End Effector for Robotic Strawberry Picker
Final Design Review

Sponsored by: California Strawberry Commission

Cory Frederickson
cory.frederickson@sbcglobal.net

James Jeffery
jimmyjeffery@sbcglobal.net

Mick Cuffe
mcuffe1@hotmail.com

Mechanical Engineering Department

California Polytechnic State University

San Luis Obispo

2018

Table of Contents

List of Figures

List of Tables

Executive Summary

In this report, we have outlined the background of the problem and need for a solution to an automated form of strawberry harvesting. The report includes our research findings, defines the scope and objectives for this project, and documents our complete design process. Also included is our final, completed prototype, and a description of the manufacturing, design verification and testing process. Also included is our conclusions and recommendations for further improvement on future iterations.

1. Introduction

The purpose of this project is to design an end effector for an automated strawberry harvesting system that will not damage the berry as well as effectively remove the berry from the peduncle. Due to the increase in the demand of fruit along with a decrease in available cheap labor, there is a need for an automated solution to fruit harvesting. Key stakeholders in this project include strawberry farmers and consumers, the manufacturer, and the sponsor of this project, the California Strawberry Commission. The team working on this project consists of three seniors from California Polytechnic State University, San Luis Obispo all majoring in Mechanical Engineering.

2. Background

Current automated strawberry pickers show promise for the future, but are not yet feasible for mass use. There are several products on the market that effectively pick strawberries in a controlled environment, but are not yet reliable enough to be commercialized (Lin). The major issue with these current products is the low yield rate when compared with human labor. While humans will pick virtually all ripe strawberries in the field, automated systems have issues successfully harvesting 75% of ripe berries (Lin). Automated systems may also damage the strawberries, making them unacceptable to consumers. In addition, many current products do not address the issue of packing the strawberries. This means that human labor is still necessary to properly pack the strawberries in the manner in which they will arrive in stores. These issues make the current products not economically advantageous over human labor. A successful product will mitigate these issues, successfully picking enough strawberries to become economically beneficial.

While hydroponic growing is becoming more common, the current prevailing method for strawberry growing is in-ground. These two different growing methods lend themselves to different methods of picking the berry. In-ground systems typically have the strawberry lying on the ground with the peduncle being oriented sideways. Hydroponic systems have the strawberries hanging down, with the peduncle extending vertically above the strawberry. A strawberry picker can be optimized for one growing method or the other, but there is also potential for a system with the ability to function for both. Below is a discussion of some of the currently existing automated systems.

- **The Pitzer Wheel**
  - **Description:** The Pitzer Wheel utilizes six silicon claws and a rotating motion to pick strawberries. It picks one strawberry, then rotates to utilize the next claw to pick the next berry. Once a claw rotates to the vertical position, it places a berry into a cup that is transferred to a central location. From here, each berry is inspected before being placed into a pack.
  - **Key Attributes:** Farmers will not have to change the way they currently grow strawberries
  - **Growing System:** In-ground

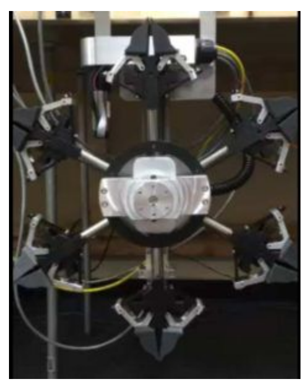


Figure 1. The Pitzer Wheel

- **Agrobot SW6010**
  - **Description:** Has 16 robotic arms that scan the plants while moving slowly. It picks strawberries and pulls them up through brushes to a conveyor belt that moves the berry to the user. User sits on top of machine as it moves
  - **Key Attributes:** End effector can switch positions to be used to retrieve strawberries that are laying on the ground and hanging, not completely autonomous (user required)
  - **Growing System:** In-ground & Hydroponic

Figure 2. Agrobot SW6010

- **Octinion**
  - **Description:** Utilizes 3d vision for detection of ripe strawberries, grabs strawberry with soft touch gripper. Simulates actual picking of a strawberry without cutting or burning stem. Has capacity of picking one strawberry every 3 seconds
  - **Key Attributes:** Simulates human picking, quality monitoring to sort strawberries according to size and quality, handles strawberries and places them in their final packaging.
  - **Growing System:** Hydroponic

Figure 3. Octinion

- **Phil Brown/DBR Vacuum System**
  - **Description:** Utilizes a vacuum to pluck apples off of trees. Identifies ripe apples with imaging and then moves the arm to the desired location before sucking apple in with vacuum
  - **Key Attributes:** Able to decelerate apple fast enough to avoid damage.
  - **Growing System:** N/A (apples)

Figure 4. DBR Vacuum System

- **Wall-ye Vineyard Picker**
  - **Description**: Utilizes AI and tracking technologies to drive around vineyards and find ripe grapes. Able to trim the vines and pick grapes.
  - **Key Attributes:** Highly mobile
  - **Growing System:** N/A (grapes)

Figure 5. Wall-ye Vineyard Picker

Table 1. Patent Search Results

Patent Name & Number	Date	Description
Mechanical picker for strawberries - US3698171A	10/17/1972	• Self-propelled strawberry harvester. Includes a series of picking fingers that move along the ground to scoop under and lift strawberry clusters to remove them from the plant.
Automated Selective Harvesting of Crops - US9480202B2	11/1/2016	• Device for selectively harvesting crops on a plant, which can include a picking apparatus that may be rotated around a central axis. Picking apparatus will open and close to pick and hold berries.
Fruit Harvesting Robot Hand - US4663925A	3/12/1987	• Fruit harvesting robot with fruit detecting sensors arranged around an opening of a cylindrical compartment for taking in fruit.
Machine for Automatically Harvesting Fruit Cultivated in Rows - US20110252760A1	10/20/2011	• Machine that automatically harvests fruit cultivated in rows. Consists of an autonomous vehicle that drives atop several crop rows. Utilizes independent robot arms to pick strawberries. End effector has a blade to remove the strawberries via cutting.
Fully-automatic Kiwi fruit picking end effector - CN103404307A	11/27/2013	• Consists of a holding device, a shearing device, a detecting device and a machine case housing. Holds the fruit in place before cutting to remove the fruit from the stem.
End effector for fruit picking robot -	12/16/2015	• Composed of a pair of scissors, a sliding rail assembly, an eccentric wheel sliding block

CN105144994A		assembly and a wireless control assembly. Extends mechanical arm and closes scissors to shear off the stem of fruit.

There have many efforts to improve automated fruit harvesting in an attempt to mitigate the issues brought about by labor shortages. This has resulted in many new products, such as the ones mentioned in Table 1, and subsequent studies on these products. One study done on a Japanese Strawberry Harvester found that the system was able to successfully harvest 67% of strawberries and properly remove the peduncles from 88% of those harvested berries (Yamamoto). Although this performance may be an improvement over past robotic harvesters, it is still not effective enough to replace human labor. In addition to strawberry harvesting, there have been numerous studies on the automated harvesting of other fruits, such as apples. One study on an apple-picking end effector meant to replicate a human hand found that 95 out of 100 apples were successfully picked. The average localization and picking time for each apple was found to be between 1.2 and 6.8 seconds (Davidson).

A large problem found in current solutions is maturity recognition to determine pickable strawberries. The current success rate in fruit color recognition is close to 83% as found in a study conducted by Yamamoto in 2013. The vision success rate maxed out at 93.6% and the technology had a good correlation with human assessment with a reliability factor of $R^2 = 0.956$. In assessing peduncle position/orientation and maturity level the success rate dropped some to 83%.

We plan on using the existing products and extensive research on this topic to help guide us in our design of an end effector. Incorporating effective ideas used in current products and learning from their drawbacks will help aid us in producing an effective solution to our problem.

3. Objectives

3.1. Problem Statement:

Increasing strawberry consumption combined with a decrease in available cheap labor has created a need for a more effective way to pick strawberries. Farmers are currently looking for an automated solution but current prototype systems have issues collecting the strawberries. Our goal is to design an end effector to pick strawberries without damage and properly separate the berry from the stem. An effective solution to this problem could help save money for both farmers and consumers as well as significantly impact the Agricultural Technology industry.

3.2. Boundary Diagram

This sketch represents our Boundary Diagram, which displays the scope of our project. As can be seen in the sketch, there is a strawberry being picked by a robotic harvester. The harvester has an imaging system mounted on it and includes an arm with multiple axes of movement. Additionally, a box of neatly packaged strawberries is next to the harvester. The scope of our project only includes the end effector of the picking system, hence the outline around just the berry and the end effector.

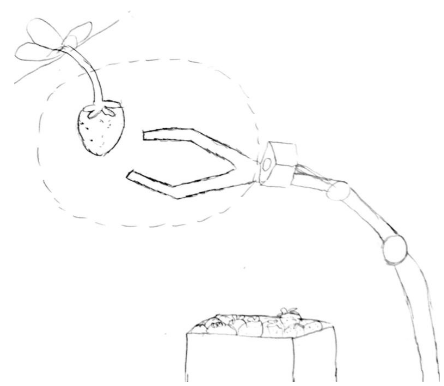


Figure 6. Boundary Diagram

3.3. Customer wants/needs

A full list of our customer needs and wants for this project has been included in Appendix B, but we will discuss the key ones here. Among customer needs, it is necessary that our end effector will: pick berries without damaging them; function in an automated system; have on/off capability; and be completed by December of 2018. Some of our customer wants are: keeping a $1,000 budget; plug-and-play capability with multiple systems; incorporating a control system (such as a pressure or force feedback).

3.4. Quality Function Deployment

For our Quality Function Deployment, we utilized a "House of Quality" to help us better understand the problem. In order to create our House of Quality, we first determined our customers, or the various individuals who will be impacted by our product. With these individuals in mind, we determined different "needs", or desired qualities for our product. We then ranked each one of these qualities by importance to each one of our customers. After doing so, we took these customer needs and created a new list, this time of engineering specifications. By placing these needs on intersecting axes to our customer needs, we were able to mark how closely related each specification was with each customer need. In doing so, we were able to see how our engineering specifications impacted the desired characteristics of customers.

Included below is a table of the key specifications of this project that we determined using our House of Quality. Their requirements or targets are listed beside them, as well as a tolerance. We have ranked the likelihood of failing to meet these targets with either a high, medium, or low level of risk (H, M, L). The column labeled "Compliance" indicates the method we will use to determine if the specification target has been met. The letters have the following meanings: A = Analysis, T = Testing, S = Similar Design, and I = Inspection.

Table 2. Engineering Specifications

Specification #	Description	Requirement / Target (Units)	Tolerance	Risk	Compliance
1	Cost	$1,000	Max	M	A
2	Weight	5 lbs	± 2.5 lbs	M	A, T
3	Max Force Applied to Stem	10 lb	Max	L	A, T
4	Power	500 W	Min	L	T
5	% Berries Damaged	10%	Max	H	T, I
6	% Successful Pick	95%	Min	H	T
7	Time-to-Pick	1.5 sec	Max	M	A, T

In table 2 we list the method we will use to determine whether or not the specification requirement has been met. We will use the following techniques specifically to measure each of the requirements:

- Cost - Adding the cost of each component we purchase during our design process
- Weight - A scale
- Max Force Applied - Load cells
- Power - Omni-Watt meter
- % Berries Damaged - Visual analysis after berry acquisition
- % Successful Picks - Counting how many picks during a test run were successful
- Time-to-Pick - Calculations as well as test runs

4. Concept Design Development

4.1 Ideation Process

The first step in our concept design development was the ideation phase. In order to simplify the problem and make it easier to find a solution, we broke down the problem into separate elements using functional decomposition. Functional decomposition involves taking the main function of a product and separating it into distinct sub-functions.

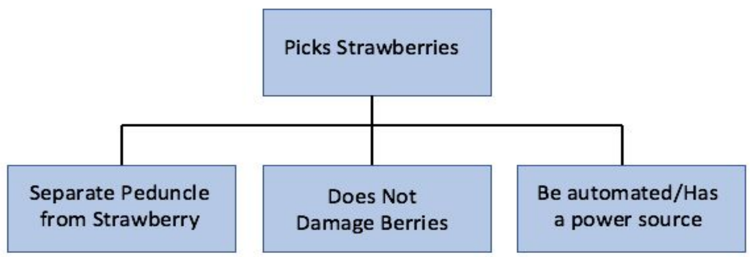


Figure 7. Functional Decomposition

Figure 7 shows how we broke down our problem into sub-functions. As can be seen, the sub-functions we felt were central to our problem were: separate peduncle from strawberry, do not damage berry, and be automated/have a power source. The rest of our ideation process revolved around these three sub-functions.

After performing functional decomposition we conducted ideation exercises to produce ideas for each of our sub-functions individually. For example, we would conduct a session to come up with different ideas for just separating the peduncle from the strawberry. Rather than try to solve the whole problem at once, this was a more methodical approach that allowed us to consider every major aspect of the problem individually. In order to produce ideas, we used several different techniques such as Brainstorming, Brainwriting, and S.C.A.M.P.E.R. (Substitute, Combine, Adapt, Modify, Purpose, Eliminate). After coming up with a number of ideas, we decided which ones we felt would be most effective for each sub-function and checked to see if any of our ideas overlapped from one sub-function to another. This helped us determine our best ideas and allowed us to move forward into our concept development.

4.2 Concept Development and Selection

Our ideation process gave us a number of ideas we felt strongly enough to move forward with. We began our concept development by making very simple prototypes for our most feasible ideas from ideation. We utilized simple materials such as foam core, dixie cups, and wooden skewers. These concept designs were not meant to be functional by any means, they were simply meant to give us a better feel for what our potential designs would look like and if they were feasible. Two of these concepts can be seen below.

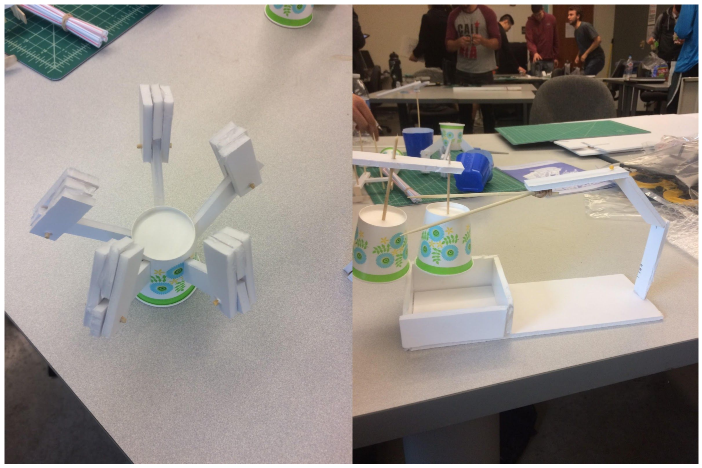


Figure 8. Simple Concept Models

After making various concept models, we moved into further concept development for the designs we felt most confident with. After some deliberation, we narrowed down our ideas to a handful of concept designs to move forward with in our selection process. They are described below, except for our selected concept which is described in the following section.

Grabber Arm Concept

Our first concept was one of the most simple and intuitive designs. It involves a claw with several arms that would softly grab the berry and pull it off of the stem. This design stood out to us initially because it is similar to the human hand and is fairly mechanically simple in nature. Our major concern with this concept was potentially damaging the berry with the grabber arms. In order to mitigate this issue, we planned on using five arms to increase contact points and thus decrease the force applied per arm. Additionally, the grabber arms could be made of a soft material in order to avoid bruising the fruit. Another major issue with this design is the motion required to separate the peduncle from the strawberry. When picking a strawberry, it cannot be simply pulled straight off of the stem; there needs to be a twisting motion. This motion is easily done by the human hand, but difficult to replicate with a robot. Using this method would require an arm that is able to perform complex movements.

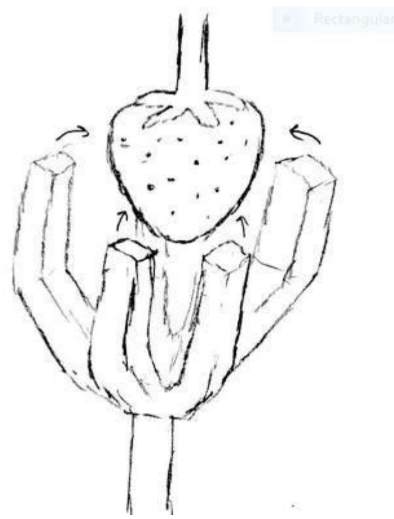


Figure 9. Grabber Arm Concept

One potential addition we felt could improve this design would be the use of some kind of tactile feedback system. The arms could be equipped with a strain gauge or pressure sensor that could ensure that the proper amount of force is applied to the berry as it is being removed. This application could even be extended further to help determine the ripeness of fruit by judging how firm it is. While this system would provide the benefit of preventing damage to the fruit, it would also add a large amount of complexity to the design.

Clamshell Concept

Another mechanical concept we developed was the clamshell design. As the name implies, this design would utilize a clamshell-like shape in order to pick the strawberry. It would open up and surround the berry before closing off and pinching the stem off. The top of the clamshell would have sharp edges to cut the peduncle as it closes. This design would be able to sever the peduncle and catch the berry in one motion. With only one hinging motion, the major benefit to this concept is its simplicity.

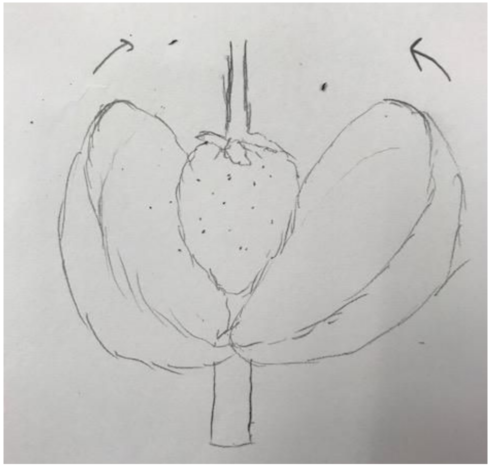


Figure 10. Clamshell Concept

While the clamshell design seemed effective in theory, it would be difficult to actually implement effectively. The major drawback to this concept is that it would only function properly for very specific growing patterns. The strawberries would need to be very spread out and hang down below the table top neatly. If berries were clumped together, the clamshell would damage fruit by closing on neighboring berries. Because of the unpredictable nature of agriculture, we cannot assume that all berries will be neatly spread out to accommodate this design.

Suction Method

The suction method is inspired by the Phil Brown/DBR Vacuum System mentioned in the background. The DBR Vacuum System utilizes suction to pluck apples off of the tree. Our suction concept would perform the same function but with strawberries rather than apples. It would use a high amount of suction to pull the berry off of the stem, and then quickly decelerate the berry to prevent damage. A benefit to utilizing this method is that it avoids potentially damaging the berry by grabbing it. If done correctly, it could also provide a very quick time-to-pick speed.

This concept would present major difficulties to effectively implement. The biggest issue is that there would be a very high risk of damaging the berries, as the suction required to remove the stem would cause the strawberry to move at a very high speed. In order to avoid damage to the berry, it would have to be decelerated very rapidly in a small amount of space. While this can be done with apples, as in the DBR system, strawberries are a much more sensitive fruit. It would likely not be feasible to decelerate the berry enough to avoid damage. In addition, utilizing suction would add more complexity to the problem and would likely require a fairly large amount of space to function properly.

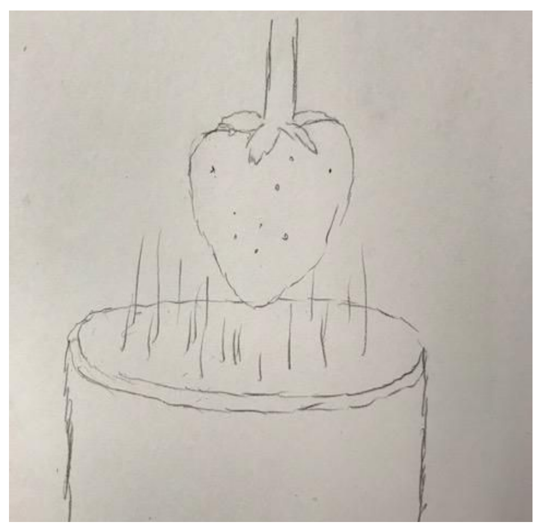


Figure 11. Suction Concept

Suction and Cut Method

Similar to the previous method, the suction and cut method would utilize suction to aid in picking the strawberry. Rather than utilizing suction to remove the peduncle from the berry, the suction would simply hold the berry in place, at which point a pair of shears would cut the stem. The suction would continue to hold onto the berry until it is transported to be packaged. This concept provides the benefit of not having anything physically grabbing onto the berry, which would mitigate potential bruising.

