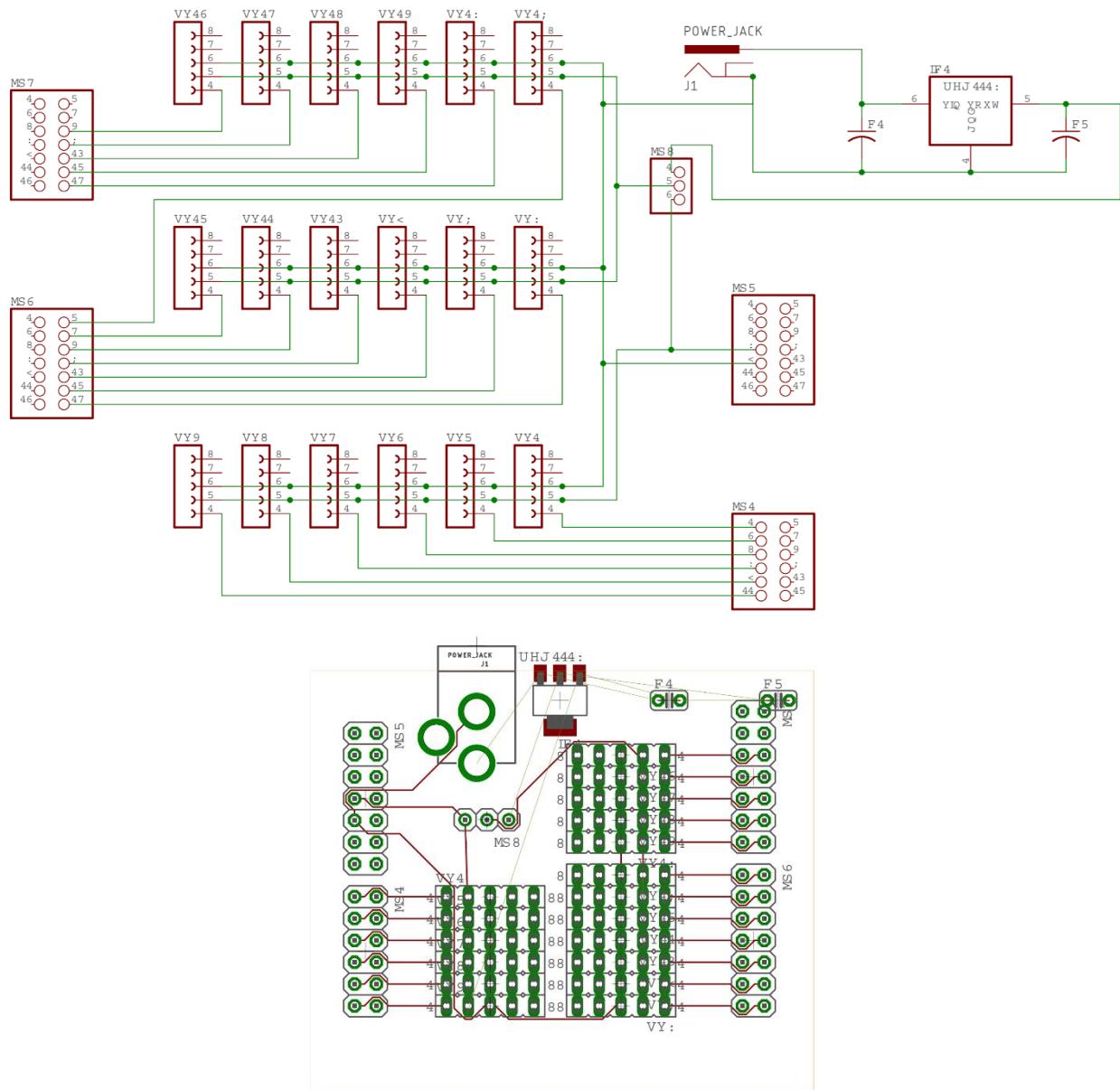


Figure 44: Daughter Board Layout

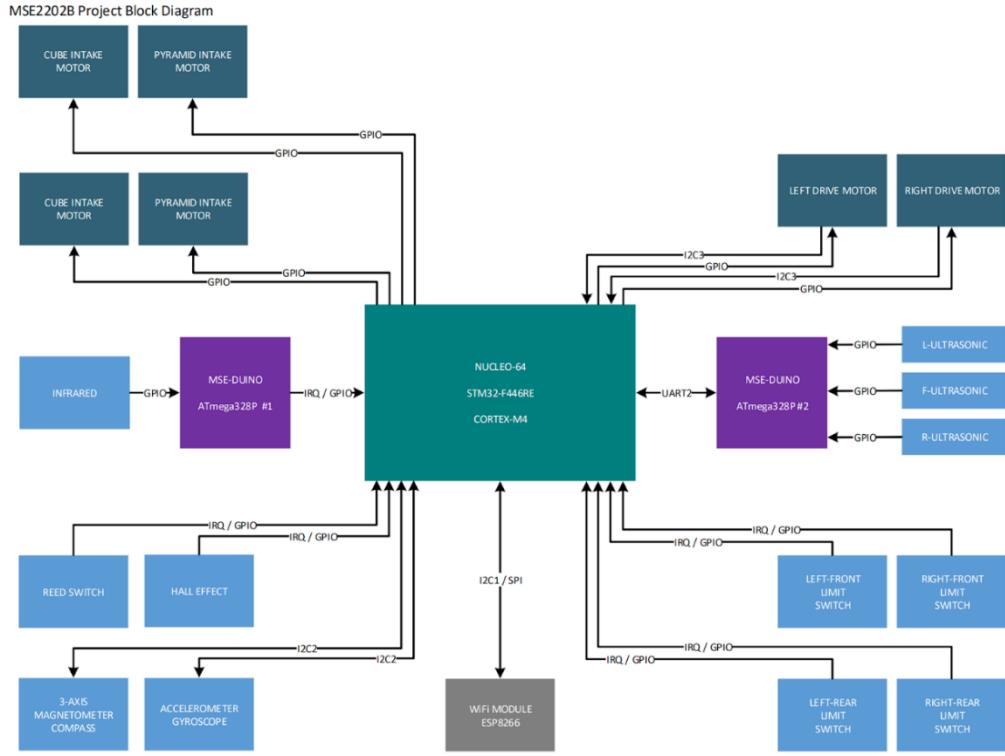


Figure 45: Planned Block Diagram

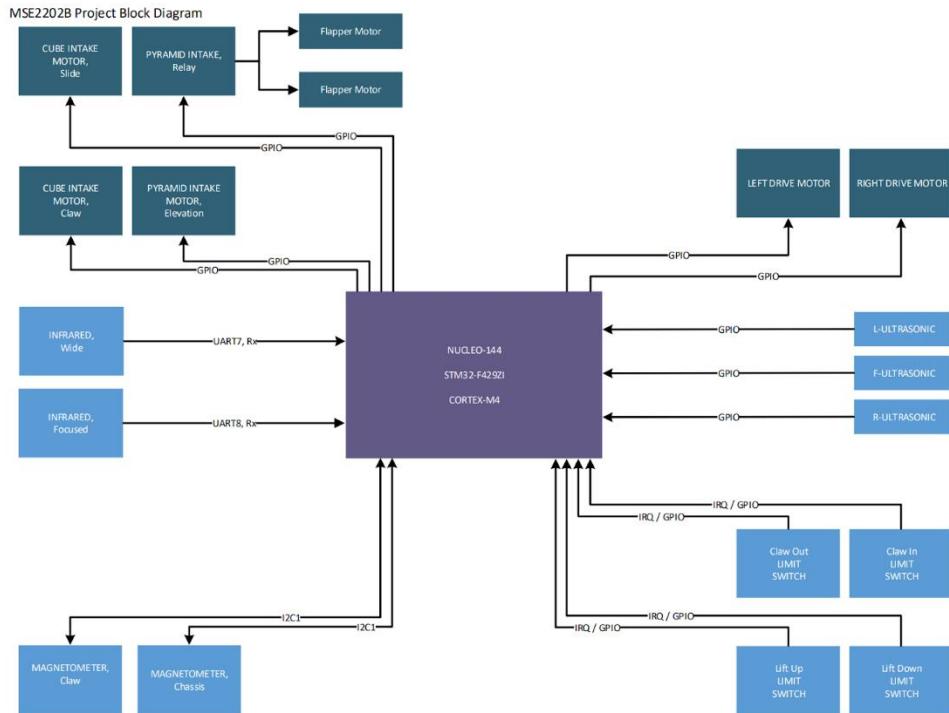


Figure 46: Block Diagram Used

Position Name	Daughter Board Low Power		Daughter Board High Power		Attached Device		I/O Structure	Voltage	Signal	Label	Alternate Functions		AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	ADC	DAC	
	STM	Header	Arduino UNType	Power	Other						AF0	AF0																
2 PC13	CN7-1/2	IO					I/T	5 V	GPIN_ExT13_B [Blue Pin/Blue]																			
3 PC14	CN7-1/3	IO					I/T	5 V	RCC_OSC2_IN																			
4 PC15	CN7-1/4	IO					I/T	5 V	RCC_OSC2_OUT																			
5 P10	CN7-1/6	IO					I/T	5 V	RCC_OSC_IN																			
6 P11	CN7-1/16	IO					I/T	5 V	RCC_OSC_OUT																			
7 Nrst	CN7-2/7	Reset																										
8 PC0	CN7-2/10	A0					I/U, I/U, I/U, OUT	5 V																				
9 PC1	CN7-2/16	A0					I/U, I/U, I/U, IN	5 V																				
10 PC2	CN7-18	IO					I/T	5 V																				
11 PC3	CN7-19	IO					I/T	5 V																				
14 PA0	A0		F				F, UTRX, OUT	5 V																				
15 PA1	CN7-2/4	A1					F, ULTRA, OUT	5 V																				
16 PA2	CN7-2/15	D1					Default Serial	5 V																				
17 PA3	CN7-1/19	D0					Default Serial	5 V																				
20 PA4	CN7-2/16	A0					I/E, I/U, I/U, IN	5 V																				
21 PA5	CN7-1/6	D0					GPIO Output	5 V																				
22 PA6	CN7-1/7	D1					LIFT MOTOR	5 V																				
23 PA7	E11	IO					DRIVE MOTOR, RIGHT	5 V																				
24 PA8	CN7-1/8	D1																										
25 PC5	CN7-2/17	A0																										
26 PS0	CN7-2/3	A3					I/F, ULTRA, OUT	5 V																				
27 PS1	CN7-2/12	IO					LIMIT_SIN_2, LEFT DOWN	5 V																				
28 PS2	CN7-2/11	IO					LIMIT_SIN_2, LEFT UP	5 V																				
29 PB0	CN7-1/3	B6					LIMIT_3, RX IS SHROUDED REAR, FT	5 V																				
33 PB12	CN7-1/2	IO																										
34 PB13	CN7-1/26	IO																										
35 PB14	CN7-2/15	IO																										
36 PB15	CN7-2/14	IO																										
37 PG6	CN7-2/25	IO																										
38 PC1	CN7-1/10	B9																										
40 PC8	CN7-1/21	IO																										
41 PA8	CN7-1/12	E7																										
42 PA9	CN7-1/11	E3					CLAW SLICE SERVO	5 V																				
43 PA10	CN7-1/17	D2					LIMIT_1, RX IS INSPIRED FRONT	5 V																				
44 PA11	CN7-2/7	IO																										
45 PA12	CN7-2/6	IO																										
46 PA13	CN7-1/7	IO																										
49 PA14	CN7-1/8	IO																										
50 PA15	CN7-1/9	IO																										
51 PC0	CN7-2/1	IO																										
52 PC11	CN7-2/1	IO																										
53 PC12	CN7-2/2	IO																										
54 PC2	CN7-2/2	IO					CLAW GRIP SERVO	5 V																				
55 PB3	CN7-1/6	D9																										
56 PB4	CN7-1/4	D5																										
57 PB5	CN7-1/5	D4																										
58 PB6	CN7-1/9	D10					DRIVE MOTOR, LEFT	5 V																				
59 PB7	CN7-1/11	IO																										
60 B000	CN7-1/4	Boot																										
61 PB8	CN7-1/2	0B1																										
62 PB9	CN7-1/3	0B4																										

Figure 47: STM32 Pinmuxing

GROUND	I/O						
POWER	OTHER						
Attached Dev	CN7	Attached Dev		Attached Dev	CN10	Attached Dev	
	CN7-1-1      CN7-2-1				CN10-1-1      CN10-2-1		
	CN7-1-2      CN7-2-2				CN10-1-2      CN10-2-2		
	CN7-1-3      CN7-2-3				CN10-1-3      CN10-2-3		
	CN7-1-4      CN7-2-4				#N/A      CN10-1-4      CN10-2-4		
	CN7-1-5      CN7-2-5				CN10-1-5      CN10-2-5		
	CN7-1-6      CN7-2-6			PYR INTAKE MOTOR	CN10-1-6      CN10-2-6		
	CN7-1-7      CN7-2-7			LIFT MOTOR	CN10-1-7      CN10-2-7		
	CN7-1-8      CN7-2-8			DRIVE MOTOR RIGHT	CN10-1-8      CN10-2-8		
	CN7-1-9      CN7-2-9			DRIVE MOTOR LEFT	CN10-1-9      CN10-2-9		
	CN7-1-10      CN7-2-10			CLAW SLIDE SERVO	CN10-1-10      CN10-2-10		
	CN7-1-11      CN7-2-11			UART3, RX IR SHROUDED REAR	CN10-1-11      CN10-2-11	LIMIT_SW_1 LIFT UP	
	CN7-1-12      CN7-2-12				CN10-1-12      CN10-2-12	LIMIT_SW_2 LIFT DOWN	
	CN7-1-13      CN7-2-13				CN10-1-13      CN10-2-13	LIMIT_SW_3 CHECK PYR	
	CN7-1-14      CN7-2-14	F_ULTRA_IN			CN10-1-14      CN10-2-14		
	CN7-1-15      CN7-2-15	F_ULTRA_OUT			CN10-1-15      CN10-2-15		
	CN7-1-16      CN7-2-16	LF_ULTRA_IN		CLAW GRIP SERVO	CN10-1-16      CN10-2-16		
	CN7-1-17      CN7-2-17	LF_ULTRA_OUT		UART1, RX IR UNSHROUDED FRONT	CN10-1-17      CN10-2-17		
#N/A	CN7-1-18      CN7-2-18	LR_ULTRA_IN			Default Serial	CN10-1-18      CN10-2-18	
	CN7-1-19      CN7-2-19	LR_ULTRA_OUT			Default Serial	CN10-1-19      CN10-2-19	

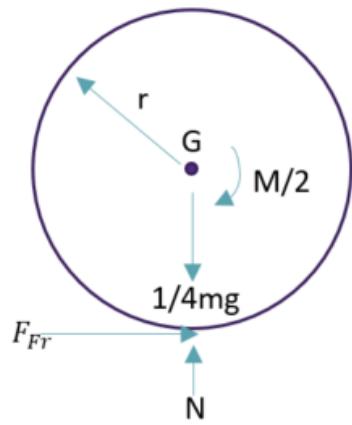
Figure 48: STM32 Custom Headers

# Engineering Analysis

## Drive

To allow the product to successfully navigate the field and maneuver conduits with no stalling, yet fast enough speeds, a strong motor with a high torque was required. As such, the NEMA 23 2250 Position-Control DC Motor from McMaster Carr was selected for its high stall torque of 23in-oz and peak power of 320W.

To verify the motor would satisfy torque requirements, a torque calculation was done on each wheel.



Assume even weight and torque distribution between wheels and a nominal  $u_s$  of 0.4

$$\sum F_G = N - \frac{1}{4}mg = 0$$

$$N = \frac{1}{4}mg \quad F = u_s N = \frac{1}{4}mgu_s$$

$$\sum M_G = F_{Fr}r - \frac{M}{2} = 0$$

$$M = \frac{1}{2}mgu_s r$$

From SolidWorks mass and COG Analysis:

$$m = 6.67\text{kg} \quad r = 0.1245\text{m}$$

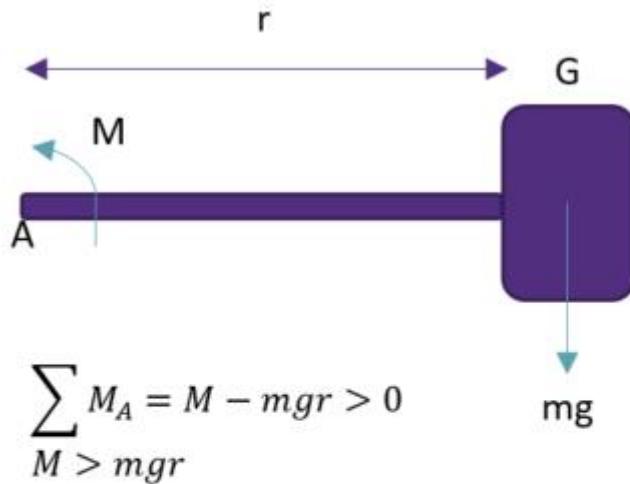
For a FOS of 2 and a 30 gearbox/chain reduction, motor stall torque should be greater than: **0.109 Nm**.

Figure 49: Drive Torque Calculations

As the NEMA motors provide 0.162Nm in stall torque, these motors satisfy the torque requirement with a factor of safety of 2.

## Lift and Claw Arm

To develop a reliable lift for the pyramid that can consistently pickup and drop the pyramid, a high torque with a large margin of safety is required. A torque analysis was done to determine the largest torque requirement from the pyramid lift.



From SolidWorks mass and COG Analysis:

$$m = 0.48kg \quad r = 0.33m$$

For a FOS of 2 and the 50:1 gearbox reduction,  
motor stall torque should be greater than **0.062 Nm**.

Figure 50: Lift Torque Calculations

Using the NEMO 14 with gearbox from McMaster Carr, it is seen that the torque requirement is met with a factor of safety of 2, as the motor has a stall torque of 0.07Nm. The same motor was selected for the Claw Arm due to similar torque requirements.

## Pyramid Intake

The pyramid intake uses two rollers moving in opposite directions spaced out enough to let a pyramid through. Due to this nature, size is of utmost importance, along with a high RPM and a reasonable torque. The Compact Round-Face DC Gearmotor was selected for its small size of 3.75x1.25x1.25in and high speed of 202 rpm. One of these on both sides will prevent any torque issues.

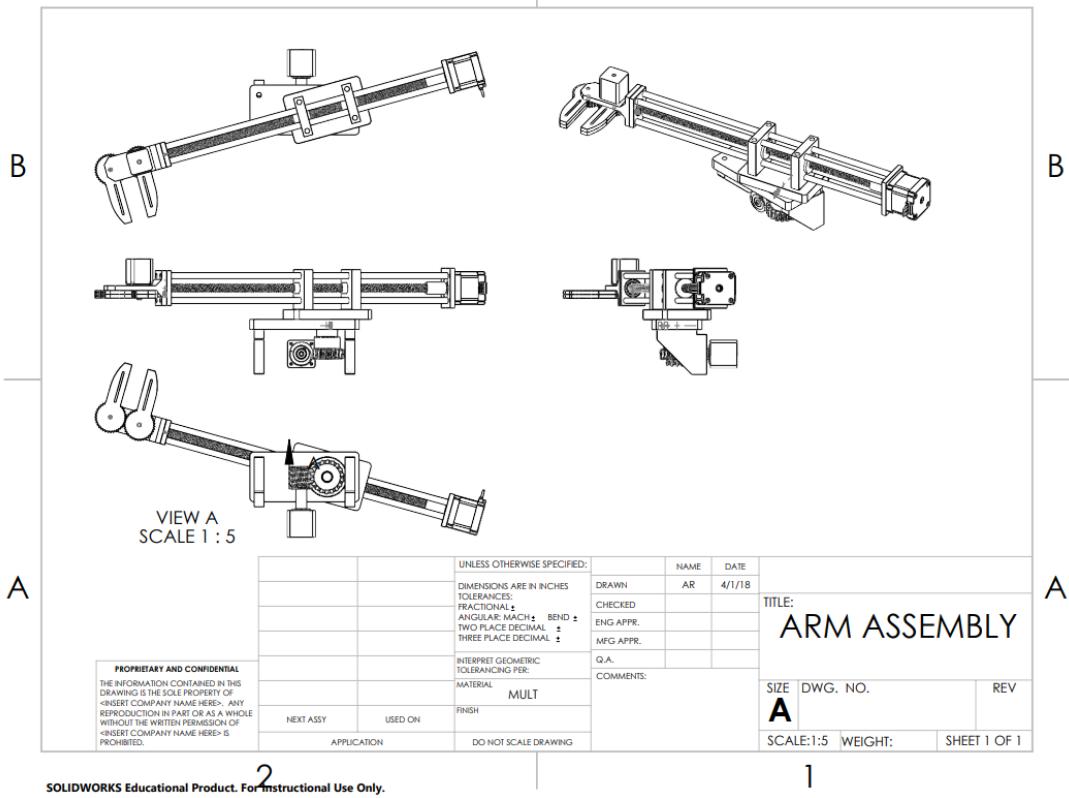
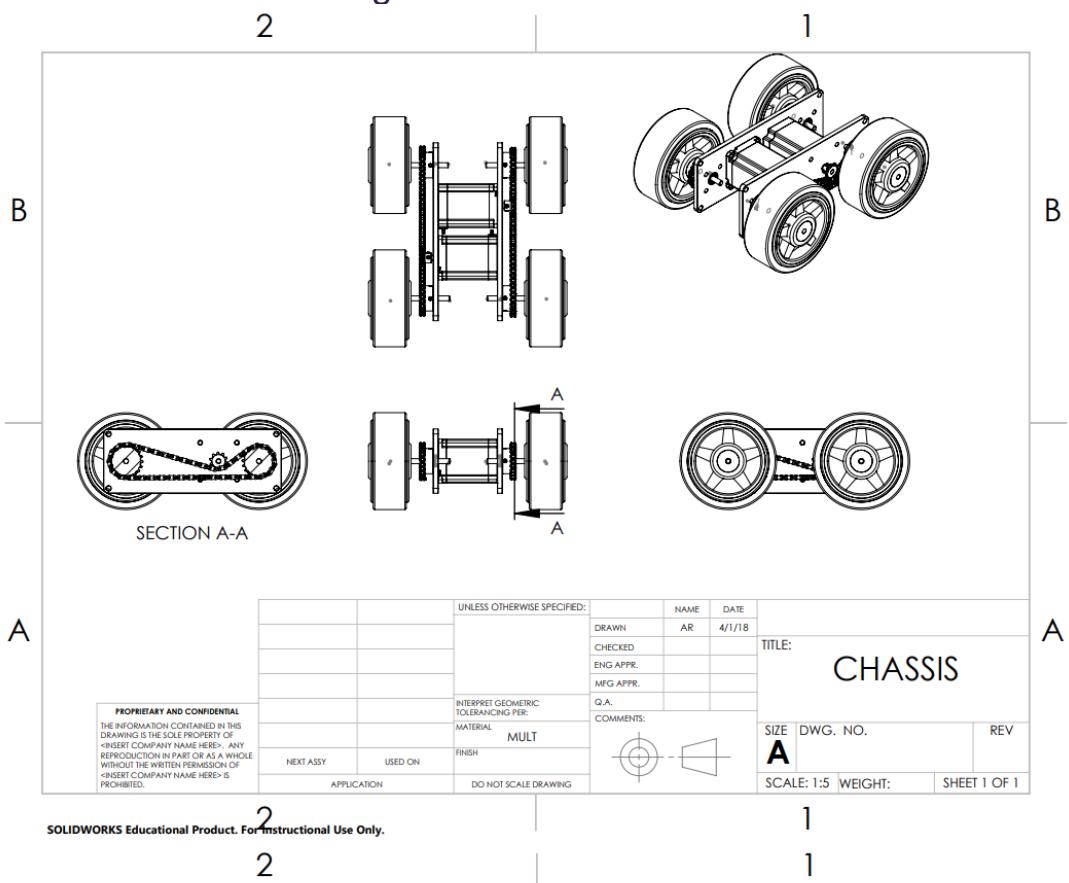
## Detailed Documentation

