TEAM 01

dextera

FINAL REPORT

MEAM 445/446, MEAM SENIOR DESIGN

SEPT 2018 - MAY 2019

Abstract

Third-hands are common tools used by professionals across multiple industries to hold a work object while the users' hands are busy manipulating the object. However, these tools are manually adjusted, which requires a user to put down their tools, break their workflow to adjust the device, and potentially risk damaging their work in the process. Jewelers are especially affected by this problem; the stakeholders surveyed will spend approximately half of their day soldering, with the majority of that time spent simply adjusting. Dextera is the first voice-activated, robotic, third-hand tool specifically optimized for jewelry soldering. The device allows users to overcome the counterintuitive and disruptive nature of using a third-hand tool by giving them hands-free control over a the position and orientation of a work object.

The arm of Dextera consists of three joints, one linear joint, which allows for up and down movement of the arm, and two rotational joints, which allow for full 360 degree rotation of the arm as well as tilting at the wrist. The device features a dual gripper system to allow users to hold all types of desired objects from one or two contact points. Dextera is packaged in a protective yet sleek exterior casing which shields the internal components from heat, dirt, and other debris while simultaneously ensuring a consumer-ready look. The compact design of the device ensures it is able to fit comfortably on a jeweler's crowded worktable.

The team was successfully able to meet user needs, creating a product that is efficient, intuitive, safe, and enjoyable. Dextera reliably holds all jewelry objects specified by stakeholders - supporting the weight of each object while ensuring no damage to the piece. Dextera responds accurately to users' commands and is able to move to a new, desired position in under 2 seconds after the user has finished speaking.

In the following report, the team seeks to describe this Senior Design Project in full, including analysis of user needs, definition of system objectives, concept down selection, design and realization decisions, efforts to validate and test the device, and resources used in the making of Dextera.

Table of Contents

1. Executive Summary

A typical jeweler spends 50% of his or her day soldering.[1] Further, 20% of this time soldering is spent adjusting the object on which he or she is working or switching between objects. To perform these manual tasks, the jeweler is forced to put down the soldering blow torch, sacrificing both the consistency of heat on the object, as well as time that could be spent completing the solder. Further, the manual labor involved in adjusting objects of such small scale causes hand cramping and could eventually lead to carpal tunnel.[2] Currently, there are no systems on the market that allow the jeweler to perform these tasks hands-free. Consequently, there is a strong need for a system that affords jewelers a continuous workflow, eliminates hand cramping and manual labor, and saves time when soldering.

Adhering to this need, the team devised a system that is physically characterized by a 5-degree-of-freedom robotic arm. With steel linkages and a sturdy base that provide both heat-resistance and loading support, the system is optimized for soldering. Additionally, it has two independent grippers, that can be used together or individually, allowing flexibility in surface area exposure, and thereby eliminating the need for the jeweler to constantly adjust the object to work on different areas. Additionally, the grippers are modular in design to accommodate for a variety of potential work objects. Each of the five degrees of freedom act as follows: one linear vertical slider for the user to adjust height of the object, one 360-degree rotational joint in the wrist for the user to adjust based on the orientation of the solder, an additional panning joint in the wrist for the user to adjust height and orientation of the object with respect to the horizontally-rigid arm, and two clamshell-like grippers that open and close to expose different areas of the object. In order to address the necessary hands-free characteristic, the system is voice controlled, and responds precisely to users' commands for movements and adjustments within the workspace. The full system is adaptable within the workspace and thus can be placed on either side of the workbench to serve both left-handed and right-handed users. Finally, in order to accomodate for necessary actuation and language processing computation, the system is powered by a 6 V power source, which is connected to a wall outlet.

In order to realize this system, the team first created proof-of-concept prototypes for each subsystem, including the mechanical components of the arm, the grippers, the voice control software, and the motor controls. Then, a preliminary integrated prototype was constructed with minimum functionality, which served as an initial validation of full system feasibility. Finally, the final device prototype of Dextera was created. It included a wrist with a 180-degree servo, protected by a stainless steel heat shield that doubled as a motor housing for the two gripper gearmotors. From the wrist, the modular grippers extended outward to the end effector, where the work objects are held. The arm was manufactured out of hollow aluminum tubing and mounted to the necessary motors via pillow blocks and clamps. At the base of the arm lies a continuous rotation servo that controls the 360-degree rotation of the arm. The continuous rotation servo is mounted to the linear slider, which is operated by a gearmotor attached to a lead screw drive. Finally, the entire body of the system is encapsulated by a sheet metal enclosure that incorporates a vertical sliding mechanism, as to not inhibit the vertical range of motion of the device.

The device utilizes a Raspberry Pi computer to perform the necessary speech-to-text and motor control computations. Motor control techniques including Servo object methods, direct PWM control, and proportional integral derivative (PID) control are employed in the operation of the device. Moreover, an overview of the software functionality of Dextera is defined as follows: first, the user voices a command, which Dextera then converts to text. The text is then processed and generates a new position for the arm to move to via an inverse kinematic systemic model. This new position is then sent as a motor command to update the physical position of the arm. With the current state of the arm fully updated, Dextera then continuously listens for its next voiced command.

In order to validate the functionality and usability of the product, the team performed a variety of tests and simulations. Namely, thermal and structural analysis simulations confirmed that Dextera could withstand the necessary heat and loading during usage and gripper testing validated that the grippers could hold the variety of objects worked on by jewelers. Additionally, software efficiency tests ensured the device could effectively improve the efficiency of soldering tasks and usability testing ensured the device was useful and helpful to jewelers, providing the full dexterity and ranges of motion necessary for soldering.

After validating the device functionality, the team reflected on initial goal parameters set forth and compared the final system form of Dextera against said parameters. With most primary goals met, the team focused future work and recommendations on better designing the product for the consumer space - most notably, adapting the design for a wider audience, further improving on voice recognition software, and incorporating gripper force feedback into the device.

Lastly, the monetary budget and resources used in the creation of Dextera, as well as the price and marketing for this product should it be brought to market, are both outlined. The following comprehensive report serves as a thorough delineation of and a tribute to the efforts put forth in creating Dextera throughout the entirety of the Senior Design course.

2. Statement of Roles

Alexis Mitchnick
Mechanical and Software Engineer
amitchn@seas.upenn.edu
Alexis was mainly responsible for the integration of the different software and hardware subsystems. She contributed to the speech to text code and the motor controls code. She was also responsible for contributing to the motor and voice control debugging. Alexis also aided in the manufacture of various prototypes and final assembly. She also did a full robotic analysis and simulation in Matlab along with Danielle.

Danielle Gelb
Mechanical Subsystem and Validation Engineer
danigelb@seas.upenn.edu
Danielle was mainly responsible for the user research and prototyping. She contacted and communicated with jewelers and visited them at their workshops, conducting interviews to guide our design choices and validate our final system. She also contributed to the design and CAD of the gripper and wrist subsystems. Danielle also laser cut and 3D printed many iterations of the prototype, as well as manufacture the final form of the grippers. She also did a full robotic analysis and simulation in Matlab along with Alexis.

Cheryl Feig
Mechanical Subsystem and Validation Engineer
cfeig@seas.upenn.edu
Cheryl was the project manager and also responsible for the thermal analysis and gripper subsystem. She created Matlab and Comsol simulations to validate the grippers geometry and ability to perform in the necessary conditions. She also assisted Dani with the user research and recorded jewelers' soldering to help validate the final system form. Cheryl also assisted with the final assembly and the formatting of all team deliverables and presentations.

Liam Cook
Fabrication and Electrical Subsystems Engineer
liamcook@seas.upenn.edu
Liam was primarily responsible for the manufacture and controls of the final system. He manufactured all of the metal parts for the final system form and installed the motors. He also wrote the motor code and helped Max with the integration of the code with the speech to text software. Liam also created much of the system in Solidworks and ran FEA tests on it.

Max Hartman
Software and Controls Engineer
mhartman@seas.upenn.edu
Max was the lead software engineer for the team. He was in charge of all software system design choices and implementation. He did all the speech to text research and tested the different platform options. He also did the debugging and microphone configuration. Max worked with Liam and Alexis to integrate the hardware and software subsystems.

Ethan Bradlow
Mechanical Engineer
bethan@seas.upenn.edu
Ethan was primarily responsible for prototyping and market research. He was responsible for the budget and determining the final price of the system from his research and analysis. He also performed user research and interviewed potential users to determine viability. Ethan laser cut iterations of prototypes and assisted in the assembly. He also aided in the FEA and strength analysis of the system.

3. Background

For many people that work with their hands, a great deal of unnecessary time and effort is wasted due to the need to put tools down periodically to adjust clamping instruments. This affects a wide range of professionals, such as engineers, electricians, mechanics, craftsmen, jewelers, and researchers. There are tools to address this general need as well as tools for specific industry use cases. Such tools are referred to as helping hands or third hands. These devices are typically adjustable, with the ability to hold one or more points of an object (or objects) in place so the user can work on them. These tools are especially useful when the user requires very specific orientations or needs to hold multiple objects in place. These third hands come in a variety of shapes and sizes for this reason.

Jewelers in particular require access to the piece they are working on from a variety of angles and locations. The team has consulted multiple stakeholders who work with jewelry to confirm the existence of this problem. A visit to Jennifer Green Custom Jewelry in Philadelphia was especially useful for the team to identify the specific jewelry work that has room for improvement. The team was lucky enough to view Jen Green working on a piece and engage in dialogue with her regarding pain points. She identified soldering as the primary task in which she is limited by only having two hands available to work.

In order to address the drawbacks of this time consuming and cumbersome task, the team built a voice-controlled robotic third arm to hold and reposition the piece while the jeweler is soldering. Jen confirmed that this would indeed streamline her soldering work, which she says consumes roughly half of her day.[1] Based on this feedback, the team strongly believes that there is significant demand for the product. The potential for substantial time reduction and improved user experience could make the arm a must-buy for jewelers, especially those that work with a large volume of pieces.

The arm represents a substantial technological upgrade over the existing solution: the metal vise shown below - a simple metal structure that resembles a pair of tweezers. The existing static metal clamp is simply inefficient for jewelry soldering, requiring the jeweler to put down at least one of the tools in order to tilt, rotate, or reposition the piece. This action represents significant wasted time in the long run and disrupts work flow. There are currently no vises on the market that integrate electronics, let alone voice-control technology. Thus, the team sees an enormous opportunity for market disruption.

Figure 2.1: Common third hand used by jewelers for soldering

The jeweler profession in America is under enormous threats from both increasing imports and rising productivity of individual jewelers (reduction in workforce size). Thus, strong competition is expected in the jeweler labor market.[3] Although demand for customized and boutique jewelry is strong, it is often difficult for independent designers to establish themselves in the market. The team believes that the increased productivity provided by the arm would represent a valuable investment for jewelry workers to stay ahead of this increasing competition.

4. Objectives

Utilizing the feedback from user research amongst several professional jewelers, both in in-person interviews and virtual surveys, as well as online jewelry-making expert articles and video tutorials,[1,2] a set of key system-level characteristics were outlined, for which design goals and necessary technical specifications for the system were devised. The goal metrics defined for each system-level characteristic are based on both the foundational jewelry-clamping needs expressed in user interviews as well as any additional value that could be feasibly added to the system, should time and resources permit. Additionally, while jewelers were the user kept top of mind, the characteristics examined were also considered in their adaptability for a wider range of use cases. Consequently, considering jewelers' basic needs and common pain points, as well as general device usefulness, the goals for each characteristic were defined. In addition to these user-driven characteristics and objectives, a collection of engineering industry standards was also devised, as a necessary set of technical qualifications and metrics to which this system must comply. These standards align with both the user-driven metrics as well as the additional metrics necessary for a consumer product at large. These standards are defined and discussed later, in SubSection 4.8. First, the chart below, Table 4.1, states the user-driven characteristics, along with their corresponding goal metrics.

Table 4.1: Goals for system-level characteristics

System-level Characteristic	Basic Goal
Degrees of Freedom	5
Gripping Capability	2 DOF gripper
Temperature Resistance	Propane Flame: 1,995°C
Static Load	3.0 N
Dynamic Load, User Input	2.0 N
Reachable Workspace	Spherical, spanning a 200 mm radius
Storage Size	400 mm^3
Weight	45 N
Median Time Between Steps	<5 sec
Cost	<$250
Energy Consumption	<20 Wh/day
Efficacy	Hands-free, with >20% time reduction per task
Reliability	Successfully holds typical ring, bracelet, necklace chain

In the following subSections, the reasoning for each characteristic and metric defined in the above Table 4.1 is discussed and rationalized, as backed by user feedback.

4.1 Reachability and Orientation

Based on visual studies of user workflow, in order for the system to be able to provide the jeweler access to all orientations and areas necessary for soldering a piece, the device must be capable of providing 360-degrees of spherical reachability at the end effector, as well as two independent access points in gripping capability. This drives the need for a two-degree-of-freedom wrist, with both panning and tilting rotational joints, as well as two additional degrees of freedom in the grippers, for providing exposure to different areas of the piece, while still maintaining a fixed hold[1]. Further, in order to adapt to a variety of user preferences and workspaces, the system should have one translational degree of freedom that can position the object at different heights in the workspace. Thus in total, this drives the need for a system with five degrees of freedom.

4.2 Heat-Resistance

Research was performed on the temperature required to solder for a variety of techniques and metals - primarily silver and gold, including various alloys. It was found that the maximum melting point of any material for soldering was for 91.8% gold, 22 carat solder compositions, with a melting point ranging from 940-960°C.[4] Additionally, the heat expelled from the propane torch used to solder these materials is 1,995°C. Thus, it was ascertained that the system must be resistant to heat, at maximum temperature of 1,995°C. Consequently, a paramount material property concern is heat-resistance, and thus the material choice performed later on will be based primarily on this heat-resistance metric.

4.3 Load-Bearing Capabilities

The amount of static load the system must be able to support depends on the weight of the objects it will hold in place. To quantify a goal supportable load, the team honed in on the average weight of rings and chains made out of both silver and gold, eliminating edge cases with heavier stones and unconventional metals or materials. It was found that 3.0 N of static load-bearing capabilities is sufficient, as the maximum mass of any researched piece was found to be 30 grams, or 0.30 N weight, for a thick-mesh gold necklace, leaving a significant safety factor of ten.[5] Next, in order to determine the magnitude and nature of dynamic loading on the system resulting from user input, stakeholders were observed soldering and setting pieces in a variety of circumstances, including ring-resizing, chain reparation, stone setting, and metal plating.[1] In each situation, there was a very minimal amount of added user force, as the process primarily involved holding the blow torch over the object, with very little physical contact with the work object or third hand itself. The primary source of user force input involved placing the pieces of solder on the work, which was only a gentle touch and required nearly negligible force. However, that said, with the design of a consumer product in mind, the team chose not to neglect this forcing to account for improper usage or rare edge cases. Thus, based on the results of the user analysis, it was determined that the maximum dynamic load on an average piece, driven by user input and thereby necessary for the system to support is at most an additional 2.0 N. As stated above, this metric was devised with all usage in mind, and thus covers both typical cases and edge cases for the user.

Finally, the need for supporting the defined loads drives another need relating to effective material choice. In order for the system to be able uphold the metrics defined for the loading characteristics, the materials chosen for design must be strong and capable of bearing the defined loading. Moreover, the team aims for an optimal material choice, that is not only heat-resistant, as determined in SubSection 4.2, and strong, as determined by the above loads, but is also relatively light and cost-effective, attributes that are valued in any well-designed consumer product.

4.4 Reachable Workspace, Storage, and Weight

In interviews with stakeholders, seven individual interviewees deemed the necessary reachable workspace of the system to be minimal - as most of the brunt-work in jeweling happens in the small space directly in front of the jeweler, at the center of his or her workbench.[1] Consequently, the metrics defined for system geometry and sizing reflected this confined space with a reachability bounded by a 200 mm-radius sphere at the work object. In addition to driving workspace metrics, the user interviews helped shape the needs for both the storage and weight characteristics. Stakeholders indicated that most, if not all, of their work is performed in their studio at their workbench, and is rarely done on-the-go. Thus, there is limited need for compact storage or an extremely light-weight system, given that portability is not of concern. However, the user still must be able to lift the system to place where desired, as well as store the system when he or she is not using it. Taking all of these factors into consideration, a reasonable system weight was determined to be approximately 45 N. Also underpinned by stakeholder feedback, the basic metric for volume of the system when in storage was determined to be 400 mm^3. In order to achieve this goal of a compact, light-weight product, the team aimed to optimize the system by creating a base as compact as possible, without comprising system stability, as well as creating a miniaturized gripping system that would fit within a jeweler's tight workspace without compromising system reachability and dexterity at the end effector.

4.5 Energy and Cost

The metrics for energy consumption of the system are driven by research done on the energy consumption of a variety of small household appliances and consumer electronics on the market. This provides a baseline for an acceptable wattage value. Additionally, outlined in user needs, this system would be used for 50% of the jeweler's workday, or approximately 4 hours per day. From the range of consumption values researched, it was determined that a reasonable maximum consumption value would be on the order of 5 W, which is the same as the average energy consumption of a smart product, like the Amazon Echo.[6] With a 5 W rating and 4 hours of usage, this equates to an energy consumption metric of 20 W-h/day. Additionally, an average smart product on standby mode has a power usage of 3 W, which would equate to an energy consumption of 12 Wh/day. Thus, taking this value into account and assuming this system will not be active during its entire daily usage, the goal metric was defined to be less than the maximum usage defined above, or <20 W-h/day. In a region with the average residential energy cost, $0.12/kWh, this In addition to observing energy consumption ranges, the team determined the optimal price point of the system. By analyzing market trends from generic and affordable to state-of-the-art jewelry equipment, the product pricing range was deduced to be fairly wide, spanning from as low as $5 to upwards of $300.

Stakeholders expressed that they, along with most typical jewelers, will aim to buy a product towards the median of this range, as to achieve a higher quality while maintaining an economic conservatism.[2] Thus, the maximum goal cost metric was determined to be $250, placed at the higher end of the above range, due to the product's technical complexity, but not too high as to exceed the upper bound of this range. Should the quality remain consistent with the maximum-cost $250 version of the product, a lesser price point would be advantageous as it could provide a competitive edge. A more affordable price could come from bringing the object from small- to large-scale manufacturing, as well as optimizing material choice, as stated in SubSection 4.4.

4.6 Time Constraint

Given the goal of creating a system that alleviates the pain points of current adjustable clamping systems for jewelry soldering, a time constraint for orientation adjustments, system movements, and switching out pieces stood out as a necessary system characteristic. As a baseline comparison, the time it takes for the system to complete one step should take less time than it does to perform said step manually. After observing jewelers at work, the amount of time to complete a manual adjustment was ascertained to be approximately 5 seconds.[1] Thus, the time it should take for the system to complete the same step should be at most 5 seconds, giving way to this as the maximum value for average time between steps. However, in effort to design a product that is more efficient and a desired alternative to the current market offerings, the time to complete a step should not only match the manual alternative, but beat it, adding an additional factor of efficiency to the system. Given this desire for efficiency, the metric was defined to be <5 seconds for an average step time, with a goal of 4 seconds, in line with the efficacy characteristics defined in SubSection 4.7. However, it is important to note that these times are merely an average, as larger motions will inherently take longer to complete than small adjustments. Additionally, these times do not necessarily correspond one-to-one with the overall time it takes to complete the series of steps necessary when breaking a soldering workflow. Particularly, the efficacy of the system, relating to the overall increased time efficiency of the system, is explained further below.

4.7 Efficacy and Reliability

Two crucial system characteristics, efficacy and reliability, are defined directly from the initial problem statement and the primary user pain point: the ability to firmly and stably hold a work object in place while soldering and adjust its orientation to reach the entire surface area of the object without breaking the user workflow. This pain point is therefore twofold - holding the object firmly in place, and adjusting orientation without breaking workflow. The reliability of the system will come from its ability to hold the object in place, without dropping it, losing grip, or giving out under user input. For a basic metric, this reliability should hold “typical” jewelry pieces. Specifically, the system should be able to hold ring sizes ranging from 4 to 13, as this is the range of sizes from the 15th percentile female to the 95th percentile male.[7] Additionally, it should be able to grip bracelet and necklace chains ranging from 1 mm to 3 mm, and support the four typical chain styles of cable, rope, wheat, and box.[8]

The latter portion of the specified pain point, adjusting orientation without breaking workflow, drives the metrics for system efficacy. Determined from user observation, it takes approximately 10 seconds to put the soldering blow torch down, adjust the object, and reheat the necessary area before continuing to solder.[1] Having a hands-free system would enable the user to maintain the necessary heat while simultaneously adjusting the object, consequently saving the user at least 5 seconds, depending on the system response time, by eliminating manual adjustments. Thus, for a basic metric, it was determined that 20% reduction in time to perform each adjustment was a feasible goal. Hitting this metric for the efficacy characteristic will depend on user testing and observation with both control and experimental systems, in which the time will be recorded for completing a necessary set of tasks on the existing manual solution as well as on the newly devised hands-free system. The efforts placed on hitting and validating these metrics, as well as the others defined in this Section, are discussed in greater detail in Section 6, Validation and Testing.

4.8 Engineering Standards

In order to be a capable consumer product, the system must not only successfully hit the objectives defined above, but also adhere to the engineering standards defined in the below discussion. The upcoming subSection will define what each of these standards entails as well as how the system’s metrics will be shaped by their adherence to them.

This product is categorized as a personal use robotic device. In the past few years, the major governing organizations have created or are working on a set of standards for these types of robots, as most of the existing standards only apply to industrial use or medical use robots. The main standard applicable to the device is ISO 13482: ”Robots and robotic devices -- Safety requirements for personal care robots.”[9] According to this standard, personal care robots are “service robots that perform actions contributing directly towards improvement in the quality of life of humans, excluding medical applications.” The standard categorizes three types of products: mobile servant robots, physical assistant robots, and person carrier robots. Physical assistant robots are defined as a robot that physically assists a user to perform required tasks by providing supplementation or augmentation of personal capabilities. This product will augment the user's task of soldering and the manipulation of the object he or she is working on. The standard goes outlines a list of potential hazards of these robots and the necessary control requirements in order to ensure adequate safety of the user. Part of this discussion includes a list of other standards which are encompassed by this new, system-level standard. The standards applicable to this product are the following:

1. ISO 4871/11202: Acoustics (noise emission values of machinery and equipment)
2. ISO 12100: Safety of machinery: general principles for design
3. ISO 13849: Safety of machinery: safety-related parts of control principles
4. ISO 13850: Safety of machinery: emergency stop
5. ISO 14118: Safety of machinery: prevention of unexpected start-up
6. IEC 60204: safety of machinery: electrical equipment of machines
7. IEC 60529: degrees of protection provided by enclosures
8. IEC 61140: protection against electrical shock
9. IEC 62061: Safety of machinery: functional safety of safety-related electrical, electronic, and programmable electronic control systems

The hazards discussed in the standard relate mainly to risks associated with human-robot interaction. These hazards are due to charging the battery, energy storage and supply, the start up and restart of operations, electrostatic potential, the robot shape, electromagnetic interference, stress, posture, usage, robot motion, insufficient durability, incorrect autonomous decisions and actions, contact with moving components, and hazardous environmental conditions. The standard gives guidelines to address these issues, taking the form of control system requirements. This will affect both the mechanical and software aspects of the product. In the design of the product, it must first be ensured that there are safety measures in place to stop the robot in case of sporadic movement or unexpected start up. Dextera has hard coded restraints on it's movements, so even if a user continues to tell it to move, it will not move past the predetermined boundaries. Next, there must be a limit to the operational space of the robot. This will unlikely affect the device as it currently stands since it cannot move its base autonomously (lays flat and still on a desk surface), which limits the operational space to a predefined area. Speed control, stability control, and force control must also be considered in the design. Speed and stability control are done by the software, but force control was not able to be achieved over the course of the project. However, the maximum force imparted by the grippers is derived from the motors used. The maximum torque output by the motors will not harm the work object or if the users fingers were in the path of the grippers.

In addition to ISO 13842, which covers general system requirements for robotic systems, it is also important to consider risks associated with specific components of the device that may require adherence to standards. First, IEC 61508-3: Functional safety of electrical/electronic/programmable electronic (E/E/PE) safety-related systems, gives guidelines to the software embedded in the device.[10] The standard requires the E/E/PE system to have minimum safety integrity level requirements. For example, it sets a numerical target failure rate for E/E/PE systems which are linked to the safety integrity levels.

Next is the ingress protection associated with the device's casing. Since the casing will house sensitive electronic components and exposed wires, it is important to prevent these parts from interacting with water. Due to the environment this device, it is not expected for the device to come in contact with powerful, constant flows of water or risk submersion into a pool of water. For this reason, IP65 is the applicable rating for this device.[11]

Finally, as this device will be frequently exposed to high-temperature flames, the flammability of the materials used in the gripper and surrounding the electronics are very important. There are a series of standards that describe this property which have now been coordinated: UL-94, IEC 60707, 60695, ISO 9772 and ISO 9773 for plastic components. According to the UL-94 standard, the device should achieve a V-0 rating (highest rating), signifying even under harsh conditions the device will not ignite or drip flaming particles.[12] Another standard, ASTM E1725 describes appropriate test conditions for fire-resistive barrier systems for electrical system components.[13] Passing these tests ensure the proper functioning of the device in the presence of flame. In order to comply with these standards, materials (stainless steel) were selected for their ability and frequent use in high heat situations. The motors and electrical components re also all housed under stainless steel shields to ensure no direct flame contact and minimize and convective heat transfer.

5. Design and Realization

Based on the research done towards user needs and primary pain points within the problem space, a single design question was honed in on: “How might we hold a work object in place securely and stably for a jeweler to solder, and simultaneously adjust the orientation to access of the object and reach its entire surface area without breaking user workflow?” Working off the underlying premise of this specific design question, the team followed a methodical process of concept selection and down-selection for the system at large, in which parameters were defined to compare newly devised solutions with both existing solutions and each other. Once a specific system-level design was chosen, there followed a process of specific sub-system definition and selection, which were determined by additional sets of parameters. After outlining the process by which the team chose both the system and corresponding sub-systems topology, this Section will also delineate the processes necessary to realize and construct the final device. Finally, the Section will end with a complete overview of the final system prototype, as well as a detailing of the internal workings and mechanisms of each of the final sub-systems.

5.1 System-level Down Selection

In order to properly address the design question at hand - “How might we hold a work object in place securely and stably for a jeweler to solder, and simultaneously adjust the orientation to access of the object and reach its entire surface area without breaking user workflow?” - the down-selection process was first analyzed from a system-level standpoint, looking at generalized solutions and potential devices to address the design criteria. For each design concept, the same set of parameters was used as a guideline for comparison and evaluation, ensuring methodical consistency. These parameters are defined as follows:

A. <u>Fulfills User Needs</u>: Holds object firmly in place, and affords the ability to adjust the work object’s position

B. <u>Alleviates Pain Points</u>: Can seamlessly move between tasks without breaking user workflow

C. <u>Feasibility</u>: The design is possible to implement given the team skillset as well as senior design time constraints, requirements, budget

D. <u>Cost</u>: Affordable point-of-purchase and maintenance cost for an average jeweler

E. <u>Ergonomic</u>: Comfortable to use, eliminating user issues such as hand cramping and development of carpal tunnel

F. <u>Weight</u>: Reasonable weight as to allow user to pick it up and move it within studio or workspace

G. Energy Consumption: Does not take too much power to run as to prevent user from using other necessary powered tools

Based on these parameters, novel solutions to this problem were devised, evaluated, and compared with each other as well as with the current solutions within the problem space. Below are detailed sketches, coupled with an explanation and parameter-based evaluation for each solution design that was included in the comparison. First, the existing solutions were evaluated in the following Section. Next, the internally-devised solutions, broken up into the specified categories, are explored. Each solution is then compared in a comprehensive table, which quantifies the adherence of the solution to the given parameter on a scale of 0-10.

5.1.1 Existing Solutions

I. Existing Third Hand: The existing third hand for jewelry soldering consists of a long set of heat-resistant tweezers, connected to a base and tightened into place using a set screw. In order to adjust the piece within the tweezers, the jeweler must manually loosen the tweezers within the base, re-orient the piece, and tighten it back into place before continuing to work. Thus, while it properly fulfills the user need of holding the object firmly in place, it fails to alleviate the pain point of allowing the user to perform adjustments without breaking workflow. In terms of cost and weight, this solution is both lightweight and cheap, making it the leading solution in the current market. Additionally, it is a passive system, eliminating energy consumption considerations. However, it is not an ergonomic solution, given the pain points it creates for the user.

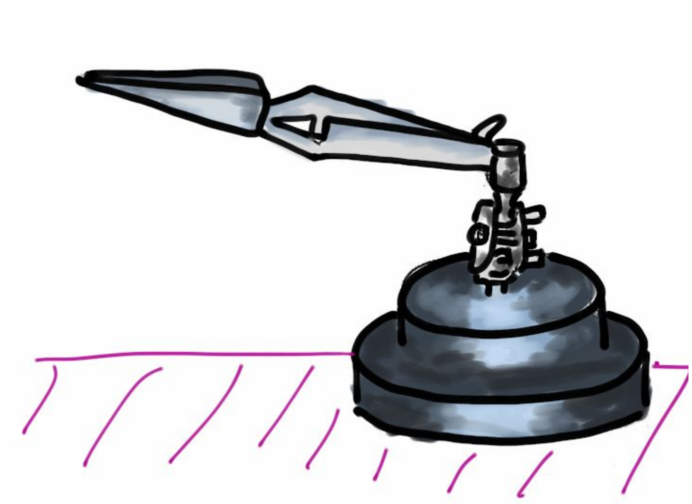


Figure 5.1: Drawing of existing third hand

II. Manual: Some jewelers opt to hold a long set of tweezers in their hand when soldering, allowing them to alter the orientation of the object while soldering in real time. However, the jeweler must still stop working when a different area of the piece must be exposed. Additionally, he or she must wear protective-wear such as heat-resistant gloves, as safety in the presence of the blowtorch is crucial. Thus, this solution only partially fulfills the user needs and does not alleviate the user pain point. Additionally, it is not an ergonomic solution, as it creates issues of hand-cramping, additional user work, and safety hazards for the user. However, it is notable that this solution inherently has no concerns of weight or energy consumption, and the cost of soldering tweezers is low.

Figure 5.2: Drawing of alternative manual solution

5.1.2 Devised Solutions

III. Locking Spherical Gimbal: This system holds the work object similarly to how a globe stand holds a spherical globe. In this solution, the work object only has the ability to rotate around a single axis but not move laterally or rotate about additional axes. The orientations available to users in this solution to orient their work objects are not sufficient to satisfy their needs. When working with rings containing colored stones, for example, jewellers need to position the stone in water to avoid heat damage from the blowtorch.[1] However, with one degree of rotational freedom, the user would be able to reach the entire surface area of the object but not sufficiently manipulate its orientation.