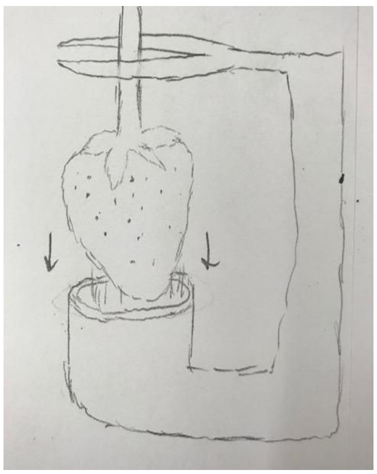


Figure 12. Suction and Cut Concept

In addition to the added complexity of utilizing suction, this concept has several downsides. For one, it would be difficult to properly orient the berry with suction before severing the stem. Our design would assume the the berry is being suctioned directly at the bottom every time, but that may not always be the case. The berry could get suctioned on the side, in which case the shears would not be in the appropriate position to sever the stem. This concept also presents the possibility of damage to other strawberries as well as the plant.

Cut and Catch Method

Our cut and catch method is another concept that would separate the peduncle from the berry via cutting with shears. The end effector would consist of a pair of shears attached to some sort of basket or catching device. It would approach a hanging berry from the side and cut the stem with the shears, with the basket oriented directly below the berry. The strawberry would fall into the basket where it could be transported to packing. The major benefit this concept provides is that there is minimal risk of damaging the berry. The only force the berry would encounter would be from the short drop into the catching basket.

Although this design is favorable in terms of forces applied to the strawberry, it does have its flaws. It would most likely only be feasible for ideal growing patterns, with the berries spread out evenly and hanging straight down. If the concept were to encounter berries that were clustered together, the end effector could easily cut adjacent berries unintentionally. Additionally, if the berries were not hanging straight down the end effector would be unable to catch the berry after cutting it. In an extremely controlled, ideal environment, this concept would likely be extremely effective. Unfortunately, as we have learned, ideal conditions are virtually impossible to obtain in agriculture.

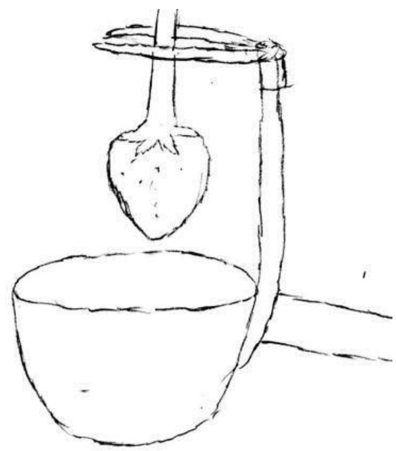


Figure 13. Cut and Catch Concept

With these design ideas, we moved forward in selecting one final concept with the aid of decision matrices (included in Appendix B). The purpose of creating these matrices was to assess and compare the ability of each of our design ideas to meet our specification requirements and other important design considerations. The specifications of higher importance were given a larger weight factor than the lesser important specifications, and then we scored each of the design ideas on their ability to meet our requirements. Through this exercise we were able to ultimately select our final design idea.

4.3 Chosen Design Concept

After considering the strengths and weaknesses of each of our concepts, we decided to move forward with a design that combined multiple methods. We liked the simplicity of the grabbing method, but wanted to avoid the issue of damaging the fruit while grabbing. The cut and catch concept navigated this problem by attacking only the stem, but the catching system required would have been too large, cumbersome, and difficult to maneuver. Our final design idea combines these two concepts, utilizing the strengths of each. The concept utilizes a grab-and-cut method, in which it grabs onto the stem before cutting it just above the grab point. It works by approaching the berry from the side, and proceeding to pinch the stem at an undetermined distance from the berry with two small arms. At this point two shears located directly on top of the arms are actuated to cut the stem. The two arms will still be holding on to the bit of stem attached to the berry and will transport the berry to the packing mechanism of the harvester. A SolidWorks model of our design can be seen in Fig. 14. .

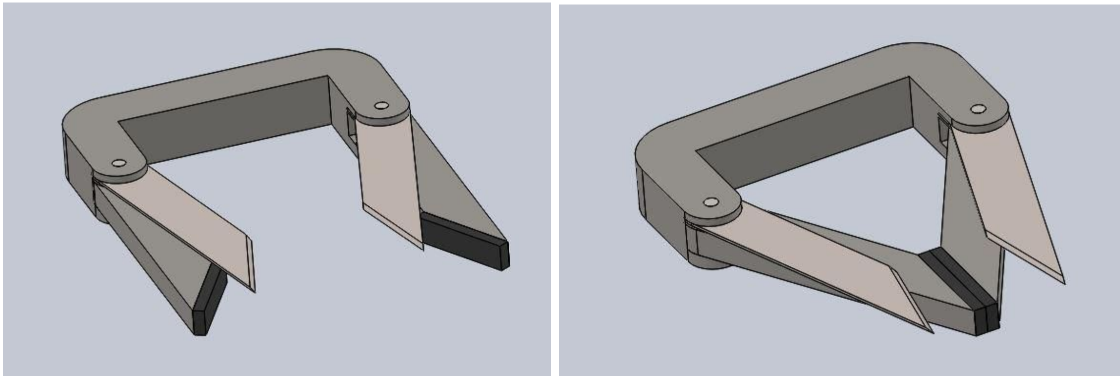


Figure 14. Preliminary SolidWorks Model of our Selected Design

As can be seen in figure 14, the end of the grabber arms will be made of a separate material, likely some sort of rubber with a high friction coefficient. Additionally, the grabber ends will be textured to enhance the gripping ability. These additions should reduce the risk of the berry being dropped after it is picked.

In order to provide the actuation for the arms to pinch the stem and the shears to cut the stem, we will attach small DC servo motors to each side of our end effector. Our plan is to incorporate a closed-loop proportional controller in order to control the magnitude and direction of motor torque as well as set the limits for the positions of the arms and shears.

We hope to be able to seamlessly integrate our end effector into various systems. The plan is to attach the base of the grabber to a robotic arm which will be responsible for orienting the end effector to grab the strawberry. Additionally, there will be a specified area to mount a camera to utilize image recognition software to locate the strawberries.

4.4 Potential Challenges

The design that we have chosen to prototype and test contains several potential challenges, the first of which is the problem of the stem itself. A system that focuses on grabbing the stem of the strawberry rather than the actual berry itself presents the problem of stem recognition. Although actual imaging and recognition of berry location and ripeness is not in the scope of this project, we still are attempting to be cognizant of how our end effector will interact with the entire system. With the research we have done, most operating systems base their imaging off of the actual strawberry, so as long as we are still able to recognize the different orientations of berries, then we would be able to find the stem located at the top of the berry. This sort of pinching end effector would require the ability to be in the correct position to move towards the stem and grip it without any interference from other parts of the plant. Our idea for this attack method is partly coming from graduate student John Sadler, who is working on a program to recognize a full plant and orient the end effector after imaging the entire profile of the plant. Since we are focusing our design on hydroponic growing systems, the issue of attacking the stem is simplified to an extent. In a hydroponic system, the berries hang off the side of a growing table, which makes the stems much more accessible and does not require the accuracy that would be needed to attack a traditional in-ground strawberry plant.

Our second potential challenge lies in the possibility of plant damage. As mentioned previously, the system would need to move in towards the stem without grabbing any other part of the plant, which requires an accurate device. If our device is using a smart enough computer that can accomplish this, then we would be able to effectively avoid any direct damage to other parts of the plant than the desired stem. This is where we would also desire to keep the profile of the gripper small in order to reduce the likelihood of interfering with undesired parts of the berry plant. This moves into part of the next set of challenges, which includes the force, speed, and size constraints. As mentioned, we would ideally like our device to be a smaller profile in order to minimize any potential damage to other berries or parts of the plant. Concerning the applied force, we imagine that our range of allowable forces applied to the stem is larger than what would be acceptable if we were attacking the actual berry. With our end effector focusing on the stem and not contacting the berry, we have minimized a large concern with forces occurring in the system. Nevertheless, we still need to figure out what forces are required to sufficiently hold on to the the stem without crushing it and dropping the berry. In addition to the force and size, we will focus on the speed of the device. With a single stem picking end effector grabbing such a small part of the plant, we will want to minimize the actual time it takes from recognition to a fully removed berry.

We will also be concerned with the ability to adapt to different grow patterns. This has been a concern from the beginning and is a challenge that we have been seeking to minimize. After visiting the tabletop facility at Driscoll's, we now recognize that there are many different strains of berries that could potentially be at the receiving end of this picking device. These variations of berries have vastly different grow patterns, with some hanging off the side of the table quite nicely, and others sticking inside the plant surrounded by leaves. The ideal situation puts the strawberries hanging off the side of the boxes in order to minimize interference from the plant and make targeting and picking more successful. Because we certainly cannot expect ideal situations at all times, we must adapt our device to fit these different growth patterns.

5.0 Final Design

5.1 Overall Description and Layout

Our design utilizes the grab and cut method we have previously discussed. The end effector first grabs the stem of the berry using grabber arms powered by a servo motor. Once the stem is firmly grasped, a cutting mechanism is actuated by a DC motor and severs the stem. At this point the berry is transported to the next step of the process to be packed. An isometric view of the completed assembly can be seen in Fig. 15.

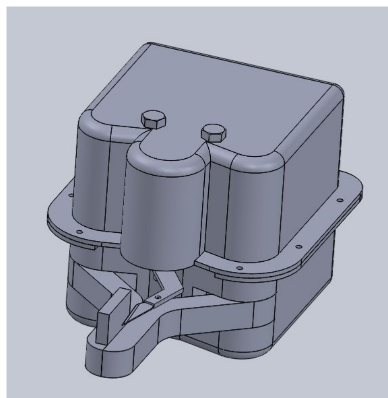


Figure 15. Final Design of End Effector

The entire assembly has a footprint of roughly 9.25 x 5.15 x 4.5 inches (length x width x height). We feel the end effector is small enough to maintain mobility and effectively reach into areas where it may need to access a stem. The assembly is composed of several major components: the upper housing, bottom housing, servo motor, DC motor, shafts, gear train, cutting device, and grabber arms. These components can be seen labeled in the exploded view in Fig. 16

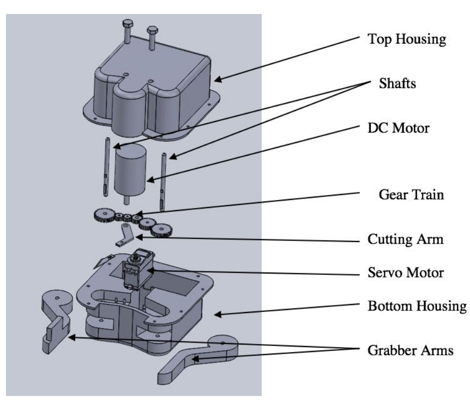


Figure 16. Exploded View of Final Design

5.2 Bottom Housing

The bottom housing, seen in Fig. 17, contains the servo motor, gear train, grabber arms, shafts and the electronics controlling the servo and DC motors. It also has a mount in the back to attach to the robotic arm. The arm slides into the mount on the bottom housing, fitting snugly before a bolt is inserted to hold the arm and housing together.

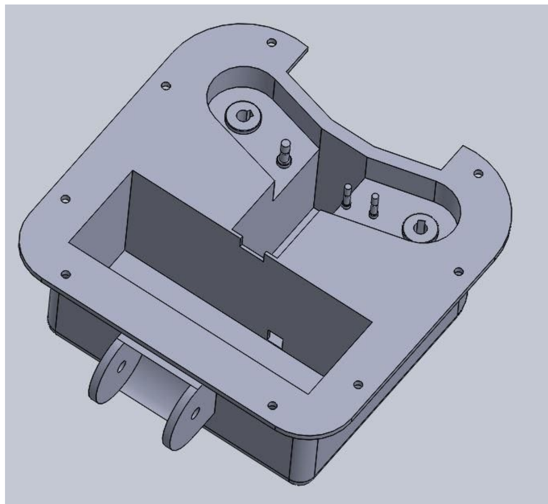
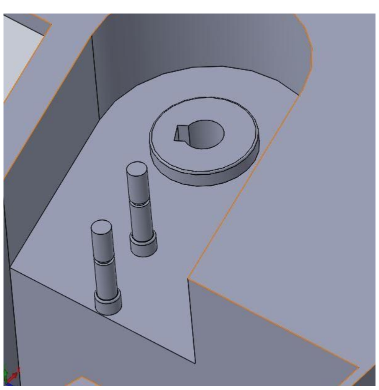


Figure 17. Bottom Housing

The bottom housing also has a cut out to hold the electronics driving the motors. This cut out is 2 inches long, 4 inches wide, and 1.5 inches deep. Note that the top housing has a cut out with the same dimensions, making the electronics housing 3 inches tall. In addition to the electronics housing, there is also a cut out to hold the servo motor. This cut is roughly the dimensions of the servo motor (1.72 x 0.88 x 1.57 inches). The servo fits snugly in this slot and allows the driving gear to be level with the rest of the gear train. There is also a small slot cut joining the servo housing and the electronics housing to allow the wiring to be connected between the servo and the controller.



The bottom housing also has the pins for mounting the gears. These pins allow the gears to spin in place while transmitting torque through the gear train. The gears are held on vertically by retaining rings or c-clips to prevent them from falling off when the end effector is turned upside down. The outer gear mounts also have a slot cut in them for the shafts to slide through. These slots have a keyway to allow the shaft's key to slide through down into the arm without interfering. Finally, the bottom housing has a flange with holes to attach to the top housing.

Figure 18. Gear Pins

5.3 Grabbing Assembly

The grabbing assembly consists of the servo motor, gear train, shafts, and grabber arms. The servo motor being utilized is a D-980TW with a weight of 2.76 oz and dimensions of 1.72” x 0.88” x 1.57”. The servo motor has a voltage range of 3.5 V to 8.4 V and a maximum torque of 611 oz-in. This torque falls in between our desired range of torques, obtained through hand calculations. This servo motor transmits torque to a driving gear, which subsequently drives a gear train that transmits torque to the grabber arms. A schematic of the gear train can be seen below.

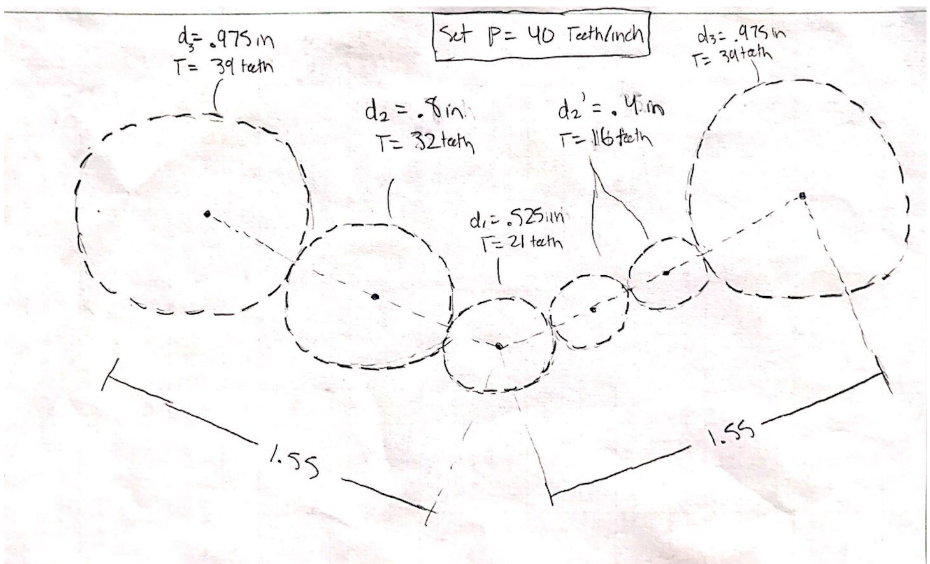


Figure 19. Schematic of Gear Train

A diametral pitch of 40 Teeth/inch was selected for the gear train, and the sizes of the gears were then chosen based on our geometric constraints and desired torque output. The outer gears driving the arms transmit torque to the shafts through a key, which subsequently drives the arms through a key with the same dimensions. The shaft and keyway dimensions can be seen in Fig. 20. Since we are dealing with relatively low torque and a grabbing system that makes no more than 45 degree turns (rather than many revolutions), we were confident that these keys would sufficiently transmit the torque, as they did. The smaller, top key transmits torque from the gears, while the larger, bottom key transmits torque to the arms. As can be seen in Fig. 20, the keyway is 0.10 inches wide and 0.05 inches long. The shafts’ top and bottom ends sits inside circular cutouts in the bottom and top housing to keep the shafts held in place. The shafts have rounded ends to reduce the friction of the shaft ends rotating against the housings.

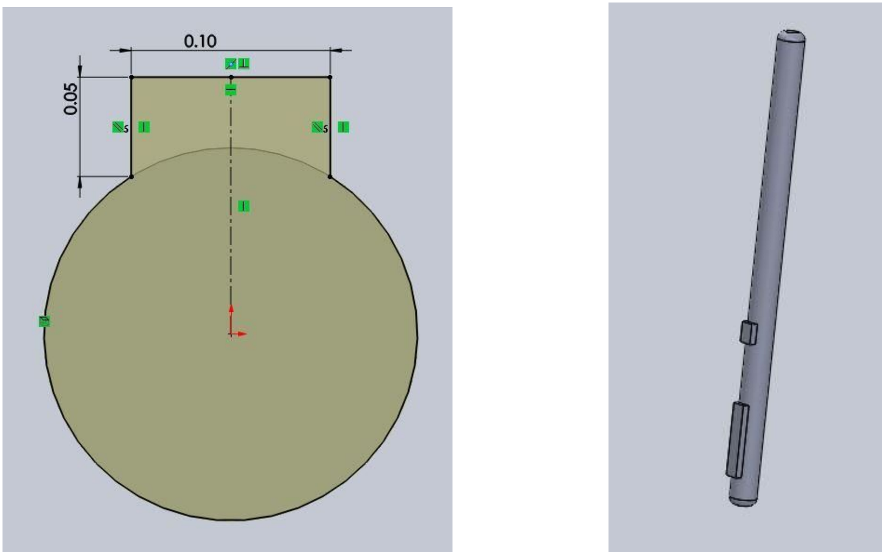


Figure 20. Shaft Schematic

The entire grabbing system fits in the bottom housing, and can be seen in Fig 21. The resulting system grabs with a force between 5-10 pounds, an acceptable range to properly grasp strawberry stems without inflicting any damage.

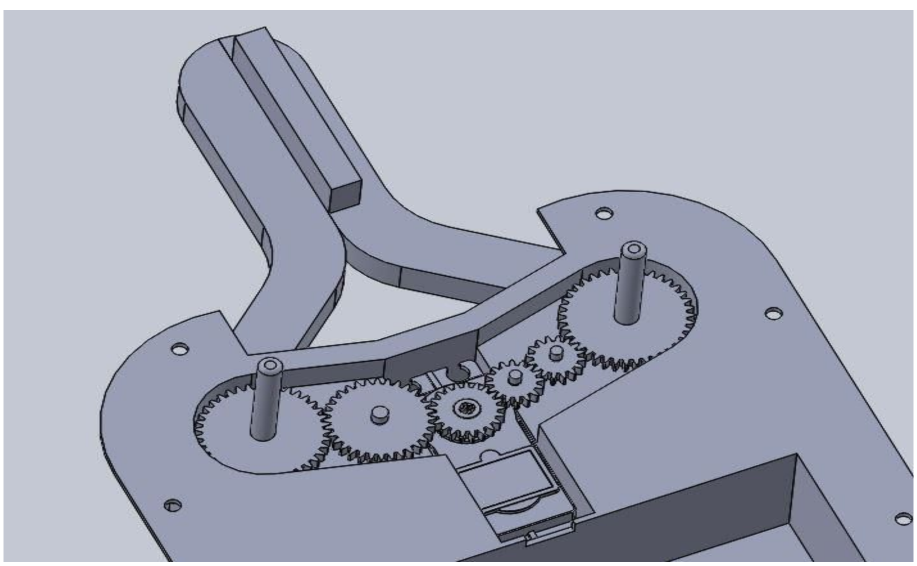


Figure 21. View Inside Grabbing Assembly

5.4 Top Housing

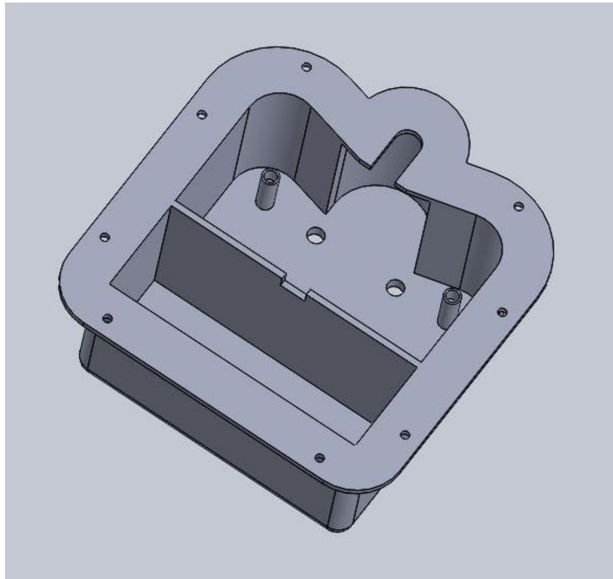


Figure 22. Top Housing

The top housing is roughly the same size as the bottom housing and holds the DC motor to actuate the shear as well as assist in holding the electronics. Similar to the bottom housing, it has a 2” x 4” x 1.5” cut out to hold the electronics. There is a small slot cut to allow wiring to run from the DC motor to the electronics driving the motor. At the front of the top housing, there is a cylindrical cut out to hold the DC motor in place. Once the DC motor is placed in this cut out, two set screws are inserted, holding the motor firmly in place to prevent it from falling out as the end effector moves.