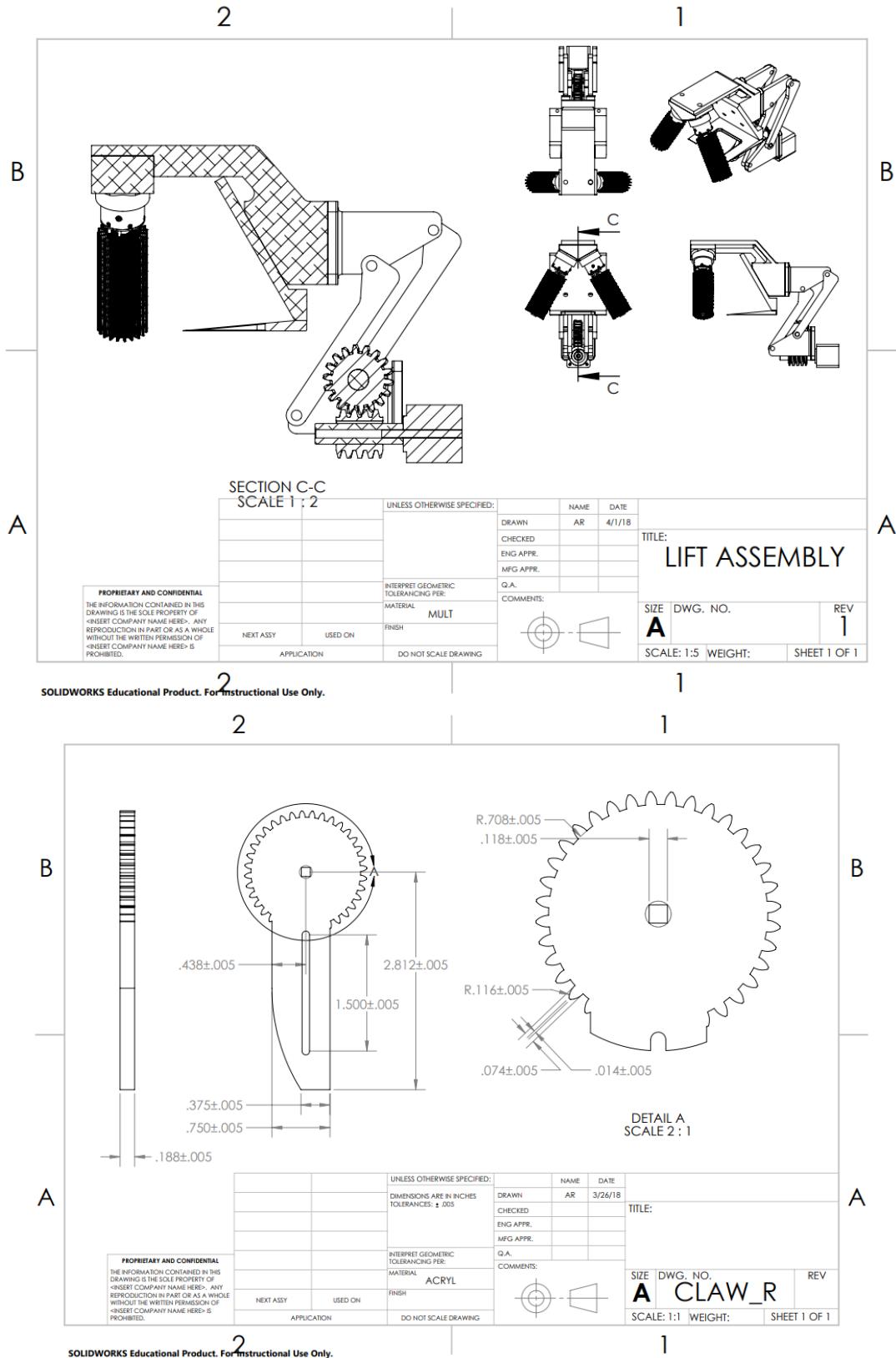
### Product BOM

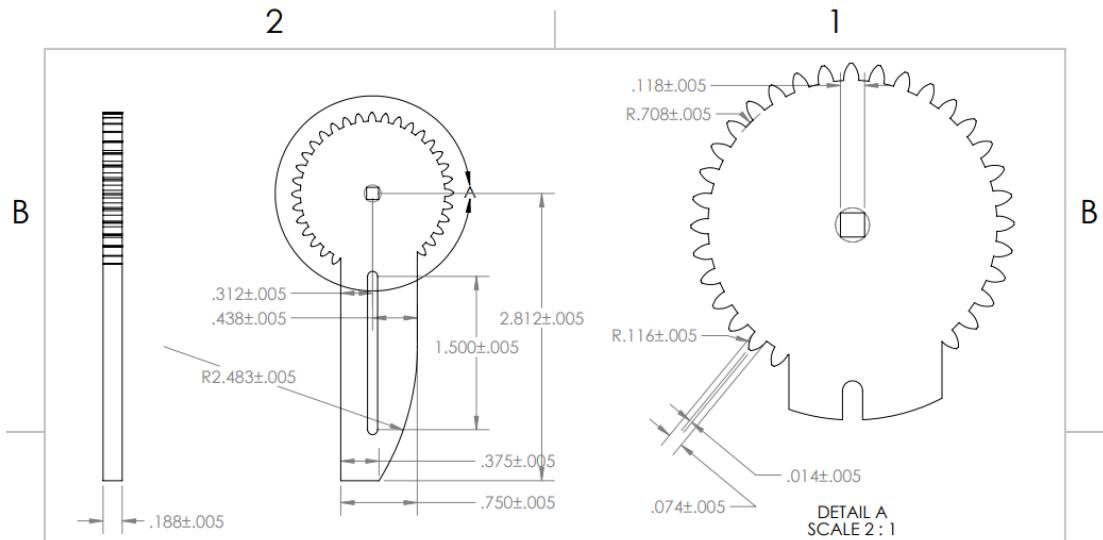
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	ChassisSidePlate		2
2	DriveStepper		2
3	60355K503		4
4	1327K113		4
5	2737T131		4
6	6261K281		78
7	2737T2		2
8	Wheel_Bushing		4
9	CasterWheel_5in.ipt		4
10	ChassisSideCover		2
11	SideSensorEntension		2
12	SideSensorMount		2
13	ChassisMotorHousing		1
14	ClawArmBearingMoun †		1
15	60355K509		1
16	ArmRotate		1
17	RackStand		2
18	93410A912		1
19	95120A592		2
20	Rail		4
21	nema_17_motor		1
22	ClawWormCoupler		1
23	ClawRailMotorBracket		1
24	60355K504		1
25	ClawBracket		1
26	ClawMount		1
27	CLAW_L		2
28	CLAW_R		2
29	Step Motor 28BYJ-48		1
30	57545K511_SPEED- REDUCING WORM GEAR FOR 90 DEG TRANSFER		1

31	57545K527_SPEED-REDUCING WORM GEAR FOR 90 DEG TRANSFER	2
32	RotationBushing	2
33	6627T38	2
34	CubeArmMountBracke †	2
35	LiftLinkageBaseMount	4
36	PyrlIntakeAngledMount	1
37	PyrlIntakeBacking	1
38	PyrlIntakeForks	1
39	LiftLinkage	4
40	PyrlIntakeFlapperMotor	2
41	PyrlIntakeMotorMount	1
42	PyrlIntakeRoller	2
43	57545K511_SPEED-REDUCING WORM GEAR FOR 90 DEG TRANSFER-cutoff	1
44	PyrlIntakeLinkageRod	1

# Product Technical Drawings



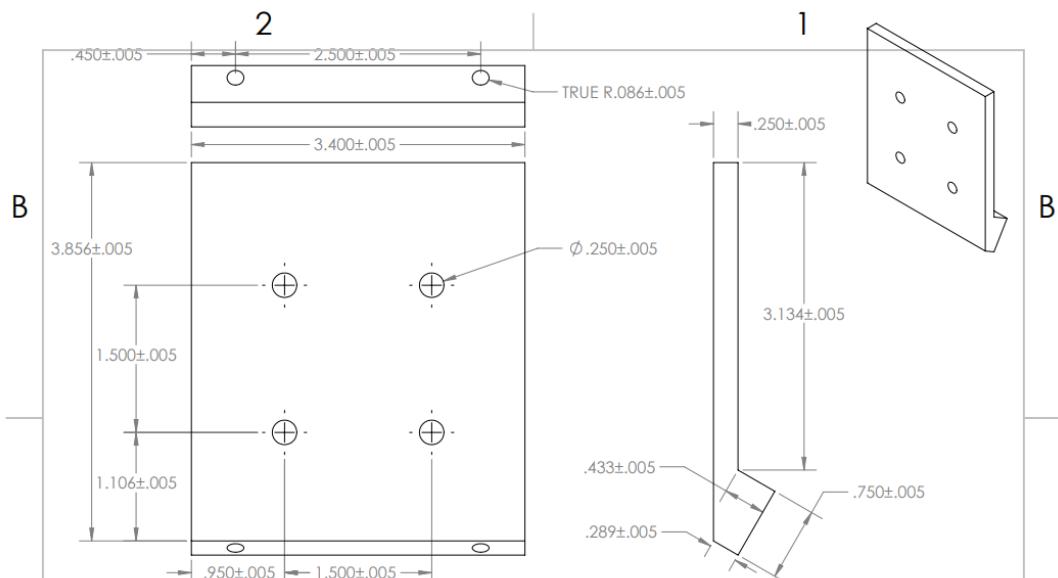




		UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: ± .005					
		DRAWN	NAME	DATE			
		AR		3/26/18			
		CHECKED					
		ENG APPR.					
		MFG APPR.					
		Q.A.					
		COMMENTS:					
		SIZE	DWG. NO.	REV			
A		<b>A</b>	<b>CLAW_L</b>				
		SCALE: 1:1 WEIGHT: SHEET 1 OF 1					

**PROPRIETARY AND CONFIDENTIAL**  
THE INFORMATION CONTAINED IN THIS DRAWING IS THE PROPERTY OF  
<INSERT COMPANY NAME HERE>. ANY  
REPRODUCTION IN PART OR AS A WHOLE  
WITHOUT THE WRITTEN PERMISSION OF  
<INSERT COMPANY NAME HERE> IS  
PROHIBITED.

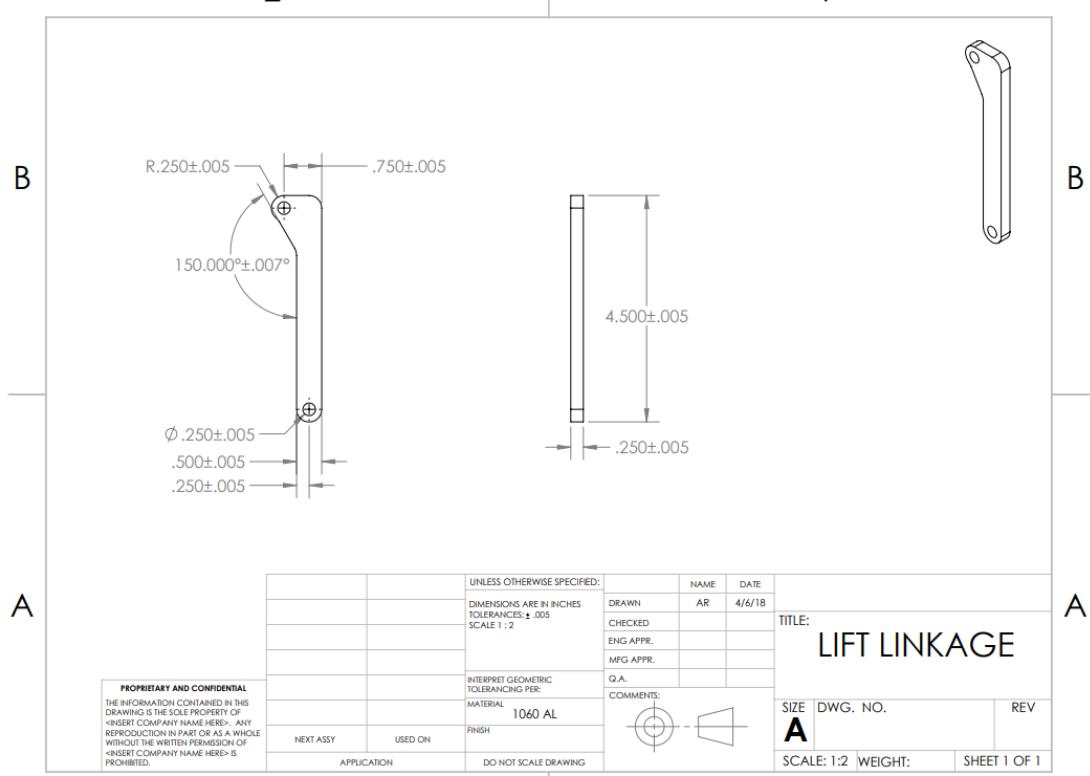
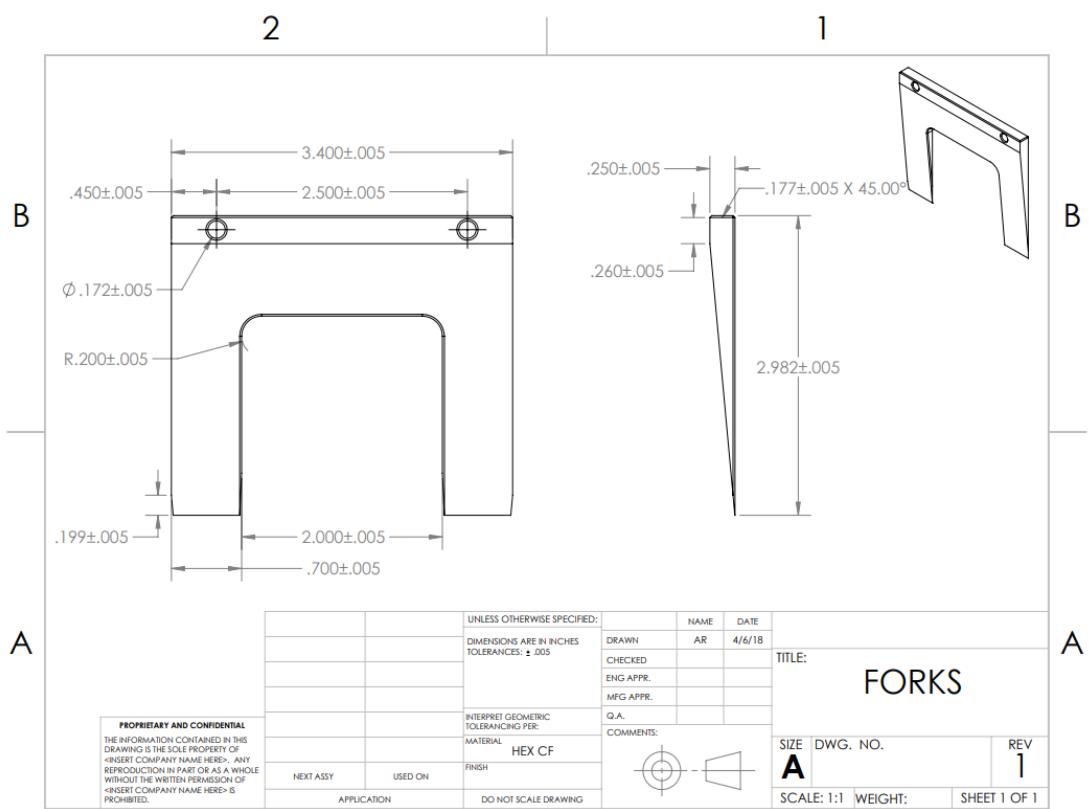
SOLIDWORKS Educational Product. For Instructional Use Only.

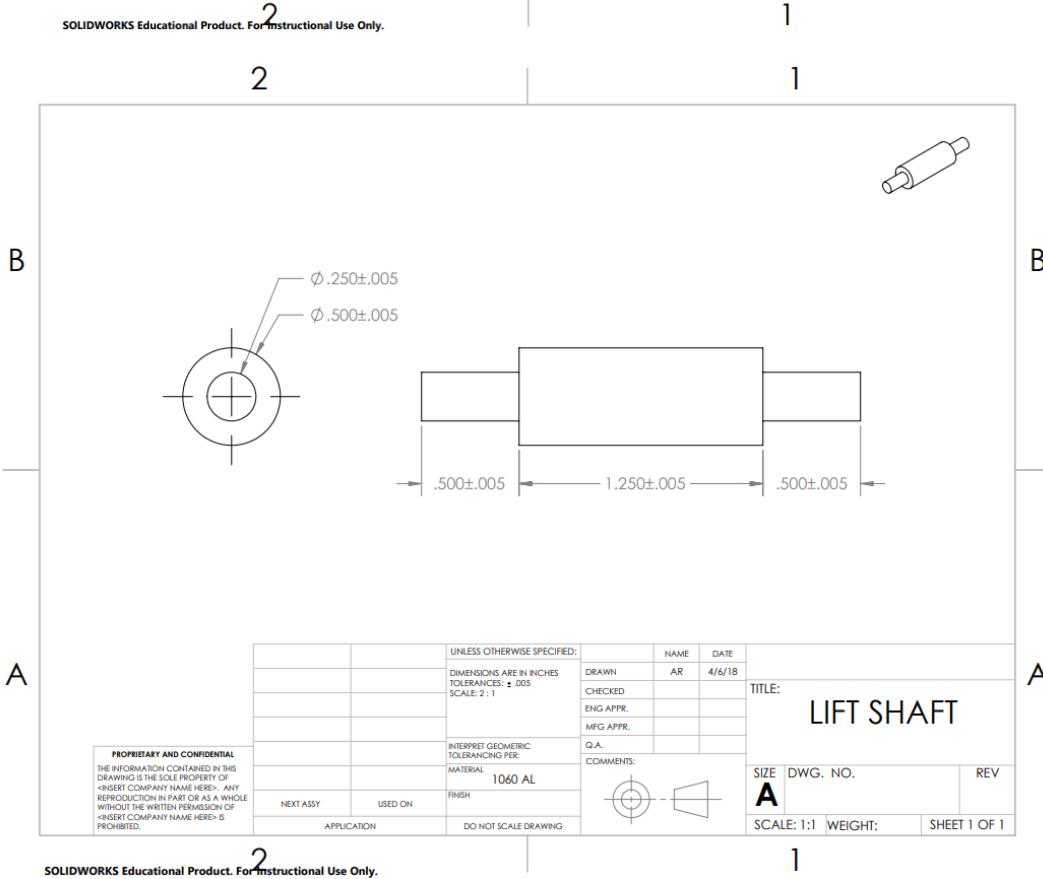
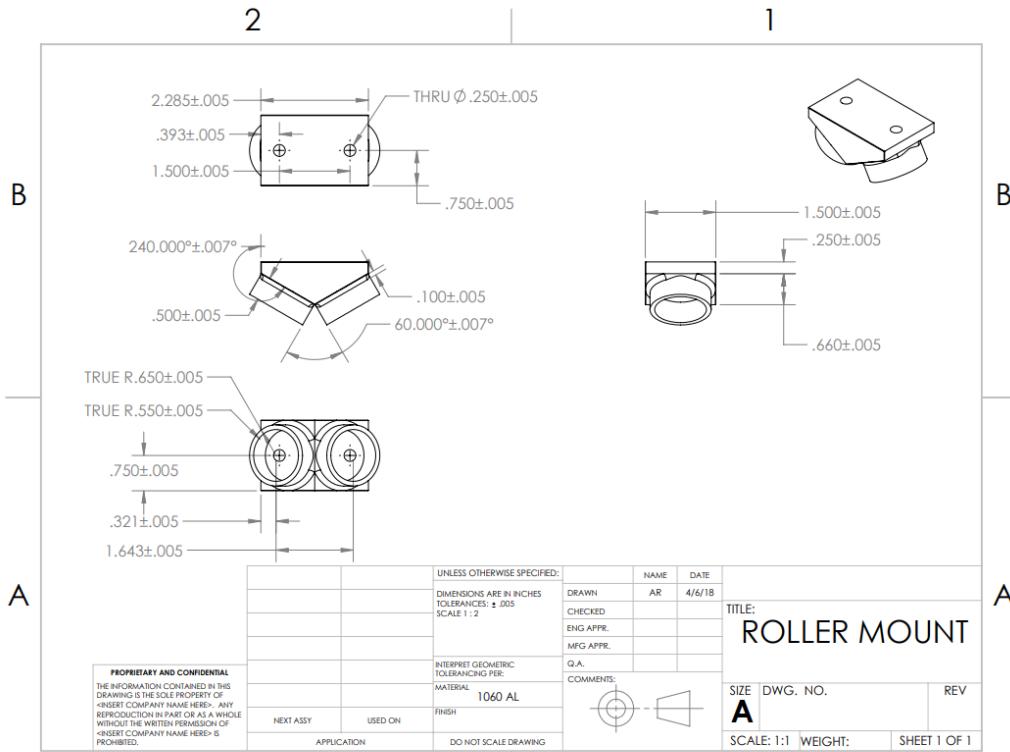