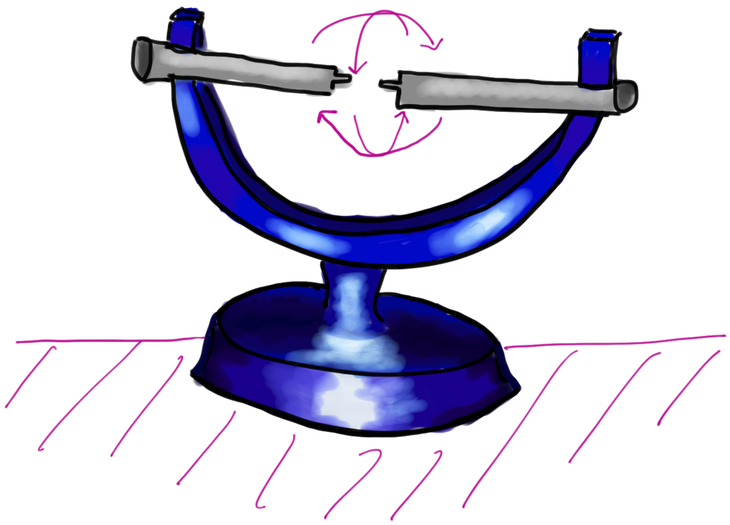


Figure 5.3: Drawing of Locking Spherical Gimbal

IV. Soft Robots: Soft robots are constructed of highly compliant and flexible materials embedded with actuators.[14] While this solution can grasp, hold, and manipulate objects with lowered risk of crushing or damaging the object , which is a challenge for rigid linkages, this solution would not provide the structural support necessary to satisfy the precise holding needs of the user.[15] Jewellery soldering is a heat-intensive process, as the jeweller utilizes a blowtorch to melt the solder.[1] Soft robots are typically made of materials such as silicon, paper, rubber, cloth. A necessary requirement of the system is heat or flame resistance. Thus, this solution needs to be constructed of a fireproof material such as mineral or glass wool.[16] However, implementing these materials are not feasible as jewellers' soldering processes are messy and would soil the material easily. Thus, the optimal solution must be constructed a material that is convenient to clean.

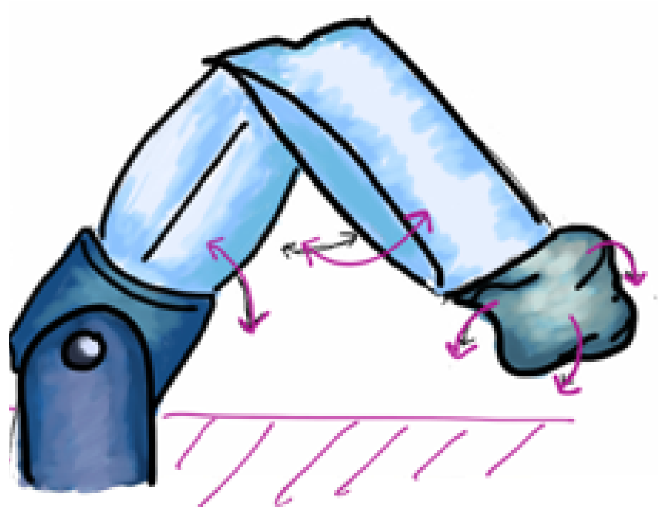


Figure 5.4: Drawing of soft robotic arm

V. 1-DOF Rotary Device: This system operates like a rotisserie device, in which the work object would have one degree of rotational freedom, similar to the locking spherical gimbal solution. However, in this solution, the user's access to the work object is limited given the orientation and type of gripping system. Thus, while this fulfills the need of holding the object in place, it fails to alleviate the pain point of accessing a variety of orientations and object surface areas. As for the weight, cost and energy consumption of this system, the need for only one point of actuation gives this solution a competitive edge for each of these characteristics in comparison with some of the other proposed, more complex systems. Additionally, it is feasible, as it is again only requires one degree of freedom, which can be provided by controlling a single motor. However, ultimately, alleviating pain points and providing an ergonomically-optimal solution are paramount to these other characteristics, and thus, this solution is not the best.

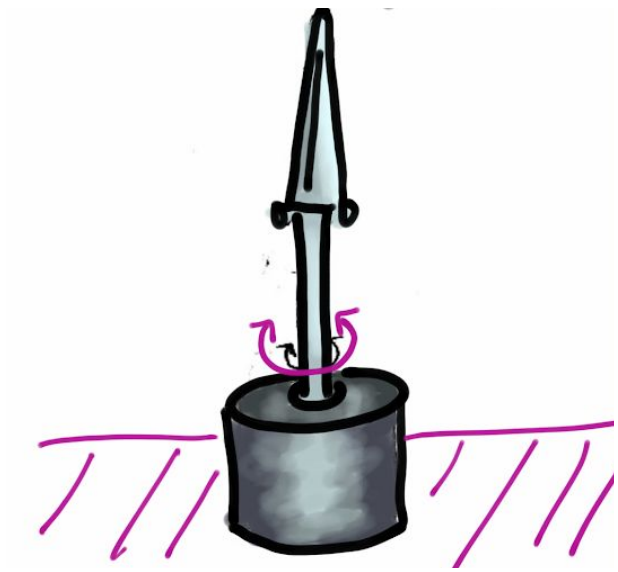


Figure 5.5: Drawing of 1-DOF Rotary Device

VI. <u>Dual Arm System</u>: This system has two arms that move in conjunction with each other. Both arms are actuated and have controllable grippers. In terms of fulfilling user need, this solution is successful, as it is theoretically able to firmly hold an object in place. Moreover, it helps to alleviate the pain point of accessing the entire surface area of the object, as it allows the user to grip with one or both grippers at a time, providing flexibility in object exposure. However, controlling both arms simultaneously poses challenges for effective human interactions, as well as technical challenges in implementation - discrediting its ergonomic quality as well as its feasibility. Thus, while it alleviates the pain point of exposing different areas of the object, it may impart an additional pain point of a steeper learning curve, impeding the efficacy of the system. The theoretical cost of the system is also greater for this solution, as it requires double the amount of motors, microcontrollers, and material as the other proposed actuated arm solutions. Similarly, the weight and energy consumption for this system would also be increased twofold. Thus, while it alleviates user pain points and fulfills user needs, it is not the best solution to the problem.

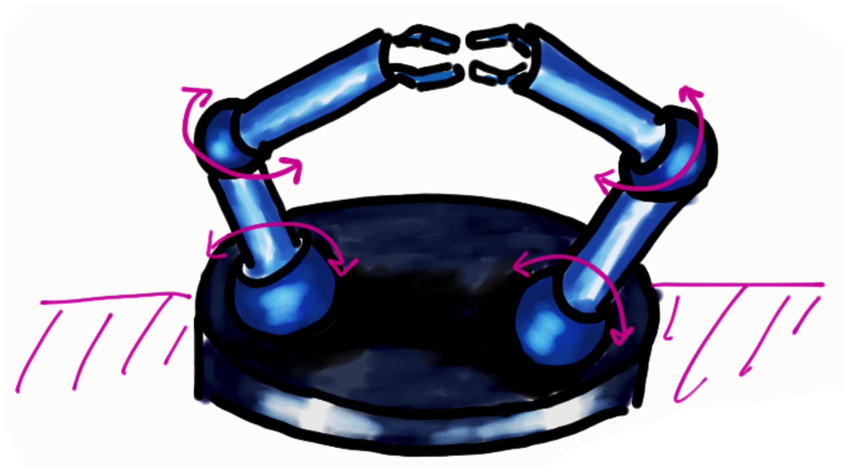


Figure 5.6: Drawing of Dual Arm System

VII. 6-DOF Revolute Robotic Arm: This system is modeled off of a typical robotic arm, similar to the LynxMotion robotic arms on the market.[17] However, it is more specially designed for this use case, as all wires and systems are enclosed, the arms are heat-resistant and load-bearing, and the overall system is downsized appropriately as to fit comfortably in the center of the jewelers workbench. This system fulfills the user needs of holding firmly in place, although, the additional torque placed on the system due to the geometry of the arm presents a tradeoff between cost, weight, energy consumption, and efficacy in fulfilling the need. If expensive, heavy-duty, high-power motors are used to compensate for this additional torque, then the user need of a firm and stable hold can be fulfilled. However, this comes at a price of increasing cost, weight, and energy consumption. Additionally, it does not alleviate the pain point of providing access to multiple surface areas, as it only has one gripper. Despite some of the other trade-offs, this solution is feasible within senior design and is ergonomically and anthropometrically sound.

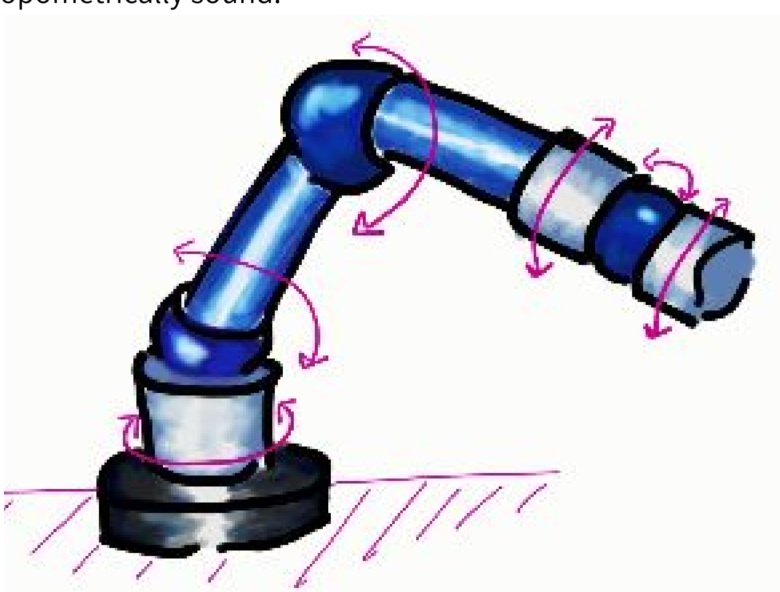


Figure 5.7: Drawing of 6-DOF Revolute Robotic Arm

VIII. Spring-Loaded Arm: The spring-loaded arm is a passive system that requires manual input to adjust, and uses a spring-loaded gripper to hold the object in place. It fulfills the user needs in the sense that it adequately holds the object in place for soldering. However, it fails to alleviate the pain point of adjusting without breaking workflow, as the only way to adjust this passive system is manually. However, it provides advantages in cost, weight, and energy consumption: with no motors, the system consequently costs less to build, becomes lighter, and does not need to be powered. Additionally, the solution is feasible to create given its purely mechanical nature. Ergonomically, it does not pose any additional user interaction challenges; however, given that it does not appropriately address the necessary pain points, it does not foster a seamless ergonomic experience.

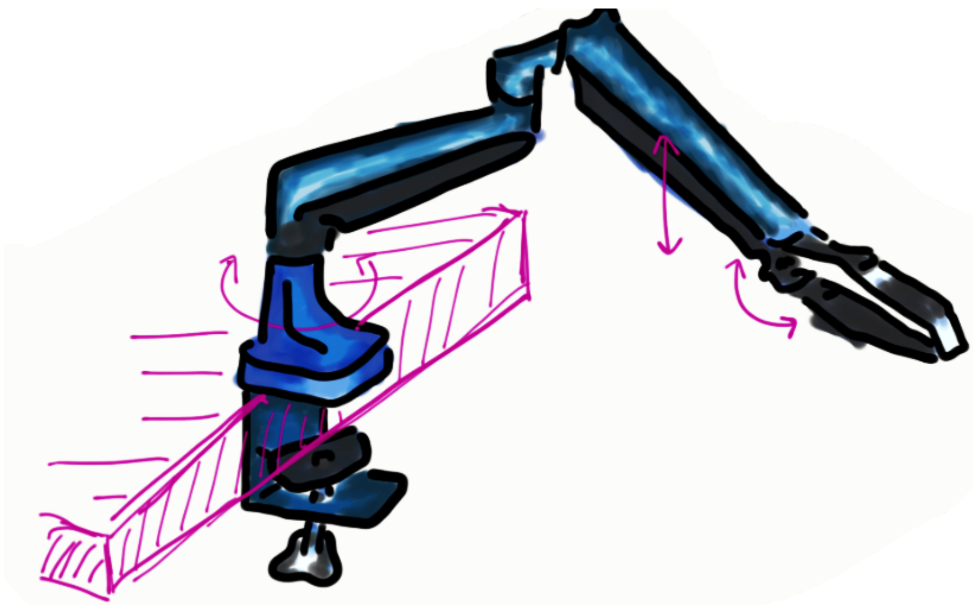


Figure 5.8: Drawing of Spring Loaded Arm

IX. 5-DOF Robotic Arm with Alternating Grippers: Finally, the last concept evaluated involves 5 independent joints, 3 linear and 2 rotational, in a spherical, 3D workspace. With two independent grippers at the end of the arm, this solution both fulfills the user need by providing a firm hold on the work object, as well as alleviates the specified pain point by allowing for different orientations and different surfaces of the object to be exposed without forcing the user to remove the object from the system. In terms of cost and energy consumption, this system would be a bit more costly and require more energy than some of the others, given its five different points of actuation. However, with efficiently-designed internal mechanisms, this consumption can be decreased, for example, when the system is not in motion. The weight of this system is on par with some of the other solutions devised, as they must all be made of the same heat-resistant materials. The physical feasibility of this system is also sound, provided the later subsystem design choices comply with the necessary system characteristics. Additionally, the capabilities of the team make this solution technically feasible as well.

Finally, in terms of ergonomic considerations, this solution adheres to the form defined by user preferences, is adaptable to multiple different types of users, and covers a wide range of anthropometric measures, making it an ergonomic solution.

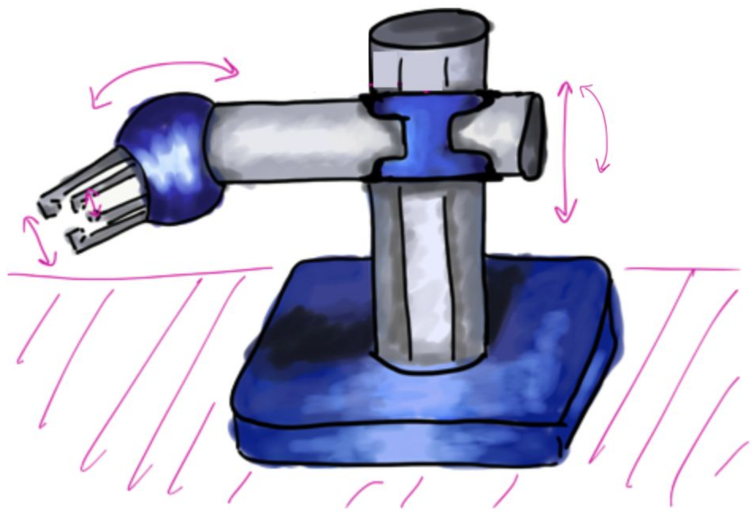


Figure 5.9: Drawing of 5-DOF Prismatic Arm

Based on these evaluations, the table below quantifies each solution as it adheres to each necessary parameter.

Table 5.1: System-Level Downselection Matrix

	Existing Solutions		Devised Solutions						
Parameter	I	II	III	IV	V	VI	VII	VIII	IX
Fulfills User Needs	7	4	4	5	5	10	8	6	10
Alleviates Pain Points	0	2	4	6	5	8	8	6	10
Feasibility	--	--	9	5	9	6	9	8	9
Cost	8	10	8	5	8	7	7	9	8
Ergonomic	5	0	4	9	6	8	8	7	10

Weight	10	10	8	10	8	7	7	10	7
Energy Consumption	10	10	8	7	8	6	7	10	8

As evidenced by the results outlined in the above table, system-level concept IX was chosen as the foundational design for Dextera. After performing the careful analysis and consideration for each design concept detailed previously, it became clear that this design best met the criteria necessary for an effective product. It is of note that this design concept served as merely a jumping-off point, for which many later iterations were designed, modeled and created.

After choosing this initial design concept, the system was further broken down into categorized subsystems, for which the team followed a similar down selection process. The subsystems of this design concept were broken down into categories based on the necessity for physical manufacturing and realization of the product, as well as software design for product functionality. Thus, the resulting subsystems were defined to be: controls, motors, and material choice. The down selection process for each of these subsystems is described in detail below.

5.2 Subsystem Down Selection

5.2.1 Controls

In addition to system design as a whole, it is crucial to the users' pain point that the solution incorporates the ability to control the device in a hands-free manner, as this is the best way to enable the user to adjust an object without disrupting workflow. Thus, below are the devised solutions for enabling hands-free control, which are then evaluated and compared by the same key parameters that are used in the system-level design comparison above.

I. Voice Activation: In this solution, the system responds to voice commands from the user. Voice recognition technology is used to process and analyze user speech and subsequently move the device accordingly. Sample commands are "move up three inches," "rotate in ten degrees", and "pan down thirty degrees." A challenge for this method of control is calibration: ensuring that the system is able to correctly identify directions as up, down, in, out, away from, or towards the user. This solution alleviates pain points by allowing the user to control the position and orientation of the work object without use of hands. In order to ensure that the user needs are fulfilled, this solution must respond quickly and reliably as well as be programmed to move Dextera's joints in desired increments.

II. Foot Pedal: In this solution, the system responds to signals sent through a foot pedal controller. It fulfills the user need as well as alleviates the pain point of enabling hands-free control; however, it may lead to several other pain points, such as limited commands, increased number of steps to achieve desired adjustment, and a steep learning curve for properly controlling the pedal. Along with this steep learning curve, the human-device interaction would be difficult and require additional attention, decreasing the ergonomic optimality of this solution. Moreover, the foot pedal adds both weight and cumbersomeness to the system, and the additional materials required increases the cost of the system. Finally, the energy consumption of the foot pedal mechanism is on par with the rest of the control techniques, as all of them require sensors of different types, which require power values on the same order of magnitude.

III. Computer Vision: In this solution, cameras are mounted to the end effector to sense where the arm is commanded to move. The user can motion Dextera to move in a certain direction or to a specific location with hand gestures. A challenge for this solution is determining the hand gestures the system is able to detect and understand; additionally, this solution causes a steep learning curve if users must execute extremely specific gestures in order accurately manipulate the arm. Another challenge with this system is ensuring high performance in all settings; the reliability of the solution depends on the lighting conditions the device is operating in as well as what the environment contains: a messy workspace or a crowded room may make gesture or movement detection difficult. Because this solution requires the user to use a hand to motion to the device, it would not alleviate the pain point of hands-free control unless the system were able to correctly interpret gestures while the user holds a tool in his or her hand.

Similarly to the design table defined for the system-level concept selection, below is a table that quantifies the possible controls solutions, as they adhere to the same necessary parameters:

Table 5.2: Controls Downselection Matrix

	Devised Solutions		
Parameter	I	II	III
Fulfills User Needs	10	10	8
Alleviates Pain Points	10	7	7

Feasibility	10	9	6
Cost	8	7	7
Ergonomic	10	6	10
Weight	9	7	9
Energy Consumption	8	9	7

As evidenced by the quantified results in the above table, in the final system form, Dextera uses voice activation as its mode of control, due to ability to effectively alleviate user pain points and its feasibility within the team's skillset and senior design constraints. In order to achieve successful and effective system responses, the team created an intricate software feedback loop that is described in detail in the upcoming Voice Controls and Software Design Section.

5.2.2 Motors

In order for the system to move smoothly, stably hold objects, have a desirable form factor, and be cost effective, the right motors must be employed. The team surveyed a range of motors and drive systems for each degree of freedom.

Firstly, the linear slide needed to have a linear speed of 1 inch per second to be able to achieve the desired time efficiency while maintaining system stability. Additionally, it needed to be able to lift the weight of the horizontal arm and steadily hold position while the user was working. In the down selection process, actuation methods such as continuous rotation servos, dc and ac gearmotors, and stepper motors were considered. Moreover, in order to translate rotational to linear motion for this translational joint, both a lead screw drive as well as a timing belt system were considered.

The ultimate gearmotor selected had a much smaller form factor and was less expensive than the equivalent continuous rotation servos considered. The brushed DC gearmotor was chosen due to its cost advantage over brushless motors, despite having the tradeoff of lower power for a given motor size and a lesser efficiency. Additionally, the lead screw drive was chosen as it was non-backdrivable, leading to a decreased power consumption and a lower required continuous stall torque of the drive motor, allowing for the usage of a smaller, less expensive motor. Further, the increased stiffness of the lead screw relative to the timing belt system, in addition to the non-backdrivability, increased the steadiness of Dextera's linear motion.

Next, the wrist rotate and pan degrees of freedom needed to have a rotational speed of 3 degrees per second to again ensure desired time efficiency and precision, as well as steadily hold position. More specifically, the pan degree of freedom drove additional motor size constraints, as it had to be located outside the main body of Dextera to effectively pan the wrist. Both servo and stepper motors were considered to achieve the desired motion for this degree of freedom, as they allow for easily implementable steady position control. Adhering to constraints, a servo motor was chosen over a stepper motor due to its size and cost efficiency.

In addition to the pan motor, the wrist was designed to include a 360-degree rotational joint. Consequently, this joint drove the need for a continuous-rotation motor: it needed to be able to rotate continuously in order to provide intuitive and high-speed control over position. For example, without a continuous-rotation motor, a user's command to rotate the arm 20 degrees clockwise could result in a 340-degree counterclockwise rotation when the arm is configured in certain orientations. Once again, with ease of control implementation in mind, both servo motors and stepper motors were considered to actuate the wrist rotation joint. Ultimately, a continuous-rotation servo motor was chosen for the same size and cost advantages stated above, as well as for consistency within device design and software execution.

Lastly, the gripper motors needed to be able to provide a controllable torque output in order to control the gripping force. Additionally, they needed to be able to fit within the wrist cover at the base of the grippers; therefore size, and torque specifications became the critical constraints for these motors. Gearmotors, servo motors, and stepper motors were all considered for gripping actuation. Ultimately, gearmotors were chosen over servo motors and stepper motors, as their output torque can be most easily controlled and their form factor is smallest. Within the gearmotor family, brushed DC motors are once again chosen due to their cost advantage over brushless motors.

Therefore, after utilizing the above down selection process, the final system form of Dextera included a brushed DC gearmotor attached to a lead screw drive for the vertical translational joint, a continuous rotation servo motor for the wrist rotation joint, a 180-degree servo motor for the wrist pan joint, and two additional brushed DC gearmotors for the actuated grippers.

5.2.3 Materials

In order for the system to be optimal for soldering, it must be able to withstand the heat of the open flame coming from the soldering blowtorch. Thus, the material chosen to construct the device was chosen largely based on its thermal properties, and was therefore considered more heavily during the validation and testing portion of the design process. Consequently, the analysis and subsequent down-selection procedure for this subsystem came later with validation of thermal properties for the system and are thus more aptly included in the later discussion regarding this thermal evaluation. The down-selection table and additional details involving the team's decisions regarding device materials are provided in Section 6.1, Thermal Analysis.

As will be shown in the later discussion, a combination of steel and aluminum was chosen as the optimal usage of materials, to create a device that is heat resistant, low-cost, and lightweight.

5.3 Prototyping and System Realization

As discussed previously, the final system form for Dextera was chosen to be a five degree-of-freedom robotic arm, with two rotational wrist joints, two independently-actuated grippers, and a linear prismatic joint, all of which were chosen to be controlled with the user's spoken command. The system was chosen to be designed with a combination of servo motors and gearmotors, and manufactured out of both steel and aluminum. To turn this design vision into a functional, consumer-ready prototype, the team operated under the design constraints defined previously, and worked through numerous models and iterations before arriving at the final prototype of Dextera.

First, the team created a bare-bones arm prototype, which included only essential motors and components to obtain a works-like, looks-like mechanistic model of the device, which is shown and described in the below Section 5.3.5. Simultaneously, the team worked towards creating a software prototype, that tested voice control and system response before connecting it to the physical device, as well as circuitry and motor control software. Sections 5.3.1 and 5.3.2 discuss these subsystems in detail. Finally, the team designed a finalized wrist assembly, grippers for manufacturing, and a main body enclosure to encapsulate the internal electronics of the device, which would complete all parts necessary for a final prototype. These three subsystems are discussed in Sections 5.3.3 through 5.3.5. The last step in the realization process came with assembling the physical prototype and integrating all subsystems - both software and hardware - to end with a final functioning prototype of Dextera, which is described in full in Section 5.3.6. As denoted above, each of the below Sections describes a subsystem of the device, its associated design process, and how it contributed to Dextera in its final form.

5.3.1 Voice Controls and Software Design

A Raspberry Pi B3+ controlled all of Dextera. This choice was made at the outset, given the Raspberry Pi's rapid-prototyping capabilities, GPIO-friendly design, form factor, computing power, and language-agnostic functionality. The software, as seen in Appendix B.2, is written exclusively in Python 3. This language was chosen for its ubiquity in current Machine Learning paradigms, Raspberry Pi support, and ample supply of pre-built libraries. The code is comprised of two main parts -- the voice and motor controls. Scripting the entire project in the same language allowed for easier integration.

The voice control was based in the Google Cloud Speech API. After thorough research and development trying both large corporation APIs (Microsoft Azure, IBM Watson, and Amazon Alexa) along with less well known ones, Google proved most effective at transcribing commands quickly and in a cost-efficient manner. The Microphone Stream class continually listened to the incoming audio file and broke up each word as they were being spoken. It then decided on various possible options, along with Google's probability distribution for the likelihood of each word. It also used the subsequent words to confirm or update the predicted word using context clues.

From the time the final word of the command was spoken, the computer would have the voice command in the form of a motor command in around 1 second. This lag time was decreased from 2-5 seconds through the removal of middle-man libraries such as Python's Speech Recognition, optimizing on a fixed dictionary of words, and the use of starting and ending hotwords. The computer knew when to start listening from hearing any of the words that start the commands and would not process outside conversation because it would not be in the proper command form. By using a concluding keyword instead of the traditional starting keyword (such as 'Hey Siri' or 'Alexa'), the computer was able to immediately know when to stop listening instead of having to wait for a multi-second pause. In the interest of following modern technology conventions, this ending hotword was the name of the software assistant inside Dextera -- chosen to be 'Graham.'

The motor control developed functionality over time. Abstracting motors and degree of freedom joints (DOFs) through the use of classes and inherited subclasses followed standard Python development practices. As a result, the code is more readable, easier to debug, and simple to adapt. Encoder and Sensor classes were built to abstract PID functionality, while a stand alone Gripper class was built independently due to its relative simplicity.

Furthermore, an Arm class instantiated joint objects, which consisted of various motors. This higher level class was chiefly responsible for turning the newly formed text command into a specific motor one, while also keeping track of the states of each joint as to ensure the motors would not go out of their safe bounds. Dextera supported both continuous commands ['Start moving up'] along with fixed commands ['Rotate towards me 10 degrees']. More discussion of the motor controls is below in Section 5.3.2.

5.3.2 Electronics and Motor Controls

The electronics and subsequent motor controls were a critical subsystem in the functionality of Dextera. In order to be able to actuate each joint in the arm as well as the grippers, the device needed to be able to process position sensor input, as well as output pulse width modulation (PWM) control. Additionally, each motor required different methods of control based on their motor type, as indicated by the existence of different motor objects in the code, defined in Section 5.3.1.

Analyzing the actuation control of the device at large, the servo motor are controlled directly by the Raspberry Pi, as external PWM control is not necessary when utilizing this motor type. For the gearmotors, onboard hardware PWM capability is used to provide repeatable position commands, as software based PWM was seen to generate inconsistent position commands, leading to twitching. Additionally, the DC gearmotors are driven by H-bridges in the circuit, packaged in quad half H-bridge drivers. The schematic of electrical connections for the circuitry that enabled these motor controls is provided in Appendix A.1. Looking at how each individual degree of freedom was actuated more closely, the controls scheme for each joint is defined as follows:

For the first joint, the position of the linear slide gearmotor is read by a quadrature encoder, which is mounted to the back of the motor and connected directly to the GPIO pins on the Raspberry Pi. Once the position taken from this sensor is read and processed in code, an appropriate motion command is sent back to the motor, using a mapping that converts the desired linear distance to number of motor rotations.

For the wrist rotation joint, the continuous-rotation servo is employed. Because the Raspberry Pi does not have analog input capability, an external analog-to-digital converter, which communicates to the Raspberry Pi over serial peripheral interface (SPI), is added to the circuit to read the analog position output of the servo. Once this is done, proportional integral derivative (PID) controls are utilized to send the servo to the appropriate updated position.

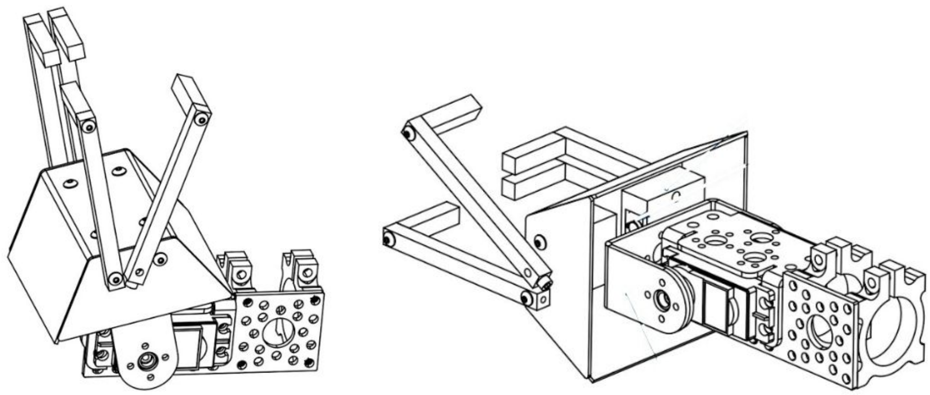
Next, the wrist pan joint, actuated with a 180-degree servo, is manipulated in software using embedded servo methods. Since the 180-degree servo has position sensing embedded in its packaging, there is no need for additional forms of control aside from sending the desired position directly to the servo's output pin.

Lastly, the gripper gearmotors are controlled directly using PWM, by setting specific power outputs to open and close the desired gripper, or stagnantly hold the work object. In order to ensure consistency, there is a universal opening time and closing time specified in the code, for which the gripper motor power is consistently on.

Along with the circuitry and accompanying software implementations necessary to control each of the motors, Dextera required circuitry to convert and distribute the power provided from the power source to both the motors and the Raspberry Pi. The motors and Raspberry Pi were powered separately, as powering the motors from the Raspberry Pi directly could overload the system and risk disrupting the digital logic. In practice, Dextera is designed to be run off of an off-the-shelf 120V AC to 6V converter, with a standard barrel jack connector. Thus, the circuit the team constructed converted this 6V input into a usable 5V power supply, off of which both the motors and the Raspberry Pi are then powered. The schematic of this circuit is provided in Appendix A.1.