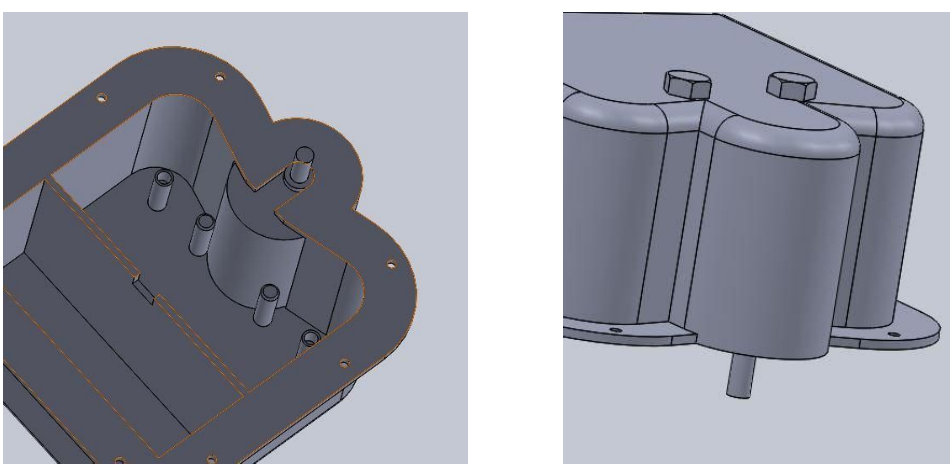


Figure 23. Top Housing DC motor mount

The housing for the DC motor has a slot cut out to allow the motor shaft to protrude outwards and drive the shear. To hold the assembly together the top housing has a flange with holds to be bolted to the bottom housing.

5.5 Cutting Assembly

The cutting assembly consists of a DC motor with an attached encoder and a cutting arm which utilizes an Xacto knife blade to cut the stem. Once the stem is held by the grabbing assembly, the DC motor actuates the blade, cutting the stem against a cutting block-like protrusion extending from one of the arms. The DC motor being utilized has a nominal voltage of 12 V, with a no load RPM of 107 rpm. The motor has a torque of 150 oz-in, and with our roughly 3 inch cutting arm, creates a cutting force of slightly over 3 lb.

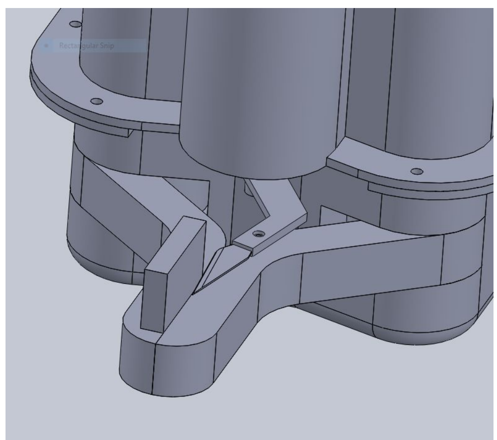


Figure 24. Cutting Mechanism

Our cutting arm is designed to fit a standard Xacto blade. The blade will be slid into the cutting arm and held in place by a bolt.

5.6 Mechatronics

To control the DC and servo motors we used a NUCLEO L-476RG microcontroller running python code along with a X-NUCLEO-IHM04A1 brush DC motor driver. The 12V DC motor is powered through the motor driver and it's position is controlled with the aid of the attached encoder by sending encoder counts through the microcontroller. The servo motor was wired to the microcontroller, through which 3.3V pulse width signals were sent in order to control the position of the servo motor. The complete code used to control the DC motor is included in Appendix J, and the code to control the servo motor is included in Appendix K.

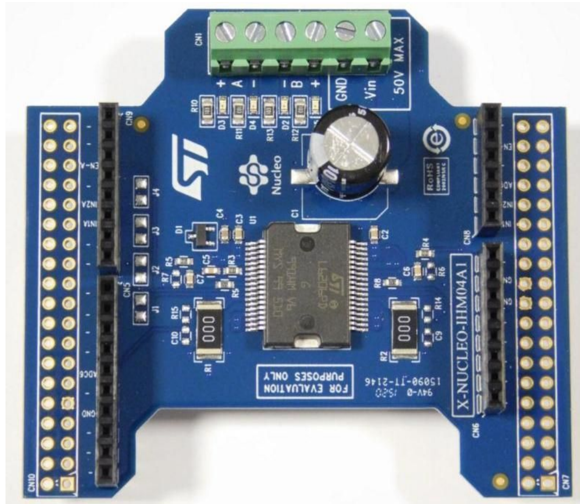


Figure 25. X-NUCLEO-IHM04A1 Dual Brush DC Motor Driver

The electrical wiring involve with the end effector is fairly simple. Altogether it will consist of a 12V power source, a 6V power source,, the microcontroller, the DC motor driver, and the DC and servo motors. The DC motor requires a 12V power source and the servo motor requires a power source with a voltage between 3.5V-8.4V. The microcontroller, motor driver, and both motors are enclosed in the end effector housing with wires coming out to the two power sources.

5.7 Cost Analysis

The California Strawberry Commission offered us a $1,000 budget for our project, and we ended up well under this number. The total cost of all of our components to be ordered was less than $350. This is due to the fact that the majority of our parts will be 3D printed. The most expensive part to be ordered is the Hitec D-980TW servo motor costing $177. This decision is justified due to the importance of choosing a high torque servo motor because it needs to provide enough torque to both grabber arms to pinch the stem with adequate force. Due to the smaller force required by the shear cutting and less need for encoder control, we decided on DC motors which are relatively cheap in comparison. In considering our other costs we based our 3D print costs off of third party sources with ratings for ABS plastic 3D printing. Attached in appendix H is a list of specific model numbers to order with the product links.

Table 3. Bill of Materials

Item	*Amount*	*Price*
HiTec D-980TW Servo Motor	1	$177.00
DC Motor	1	$28.75
Microcontroller / Motor Driver	1	$37.48
Breadboard	1	$7.96
Wires	1	$6.98
Shears	1	$29.95
Power Supply	2	$30.00
Total Price		**$318.12**

5.8 Material, Geometry, and Component Choices

For our material we selected ABS plastic to 3D print. This conclusion comes from the cheaper cost of the material and the relatively strong nature of the plastic. We are not exerting high forces on any of our parts and believed this would be able to hold up to the tolerances we require, which it did. With a selection of a cheaper plastic we are also able to produce multiple iterations or upgrade the material due to the low cost of our initial choice and being well under budget.

The geometry of the end effector originates from the need to fit components but also minimize the footprint. We were cognizant of the risk resulting from a larger berry picker and how it could potentially impact the plant and cause damage. In avoiding this risk we kept our grabber arms small and with a shorter profile while attempting to keep the grabbing surface larger. In addition the main aspects to design around were the motors and electrical components with the drivetrain orienting the front part of the body.

5.9 Changes to Final Design

For the most part, the final design remained relatively unchanged after the CDR. Most of the changes made were minor adjustments to the size of the parts to ensure everything fit together properly. The assembly as a whole increased in length by 0.5 inches, and the height increased by roughly a quarter inch.

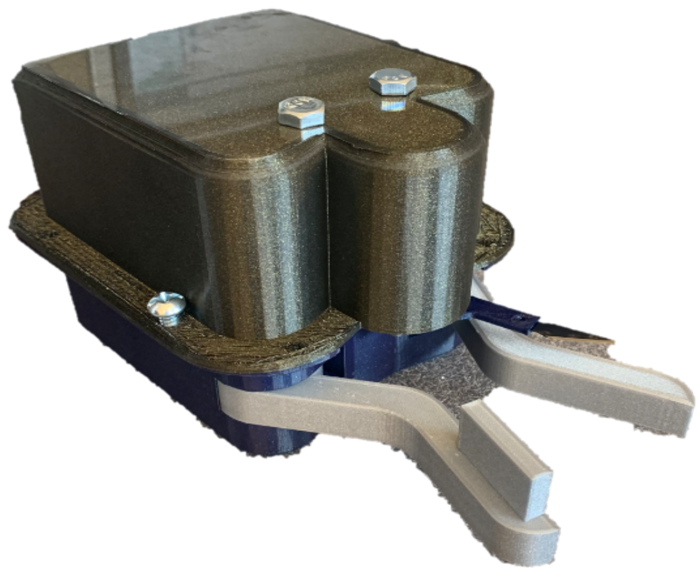


Figure 26. Final Assembly

Bottom Housing

The part with the most alterations was the bottom housing. The overall profile was lengthened by a half inch to increase the size of the slot holding the electronics. In addition, the thin wall separating the slot for the servo and the slot for the electronics was removed. This was done to allow the servo to be slid forward into place, rather than put downward into the slot. The reason this was necessary was because an overhang was added above the servo slot. The servo used was wider than anticipated, so in order to keep the desired gear geometry the overhang was added. This overhang also had the added benefit of holding the servo accurately in place to ensure the pinion gear is in the proper location to mesh with the gear train.

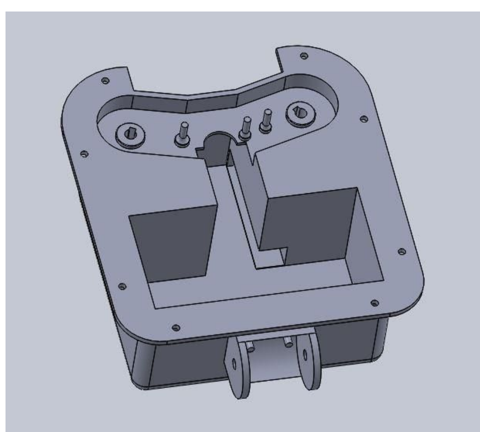


Figure 27. Final Bottom Housing

The rest of the changes made to the bottom housing involved adjusting the geometry to ensure everything fit properly. The shell cut-out that contained the gear train was slightly increased in size to make appropriate room for the gears. In addition, the slot on the lower right of the servo housing was increased to make sufficient room for the servo wiring. Finally, the three gear pins were given a fillet at their base to strengthen them and prevent them from snapping off.

Top Housing

The top housing remained completely unchanged with the exception of two minor geometry adjustments. The length of the housing was increased by a half inch to make more room for the electronics, as discussed in the changes to the bottom housing. In addition, the height of the housing was increased by roughly a quarter inch to make room for the DC motor.

Cutting Assembly

Only change was incorporated in the cutting assembly, and that was modifying the arm that the blade cut against. As can be seen in Fig. 28, a slot was added to the cutting block to allow the knife blade to pass completely through the stem. This allowed for the DC motor to make more effective cuts through the stem.

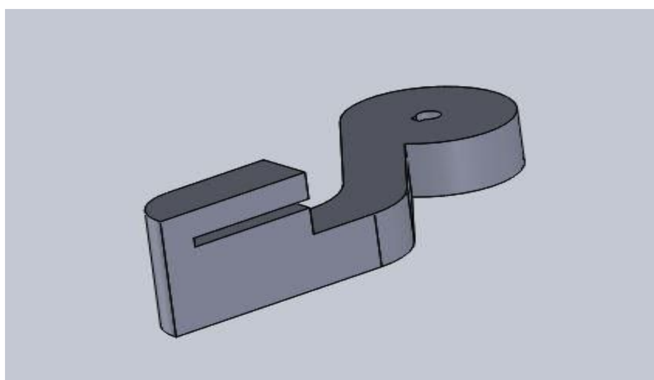


Figure 28. Final Right Grabber Arm

Grabbing Assembly

The main change in the grabbing assembly was the change to the right grabber arm, as discussing in the cutting assembly. The geometry of the gear train stayed exactly the same, and the only other changes were minor adjustments to the sizes of the gear bores and the shafts to ensure a proper fit.

6.0 Manufacturing

6.1 Procurement

We ordered the DC and servo motors, microcontroller, motor driver, shear, fasteners, the shear and all the circuitry. The microcontroller and motor driver are both NUCLEO model from the same manufacturer ST and were ordered from their website. The Hitec servo motor was ordered from ServoCity and the DC motor ordered from RobotShop. The circuitry (including the breadboard, resistors, switch and wires) was ordered through Amazon. Bolts and set screws were ordered from McMaster Carr.

6.2 Manufacturing

The majority of the components in our design were 3D printed with ABS plastic. The decision to use ABS plastic comes from the lower cost of the material as well as the relatively good strength and finish properties. The 3D printed components include the top and bottom housing, gears, shafts, and grabber arms. We 3D printed some of the parts through the resources available on Cal Poly's campus. We were satisfied with the quality of the 3D prints from Cal Poly's resources so there was no need to use costlier outside methods.

6.3 Assembly

The assembly process began after obtaining all parts. After wiring the electrical components correctly, the servo was placed into the bottom housing and the electronic components into their separate space. The gears were then put into their respective places, with the pinion gear being screwed onto the output shaft of the servo. The arms were aligned and the shafts slid through the openings to align the key into the keyways in order to transmit torque. The top housing was fitted with the DC motor into the cylindrical section and two set screws placed in from the top to hold the motor in place. The shear was then attached to the driver arm of the motor and the housings fitted with screws attaching the flanges. A more detailed description of the assembly process can be found in

With the relatively uncomplicated assembly process resulting from 3D printing more complex parts, few complications occurred. A few complications arose from the accuracy and tolerancing from the printed parts, but after multiple iterations of 3D prints all of the parts fit perfectly and were used in the final end effector system.

7.0 Design Verification Plan / Testing Results

After obtaining all our parts and assembling our prototype, we were able to perform the testing noted in our DVPR sheet in Appendix F.

7.1 Gear Test

After hooking up the servo motor and attaching the driver gear, the remaining gears were placed on their pins and the servo was ran to make sure the mating of gears and movement was correct. For testing these systems, we fully rotated the gears to ensure there were no surfaces that did not properly mate and that the forces were transmitted correctly. Another important aspect to check was that the gears rotated easily on the small shafts they are set on. In performing these tests, we kept in mind that when the gears rotate in our operating system they only perform partial, or small rotations to close and open the arms a small amount.

7.2 Gear Test Results

Upon our first test of running the gear train, we were able to move all of the gears together but not as smoothly as was desired. After removing all of the gears it was discovered that there were burrs on some of the edges of the gears left over from the 3D printing job. We then removed all of the burrs and the gear train ran as smoothly as planned.

7.3 Grab Test

One of the most important tests we conducted was running the servo motor to actuate the entire grabbing system and test the gripping force acting on the stem. This test included grabbing several stems and ensuring that the stem was gripped firmly enough through the cutting and transporting motions. We wanted to ensure that the force applied was appropriate and we could meet our desired specifications of under 10 lbs of force. This required us to test the code which is discussed in section *7.7* and is a major part of our system operating as intended.

7.4 Grab Test Results

Since the gear train was tested and ran smoothly before installing the grabber arms to the system, the grabber arms had no problems with their motion. The code successfully controlled the servo motor, which actuated the grabber arms as expected. In order to determine the specific pulse width signals needed to control the open and close motioning of the arms, first a certain pulse width signal was set which was associated with the grabber arms in the position of closed but without any grabbing force. We then decremented the pulse width signal incrementally until we found them in a position open enough we saw fit to set as the open signal. We were able to set the closed signal by placing one of our fingers between the grabber arms and setting the signal to the position of closed without any grabbing force. We then incremented the signal until we felt the grabber arms were using enough force to firmly grasp a stem without using excessive force.

7.5 Shear Test

Part of the success of the design rests in the shearing of the stem from the berry. A preliminary test was accomplished to determine the average force required to cut a stem. This test was done starting from rest with no initial momentum. When the DC motor and 3D printed parts were received we performed another cutting test starting with the applied torque from the DC motor and the momentum that would result with the blade starting further away form the stem. This demonstration aimed to show that we could accomplish our cutting action with a variety of stems and their varying thicknesses.

7.6 Shear Test Results

We implemented a method to control the position of the DC motor such that it completes a sweeping motion into the slot of the grabber arm and without using four extra wires for an encoder to crowd up our electronics. To do this we set a max duty cycle on the motor (for maximum cutting force) and wrote code to set the duty cycle to zero after enough time has passed that is has reached the slot by the use of a timer in the code.

To test the ability of the attached blade to cut a stem, we grabbed multiple stems of varying thickness and hardness to attempt to cut. We then had the grabber arms hold a stem and commanded the DC motor to sweep the blade into the slot. We were successful in cutting through some of the thinner and weaker stems but were only able to partially cut through the thicker, harder stems. We are confident that our system would be able to cut through the majority of strawberry stems it would encounter, but in order to cut through tougher stems a sharper blade should be attached to the DC motor.

7.7 Code Test

In order to execute a pick successfully we needed our code to accurately control our motors. The code was written with Python and ran on the NUCLEO L-746RG microcontroller. After the code was completed and our end effector fully assembled and wired, we conducted testing in the mechatronics lab. The code was ran from one of our laptops using a remote terminal (specifically PuTTY) in which we had simple commands established for opening/closing the grabber arms and activating/deactivating the shear.

The code used to control the DC motor established a connection between the DC motor and the X-NUCLEO-IHM04A1 motor driver as well as between the attached hall encoder and the microcontroller. The position DC motor is measured in encoder counts and controlled by sending specific encoder counts corresponding to the deactivated and activated positions of the shear. The code to control the servo motor is simpler. It establishes the connection between the servo motor and microcontroller through which the servo motors position is controlled by sending specific 3.3V pulse width signals corresponding to an open or closed grabber arm position. In order to increase the precision in which both of the motors are controlled we incorporated proportional-integral (PI) controllers in our code.

7.8 Code Test Results

Upon connecting to and setting up serial port communication with the microcontroller, we were able to test sending commands to the servo and DC motor. Upon our first attempts we had trouble due to faulty code. However after debugging and rewriting the code we were able to successfully command the end effector system to grab a stem and cut it with one command.

8.0 Project Management

Our timeline of our project followed the major dates highlighted in Table 4. With our design completed and finalized by the time of the Critical Design Review we moved on to the build and test phase. While manufacturing our parts we learned the turnaround time for part production with the 3D printing was longer than anticipated. We were expecting to receive parts within several days of submission and ended up with at least a week between part submission and actual print production. This caused our project timeline to be adjusted as we needed to make several iterations to our design as we learned more about the assembly and function of the end effector.

Sticking to our timeline of producing a final product November 30th we planned out several steps for printing and testing through the months of October and November. With our first couple prints we were forced to acknowledge some issues with our gear pins and the fact that they would sheer if put under excess stress. By breaking the pins on these parts we needed to adjust again our flow of manufacturing to account for additional bases to be printed with the allotted time for each successive print. This onset of unforseen problems with rapid prototyping caused several pushbacks on our testing and final production.

In addition to the 3D print issues and the challenges they posed to our timeline, we needed to adjust for part ordering and shipping time. With the servo motor and DC arriving we were able to start testing the code to run these devices. Testing our electrical systems was crucial in our timeline towards the final product because this required that everything was able to run and properly execute commands.

Once all parts were received and printed, we were able to begin our testing with the assembled device. This required ample time for set up and running the device with live interactions of grabbing and cutting. We allotted the last two weeks to this, giving us time to trouble shoot both code and hardware.

With the timeline we set up and the several setbacks and adjustments made, we were able to produce our final product by the November 30th deadline and present our design and findings. In future design projects we will make allowances for setbacks and unforeseen issues to arise. The timing that we thought we could have if everything worked as expected would have allowed for more testing and an additional iteration to be designed and presented. Expecting the unexpected in a manufacturing design project is always the smarter idea as to better predict your timeline, and if there are not issues you come out ahead of schedule.

Table 4. Project Timeline of Key Deliverables

Deliverable	Date
Critical Design Review	May 1st, 2018
Manufacturing and Test Status	June 5th, 2018
Hardware and Safety Demo	October 18th, 2018
Final Product	November 30th, 2018

9. Conclusion & Recommendations

This document was created in order to outline our process in creating an end effector for an automated strawberry harvesting system, sponsored by John Lin and the California Strawberry Commission. In it we have aimed to explain and justify the design we have made for our end effector. While we feel we have created a satisfactory final design, there were certainly lessons learned and areas where we saw opportunities for improvement.