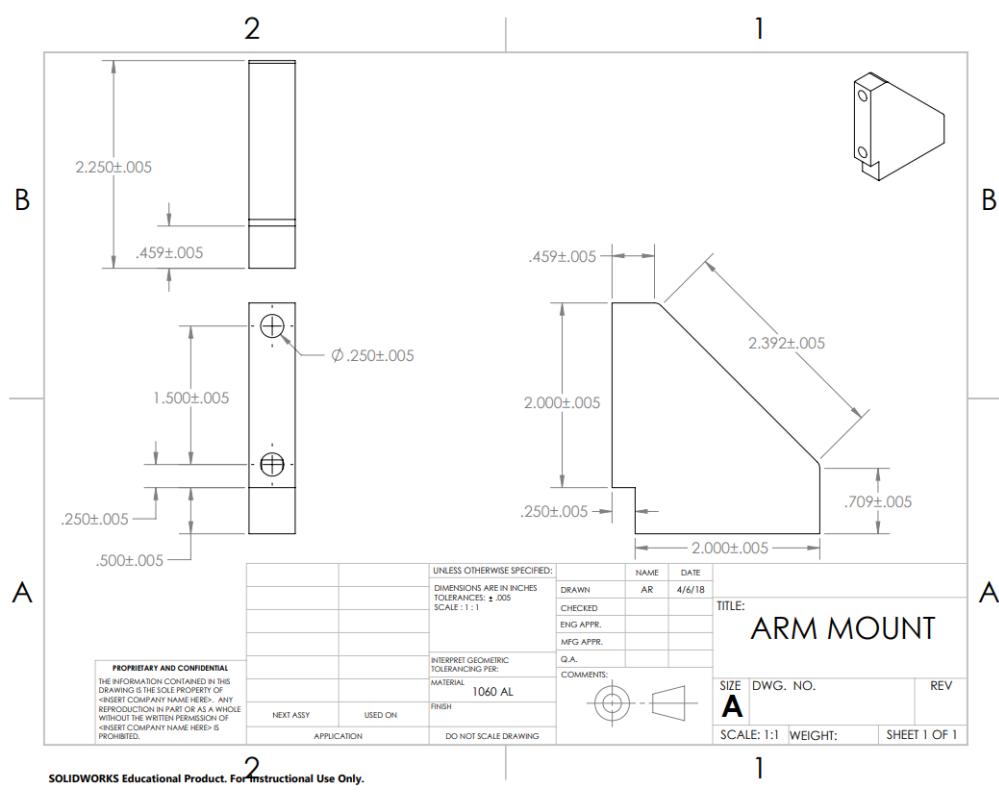
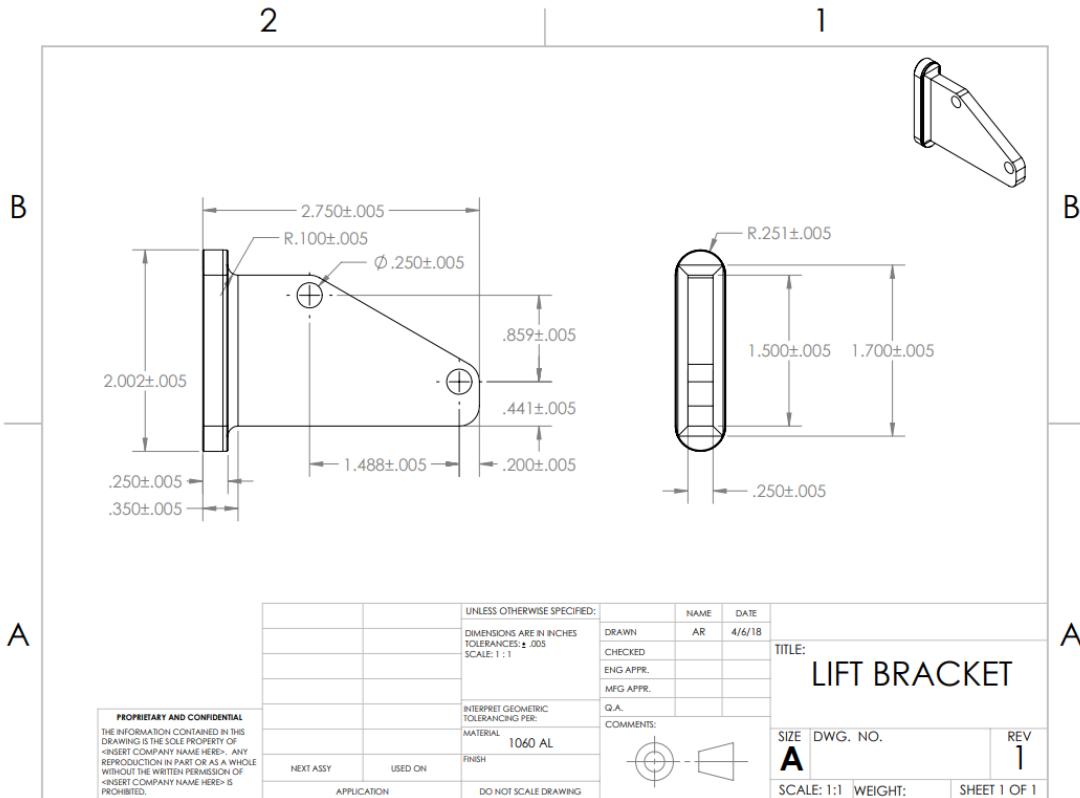
		UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: ± .005					
		DRAWN	NAME	DATE			
		AR		4/6/18			
		CHECKED					
		ENG APPR.					
		MFG APPR.					
		Q.A.					
		COMMENTS:					
		SIZE	DWG. NO.	REV			
A		<b>A</b>	<b>LIFT BACKING</b>				
		SCALE: 1:1 WEIGHT: SHEET 1 OF 1					

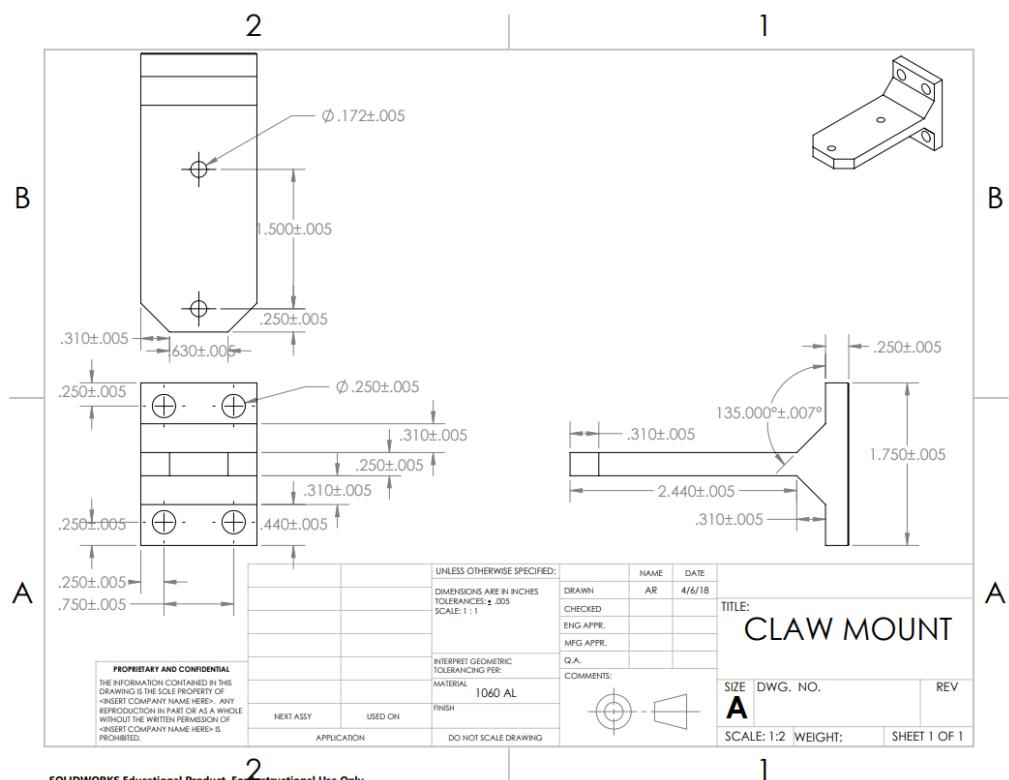
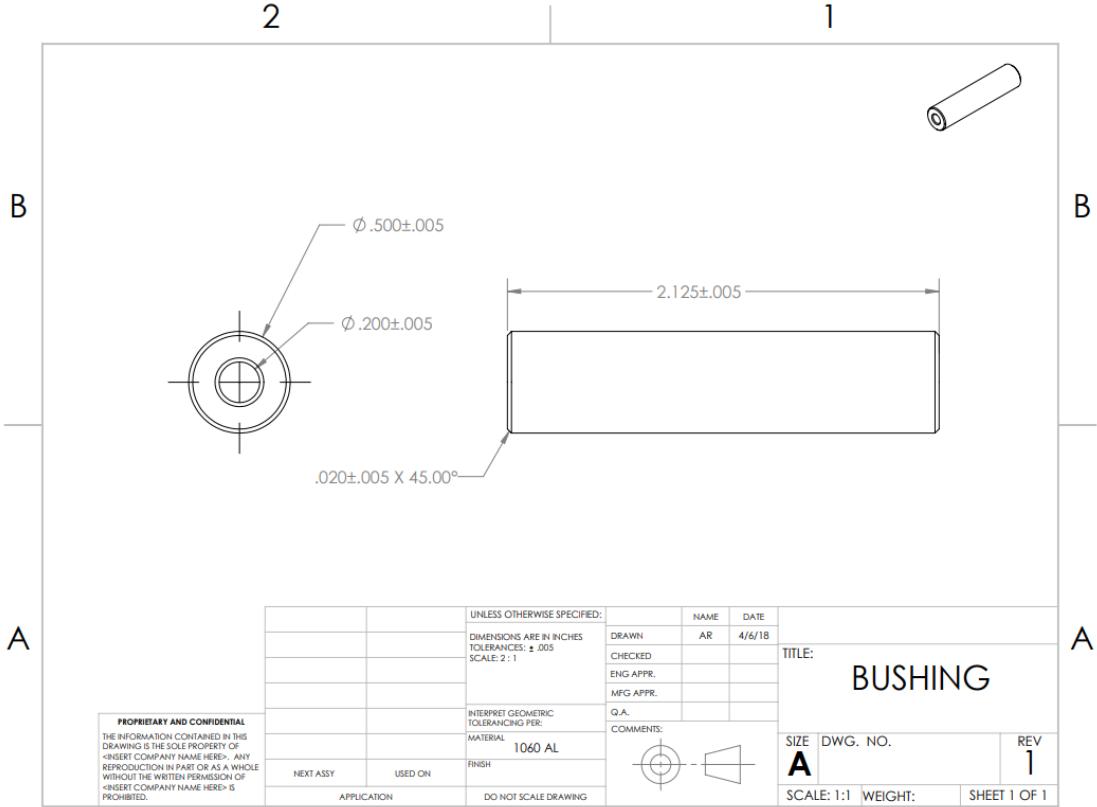
**PROPRIETARY AND CONFIDENTIAL**  
THE INFORMATION CONTAINED IN THIS DRAWING IS THE PROPERTY OF  
<INSERT COMPANY NAME HERE>. ANY  
REPRODUCTION IN PART OR AS A WHOLE  
WITHOUT THE WRITTEN PERMISSION OF  
<INSERT COMPANY NAME HERE> IS  
PROHIBITED.

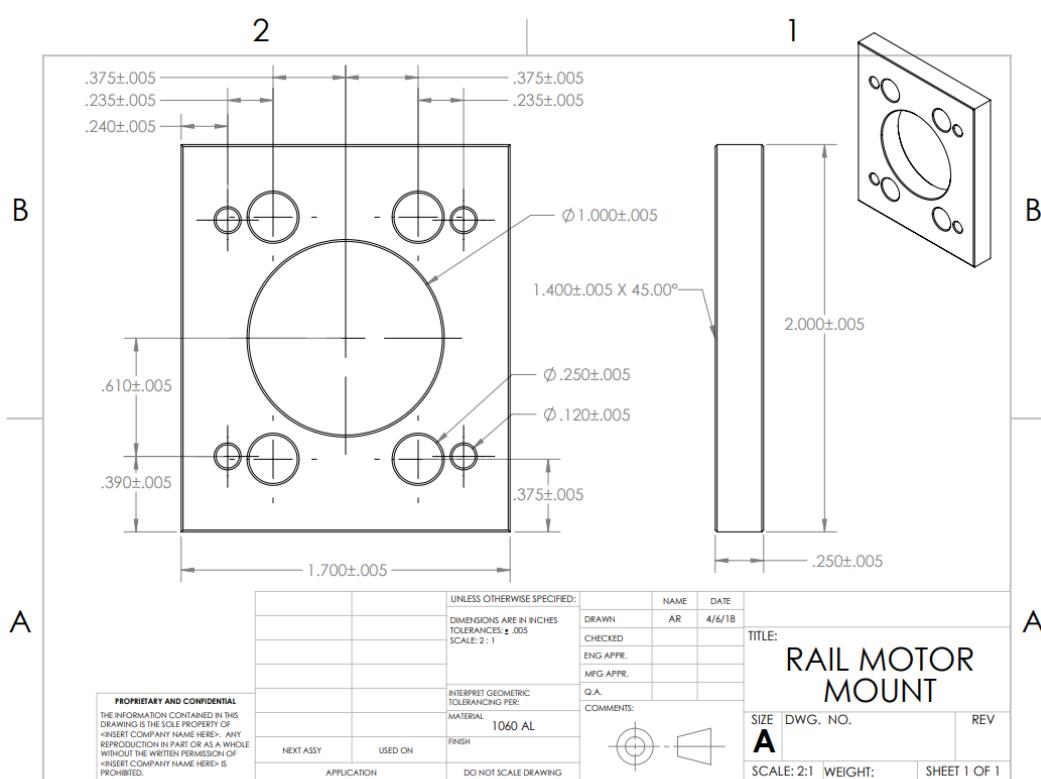
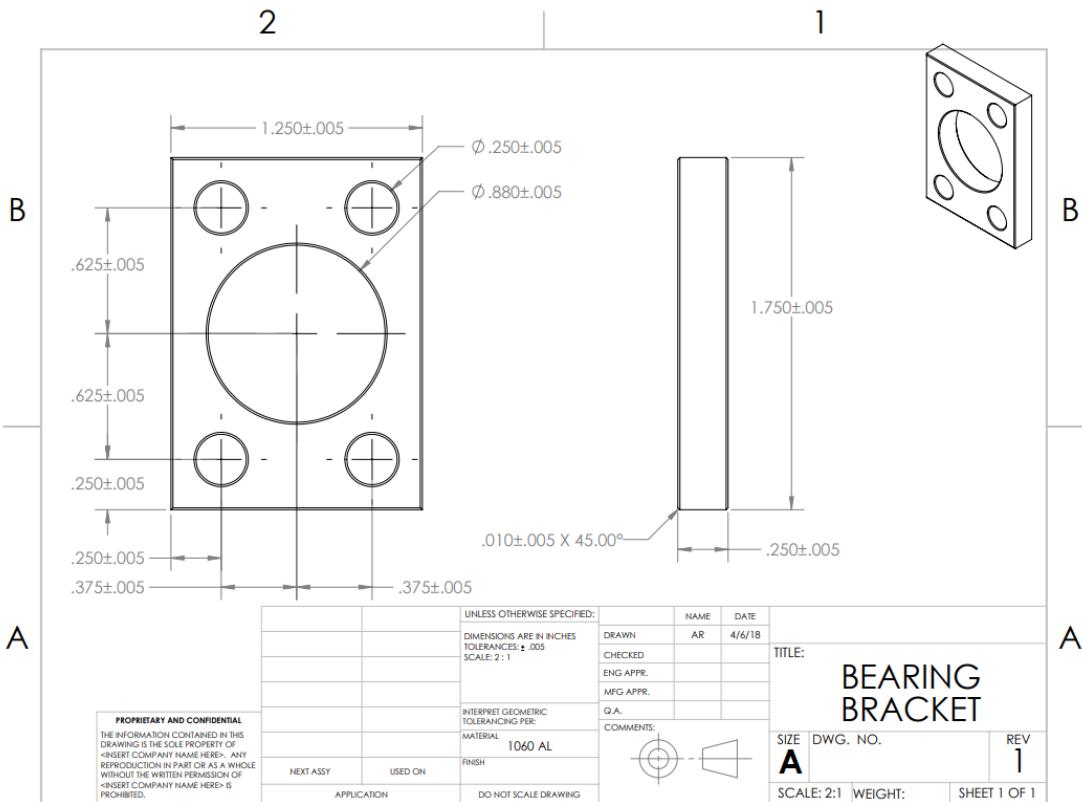
SOLIDWORKS Educational Product. For Instructional Use Only.

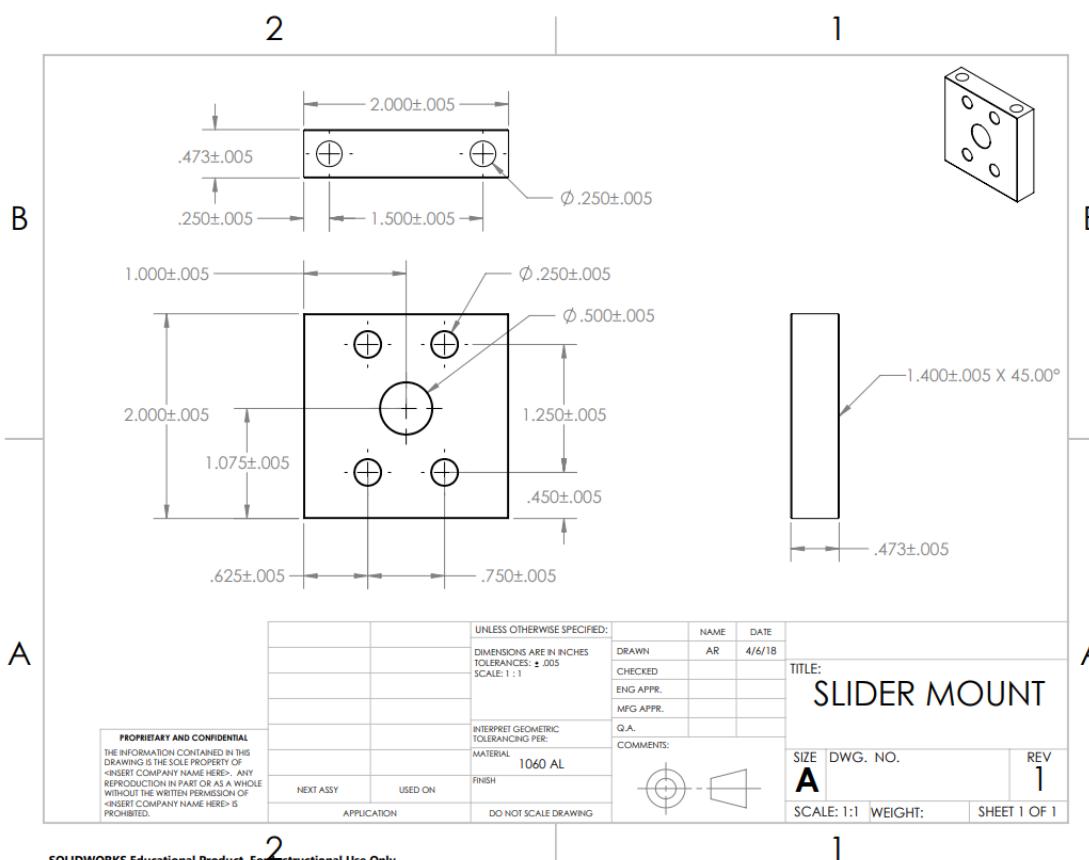
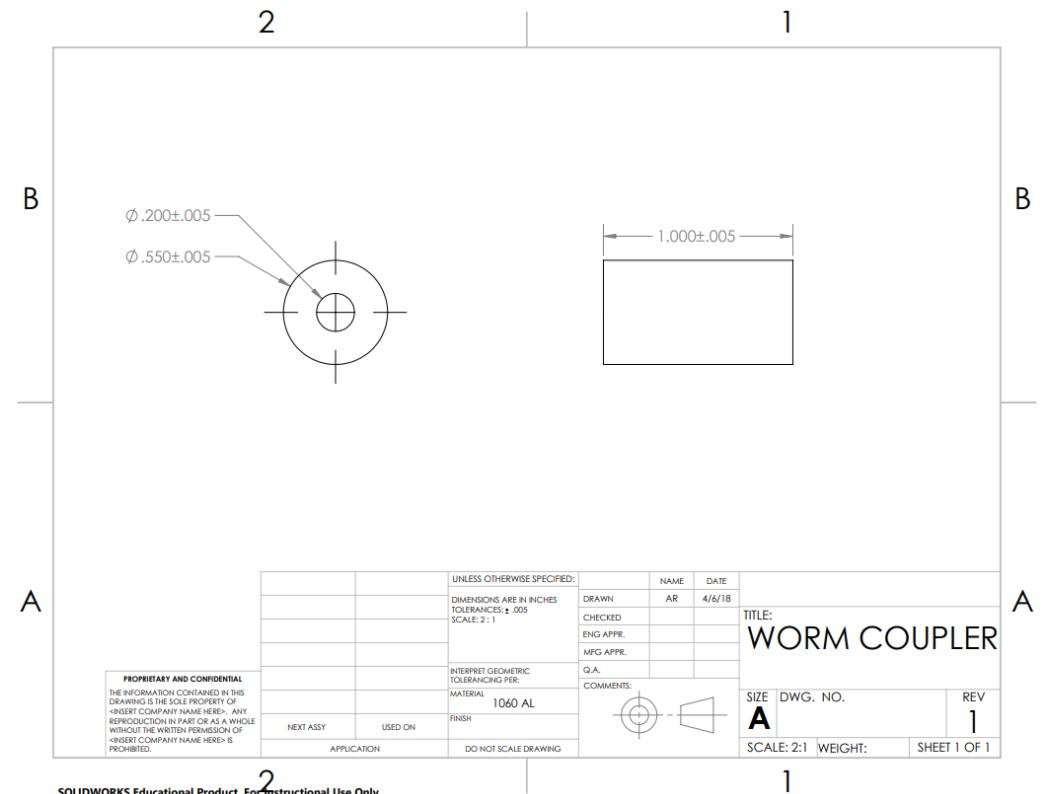