5.3.3 Wrist Design and Manufacturing

The wrist subsystem allows Dextera to achieve the range of orientations that users require. When designing the wrist, important considerations included minimizing weight, compactly organizing the motors, creating a method for attaching the wrist motors to the gripper motors, and protecting the motors from the flame of the jeweler's blowtorch. To solve these issues, a heat shield was created that also doubled as a motor mount. This wraps around the wrist motors such that direct flames will not damage them. Additionally, the gripper motor encasings are attached to the inside of the shield, and the gripper motors are then mounted to these. The gripper motor shafts extend out from either side of the shield, and the two rotating grippers are attached to each respective shaft. The stationary gripper is attached directly to the side of the shield. The heat shield was manufactured in-house, bent from a sheet of stainless steel. Additionally, the wrist motors are attached to one another using standard servo brackets and mounts that were purchased. The full wrist assembly can be viewed in the schematics below.

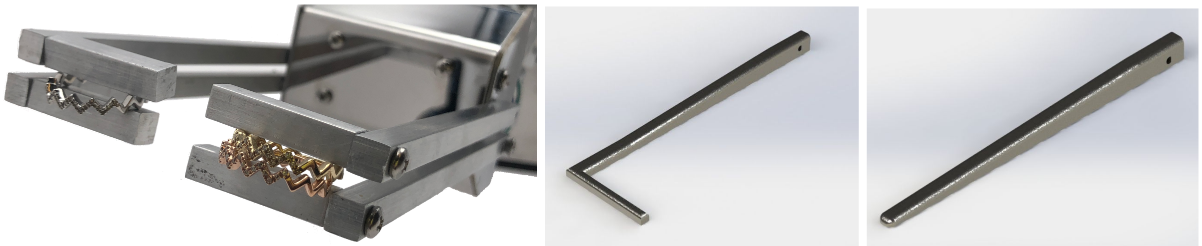


Figures 5.10-5.11: Wrist subassembly and heat shield

Figures 5.10 and 5.11 show how the heat shield interfaces with the wrist motors and gripper motors as well as the tilting capability of the wrist.

5.3.4 Gripper Design and Manufacturing

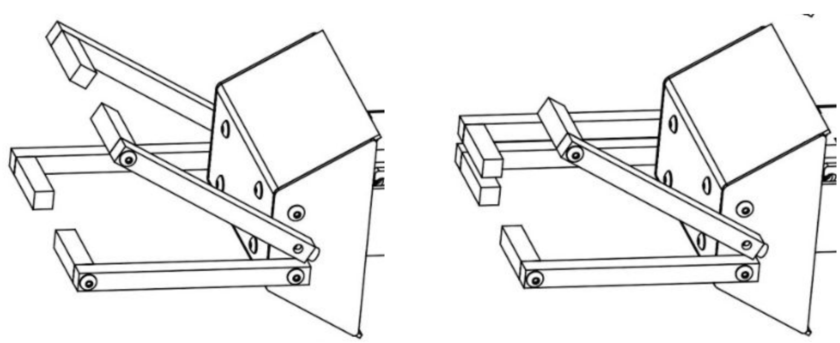
The gripper subsystem is one of the most crucial aspects of the device because it is the interaction point between Dextera and the user's work object. The main function of the grippers is to hold the object securely and safely while providing jewelers full accessibility to the object. However, the gripper design was also optimized to ensure that Dextera's temperature sensitive components would remain safe from thee open flame of a jeweler's blowtorch. The current third hand solution utilizes grippers that are thin and tweezer-like in shape. This design is optimal because the thinness minimizes material costs and a small contact point with the work object ensures maximum exposure of the object to the user. The main drawback to the gripper design of current solutions is that it requires users to unclamp the "tweezers" and remove, reorient, and then reinsert the work object in order to access all faces of the object. Thus, Dextera utilizes a dual gripper system to overcome this issue. By having two sets of grippers, each independently controlled, Dextera can accommodate a wide range of work objects, successfully holding and manipulating any type of jewelry as well as objects for unrelated use cases. Finally, Dextera's grippers are modular; users can easily swap different gripper styles to ensure the desired distance between contact points on the work object, allowing Dextera to cater to a wider variety of work objects.



Figures 5.12-5.14: Physical prototype grippers holding multiple objects, renders of different gripper designs

Figure 5.12 shows the dual-gripper functionality - a user can hold multiple objects at once. Users could also, instead, hold a single ring from two contact points, and thus gain access to all of the ring's surface area by alternating which gripper holds and supports it. Figures 5.13 and 5.14 are renders of two different gripper designs. Additionally, a jeweler can solder a long chain by extending a Section of it in between the two sets of a grippers. A bracelet can similarly be manipulated.

Initial 3D-printed gripper prototypes featured a tapered geometry and a minimum thickness of 0.1 inches. However, this design was not feasible to manufacture in the final material in-house using manual mill machining, so the final prototype features grippers of a uniform, 0.25" cross Section. The increase in cross Sectional height has no impact on user accessibility to the work object; however, a thinner, tapered end is more aesthetically pleasing. Thus, if Dextera were to be produced at a larger scale, the tapered profile could be reintroduced using a variety of casting processes, which would additionally lower the cost per part. On the final prototype, the mill-machined grippers achieved modularity through an attachment at the end of each gripper that extended inwards, towards the other pair of grippers. This attachment allowed users to hold objects of different sizes and to bring contact points closer together.



Figures 5.15-5.16: Dual gripper functionality

Figures 5.15 and 5.16 above demonstrate the dual gripper functionality: each gripper set operates independently of the other, so the user can open one or both at a time.

With the grippers designed, the next crucial step in realizing the final prototype was creating a main body enclosure to cover and protect the internal electronics described previously. Thus, in the upcoming Section, the design for the device enclosure and its accompanying manufacturing process is delineated.

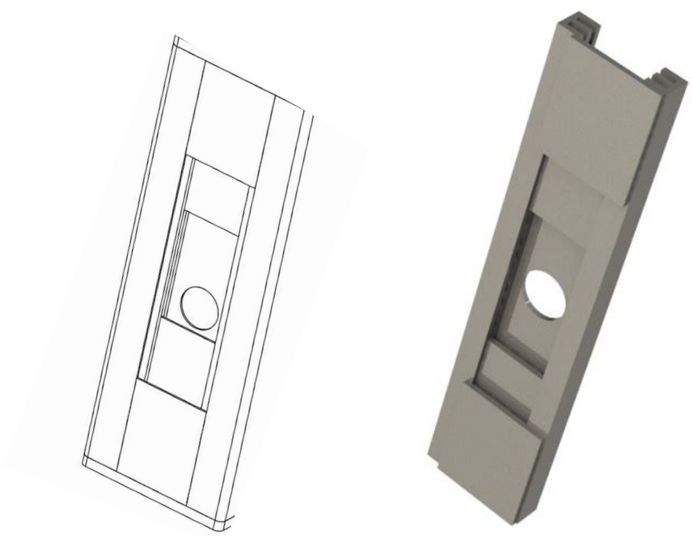
5.3.5 Enclosure Design and Manufacturing

Some of the design constraints the team considered for a device enclosure included heat resistance, aesthetic appeal, ease of manufacturing, and ease of access to internal components for debugging. Keeping these constraints at top of mind, the enclosure was made from a single piece of stainless steel sheet. Using this material, the manufacturing process entailed cutting the stock piece of sheet metal into a rectangle, then adding four sheet metal bends, creating a rounded-edge, rectangular prism shaped enclosure. With this simple process, the ease-of-manufacturing constraint was adhered to: the part was feasible to prototype and would be inexpensive to manufacture at scale. Moreover, the resulting part was simple, clean, and neat in appearance, with a polished exterior and minimalistic design, achieving the desired aesthetic for the product. Not only did the polished, stainless steel sheet metal used attribute to device aesthetic, it also ensured that the enclosure was heat resistant. Finally, incorporating acrylic magnetic caps for the top and bottom plates of the enclosure, discussed further in Section 5.3.5, adhere to the need for ease of internal access for debugging. The images below help depict this manufactured enclosure on the final device prototype.

Figures 5.17-5.18: Side View and Close-up View of Body Enclosure

One critical aspect of designing the enclosure was ensuring it did not hinder the vertical motion of the arm, while still completely covering and protecting the internal electronics. Thus, a number of covering mechanisms were analyzed; most notably, the team looked into using a bellow covering or designing a metal sliding mechanism. Ultimately, the metal sliding covering mechanism was chosen over bellows because it kept more in line with the aesthetics of the rest of the device, ensured temperature resistance at the base of the arm, and was less expensive to manufacture. While a bellow design provides a stronger degree of ingress protection, the team determined the sliding mechanism would still allow for protection from water and external object ingress to the level deemed necessary in a jeweler’s workshop. A larger discussion on this device protection as it relates to engineering standards is presented later in Section 7.2.

In order to make this sliding mechanism, two metal rails, each with a front and back track on them, were inserted on the inside of the sheet metal enclosure, at either side of the opening where the horizontal arm would be placed. Then, a sheet metal piece with a circular opening, where the arm is inserted into the enclosure, slides onto the back tracks. Finally, two additional sheet metal pieces slide onto the front tracks and connect to the top and bottom of the main piece, completing the full range of coverage for the vertical sliding motion. This sliding mechanism can be better visualized in the rendering and engineering drawing shown below.

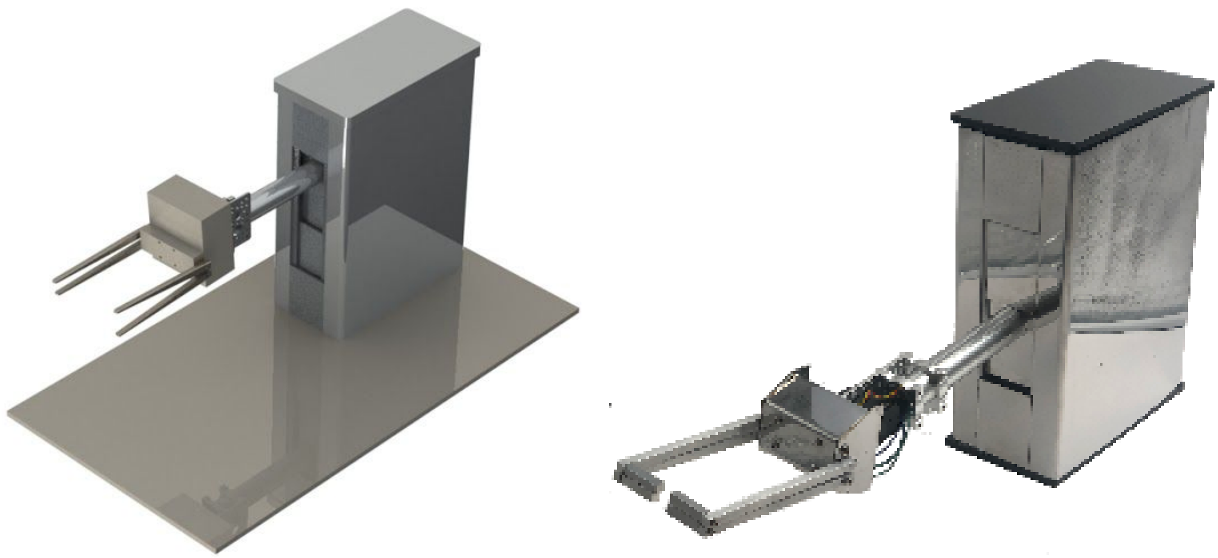


Figures 5.19-5.20: Engineering Drawing and Rendering of Sliding Mechanism

After designing and manufacturing the enclosure, the final prototype assembly could be completed and the device tested. The accompanying final form of the product, its assembly, and its operation are described in the Section below.

5.3.6 Final System Form, Device Assembly and Operation

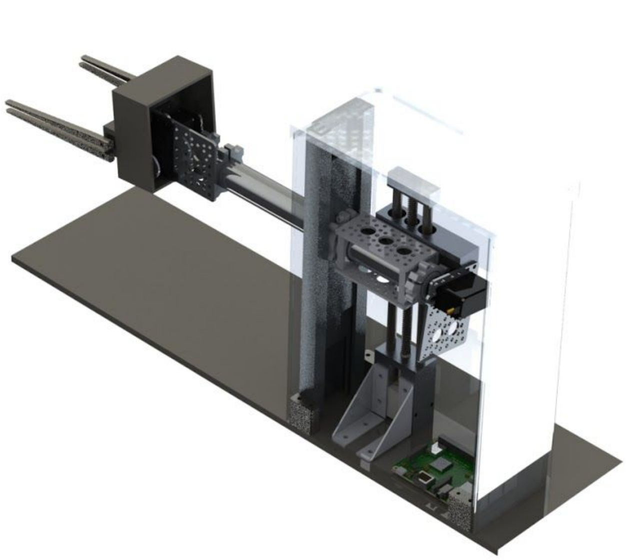
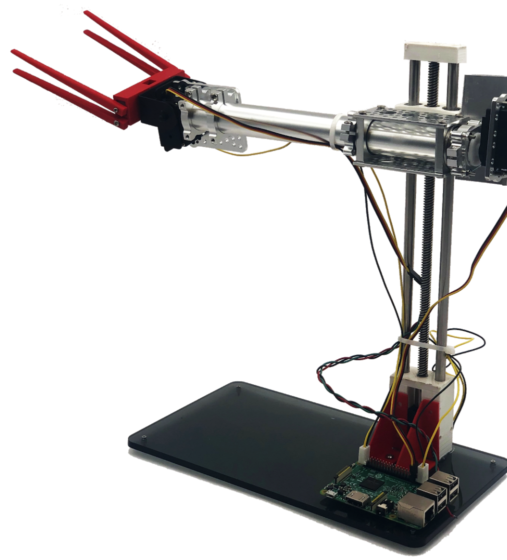
In Dextera's complete final form, each of the subsystems detailed above was integrated into a polished, encapsulated product. Both the base and the lid for the device were made out of laser-cut acrylic and magnetically snap to the main enclosure for ease of attachment and removal. The final iteration of the device did not include a base that extended underneath the arm, rather it included only a base on the main enclosure, as this was cleaner and more compact for the users' tight workspaces. The sheet-metal casing, described in detail in Section 5.3.4, houses and protects the vertical actuation mechanism, the arm's continuous rotation servo, and the Raspberry Pi computer, as well as holds the arm rigidly in place. Additionally, it incorporates a steel vertical sliding mechanism, as to not inhibit Dextera's range of vertical motion with the encasing. This enclosure design was motivated by the goal to protect the device, the user, and the object the user is working on from both heat and debris. At the end of the arm sits the wrist pan servo, as well as the two gripper gearmotors, mounted behind a steel heat shield, which serves as another mode of protection against the open flame of the blowtorch. Finally, the dual grippers extend past the heat shield to the end effector, where the work objects are held. In order to realize this final design, a complete model assembly was created in SolidWorks CAD. Including both parts aimed to be manufactured in the Precision Machining Lab, as well as parts ordered from external sources, this assembly heavily informed the final, as well as the intermediary forms of the Dextera prototype. To visualize the adherence of the physically constructed prototypes to the CAD model, the side-by-side renderings and prototype images are offered for comparison below.



Figures 5.21-5.22: SolidWorks Rendering and Final Prototype of Dextera

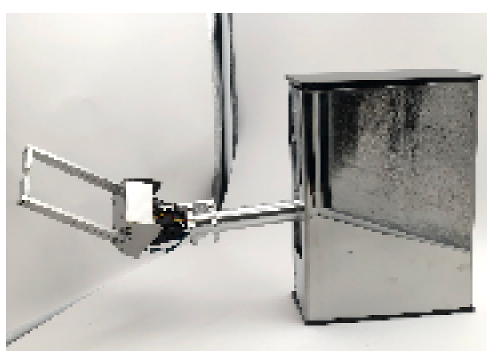
The above figures 5.21 and 5.22 display a final side-by-side view of the complete SolidWorks assembly rendering, including the main body enclosure, with the final Design Day prototype of Dextera. The minor differences between the two figures is evident in the lack of an extended base on the final prototype, and the variation between gripper designs. The sheet metal base was eliminated, as it simply added unnecessary bulk and weight to the device once physically realized. Additionally, the grippers included an extended L arm to display the modularity of the gripper design. Moreover, the gripper design was modified for ease of manufacturing, eliminating a tapered profile - in a consumer-ready version of this product with industrial-level manufacturing techniques, the tapered profile would be reintroduced, as explained previously in Section 5.3.3.

For an earlier look into the design process and how the team adhered to the SolidWorks design throughout, the below figures exhibit an additional side-by-side view, which portrays the internal mechanisms created in the CAD assembly, next to the initial, bare-bones prototype of Dextera, which served as a proof-of-concept of preliminary form for the product, as well as said internal mechanisms.

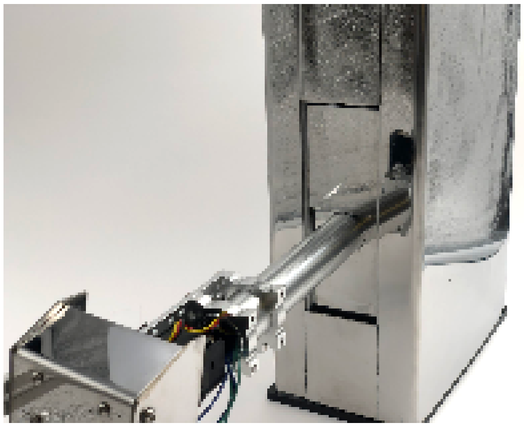



Figures 5.23-5.24: Transparent SolidWorks Rendering and Initial Prototype of Dextera

As shown, the prototypes made throughout the entirety of Senior Design work towards a final near-match with the intended product driven by virtual design. To show this in greater detail, below is an array of images of the final product, taken from a variety of viewpoints in order to thoroughly display the assembly and ultimate design. Further, engineering drawings of the entire system and all necessary subsystems, as well as a bill of materials that was utilized in order to achieve this final form, are provided in Appendix A.2 and A.3 respectively, to display the engineering behind and manufacturing of the prototype in greater technical detail.


Figures 5.25-5.27: Front, Side, and Isometric View of Dextera Final Prototype




Figures 5.28-5.29: Zoomed View of Wrist and Sliding Mechanism, and Grippers

In assembling the device, first, the vertical translation mechanism, which includes the gearmotor mounted to the lead screw drive, was adhered to the acrylic base via two 3D-printed brackets. The continuous-rotation servo for the wrist rotation joint, mounted inside a servo plate, was then attached to the top mount of the lead screw drive with a metal block adhered to the servo plate. This entire apparatus is placed inside the enclosure and the base magnetically snapped to the bottom of said enclosure. The arm is then slid through the circular opening in the front of the enclosure and secured to the continuous-rotation servo with an aluminum channel, two pillow blocks, and a clamping hub. At the end of the arm, the wrist pan servo motor is mounted to a servo plate, which attaches to the arm via two other pillow blocks. A mounting bracket extends from the servo to attach the two gripper motors, which are screwed into the heat shield. Finally, the non-actuated grippers are screwed into the heat shield, and the actuated ones are connected to their respective gearmotor shafts via set screws. Once all motors are in place, the wires are all threaded through the hollow tubing of the arm and plugged into their necessary pins in the perfboard that connects to the GPIO pins of the Raspberry Pi. The Pi and its connected perf board and the power converter circuit perfboard are rested inside the enclosure and, finally, the magnetic lid is snapped to the top of the enclosure. With necessary power cords and a motor power On/Off switch are threaded through a small notch in the bottom of the base, Dextera is then ready for usage.

As for the functionality and usage of the device, it is shown in the later validation discussion that the operation of the device is easily learned and understood, even from a layman's perspective. Dextera is plugged into a 6 V power source to activate both the Raspberry Pi and motor power. Dextera includes a voltage converter circuit that converts the 6 V power down to 5 V, as explained in Section 5.3.2, for proper digital 5V-logic. With a bluetooth headset connected to the device, the user states “Power up, Dextera,” into the headset and the device moves to the universal starting position. Dextera then continuously listens for its next command - the user simply can continue to speak his or her commands into the wearable microphone, and Dextera reacts with the appropriate motion in the desired time frame of just two seconds. As explained in Section 5.3.1 above, the user must state ‘Dextera’ at the end of his or her command to indicate to the device that the command is complete. Then, when the user is done working, he or she may state “Power down, Dextera,” for the device to move to its universal resting position, before turning off the power supply.

Internally, the subsystems utilized above - namely the voice controls and software, as well as the electronics and motor controls - are all integrated such that this command-response interaction is possible. In terms of gripping operation, the device is able to effectively hold a variety of rings, bangles, chains, and hoop earrings, with continuous actuation of the gripper motors to securely hold the objects in place. In the accompanied media package for this report, this operation of Dextera is exhibited. Additionally, below is a still image that exhibits the form and feel for the above described device operation.

Figure 5.30: Dextera in operation, being given a command

After defining and then realizing Dextera's final system form and operation, as it is shown in the mid-operation image shown above, as well as the multiple other renderings and images provided throughout this Section, the device was tested for a variety of important metrics, including efficiency, stability, thermal and structural integrity, usability, and reliability. The following Section describes in detail the procedures carried out to test the system and validate the many facets of its functionality.

6. Validation and Testing

Throughout the design process, the team continuously tested and validated each individual subsystem. Additionally, once the team had made significant progress, efforts focused on integration and full system validation. This Section will cover the testing and validation of specific areas, such as thermal analysis of the grippers and efficiency of the software. Additionally, subSections such as usability testing apply to the product as a whole.

6.1 Thermal Analysis

Thermal simulations were performed to confirm the gripper's ability to withstand constant, direct flame contact for nearly 20 minutes before becoming pliable. Additionally, through interviews with professional jewelers, it was discovered that the longest time a jeweler has reported continuously soldering for is ~3 minutes, demonstrating the large margin of safety built into the product. In order to ensure this level of reliability, each relevant factor was considered—material selection, gripper geometry, flame type, and jeweler behavior.

The first important consideration when designing to withstand high temperatures is the material choice. A range of popular metals used in high temperature applications were selected and evaluated to compare their characteristics against Dextera's needs, specifically thermal conductivity, melting point, and price. The system needs a material that minimizes thermal conductivity and price, but maximizes melting point. The table below shows these materials with their respective values in each category.[22, 23, 24, 25]

Table 6.1: Material Characteristics

Material	Thermal Conductivity (K, W/mK)	Melting Point (C)	Price* ($/lb)
Stainless Steel	15	1500	1.04
Aluminum Bronze	76	1038	1.67
Tungsten	163	3400	13.75
Titanium	20	1670	25.6
Nickel Alloy	61	1453	14.91

Stainless Steel is the best choice, and will be the material used in Dextera's grippers and body of the device. It has the lowest thermal conductivity and price of all the materials, and the melting point is close in range with other materials. There is no close second. Aluminum Bronze has too low of a melting point, Tungsten has too high of a thermal conductivity, Titanium is too expensive, and Nickel alloy is expensive and also has a relatively high thermal conductivity. Additionally, stainless steel is easy to access, easy to manufacture, and offers other benefits like corrosion resistance. This can be especially important in jewelry applications, as pieces are often coated in an acidic 'pickling fluid' when setting gemstones.

Once stainless steel was selected, simulations were run to ensure the grippers could withstand prolonged exposure to the soldering flame. The 3D geometry of the grippers in a clamped position was imported into both Comsol and Matlab and tested over time intervals ranging as long as 60 minutes with direct flame contact. The flame was modeled as constant heat flux at the left faces of the gripper. The simulation was first run for propane torches, which are commonly used by jewelers, and are estimated to have a heat flux of 100,000 Watts per meter squared. Butane torches were also looked at, which are used by some jewelers since they reach higher temperatures needed for certain metals, and have an estimated heat flux of 230,000 Watts per meter squared. The image shown here depicts the temperature gradient in the grippers after constant exposure for 10 minutes.

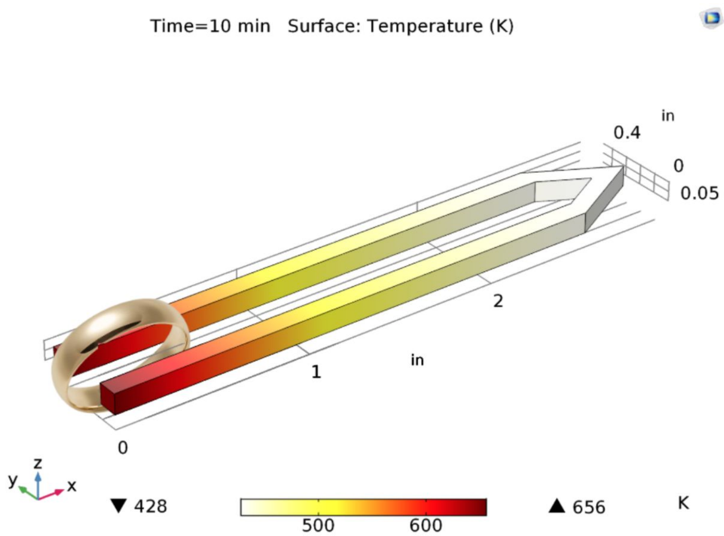


Figure 6.1: Temperature gradient in flame exposed grippers

In order to validate that the grippers will perform reliably, the maximum temperature in the grippers was tracked over time. This allows us to find the point in time at which the gripper will start to exhibit changes and ultimately melt under the flame. From this graph, it can be shown that the temperature at which it starts to become pliable, about 500 degrees celsius (or almost 800 Kelvin) doesn't happen until around the 20 minute mark. It is very unlikely there will be a case that is this long. From the user research, it is clear that soldering is only done for a few minutes at a time. The users are also trained in how to use these tools and were very clear with the team that you always direct the flame onto the jewelry, not the third hand. However, this design allows for improper use with a safe margin.

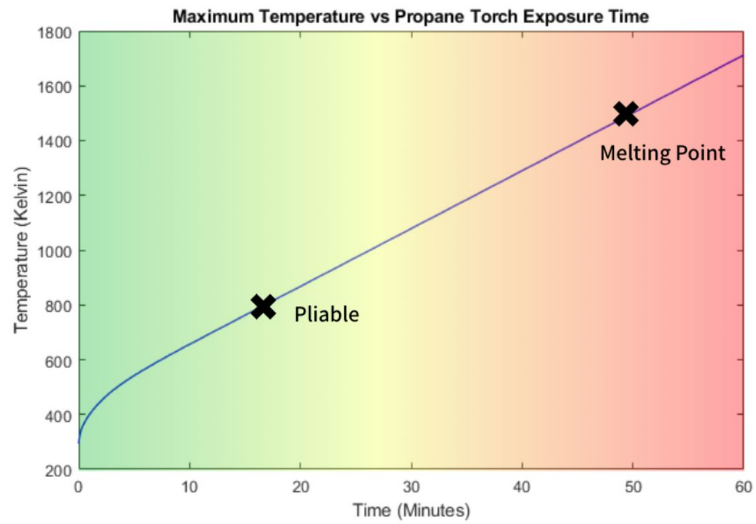


Figure 6.2: Gripper internal temperature when exposed to propane torch

In a similar analysis to the propane torch, results show that the grippers are able to withstand around 7 minutes of direct contact with a butane torch. This is still past the expected contact time, however it comes closer. It will be important to consider additional methods to extend this time as well as pin down exactly how common it is for jewelers to use butane torches and better estimate their use time as well.