Our largest issue in our final design was the compliance in the grabber arms of the assembly. When the gear train was in position to have the arms closed, there was a good amount of wiggle room in the arms, resulting in a loose grab. This issue could be overcome in the grabbing motion by tightening the arms past the natural closed position. Although this resulted in sufficient force to grab the stem, it actually created issues with the cutting motion. When the knife made contact with the stem, it would push against the grabber arms. Because of the compliance in the arms, the grabbing assembly was not able to close tight enough to resist the force of the knife against the stem. This made it very difficult for the stem to be cut properly. We would certainly recommend eliminating the compliance between the shaft and the arms in order to prevent this issue. One way this could be done is by making larger shafts and larger keys in the shafts. This would allow the keys to fit better and have less free rotation in the arms.

Another recommendation we would make is to improve the handling of the electronics and wires. There is a large amount of wiring involved in our design, and it was difficult to deal with when assembling the prototype. For the most part, the wires were folded up and placed in the housing any way they could fit. In a future iteration, it would be useful to adjust the housing to more effectively hold the electronics and wiring. A good first step would be to make a slot that holds the electronics firmly in place. From there different slots and/or pins could be added to help align and hold the wires in a convenient manner.

The way that the pinion gear was attached to the servo motor was another point to be improved upon. In our design, the pinion gear simply sits on top of the servo output shaft and is tightened on by a screw. If the grabbing motion was actuated tightly enough it would loosen the screw and the pinion gear would no longer be able to transmit torque. This could be improved by making a pinion gear that properly meshes with the output shaft of the servo. The best way to do this would be to make splines on the inside of the pinion gear that allow the gear to snugly fit onto the output shaft.

If this product was created on a full production scale, we would recommend making it out of metal rather than plastic. This would make the final product must more durable and robust, as well as improve the issue of the compliance in the arms discussed earlier. Some of the parts would be difficult to machine, but they could be cast instead.

We hope the design we have put forward will help make progress in solving the issue of harvesting strawberries in some manner. We feel the research, existing technologies, and concept designs we compiled are an excellent base of knowledge in the automated strawberry harvesting industry and will hopefully inspire ideas for further development. We hope our final design, while certainly not the end-all be-all solution to the problem, will spark future development and serve as a building block to work towards further improvements.

Works Cited

Defterli, Sinem Gozde, et al. "Review of Robotic Technology for Strawberry Production."*Applied Engineering in Agriculture*, American Society of Agricultural and Biological Engineers, 1 Jan. 1970, elibrary.asabe.org/azdez.asp?search=0&JID=3&AID=46709&CID=aeaj2016&v=32&i=3&T=2.

Zhang, Xiuxia, et al. *Study on the Design and Control System for Wolfberry Harvesting Robot* . ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7532069.

Edan, Yael, and Gaines E Miles. "Systems Engineering of Agricultural Robot Design." *IEEE Xplore*, ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=299707.

Gao, Yang. *Towards Autonomous Robotic Systems: 18th Annual Conference, TAROS 2017, Guildford, UK, July 19, 2017: Proceedings*. Springer, 2017.

Yamamoto, Satoshi, et al. *Development of a Stationary Robot Strawberry Harvester with a Picking Mechanism That Approaches the Target Fruit from Below*. www.jstage.jst.go.jp/article/jarq/48/3/48_261/_pdf/-char/ja.

Davidson, Jospeh R, et al. "Proof-of-Concept of a Robotic Apple Harvester." *IEEE Explore*, ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7759119&tag=1.

Lin, John Ph.D. Personal Interview. 23 Jan. 2018

Appendix A: Quality Function Deployment - House of Quality

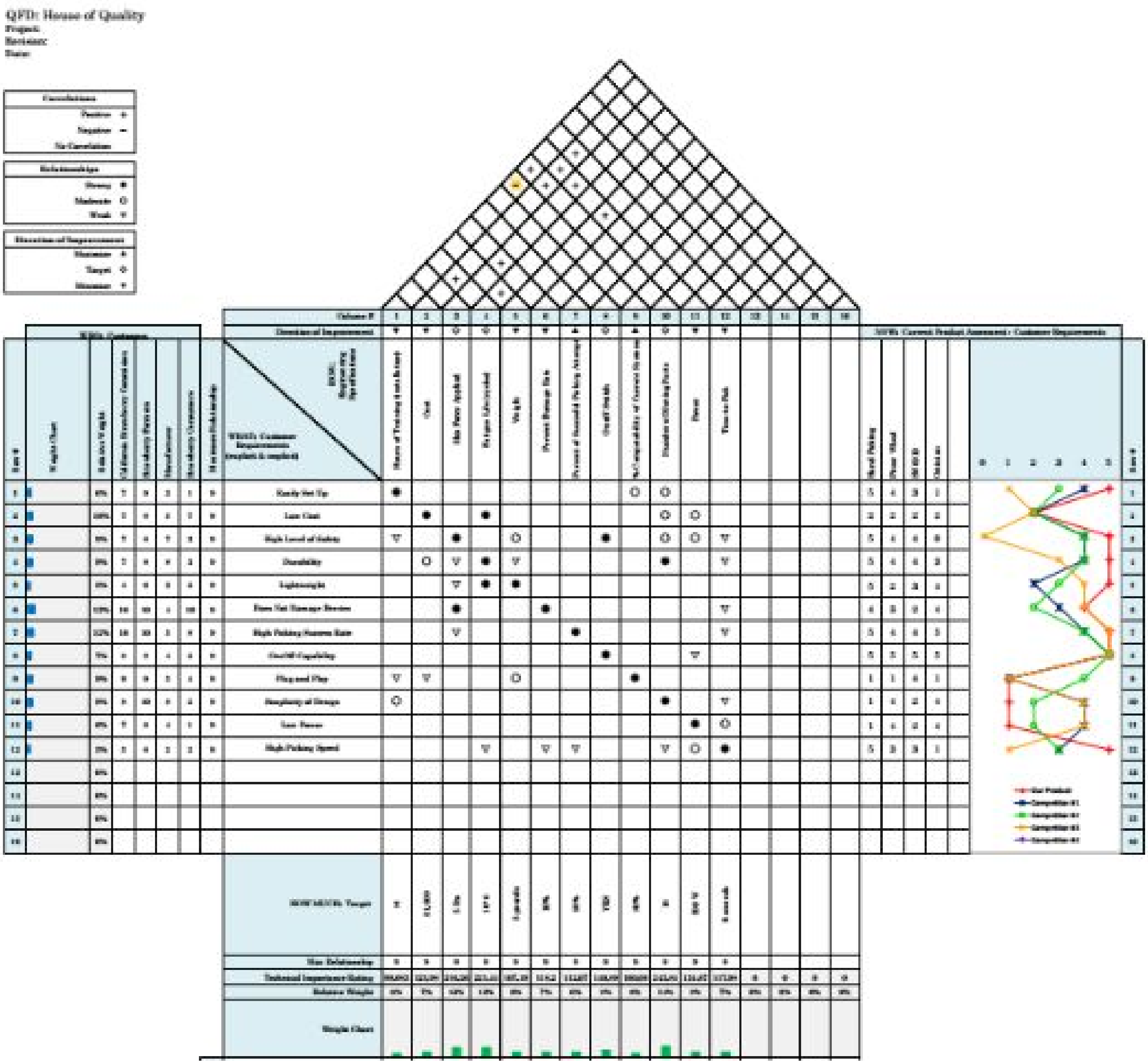


Figure A-1. House of Quality

<u>Appendix B:</u> Decision Matrices

Table B-1. Decision Matrix for Overall Design

Design	Cost	# of Moving Parts	Time-to-pick	Weight	Damage Rate	Successful Pick Rate		**Total**
Weight	2	3	4	1	5	5		
Grab berry and pull	4	6	12	4	10	15		**51**
Cut and catch	8	12	16	2	15	15		**68**
Grab and cut stem	6	9	16	5	20	15		**71**
Suction	4	12	16	2	10	10		**54**

Table B-2. Table for Sub-Function “Minimal Damage to Berry”

Design	Points of Contact	Force Applied to Berry	Potential for Drop Damage	Potential for Plant Damage	Footprint		**Total**
Weight	3	3	2	4	2		
Grab berry and pull	6	3	8	12	8		**37**
Cut and catch	12	9	4	8	4		**37**
Grab and cut stem	15	12	4	12	10		**53**
Suction	12	6	6	12	6		**42**

Table B-3: Decision Matrix for Sub-Function "Power Source"

Type of Power Source	Space Required	Cost	Life Span	Complexity of Integration	Amount of Available Power		**Total**
Weight	3	4	2	5	2		
Electrical	12	20	8	20	8		**68**
Hydraulic	6	8	6	5	6		**31**
Pneumatic	9	8	4	10	4		**35**
Fuel Cell	6	4	8	10	10		**28**

Appendix C: Full List of Customer Needs and Wants

Needs:
- Automated System
- Able to be attached to existing systems
- Completed by December 2018
- On/off capability
- Pick strawberries of all shapes and sizes without damage
- Operation in various environmental conditions (dusty, rainy, etc.)
- Weather resistant material
- Safe

Wants:
- $1,000 budget
- Plug-and-play with different systems
- Programmed control system (pressure/force feedback, voltage regulation, etc.
- Other personal interests incorporated
- Email correspondence once a week, in-person meetings bi-weekly
- Packs strawberries
- Operational in different strawberry growing patterns
- Lightweight, easy to move
- Durable and robust

<u>Appendix D:</u> Design Hazard Checklist

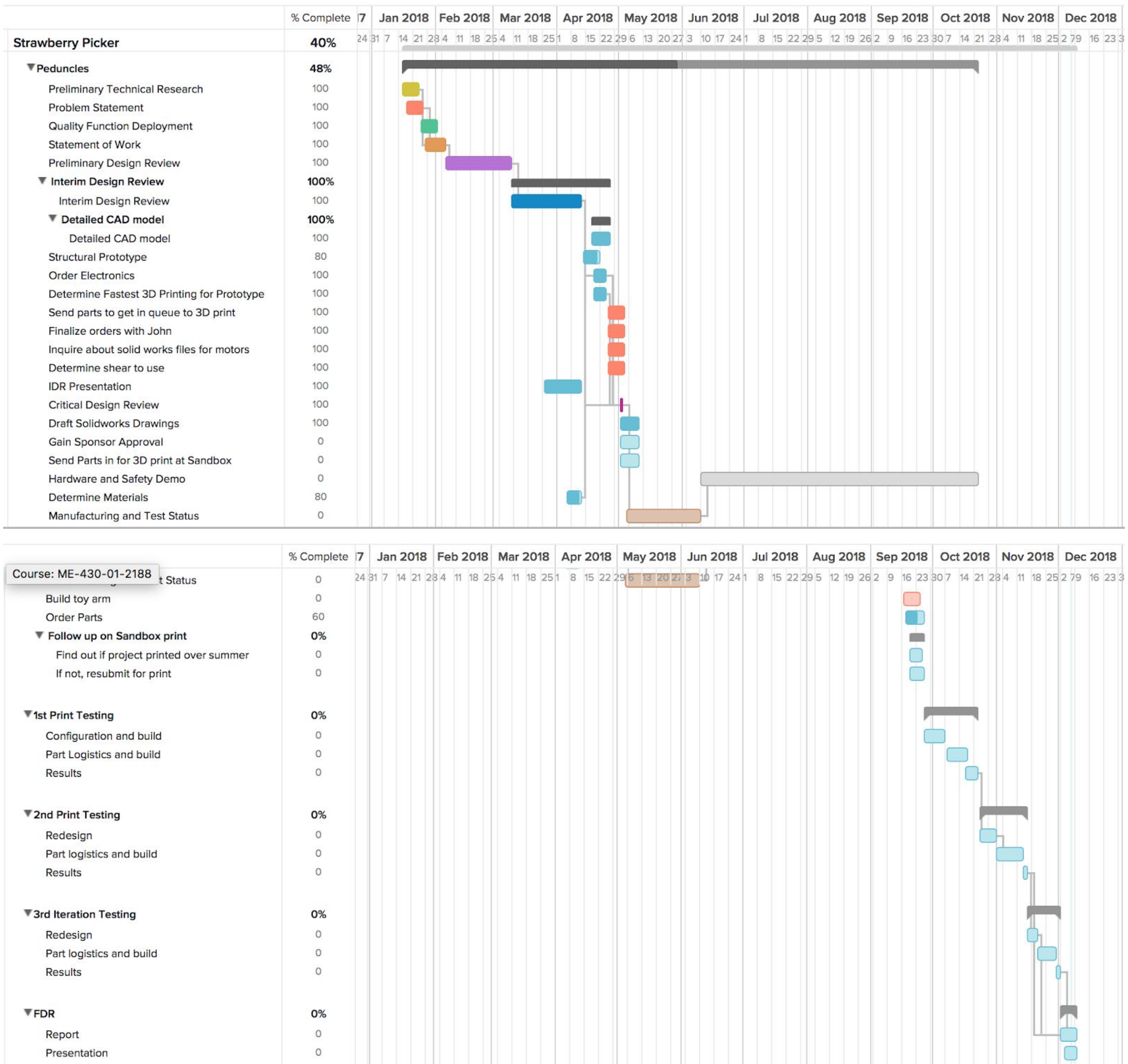
Team: Peduncles Advisor Dr. Self Date: Thursday, March 1

Y	N	
	x	1. Will the system include hazardous revolving, running, rolling, or mixing actions?
x		2. Will the system include hazardous reciprocating, shearing, punching, pressing, squeezing, drawing, or cutting actions?
	x	3. Will any part of the design undergo high accelerations/decelerations?
	x	4. Will the system have any large (>5 kg) moving masses or large (>250 N) forces?
	x	5. Could the system produce a projectile?
	x	6. Could the system fall (due to gravity), creating injury?
	x	7. Will a user be exposed to overhanging weights as part of the design?
x		8. Will the system have any burrs, sharp edges, shear points, or pinch points?
	x	9. Will any part of the electrical systems not be grounded?
	x	10. Will there be any large batteries (over 30 V)?
	x	11. Will there be any exposed electrical connections in the system (over 40 V)?
	x	12. Will there be any stored energy in the system such as flywheels, hanging weights or pressurized fluids/gases?
	x	13. Will there be any explosive or flammable liquids, gases, or small particle fuel as part of the system?
	x	14. Will the user be required to exert any abnormal effort or experience any abnormal physical posture during the use of the design?
	x	15. Will there be any materials known to be hazardous to humans involved in either the design or its manufacturing?
	x	16. Could the system generate high levels (>90 dBA) of noise?
x		17. Will the device/system be exposed to extreme environmental conditions such as fog, humidity, or cold/high temperatures, during normal use?
	x	18. Is it possible for the system to be used in an unsafe manner?
	x	19. For powered systems, is there an emergency stop button?
	x	20. Will there be any other potential hazards not listed above? If yes, please explain on reverse.

For any "Y" responses, add (1) a complete description, (2) a list of corrective actions to be taken, and (3) date to be completed on the reverse side.

Description of Hazard	Planned Corrective Action	Planned Date	Actual Date
Hazardous Cutting and Pinching of Stem	Shears will have a protective guard on them to avoid cutting anyone.	4/21	
The device may be exposed to environmental conditions such as fog, humidity, and cold/high temperatures	The device will be made of materials resistant to adverse environmental conditions	3/20	
Pinch point behind grabber arms	Place guard behind grabber arm	4/17	

<u>Appendix E:</u> Gantt Chart



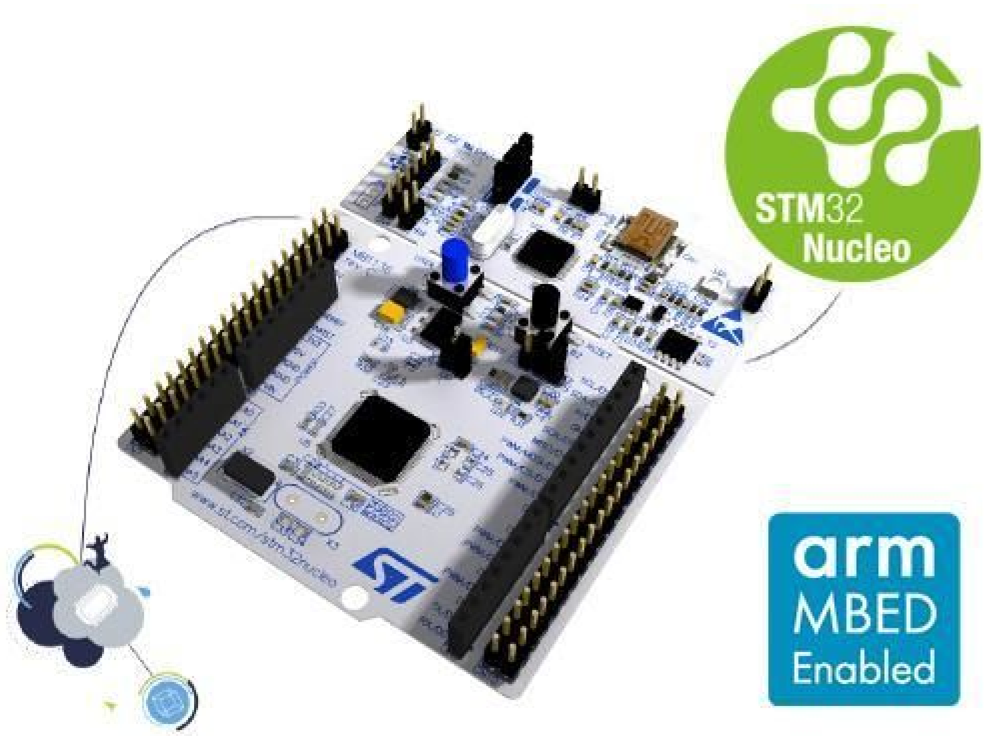
Appendix F: DVP&R Spreadsheet

Senior Project DVP&R

Date: 5-3-18 | Team: Peduncles | Sponsor: California Strawberry Commission | Description of System: Autonomous Strawberry Picker End Effector | DVP&R Engineer:

TEST PLAN										TEST REPORT			
Item No	Specification #	Test Description	Acceptance Criteria	Test Responsibility	Test Stage	SAMPLES Quantity	Type	TIMING Start date	Finish date	TEST RESULTS Test Result	Quantity Pass	Quantity Fail	NOTES
1	1	Hours of Training Requruied (installation)	<1 hour	Mick	SP		Sys						
2	2	Cost	<$1000	Mick	CP		Sys	4/29/18	5/3/18	$300			
3	3	Max Force Applied	<10 lbs	Jimmy	SP		Sub						
4	4	Fatigue Life (cycles)	Cycles	Cory	FP		C						
5	5	Weight	< 5lbs	Cory	FP		Sys						
6	6	% Damage Rate	<5%	Mick	FP		Sub						
7	7	% of Successful Picking Attempts		Jimmy	FP		Sub						
8	8	Off-On Switch		Jimmy	CP		C						
9	9	% Compatability w/ Current System		Cory	CP		Sys						
10	10	Number of Moving Parts		Cory	FP		Sys						
11	11	Power	<500 Watts	Jimmy	CP		Sys						
12	12	Time-To-Pick	< 3 sec	Mick	FP		Sub						
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													

<u>Appendix G:</u> NUCLEO L-746RG Microcontroller



Key Features

- STM32 microcontroller in LQFP64 package
- External SMPS to generate Vcore logic supply (only available on '-P' suffixed boards)
- 1 user LED shared with Arduino™
- 1 user and 1 reset push-buttons
- 32.768 kHz LSE crystal oscillator
- Board expansion connectors:
  - Arduino™ Uno V3
  - ST morpho extension pin headers for full access to all STM32 I/Os
  - External SMPS experimentation dedicated connector (only available on '-P' suffixed boards)
- Flexible power-supply options: ST-LINK USB VBUS or external sources
- On-board ST-LINK/V2-1 debugger/programmer with USB re-enumeration capability. Three different interfaces supported on USB: mass storage, virtual COM port and debug port
- Comprehensive free software libraries and examples available with the STM32Cube MCU Package

- Support of a wide choice of Integrated Development Environments (IDEs) including IAR™ , Keil® , GCC-based IDEs, Arm® Mbed™
- Arm® Mbed Enabled™ compliant (only for some Nucleo part numbers)

The STM32 Nucleo board provides an affordable and flexible way for users to try out new concepts and build prototypes with the STM32 microcontroller, choosing from the various combinations of performance, power consumption and features. For the compatible boards, the SMPS significantly reduces power consumption in Run mode.

The Arduino™ Uno V3 connectivity support and the ST morpho headers allow the easy expansion of the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.

The STM32 Nucleo board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger and programmer.

The STM32 Nucleo board comes with the STM32 comprehensive software HAL library together with various packaged software examples, as well as direct access to the Arm® Mbed™ online resources at http://mbed.org.