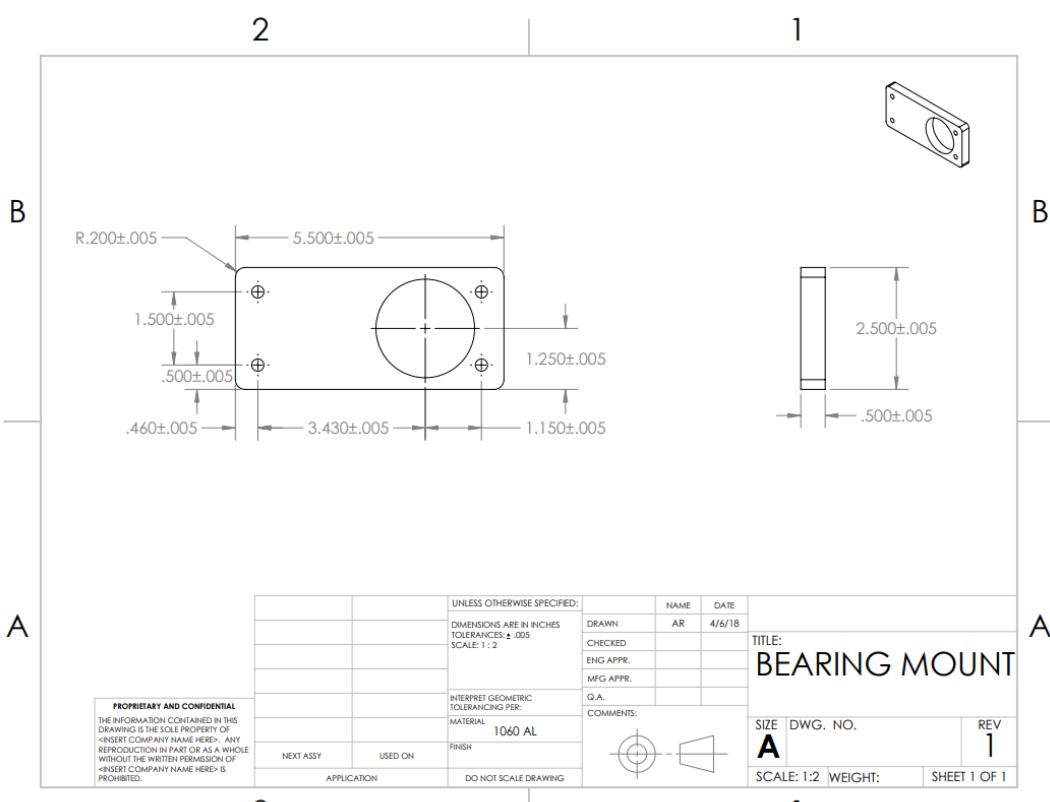
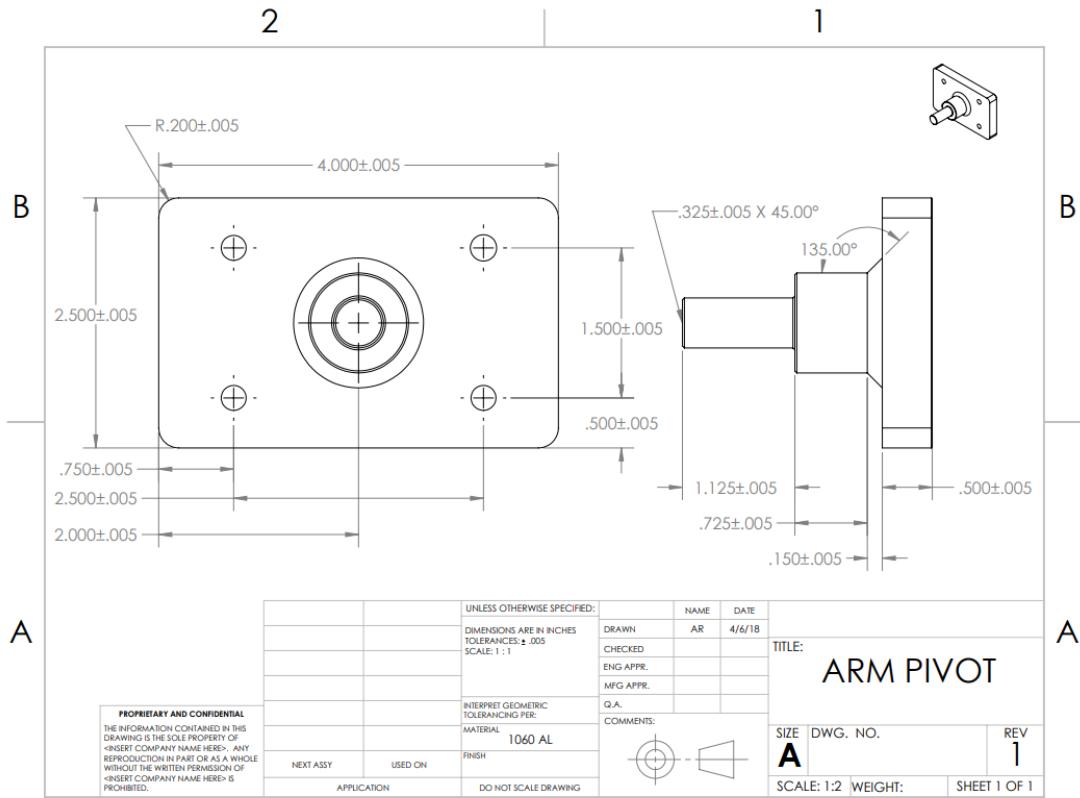


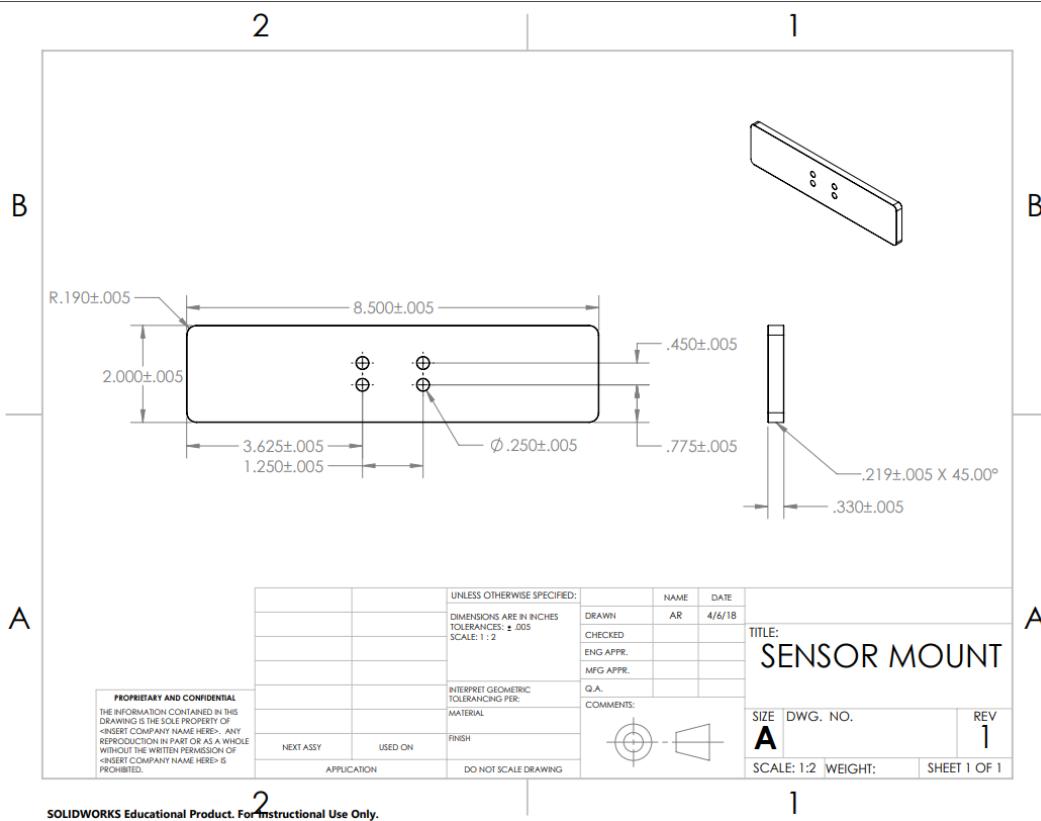
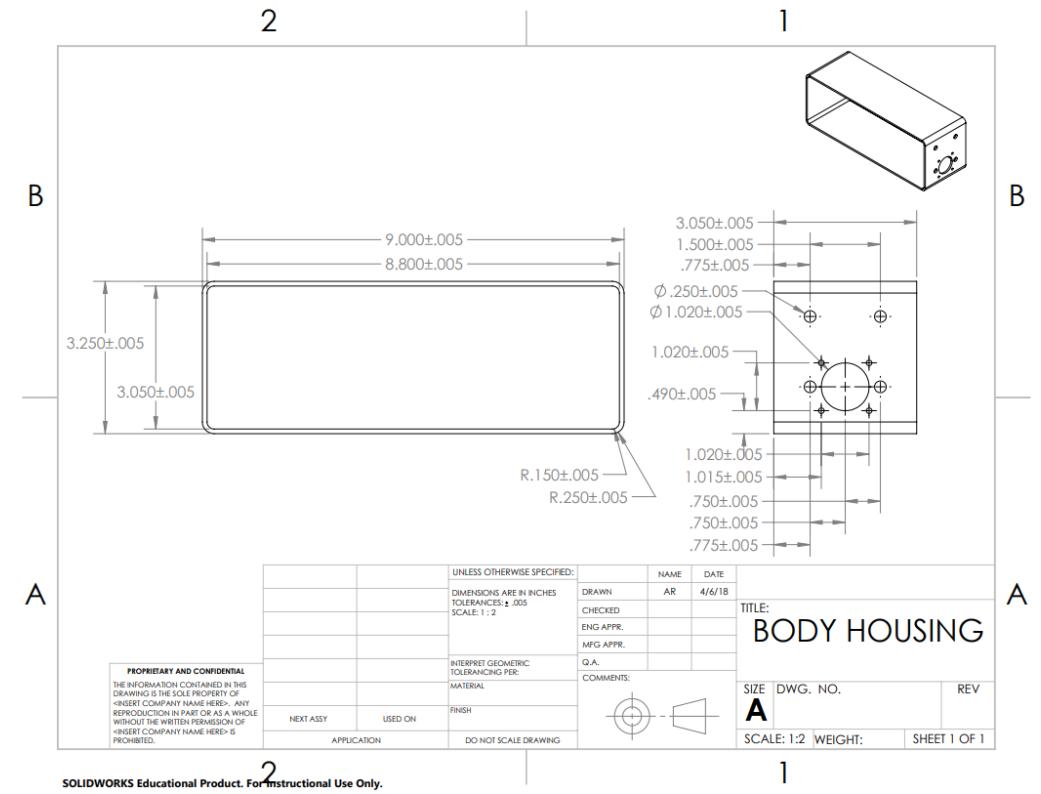


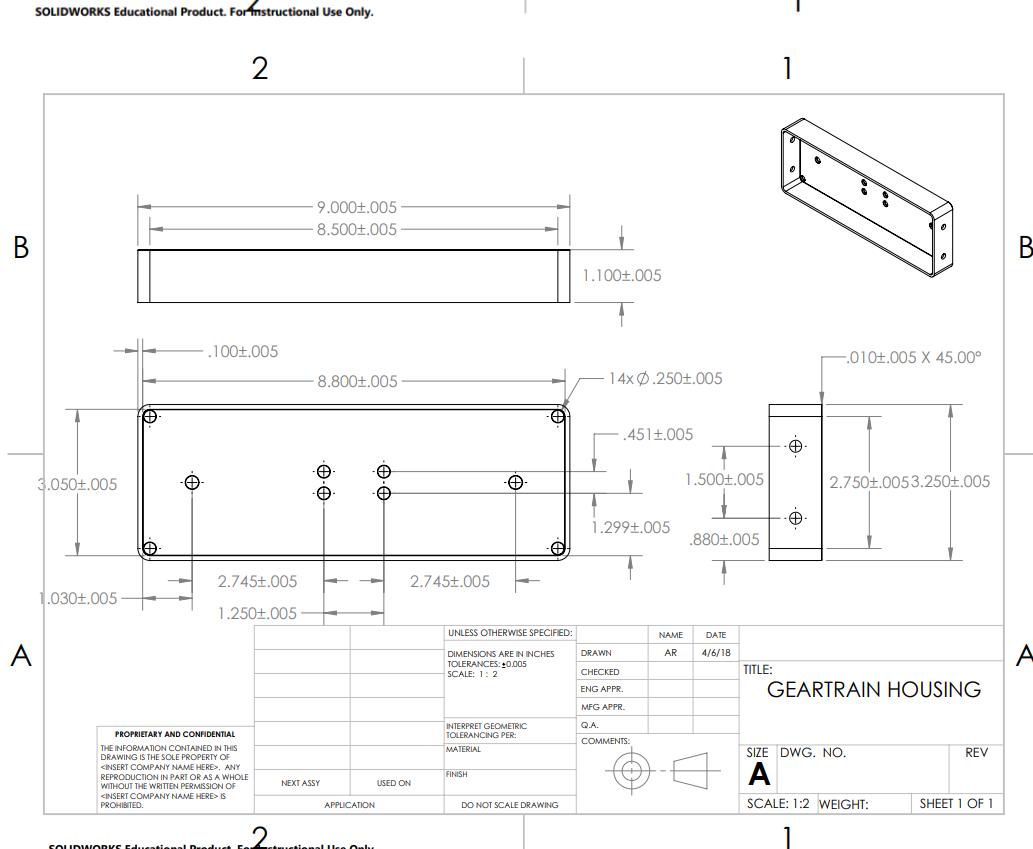
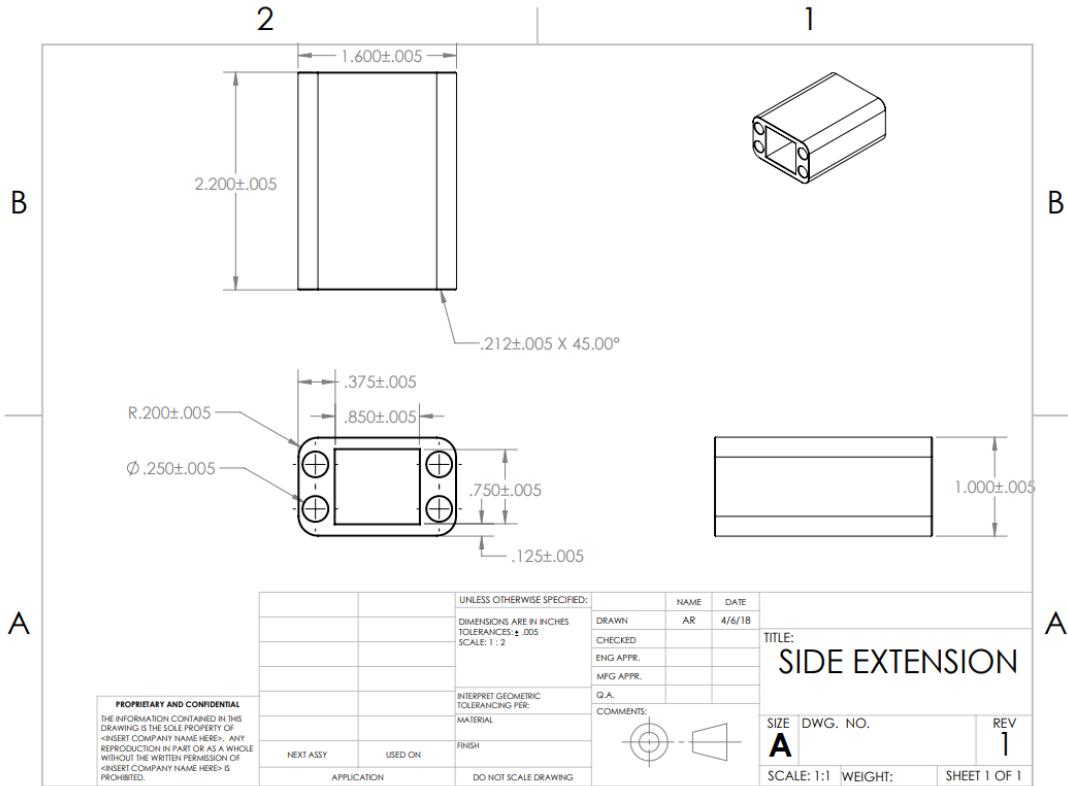


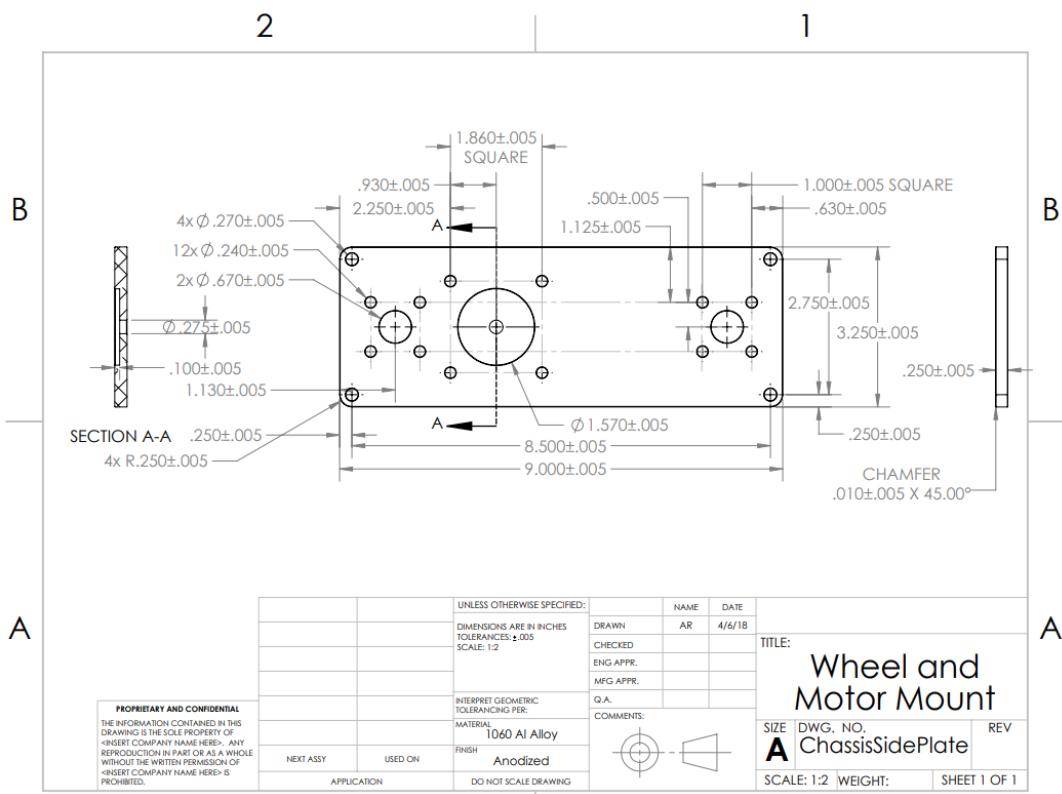
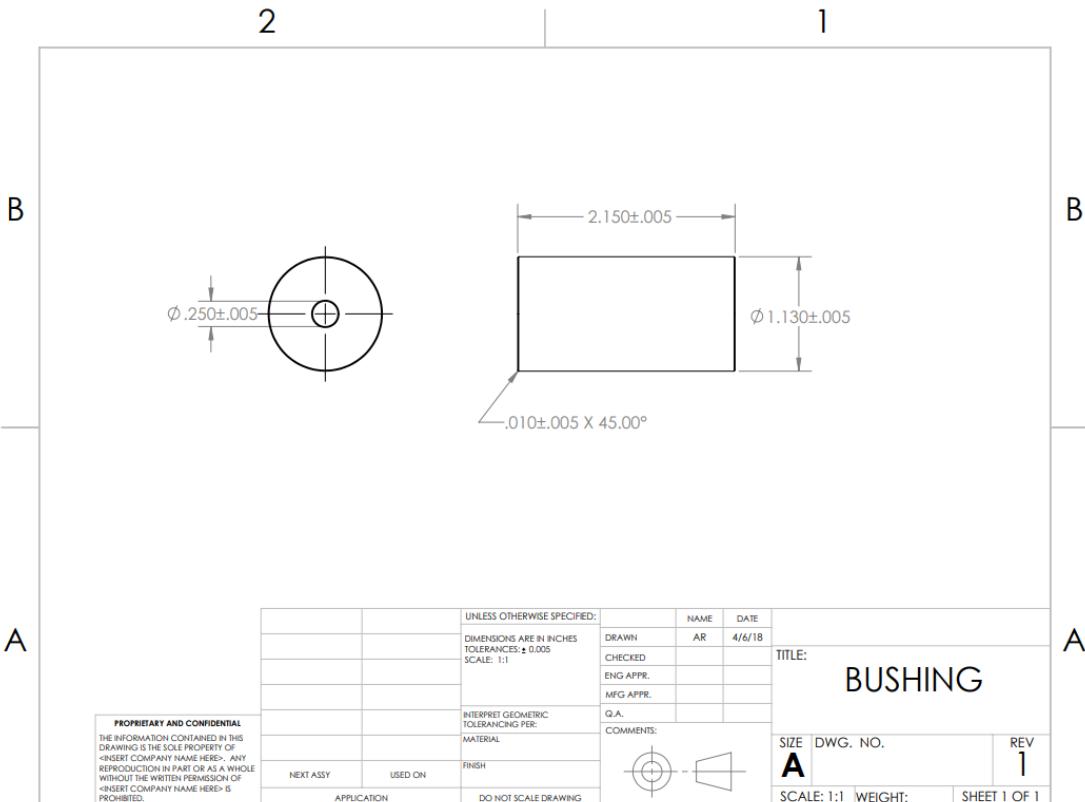