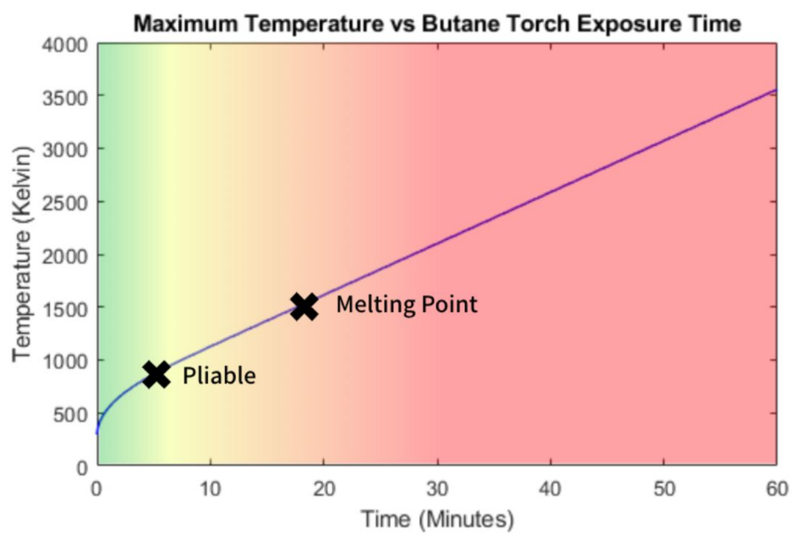
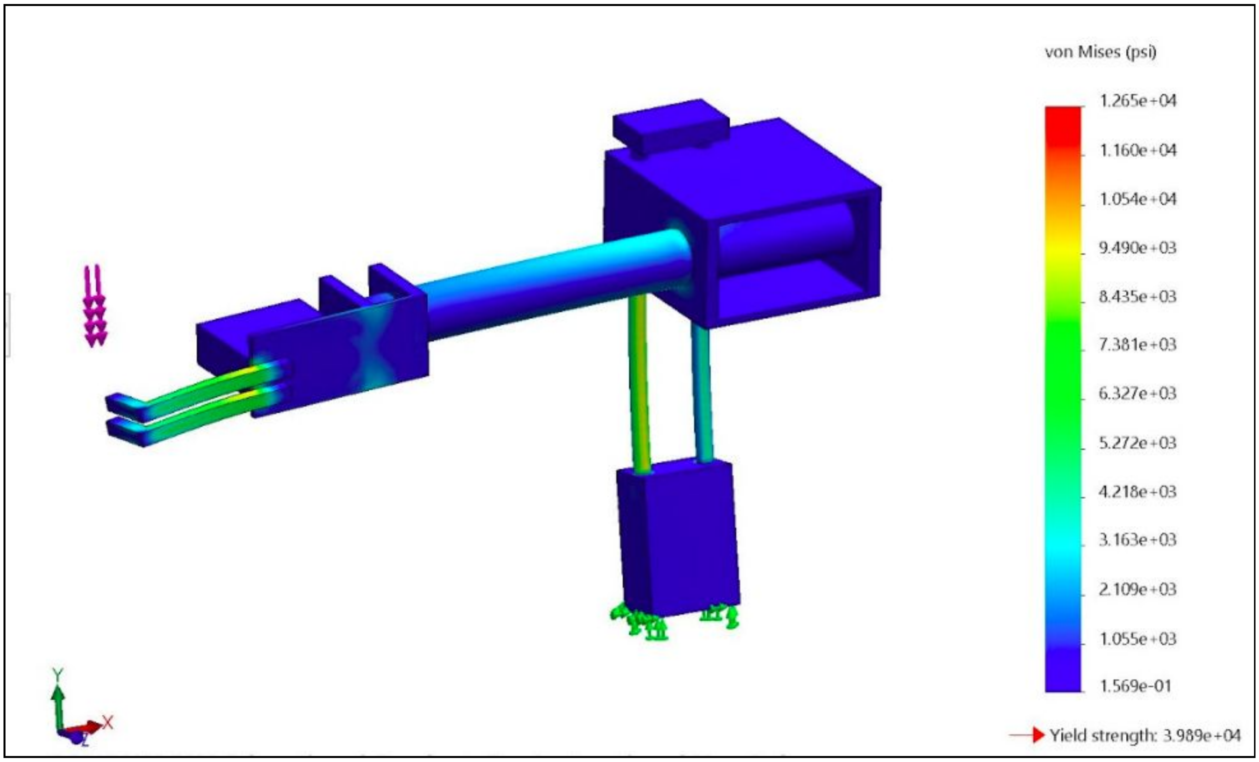
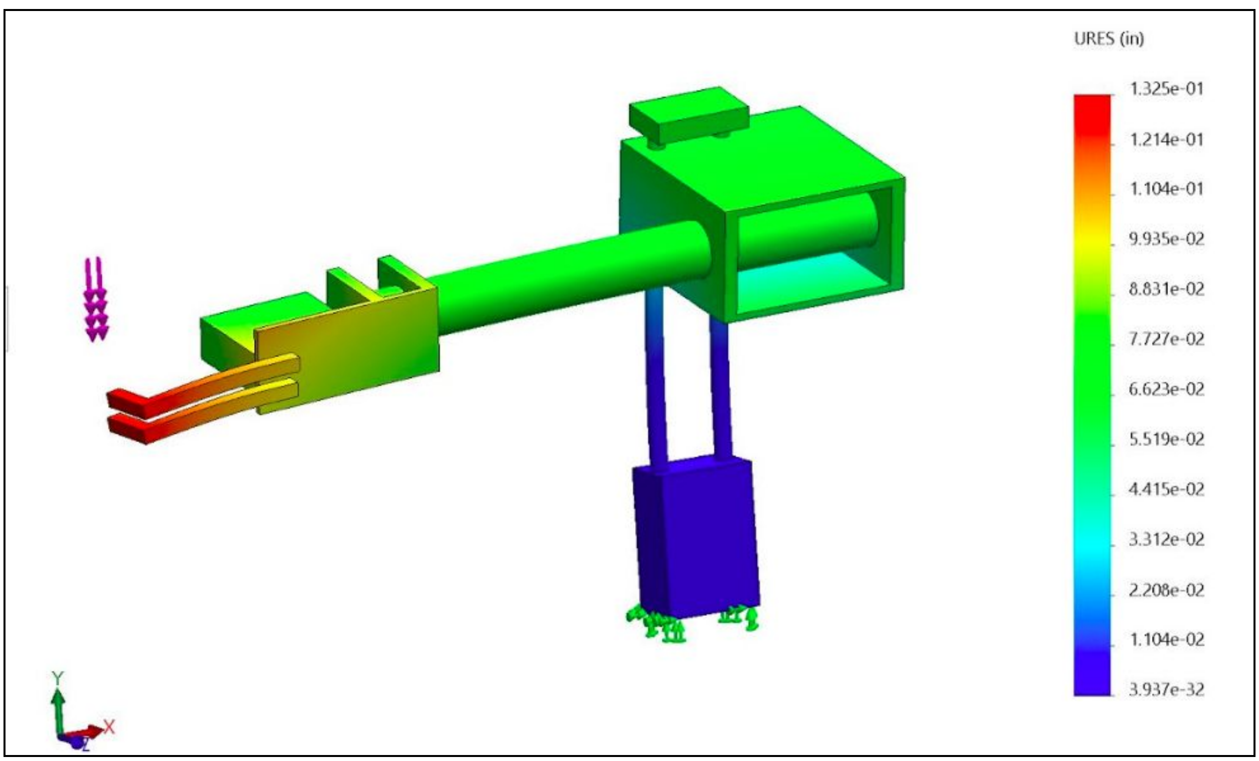


Figure 6.3: Gripper internal temperature when exposed to butane torch

6.2 Finite Element Analysis of Structure

In order to design the system to withstand the necessary loads and be stiff enough to allow jewelers to work on their objects from a stable platform, we used finite element analysis to analyze the structure of our system. We also performed calculations to size the bearings, fasteners, motors, and other components. Further, we calculated joint torques and compared them to the maximum torques of the servos. Lastly, we hung weights off of the end of the structure simply to verify our calculated results before moving on to assembling the final system.





Figures 6.3-6.4: Finite Element Analysis with Force Applied at Grippers

6.3 Software Efficiency

The final version of the software performed at a minimum of up to par and passed standards in key areas. The important metrics focused on were accuracy breakdown and end-to-end processing speed. More specifically, accuracy breakdown consisted of what percentage of commands were properly executed, improperly executed, or missed entirely. In a test of 40 input commands, 32 were properly executed, 2 were improperly executed, and 6 were missed entirely and needed to be repeated. Additionally, the end-to-end processing speed for a correct command was, on average, 3.5 seconds. This consists of roughly equal time for converting the speech to text (less than 2 seconds) and moving the motors (1-2 seconds).

From this, a calculation of the average time of each correct command can be calculated. The time to perform a correct or incorrect operation is stated to be 3.5 seconds, while the missed commands took 2 seconds before another command could be given. Thus, for the 40 commands, the total time can be determined. Dividing this value by the number of properly completed commands yields the average time per correct command.

$$\begin{aligned} T &= (32 * (3.5)\ +\ 2 * (3.5)\ +\ 6 * (2.0))\ /\ 32 \\ &= 131\ /\ 32 \\ &= 4.09 \text{ seconds} \end{aligned}$$

This is significantly faster on average than the 5 seconds it took the jeweler to perform the operation manually, not even factoring in the safety benefits of continually holding the tools as opposed to putting them down constantly.

6.4 Usability Testing

Concept and usability testing were performed throughout the entire design process. The team met with multiple jewelers of different specialties and expertise: from a jewellery making apprentice to seasoned metalworking experts to jewelry distributors. Firstly, the team validated the concept of the product with jewelry making experts, ensuring the need is real and the demand is high. The team also tested different voice commands with potential users to ensure that user experience of the device was simple and intuitive. Jewelers specified a desire to issue exact commands, stating that specialized jewelers understand what a precise position or orientation, such as “2 mm” or “30 degrees,” looks like. The team then observed the process of making various pieces of jewelry, recorded videos of jewelers performing these tasks, and then used this information to recreate the process using Dextera. The team compared time duration of the jewelers’ operations to Dextera’s operations. Dextera was able to perform 5 times faster than jewelers for a simple soldering task performed on a ring: Dextera completed the necessary movements, pausing to allow the jeweler to use the blowtorch, in 26 seconds while the jeweler completed the process in 2:14 minutes. Much of the jeweler’s time was not spent soldering, instead it was spent adjusting and studying the ring.

Another important consideration for user testing included gripper reliability. The grippers must hold any object a jeweler would like to work on, without dropping or damaging the piece. To test this functionality, many different types of jewelry were placed into the grippers including rings, bracelets, watches, and necklaces. Then, Dextera simulated the movements that jewelers require to perform various tasks. The system was observed before and after the motion to check that the hold on each object was strong. After the objects were released, each pieces was checked to ensure no damage.

6.5 Robotic Analysis

In order to validate the dexterity, motion, and reachability of the system, a complete robotic analysis of the manipulator arm was performed. This entailed creating a symbolic diagram of the arm to define joint variables, derive homogeneous transformation matrices, derive forward and inverse kinematics models, and, from these, a velocity kinematics model that would inform the available motion of the arm. Additionally, using the computed kinematics model described, the team created an in-depth three-dimensional simulation of Dextera to visualize and test the motion of the product. All of the computations and simulations were done in MATLAB, and the corresponding code utilized is provided in Appendix B.1. Below is an outline of how the team was able to develop this robotic framework, as well as how it informed and validated system dexterity.

Firstly, the team devised the proper robotic diagram to define the system, shown below. As portrayed in the diagram and previously stated, this is a prismatic-revolute-revolute (PRR) manipulator arm with three degrees of freedom at the end effector.

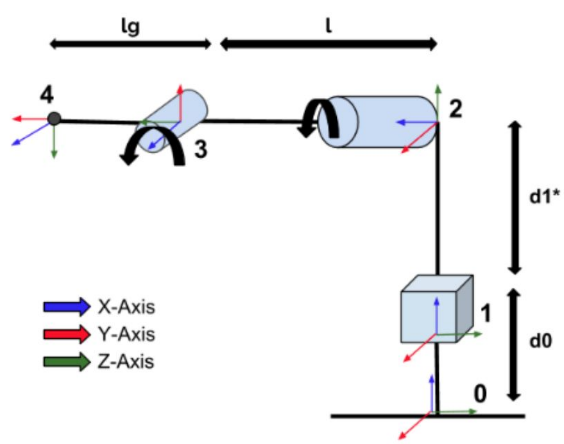


Figure 6.4: Symbolic robotic diagram of Dextera

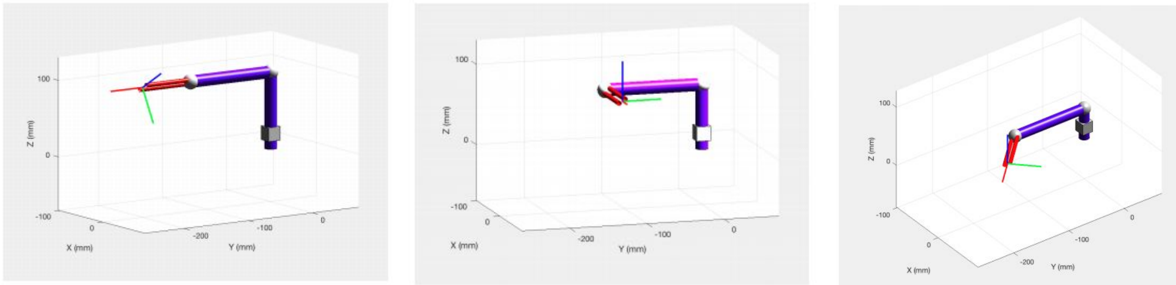
Utilizing the figure above, the team was able to define the forward kinematics of Dextera, which is described as follows. The first step in deriving the forward kinematics of the arm was determining the set of Denavit-Hartenberg parameters, which is a standardized practiced used in the Robotics industry for defining manipulator arm joint variables. Next, using these variables, the homogeneous transformation matrices corresponding to the transformations between each consecutive joint frame were derived. Both the parameters used, as well as the associated transformation matrices are provided in Appendix C.1. Multiplying each of these transformation matrices yields a resulting transformation matrix that converts the position and orientation of the tip of the gripper back into their corresponding values with respect to the base world frame given a set of joint inputs - this allowed for the conversion from joint positions and angles to gripper positions, which was used in the system's feedback loop to effectively update Dextera's configuration in software, as mentioned previously.

After defining the forward kinematics of the system, the inverse kinematics equations were derived, which allows for conversion from base world frame positions back into corresponding joint angles. This provided another bridge in the feedback loop, allowing a user's position command to be effectively turned into the necessary motor command. In order to solve for the inverse kinematics of the manipulator arm, we utilized the end effector, or gripper tip, position, which was found from the forward kinematics final transformation matrix. In doing so, the equations defining the x, y, and z positions of the end effector were delineated, and from these the corresponding joint angles were solved for. Each of these equations can also be found in Appendix C.1.

Finally, the last step in completing the kinematic model for Dextera was deriving the velocity forward and inverse kinematics of the arm, which assigns how the motion of each joint affects the motion of the end effector, and vice versa. The formulas derived in determining the inverse kinematics were utilized to compute the Jacobian matrix of the system, and from that, the linear and angular velocity kinematics of the arm. The Jacobian, as well as the equations used to define the linear and angular joint velocities are provided in Appendix C.1.

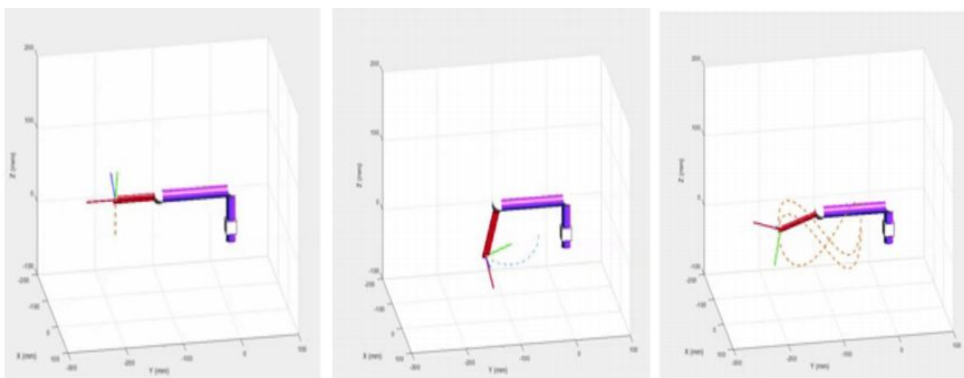
After computing both the forward and inverse position kinematics, as well as the forward and inverse velocity kinematics, the robotic system by which to control the position and motion of Dextera is now fully defined. Utilizing this analysis, the team was able to create a three-dimensional simulation that allows for better visualization and validation of system motion, as well as test the soundness of the kinematics calculations. This simulation and subsequent observations and results are outlined below.

In order to understand and visualize the feasible configurations, motions, and corresponding reachable workspace of the device, the team developed a three-dimensional Matlab simulation. Given a particular input of joint positions and angles, or a specified end effector location, the simulation displays Dextera's resulting form. This gave the ability to test a multitude of configurations to determine the access and feasibility of soldering. In addition to testing static positions, the simulation can show the movement capabilities of the arm if the inputs are joint velocities instead of positions or angles.



Figures 6.5-6.7: Dextera at different orientations.

The images above show different configurations of Dextera in the simulation. The linear joint is represented by a grey cube while the two rotational joints are grey spheres. The arms are purple cylinders and the grippers are the red end effectors. In figure 6.5, joint 1 is extended 80 mm, joint 2 is set to 30 degrees , and joint three is set to 0 degrees. In Figure 6.6, joint 1 is extended at 60 mm while both joints 2 and 3 are set to 90 degrees. In Figure 6.7, joint 1 is extended 30 mm, joint 2 is set to 90 degrees while joint 3 is set to 45 degrees.



Figures 6.8-6.10: Simulating joint velocities

Finally, in the above images, the simulation shows the path that the end effector traces out as the robot moves given a set of joint velocities. In Figure 6.8, joint 1 is set to 25 mm/s, joint 2 is set to 10 rad/s and joint 3 is set to 0 rad/s; thus, the arm moves upward while the second link rotates continuously as the end effector's path traces a straight line upwards. In Figure 6.7, joint 1 is set to 10 mm/s, joint 2 is set to 0 rad/s, and joint 3 is set to 10 rad/s; an arc is formed by the end effector. In figure 6.10, joint 1 is set to 25 mm/s, while joints 2 and 3 are set to 10 rad/s. The end effector traces an arced figure eight while moving upwards.

7. Discussion

Once each subsystem, as well as the device as a whole was validated using the methods described above, the team was able to reflect upon the successes and failures of the final product, as the metrics of the prototype compare to the earlier defined system-level objectives. In the Sections below, each of the earlier objectives, defined in Section 4, is analyzed with respect to the final prototype. Additionally, the product's compliance with the necessary engineering standards, which are also outlined in Section 4, is looked into. Finally, reflections on the work completed throughout the scope of Senior Design and the successes and failures of Dextera at large are presented via recommendations for future work and device improvements.

7.1 Meeting Desired Objectives

The team was able to meet its desired objectives with respect product capability and ability to meet user needs. The following Sections highlight the important characteristics of the device that pertain to usability. Specifically, this includes reachability, heat resistance, load bearing capabilities, product form factor, efficiency, and reliability. Characteristics such as energy were non-issues for the team as the device uses minimal energy, easily powered by a standard wall outlet, which is available to all stakeholders interviewed.

7.1.1 Reachability and Orientation

The final system is able to reach every orientation a jeweler may require. The team took videos of jewelers working on various pieces and were able to recreate the exact orientations and transitions using Dextera. The final system is six degrees of freedom all together--the up and down motion of the arm, the 360 degree rotation of the arm, the 180 degree rotation of the wrist, and the individual open and close motion of each gripper. This allows jewelers to access every point on the work object with the voice controls, which was the original goal of the project.

7.1.2 Heat-Resistance

By using stainless steel on the grippers with a stainless steel heat shield at the wrist, the team was able to ensure a minimum continuous soldering time of 15 minutes. From interviews with jewelers, this soldering time was confirmed to be in excess of what they would require to finish working on an object. Even with hotter temperature flames, Dextera can remain reliable for minutes at a time. The casing around the internal components also adds to the heat resistance of the system, since it will prevent any accidental melting if a user inadvertently points the flame where it shouldn't be.

7.1.3 Load-Bearing Capabilities

The original goal metric for this system characteristic was defined by 3.0 N of static force, with an assumed additional 2.0 N for dynamic forces from user-interfacing with the device. As shown in the Validation Section 6.2, the final design of Dextera was confirmed to be able to support these identified target loads through both finite element analysis (FEA) methods and physical testing. Described in greater detail earlier, the FEA on the device was done in SolidWorks, on the assembly of the internal arm structure. Alternatively, physical testing included placing a number of jewelry objects in the grippers, simulating additional user loading by pressing on the grippers at the end effector and on various areas of the arm, and cycling Dextera through a number of movements while holding various work objects. Through this physical testing, along with the more detailed structural validation described earlier, the team confirmed that the final prototype successfully hit load-bearing capability metrics.

7.1.4 Reachable Workspace, Storage, and Weight

Stated previously in the objectives Section, and later confirmed with additional user interviews and jeweler workshop visits, the reachable workspace and storage size of the device are primarily limited by the small size and cluttered area that defines the jeweler's workbench. Thus, a major concern throughout Senior Design was designing a miniaturized product. After successfully designing towards this metric, the final prototype for Dextera was confirmed to be small enough to fit within the jeweler's confined space. Additionally, the wrist of Dextera was designed with the goal of a spherical reachable workspace in mind - to cater to the observed types of motions jewelers must perform while soldering. With the inclusion of both 360-degree rotation and 180-degree panning capabilities, this goal metric was also achieved. Finally, the goal weight of the system, for less than 45 N, was driven by the need for the user to be able to lift it and move it around his or her workbench. The final prototype weighed in at 33.36 N, meeting this goal as well. Thus, analyzing adherence to the earlier defined goals of a spherically-bounded reachable workspace, minimal storage size, and a light weight, Dextera meets or exceeds all of the goal metrics for these specific system characteristics.

7.1.5 Time Constraint

As discussed earlier, meeting the time constraints was one of the most important benchmarks to hit. It was a primary reason for undergoing the project at the outset, and, should the standard not be met, the device would be rendered impractical. No jeweler would sacrifice time unless the physical gains in relieving pain from manual labor were exorbitantly significant.

Dextera was able to perform the average command in under the 5 second requirement and almost at par with the 4 second goal (at 4.09 seconds). However, one metric that was not in a benchmark was the worst case time to complete a task. Given the success rate of 80%, it is highly unlikely that there would be 3 errors in a row (0.8%), and almost impossible for there to be 4 in a row (0.16%) -- and these errors are likely missed commands and not incorrect commands, which have a far less severe time penalty. This yields a worst case scenario time of approximately 11 seconds. Discussing more likely scenarios though, Dextera took 7 seconds to complete a task after 1 incorrect operation, compared to the roughly 5 second time the jeweler took in the best and worst case scenarios.

7.1.6 Efficacy and Reliability

Just like the time constraint above, efficacy and reliability are vital to the success of Dextera. If it could not hold items properly, Dextera would run the risk of breaking or damaging expensive pieces. Furthermore, the product's downside would outweigh the gains made, causing the product to have little real world use.

Overall, Dextera did well in this area. The motor design allowed for the piece to be held on firmly, without a significant risk of damaging the item. The gripper design allowed for many different configurations of placement and holding style, such as between both sets of grippers, wrapped around one gripper prong, between one set of grippers, among others. A risk seen here, though, is that misuse by the user may cause reliability issues.

Further iterations of gripper design would make this aspect of dextera even more concrete, eliminating aforementioned misuse.

7.2 Measuring up to Engineering Standards

While no official tests were performed to determine if Dextera met the relevant engineering standard, these standards were used to shape the design of the final system form. As mentioned in the previous subSection on engineering standards, the team made sure to take the necessary steps in order to ensure that the product would have succeeded in these tests. The software has safety limits on the user's ability to control the movement as well as the system's ability to move on its own. This ensures that Dextera does not move too fast or too far from the designated working area. The casing of the system was also carefully designed to ensure maximum flame resistance and minimize heat transfer throughout. Any flammable or susceptible components had heat shields in front of them as well.

7.3 Recommendations

Taking a step back and analyzing the above discussion of each design objective, it can be noted that Dextera has met, and occasionally exceeded, nearly all system goals and metrics that were initially set forth. Theoretical and physical testing methods showed the device could withstand the required thermal and structural loads, that it can be built at the desired price point, and that it can respond accurately to spoken commands. Further testing also showed that Dextera has the flexibility to reach the appropriate orientations for jewelers, the target user, and help reduce the time it takes for them to solder. However, if this product were to be brought to market, there are several other features that would be beneficial to implement as well as tests that would be necessary to perform.

Firstly, a concern of the target user that the team was unable to validate with this prototype was the assurance of protection of fragile work objects. Many of the items that jewelers work with are small, delicate, and fragile, which makes them susceptible to damage under high loads. While the testing performed, described in Section 6.5, confirmed that the forces imposed by the grippers would not in fact damage the objects it held, this could be taken a step further, with the incorporation of a softer hold or force feedback. For example, the grippers could have been manufactured with a more pliable heat-resistant material at the tip, such as a layer of metal mesh. Alternatively, there could be sensors placed at the tip of each gripper, for the user to be able to specify a level of gripping strength or a forcing limit, with which the system software would be able to process and adhere to via the feedback from the attached sensors.

Additionally, while testing showed that the system responded to commands with a reasonably high degree of certainty as stated in Section 6.4, in an updated iteration of the prototype, this would be improved upon even further by expanding the types of allowable commands and including a dictionary of only commonly used words and phrases. As the software currently stands, the recognizable words and phrases are processed with Google's dictionary, since it is developed with Google's Speech-to-Text Software Development Kit (SDK) - the process by which this is done is described in greater detail in Section 5.3.1. It could be effective to instead create a smaller dictionary of only words that would be necessary to give Dextera commands, eliminating superfluous words and thereby minimizing the potential for parsing speech incorrectly. This dictionary could be built into the currently-used program, on top of the Google SDK, as to not decrease from the present software functionality. Additionally, allowing a more flexible command structure would increase the usability of the device. For example, incorporating more relative commands, including prepositional modifiers, such as "to me," "away from me," may provide a more intuitive interface with the product. Including both a smaller dictionary, as well as expanding the set of allowable commands would take much thought, effort, and iteration in order to ensure that all commands and uses are appropriately accounted for.

Lastly, in addition to providing force feedback on the grippers and refining the voice recognition clarity, if the team were to propose future work for this project, it would be largely focused on adapting the product to be applicable for a wider range of use cases. Jewelers were determined to be an impactful user group to hone in on for the entirety of Senior Design, as they provided the most deliberate and niche use cases, as well as the most particular specifications of the groups considered. Thus, the team focused on attending to this user group for the first final prototype of Dextera, with the intention of later scaling and adapting for use by other groups. Thus, with adaptability in mind, the design could be modified to have more modular components, particularly in the wrist and gripper area, that would allow Dextera to hold and manipulate a variety of other objects - perfboards and wires for engineers and eye glasses for opticians, to name a few. Additionally, thinking on a broader scope, this first prototype of Dextera could be just a foundational product in a line of several voice-controlled robotic arm soldering assistants, with others tailored specifically to the other researched use cases.

Had these additional features, modifications, and potential new products been prototyped, the last additional steps the team would have taken to make Dextera a consumer-ready product would have been getting it in the hands of real users to work with in industry. Unfortunately, given the time and resource constraints that accompanied working with jewelers throughout South Philadelphia, the team was unable to physically test the model in the hands of an actual stakeholder. While the alternative methods used for usability validation were effective as a foundation, putting the device to test with a real user would be a crucial step before bringing Dextera to market.

Thus, despite delivering a nearly-consumer-ready prototype in the Senior Design timeline, there are still a few recommendations for future work that would have enhanced and reinforced the robustness, scalability, and usability of the system. Namely, redefining the gripper design to incorporate a soft-touch tip or force-sensor feedback, constructing a usage-specific dictionary and a broader command range for voice control, modularizing or re-creating Dextera with additional use cases in mind, and finally, getting the prototype in the hands of an actual user, to confirm system usability in industry. Should time and resources have permitted, the team would have began embarking on these ventures to turn Dextera from a polished prototype to a sellable consumer product.

8. Budget and Resources

8.1 Senior Design Budget and Spending

The budget consisted of $2,400 given by the Mechanical Engineering and Applied Mechanics (MEAM) department at the University of Pennsylvania. The general breakdown of the distribution of these funds is displayed in the table below. Items that represented a significant portion of the cost of the final design are in bold. These larger purchases include 3 gearmotors, 2 linear actuators, machining stock, the Raspberry Pi, and the headset. Total spending was $655.11 and thus did not require any funding outside of the budget given by the MEAM Department. All work was completed with the machinery and facilities available in Penn Engineering.

Table 8.1: Total Itemized Team Spending

Item	Price	Quantity	Expense
Asana Software	$75.00	1	$75.00
8mm Stainless Steel Precision Shafting	$4.69	4	$18.76
8mm Lead Screw, 650mm Long	$13.99	1	$13.99
8mm Lead Screw Barrel Nut	$4.99	2	$9.98
8mm Linear Ball Bearings (2 pack)	$6.99	2	$13.98
Micro Gear Motor Enclosure	$0.99	2	$1.98
1/8" ID x 5/16" OD Flanged Ball Bearing (2 pack)	$1.99	2	$3.98
Set Screw Shaft Coupler, 1/8 to 3mm	$4.99	1	$4.99
Lynxmotion Pan and Tilt Kit/Aluminium	$29.93	1	**$29.93**
1" Bore Side Tapped Clamping Mount	$5.99	2	$11.98
1" Bore Side Tapped Pillow Block	$9.99	2	$19.98

1" Bore, Face Tapped Clamping Hub, 1.50" Pattern	$5.99	2	$11.98
3.00” Aluminum Channel	$3.99	1	$3.99
1" Aluminum Tubing, 10" Length	$3.49	1	$3.49
Standard Servo Plate D	$6.99	2	$13.98
1000:1 Micro Metal Gearmotor HPCB 6V with Extended Motor Shaft	$25.96	2	**$51.92**
Magnetic Encoder Pair Kit for Micro Metal Gearmotors, 12 CPR, 2.7-18V (HPCB compatible)	$8.95	3	**$26.85**
Raspberry Pi 3 - Model B+ 1.4 GHz Cortex- A53 with 1GB RAM	$35.00	1	**$35.00**
Feedback 360 Degree - High Speed Continuous Rotation Servo	$27.99	1	**$27.99**
Standard Servo Plate C	$5.99	1	$5.99
Magnetic Encoder Pair Kit for Micro Metal Gearmotors, 12 CPR, 2.7-18V (HPCB compatible)	$8.95	2	$17.90
250:1 Micro Metal Gearmotor HP 6V with Extended Motor Shaft	$16.95	1	$16.95
STAINLESS #8 SHEET 304 ANNEALED - Thickness - 0.045" - 12" x 36"	$59.70	1	**$59.70**

COLD DRAWN STAINLESS RECTANGLE 303 ANNEALED - two foot length	$15.48	1	$15.48
Plantronics Savi W430 Wireless Headset - Silver/Black	$159.34	1	**$159.34**
Total			**$655.11**

8.2 Product Cost Projections

Not including the purchase of Asana productivity software and the Wireless Headset, the team spent only $495.77 to make Dextera. To make the product in mass quantities, the team feels extremely confident that the unit cost could be brought down to at least $250. This cost reduction would come largely from economies of scale and the elimination of the margins taken by suppliers, but it also would come from more creative cost-saving methods. For instance, only some of the components of the off-the-shelf Raspberry Pi purchased are actually required for the system, and thus a significantly pared down version would be used when producing the product at scale. Another cost-saving measure would be the microphone. To keep the price that the consumer pays for Dextera as low as possible, the product would be sold without the expensive headset and allow the user to utilize whatever bluetooth microphone/headset they may already have. For reasons such as these, the final product has a higher cost than the market-ready product would, as is typical for prototypes. The high-level cost projection at scale to arrive at the $250 total is shown below.

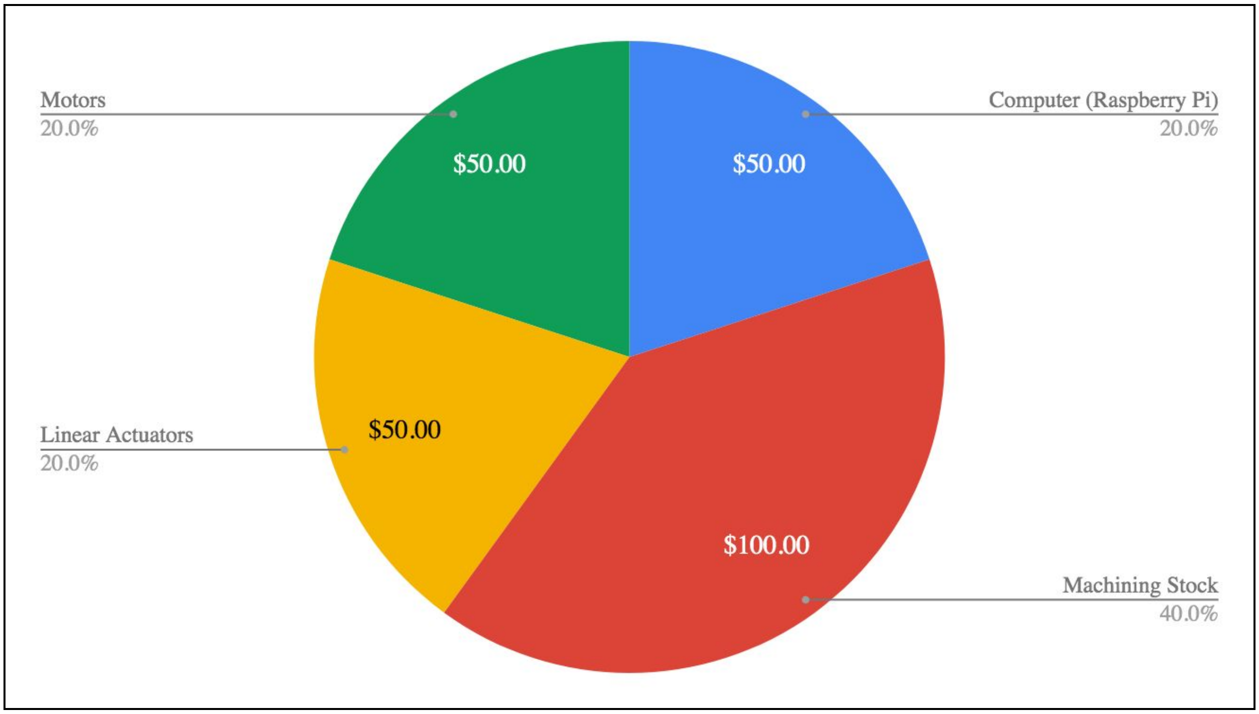


Figure 8.1: Projected Unit Cost at Scale

8.3 Pricing and Market Sizing

According to a recent McKinsey study, worldwide jewelry sales should reach over $250B by next year, growing at a healthy clip of 5 to 6 percent each year.[16] With the rise of non-commercial jewelry manufacturing such as crafted products on Etsy, the jewelry manufacturing industry is increasingly becoming hands-on. There are roughly 20,000 potential US customers for Dextera just from non-commercial jewelry manufacturers alone. A consumer-facing product such as Dextera would disrupt this large and growing market, becoming an essential tool for jewelers to make large quantities of jewelry in a cost efficient and convenient manner.