- Information from http://www.st.com/en/evaluation-tools/nucleo-l476rg.html

Appendix H: Linked Parts to Order

Servo for grabber arms
https://www.servocity.com/d980tw-servo

DC motor for shear
https://www.robotshop.com/en/12v-dc-motor-251rpm-encoder.html

Microcontroller
http://www.st.com/en/evaluation-tools/nucleo-l476rg.html

Motor Driver
http://www.st.com/en/ecosystems/x-nucleo-ihm04a1.html

Breadboard
https://www.amazon.com/Elegoo-6PCS-tie-points-Breadboard-Arduino/dp/B01EV6SBXQ/ref=sr_1_4?s=electronics&ie=UTF8&qid=1524520586&sr=1-4&keywords=small%2Bbreadboard&th=1

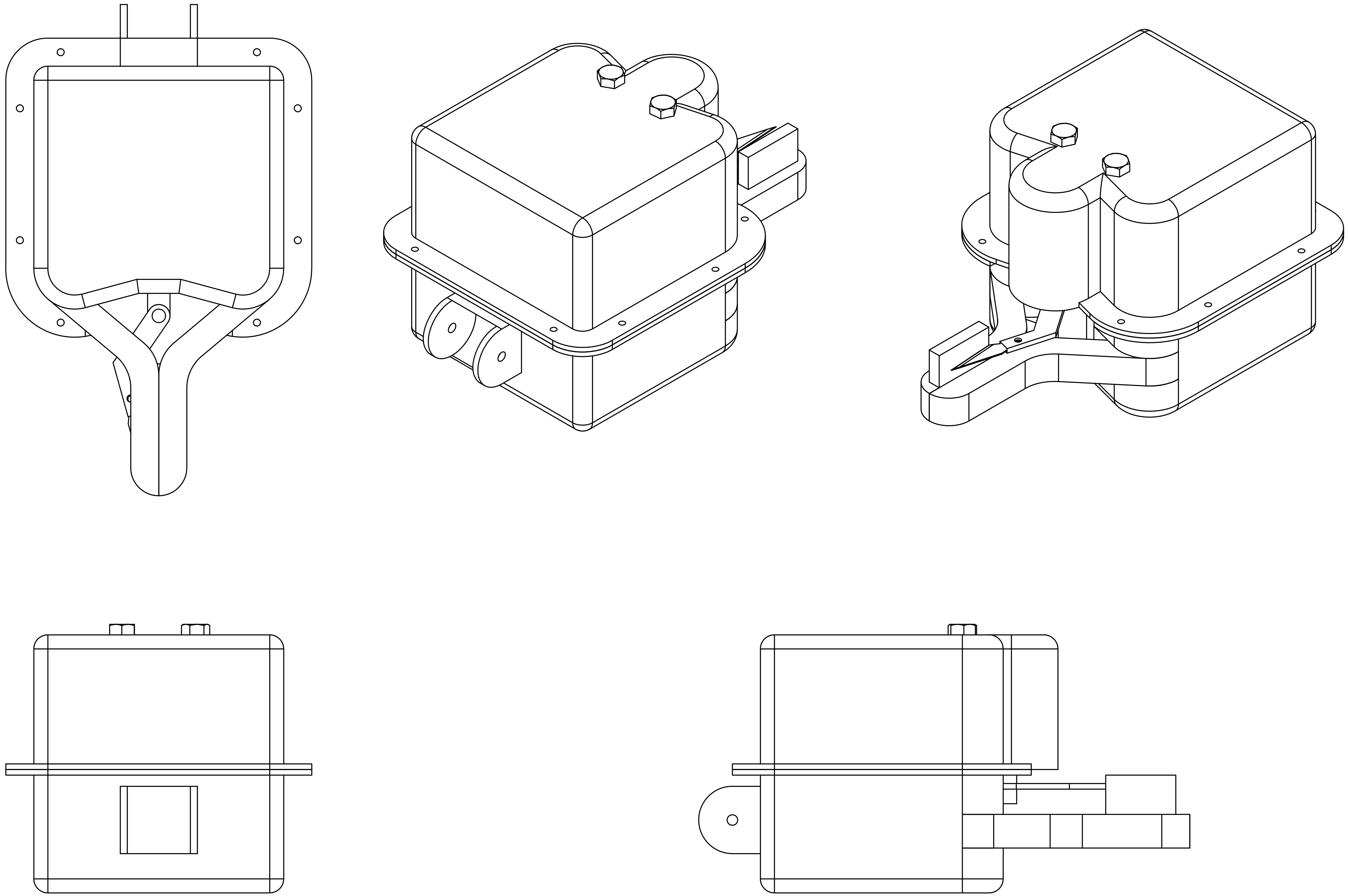
Wires
https://www.amazon.com/dp/B01EV70C78

Shears
https://www.amazon.com/X-ACTO-X-Life-Classic-Blades-X611/dp/B00006ICJV

On/off switch
https://www.zoro.com/carling-technologies-rocker-switch-spst-2-connections-ra911-vb-b-1-v/i/G3103694/#description

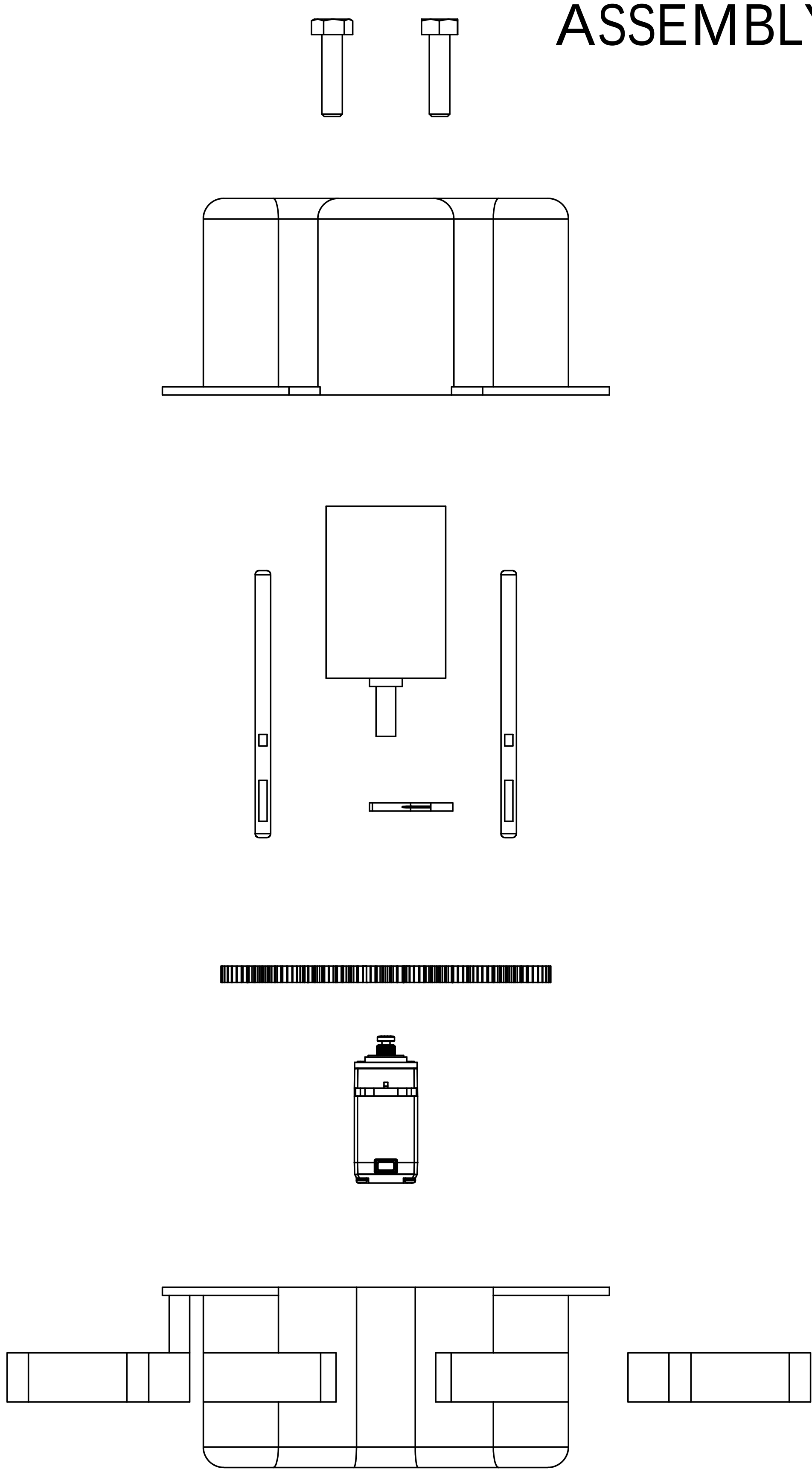
Appendix I: Engineering Drawings

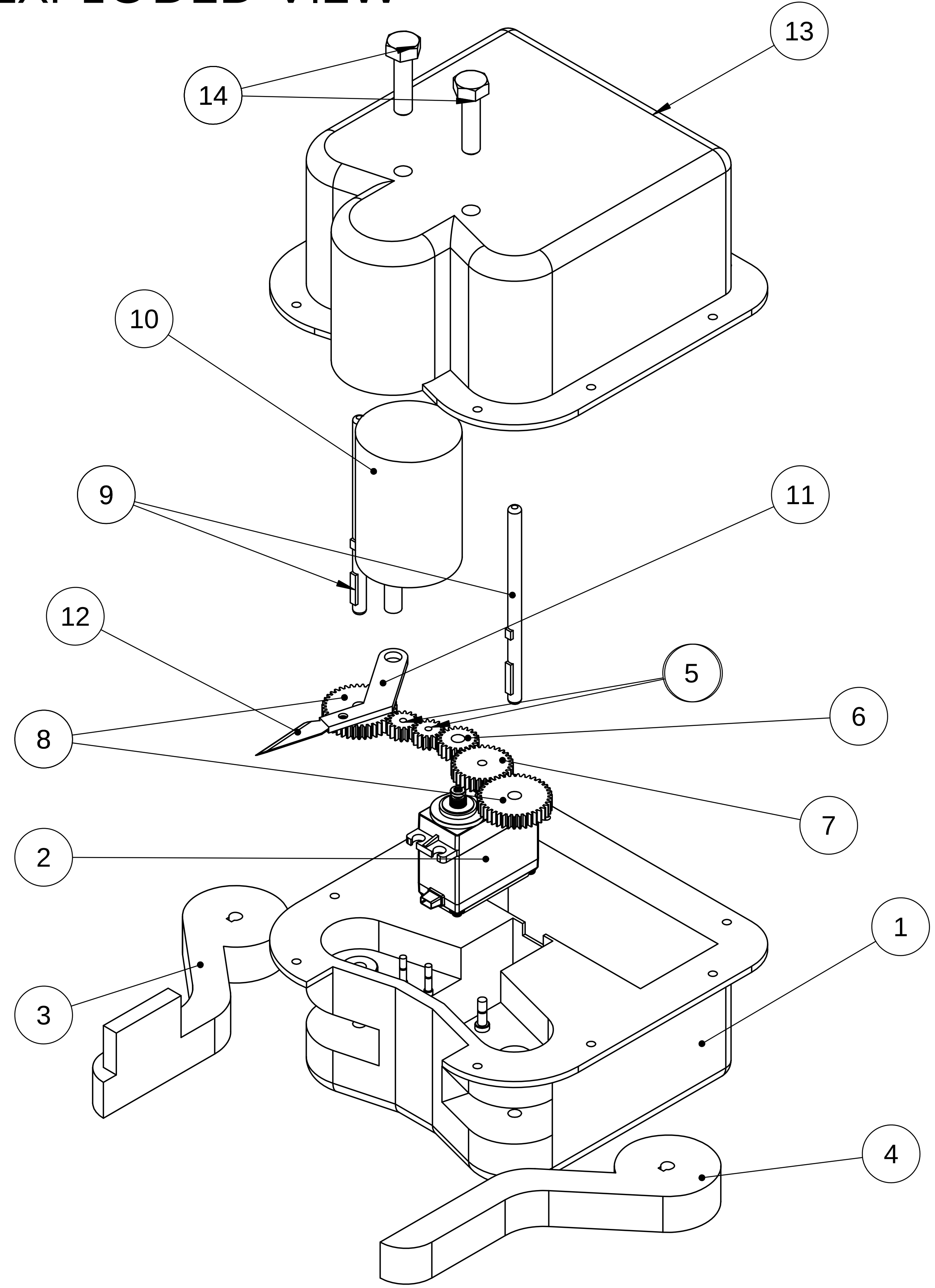
ASSEMBLY DRAWING



Cal Poly Mechanical Engineering	Lab Section:	Assignment #	Title: Assembly Drawing		Drwn. By:
ME 429 Spring 2018	Dwg. #: 1	Nxt Asb:	Date:	Scale: 1:2	Chkd. By: ME STAFF

ASSEMBLY EXPLODED VIEW

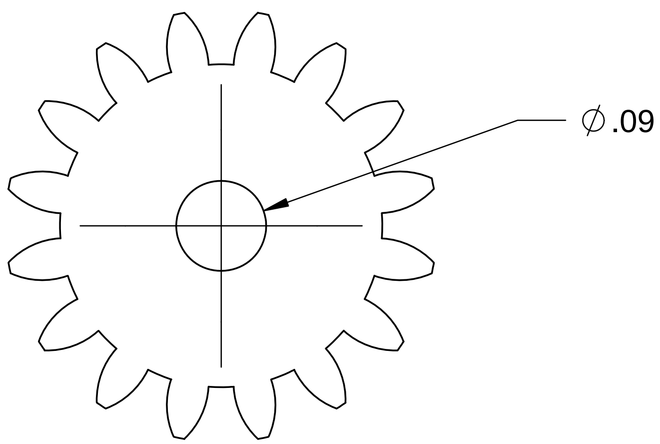


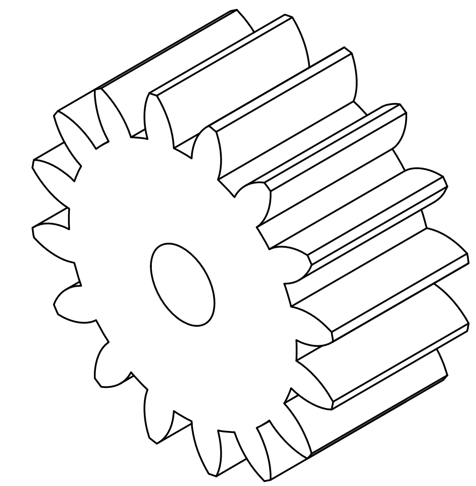


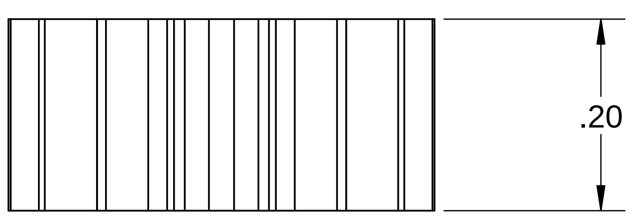
ITEM NO.	PART NUMBER	QTY.
1	Bottom Housing	1
2	Servo Motor	1
3	Right Grabber Arm	1
4	Left Grabber Arm	1
5	16T Spur Gear	2
6	21T Spur Gear	1
7	32T Spur Gear	1
8	39T Spur Gear w/ Keyway	2
9	Shaft	2
10	DC motor	1
11	Cutting Arm	1
12	Xacto blade - No 11	1
13	Top Housing	1
14	HBOLT 0.2500-20x1x1-N	2

Cal Poly Mechanical Engineering	Lab Section:	Assignment #	Title: Assembly Exploded View		Drwn. By:
ME 429 Spring 2018	Dwg. #: 1	Nxt Asb:	Date:	Scale: 1:2	Chkd. By: ME STAFF

16 T SPUR GEAR
DIAMETRAL PITCH: 40
TEETH NUMBER: 16
DIAMETER: 0.40 IN





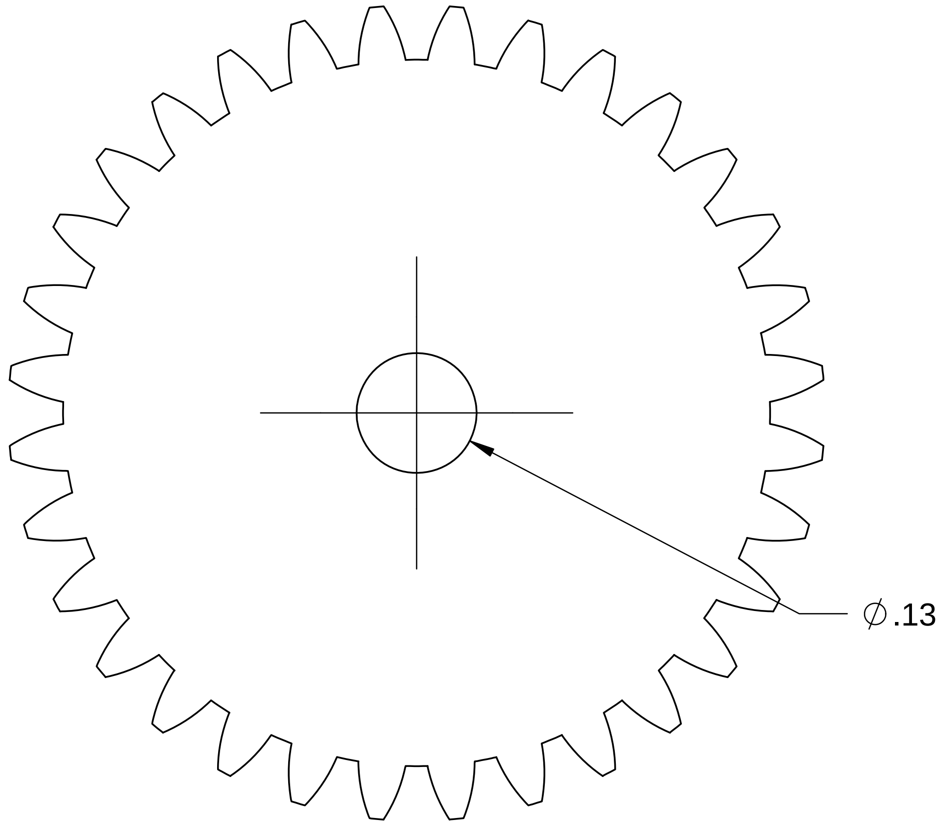


NOTES:

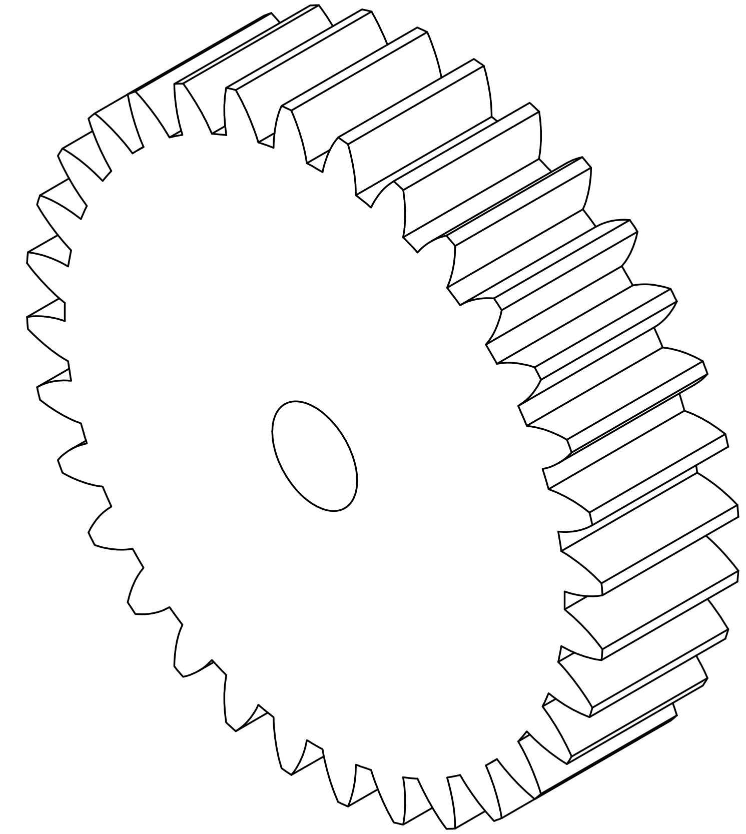
UNLESS OTHERWISE SPECIFIED:

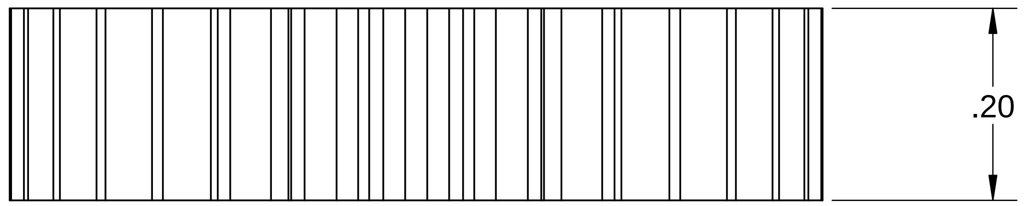
1. ALL DIMS IN INCHES
2. TOLERANCES:
   X.XX = ±0.01
   ANGLES = ±1°
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED