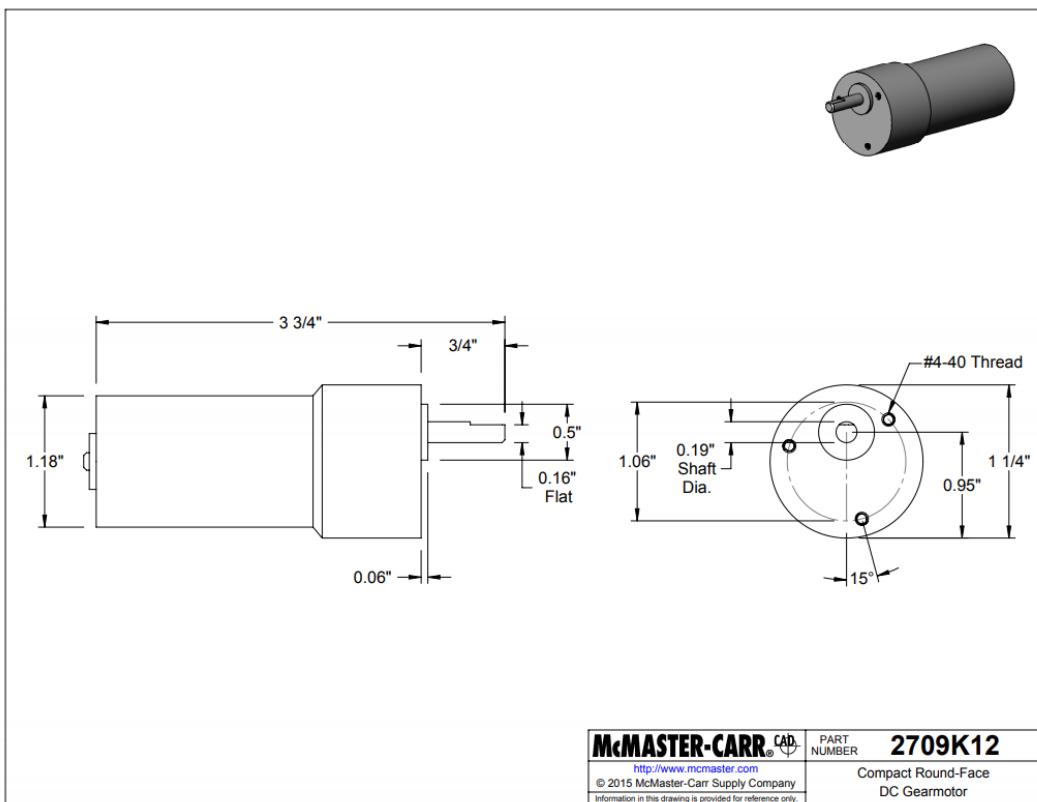
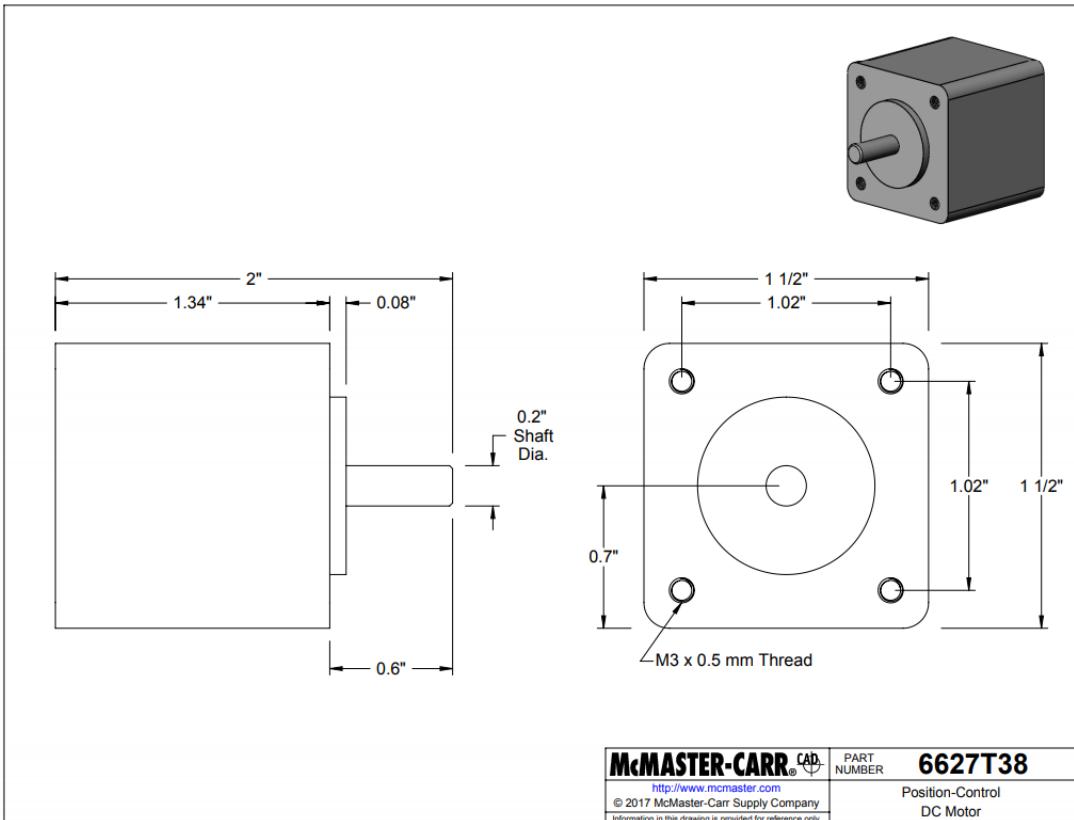


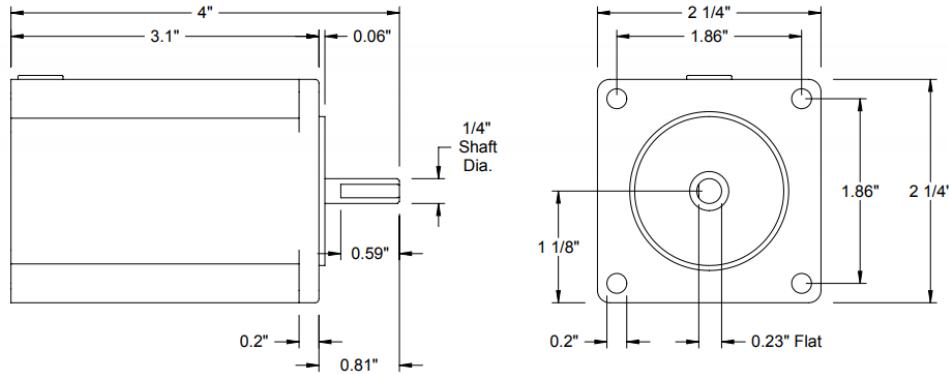
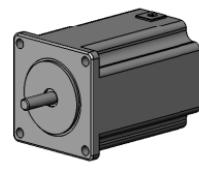










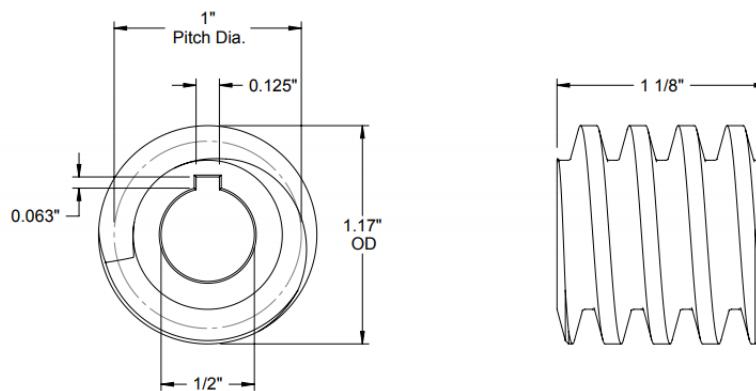


**McMASTER-CARR CAD**  
http://www.mcmaster.com  
© 2015 McMaster-Carr Supply Company  
Information in this drawing is provided for reference only.

PART NUMBER **6627T54**  
Position-Control  
DC Motor

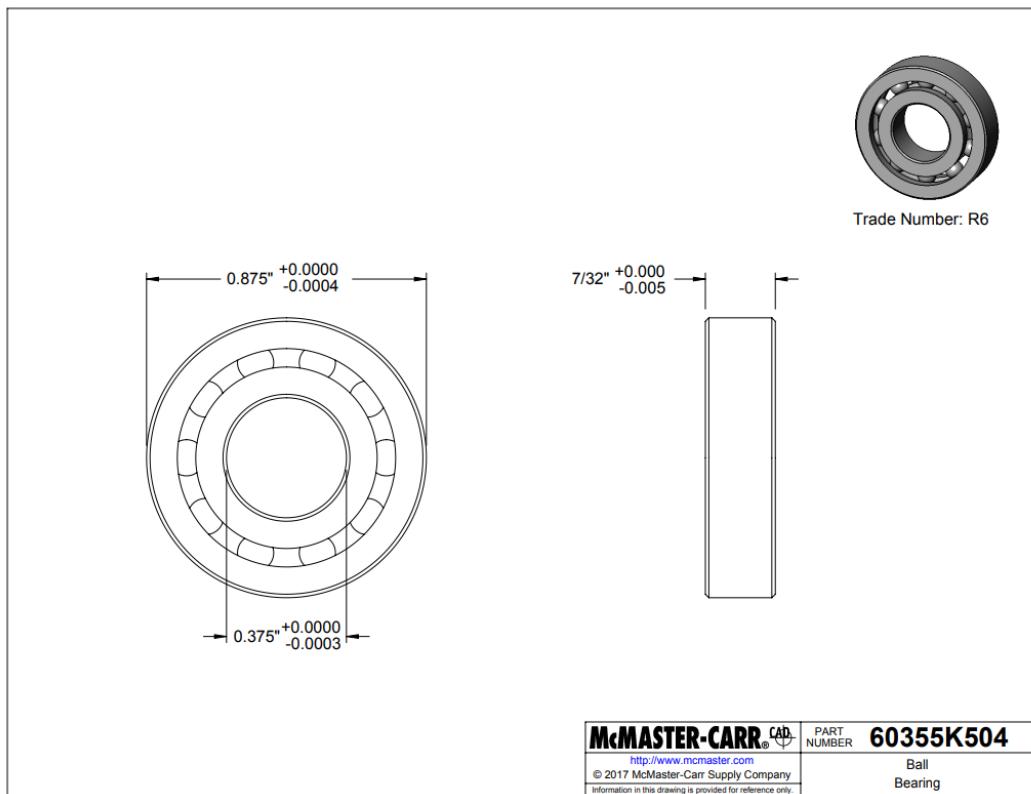
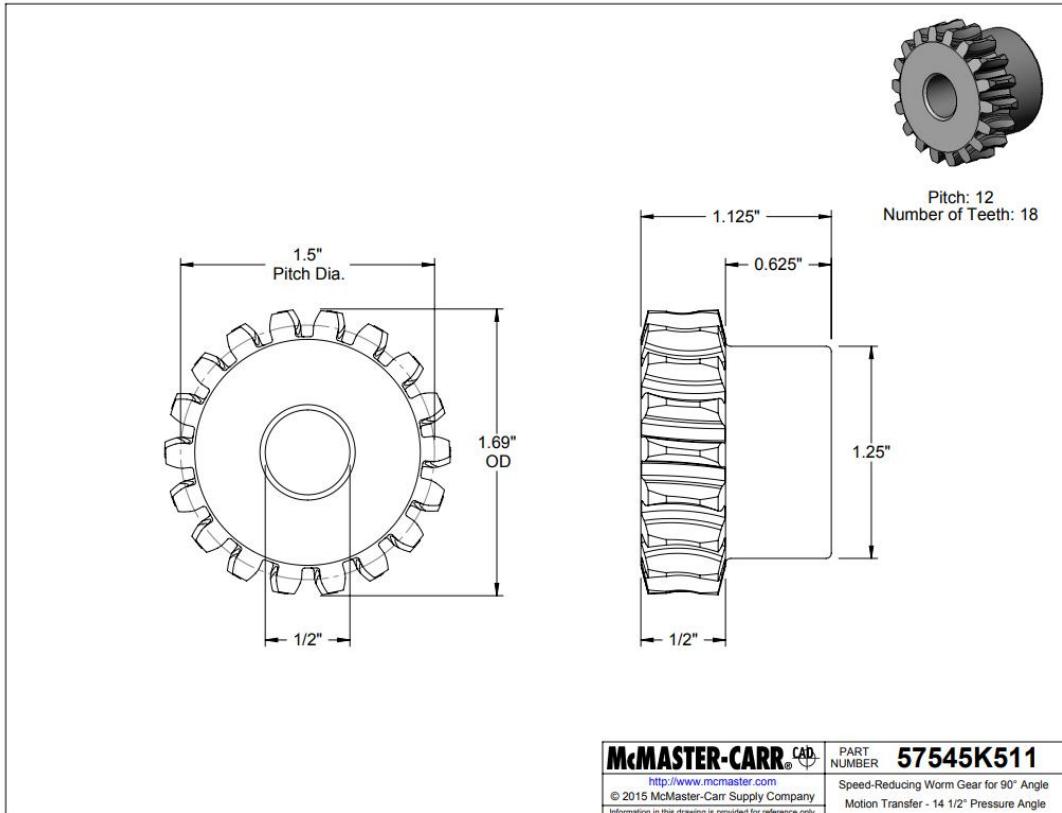


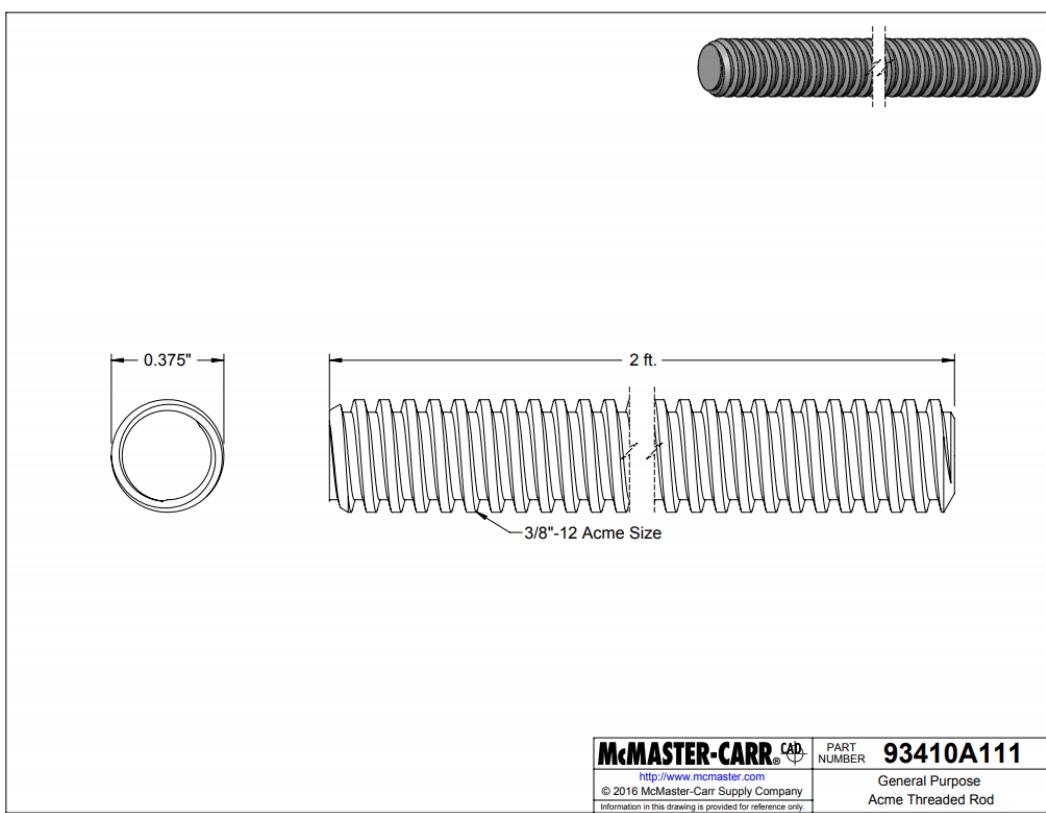
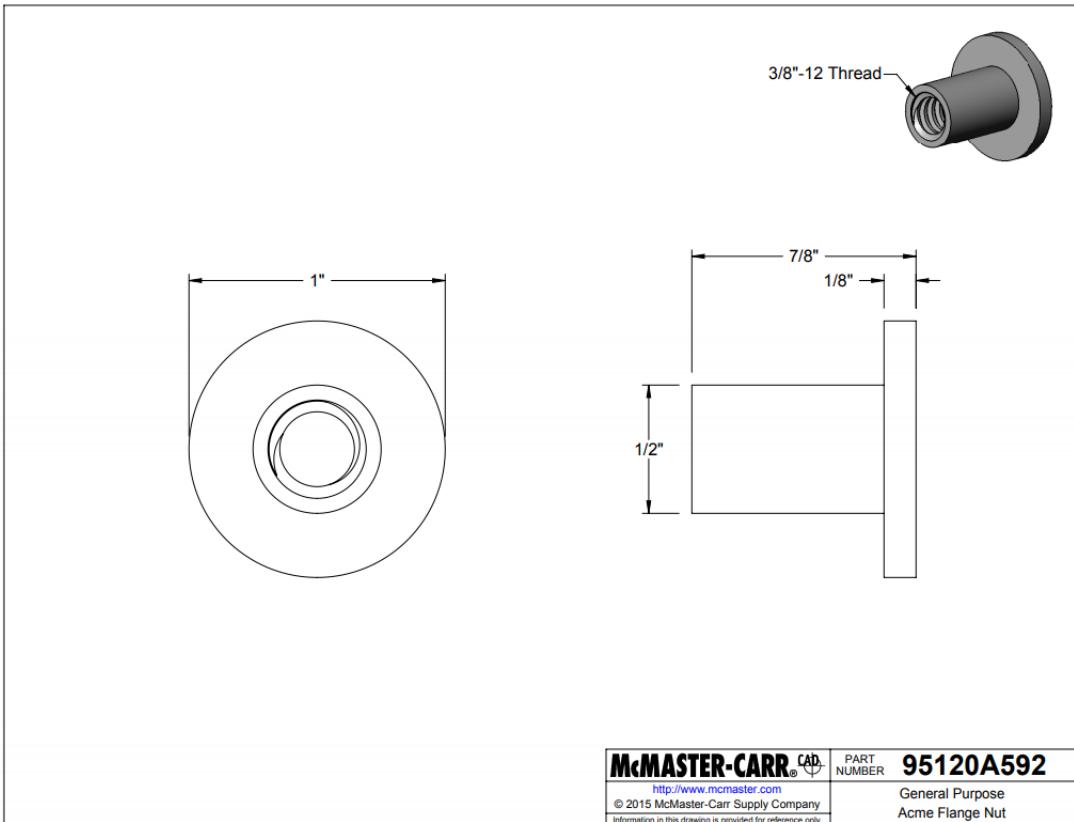
Pitch: 12

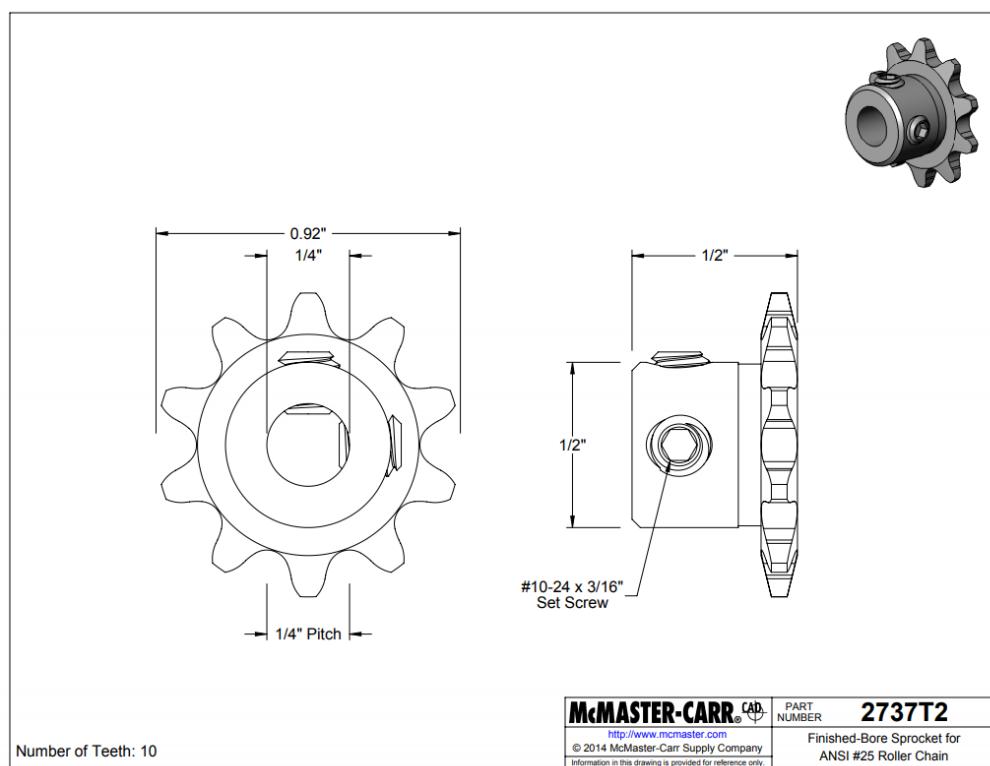
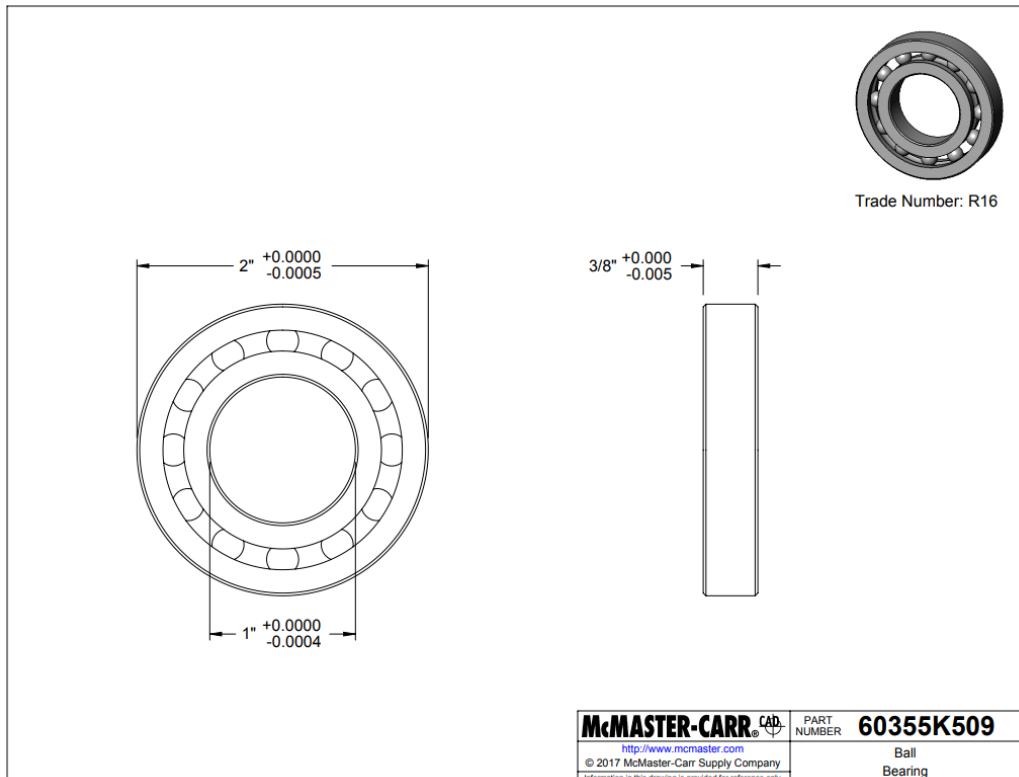


