SCARA DRAWING ROBOT

By

Tianhao Chi

Chenghao Duan

Bingzhe Wei

Final Report for ECE 445, Senior Design, Spring 2018

TA: Dongwei Shi

May 1, 2018

Project No. 19

Abstract

We report on the design and implementation of a SCARA-style robotic arm image drawing system capable of emulating arbitrary artistic styles. Our system consists of a custom-designed, 3D printed SCARA-style robotic arm with joints driven by stepper motors and model servos, motor control electronics, and PC-side image processing software utilizing deep neural networks. To use the system, a user inputs both a desired image to draw, and another image with the desired artistic style to emulate. A deep neural network based optimization method then transforms the image to draw to the desired artistic style. The image is then vectorized, and inverse kinematics determines the necessary SCARA arm poses. The motor control electronics are then commanded to actuate appropriate motors to place the SCARA arm in the necessary poses to draw.

Contents

1. Introduction

1.1 Objective

Since the advent of computers and subsequently artificial intelligence, a key issue, and one that remains particularly elusive, has been creativity, the ability to generate output that is both novel and of significance as would be recognizable by humans. Although alternative programming paradigms that require less strict process specification and that can infer certain details automatically have been developed, these logic-based paradigms still require relative stringent and particular specifications as compared to natural language descriptions and are more difficult to use, and hence most computing today is still based on the imperative paradigm, resulting in algorithms can solve well-defined specific problems for which they have been designed to solve with high computational efficiency, but require manual algorithm discovery and implementation.

Whereas robotics is now commonly used in many industrial applications, there have been far fewer applications of robotics for artistic purpose, and we are yet to see a robot artist. As compared to other robotic arm designs, the Selective Compliance Assembly (SCARA) robotic arm design, originally designed for assembly line work, enables very high manipulator velocities with high positioning accuracy [1]. We thus seek to combine robotics, specifically, a SCARA robotic arm, together with deep neural networks, to create a robot capable of learning the art style of any image, applying the learned style, or a synthesized novel style, to any given input image, and finally output the transformed output image onto a physical medium, and hence demonstrate a significantly higher level of creative behavior by an artificial intelligence system.

1.2 High-Level Requirements List

a. The robotic arm should be able to draw an image of size bounded by the size of a sheet of Letter-sized paper with arbitrary content onto a sheet of Letter-sized paper.
b. The PC-side processing program should be able to produce an artistically appealing output image for any user given input image.
c. The PC-side processing program should be able be transform the above output image into a set of commands for the robotic arm.