Assuming the cost reductions described in Section 8.2 could be achieved, the team believes that a price point of $329.99 would be reasonable for Dextera. This represents a gross margin of 34%, which is consistent with similar industrial electronics products on the market.[27] Interviews with stakeholders have shown that potential users are willing to pay this value.

References

[1] J. Green, Jennifer Green Custom Jewelery, 2018.

[2] T. Caputo, John M. Huntsman Hall 526.7, 2018.

[3] "Jewelers and Precious Stone and Metal Workers : Occupational Outlook Handbook", Bls.gov, 2018. [Online]. Available: https://www.bls.gov/ooh/production/jewelers-and-precious-stone-and-metal-workers.htm#tab-6. [Accessed: 04- Nov- 2018].

[4] S. Sanford, "Soldering 101: Gold & Silver Solder Melting Points & More", Interweave, 2011. [Online]. Available: https://www.interweave.com/article/jewelry/solder-101-its-forms-and-melting-temperatures-for-successfully-soldering-jewelry/. [Accessed: 04- Nov- 2018].

[5] "Quickly estimate the weight of your jewellery", Goldbestprice.net, 2015. [Online]. Available: http://goldbestprice.net/Weight-jewellery. [Accessed: 04- Nov- 2018].

[6] A. Williams, "How much power does your smart home tech really use?", The Ambient, 2018. [Online]. Available: https://www.the-ambient.com/features/power-smart-home-tech-yearly-cost-374. [Accessed: 04- Nov- 2018].

[7] P. Gavin, "Average Ring Size for Men and Women", Myirishjeweler.com, 2018. [Online]. Available: https://www.myirishjeweler.com/blog/what-is-the-average-ring-size. [Accessed: 04- Nov- 2018].

[8]A. Gregson, "A Simple Guide to Choosing Your Chain [INFOGRAPHIC] - Stuller Blog", Stuller Blog, 2018. [Online]. Available: https://blog.stuller.com/simple-guide-choosing-your-chain/. [Accessed: 04- Nov- 2018].

[9] Robots and robotic devices--Safety requirements for personal care robots, 1st ed. Geneva: International Standard Organization, 2014, pp. 1-11.

[10] Functional safety of electrical/electronic/ programmable electronic safety-related systems – Part 3: Software requirements, 1st ed. International Electrotechnical Commission, 1998, pp. 1-11.

[11] B. Haas, "IP 66, IP 65 and IP 67 Ratings Explained", Bud Industries Blog, 2014. [Online]. Available: http://www.budind.com/blog/2014/02/the-mysteries-of-ip-rated-enclosures-explained/. [Accessed: 04- Nov- 2018].

[12] "UL - 94 Standard for Tests for Flammability of Plastic Materials for Parts in Devices and Appliances", Standardscatalog.ul.com, 2013. [Online]. Available: https://standardscatalog.ul.com/standards/en/standard_94_6. [Accessed: 04- Nov- 2018].

[13] "ASTM E1725 - 14e1 Standard Test Methods for Fire Tests of Fire-Resistive Barrier Systems for Electrical System Components", Astm.org. [Online]. Available: https://www.astm.org/Standards/E1725.htm. [Accessed: 04- Nov- 2018].

[14] "Soft Robotics", Biodesign.seas.harvard.edu, 2018. [Online]. Available: https://biodesign.seas.harvard.edu/soft-robotics. [Accessed: 04- Nov- 2018].

[15] A. Conner-Simons, "What is a soft robot?", World Economic Forum, 2015. [Online]. Available: https://www.weforum.org/agenda/2015/10/what-is-a-soft-robot/. [Accessed: 04- Nov- 2018].

[16] A. Moultry, "What Materials Are Fireproof? | Hunker", Hunker.com, 2018. [Online]. Available: https://www.hunker.com/12444429/what-materials-are-fireproof. [Accessed: 04- Nov- 2018].

[17] "Lynxmotion - Robotic Arms", Lynxmotion.com, 2018. [Online]. Available: http://www.lynxmotion.com/c-27-robotic-arms.aspx. [Accessed: 04- Nov- 2018].

[18]"Pololu-1000:1 Micro Metal Gearmotor HPCB 6V with Extended Motor Shaft", *Pololu Robotics & Electronics.* [Online]. Available: https://www.polulu.com/product/2080/specs. [Accessed: 04-Nove-2018]

[19]"Nomex Fabric", Sternandstern.com, 2018. [Online]. Available: http://www.sternandstern.com/category/53/Nomex-Fabric.html. [Accessed: 04- Nov- 2018].

[20] K. O'Donahue, "What Temperature Is a Propane Torch?", Sciencing, 2018. [Online]. Available: https://sciencing.com/temperature-propane-torch-5911.html. [Accessed: 04- Nov- 2018].

[21] B. Donald, *Forward Kinematics: The Denavit-Hartenberg Convention*. 2011.

[22] "Metal Prices - Steel, Aluminum, Copper, Stainless, Rare Earth, Metal Prices, Forecasting | MetalMiner", *Steel, Aluminum, Copper, Stainless, Rare Earth, Metal Prices, Forecasting | MetalMiner*, 2018. [Online]. Available: https://agmetalminer.com/metal-prices/. [Accessed: 11- Dec- 2018].

[23] "Metals - Melting Temperatures", *Engineeringtoolbox.com*, 2018. [Online]. Available: https://www.engineeringtoolbox.com/melting-temperature-metals-d_860.html. [Accessed: 11- Dec- 2018].

[24] "Thermal Conductivity of Metals", *Engineeringtoolbox.com*, 2018. [Online]. Available: https://www.engineeringtoolbox.com/thermal-conductivity-metals-d_858.html. [Accessed: 11- Dec- 2018].

[25] "Tungsten Price - Metalary", *Metalary*, 2018. [Online]. Available: https://www.metalary.com/tungsten-price/. [Accessed: 11- Dec- 2018].

[26] "A Multifaceted Future: The Jewelry Industry In 2020". *Mckinsey & Company*, 2019, https://www.mckinsey.com/industries/retail/our-insights/a-multifaceted-future-the-jewelry-industry-in-2020.

[27] "Operating And Net Margins". *Pages.Stern.Nyu.Edu*, 2019, http://pages.stern.nyu.edu/~adamodar/New_Home_Page/datafile/margin.html.

Appendix

Appendix A.1 Circuit Diagrams

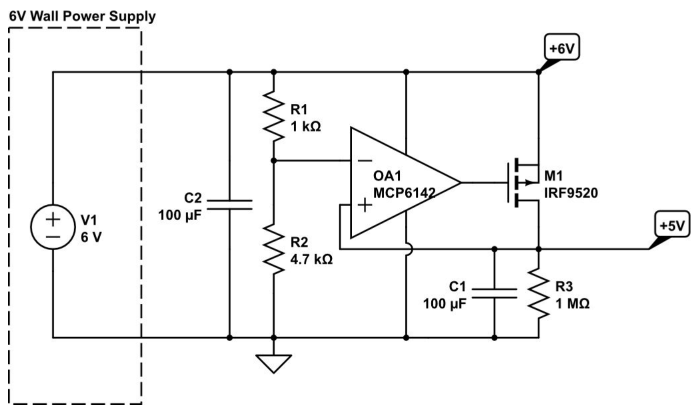


Figure A.1.1: 5V Regulator Schematic

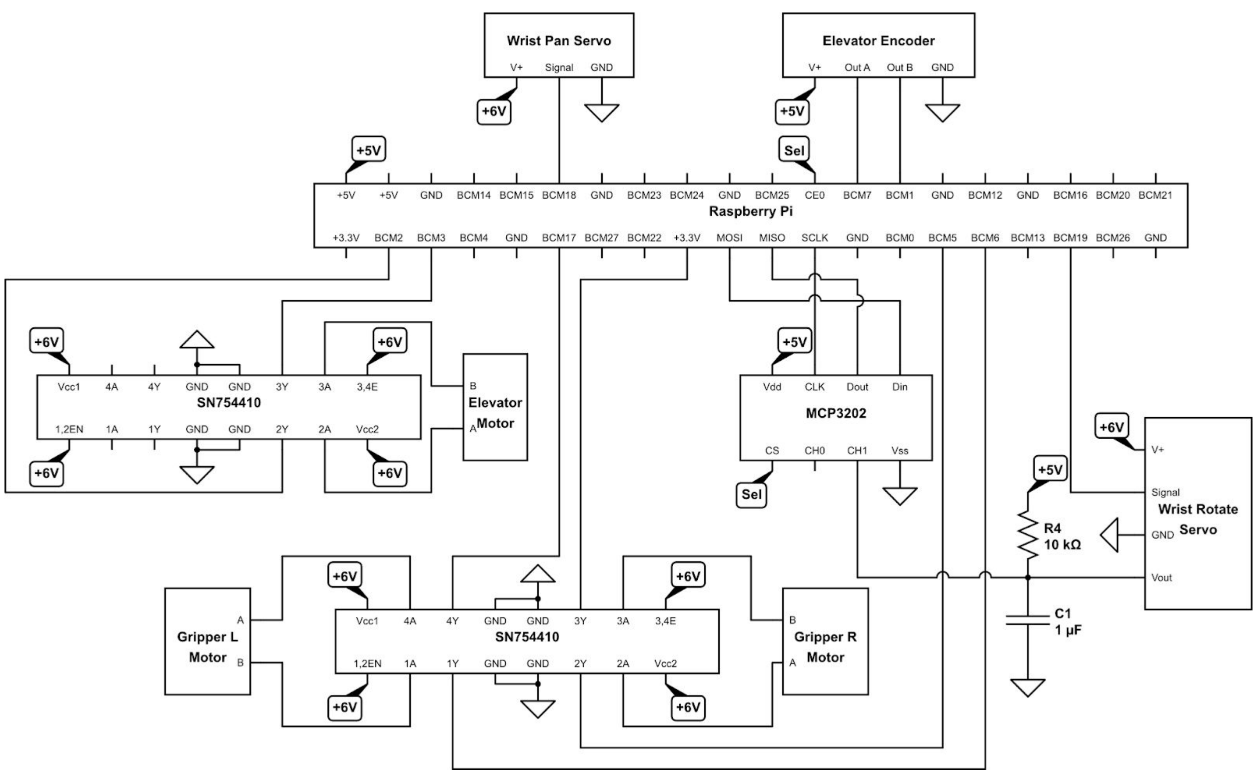
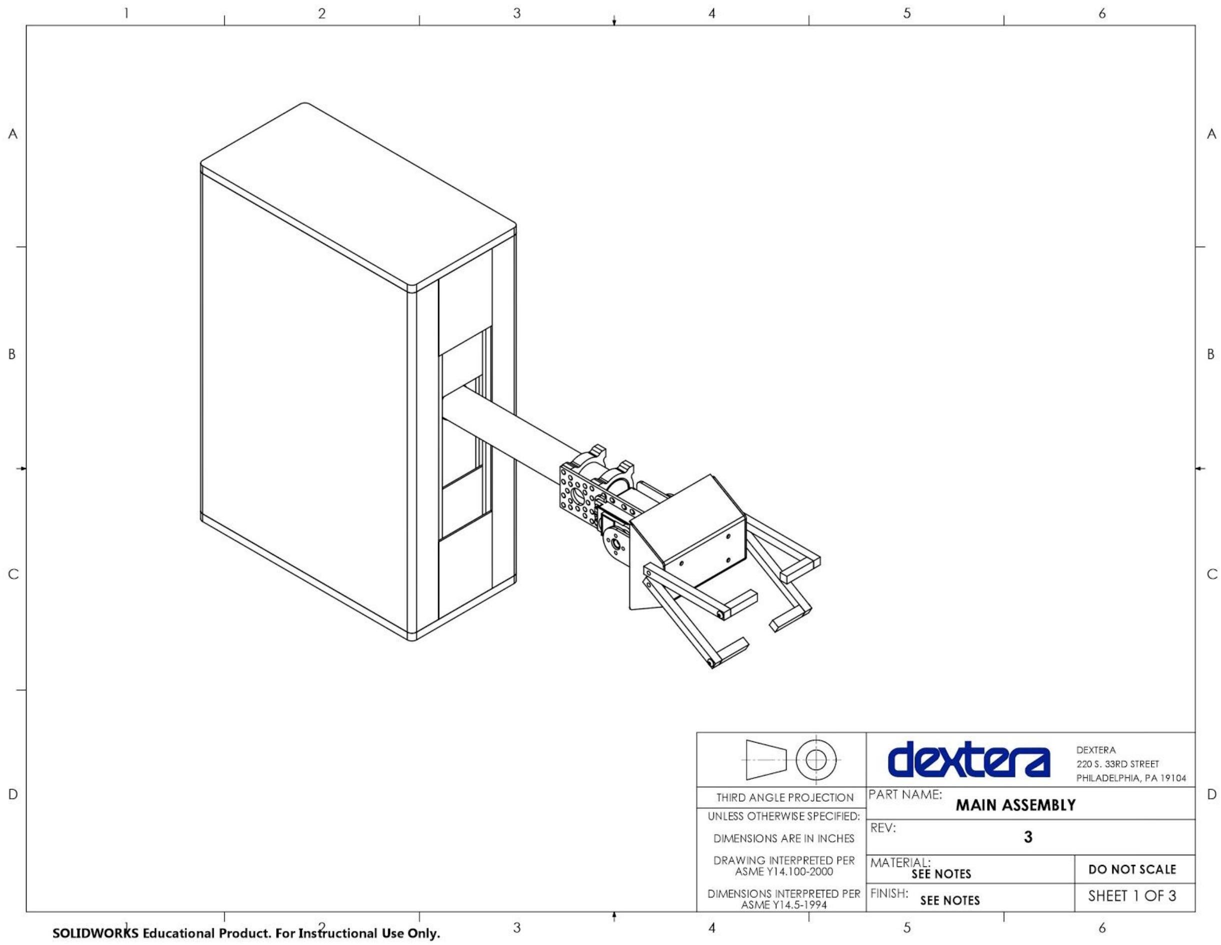
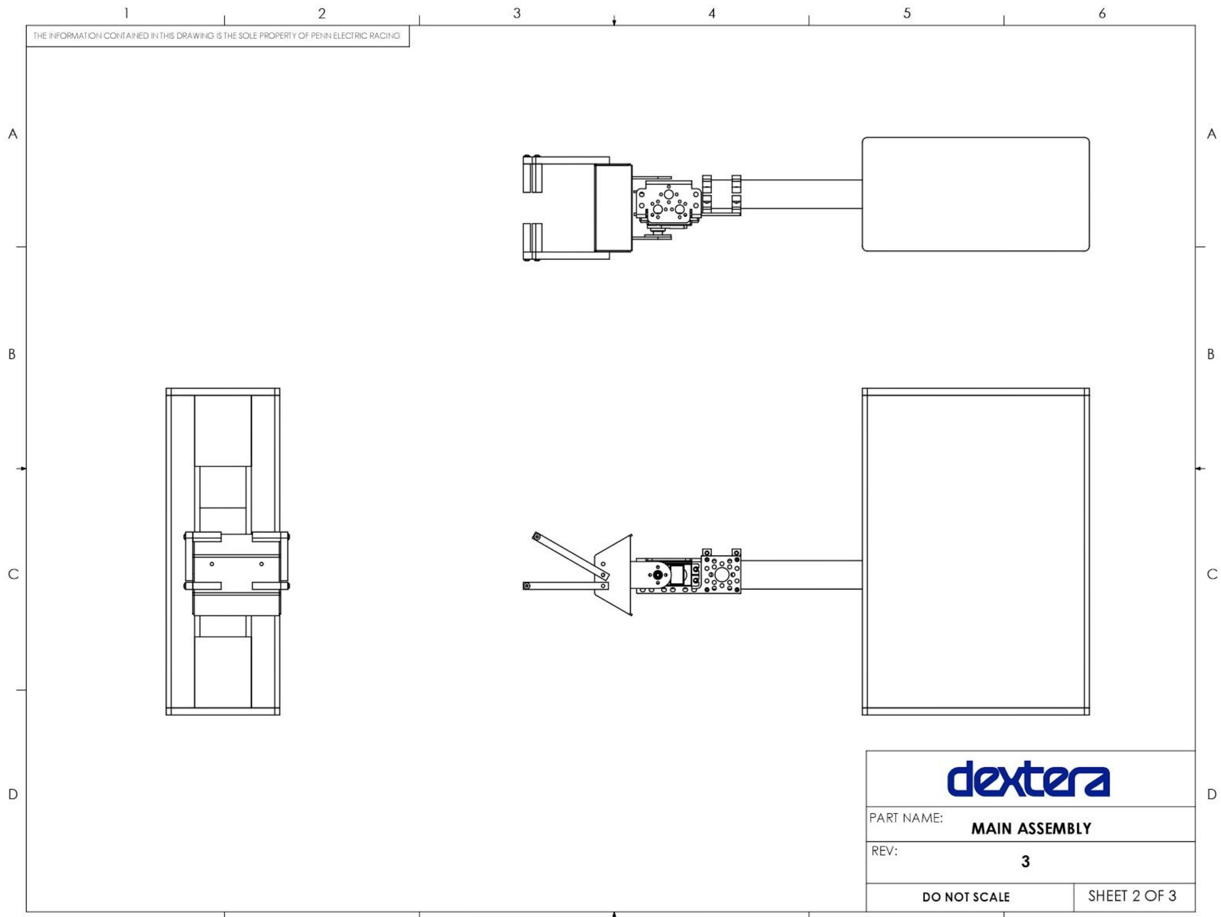


Figure A.1.2: Main Electronics Schematic

Appendix A.2 Engineering Drawings

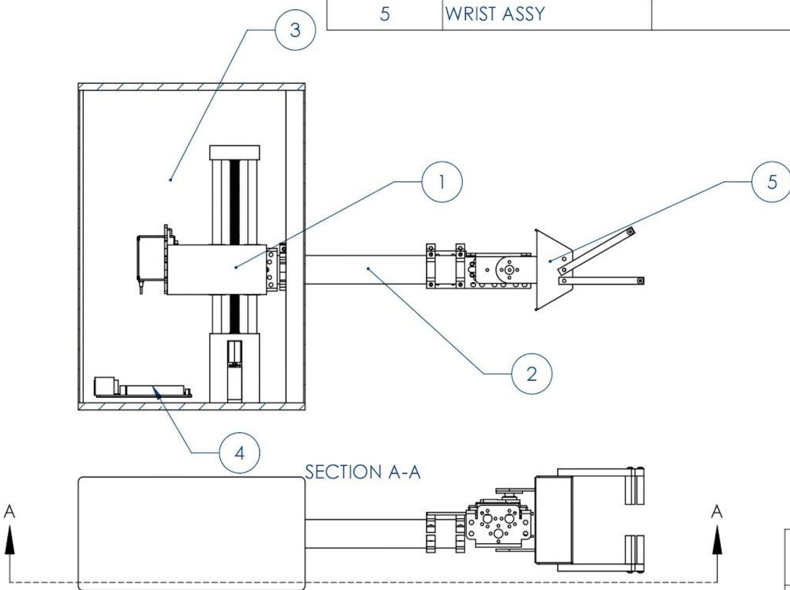




SOLIDWORKS Educational Product. For Instructional Use Only.

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF PENN ELECTRIC RACING

ITEM NO.	PART NUMBER	DESCRIPTION	Default/ QTY.
1	VERTICAL TOWER		1
2	HORIZONTAL TUBE		1
3	CASING ASSY		1
4	Rpi3Bplus		1
5	WRIST ASSY		1



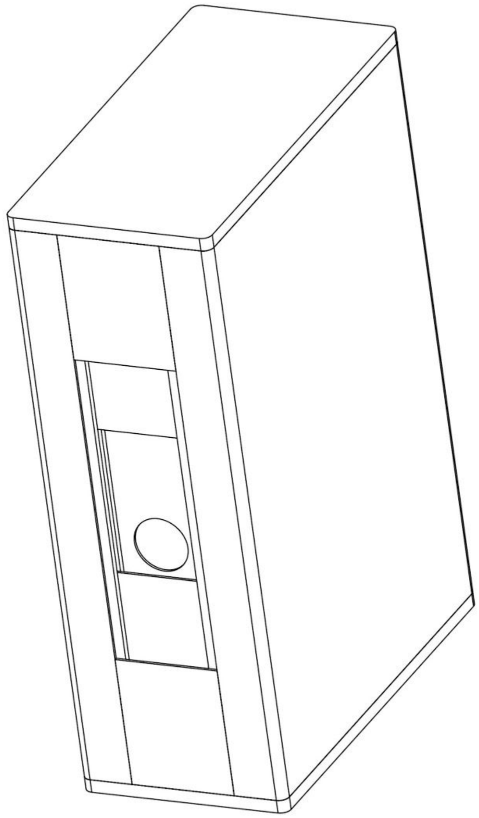
dextera

PART NAME: **MAIN ASSEMBLY**

REV: **3**

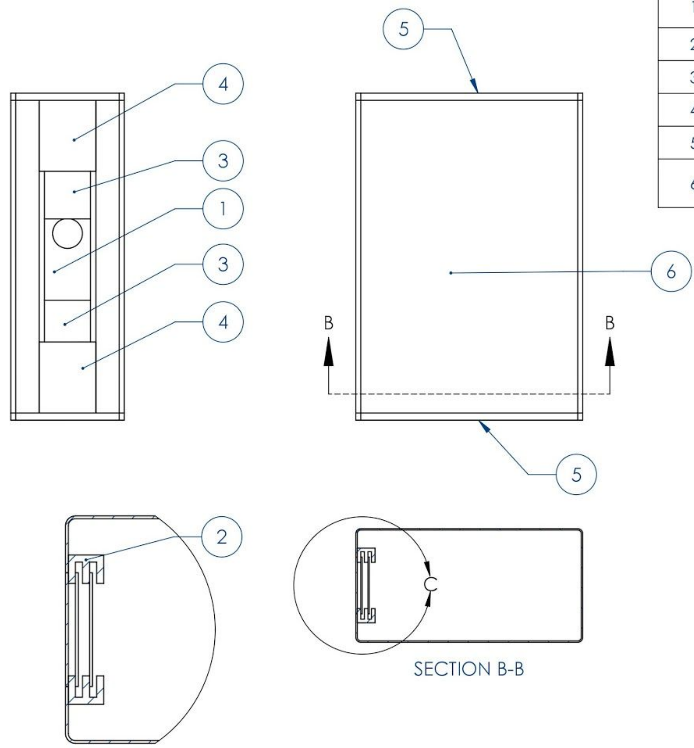
DO NOT SCALE | SHEET 3 OF 3

SOLIDWORKS Educational Product. For Instructional Use Only.



THIRD ANGLE PROJECTION	DEXTERA 220 S. 33RD STREET PHILADELPHIA, PA 19104	
UNLESS OTHERWISE SPECIFIED:	PART NAME: MAIN ASSEMBLY	
DIMENSIONS ARE IN INCHES	REV: 3	
DRAWING INTERPRETED PER ASME Y14.100-2000	MATERIAL: SEE NOTES	DO NOT SCALE
DIMENSIONS INTERPRETED PER ASME Y14.5-1994	FINISH: SEE NOTES	SHEET 1 OF 2

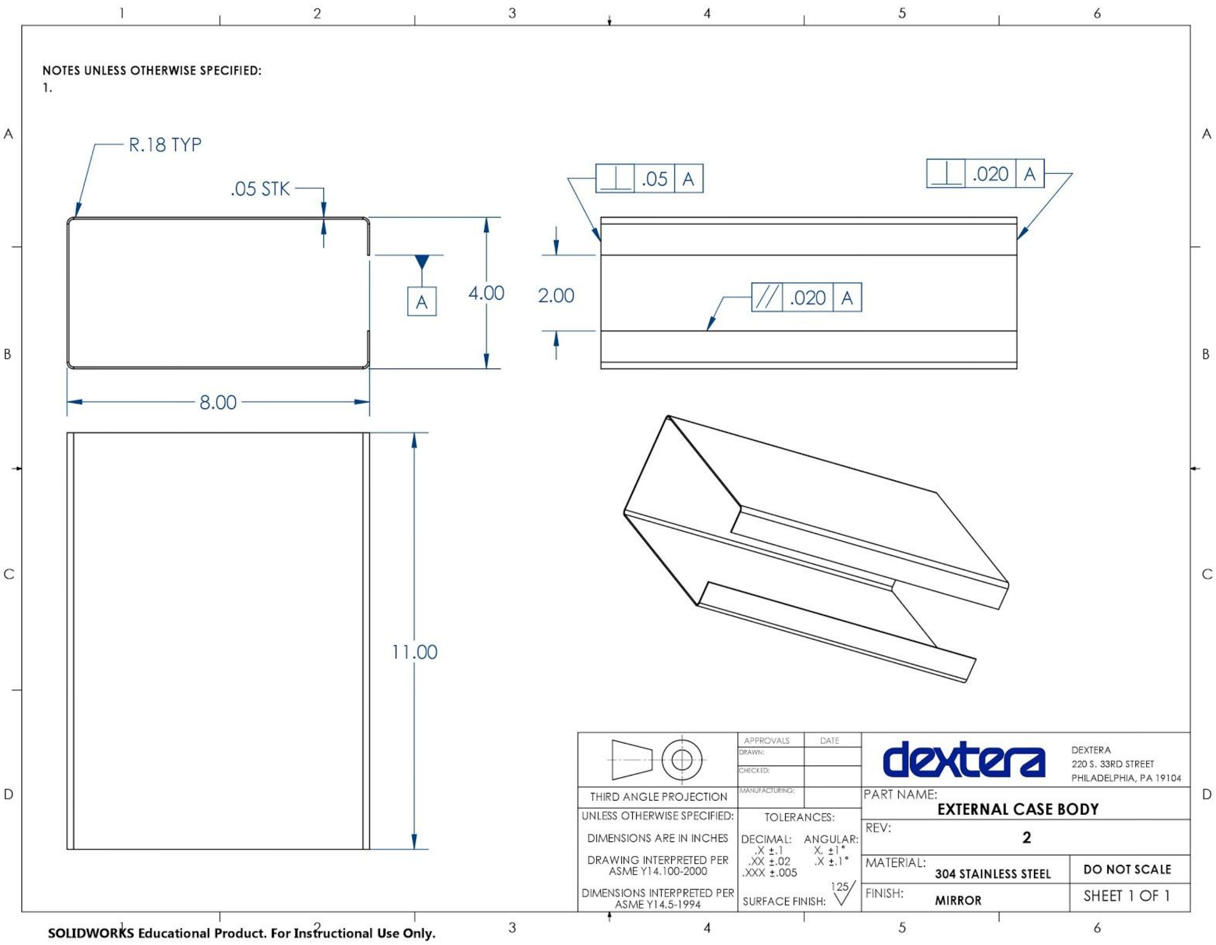
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	VERTICAL SLIDE MIDDLE COVER		1
2	GUIDE RAIL		2
3	SLIDING COVER		2
4	FIXED COVER		2
5	LIID		2
6	EXTERNAL CASE BODY		1