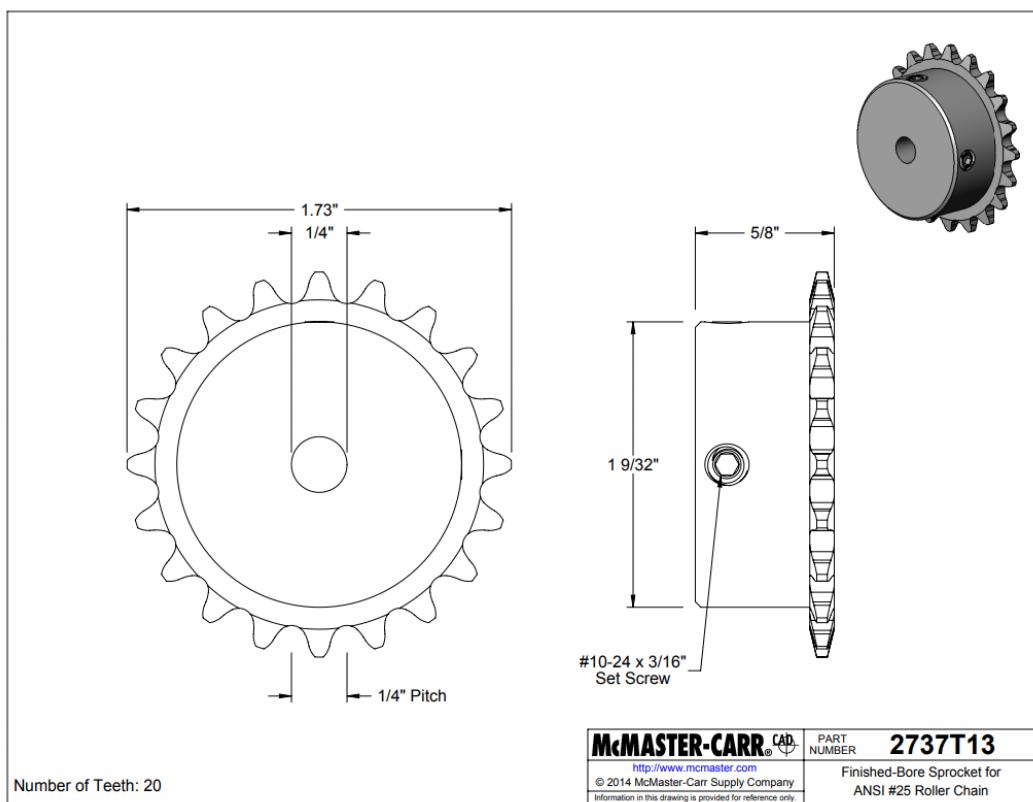
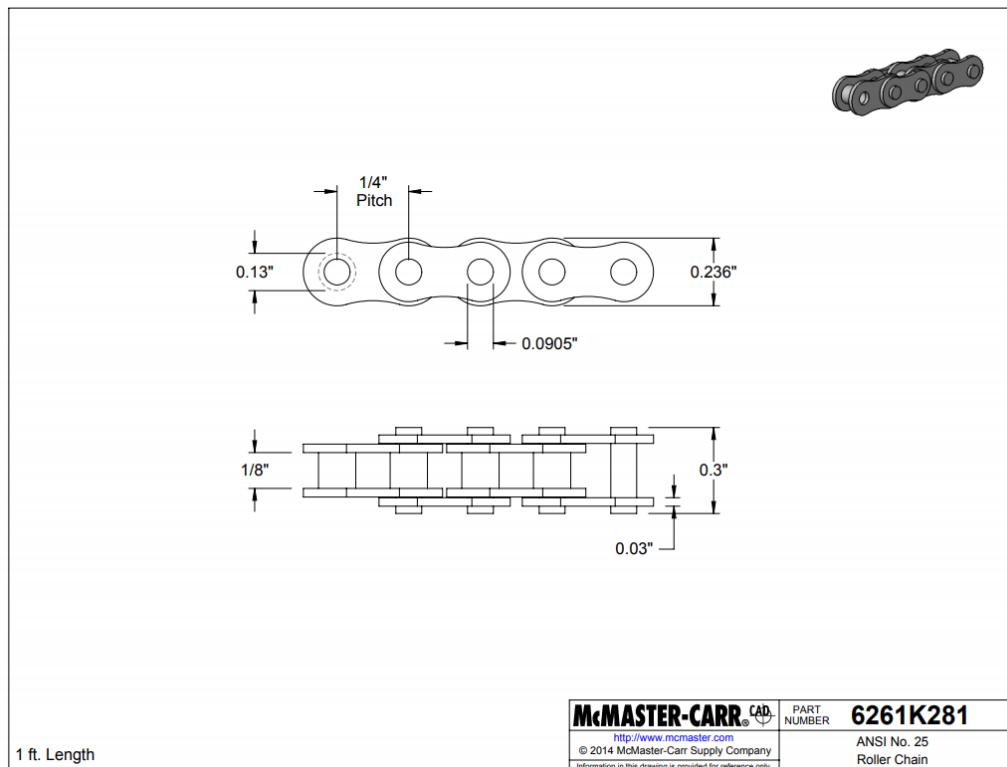
**McMASTER-CARR CAD**  
http://www.mcmaster.com  
© 2015 McMaster-Carr Supply Company  
Information in this drawing is provided for reference only.

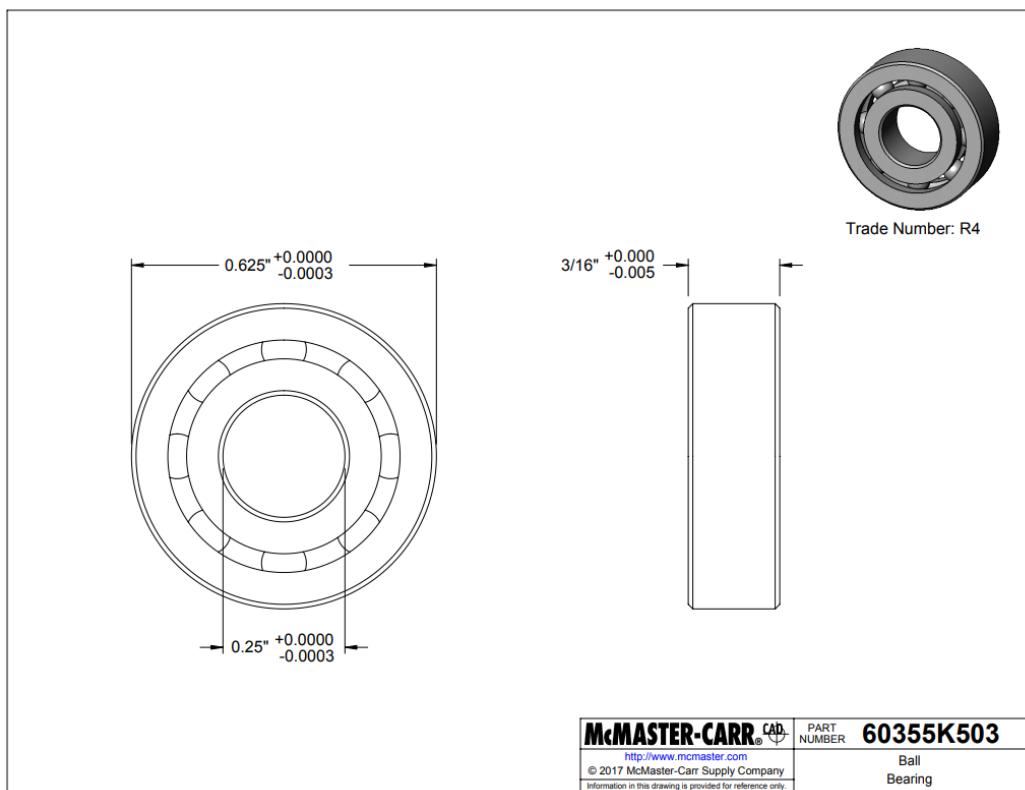
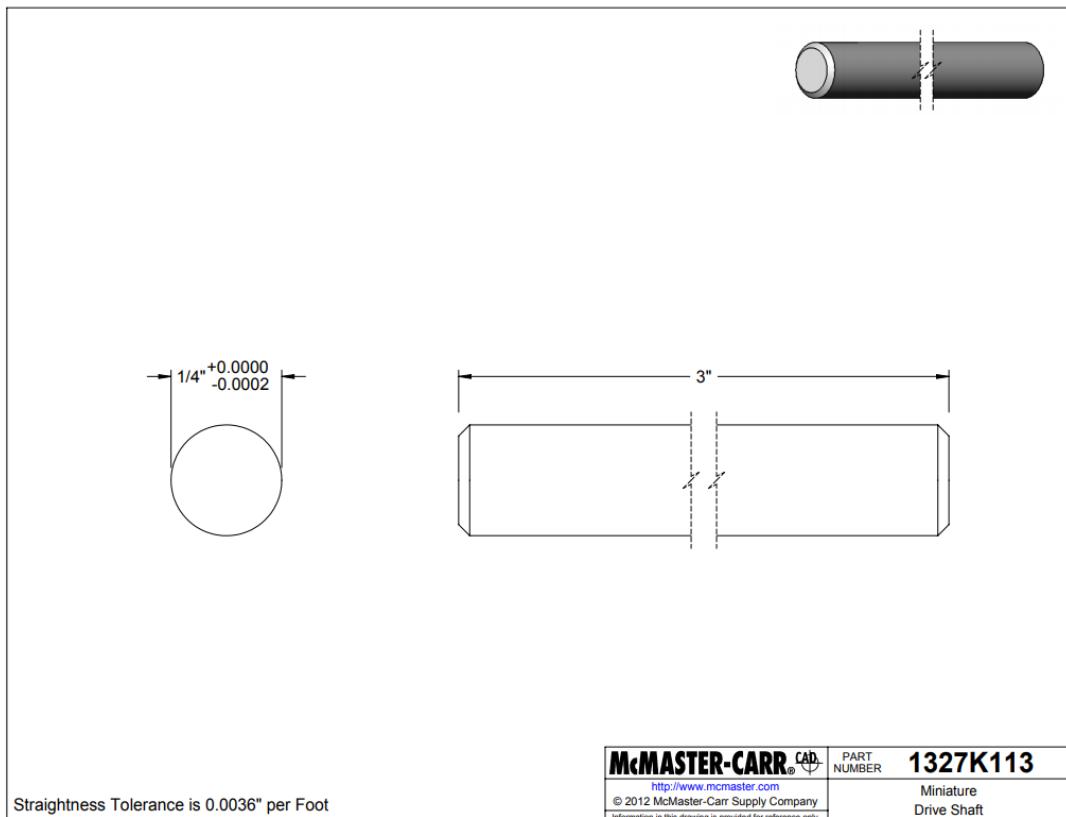
PART NUMBER **57545K527**  
Worm for Speed-Reducing Worm Gears for 90°  
Angle Motion Transfer - 14 1/2° Pressure Angle