1.3 Block Diagram

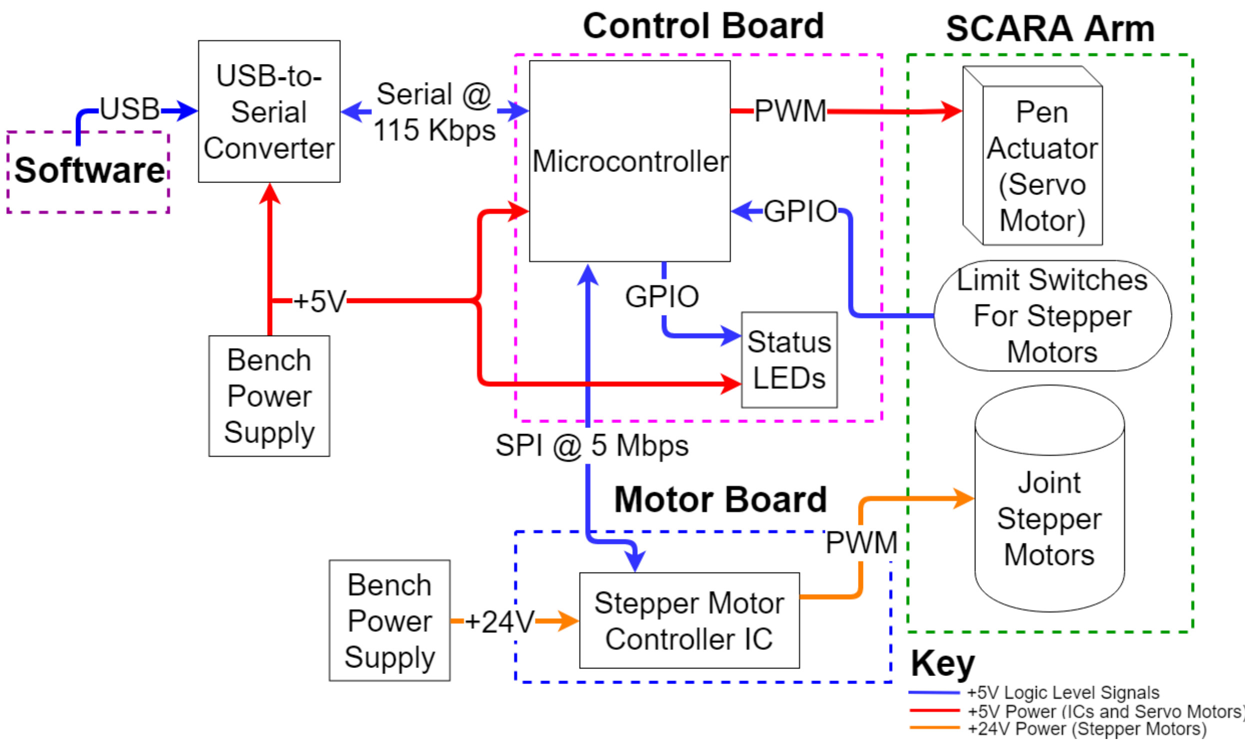


Figure 1: Block Diagram

Overall, the system consists of four components: software running on a PC performing all image processing steps, a control board with a microcontroller to coordinate all motor movements, a motor driver board to control the stepper motors required due to the high current and voltage needed to drive the stepper motors, and the mechanical components of the SCARA robot arm. Given an input image and a style image, the PC software first performs style transformation on the input image using a deep neural network, then converts the result into a series of arm angles by clustering colors and thresholding to get color regions, which are then converted into a series of lines via a vectorization algorithm, before finally transforming the lines into a list of arm angles via inverse kinematics. The result is sent to the control board via a USB serial port provided by a USB-to-Serial IC, and subsequently, the control board sends stepper motor positioning commands via SPI to the stepper motor controller ICs on the motor board, and pen actuator commands directly as PWM waveforms. The resulting sequence of motor motions then produces physical output on a piece of Letter-sized paper placed appropriately beforehand in the robotic arm's workspace.