Cal Poly Mechanical Engineering	Lab Section:	Assignment #	Title: 16 T Gear		Drwn. By:
ME 429 Spring 2018	Dwg. #:4	Nxt Asb:	Date:	Scale: 5:1	Chkd. By: ME STAFF



21 T SPUR GEAR
DIAMETRAL PITCH: 40
TEETH NUMBER: 21
DIAMTER: .525 IN

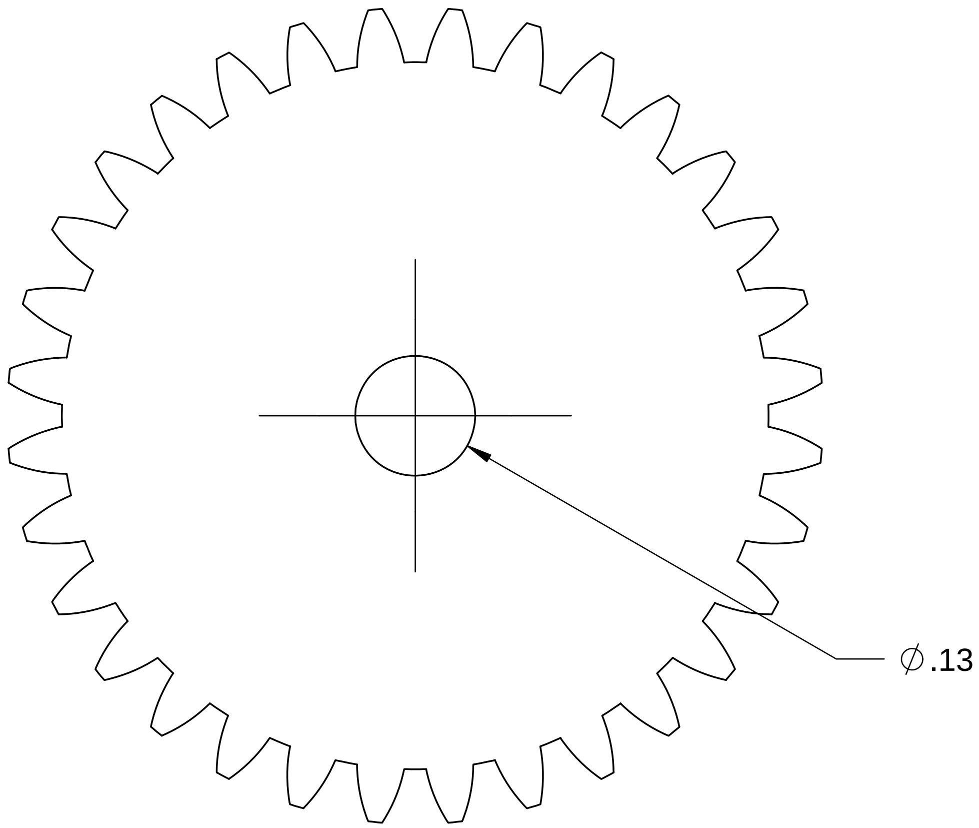


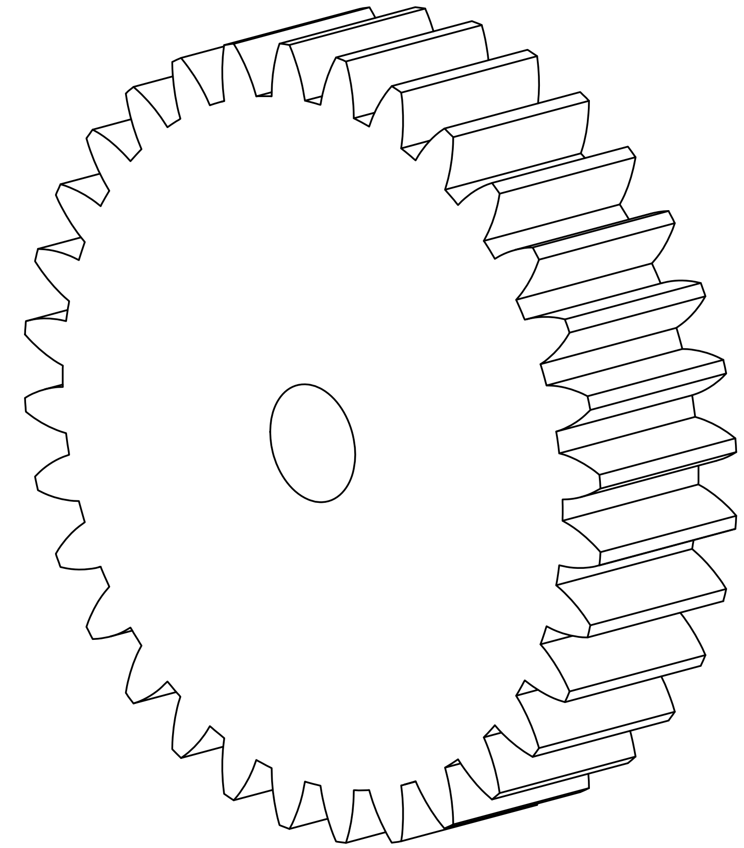


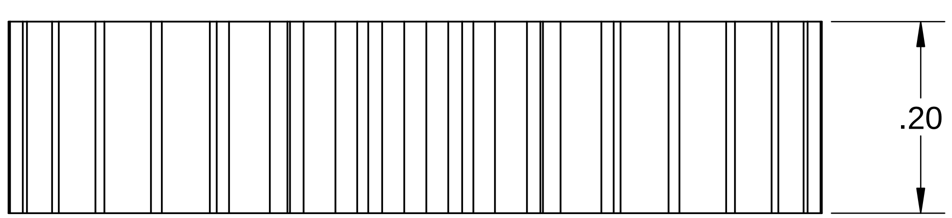
NOTES:
UNLESS OTHERWISE SPECIFIED:
1. ALL DIMS IN INCHES
2. TOLERANCES:
X.XX = ±0.01
ANGLES = ±1°
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED

Cal Poly Mechanical Engineering	Lab Section:	Assignment #	Title: 21 T Gear		Drwn. By:
ME 429 Spring 2018	Dwg. #: 4	Nxt Asb:	Date:	Scale: 5:1	Chkd. By: ME STAFF

32 T SPUR GEAR
DIAMETRAL PITCH: 40
TEETH NUMBER: 32
DIAMETER: 0.80 IN







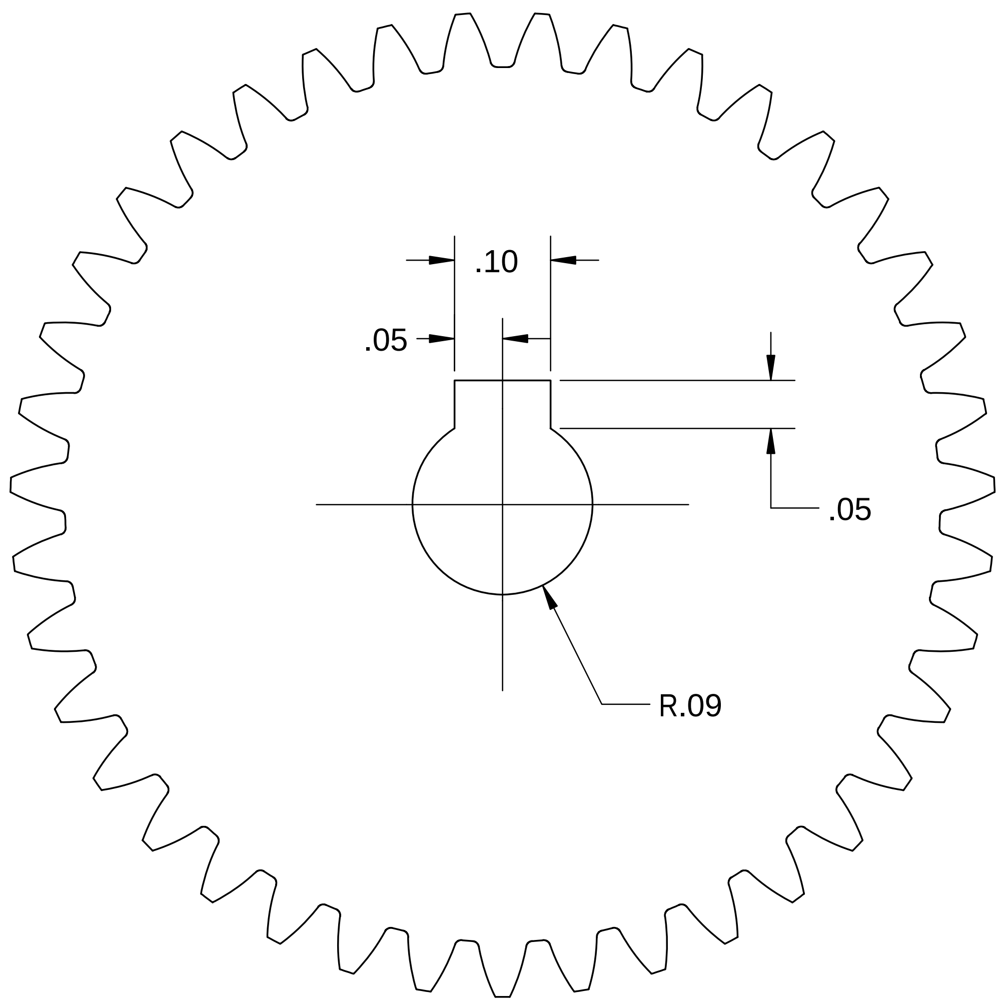
NOTES:

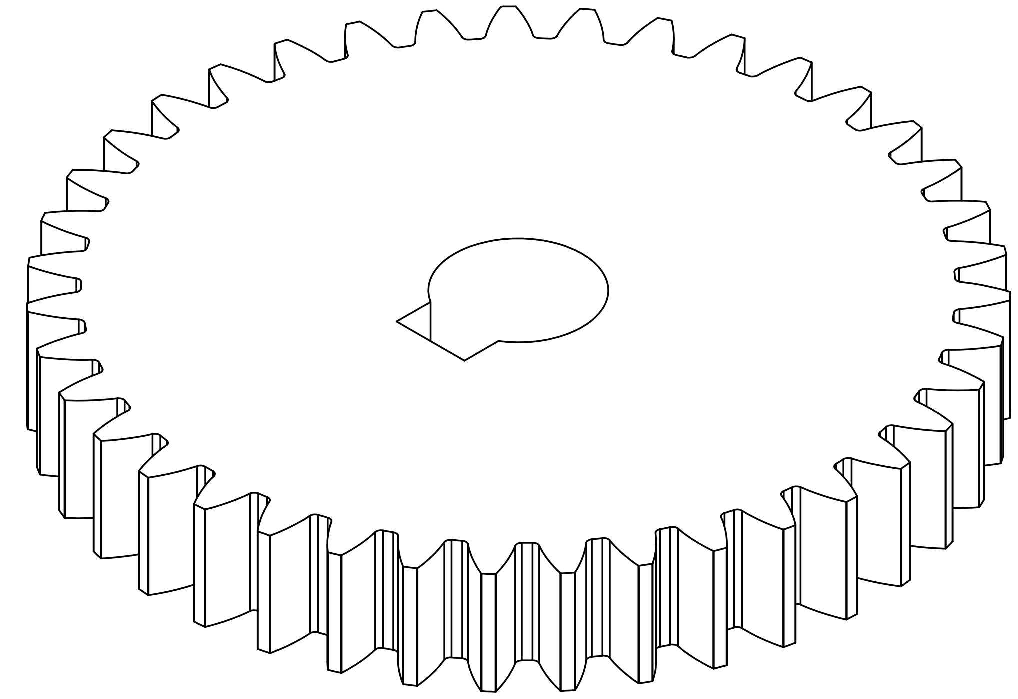
UNLESS OTHERWISE SPECIFIED:

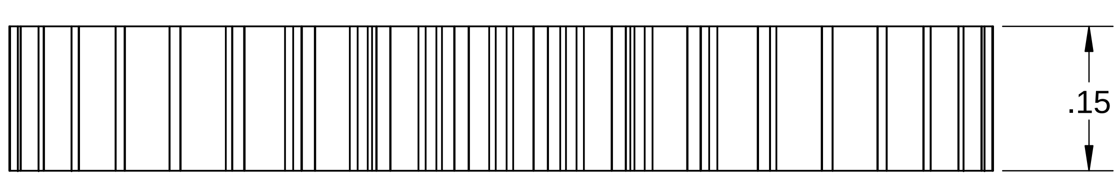
1. ALL DIMS IN INCHES
2. TOLERANCES:
   X.XX = ±0.01
   ANGLES = ±1°
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED

Cal Poly Mechanical Engineering	Lab Section:	Assignment #	Title: 32 T GEAR		Drwn. By:
ME 429 Spring 2018	Dwg. #: 4	Nxt Asb:	Date:	Scale: 5:1	Chkd. By: ME STAFF

39 T SPUR GEAR
DIAMETRAL PITCH: 40
NUMBER OF TEETH: 39
DIAMETER: .975 IN





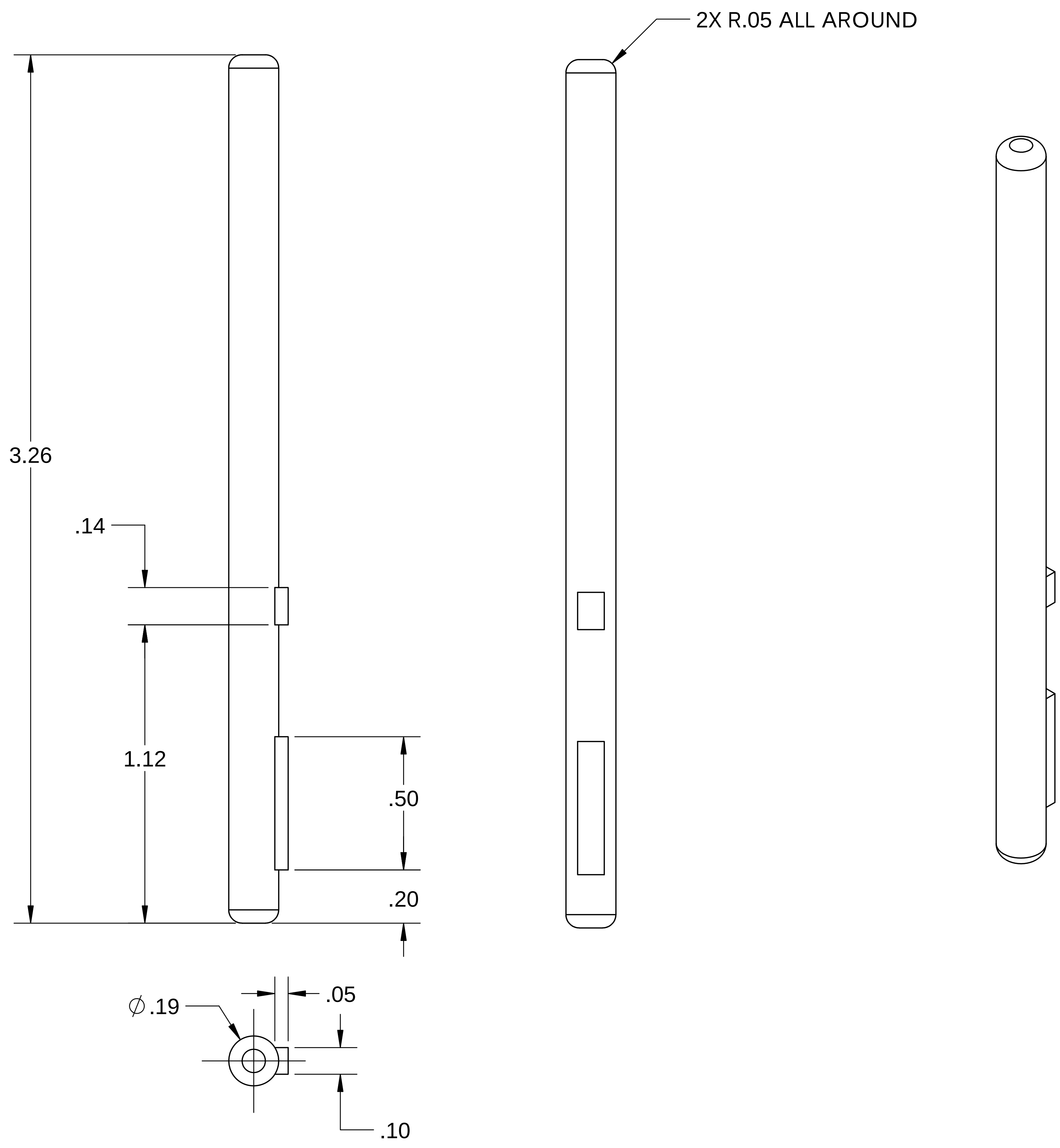


NOTES:

UNLESS OTHERWISE SPECIFIED:

1. ALL DIMS IN INCHES
2. TOLERANCES:
   X.XX = ±0.10
   ANGLES = ±1°
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED

Cal Poly Mechanical Engineering	Lab Section:	Assignment #	Title: 32 T Gear w/ Keyway	Drwn. By:
ME 429 Spring 2018	Dwg. #: 4	Nxt Asb:	Date: / Scale: 5:1	Chkd. By: ME STAFF

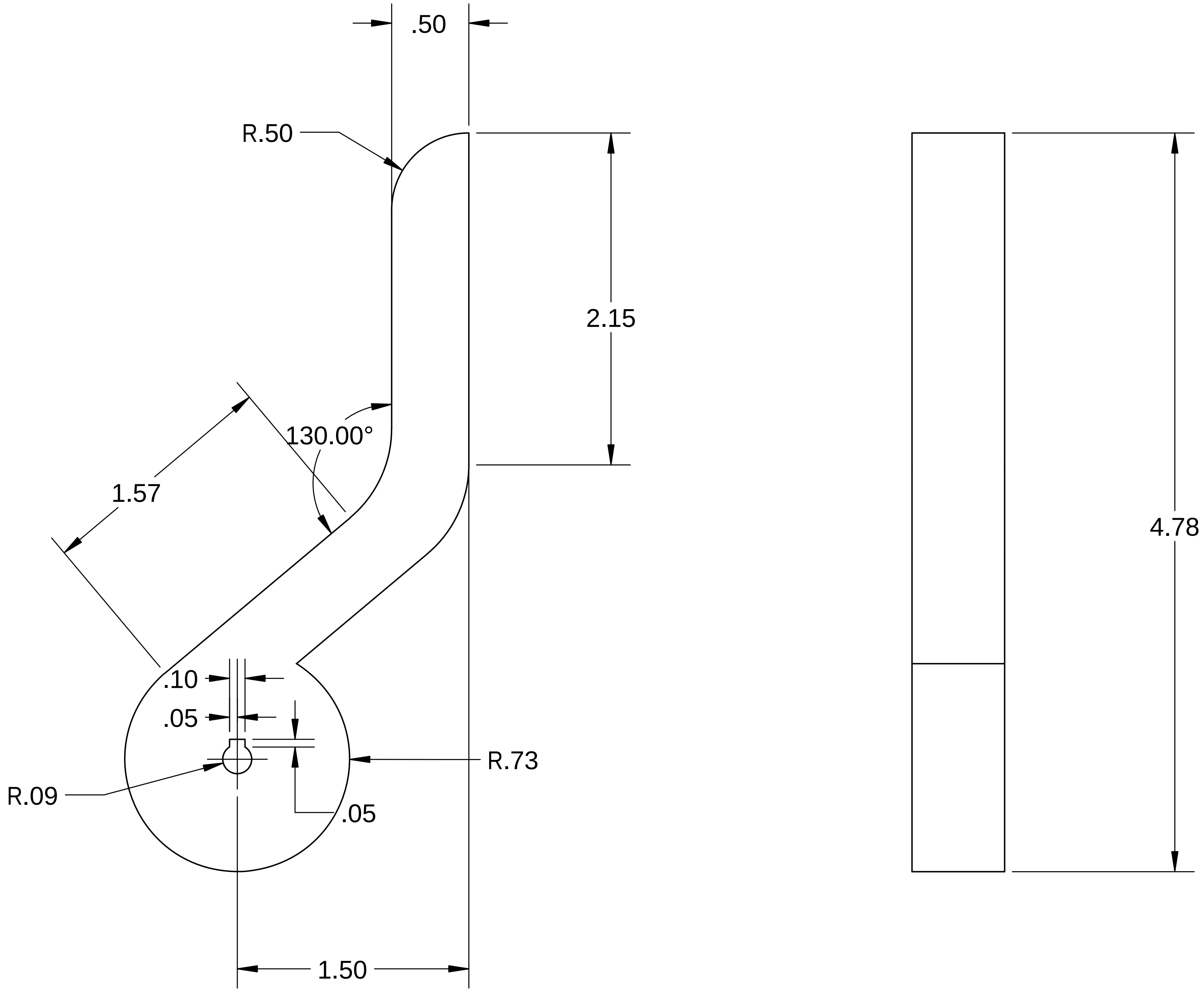


NOTES:

UNLESS OTHERWISE SPECIFIED:

1. ALL DIMS IN INCHES
2. TOLERANCES:
   X.XX = ±0.01
   ANGLES = ±1°
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED

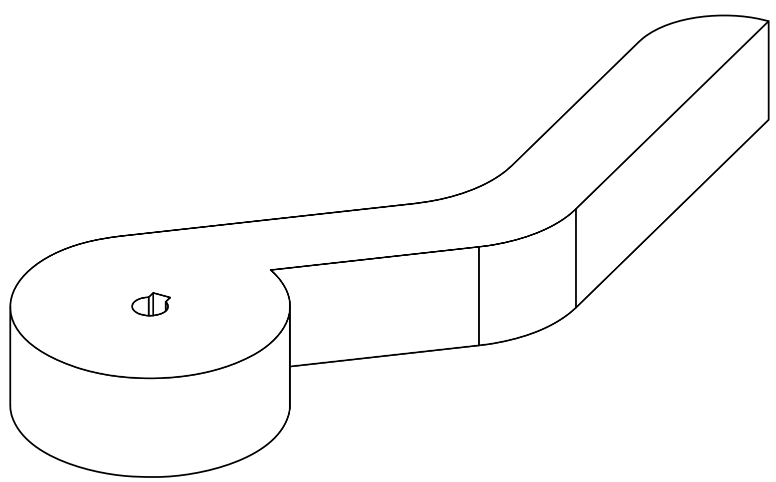
Cal Poly Mechanical Engineering	Lab Section:	Assignment #	Title: Shaft		Drwn. By:
ME 429 Spring 2018	Dwg. #: 2	Nxt Asb:	Date:	Scale: 2:1	Chkd. By: ME STAFF



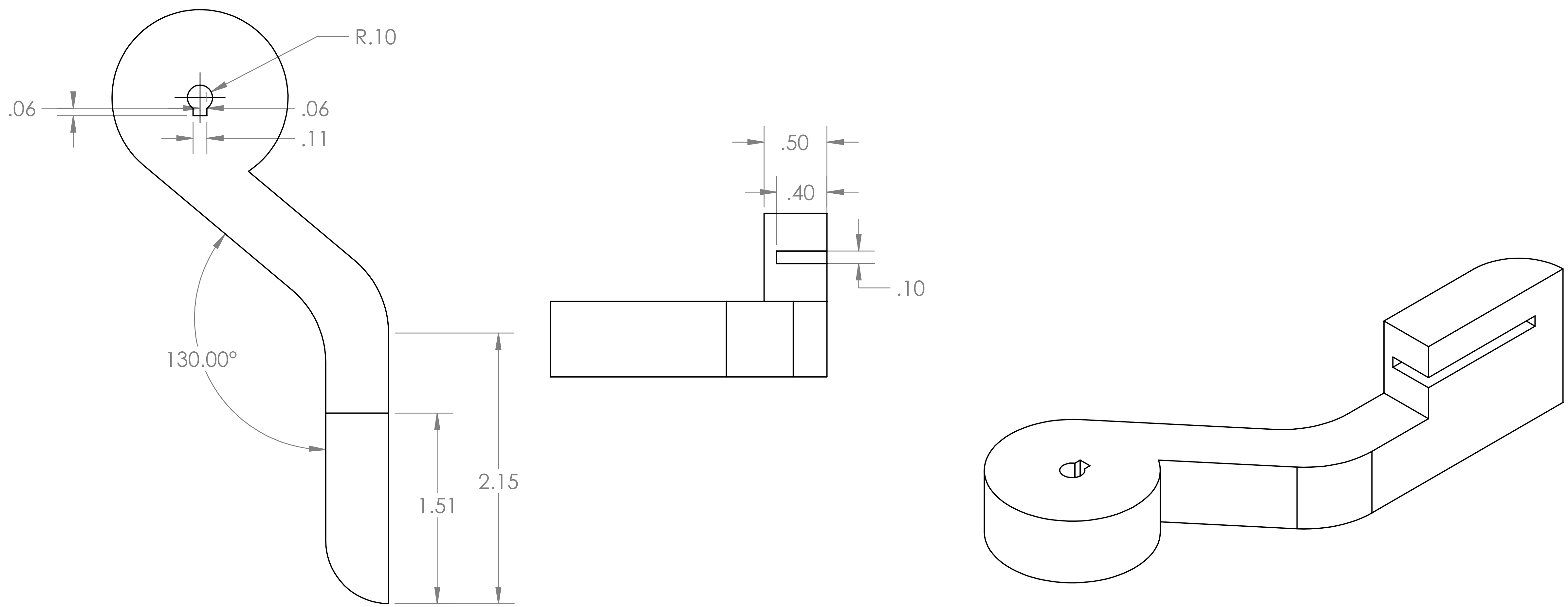
NOTES:

UNLESS OTHERWISE SPECIFIED:

1. ALL DIMS IN INCHES
2. TOLERANCES:
   X.XX = ±0.01
   ANGLES = ±1°
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED



Cal Poly Mechanical Engineering	Lab Section:	Assignment #	Title: Arm		Drwn. By:
ME 429 Spring 2018	Dwg. #: 3	Nxt Asb:	Date:	Scale: 1:1	Chkd. By: ME STAFF

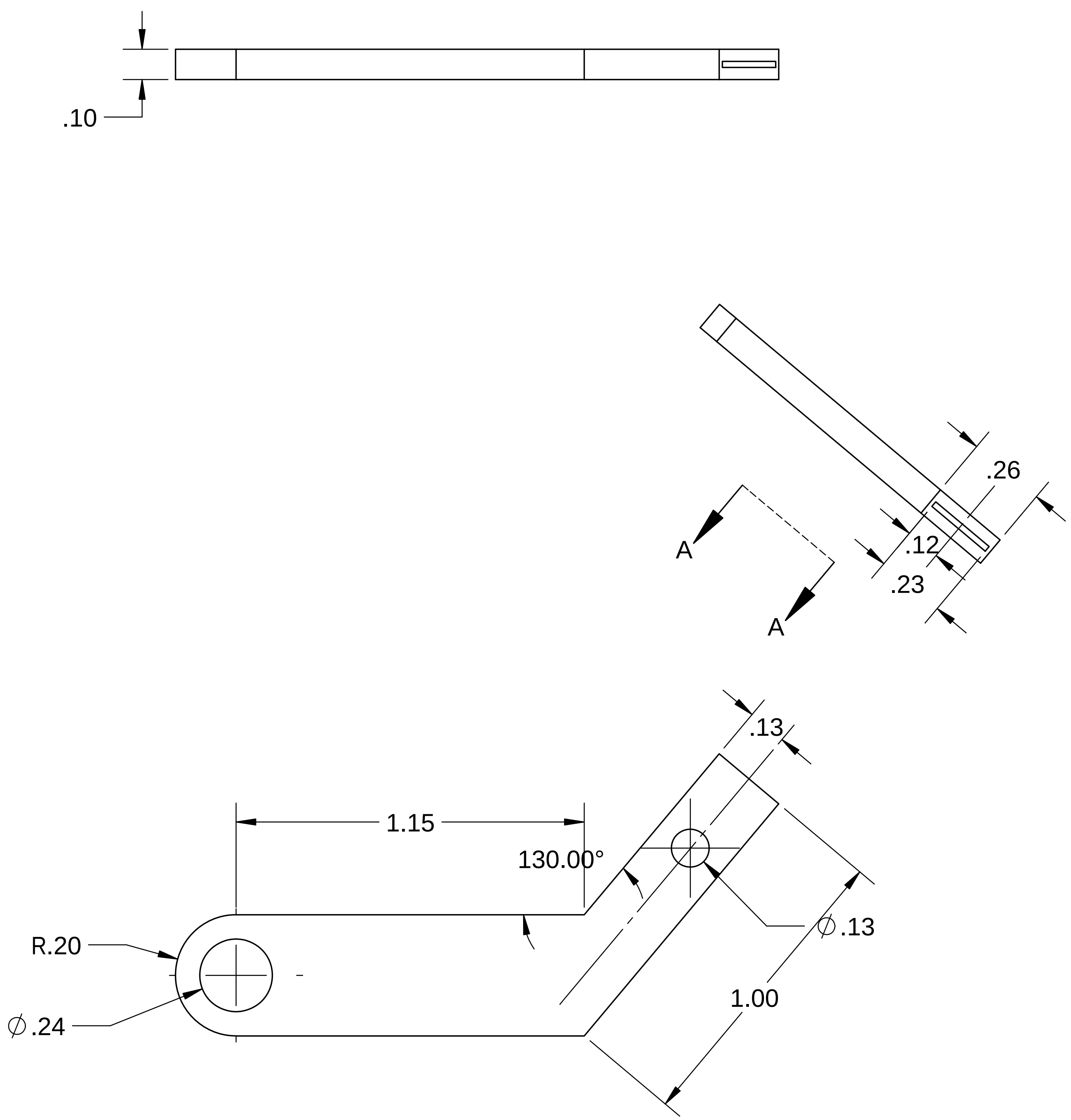


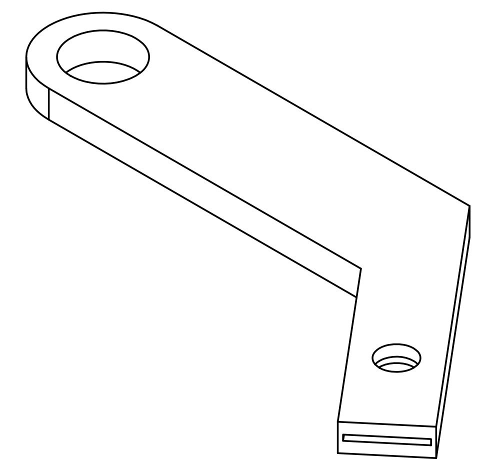
NOTES:

UNLESS OTHERWISE SPECIFIED:

1. ALL DIMS IN INCHES
2. TOLERANCES:
   X.XX = ±0.01
   ANGLES = ±1°
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED

Cal Poly Mechanical Engineering ME 429 Spring 2018	Lab Section:	Assignment #	Title: Arm		Drwn. By:
	Dwg. #: 3	Nxt Asb:	Date:	Scale: 1:1	Chkd. By: ME STAFF



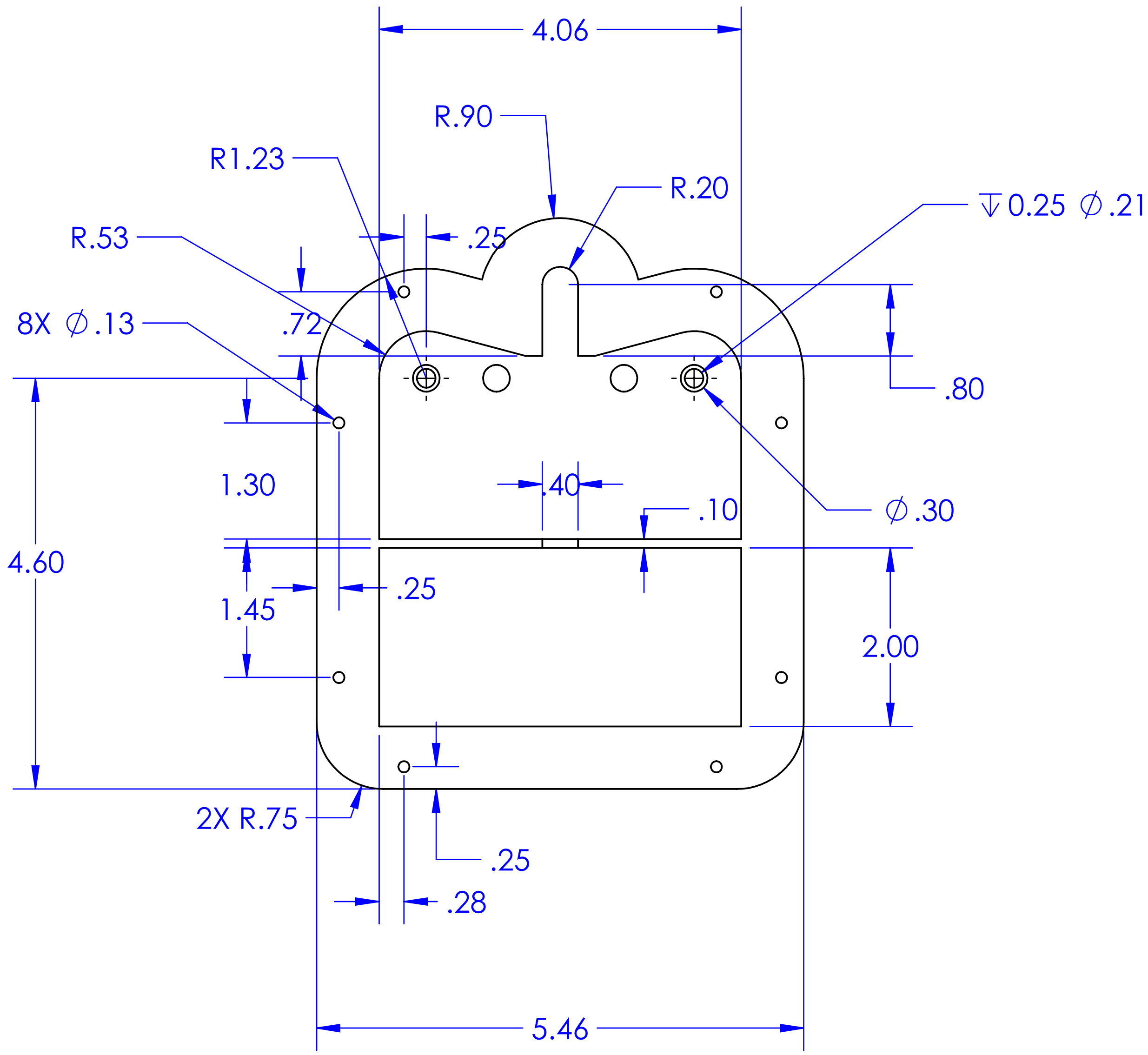


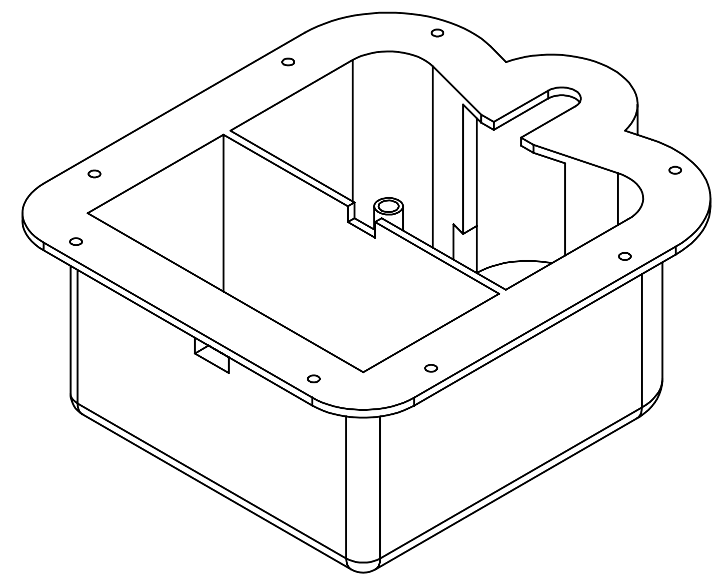
NOTES:

UNLESS OTHERWISE SPECIFIED:

1. ALL DIMS IN INCHES
2. TOLERANCES:
   X.XX = ±0.10
   ANGLES = ±1°
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED

Cal Poly Mechanical Engineering	Lab Section:	Assignment #	Title: Cutting Arm		Drwn. By:
ME 429 Spring 2018	Dwg. #: 7	Nxt Asb:	Date:	Scale: 2:1	Chkd. By: ME STAFF

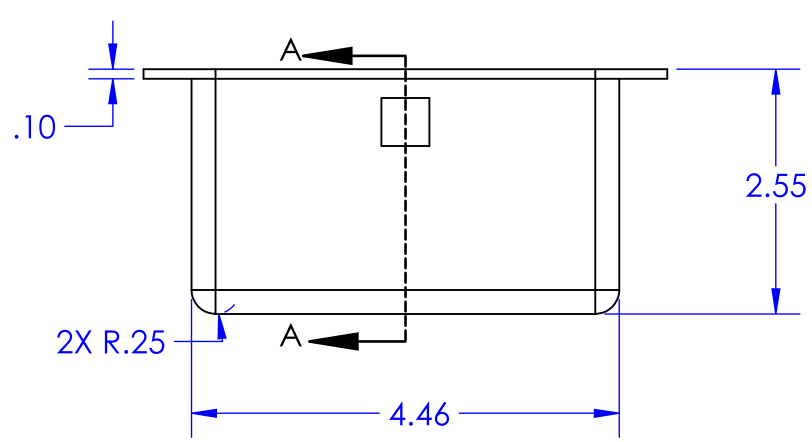


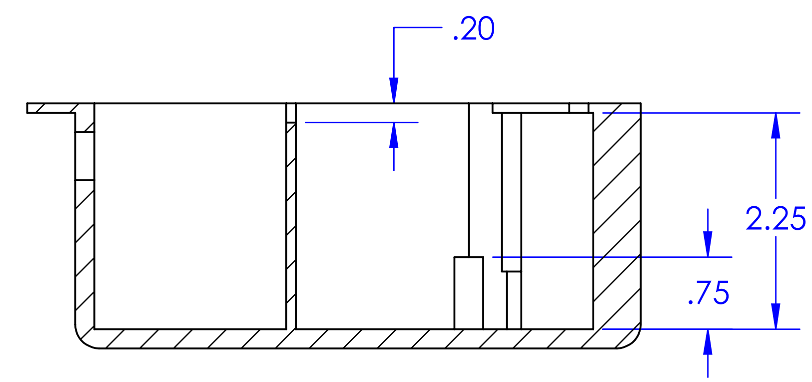


NOTES:

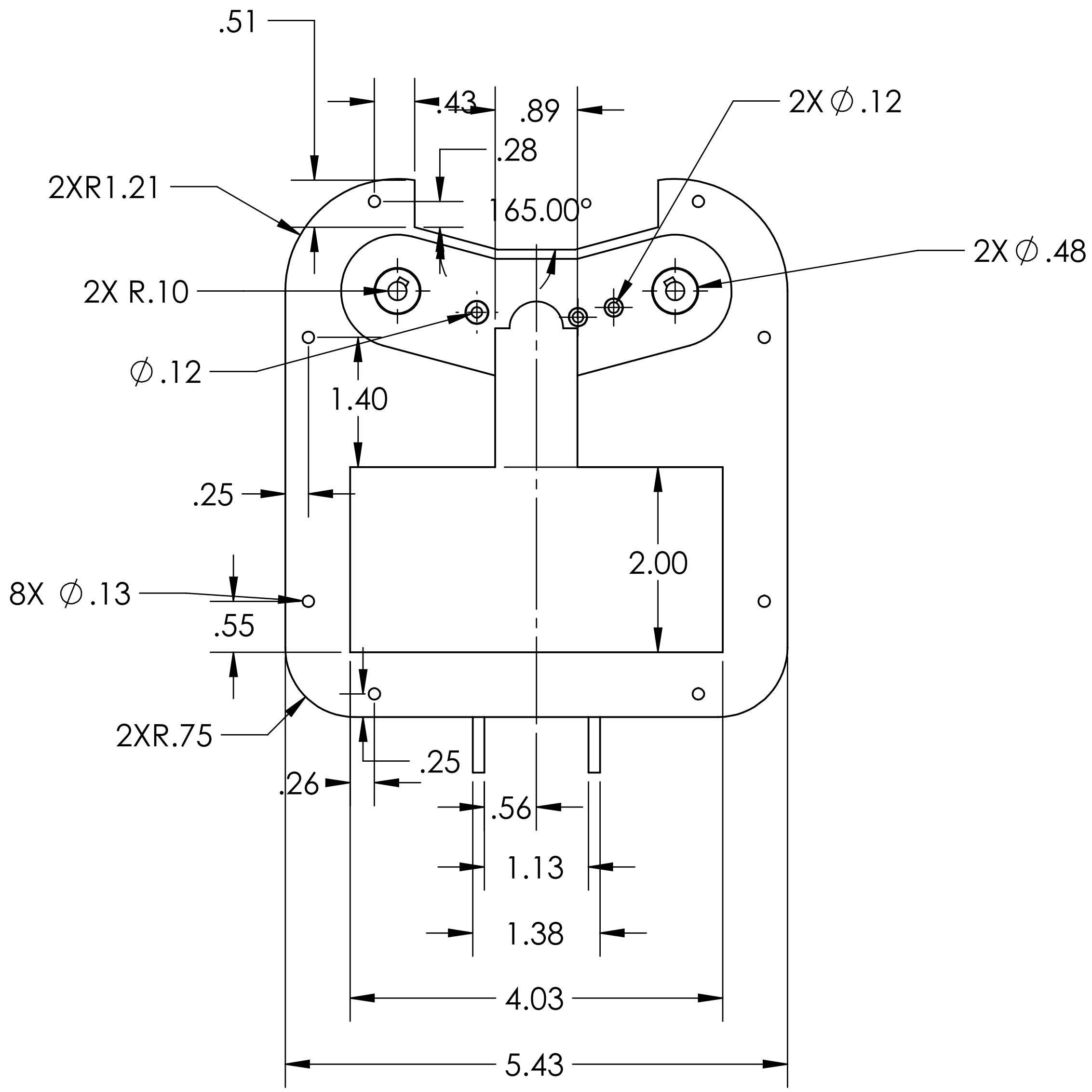
UNLESS OTHERWISE SPECIFIED:

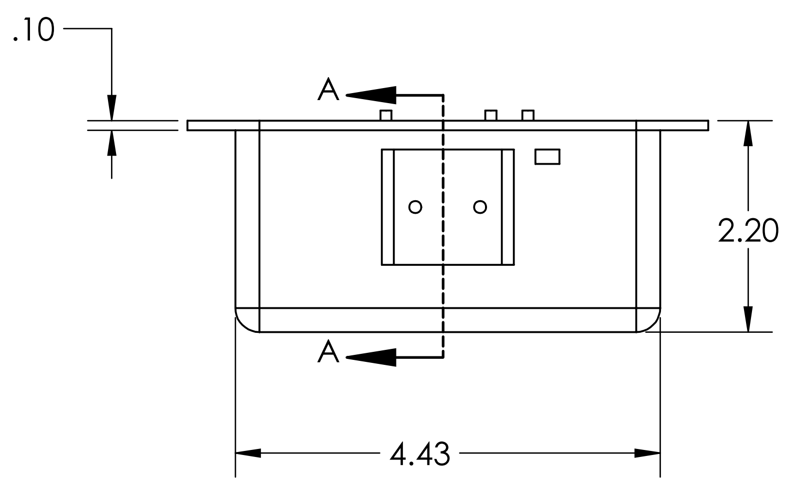
1. ALL DIMS IN INCHES
2. TOLERANCES:
   X.XX = ±0.01
   ANGLES = ±1 °
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED

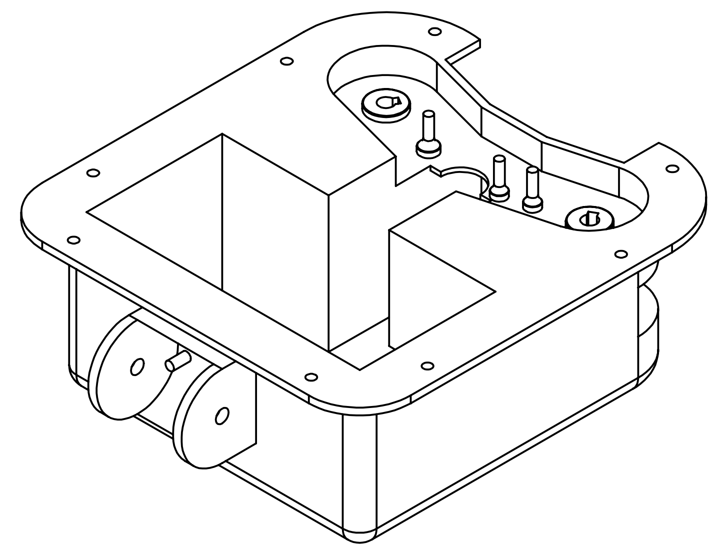


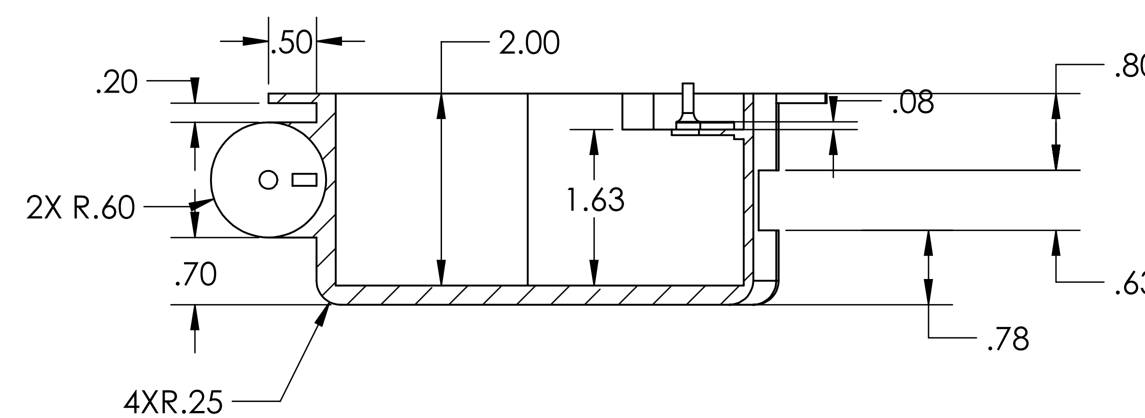


Cal Poly Mechanical Engineering ME 429 Spring 2018	Lab Section:	Assignment #	Title: Upper Housing		Drwn. By:
	Dwg. #: 6	Nxt Asb:	Date:	Scale: 1:1	Chkd. By: ME STAFF









NOTES:

UNLESS OTHERWISE SPECIFIED:

1. ALL DIMS IN INCHES
2. TOLERANCES:
   X.XX = ±0.01
   ANGLES = ±1°
3. MATERIAL: ABS PLASTIC
4. MANUFACTURING: 3D PRINTED

Cal Poly Mechanical Engineering ME 429 Spring 2018	Lab Section: 01	Assignment #	Title: Lower Housing		Drwn. By:
	Dwg. #: 5	Nxt Asb:	Date: 4-26-18	Scale: 1:1	Chkd. By:

Appendix J: Python Code used to Control DC Motor

```
# -*- coding: utf-8 -*-
"""
@author: James Jeffery
"""
import pyb
import time

class MotorDriver:
    '''This class implements a motor driver for the ME 405 board'''

    def __init__ (self,echo=True,pin1=pyb.Pin.board.PA0, pin2=pyb.Pin.board.PA1,
mCPUPIN=pyb.Pin.board.PC1, timerChan=5):
            # removed input -> mCPUPIN=pyb.Pin.board.PA10, ADD BACK IN IF FAILS
        '''Creates a motor driver by initializing GPIO pins and turning the motor off for safety. '''
        self.echo = echo
        pinENA = pyb.Pin(mCPUPIN, pyb.Pin.OUT_OD, pyb.Pin.PULL_UP)
        ##Sets pin IN1A to output
        self.pinIN1 = pyb.Pin(pin1, pyb.Pin.OUT_PP)
        ##Sets pin IN2A to output
        self.pinIN2 = pyb.Pin(pin2, pyb.Pin.OUT_PP)

        tim5 = pyb.Timer(timerChan, freq=20000)
        ##Creates a timer channel to allow PWM output on pin IN1A
        self.ch1 = tim5.channel(1, pyb.Timer.PWM, pin=self.pinIN1)
        ##Creates a timer channel to allow PWM output on pin IN2A
        self.ch2 = tim5.channel(2, pyb.Timer.PWM, pin=self.pinIN2)

        pinENA.high()

        self.pinIN1.low()
        self.pinIN2.low()
        if echo:
            print ('Creating a motor driver')
```

```
    def set_duty_cycle (self,level):
        '''This method sets the duty cycle to be sent to the motor to the given level. Positive values
cause torque in one direction, negative values in the opposite direction.
        @param level A signed integer holding the duty cycle of the voltage sent to the motor'''


        if level < 0:
            level = level * -1
            if level > 100:
                level = 100
            self.ch1.pulse_width_percent (0)
            self.ch2.pulse_width_percent (level)
            level = level * -1
        else:
            if level > 100:
                level = 100
            self.ch2.pulse_width_percent (0)
            self.ch1.pulse_width_percent (level)

        if self.echo:
            print ('Setting duty cycle to ' + str(level))


    def cut(self,level):

        self.ch2.pulse_width_percent(level)

        time.sleep_ms(150)

        self.ch2.pulse_width_percent(0)

        time.sleep_ms(750)

        self.ch1.pulse_width_percent(level)

        time.sleep_ms(150)

        self.ch1.pulse_width_percent(0)
```

```
def jogfow(self):

    self.ch2.pulse_width_percent(25)

    time.sleep_ms(50)

    self.ch2.pulse_width_percent(0)

def jogback(self):

    self.ch1.pulse_width_percent(25)

    time.sleep_ms(50)

    self.ch1.pulse_width_percent(0)
```

Appendix K: Python Code used to Control Servo Motor

```
# -*- coding: utf-8 -*-
"""
@author: James Jeffery
"""
import pyb
import time

class ServoController:
    """ This class is used to conrol the D-980TW servo used for our senior project
        The servo is 180deg control.
        pulsewidth of 500 = 90deg one way
        pulsewidth of 2500 = -90deg
    """
    #self.currPW = 0
    def __init__ (self, pwmPin=pyb.Pin.board.PA5, tim = 2, ch = 1, initialSig=1500):
        ''' Initialization
            Requires one pin for the signal. Ground must be connected to a comm
            ground on the pyboard. Signal pin requires timer function on the same pin
        '''
        self.sigPin = pyb.Pin(pwmPin, pyb.Pin.OUT_PP)
        self.timer = pyb.Timer(tim, prescaler=79, period=19999)
        self.chan = self.timer.channel(ch, pyb.Timer.PWM, pin=self.sigPin)

        self.PW = 1500
        self.nextPW = 1500

        self.initPos()

        print('Servo Motor Initialized')

    def initPos(self):
        '''
        '''
        self.nextPW = 1500

        while self.PW != self.nextPW:
```

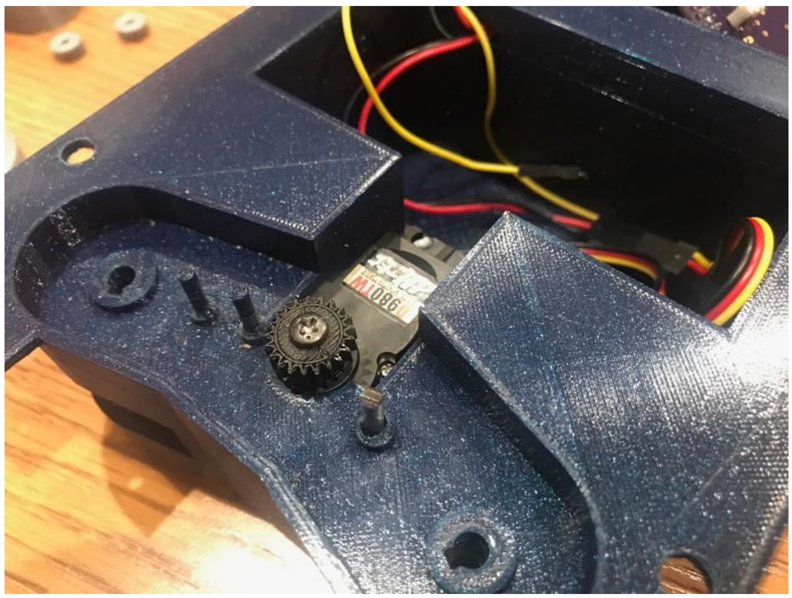
```
            # long expression used to determine sign
            self.PW = self.PW + 50 * int((self.nextPW-self.PW)/abs(self.nextPW-self.PW))
            self.chan.pulse_width(self.PW)
            time.sleep_ms(20)
        self.chan.pulse_width(1500)
        self.PW = 1500

    def lowPos(self):
        '''
        '''
        self.nextPW = 1000

        while self.PW != self.nextPW:
            #
            self.PW = self.PW - 50
            self.chan.pulse_width(self.PW)
            time.sleep_ms(20)
        self.PW = 1000
        self.chan.pulse_width(self.PW)

    def highPos(self):
        '''
        '''
        self.nextPW = 1600
        while self.PW != self.nextPW:
            self.PW = self.PW + 50
            self.chan.pulse_width(self.PW)
            time.sleep_ms(20)
        self.PW = 1600
        self.chan.pulse_width(self.PW)

    def Open(self):
        '''
        '''
        self.nextPW = 1100

        while self.PW != self.nextPW:
            self.PW = self.PW - 50
            self.chan.pulse_width(self.PW)
```

```
            time.sleep_ms(20)
        self.PW = 1100
        self.chan.pulse_width(self.PW)

    def Close(self):
        '''
        '''
        self.nextPW = 1700
        while self.PW != self.nextPW:
            self.PW = self.PW + 50
            self.chan.pulse_width(self.PW)
            time.sleep_ms(20)
        self.PW = 1700
        self.chan.pulse_width(self.PW)

    def goupto(self,signal):
        '''
        '''
        self.nextPW = signal

        while self.PW != self.nextPW:
            self.PW = self.PW + 50
            self.chan.pulse_width(self.PW)
            time.sleep_ms(20)
        self.PW = signal
        self.chan.pulse_width(self.PW)

    def godownto(self,signal):
        '''
        '''
        self.nextPW = signal
        while self.PW != self.nextPW:
            self.PW = self.PW - 50
            self.chan.pulse_width(self.PW)
            time.sleep_ms(20)
        self.PW = signal
        self.chan.pulse_width(self.PW)
```

<u>Appendix L</u>: Assembly Guide/Operator's Manual

Hardware Assembly

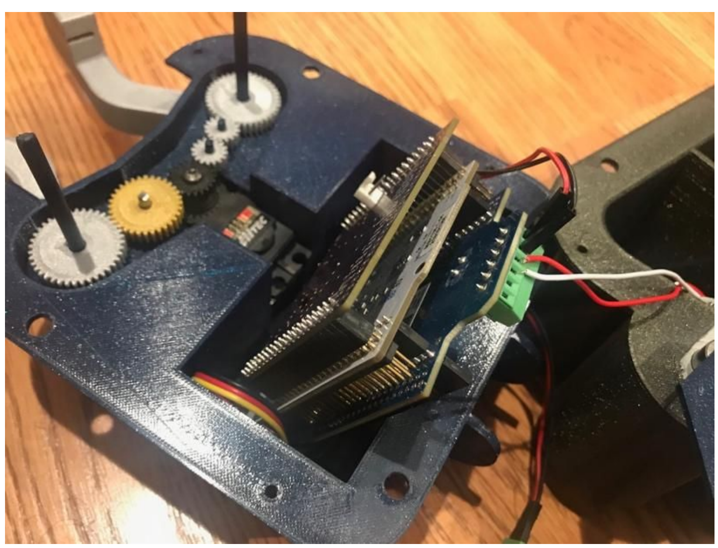
To assemble the strawberry picker end effector, first begin by placing the servo motor inside the designated slot in the bottom housing. Place the motor flat on the base of the bottom housing, with the end with output shaft facing the front of the bottom housing. Slide the servo motor into its designated slot, making sure the rounded edge on the motor presses up against the rounded portion of the overhang on the bottom housing. The motor should fit snugly inside and should not slide back and forth. Make sure the wiring for the motor is wrapped around the right side, so when it is slid in the wire will fit inside the small slot to the right of the motor.



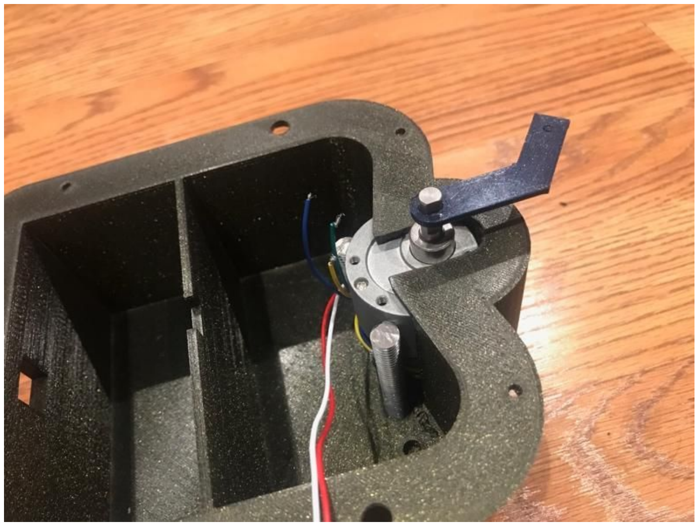
After the servo motor is placed, the gear train should begin to be assembled. The 32 tooth gear should be placed on the intermediate pin on the left, and the 16 teeth gears should be placed on the two intermediate pins on the right.

The next gears of the gears to be placed are the outer, 39 tooth gears. These are the gears that drive the shaft and thus the grabber arms. The first step is to place the arms in their slots, taking care to line up the holes in the arms with the holes in the bottom housing. Once these holes are lined up, the shafts should be slid through. The key of the shafts will need to be lined up with the keyway in the arms and the bottom housing. After the shafts are in place, the outer gears should be slid on over the shaft, with the keyway lining up with the key in the shaft. Once the outer gears are in place, the pinion gear should be placed on top of the output shaft of the servo motor and tightened in place with a screwdriver.

The next step will be to attach the electronics. The motor controller and motor driver need to be placed inside the large slot at the back of the bottom housing. These will be plugged in to the servo motor and the DC motor.



With the electronics in place, the top housing will now need to be used. The DC motor should be placed inside the top housing, and the set screws should be placed to hold the motor in place. The shaft of the DC motor should be protruding out of the housing. The cutting arm should be slid onto the shaft, and should be tight enough that it does not wiggle. The X-Acto knife should then be slid into the opening at the end of the cutting arm and held firmly.



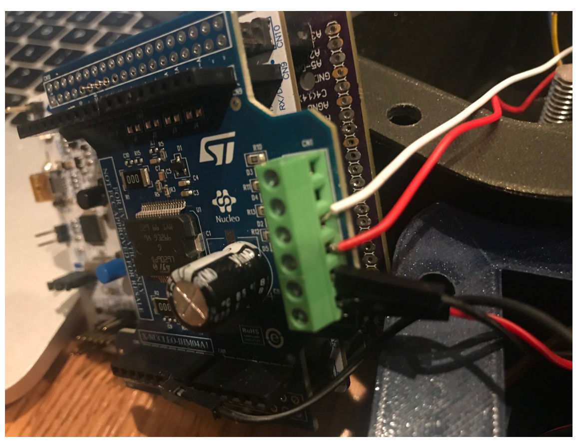
At this point, the DC motor should be hooked up to its electronics. The 5 wires (2 black and 3 red) that need to be attached to exterior power sources should be fed through the small opening in the back of the bottom housing. These cables should then be hooked up to their power sources. The USB cable should be plugged into the controller, fed out the back through the hole in the top housing, and plugged into a computer.

The top housing should then be placed on top of the bottom housing. Be careful to line up the shafts with the two slots in the top housing. Once the top housing is in place, it should be secured with bolts on the outside flange of the assembly.

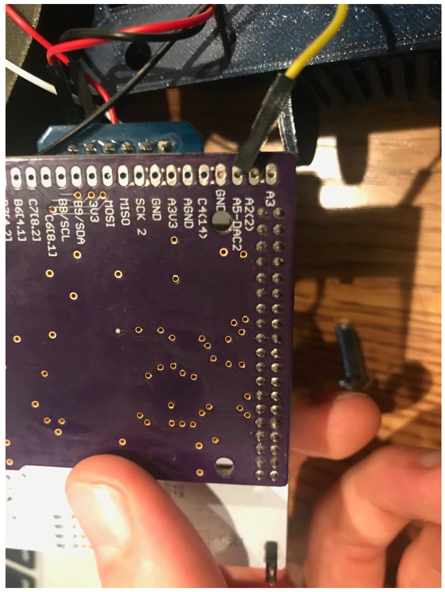
The assembly is now complete and is ready to be used.

Electronics Assembly

To assemble the electronics of the end effector begin by connecting a mini-USB cable from a computer to the bottom-most board on of the electronic system. This allows for microcontroller to receive commands from the computer. Next connect a 12V power supply into the two screw terminals on the motor driver (top-most board of the electronic system) labeled "+" and "-".



Next is to connect the DC motor to the motor driver. Take the red wire from the DC motor and connect it to the positive B screw terminal and connect the white wire into the negative B screw terminal. You can ignore the other 4 wires from the DC motor, they are not needed for our electronics.



The last step is to complete the wiring for the servo motor. Connect the red (+) and black (GND) wires from the servo motor to a 6V power supply. The yellow wire connects to a pin labeled "A5" on the bottom-most board of the electronics. This allows for pulse-width signals to be sent to the servo motor which control its position.

The electronics of the end effector are now fully assembled.

<u>Appendix M</u>: Bill of Materials

Item	*Amount*	*Price*
HiTec D-980TW Servo Motor	1	$177.00
DC Motor	1	$28.75
Microcontroller / Motor Driver	1	$37.48
Breadboard	1	$7.96
Wires	1	$6.98
Shears	1	$29.95
Power Supply	2	$30.00
Total Price		**$318.12**