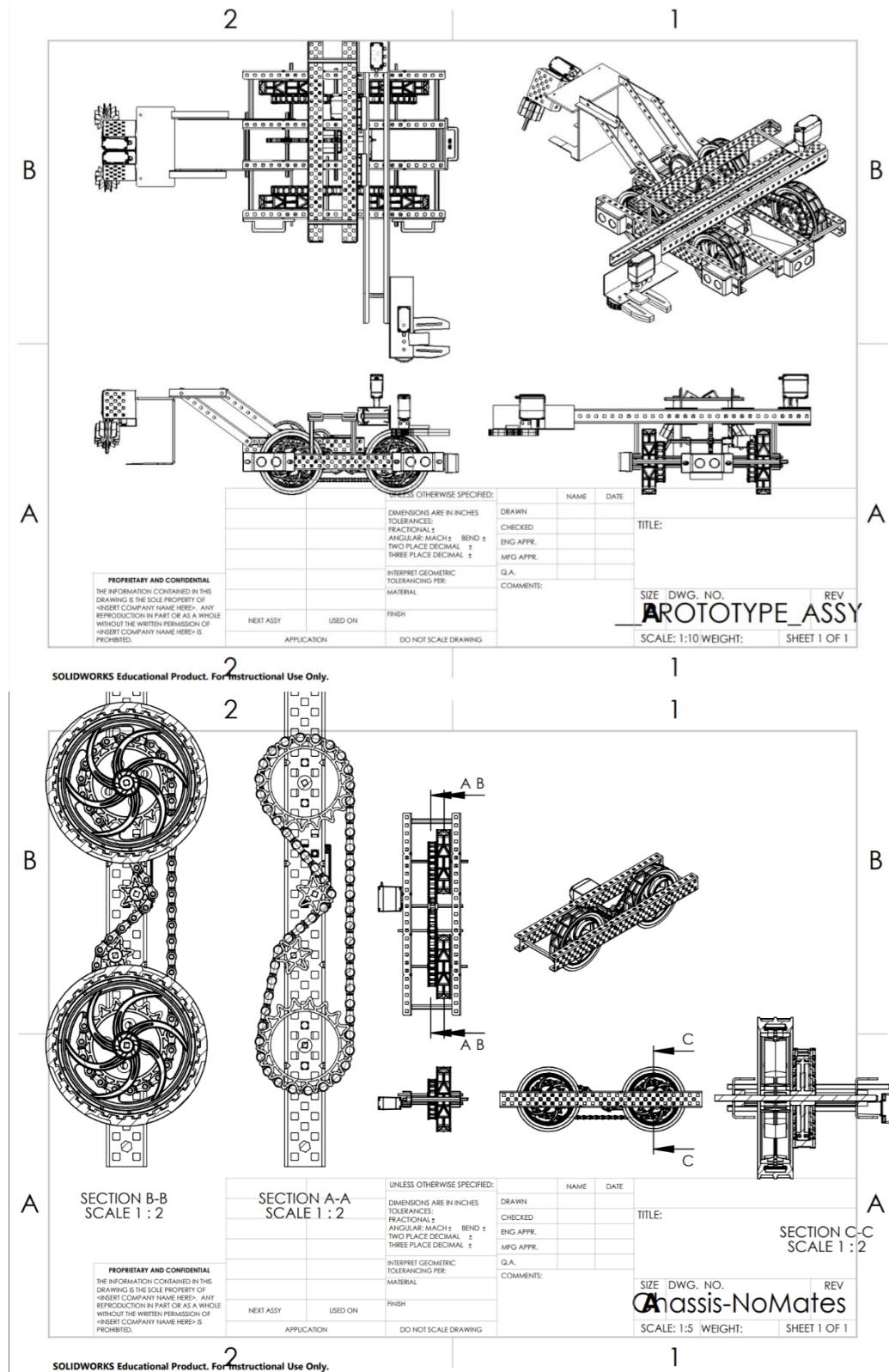


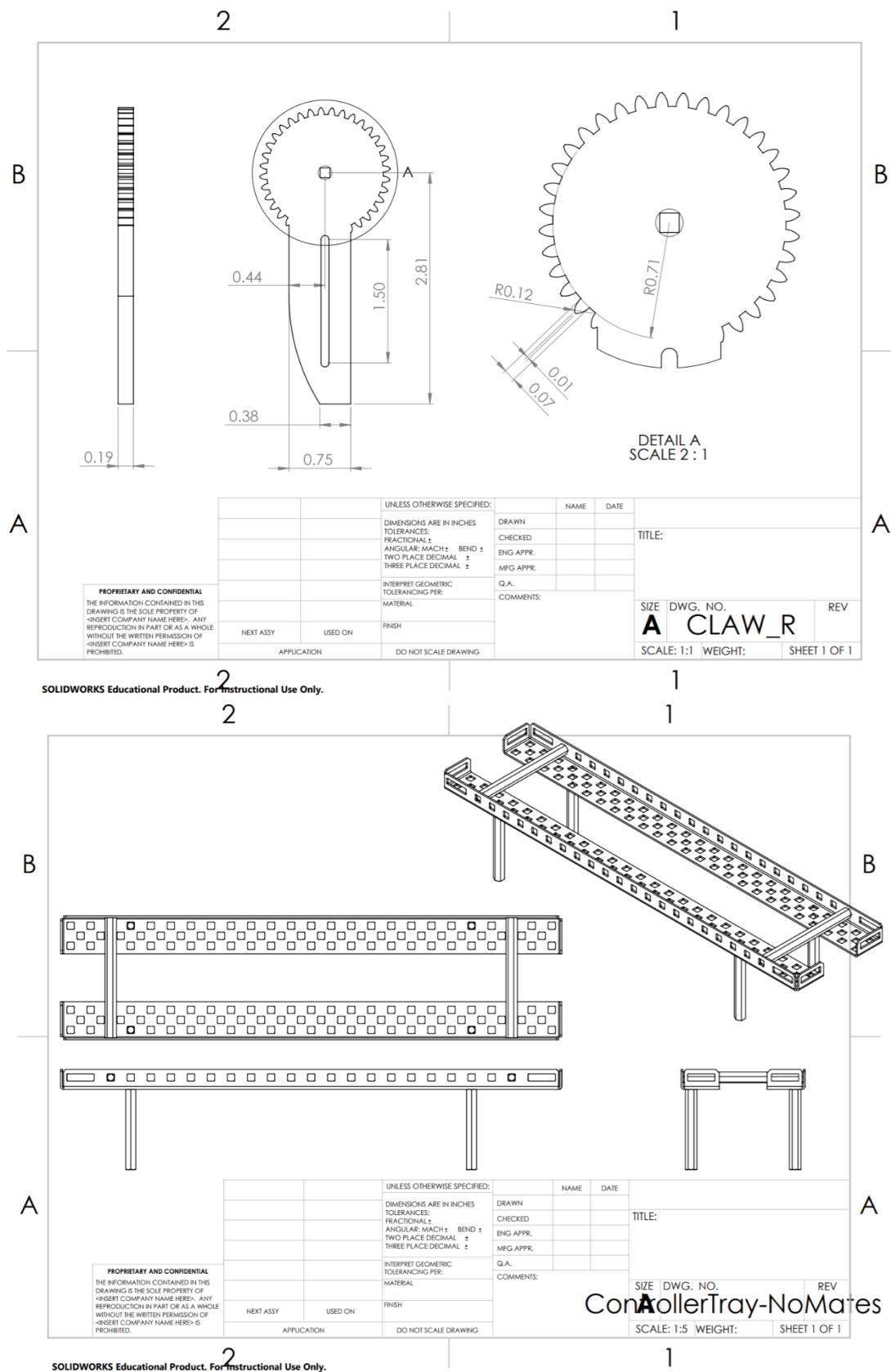
## Prototype BOM

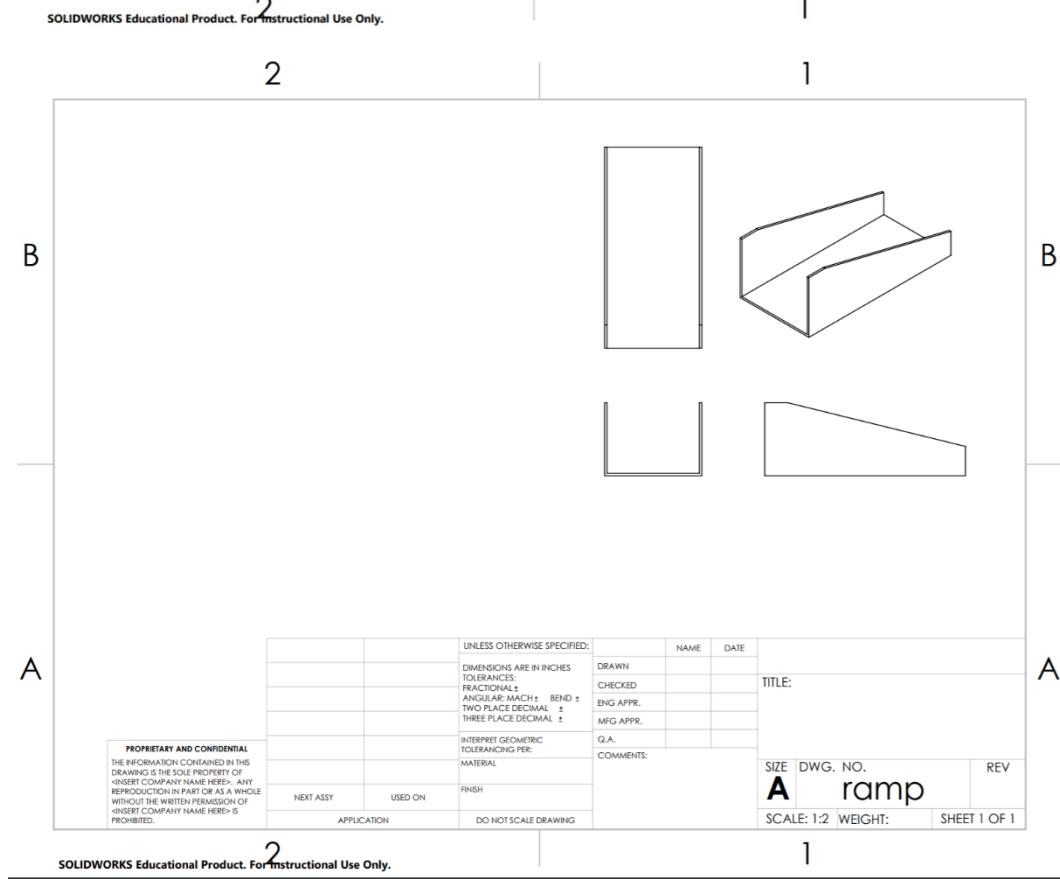
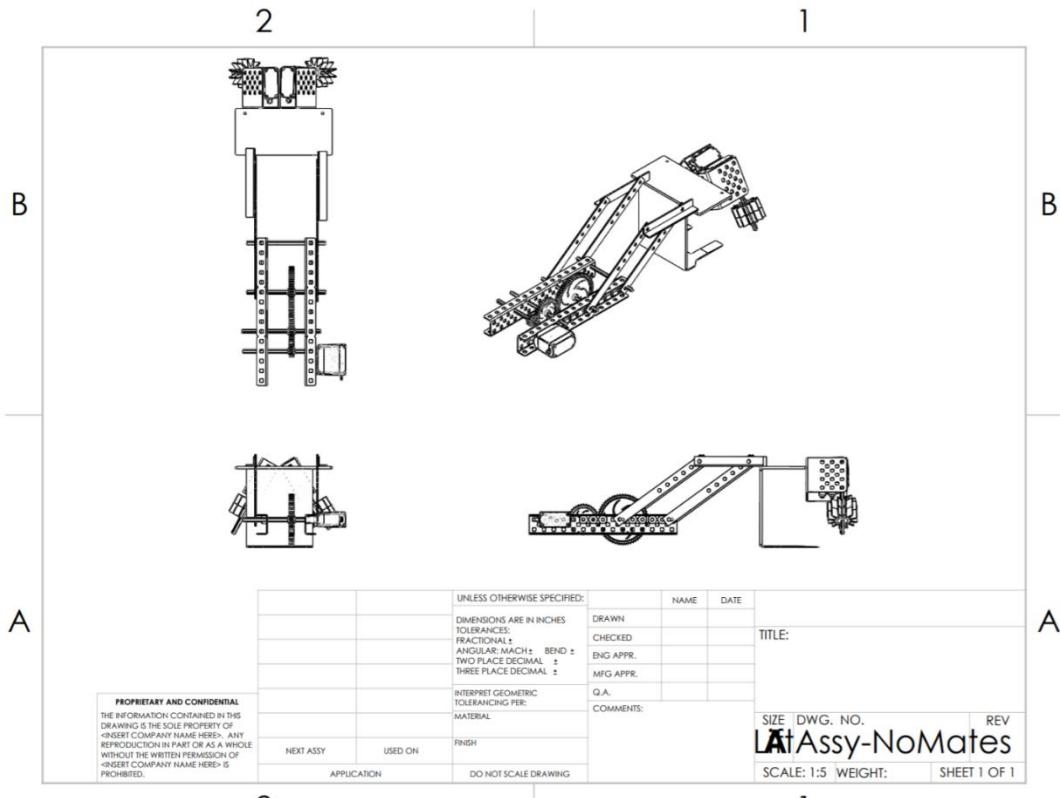
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	VL-CHAN-121-25 RevA	Chassis Structure	4
2	BEAM-3000	3" Standoff	10
3	BEARING-FLAT	Shaft Bearing	14
4	VEX-MEDIUM-WHEEL	4" Wheel	4
5	SHAFT-4000	4" Shaft	14
6	VEX-2WIRE-MOTOR	VEX DC Motor	6
7	sprocket-hs-06-tooth	6 tooth roller chain sprocket	4
8	sprocket-hs-18-tooth	18 tooth roller chain sprocket	4
9	chain-hs-link	Roller Chain Link	110
10	VL-RAIL-21-25 RevA	Liner Slide Rail	2
11	BEAM-2000	2" Standoff	4
12	276-1926-002 Rev1	Liner Slide Hardware	1
13	276-1926-004 Rev1	Liner Slide Hardware	3
14	276-1926-005 Rev1	Liner Slide Hardware	2
15	BEAM-1500	1.5" Standoff	2
16	Claw-Bracket	Bracket for claw	1
17	CLAW_L	Left Claw	2
18	CLAW_R	Right Claw	2
19	VEX-MOTOR	Vex Servo	1
20	magmount	mount for LSM303	1
21	VL-CHAN-121-15 RevA	Structure	2
22	VEX-12-TOOTH-GEAR	12 tooth gear	1
23	VEX-36-TOOTH-GEAR	36 tooth gear	1
24	VEX-60-TOOTH-GEAR	60 tooth gear	1
25	backForkLift	back of pyr intake	1
26	BottomForklift	bottom of pyr intake	1
27	Circles 5.5in Inch Shaft Bar	pyr linkage	4
28	Circles 3.5 inch Linkage	pyr linkage	1
29	topForkLift	top of pyr intake	1
30	VL-RACK-BRK RevA	Liner Slide Hardware	2
31	INTAKE-ROLLER	Pyr Hardware	4

32	Circles 3.5 inch Linkage-o	pyr linkage	1
33	rampCube	slide for cube	1
34	Ultrasonic_Range_Finder_276-2155	ultrasonic sensor	3
35	IR	IR Sensors	2
36	STM32F429ZI	Cortex M4 MCU	1
37	Daughter Board for MCU	Connect sensors and motors to MCU	1
38	Elastic Bands	stress relif of pyr intake	4
39	Limit Switches		3
40	Misc Hardware	Nuts and Bolts	...

## Prototype Technical Drawings







## Final Code

MSE-bot.h

//Main Class

//Compile Options and Reqs:

// Board: Nucleo-64

// Part Number: F446RE

// Upload Method: STLink

#include <Wire.h>

#include <Stream.h>

#include <SPI.h>

#include <Servo.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_LSM303\_U.h>

#include <math.h>

//pin assignments

#define LEFT\_MOTOR D10

#define RIGHT\_MOTOR D9

#define LR\_ULTRASONIC\_IN A1

#define LR\_ULTRASONIC\_OUT A0

#define LF\_ULTRASONIC\_IN A3

#define LF\_ULTRASONIC\_OUT A2

#define F\_ULTRASONIC\_IN A5

#define F\_ULTRASONIC\_OUT A4

#define CUBE\_INTAKE\_ARM D12

#define CUBE\_INTAKE\_CLAW D8

```

#define PYR_INTAKE_LIFT          D13
#define PYR_INTAKE_WHEELS        D3

//##define TARGET_PYL_SW          PB15
#define START_SW                  PC13
#define PYR_INTAKE_SW             PG0
#define LIFT_LIMIT_SW0            PB2 //upper switch for retracted position
#define LIFT_LIMIT_SW1            PB1 //lower switch for extended position

//Program Parameters
#define WALL_TARGET_DIST          13 * 58 // 11 cm * 58 = 638
#define WALL_TARGET_TOLERANCE      30 // 1cm
#define PARALLEL_TOLERANCE         60 // 1.5cm
#define TURN_THRESHOLD              600
#define SWERVE_DELAY                2000
#define PING_MS                     10

#define FORWARD_SPEED_FAST         1800
#define REVERSE_SPEED_FAST         1200
#define FORWARD_SPEED_SLOW          1650
#define REVERSE_SPEED_SLOW          1350
#define STOP_VALUE                  1500

#define CUBE_MAG_GEN_THRESH        2000
#define CUBE_MAG_ACCURATE_THRESH    4500
#define IR_TIME_TOLERANCE           2000
#define CUBE_INTAKE_OPEN             65
#define CUBE_INTAKE_CLOSE            105

```

```
class MSEBot {
```

```
public:  
    bool _speedMode = 1;  
  
    unsigned int _LR_ultrasonic_dist;  
    unsigned int _LF_ultrasonic_dist;  
    unsigned int _F_ultrasonic_dist;  
    unsigned int _LRecho;  
    unsigned int _LFecho;  
    unsigned int _Fecho;  
    unsigned int _LastIRTime = 0;  
  
    char _IRValues[4] = {'A', 'E', 'I', 'O'};  
    bool _IRsw = 1; // 1 for AE, 0 for IO  
    bool _ArmPosition = 0;  
    bool _LiftPosition = 1;  
    bool _TurnCount = 0; // counter for turn direction for finding pyramid  
  
    int _compassHeading;  
    int _compassMagnitude;  
  
    sensors_event_t compassData;  
  
    Servo _leftMotor;  
    Servo _rightMotor;  
    Servo _armMotor;  
    Servo _clawMotor;  
    Servo _liftMotor;  
    Servo _intakeMotor;  
  
    Adafruit_LSM303_AccelMag;
```

```
public:  
    void init();  
  
    void findWall();  
    void PingUltra();  
    void PingFront();  
    void TurnOnAxisL();  
    void TurnOnAxisR();  
    void goForward();  
    void goReverse();  
    void moveIn();  
    void moveOut();  
    void StopDrive();  
    bool scanIRFocused();  
    bool scanIRWide();  
    bool hasPyramid();  
    void readCompass();  
    short checkForCube();  
    void checkForPyramid();  
    void parallelFollow();  
    void scanField();  
    void intakeOn();  
    void placePyramid();  
    void setSpeed(bool speed);  
    void closeClaw();  
    void openClaw();  
    void moveArmIn();  
    void moveArmOut();  
    void moveLift(bool position);  
  
};
```

## MSE-bot.cpp

```
#include "MSE-Bot.h"
```

```
void MSEBot::init(){
    //Robot Initialization Function, Call this in setup() of the Arduino Sketch
    int initTime = millis();

    //Initialize Buses
    Serial.begin(9600);
    Serial7.begin(2400); //UART7 for pyramid IR sensor
    Serial8.begin(2400); //UART8 for front IR sensor
    Wire.begin(); //I2C1 for Compass and Accel
    Wire.setClock(400000);

    //Initialize GPIO Pins
    //pinMode(13, OUTPUT);
    pinMode(LR_ULTRASONIC_IN, OUTPUT);
    pinMode(LF_ULTRASONIC_IN, OUTPUT);
    pinMode(F_ULTRASONIC_IN, OUTPUT);
    pinMode(LR_ULTRASONIC_OUT, INPUT);
    pinMode(LF_ULTRASONIC_OUT, INPUT);
    pinMode(F_ULTRASONIC_OUT, INPUT);
    pinMode(LEFT_MOTOR, OUTPUT);
    pinMode(RIGHT_MOTOR, OUTPUT);
    pinMode(CUBE_INTAKE_ARM, OUTPUT);
    pinMode(CUBE_INTAKE_CLAW, OUTPUT);
    pinMode(PYR_INTAKE_LIFT, OUTPUT);
    pinMode(PYR_INTAKE_WHEELS, OUTPUT);
    pinMode(PYR_INTAKE_SW, INPUT);
    digitalWrite(PYR_INTAKE_SW, HIGH);
```

```

pinMode(LIFT_LIMIT_SW0, INPUT);
pinMode(LIFT_LIMIT_SW1, INPUT);
pinMode(START_SW, INPUT);

//Initialize Actuators
_leftMotor.attach(LEFT_MOTOR);
_rightMotor.attach(RIGHT_MOTOR);
_armMotor.attach(CUBE_INTAKE_ARM);
_clawMotor.attach(CUBE_INTAKE_CLAW);
_liftMotor.attach(PYR_INTAKE_LIFT);
_intakeMotor.attach(PYR_INTAKE_WHEELS);
_clawMotor.write(CUBE_INTAKE_OPEN); // open position

//Initialize Sensors
AccelMag.begin();

_liftMotor.writeMicroseconds(1600);
Serial.print("Initialized in "); Serial.print(initTime-millis()); Serial.println("ms! Starting operation on request!");

//Wait for start button to be pressed
while(!digitalRead(START_SW)) {}

Serial.println("Starting...");

//Initialize Intake Positions
//moveLift(0); // up position

```

```

/*
digitalWrite(13, HIGH);
digitalWrite(START_SW, HIGH);
*/
}

void MSEBot::PingUltra(){ // Ping ultrasonic sensors and record values in member variables
    digitalWrite(F_ULTRASONIC_IN, HIGH);
    delayMicroseconds(PING_MS);
    digitalWrite(F_ULTRASONIC_IN, LOW);
    _Fecho = pulseIn(F_ULTRASONIC_OUT, HIGH, 10000);
    if(_Fecho) _F_ultrasonic_dist = _Fecho;

    digitalWrite(LF_ULTRASONIC_IN, HIGH);
    delayMicroseconds(PING_MS);
    digitalWrite(LF_ULTRASONIC_IN, LOW);
    _LFecho = pulseIn(LF_ULTRASONIC_OUT, HIGH, 10000);
    if(_LFecho) _LF_ultrasonic_dist = _LFecho;

    digitalWrite(LR_ULTRASONIC_IN, HIGH);
    delayMicroseconds(PING_MS);
    digitalWrite(LR_ULTRASONIC_IN, LOW);
    _LRecho = pulseIn(LR_ULTRASONIC_OUT, HIGH, 10000);
    if(_LRecho) _LR_ultrasonic_dist = _LRecho;

    //Serial.print(" F: "); Serial.print(_F_ultrasonic_dist); Serial.print(" LF: ");
    Serial.print(_LF_ultrasonic_dist); Serial.print(" LR: "); Serial.println(_LR_ultrasonic_dist);
}

```

```
void MSEBot::PingFront(){

    digitalWrite(F_ULTRASONIC_IN, HIGH);

    delayMicroseconds(PING_MS);

    digitalWrite(F_ULTRASONIC_IN, LOW);

    _Fecho = pulseIn(F_ULTRASONIC_OUT, HIGH, 10000);

    if(_Fecho) _F_ultrasonic_dist = _Fecho;

}

}
```

```
void MSEBot::TurnOnAxisL(){ // Set speeds for turning left

    if(_speedMode){

        _leftMotor.writeMicroseconds(FORWARD_SPEED_FAST);

        _rightMotor.writeMicroseconds(VERSE_SPEED_FAST);

    }

    else{

        _leftMotor.writeMicroseconds(FORWARD_SPEED_SLOW);

        _rightMotor.writeMicroseconds(VERSE_SPEED_SLOW);

    }

}

}
```

```
void MSEBot::TurnOnAxisR(){ // Set speeds for turning right

    if(_speedMode){

        _leftMotor.writeMicroseconds(VERSE_SPEED_FAST);

        _rightMotor.writeMicroseconds(FORWARD_SPEED_FAST);

    }

    else{

        _leftMotor.writeMicroseconds(VERSE_SPEED_SLOW);

        _rightMotor.writeMicroseconds(FORWARD_SPEED_SLOW);

    }

}

}
```

```

void MSEBot::StopDrive(){ // Set speeds to stop
    _leftMotor.writeMicroseconds(STOP_VALUE);
    _rightMotor.writeMicroseconds(STOP_VALUE);
}

void MSEBot::goForward(){ // Set speeds to go forward
    if(_speedMode){
        _leftMotor.writeMicroseconds(FORWARD_SPEED_FAST);
        _rightMotor.writeMicroseconds(FORWARD_SPEED_FAST);
    }
    else{
        _leftMotor.writeMicroseconds(FORWARD_SPEED_SLOW);
        _rightMotor.writeMicroseconds(FORWARD_SPEED_SLOW);
    }
    //Serial.println("Forward");
}

void MSEBot::goReverse(){ // Set values to go in reverse
    if(_speedMode){
        _leftMotor.writeMicroseconds(VERSE_SPEED_FAST);
        _rightMotor.writeMicroseconds(VERSE_SPEED_FAST);
    }
    else{
        _leftMotor.writeMicroseconds(VERSE_SPEED_SLOW);
        _rightMotor.writeMicroseconds(VERSE_SPEED_SLOW);
    }
    //Serial.println("Reverse");
}

```

```

void MSEBot::moveIn(){ // Move in towards right side
    if(_speedMode){
        _leftMotor.writeMicroseconds(STOP_VALUE);
        _rightMotor.writeMicroseconds(FORWARD_SPEED_FAST);
    }
    else{
        _leftMotor.writeMicroseconds(STOP_VALUE);
        _rightMotor.writeMicroseconds(FORWARD_SPEED_SLOW);
    }
    //Serial.println("in");
}

```

```

void MSEBot::moveOut(){ // Move out away from right side
    if(_speedMode){
        _leftMotor.writeMicroseconds(FORWARD_SPEED_FAST);
        _rightMotor.writeMicroseconds(STOP_VALUE);
    }
    else{
        _leftMotor.writeMicroseconds(FORWARD_SPEED_SLOW);
        _rightMotor.writeMicroseconds(STOP_VALUE);
    }
    //Serial.println("out");
}

```

```

void MSEBot::moveArmOut() {
    Serial.println("Moving Arm");
    _armMotor.writeMicroseconds(1100);
    delay(5000);
    _armMotor.writeMicroseconds(STOP_VALUE);
}

```

```

void MSEBot::moveArmIn() {
    Serial.println("Moving Arm");
    _armMotor.writeMicroseconds(1800);
    delay(5000);
    _armMotor.writeMicroseconds(STOP_VALUE);
}

void MSEBot::moveLift(bool position) {
    if(position) { // if we want lift down
        if(!_LiftPosition) { // if lift up
            //while(!digitalRead(LIFT_LIMIT_SW1)) {
                _liftMotor.writeMicroseconds(REVERSE_SPEED_SLOW); // move down until hits outer
switch
                delay(2000);
                _liftMotor.writeMicroseconds(STOP_VALUE);
                _LiftPosition = 1;
            }
        }
    } else { // if we want lift retracted or up
        if(_LiftPosition) {
            //while(!digitalRead(LIFT_LIMIT_SW0)) {
                _liftMotor.writeMicroseconds(FORWARD_SPEED_SLOW); // move up until hits inner
switch
                delay(2000);
                _liftMotor.writeMicroseconds(STOP_VALUE);
                _LiftPosition = 0;
            }
        }
    }
}

```

```

bool MSEBot::scanIRFocused(){ // Scan using rear shrouded IR
    char value = Serial7.read();
    int idx1, idx2;
    if(_IRsw){
        idx1 = 0;
        idx2 = 1;
    }
    else{
        idx1 = 2;
        idx2 = 3;
    }
    if(value == _IRValues[idx1] || value == _IRValues[idx2]){
        _LastIRTTime = millis();
        return 1;
    }
    else{
        return 0;
    }
}

```

```

bool MSEBot::scanIRWide(){ // Scan using front unshrouded IR
    char value = Serial8.read();
    int idx1, idx2;
    if(_IRsw){
        idx1 = 0;
        idx2 = 1;
    }
    else{
        idx1 = 2;

```

```

idx2 = 3;
}

if(value == _IRValues[idx1] || value == _IRValues[idx2]){
    _LastIRTTime = millis();
    return 1;
}
else{
    return 0;
}
}

bool MSEBot::hasPyramid(){ // Read if we have pyramid using limit switch
    return digitalRead(PYR_INTAKE_SW);
}

void MSEBot::readCompass(){ // Obtain magnetometer readings
    AccelMag.read();

    _compassMagnitude = sqrt(pow(AccelMag.magData.x, 2)+pow(AccelMag.magData.y, 2)+pow(AccelMag.magData.z, 2));
    //_compassHeading = atan2(AccelMag.magnetic.x, AccelMag.magnetic.y)*180/3.14159;
}

short MSEBot::checkForCube(){ // Compare magnetometer readings to see how close cube is
    readCompass();

    if(_compassMagnitude > CUBE_MAG_ACCURATE_THRESH) return 1; //return 1 if the mag field exceeds the threshold of the cube being held in the claw

    if(_compassMagnitude > CUBE_MAG_GEN_THRESH) return 2; //return 2 if the mag field exceeds the threshold of the cube being in the vicinity of the claw

    else return 0; //return 0 if the cube is not near by
}

```

```

void MSEBot::checkForPyramid(){ // Checks previous instance of IR reading to see if robot is
still pointing towards pyr

    while(_LastIRTime - millis() > IR_TIME_TOLERANCE) {
        scanIRFocused();
        TurnOnAxisL();
    }
}

void MSEBot::findWall(){

    PingUltra();
    goForward();
    while(_F_ultrasonic_dist > TURN_THRESHOLD) {
        PingUltra();
    }
    StopDrive();

}

void MSEBot::parallelFollow(){ // Follow walls at a set distace, parallel to wall, turn as necessary

    PingUltra();

    if(_F_ultrasonic_dist < TURN_THRESHOLD) {
        StopDrive();
        /*
        while(_LR_ultrasonic_dist > WALL_TARGET_DIST){ //Turn until Front Ultrasonic is
measuring a large distance
            TurnOnAxisL();
            PingUltra();
        }
        */
    }
}

```

```

TurnOnAxisL();
delay(100);
}

unsigned int parallel = abs(_LF_ultrasonic_dist - _LR_ultrasonic_dist);
/* unsigned int distance = abs(_LF_ultrasonic_dist - WALL_TARGET_DIST);

if(parallel < PARALLEL_TOLERANCE && distance < WALL_TARGET_TOLERANCE){
//Everything is fine and dandy
    goForward();
}
else if(_LF_ultrasonic_dist > PARALLEL_TOLERANCE + _LR_ultrasonic_dist){ //Distance is
ok, Robot is not parallel
    moveIn();
}
else if(_LF_ultrasonic_dist + PARALLEL_TOLERANCE < _LR_ultrasonic_dist){ //Distance is
ok, Robot is not parallel
    moveOut();
}
*/
if(parallel < PARALLEL_TOLERANCE){
    goForward();
}
else if(_LF_ultrasonic_dist > _LR_ultrasonic_dist){
    moveIn();
}
else{
    moveOut();
}
}

```

```

void MSEBot::scanField() { // Swerve and scan field for pyramid as robot drives across it back
andforth

    TurnOnAxisL();

    delay(300);

    goForward();

    delay(300);

    /*

        if(scanIRWide()) {

            StopDrive();

            return;

        }

        PingUltra();

        if(_F_ultrasonic_dist < 3 * TURN_THRESHOLD) { // turn at walls

            if(_TurnCount) {

                TurnOnAxisR();

                unsigned long triggerTime= millis(); // turn 160 degrees

                while(millis() - triggerTime < 3000) {

                    if(scanIRWide()) {

                        StopDrive();

                        return;

                    }

                    if(scanIRWide()) {

                        StopDrive();

                        return;

                    }

                    if(scanIRWide()) {

                        StopDrive();

                        return;

                    }

                }

            }

        }

    }

}