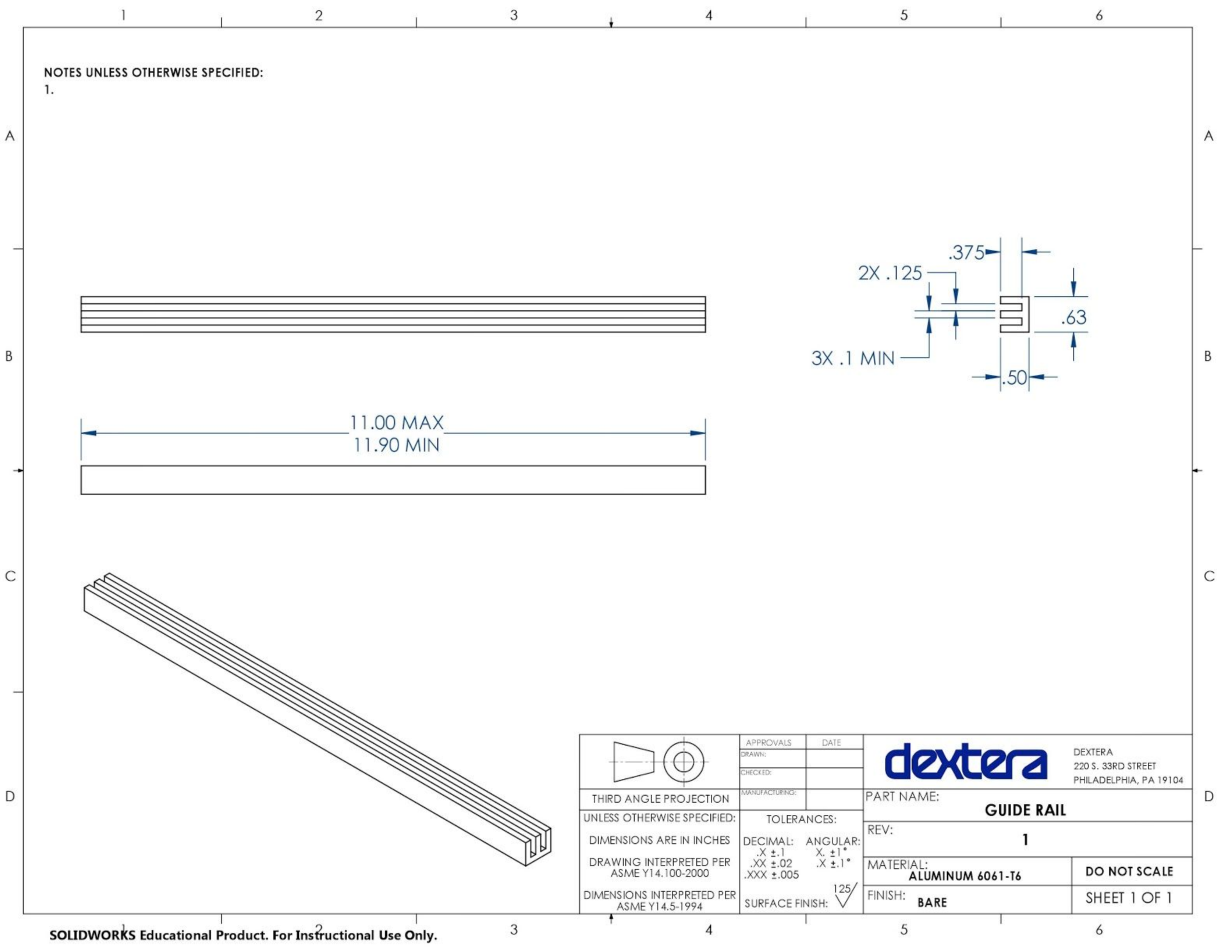


DETAIL C

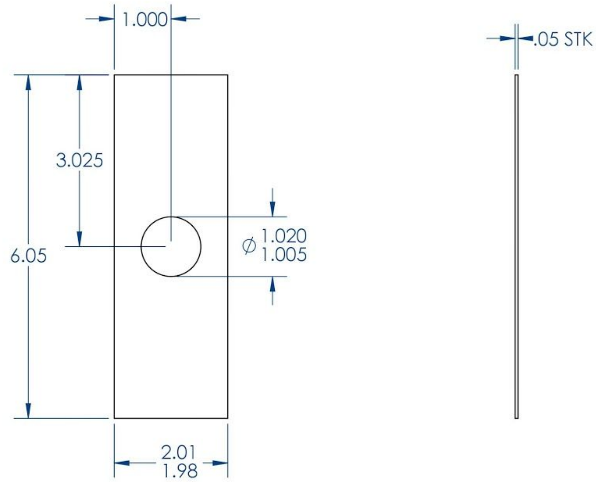
dextera

PART NAME:	**MAIN ASSEMBLY**
REV:	**3**
DO NOT SCALE	SHEET 2 OF 2

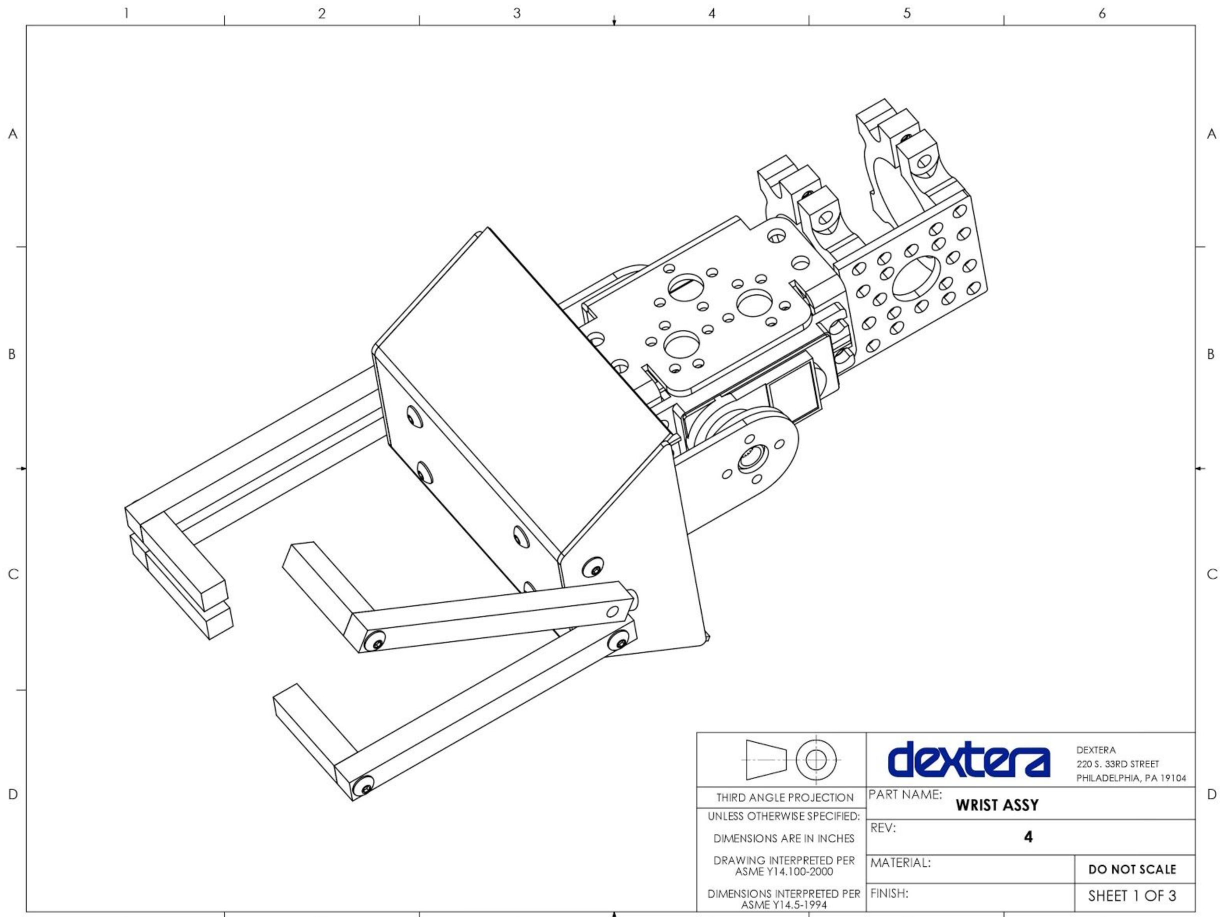

NOTES UNLESS OTHERWISE SPECIFIED:
1.
R.18 TYP
.05 STK
A
4.00
2.00
8.00
11.00
.05 A
.020 A
.020 A
THIRD ANGLE PROJECTION
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
DRAWING INTERPRETED PER ASME Y14.100-2000
DIMENSIONS INTERPRETED PER ASME Y14.5-1994
TOLERANCES:
DECIMAL: .X ±.1 .XX ±.02 .XXX ±.005
ANGULAR: X. ±1° .X ±.1°
SURFACE FINISH: 125
APPROVALS
DATE
DRAWN:
CHECKED:
MANUFACTURING:
dextera
DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104
PART NAME:
EXTERNAL CASE BODY
REV:
2
MATERIAL:
304 STAINLESS STEEL
DO NOT SCALE
FINISH:
MIRROR
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

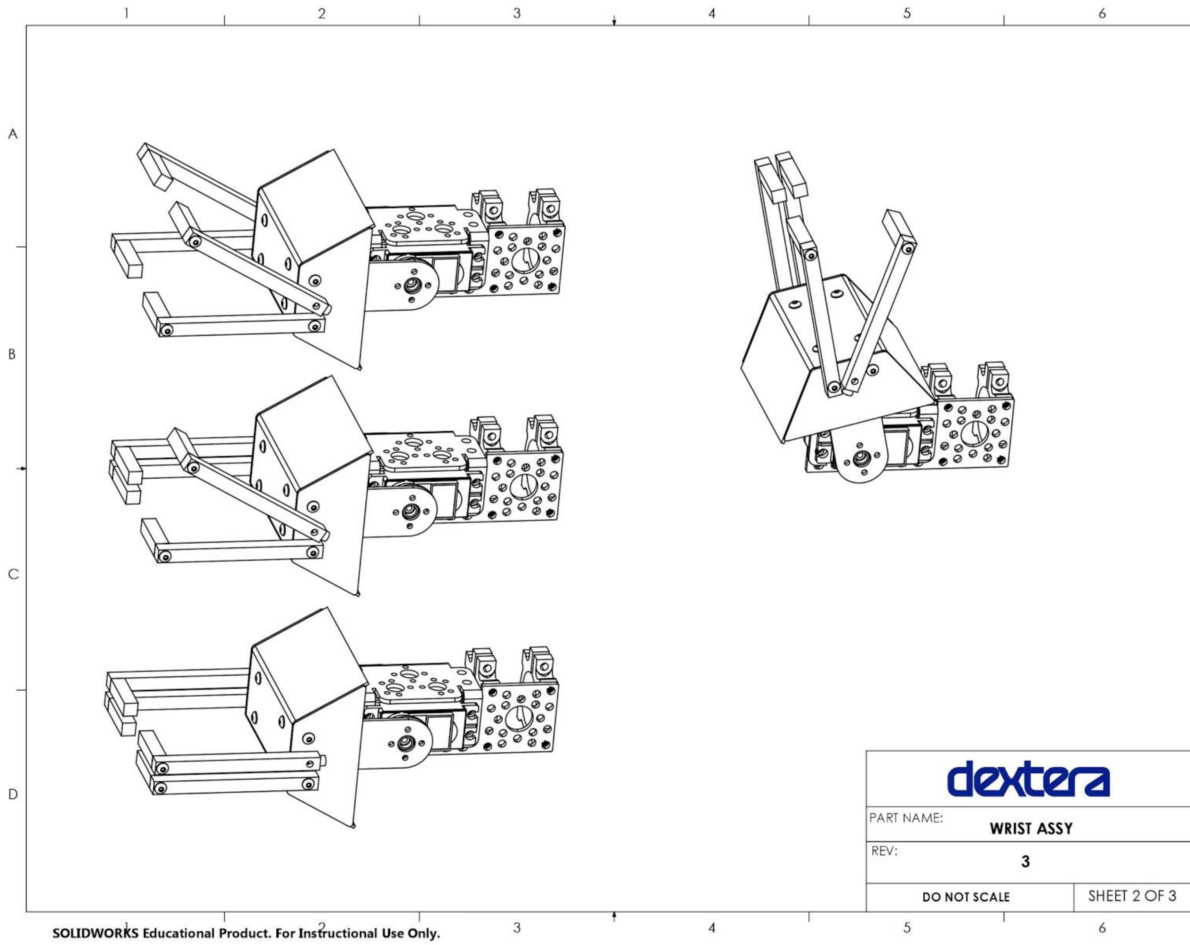

NOTES UNLESS OTHERWISE SPECIFIED:
1.
.375
2X .125
.63
3X .1 MIN
.50
11.00 MAX
11.90 MIN
APPROVALS
DATE
DRAWN:
CHECKED:
MANUFACTURING:
THIRD ANGLE PROJECTION
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
DRAWING INTERPRETED PER ASME Y14.100-2000
DIMENSIONS INTERPRETED PER ASME Y14.5-1994
TOLERANCES:
DECIMAL: .X ±.1 .XX ±.02 .XXX ±.005
ANGULAR: X. ±1° .X ±.1°
SURFACE FINISH: 125
dextera
DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104
PART NAME:
GUIDE RAIL
REV:
1
MATERIAL:
ALUMINUM 6061-T6
DO NOT SCALE
FINISH:
BARE
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

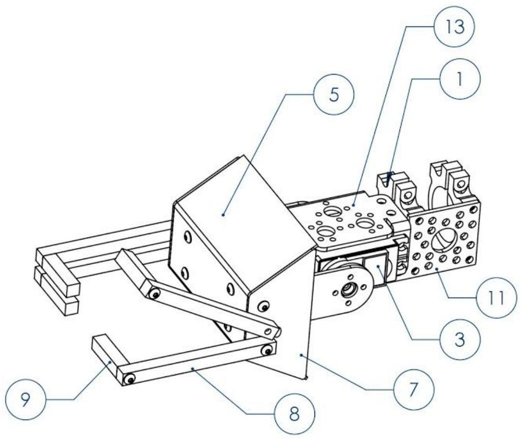
NOTES UNLESS OTHERWISE SPECIFIED:
1.



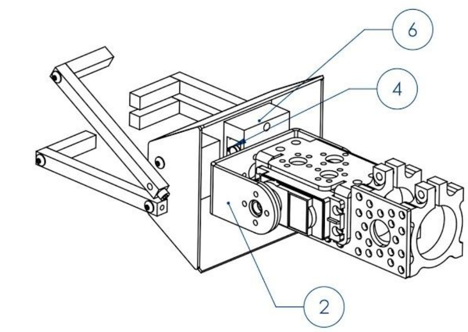
THIRD ANGLE PROJECTION	APPROVALS / DATE DRAWN: CHECKED: MANUFACTURING:	dextera	DEXTERA 220 S. 33RD STREET PHILADELPHIA, PA 19104
UNLESS OTHERWISE SPECIFIED:	TOLERANCES:	PART NAME: **VERTICAL SLIDE MIDDLE COVER**	
DIMENSIONS ARE IN INCHES	DECIMAL: .X ±.1 .XX ±.02 .XXX ±.005 ANGULAR: X. ±1° .X ±.1°	REV: **1**	
DRAWING INTERPRETED PER ASME Y14.100-2000		MATERIAL: **304 STAINLESS STEEL**	**DO NOT SCALE**
DIMENSIONS INTERPRETED PER ASME Y14.5-1994	SURFACE FINISH: 125	FINISH: **MIRROR**	SHEET 1 OF 1


dextera
DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104
THIRD ANGLE PROJECTION
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
DRAWING INTERPRETED PER ASME Y14.100-2000
DIMENSIONS INTERPRETED PER ASME Y14.5-1994
PART NAME:
WRIST ASSY
REV:
4
MATERIAL:
FINISH:
DO NOT SCALE
SHEET 1 OF 3


dextera
PART NAME:
WRIST ASSY
REV:
3
DO NOT SCALE
SHEET 2 OF 3
SOLIDWORKS Educational Product. For Instructional Use Only.



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	SERVO CITY 545620	SIDE TAPPED CLAMP MOUNT	2
2	ROBOT SHOP RB-Lyn-74	U BRACKET	1
3	SERVO CITY 31422S	HS-422 SERVO	1
4	POLOLU 3070	MICRO METAL GEARMOTOR 1000:1	2
5	FRONT MOTOR COVER		1
6	SERVO CITY 555120	MOTOR ENCLOSURE	2
7	SIDE MOTOR COVER		2
8	LONG GRIPPER ARM		4
9	SHORT GRIPPER ARM		4
11	SERVO CITY 575144	SERVO MOUNTING PLATE	1
13	ROBOT SHOP RB-Lyn-74	MULTI PURPOSE BRACKET	1

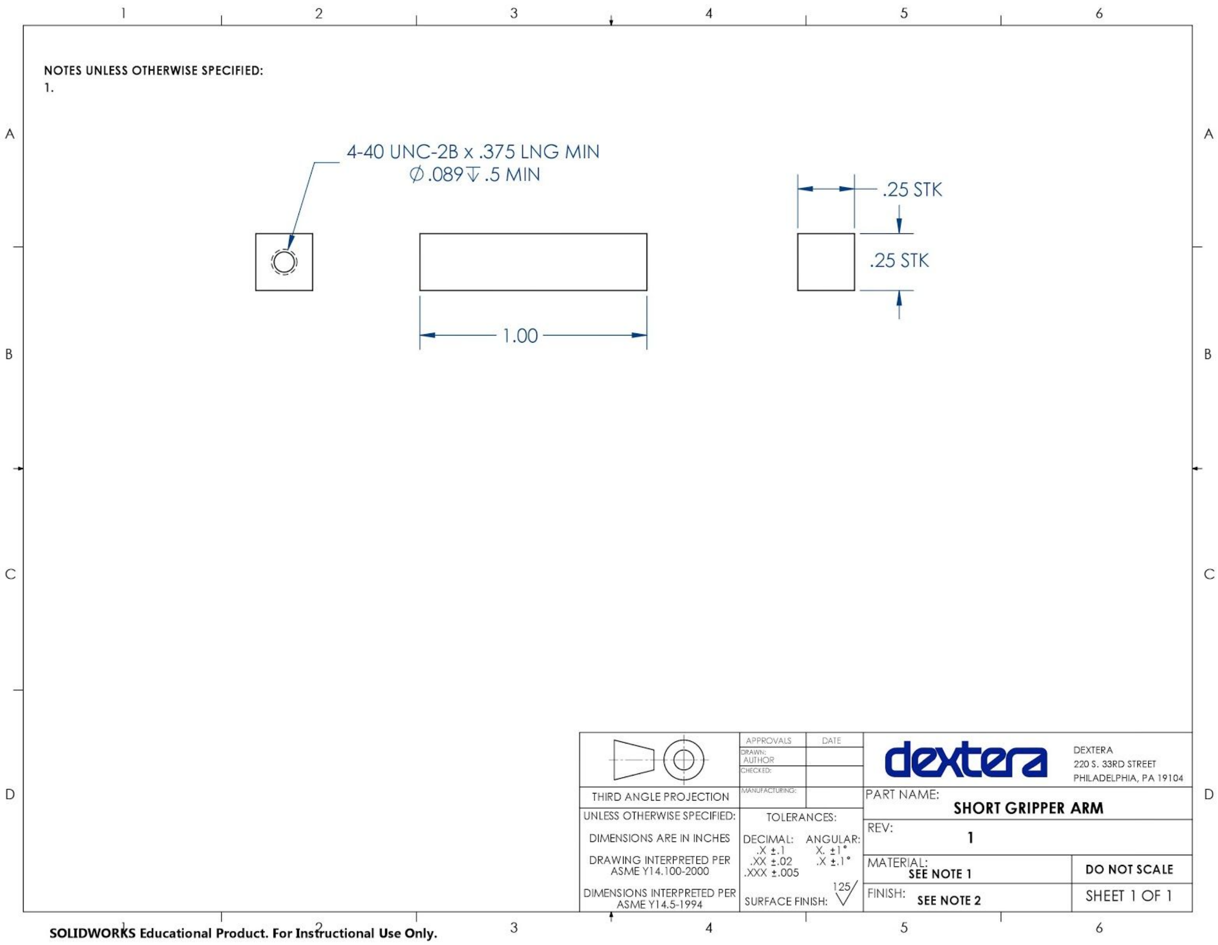


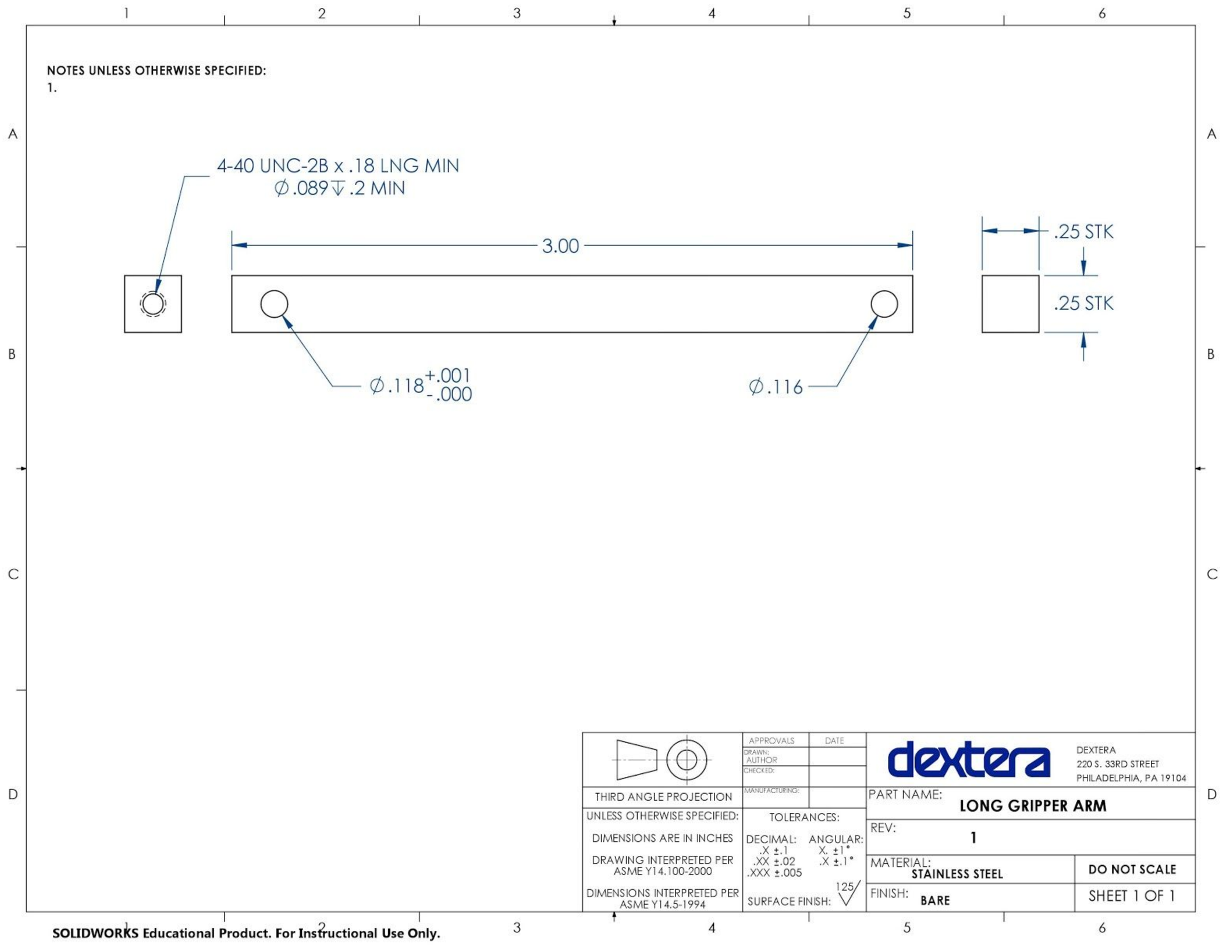
dextera

PART NAME: **WRIST ASSY**

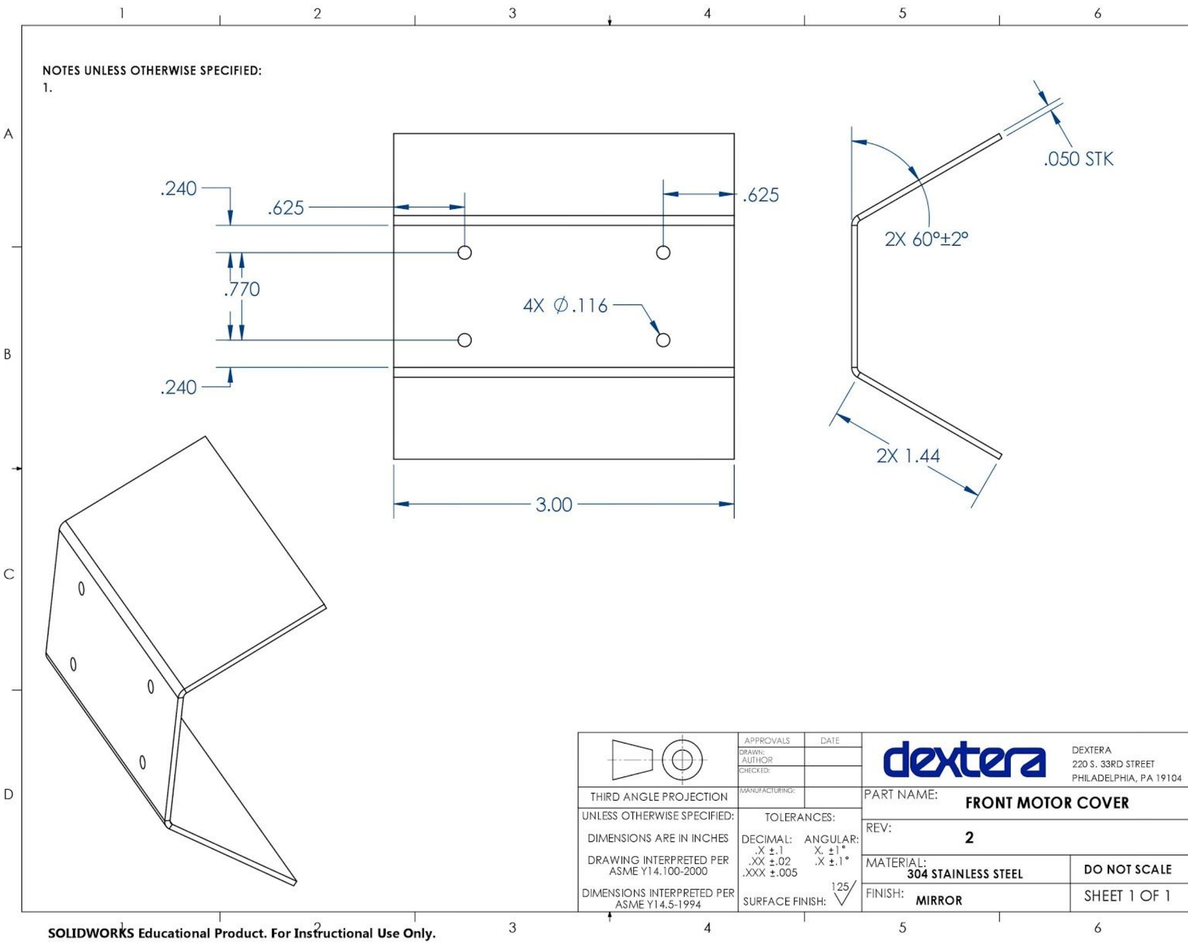
REV: **3**

DO NOT SCALE | SHEET 3 OF 3

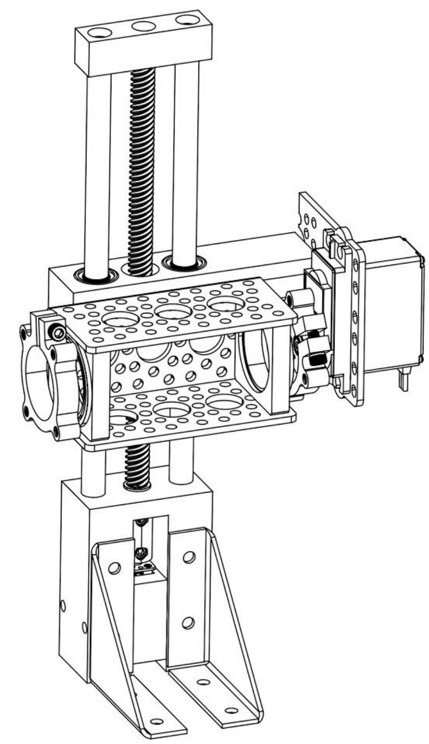

NOTES UNLESS OTHERWISE SPECIFIED:
1.
4-40 UNC-2B x .375 LNG MIN
∅ .089 ↧ .5 MIN
.25 STK
.25 STK
1.00
APPROVALS
DATE
DRAWN:
AUTHOR
CHECKED:
MANUFACTURING:
THIRD ANGLE PROJECTION
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
DRAWING INTERPRETED PER ASME Y14.100-2000
DIMENSIONS INTERPRETED PER ASME Y14.5-1994
TOLERANCES:
DECIMAL: .X ±.1 .XX ±.02 .XXX ±.005
ANGULAR: X. ±1° .X ±.1°
SURFACE FINISH: 125
dextera
DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104
PART NAME:
SHORT GRIPPER ARM
REV:
1
MATERIAL:
SEE NOTE 1
DO NOT SCALE
FINISH:
SEE NOTE 2
SHEET 1 OF 1



NOTES UNLESS OTHERWISE SPECIFIED:
1.
4-40 UNC-2B x .18 LNG MIN
∅.089↧.2 MIN
3.00
.25 STK
.25 STK
∅.118 +.001 -.000
∅.116
APPROVALS
DATE
DRAWN:
AUTHOR
CHECKED:
MANUFACTURING:
THIRD ANGLE PROJECTION
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
DRAWING INTERPRETED PER ASME Y14.100-2000
DIMENSIONS INTERPRETED PER ASME Y14.5-1994
TOLERANCES:
DECIMAL: .X ±.1 .XX ±.02 .XXX ±.005
ANGULAR: X. ±1° .X ±.1°
SURFACE FINISH: 125
dextera
DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104
PART NAME: LONG GRIPPER ARM
REV: 1
MATERIAL: STAINLESS STEEL
DO NOT SCALE
FINISH: BARE
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.



NOTES UNLESS OTHERWISE SPECIFIED:
1.
.240
.625
.625
.770
4X ∅.116
.240
3.00
.050 STK
2X 60°±2°
2X 1.44
APPROVALS
DATE
DRAWN:
AUTHOR
CHECKED:
MANUFACTURING:
THIRD ANGLE PROJECTION
dextera
DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104
PART NAME:
FRONT MOTOR COVER
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
DRAWING INTERPRETED PER ASME Y14.100-2000
DIMENSIONS INTERPRETED PER ASME Y14.5-1994
TOLERANCES:
DECIMAL: .X ±.1 .XX ±.02 .XXX ±.005
ANGULAR: X. ±1° .X ±.1°
SURFACE FINISH: 125
REV:
2
MATERIAL:
304 STAINLESS STEEL
DO NOT SCALE
FINISH:
MIRROR
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.



THIRD ANGLE PROJECTION

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES

DRAWING INTERPRETED PER ASME Y14.100-2000

DIMENSIONS INTERPRETED PER ASME Y14.5-1994

DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104

PART NAME: **VERTICAL TOWER ASSY**

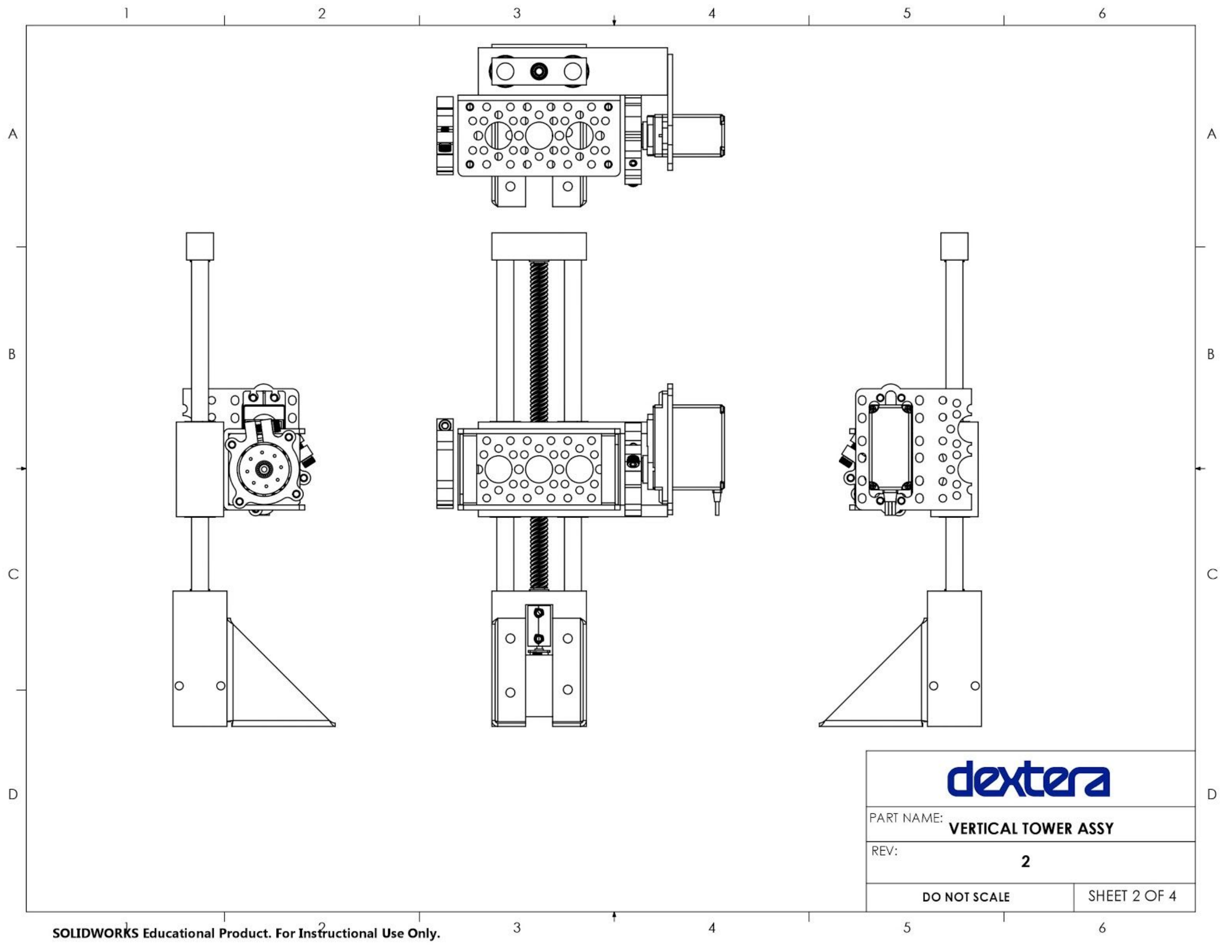
REV: **2**

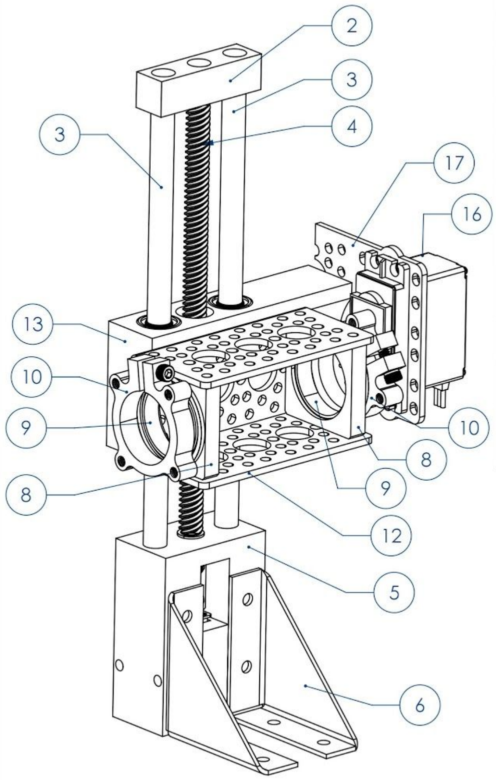
MATERIAL:

FINISH:

DO NOT SCALE

SHEET 1 OF 4


dextera
PART NAME: VERTICAL TOWER ASSY
REV: 2
DO NOT SCALE
SHEET 2 OF 4
SOLIDWORKS Educational Product. For Instructional Use Only.



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	SERVO CITY 555120	MOTOR ENCLOSURE	1
2	VERTICAL CAP BLOCK		1
3	SERVO CITY 634220	3/8" SHAFTING	2
4	SERVO CITY 3501-0804-0650	LEAD SCREW (MODIFIED)	1
5	BASE MOTOR MOUNTING		1
6	McMasterCarr 1088A31	MOUNTING BRACKET	2
8	SERVO CITY 535178	1.245" ID BEARING BLOCK	2
9	SERVO CITY 535051	1" ID BEARING	2
10	SERVO CITY 545352	CLAMPING HUB	2
12	SERVO CITY 585442	ALUMINUM CHANNEL	1
13	VERTICAL HORIZONTAL COUPLING BLOCK		1
14	SERVO CITY 545314	LEAD SCREW BARREL NUT	1
15	SERVO CITY 535226	LINEAR BEARING	2
16	ADAFRUT FS5103R	CONTINOUS ROTATION SERVO	1
17	SERVO CITY 575148	SERVO MOUNT PLATE	1
18	SERVO CITY 535037	1/8" ID BEARING	2
19	SERVO CITY 625158	SHAFT COUPLER	1
21	POLOLU 2217	MICRO METAL GEARMOTOR, 250:1	1

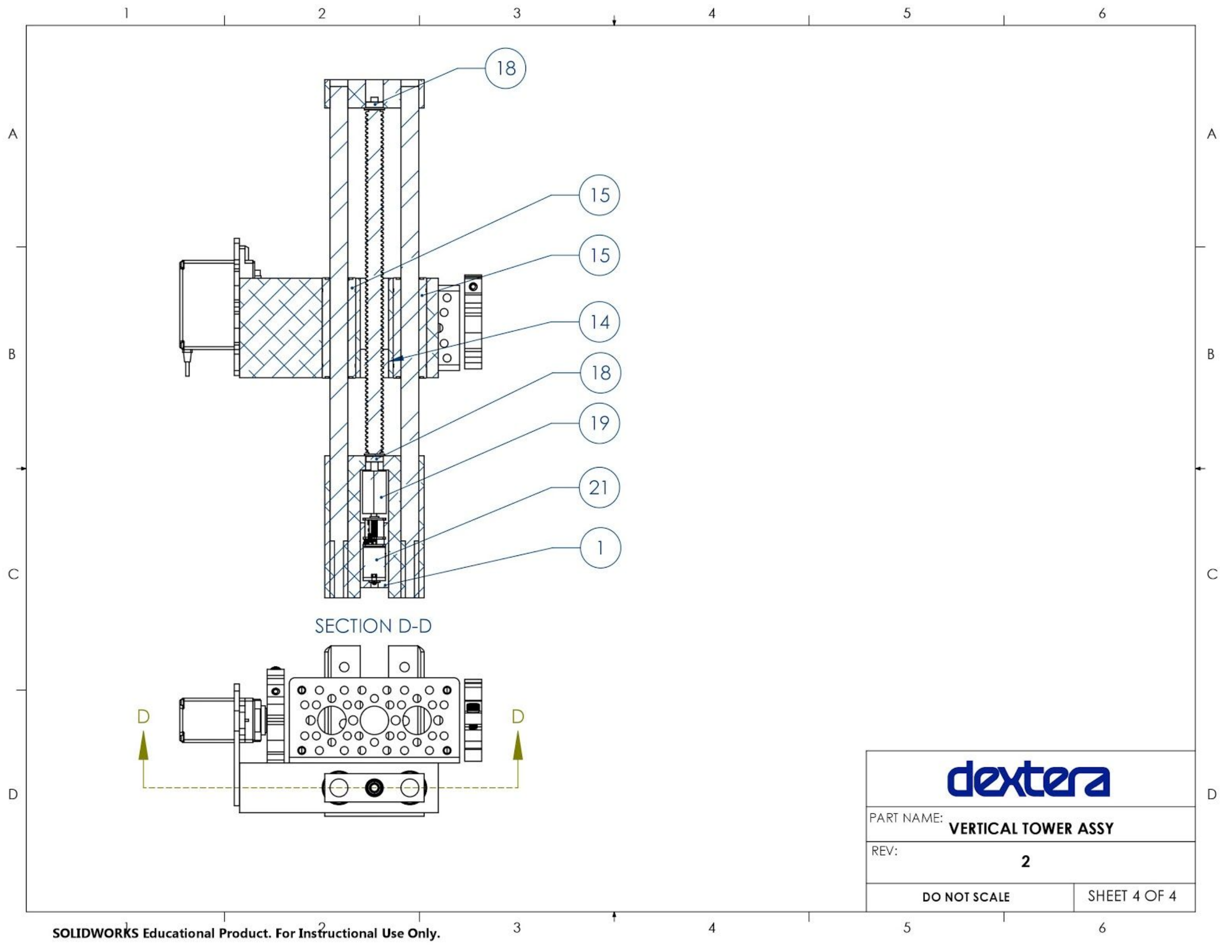
INTERNAL COMPONENTS SHOWN IN SECTION VIEW ON NEXT PAGE

dextera

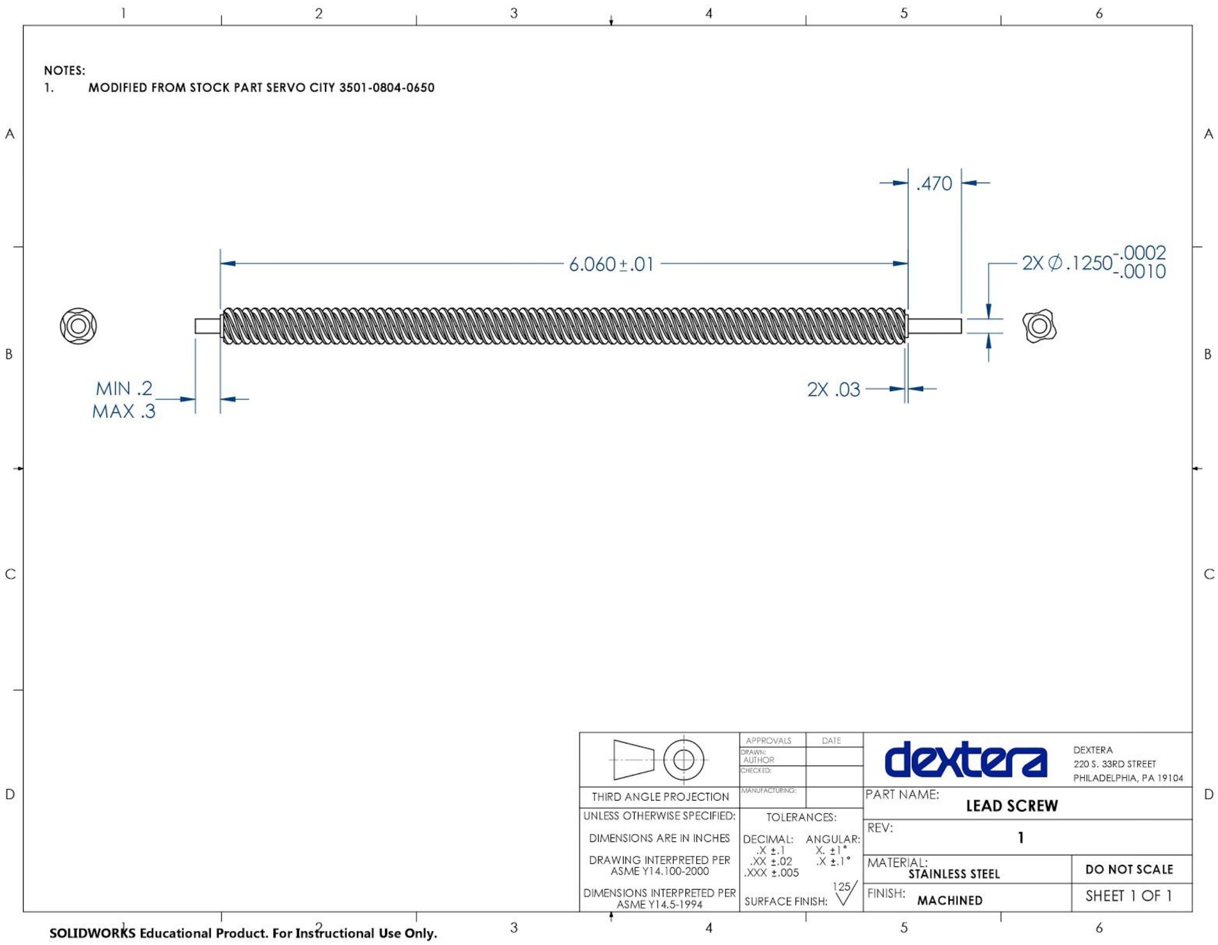
PART NAME: **VERTICAL TOWER ASSY**

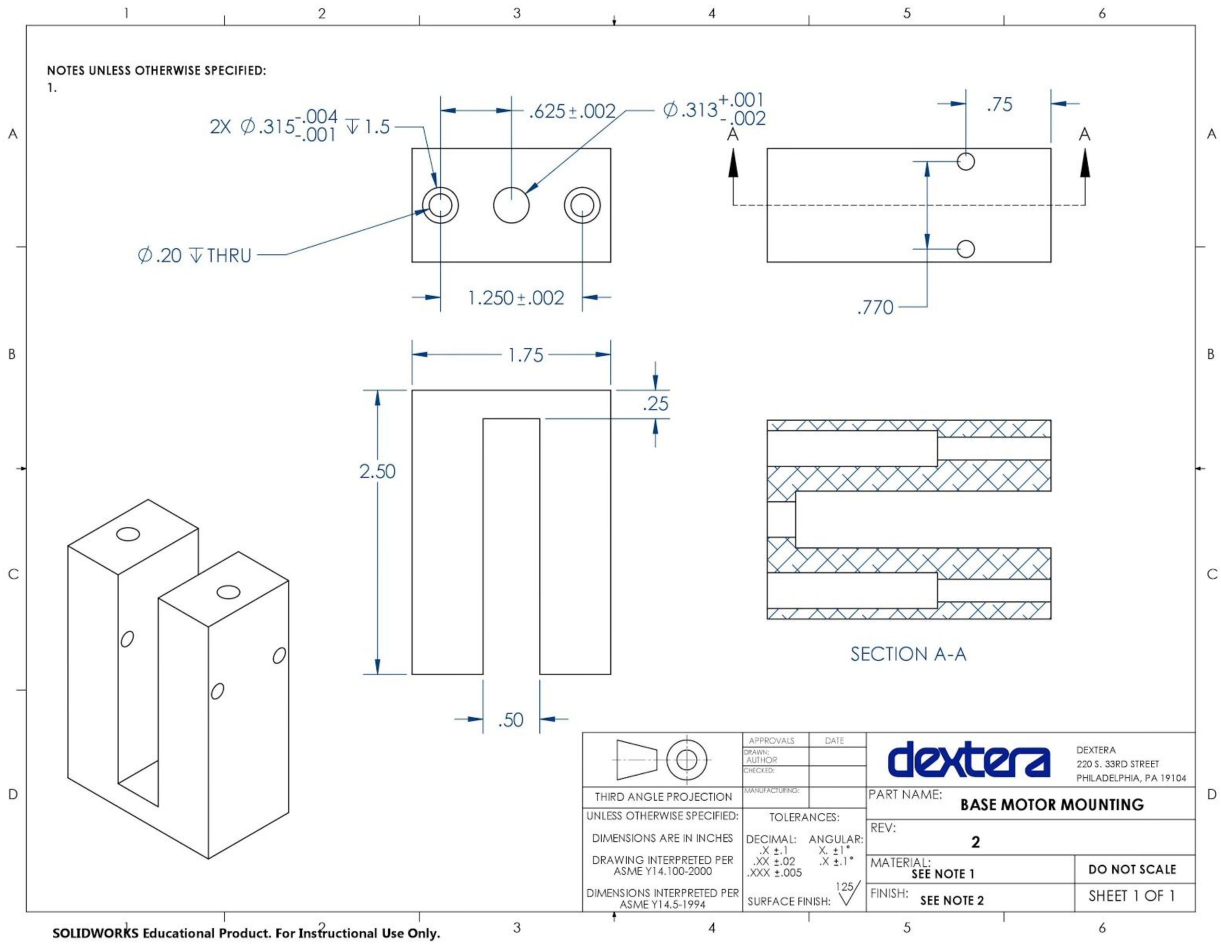
REV: **2**

DO NOT SCALE SHEET 3 OF 4

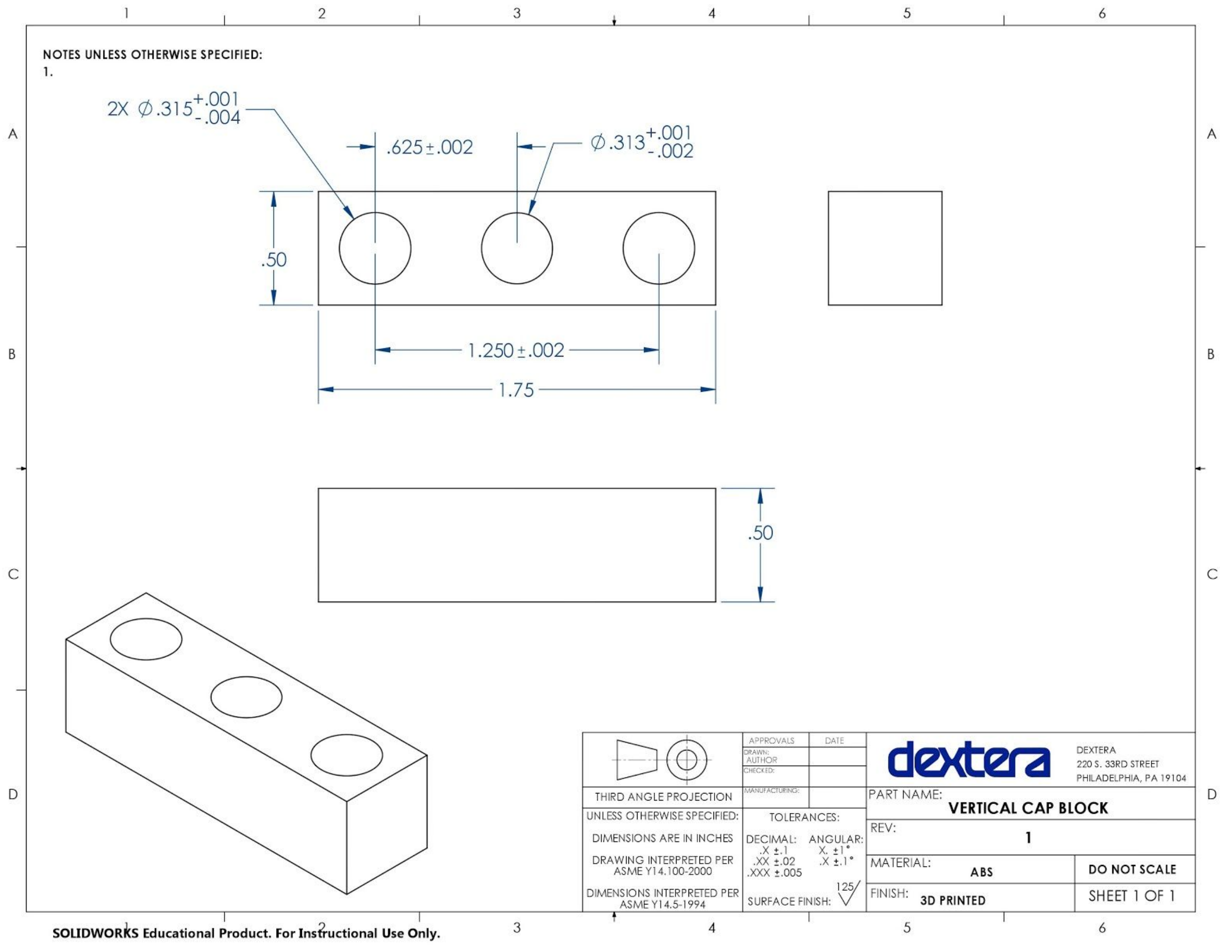


18
15
15
14
18
19
21
1
SECTION D-D
D
D
dextera
PART NAME: VERTICAL TOWER ASSY
REV: 2
DO NOT SCALE
SHEET 4 OF 4
SOLIDWORKS Educational Product. For Instructional Use Only.


NOTES:
1. MODIFIED FROM STOCK PART SERVO CITY 3501-0804-0650
.470
6.060±.01
2X ∅.1250 -.0002 -.0010
MIN .2
MAX .3
2X .03
THIRD ANGLE PROJECTION
APPROVALS
DATE
DRAWN:
AUTHOR
CHECKED:
MANUFACTURING:
dextera
DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104
PART NAME:
LEAD SCREW
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
DRAWING INTERPRETED PER ASME Y14.100-2000
DIMENSIONS INTERPRETED PER ASME Y14.5-1994
TOLERANCES:
DECIMAL: .X ±.1 .XX ±.02 .XXX ±.005
ANGULAR: X. ±1° .X ±.1°
SURFACE FINISH: 125
REV:
1
MATERIAL:
STAINLESS STEEL
DO NOT SCALE
FINISH:
MACHINED
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.



NOTES UNLESS OTHERWISE SPECIFIED:
1.
2X Ø.315 -.004 -.001 ↧1.5
.625±.002
Ø.313 +.001 -.002
Ø.20 ↧THRU
1.250±.002
.75
A
A
.770
1.75
.25
2.50
.50
SECTION A-A
THIRD ANGLE PROJECTION
APPROVALS
DATE
DRAWN:
AUTHOR
CHECKED:
MANUFACTURING:
dextera
DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104
PART NAME:
BASE MOTOR MOUNTING
UNLESS OTHERWISE SPECIFIED:
TOLERANCES:
DIMENSIONS ARE IN INCHES
DECIMAL: .X ±.1 .XX ±.02 .XXX ±.005
ANGULAR: X. ±1° .X ±.1°
REV:
2
DRAWING INTERPRETED PER ASME Y14.100-2000
MATERIAL: SEE NOTE 1
DO NOT SCALE
DIMENSIONS INTERPRETED PER ASME Y14.5-1994
SURFACE FINISH: 125
FINISH: SEE NOTE 2
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


NOTES UNLESS OTHERWISE SPECIFIED:
1.
2X ∅.315 +.001 -.004
.625±.002
∅.313 +.001 -.002
.50
1.250±.002
1.75
.50
THIRD ANGLE PROJECTION
APPROVALS
DATE
DRAWN: AUTHOR
CHECKED:
MANUFACTURING:
dextera
DEXTERA
220 S. 33RD STREET
PHILADELPHIA, PA 19104
PART NAME:
VERTICAL CAP BLOCK
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
DRAWING INTERPRETED PER ASME Y14.100-2000
DIMENSIONS INTERPRETED PER ASME Y14.5-1994
TOLERANCES:
DECIMAL: .X ±.1 .XX ±.02 .XXX ±.005
ANGULAR: X. ±1° .X ±.1°
SURFACE FINISH: 125
REV: 1
MATERIAL: ABS
DO NOT SCALE
FINISH: 3D PRINTED
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

Appendix A.3 Bill of Materials

Description	Vendor	Part #	Qty	Cost	Total Cost	Assy	Stock Material	MFG Equipment
Vertical Slide Middle Cover	In House, PML	D001	1			Casing Assy	.045 304 Stainless Steel Sheet	Sheet Metal Shear, Mill, Endmill
Guide Rail	In House, PML	D002	2			Casing Assy	0.5"x0.63" Aluminum 6061-T6 Bar	Mill, Slitting Saw
Sliding Cover	In House, PML	D003	2			Casing Assy	.045 304 Stainless Steel Sheet	Sheet Metal Shear
Fixed Cover	In House, PML	D004	2			Casing Assy	.045 304 Stainless Steel Sheet	Sheet Metal Shear
Lid	In House, RPL	D005	2			Casing Assy	Acrylic	Laser Cut
External Case Body	In House, PML	D006	1			Casing Assy	.045 304 Stainless Steel Sheet	Sheet Metal Shear, Brake
Raspberry Pi 3 - Model B+	Adafruit	3775	1	35	35	Electronics		
Plantronics Savi W430 Wireless Headset - Silver/Black	Amazon	NA	1	159.34	159.34	Electronics		
1" Aluminum Tubing, 10" Length	Servo City	635118	1	3.49	3.49	Top Assembly		
Standard Servo Plate D	Servo City	575148	1	6.99	6.99	Vertical Tower		
3.00” Aluminum Channel	Servo City	585442	1	3.99	3.99	Vertical Tower		
1" Bore Side Tapped Pillow Block	Servo City	535178	2	9.99	19.98	Vertical Tower		
1" Bore, Face Tapped Clamping Hub, 1.50" Pattern	Servo City	545352	2	5.99	11.98	Vertical Tower		
8mm Stainless Steel Precision Shafting	Servo City	634314	2	4.69	9.38	Vertical Tower		

8mm Lead Screw, 650mm Long	Servo City, Modified In House	3501-08 04-0650	1	13.99	13.99	Vertical Tower		Turned
8mm Lead Screw Barrel Nut	Servo City	545314	1	4.99	4.99	Vertical Tower		
8mm Linear Ball Bearings (2 pack)	Servo City	535226	1	6.99	6.99	Vertical Tower		
1/8" ID x 5/16" OD Flanged Ball Bearing (2 pack)	Servo City	535037	2	1.99	3.98	Vertical Tower		
Set Screw Shaft Coupler, 1/8 to 3mm	Servo City	625158	1	4.99	4.99	Vertical Tower		
Magnetic Encoder Pair Kit for Micro Metal Gearmotors, 12 CPR	Pololu	3081	1	8.95	8.95	Vertical Tower		
High Speed Continuous Rotation Servo	Adafruit	3614	1	27.99	27.99	Vertical Tower		
Vertical Cap Block	In House, Tazbot	D011	1			Vertical Tower	ABS	3D Printing
Base Motor Mounting	In House, Tazbot	D012	1			Vertical Tower	ABS	3D Printing
Vertical Horizontal Coupling Block	In House, Tazbot	D013	1			Vertical Tower	ABS	3D Printing
1" Bore Side Tapped Clamping Mount	Servo City	545620	2	5.99	11.98	Wrist		
1000:1 Micro Metal Gearmotor HPCB 6V with Extended Motor Shaft	Pololu	3080	2	25.95	51.9	Wrist		
Front Motor Cover	In House, PML	D007	1			Wrist	.045 304 Stainless Steel Sheet	Sheet Metal Shear, Brake
Side Motor Cover	In House, PML	D008	2			Wrist	.045 304 Stainless Steel Sheet	Sheet Metal Shear, Drill
Long Gripper Arm	In House, PML	D009	4			Wrist	.25x.25 Aluminum 6061-T6 Bar	CNC, Drill, Tap
Short Gripper Arm	In House, PML	D010	4			Wrist	.25x.25 Aluminum 6061-T6 Bar	CNC, Drill, Tap

250:1 Micro Metal Gearmotor HP 6V with Extended Motor Shaft	Pololu	2217	1	16.95	16.95	Wrist Assembly		
Standard Servo Plate C	Servo City	575144	1	5.99	5.99	Wrist Assembly		
Micro Gear Motor Enclosure	Servo City	555120	3	0.99	2.97	Wrist Assy x2, Vertical Tower x1		
	Total Cost (Purchased Parts for Prototype)				**411.82**			

Appendix A.3: Order Summary

Qty	Part #	Description	Cost	Total Cost
2	545620	1" Bore Side Tapped Clamping Mount	5.99	11.98
2	535178	1" Bore Side Tapped Pillow Block	9.99	19.98
2	545352	1" Bore, Face Tapped Clamping Hub, 1.50" Pattern	5.99	11.98
1	585442	3.00” Aluminum Channel	3.99	3.99
1	635118	1" Aluminum Tubing, 10" Length	3.49	3.49
2	575148	Standard Servo Plate D	6.99	13.98
2	3080	1000:1 Micro Metal Gearmotor HPCB 6V with Extended Motor Shaft	25.96	51.92
1	RB-Lyn-74	Lynxmotion Pan and Tilt Kit/Aluminium	29.93	29.93
4	634314	8mm Stainless Steel Precision Shafting	4.69	18.76
1	3501-0804-0650	8mm Lead Screw, 650mm Long	13.99	13.99
2	545314	8mm Lead Screw Barrel Nut	4.99	9.98
2	535226	8mm Linear Ball Bearings (2 pack)	6.99	13.98
2	555120	Micro Gear Motor Enclosure	0.99	1.98
2	535037	1/8" ID x 5/16" OD Flanged Ball Bearing (2 pack)	1.99	3.98

1	625158	Set Screw Shaft Coupler, 1/8 to 3mm	4.99	4.99
3	3081	Magnetic Encoder Pair Kit for Micro Metal Gearmotors, 12 CPR, 2.7-18V (HPCB compatible)	8.95	26.85
1	3775	Raspberry Pi 3 - Model B+ 1.4 GHz Cortex- A53 with 1GB RAM	35	35
1	3614	Feedback 360 Degree - High Speed Continuous Rotation Servo	27.99	27.99
1	22606	STAINLESS #8 SHEET 304 ANNEALED - Thickness - 0.045" - 12" x 36"	59.7	59.7
1	588	COLD DRAWN STAINLESS RECTANGLE 303 ANNEALED - two foot length	15.48	15.48
2	3081	Magnetic Encoder Pair Kit for Micro Metal Gearmotors, 12 CPR, 2.7-18V (HPCB compatible)	8.95	17.9
1	2217	250:1 Micro Metal Gearmotor HP 6V with Extended Motor Shaft	16.95	16.95
1	575144	Standard Servo Plate C	5.99	5.99

1	N/A	Plantronics Savi W430 Wireless Headset - Silver/Black	$159.34	$159.34
	Total Budget Used			580.11

Appendix B.1 MATLAB Scripts for Robotic Kinematics Model

Forward Kinematics:

```
% Forward Kinematics

function [jointPositions, T0e] = FK(q)

% Arm constants in mm
d0 = 20; % base height (table to center of joint 1)
l = 125; % length of horizontal link
lg = 75; % length of grippers

%Frame 1 w.r.t Frame 0
T1 = [1  0  0  0;
      0  1  0  0;
      0  0  1  d0;
      0  0  0  1];

%Frame 2 w.r.t Frame 1
T2 = [1  0  0   0;
      0  0 -1   0;
      0  1  0  q(1);
      0  0  0   1];

%Frame 3 w.r.t Frame 2
T3 = [cos(q(2)+pi/2)   0  sin(q(2)+pi/2)  0;
      sin(q(2)+pi/2)   0 -cos(q(2)+pi/2)  0;
            0          1       0          l;
            0          0       0          1];

%Frame 4 w.r.t Frame 3
T4 = [cos(q(3)+pi/2)  -sin(q(3)+pi/2)  0  lg*cos(q(3)+pi/2);
      sin(q(3)+pi/2)   cos(q(3)+pi/2)  0  lg*sin(q(3)+pi/2);
         0                  0          1           0;
         0                  0          0           1];

T01 = T1;
T02 = T1*T2;
T03 = T1*T2*T3;
T04 = T1*T2*T3*T4;

%Position of Base
X(1,:) = [0 0 0];

%Position of First Joint (Prismatic)
X(2,:) = (T01(1:3,4))';

%Position of Second Joint (Wrist Revolute 1)
X(3,:) = (T02(1:3,4))';

%Position of Third Joint (Wrist Revolute 2)
X(4,:) = (T03(1:3,4))';

%Position of Grippers
X(5,:) = (T04(1:3,4))';

%Outputs the 5x3 of the locations of each joint in the Base Frame
jointPositions = X;

T0e = T1*T2*T3*T4;

end
```

Inverse Kinematics:

```
% Inverse Kinematics

function [q, is_possible] = IK(o)
% Assume pose is reachable, check along the way
is_possible = 1;
q = [0,0,0];

% Arm constants in mm
d0 = 60; % base height (table to center of joint 1)
l = 125;
lg = 75;

% Grippers frame origin
o_x = o(1);
o_y = o(2);
o_z = o(3);

% q3 in terms of end effector origin location
q3 = asin((o_y+l)/lg) - pi/2;

if(mod(q3,pi) == 0)
  q2 = 0; % infinite solutions when q3 is mult of pi, set to 0
else
  % q2 in terms of end effector origin location
  q2 = acos(o_x/(lg*cos(q3+pi/2))) - pi/2;
end

% q1 in terms of end effector origin location
q1 = o_z - d0 - lg*sin(q2+pi/2)*cos(q3+pi/2);

q = [q1 q2 q3];

end
```