```

```

    }
}

_TurnCount = 0;
}

else {
    TurnOnAxisL();
    unsigned long triggerTime= millis(); // turn 160 degrees
    while(millis() - triggerTime < 3000) {
        if(scanIRWide()) {
            StopDrive();
            return;
        }
        if(scanIRWide()) {
            StopDrive();
            return;
        }
        if(scanIRWide()) {
            StopDrive();
            return;
        }
    }
    _TurnCount = 0;
}

if(scanIRWide()) {
    StopDrive();
    return;
}

```

```

if((int)(millis() / SWERVE_DELAY) % 2) { // alternate between swerving left and right in real
time

    _leftMotor.writeMicroseconds(FORWARD_SPEED_FAST);
    _rightMotor.writeMicroseconds(FORWARD_SPEED_FAST);
}

else {

    _leftMotor.writeMicroseconds(FORWARD_SPEED_FAST);
    _rightMotor.writeMicroseconds(FORWARD_SPEED_FAST);
}

if(scanIRWide()) {

    StopDrive();
    return;
}

*/
}

void MSEBot::intakeOn() { // get intake in position for retrieving pyramid

    _intakeMotor.writeMicroseconds(VERSE_SPEED_FAST);

    moveLift(1);

}

void MSEBot::placePyramid() { // routine for putting cube in pyramid

    _intakeMotor.writeMicroseconds(STOP_VALUE);

    delay(1000);

    moveLift(0);

    openClaw(); // drop cube into funnel

    delay(1000);

    goReverse(); // back up a bit

    delay(2000); // test this value so cube under pyr
}

```

```

StopDrive();
delay(1000);
moveLift(1);
delay(1000);
goReverse();

_intakeMotor.writeMicroseconds(FORWARD_SPEED_FAST); // push pyramid back out with
cube underneath

delay(2000);
StopDrive();
}

```

```
void MSEBot::setSpeed(bool speed){
```

```
    if(speed){ //speed == 1: FAST
```

```
        _speedMode = 1;
```

```
}
```

```
    else{ //speed == 0: SLOW
```

```
        _speedMode = 0;
```

```
}
```

```
}
```

```
void MSEBot::closeClaw(){ // close claw
```

```
    _clawMotor.write(CUBE_INTAKE_CLOSE);
```

```
}
```

```
void MSEBot::openClaw(){ // OPEN CLAW
```

```
    _clawMotor.write(CUBE_INTAKE_OPEN);
```

```
}
```

## Main.ino

```
#include "MSE-bot.h"
```

```
MSEBot Robot;
```

```
void setup() {
```

```
    Robot.init();
```

```
    Robot.moveLift(1);
```

```
    Robot.moveArmOut();
```

```
    Robot.openClaw();
```

```
    Robot.findWall();
```

```
}
```

```
void loop() {
```

```
    bool hasCube= 0;
```

```
    bool hasPyr= 0;
```

```
    while (!hasCube) { //Loop until robot finds the cube
```

```
        Robot.parallelFollow(); // Follows walls of arena and scans for cube
```

```
        Robot.readCompass();
```

```
        Serial.println(Robot._compassMagnitude);
```

```
        if (Robot._compassMagnitude> 5000) { // Finds cube within gripping distance
```

```
            delay(500);
```

```
            Robot.StopDrive();
```

```
            delay(1000);
```

```
            Robot.closeClaw(); // Pulls cube into robot
```

```
            delay(500);
```

```
            Robot.moveArmIn();
```

```
            hasCube = 1;
```

```
            break;
```

```
}
```

```

}

while(!hasPyr){

bool focused, wide;

//Loop until finds the pyramid

while(!wide || !focused) {

    wide = Robot.scanIRWide();

    focused = Robot.scanIRFocused();

    Robot.TurnOnAxisL();

    delay(300);

    Robot.goForward();

    delay(300);

    if(focused){

        Robot.intakeOn();

        delay(100);

        Robot.goForward();

        delay(500);

        Robot.StopDrive();

        Robot.moveLift(0);

        Robot.openClaw();

    }

    if(wide){

        Robot.TurnOnAxisL();

        delay(7500);

        Robot.intakeOn();

        delay(100);

        Robot.goForward();

        delay(500);

    }

}

```

```

Robot.StopDrive();
Robot.moveLift(0);
delay(100);
Robot.openClaw();

}

}

Robot.StopDrive();

// Slowly pinpoint pyramid with shrouded IR
while(!Robot.scanIRFocused()) {
    if(Robot.scanIRFocused()){
        break;
    }
    Robot.TurnOnAxisL();
    if(Robot.scanIRFocused()){
        break;
    }
    if(Robot.scanIRFocused()){
        break;
    }
}

Robot.StopDrive();
Robot.intakeOn();

```

```
delay(500);

// Push into pyramid with intake until it's in the intake
while(!Robot.hasPyramid()){

    Robot.goForward();
    delay(10000);
    break;
}

Robot.placePyramid();

}

/*
}
```

## Product Evaluation

The performance of the prototype and the product can not be equated as many of the factors that negatively affected the performance of the prototype will not be present in the actual product. This is because in spite of our best efforts several aspects of the product were not accurately replicated and represented on the prototype. For example the motors used on the prototype are much weaker than those used of the product. The motors are also bulkier which limit the space on the robot. The wheels on the prototype are also poor representations of those that were selected for the product. The 4" rubber wheels being used on the product are soft and offer much more traction than the VEX wheels used on the prototype, even with the added traction from the racket grip tape.

Most of the components of the prototype functioned individually. However, our prototype had difficulty locating the correct pyramid. Testing various components and mechanisms on the robot showed that the mechanical design of the prototype was functional. Unfortunately, problems with the code and the IR sensors used resulted in inconsistent results when the robot was navigating the course.

Overall, spending more time debugging and refining the code would allow the prototype to function as intended.

The final product designed meets most but unfortunately not all of the design requirements specified by the group. The rubber wheels would allow for effective navigation of the field, however the large turn radius due to the larger size of the robot and the 4 wheel drive results in wide turns around the corners of the course which would cause the tesseract to be missed if it was placed close to a corner. The cube is picked up and held on to firmly by the claws designed due to the addition of the grip material as required by the specifications stated at the beginning of the project. Finally, the intake used for the pyramid was proven to be able to pick up the pyramid from different angles as long as the bottom of the forklift intake is at a 10 degree incline and the intake is free to open up more when the pyramid is being picked up from its wide side. The addition of the elastics to the intake also allows the wheels to apply enough force to the pyramid to intake it when it is being picked up from its thinner side. Finally, the product might take longer than three minutes to complete the task depending on the location of the pyramid due to the fact that the IR sensors have around a 2 ft effective range so the robot must navigate the field looking for them.

To ensure that the product is consistently meeting all requirements when in the field, the code must be refined to include safety checks in case a step is not completed properly. Aspects of the code must also be debugged to ensure that locating the correct pyramid is done consistently, no matter the location of the pyramid.

## Cost of External Parts sourced for Prototype

Name	Sourced from	Cost	Quantity	Total Cost	Receipt/ Proof of sale
Microcontroller (STM32 F429ZI)	ST Microelectronics	25	1	25	Contacted ST Sales office
Tennis Racket Grips (Head Hydrosorb)	Sport Check	11.99	1	11.99	Receipt
Tennis Racket Overgrip (Wilson Pro)	Sport Check	7.99	1	7.99	Receipt
1/2 inch aluminum L braket	Home Depot	6.48	1	6.48	Screenshot
1 inch aluminum L braket	Home Depot	10.58	1	10.58	Screenshot
Magenetometer	DigiKey	20.33	1	20.33	Receipt
Nuts and bolts	Home Depot	10.48	1	10.48	Receipt
<b>Total</b>				<b>92.85</b>	

Figure 51: Prototype Cost Breakdown

## Sourcing of External Parts

Part Category	Chosen?	Part	Part Manufacturer	Currently on hand	Source	SKU	Value	Price Paid	CAD/USD	Specs:
Main Controller	1x	STM32 F103ZI	ST Microelectronics	yes	ST Sales rep	497-16280-ND	\$25.00	\$0.00	USD	ARM®@32-bit Cortex®@M4 CPU with FPU, 180 MHz
	0x	STM32 F446RE	ST Microelectronics	yes	ST Sales rep		15	0		ARM®@32-bit Cortex®@M4 CPU with FPU, 180 MHz
	0x	Arduino Mega 2560	Atmel	yes			20			16 MHz
Secondary Controller?	0x	MSE Duino	Atmel	yes						16 MHz
Drive Motors	5x	Vex Integrated Encoder	...	no	Digikey	1528-1062-ND	14	14	USD	1.645" x 1.645" - 330mA, 12V, 1.8deg stepper, Driver needed
Servo Motors	0x	Vex Servo	Adafruit Industries LLC	...						
	1x		Adafruit	...						
Magnetic Sensors	0x	STEVAL-MKI181V1	ST Microelectronics	in transit	ST Sales rep	1528-1173-ND	?	0		
	1x	LSN1303	Adafruit Industries LLC	yes	Digikey	1528-1173-ND	20.53	20.53	CAD	
	0x	MAG3110	Sparkfun	yes	Digikey	1568-1030-ND	20.53	20.53	cad	
Photosensitive Sensors	2x	The IR one they give us		yes	raish		0	0		
Ultrasonic Sensors	3x	The VEX one they give us		yes	raish		0	0		
Arduino WiFi shield	0x	EXP866		no	sparkfun	WRL-13678		7		
Misc Hardware	...	1.5" Standoffs		no						
Suction Cup	0x			no						
Suction Cup Vacuum Pump	0x			no			3			
Sensor Moduled	0x					1528-1173-ND	15		USD	

Figure 52: External Parts Sourcing

## Receipts

 <p><b>701 Brooks Ave South, PO Box 677 Thief River Falls, MN 56701-0677</b></p>		<p><b>www.digikey.ca Orders 1-800-344-4539 Fax 218-681-3380</b></p>		<p><b>Invoice # 61915590 Completed Salesorder CANADIAN \$</b></p>																																
<b>Bill To:</b> ANDREW RANDELL 1140 WESTERN RD 351B LONDON HALL LONDON ON N6G0A3 CANADA		<b>Ship To:</b> ANDREW RANDELL 1140 WESTERN RD 351B LONDON HALL LONDON ON N6G0A3 CANADA		<b>Salesorder / Packlist:</b> Customer: 54121411 / PL1 Payment Terms: 10773995 Shipping Method: AMER EXPRESS / 2002 Tracking #: XFLD Order Source: 430759290844 Order Date: INTERNET 16-Mar-2018 Invoice Date: 19-Mar-2018 Ship Date: 19-Mar-2018 Document Date: 19-Mar-2018/AUTO																																
<b>Buyer:</b> ANDREW RANDELL 534 SANDBANKS CRESCENT WATERLOO ON N2V2K2 CANADA		<b>Ship From:</b> DIGI-KEY 701 BROOKS AVE. SOUTH P.O. BOX 677 THIEF RIVER FALLS MN 56701-0677																																		
<table border="1"> <thead> <tr> <th>Line Item</th> <th>Ordered</th> <th>Cancelled</th> <th>Shipped</th> <th>Item Number/ Description</th> <th>Unit Price CANADIAN \$</th> <th>Amount CANADIAN \$</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td> <b>PART: 1528-1173-ND</b>            COO : UNITED STATES            LEAD FREE ROHS COMP REACH UNKNOWN  <b>Mercury:</b> Cert on File. For more information contact RoHS@DigiKey.com         </td> <td>DESC: BOARD ACCEL/COMPASS 3AXIS LSM303 ECCN: EAR99 HTSUS: 8473.30.1180</td> <td>20.53000</td> <td>20.53 T</td> </tr> <tr> <td>2</td> <td>1</td> <td>0</td> <td>1</td> <td> <b>PART: 1568-1030-ND</b>            COO : UNITED STATES            LEAD FREE ROHS COMP REACH UNKNOWN  <b>Mercury:</b> Cert on File. For more information contact RoHS@DigiKey.com         </td> <td>DESC: EVAL BOARD FOR MAG3110 ECCN: EAR99 HTSUS: 8473.30.1180</td> <td>20.53000</td> <td>20.53 T</td> </tr> <tr> <td>3</td> <td>1</td> <td>0</td> <td>1</td> <td> <b>PART: 1528-1223-ND</b>            COO : CHINA            LEAD FREE ROHS COMP REACH UNKNOWN  <b>Mercury:</b> Cert on File. For more information contact RoHS@DigiKey.com         </td> <td>DESC: ESP8266 HUZZAH BREAKOUT BOARD ECCN: SA992C HTSUS: 8473.30.1180</td> <td>13.66000</td> <td>13.66 T</td> </tr> </tbody> </table>						Line Item	Ordered	Cancelled	Shipped	Item Number/ Description	Unit Price CANADIAN \$	Amount CANADIAN \$	1	1	0	1	<b>PART: 1528-1173-ND</b> COO : UNITED STATES LEAD FREE ROHS COMP REACH UNKNOWN <b>Mercury:</b> Cert on File. For more information contact RoHS@DigiKey.com	DESC: BOARD ACCEL/COMPASS 3AXIS LSM303 ECCN: EAR99 HTSUS: 8473.30.1180	20.53000	20.53 T	2	1	0	1	<b>PART: 1568-1030-ND</b> COO : UNITED STATES LEAD FREE ROHS COMP REACH UNKNOWN <b>Mercury:</b> Cert on File. For more information contact RoHS@DigiKey.com	DESC: EVAL BOARD FOR MAG3110 ECCN: EAR99 HTSUS: 8473.30.1180	20.53000	20.53 T	3	1	0	1	<b>PART: 1528-1223-ND</b> COO : CHINA LEAD FREE ROHS COMP REACH UNKNOWN <b>Mercury:</b> Cert on File. For more information contact RoHS@DigiKey.com	DESC: ESP8266 HUZZAH BREAKOUT BOARD ECCN: SA992C HTSUS: 8473.30.1180	13.66000	13.66 T
Line Item	Ordered	Cancelled	Shipped	Item Number/ Description	Unit Price CANADIAN \$	Amount CANADIAN \$																														
1	1	0	1	<b>PART: 1528-1173-ND</b> COO : UNITED STATES LEAD FREE ROHS COMP REACH UNKNOWN <b>Mercury:</b> Cert on File. For more information contact RoHS@DigiKey.com	DESC: BOARD ACCEL/COMPASS 3AXIS LSM303 ECCN: EAR99 HTSUS: 8473.30.1180	20.53000	20.53 T																													
2	1	0	1	<b>PART: 1568-1030-ND</b> COO : UNITED STATES LEAD FREE ROHS COMP REACH UNKNOWN <b>Mercury:</b> Cert on File. For more information contact RoHS@DigiKey.com	DESC: EVAL BOARD FOR MAG3110 ECCN: EAR99 HTSUS: 8473.30.1180	20.53000	20.53 T																													
3	1	0	1	<b>PART: 1528-1223-ND</b> COO : CHINA LEAD FREE ROHS COMP REACH UNKNOWN <b>Mercury:</b> Cert on File. For more information contact RoHS@DigiKey.com	DESC: ESP8266 HUZZAH BREAKOUT BOARD ECCN: SA992C HTSUS: 8473.30.1180	13.66000	13.66 T																													
Total Invoiced 54.72 Shipping charges applied 8.00 ** Charges subtotal ** 62.72 HST on taxable amount: 54.72 Tax Rate: 13.00 7.11 Total charged to credit card 69.83 <b>CANADIAN \$</b> <small>Incoterm 2010: DDP LONDON, ON, CANADA T indicates taxable amounts</small>																																				
<small>* One or more items on this order are controlled for export.            * NO EEI 30,36            * These items are controlled by the U.S. Government and authorized for export only to the country of ultimate destination for use by the ultimate consignee or end-user(s) herein identified. They may not be resold, transferred, or otherwise disposed of, to any other country or to any person other than the authorized ultimate consignee or end-user(s), either in their original form or after being incorporated into other items, without first obtaining approval from the U.S. Government or as otherwise authorized by U.S. law and regulations.</small>																																				
<b>Box(es) :</b> <table border="1"> <tr> <td>1</td> <td>XIFR</td> <td>430759290844</td> <td>0.17 kg / 6 oz</td> <td></td> <td>1 1568-1030-ND</td> </tr> <tr> <td></td> <td></td> <td>1 1528-1173-ND</td> <td></td> <td></td> <td>1 1528-1223-ND</td> </tr> </table>						1	XIFR	430759290844	0.17 kg / 6 oz		1 1568-1030-ND			1 1528-1173-ND			1 1528-1223-ND																			
1	XIFR	430759290844	0.17 kg / 6 oz		1 1568-1030-ND																															
		1 1528-1173-ND			1 1528-1223-ND																															
<small>Claims for pricing errors, shortages, and defective product must be reported within 30 days of invoice date.  <b>Contact Customer Service at 1-800-858-3616</b>            All transactions with Digi-Key Electronics, including its subsidiaries and/or affiliates, are subject to Digi-Key's Terms of Use and Conditions of Order, available at <a href="http://www.digikey.ca">www.digikey.ca</a>.            DIGI-KEY NRI #: 895173490 DIGI-KEY GST/HST#: 895173490RT</small>																																				

Page: 1 of 2

 <p><b>701 Brooks Ave South, PO Box 677 Thief River Falls, MN 56701-0677</b></p>		<p><b>www.digikey.ca Orders 1-800-344-4539 Fax 218-681-3380</b></p>		<p><b>Invoice # 61915590 Completed Salesorder CANADIAN \$</b></p>	
---	--	---	--	---	--

General - WEB ORDER ID: 187457872  
 CUSTOMER 10773995 INDICATED THAT PRODUCT WILL NOT BE EXPORTED OUTSIDE OF CANADA. AOFX

**CERTIFICATE OF COMPLIANCE:**  
 The Digi-Key components included in the above shipment are genuine components and were provided by the applicable manufacturer to Digi-Key. Test reports (chemical, physical, electrical, etc. together with results of any tests performed by the manufacturer) are on file (either here or in the plant of the manufacturer) and will be made available upon request. These components have been handled in accordance with the requirements of applicable quality standards. This certification is valid only to the original customer and is not transferable. Contact Customer Service at 800-858-3616 if you have any questions.

*Kim Gilbert*

Kim Gilbert, Director, Customer Service

Figure 53: Digikey Receipt

**SPORTCHEK**

MASONVILLE PLACE, UNIT Y008, 1680 RICHMOND  
STREET NORTH  
LONDON, Ontario, N6G 3Y9  
1-519-645-0350

SALE

Date: 2018/03/28 12:03  
Cashier: \*\*\*\*\*4683 NATASHA

726423696698	\$11.99 H
HEAD HYDROSORB PRO REPLACEMENT GRIP BLACK	
99 NO COLOR N/S	
Qty: 1 Price: \$11.99	

726423696698	\$11.99 H
HEAD HYDROSORB PRO REPLACEMENT GRIP BLACK	
99 NO COLOR N/S	
Qty: 1 Price: \$11.99	

887768146719	\$7.99 H
WILSON PRO OVERGRIP SILVER	
99 NO COLOR N/S	
Qty: 1 Price: \$7.99	

887768146719	\$7.99 H
WILSON PRO OVERGRIP SILVER	
99 NO COLOR N/S	
Qty: 1 Price: \$7.99	

Sub total	\$39.96
ONFedHST 5.000%	\$2.00
ONProvHST 8.000%	\$3.20

Total	\$45.16
-------	---------

Anex	\$45.16
------	---------

TYPE	PURCHASE
------	----------

ACCT	AMEX
AMOUNT	\$45.16

CARD NUMBER	*****#2002
DATE/TIME	28 Mar 2018 12:03:58
REFERENCE #	663866380010010011 T
TERMINAL #	43570102
AUTH #	507048

00 APPROVED - THANK YOU 025

\*Important - retain this copy for your records

GST/HST #: 869618785

\*\*\*\*\*  
TELL US HOW WE DID TODAY!  
Take a short Survey & enter our monthly  
draw for 1 of 2 \$500 gift cards! Go to

Figure 53: SportsChek Receipt


**THE HOME DEPOT**  
 More saving.  
 More doing.<sup>SM</sup>

600 Fanshawe Park E., N. London, ONT  
 KATHLEEN FAULKNER 519-850-5900  
 7009 00057 87015 03/04/18 12:45 PM  
 SELF CHECK OUT  
 771878737113 PAPER 1/16X1X <A> 10.48  
 771878737670 PAPER 1/16X3/ <A> 12.48  
 SUBTOTAL 22.94  
 GST/HST 2.98  
 TOTAL \$25.92  
 XXXXXXXXXXXXXXXXX9050 DEBIT C40\$ 25.92  
 AUTH CODE 153012  
 Chip Read Verified By PIN  
 AED 00000002771010 Interac  
 TWR 02800085000  
 TAD 004855400410000040094000400000000000000  
 000000  
 TSI FB00  
 ARC 00



7009 57 87015 03/04/2018 0305

13% HST R135772911

RETURN POLICY DEFINITIONS  
 POLICY ID: DAYS POLICY EXPIRES ON  
 A 90 02/07/2018  
 THE HOME DEPOT RESERVES THE RIGHT TO  
 LIMIT / DENY RETURNS. PLEASE SEE THE  
 RETURN POLICY SIGN IN STORES FOR  
 DETAILS.

KEEP YOUR RECEIPT FOR FASTER RETURNS  
 SHOP ONLINE AT [WWW.HOMEDEPOT.CA](http://WWW.HOMEDEPOT.CA)  
 More saving. More Doing.

\*\*\*\*\*  
 ENTER FOR A CHANCE  
 TO WIN A \$3,000  
 HOME DEPOT GIFT CARD!

Tell us about your store visit!  
 Complete our short survey and  
 enter for a chance to win at:  
[www.homedepot.com/survey](http://www.homedepot.com/survey)

You will need the following to enter  
on-line:

User ID:  
 2PX2 181328 174376  
 Password:  
 18203 174319

Entries must be completed within 14 days  
 of purchase. Entrants must be 18 or  
 older to enter. See complete rules on  
 website. No purchase necessary.

(Le sondage est également offert en  
français sur le Web.)

Figure 53: Home Depot Receipt