Calculate Jacobian:

```
% Jacobian calculations

syms d1 q2 q3 d0 l lg real

% FK Transforms
%Frame 1 w.r.t Frame 0
T1 = [1  0  0  0;
      0  1  0  0;
      0  0  1  d0;
      0  0  0  1];

%Frame 2 w.r.t Frame 1
T2 = [1  0  0   0;
      0  0 -1   0;
      0  1  0  d1;
      0  0  0   1];

%Frame 3 w.r.t Frame 2
T3 = [cos(q2+pi/2)   0  sin(q2+pi/2)  0;
      sin(q2+pi/2)   0 -cos(q2+pi/2)  0;
            0        -1       0          l;
            0         0       0          1];

%Frame 4 w.r.t Frame 3
T4 = [cos(q3+pi/2)  -sin(q3+pi/2)  0  lg*cos(q3+pi/2);
      sin(q3+pi/2)   cos(q3+pi/2)  0  lg*sin(q3+pi/2);
         0                 0          1           0;
         0                 0          0           1];

T0 = zeros(4,4,'sym');
T0(:,:,1) = T1;
T0(:,:,2) = T1*T2;
T0(:,:,3) = T1*T2*T3;
T0(:,:,4) = T1*T2*T3*T4;
```


```
T0 = zeros(4,4,'sym');
T0(:,:,1) = T1;
T0(:,:,2) = T1*T2;
T0(:,:,3) = T1*T2*T3;
T0(:,:,4) = T1*T2*T3*T4;
```


```
% ANGULAR VELOCITY JACOBIAN
% Approach: grab z vectors
Jw = zeros(3,3,'sym');
for i = 1:3
    Jw(:,i) = T0(1:3,3,i+1);
end


% LINEAR VELOCITY JACOBIAN
% Approach: cross products
Jv = zeros(3,3,'sym');
for i = 1:3
    Jv(:,i) = cross(T0(1:3,3,i+1), T0(1:3,4,end)-T0(1:3,4,i+1));
end

% Compose Jacobian Matrix and calc end effector velocities
J_CS = [Jv;Jw];
```

Velocity Forward Kinematics

```
%Velocity_FK

function e_vel = Velocity_FK(q, qdot)
% Input: q - 1 x 3 vector of joint inputs [d1,q2,q3]
%        qdot - 1 x 3 vector of joint velocities [d1dot,q2dot,q3dot]

% Outputs:  e_vel - 6 x 1 vector of end effector velocities, where
%                   e_vel(1:3) are the linear velocity
%                   e_vel(4:6) are the angular velocity

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d1 = q(1);
q2 = q(2);
q3 = q(3);
lg = 75;

% Jacobian from calcJacobian output
J(1,:) = [0, -lg*cos(q3 + pi/2)*sin(q2 + pi/2), ...
    -lg*cos(q2 + pi/2)*sin(q3 + pi/2)];
J(2,:) = [0, 0, -lg*cos(q3 + pi/2)];
J(3,:) = [1, lg*cos(q2 + pi/2)*cos(q3 + pi/2), ...
    -lg*sin(q2 + pi/2)*sin(q3 + pi/2)];
J(4,:) = [0,  0, sin(q2 + pi/2)];
J(5,:) = [0, -1, 0];
J(6,:) = [0,  0, -cos(q2 + pi/2)];

% full-body velocity  = Jacobian * qdot
e_vel = J * transpose(qdot);

end
```

Velocity Inverse Kinematics

```
%Velocity_IK

function qdot = Velocity_IK(q, e_vel)
% Input: q - 1 x 3 vector of joint inputs [d1,q2,q3]
%        e_vel - 6 x 1 vector of end effector velocities, where
%                   e_vel(1:3) are the linear velocity
%                   e_vel(4:6) are the angular velocity

% Output: qdot - 1 x 3 vector of joint velocities [d1dot,q2dot,q3dot]


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d1 = q(1);
q2 = q(2);
q3 = q(3);

J = calcJacobian(d1,q2,q3);

% get qdot, (psuedo-inverse jacobian * vel of end effector)
qdot = (J\transpose(e_vel))';

end
```

Initialization function:

```
% Initialization function

function dexteraInitialize()
global arm q

arm.firstFrame = true;
arm.showGrippers = false;

% Home pose
q = [60,0,0];
dexteraSim(q);

end
```

Dextera Simulation:

```
% Simulation function

function dexteraSim(q)

frameSize = 50; % end effector frame size
axisWidth = 2; % width of axes
C = [0.28 0 0.8]; % link color
C_joint = [0.64 0.64 0.64]; % joint color

global arm

% get joint positions and origin of each joint frame
[jPos, T0e] = FK(q);

o0 = [jPos(1,1) jPos(1,2) jPos(1,3)];
o1 = [jPos(2,1) jPos(2,2) jPos(2,3)];
o2 = [jPos(3,1) jPos(3,2) jPos(3,3)];
o3 = [jPos(4,1) jPos(4,2) jPos(4,3)];
o4 = [jPos(5,1) jPos(5,2) jPos(5,3)];

if(arm.firstFrame) %We need to create the plots
    clf; %Clear any previous plot

    % configure plot
    xlabel('X (mm)')
    ylabel('Y (mm)')
    zlabel('Z (mm)')
    set(gca,'xtick',-1000:100:1000, 'ytick',-1000:100:1000,...
        'ztick',-1000:100:1000)
    grid on
    view(80,20)

    axis equal vis3d
    xlim([-200 100]);
    ylim([-400 100]);
    zlim([-100 200]);

% create cylinders for each link, grippers
arm.link1 = Cylinder(o0,o1,8,20,C,1,0);
arm.link2 = Cylinder(o1,o2,8,20,C,0,0);
arm.link3 = Cylinder(o2,o3,8,20,C,0,0);
arm.gripper1 = Cylinder(o3+[3 3 3],o4+[3 3 3],3,20,'r',0,0);
arm.gripper2 = Cylinder(o3-[3 3 3],o4-[3 3 3],3,20,'r',0,0);

hold on;
% draw each revolute joint
[x,y,z] = sphere(100);
arm.joint2 = surf((x*10)+o2(1),(y*10)+o2(2), (z*10)+o2(3),...
    'FaceColor',C_joint,'EdgeColor','none');
arm.joint3 = surf((x*10)+o3(1),(y*10)+o3(2), (z*10)+o3(3),...
    'FaceColor',C_joint,'EdgeColor', 'none');

% draw prismatic joint
L = 18; % prismatic joint size (length of an edge)

X_cube = [0 0 0 0 0 1; 1 0 1 1 1 1; 1 0 1 1 1 1; 0 0 0 0 0 1];
Y_cube = [0 0 0 0 1 0; 0 1 0 0 1 1; 0 1 1 1 1 1; 0 0 1 1 1 0];
Z_cube = [0 0 1 0 0 0; 0 0 1 0 0 0; 1 1 1 0 1 1; 1 1 1 0 1 1];

X_cube = L*(X_cube-0.5) + o1(1);
Y_cube = L*(Y_cube-0.5) + o1(2);
Z_cube = L*(Z_cube-0.5) + o1(3);

% draw prismatic joint
arm.prismatic = patch(X_cube,Y_cube,Z_cube,C_joint);
```

```
    % draw end effector frame axes
    EExAxis = T0e(1:3,1)';
    EEyAxis = T0e(1:3,2)';
    EEzAxis = T0e(1:3,3)';
    EExAxis = frameSize*EExAxis;
    EEyAxis = frameSize*EEyAxis;
    EEzAxis = frameSize*EEzAxis;
    arm.EEx = line([T0e(1,4),T0e(1,4)+EExAxis(1)],...
        [T0e(2,4),T0e(2,4)+EExAxis(2)],[T0e(3,4),T0e(3,4)+EExAxis(3)],...
        'Color','r','LineWidth',axisWidth);
    arm.EEy = line([T0e(1,4),T0e(1,4)+EEyAxis(1)],...
        [T0e(2,4),T0e(2,4)+EEyAxis(2)],[T0e(3,4),T0e(3,4)+EEyAxis(3)],...
        'Color','g','LineWidth',axisWidth);
    arm.EEz = line([T0e(1,4),T0e(1,4)+EEzAxis(1)],...
        [T0e(2,4),T0e(2,4)+EEzAxis(2)],[T0e(3,4),T0e(3,4)+EEzAxis(3)],...
        'Color','b','LineWidth',axisWidth);

    % generate lighting
    l = light('Position',[0.4 0.4 0.9],'Style','infinite');
    lighting gouraud
    material shiny

    arm.firstFrame = false;
else
    % reset everything besides the plot itself
    delete(arm.link1);
    delete(arm.link2);
    delete(arm.link3);
    delete(arm.gripper1);
    delete(arm.gripper2);
    delete(arm.prismatic);
    delete(arm.joint2);
    delete(arm.joint3);


  % redraw links and grippers
  arm.link1 = Cylinder(o0,o1,8,20,C,1,0);
  arm.link2 = Cylinder(o1,o2,8,20,C,0,0);
  arm.link3 = Cylinder(o2,o3,8,20,C,0,0);
  arm.gripper1 = Cylinder(o3+[3 3 3],o4+[3 3 3],3,20,'r',0,0);
  arm.gripper2 = Cylinder(o3-[3 3 3],o4-[3 3 3],3,20,'r',0,0);

  hold on;

  % redraw revolute joints
  [x,y,z] = sphere(100);
  arm.joint2 = surf((x*10)+o2(1),(y*10)+o2(2), (z*10)+o2(3),...
      'FaceColor',C_joint,'EdgeColor','none');
  arm.joint3 = surf((x*10)+o3(1),(y*10)+o3(2), (z*10)+o3(3),...
      'FaceColor',C_joint,'EdgeColor', 'none');

  % redraw prismatic joint
  L = 18; % prismatic joint size (length of an edge)
  X_cube = [0 0 0 0 0 1; 1 0 1 1 1 1; 1 0 1 1 1 1; 0 0 0 0 0 1];
  Y_cube = [0 0 0 0 1 0; 0 1 0 0 1 1; 0 1 1 1 1 1; 0 0 1 1 1 0];
  Z_cube = [0 0 1 0 0 0; 0 0 1 0 0 0; 1 1 1 0 1 1; 1 1 1 0 1 1];

  X_cube = L*(X_cube-0.5) + o1(1);
  Y_cube = L*(Y_cube-0.5) + o1(2);
  Z_cube = L*(Z_cube-0.5) + o1(3);

  % draw prismatic joint
  arm.prismatic = patch(X_cube,Y_cube,Z_cube,C_joint);

  % reset lighting
  l = light('Position',[0.4 0.4 0.9],'Style','infinite');
  lighting gouraud
  material shiny
```

```
    % set data for end effector axes
    EExAxis = T0e(1:3,1)';
    EEyAxis = T0e(1:3,2)';
    EEzAxis = T0e(1:3,3)';
    EExAxis = frameSize*EExAxis;
    EEyAxis = frameSize*EEyAxis;
    EEzAxis = frameSize*EEzAxis;
    set(arm.EEx,'xdata',[T0e(1,4),T0e(1,4)+EExAxis(1)],'ydata',...
        [T0e(2,4),T0e(2,4)+EExAxis(2)],'zdata',[T0e(3,4),...
        T0e(3,4)+EExAxis(3)]);
    set(arm.EEy,'xdata',[T0e(1,4),T0e(1,4)+EEyAxis(1)],'ydata',...
        [T0e(2,4),T0e(2,4)+EEyAxis(2)],'zdata',[T0e(3,4),...
        T0e(3,4)+EEyAxis(3)]);
    set(arm.EEz,'xdata',[T0e(1,4),T0e(1,4)+EEzAxis(1)],'ydata',...
        [T0e(2,4),T0e(2,4)+EEzAxis(2)],'zdata',[T0e(3,4),...
        T0e(3,4)+EEzAxis(3)]);

end

drawnow
end
```

Velocity simulation:

```
% Velocity Simulatioon

function trajectory = DexVelocitySim(q)

dt = 0.1; % time step
[pos, ~] = FK(q); % get start configuration
e_pos = pos(5,:);

% initialize x, y, z trajectory plots
x_pos = [];
x_pos(end+1) = e_pos(1);

y_pos = [];
y_pos(end+1) = e_pos(2);

z_pos = [];
z_pos(end+1) = e_pos(1);

trajectory = [];
trajectory(end+1,:) = e_pos;

% constant qdot
qdot = [20 20 0];

% draw first figure
dexteraInitialize();

for t = 0:dt:1
    e_vel = Velocity_FK(q, qdot); % find velocity FK

    dexteraSim(q); % simulate current config
    hold on;
    plot3(x_pos,y_pos,z_pos,'--','LineWidth',2); % plot trajectory
    drawnow;


    % calc next pose from velocity FK
    e_pos = e_pos + transpose(e_vel(1:3)).*dt;
    x_pos(end+1) = e_pos(1);
    y_pos(end+1) = e_pos(2);
    z_pos(end+1) = e_pos(3);

    q = q + qdot.*dt; % update configuration
    trajectory(end+1,:) = e_pos; % add end effector location to trajectory
end
```

Cylinder:

```
% THIS IS AN OPEN SOURCE FUNCTION TAKEN FROM THE FOLLOWING URL:
% https://www.mathworks.com/matlabcentral/fileexchange/13995-cylinder

function [Cylinder, EndPlate1, EndPlate2] = Cylinder(X1,X2,r,n,...
                                            cyl_color,closed,lines)
%
% This function constructs a cylinder connecting two center points
%
% Usage :
% [Cylinder EndPlate1 EndPlate2] = Cylinder(X1+20,X2,r,n,'r',closed,lines)
%
% Cylinder-------Handle of the cylinder
% EndPlate1------Handle of the Starting End plate
% EndPlate2------Handle of the Ending End plate
% X1 and X2 are the 3x1 vectors of the two points
% r is the radius of the cylinder
% n is the no. of elements on the cylinder circumference (more--> refined)
% cyl_color is the color definition like 'r','b',[0.52 0.52 0.52]
% closed=1 for closed cylinder or 0 for hollow open cylinder
% lines=1 for displaying the line segments on the cylinder 0 for only
% surface
%
% Typical Inputs
% X1=[10 10 10];
% X2=[35 20 40];
% r=1;
% n=20;
% cyl_color='b';
% closed=1;

% NOTE: There is a MATLAB function "cylinder" to revolve a curve about an
% axis. This "Cylinder" provides more customization like direction and etc
% Calculating the length of the cylinder
length_cyl=norm(X2-X1);
% Creating a circle in the YZ plane
t=linspace(0,2*pi,n)';
x2=r*cos(t);
x3=r*sin(t);
% Creating the points in the X-Direction
x1=[0 length_cyl];
% Creating (Extruding) the cylinder points in the X-Directions
xx1=repmat(x1,length(x2),1);
xx2=repmat(x2,1,2);
xx3=repmat(x3,1,2);
% Drawing two filled cirlces to close the cylinder
if closed==1
    hold on
    EndPlate1=fill3(xx1(:,1),xx2(:,1),xx3(:,1),'r');
    EndPlate2=fill3(xx1(:,2),xx2(:,2),xx3(:,2),'r');
end
% Plotting the cylinder along the X-Direction with required length starting
% from Origin
Cylinder=mesh(xx1,xx2,xx3);
% Defining Unit vector along the X-direction
unit_Vx=[1 0 0];
% Calulating the angle between the x direction and the required direction
% of cylinder through dot product
angle_X1X2=acos(dot(unit_Vx,(X2-X1))/(norm(unit_Vx)*norm(X2-X1)))*180/pi;
% Finding the axis of rotation (single rotation) to roate the cylinder in
% X-direction to the required arbitrary direction through cross product
axis_rot=cross([1 0 0],(X2-X1) );
```

```
% Rotating the plotted cylinder and the end plate circles to the required
% angles
if angle_X1X2~=0 % Rotation is not needed if required direction is along X
    rotate(Cylinder,axis_rot,angle_X1X2,[0 0 0])
    if closed==1
        rotate(EndPlate1,axis_rot,angle_X1X2,[0 0 0])
        rotate(EndPlate2,axis_rot,angle_X1X2,[0 0 0])
    end
end
% Till now cylinder has only been aligned with the required direction, but
% position starts from the origin. so it will now be shifted to the right
% position
if closed==1
    set(EndPlate1,'XData',get(EndPlate1,'XData')+X1(1))
    set(EndPlate1,'YData',get(EndPlate1,'YData')+X1(2))
    set(EndPlate1,'ZData',get(EndPlate1,'ZData')+X1(3))

    set(EndPlate2,'XData',get(EndPlate2,'XData')+X1(1))
    set(EndPlate2,'YData',get(EndPlate2,'YData')+X1(2))
    set(EndPlate2,'ZData',get(EndPlate2,'ZData')+X1(3))
end
set(Cylinder,'XData',get(Cylinder,'XData')+X1(1))
set(Cylinder,'YData',get(Cylinder,'YData')+X1(2))
set(Cylinder,'ZData',get(Cylinder,'ZData')+X1(3))
% Setting the color to the cylinder and the end plates
set(Cylinder,'FaceColor',cyl_color)
if closed==1
    set([EndPlate1 EndPlate2],'FaceColor',cyl_color)
else
    EndPlate1=[];
    EndPlate2=[];
end
% If lines are not needed making it disapear
if lines==0
    set(Cylinder,'EdgeAlpha',0)
end
end
```

Appendix B.2 Speech-to-Text Code, Main System Software

Arm Class:

```
class Arm:
    q = []  # joint angle positions (length 3) VERTICAL, ROTATE, PAN
    o_curr = []  # x, y, z in space
    power_up = False  # variable to see if the power is up or down

    GPIO.setmode(GPIO.BCM)

    VERTICAL_MOTOR = 0
    ROTATE_MOTOR = 1
    PAN_MOTOR = 2

    ROTATE_FACTOR = 1
    GRIPPER_L_PIN_A = 0
    GRIPPER_L_PIN_B = 22
    GRIPPER_R_PIN_A = 17
    GRIPPER_R_PIN_B = 5

    ELEVATOR_MOTOR_PIN_A = 2
    ELEVATOR_MOTOR_PIN_B = 3
    ELEVATOR_ENCODER_PIN_A = 7
    ELEVATOR_ENCODER_PIN_B = 1

    NAME = 'Graham'
    OFFSET = 0

    WRIST_PAN_PIN = 18
    WRIST_ROTATE_SENSOR_PIN = 0
    WRIST_ROTATE_SERVO_PIN = 19

    START_Q = [0, 500, 0]
    OFF_Q = [0, 500, 0]

    MOVE_WORD = 'up'
    ROTATE_WORD = 'here'
    PAN_WORD = 'up'

    gripper_closed = [True, True]


    def __init__(self):
        pi = pigpio.pi()
        encoder = Encoder(pi, self.ELEVATOR_ENCODER_PIN_A, self.ELEVATOR_ENCODER_PIN_B)
        wrist_angle_sensor = AnalogInput(pi)
        gripper_motor_r = GearMotor(pi, self.GRIPPER_L_PIN_A, self.GRIPPER_L_PIN_B)
        gripper_motor_l = GearMotor(pi, self.GRIPPER_R_PIN_A, self.GRIPPER_R_PIN_B)
        wrist_rotate_motor = ServoMotor(pi, self.WRIST_ROTATE_SERVO_PIN)
        self.elevator_motor = GearMotor(pi, self.ELEVATOR_MOTOR_PIN_A,
                                self.ELEVATOR_MOTOR_PIN_B)

        self.pan = ServoDOF(pi, self.WRIST_PAN_PIN)
        self.rotate = MotorPIDDOF(pi, wrist_rotate_motor, wrist_angle_sensor, 1,
                        kp=-.001, ki=-.000, kd=-.000, MIN=0, MAX=1000)
        self.vertical = MotorPIDDOF(pi, self.elevator_motor, encoder, 0,
        kp=0.05, ki=0, kd=0, MIN=-9000, MAX=0)

        self.gripper_1 = Gripper(gripper_motor_l)
        self.gripper_2 = Gripper(gripper_motor_r)
        self.o_curr = FK(self.q)
        return

    def __full_set_position(self):
        self.pan.set_position(self.q[self.PAN_MOTOR])
        self.vertical.set_position(self.q[self.VERTICAL_MOTOR])
        self.rotate.set_position(self.q[self.ROTATE_MOTOR])
        self.gripper_1.close()
        self.gripper_2.close()
        return

    def __update_q(self, new_q):
        self.q = new_q
        return
```

DOF Superclass:

```
# DOFs
class DOF(object):

    def __init__(self):
        pass

    def set_position(self, position):
        pass

    def set_velocity(self, velocity):  # will effectively just set power
        pass

    def get_position(self):
        pass
```

Servo DOF Class:

```
class ServoDOF(DOF):

    FREQUENCY = 100
    pi = None
    pin = None

    MIN_LIMIT = 500
    MAX_LIMIT = 2500

    MIN_DEGREE = -90
    MAX_DEGREE = 90

    def __init__(self, pi, pin):
        super(ServoDOF, self).__init__()
        self.pi = pi
        self.pin = pin
        self.pi.set_mode(self.pin, pigpio.OUTPUT)
        return

    def __convert_to_degrees(self, thousands_val):
        thous_range = self.MAX_LIMIT - self.MIN_LIMIT
        deg_range = self.MAX_DEGREE - self.MIN_DEGREE
        new_value = (thousands_val - self.MIN_LIMIT) * deg_range
                    / thous_range + self.MIN_DEGREE
        return int(new_value)

    def __convert_to_thousands(self, degree_val):
        thous_range = self.MAX_LIMIT - self.MIN_LIMIT
        deg_range = self.MAX_DEGREE - self.MIN_DEGREE
        new_value = (degree_val - self.MIN_DEGREE) * thous_range
                    / deg_range + self.MIN_LIMIT
        return int(new_value)

    def __in_limits(self, thous_position):
        return self.MIN_LIMIT <= thous_position <= self.MAX_LIMIT

    def __out_of_limits(self, thous_position):
        print('Position input exceeds limits ' +
            str(self.__convert_to_degrees(thous_position)))
        return

    def set_position(self, deg_position):
        thous_position = self.__convert_to_thousands(deg_position)
        if self.__in_limits(thous_position):
            self.pi.hardware_PWM(self.pin, self.FREQUENCY, int(thous_position*100))
            print(thous_position*50)
            print('pos in limits')
            return True
        self.__out_of_limits(thous_position)
        return False
```

```python
def enable_pid(self):
    if self.pid_enabled == 0:
        self.pid_enabled = 1

def disable_pid(self):
    self.pid_enabled = 0
    self.motor.set_power(0)
    self.last_time = None

def get_counter(self):
    return self.counter

def interrupt_callback(self, gpio, level, tick):
    self.update_pid()

def update_pid(self):
    if self.pid_enabled == 0:
        return -1
    if (self.continuous == 1):
        rotationDistance = 1024
        sensor = self.sensor.get_value()

        errorCCWRot = (self.target - rotationDistance) - sensor
        errorNoRot = self.target - sensor
        errorCWRot = (self.target + rotationDistance) - sensor

        error = 0

        if (abs(errorCCWRot) < abs(errorNoRot)) and
            (abs(errorCCWRot) < abs(errorCWRot)):
            error = errorCCWRot
        if (abs(errorNoRot) < abs(errorCCWRot)) and
            (abs(errorNoRot) < abs(errorCWRot)):
            error = errorNoRot
        if (abs(errorCWRot) < abs(errorCCWRot)) and
            (abs(errorCWRot) < abs(errorNoRot)):
            error = errorCWRot

    else:
        error = self.target - self.sensor.get_value()

    if self.last_time is not None:
        d_t = time.clock() - self.last_time
        self.integral = self.integral + error+d_t
        derivative = (error - self.last_error)/d_t

        power = self.kp*error + self.ki*self.integral + self.kd*derivative
        self.motor.set_power(max(-1, min(1, power)))
        #print(power)
        if power > 0:
            self.sensor.set_direction(1)
        if power < 0:
            self.sensor.set_direction(-1)
    else:
        self.integral = 0
```

Microphone Stream Class:

```
class MicrophoneStream(object):
    def __init__(self, rate, chunk):
        self._rate = rate
        self._chunk = chunk

        # Create a thread-safe buffer of audio data
        self._buff = queue.Queue()
        self.closed = True

    def __enter__(self):
        self._audio_interface = pyaudio.PyAudio()
        self._audio_stream = self._audio_interface.open(
            format=pyaudio.paInt16,

            channels=1, rate=self._rate,
            input=True, frames_per_buffer=self._chunk,

            stream_callback=self._fill_buffer,
        )

        self.closed = False

        return self

    def __exit__(self, type, value, traceback):
        self._audio_stream.stop_stream()
        self._audio_stream.close()
        self.closed = True

        self._buff.put(None)
        self._audio_interface.terminate()

    def _fill_buffer(self, in_data, frame_count, time_info, status_flags):
        """Continuously collect data from the audio stream, into the buffer."""
        self._buff.put(in_data)
        return None, pyaudio.paContinue

    def generator(self):
        while not self.closed:

            chunk = self._buff.get()
            if chunk is None:
                return
            data = [chunk]

            while True:
                try:
                    chunk = self._buff.get(block=False)
                    if chunk is None:
                        return
                    data.append(chunk)
                except queue.Empty:
                    break

            yield b''.join(data)


def listen_print_loop(responses):

    transcript = ''
    num_chars_printed = 0
    for response in responses:
        if not response.results:
            continue

        result = response.results[0]
        if not result.alternatives:
            continue

        transcript = result.alternatives[0].transcript

        overwrite_chars = ' ' * (num_chars_printed - len(transcript))

        if not result.is_final:
            sys.stdout.write(transcript + overwrite_chars + '\r')
            sys.stdout.flush()

            num_chars_printed = len(transcript)

        else:
            print(transcript + overwrite_chars)

            if re.search(r'\b(Graham|quit)\b', transcript, re.I):
                print('Properly Formatted Command')
            else:
                transcript = ''
            break

    return transcript
```

Main Class:

```python
name = 'Graham'
key_words_arr = [name, 'move up', 'move down', 'inches', 'degrees',
                 'rotate in', 'rotate out', 'pan up', 'pan down']
language_code = 'en-US'  # a BCP-47 language tag

client = gc.speech.SpeechClient()
config = gc.types.RecognitionConfig(
    encoding=gc.enums.RecognitionConfig.AudioEncoding.LINEAR16,
    sample_rate_hertz=gc.RATE,
    language_code=language_code,
    model='command_and_search',
    speech_contexts=[gc.speech.types.SpeechContext(
        phrases=key_words_arr)])
streaming_config = gc.types.StreamingRecognitionConfig(
    config=config,
    single_utterance=True,
    interim_results=True)

dextera = Arm()

while True:
    try:
        with gc.MicrophoneStream(gc.RATE, gc.CHUNK) as stream:
            audio_generator = stream.generator()
            requests = (gc.types.StreamingRecognizeRequest(audio_content=content)
                        for content in audio_generator)

            responses = client.streaming_recognize(streaming_config, requests, timeout=15)
            transcript = gc.listen_print_loop(responses)
            dextera.parse_text(transcript)
            if transcript == '':
                print ('NO VALID COMMAND GIVEN')
            else:
                print("Google thinks you said: " + transcript)
            if transcript == 'stop listening ' + name:
                break
    except Exception as exception:
        print('Exception handle: Exceed max stream of 15 seconds')
```

Appendix C.1 Robotic Kinematics Calculations

Denavit-Hartenberg Parameters:

Link	a	d	α	θ
1	0	d_0	0	0
2	0	d_1^*	$\pi/2$	0
3	0	l	$\pi/2$	$\theta_2^* + \pi/2$
4	lg	0	0	$\theta_3^* + \pi/2$

$$d_0 = 60mm, \text{ base height}$$
$$l = 125mm, \text{ length of arm}$$
$$lg = 75mm, \text{ length of gripper}$$

Homogeneous Transformation Matrix, General form for joints i and i + 1:

$$\mathrm{T}_{i+1}^{i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Transformation Matrices for Dextera's 3 joints and 4 corresponding reference frames:

$$T_1^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & q_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} cos(q_2+\pi/2) & 0 & sin(q_2+\pi/2) & 0 \\ sin(q_2+\pi/2) & 0 & -cos(q_2+\pi/2) & 0 \\ 0 & 1 & 0 & l \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = \begin{bmatrix} cos(q_3+\pi/2) & -sin(q_3+\pi/2) & 0 & lg*cos(q_3+\pi/2) \\ sin(q_3+\pi/2) & cos(q_3+\pi/2) & 0 & lg*sin(q_3+\pi/2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Total Transformation Matrix Equation:

$$\mathbf{T}_e^0 = \mathrm{T}_4^0 = \mathrm{T}_1^0\ \mathrm{T}_2^1\ \mathrm{T}_3^2\ \mathrm{T}_4^3$$

End Effector Position Vector Equations, Taken from result of above equation:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} lg\cos(\theta_2^* + \pi/2)\cos(\theta_3^* + \pi/2) \\ lg\sin(\theta_3^* + \pi/2) - l \\ lg\sin(\theta_2^* + \pi/2)\cos(\theta_3^* + \pi/2) + d_1^* + d_0 \end{bmatrix}$$

Joint Positions from End Effector Position Vector:

$$\theta_3^* = sin^{-1}((o_y + l)/lg) - \pi/2$$

$$\theta_2^* = cos^{-1}(o_x/(lg * cos(q_3 + \pi/2))) - \pi/2$$

$$d_1^* = o_z - d_0 - lg * sin(q_2 + \pi/2) * cos(q_3 + \pi/2)$$

Jacobian Matrix:

$$J = \begin{bmatrix} 0 & -lg\cos(q_3 + \pi/2)\sin(q_2 + \pi/2) & -lg\cos(q_2 + \pi/2)\sin(q_3 + \pi/2) \\ 0 & 0 & lg\cos(q_3 + \pi/2) \\ 1 & lg\cos(q_2 + \pi/2)\cos(q_3 + \pi/2) & -lg\sin(q_2 + \pi/2)\sin(q_3 + \pi/2) \\ 0 & 0 & \sin(q_2 + \pi/2) \\ 0 & -1 & 0 \\ 0 & 0 & -\cos(q_2 + \pi/2) \end{bmatrix}$$

Linear and Angular Velocity, Using above Jacobian:

$$\vec{v}_e^0 = J_v(\vec{q})\dot{\vec{q}}$$

$$\vec{\omega}e^0 = J_\omega(\vec{q})\dot{\vec{q}}$$

Joint Velocity, Using above equations and inverse of Jacobian:

$$\dot{\vec{q}} = (J_v(\vec{q}))^{-1}\vec{v}$$