2. Design

2.1 Physical Design

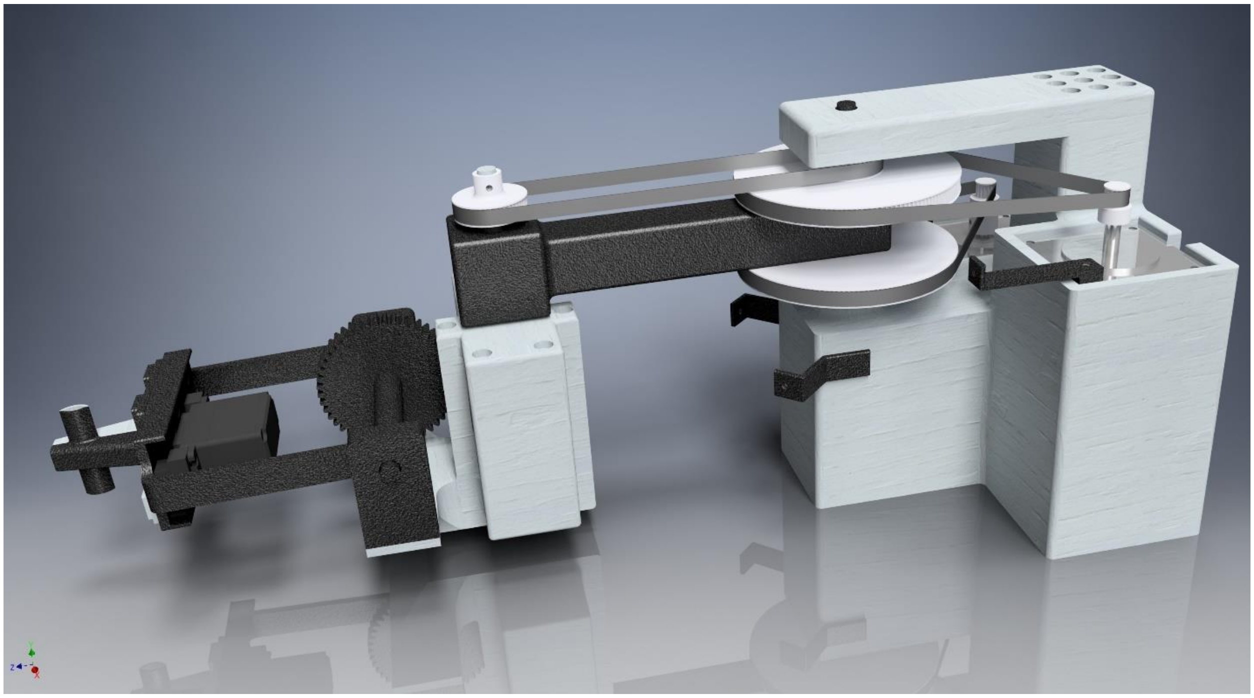


Figure 2: SCARA Robotic Arm Proposed Design

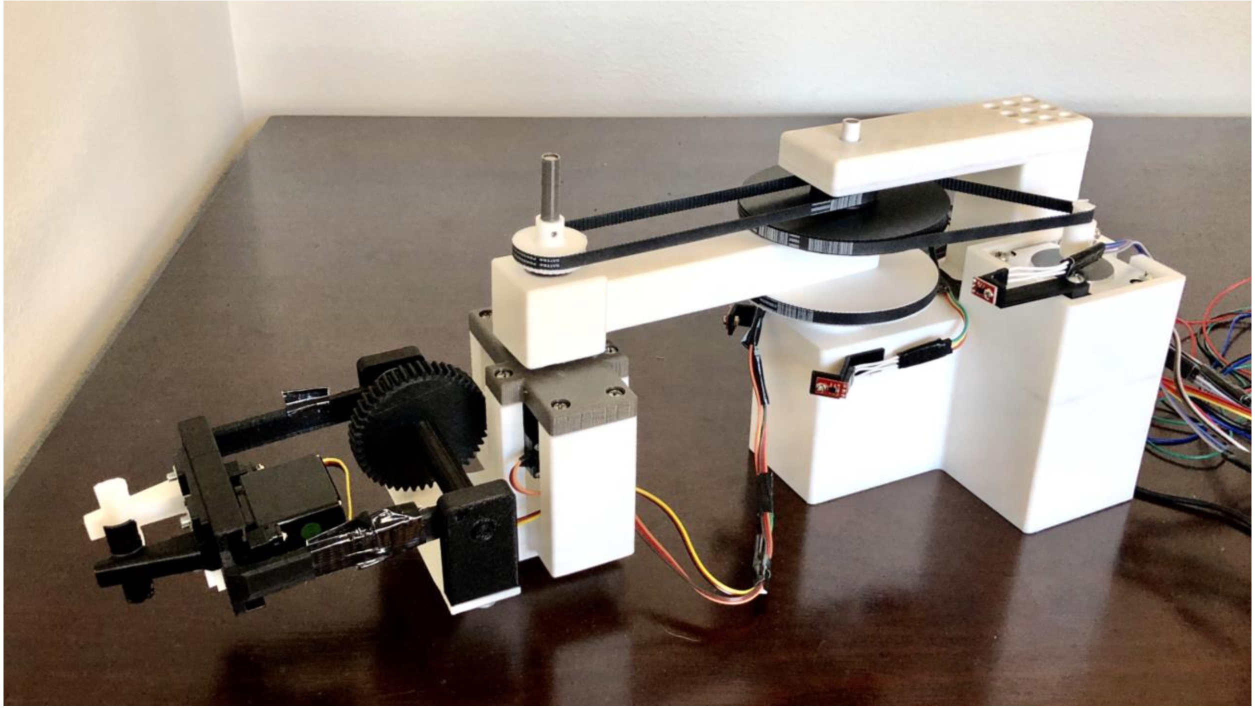


Figure 3: SCARA Robotic Arm Implemented Result

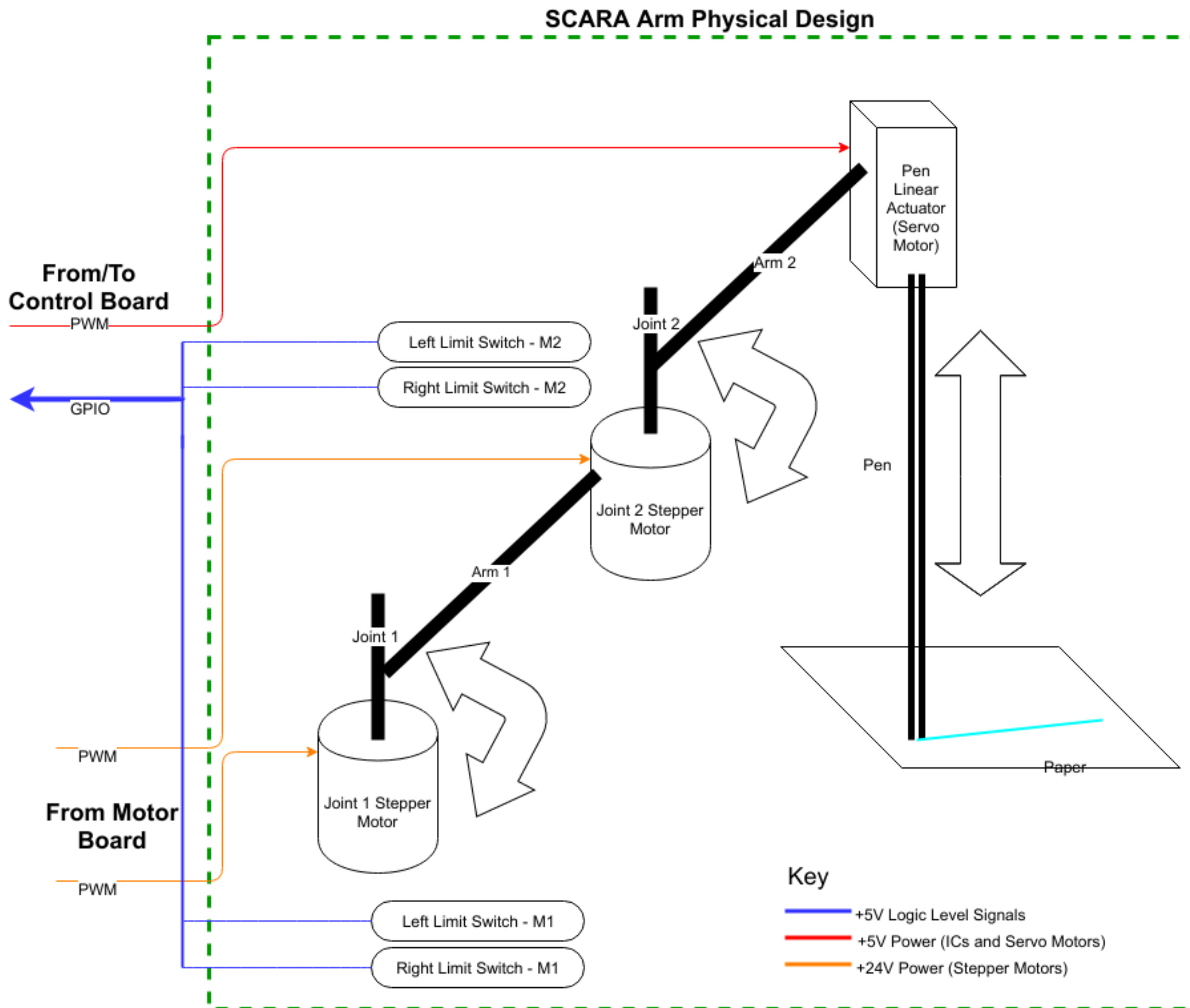


Figure 4: SCARA Robotic Arm Block Diagram

The physical design of the proposed robot arm is shown in Figure 2, while the block diagram of the SCARA robotic arm is shown in Figure 4. The SCARA arm type consists of a chain of two arm segments interconnected with a revolute joint, with a rotational platform controlling a pen located at one end and another revolute joint at the other end that joins the arm with a base. Each revolute joint is driven via stepper motors though a short timing belt and aluminum gears and supported via bearings to ensure accurate movement with low slippage, while rotational platform is driven by a standard size servo to allow for pen color changes, with three sub-micro servos each attached to a pen to enable accurate positioning with minimal weight and size. The pen is placed in and out of contact with the paper by rotating the sub-micro servo attached to that pen. As compared to other robotic arm designs, the SCARA robotic arm design, originally designed for assembly line work, enables very high manipulator velocities with high positioning accuracy [1], as all joints are positioned such that self-collisions are impossible, enabling the ability to move all joints independently and simultaneously. Meanwhile, the arm provides compliance only in the x-y plane [2], ensuring that the pen will always stay perpendicular to the work surface and thus further enhancing accuracy. As the workspace of a SCARA robotic arm is a distorted cardioid-like shape, we plan to have each arm segment be eight inches long, such that a piece of Letter-sized paper fits within the workspace of the robotic arm. We constructed the robotic arm via 3D printing and bolted connections. Fast and accurate positioning, with very low vibration, is required to produce a high-quality output with true edges instead of wavy lines in a reasonable amount of time, which necessitates not only high-powered motors but also smooth, low-slip, low-backlash joints and transmissions, and we have used synchronous belt drives to drive each arm segment. As can be seen from Figure 2, Arm Segment 1 is driven by a 1:3 ratio synchronous belt drive, while Arm Segment 2 is driven by two 1:3 ratio synchronous belt drive in series.

2.1.1 SCARA Robotic Arm

The SCARA arm will be responsible for translating the arm angles that the PC program outputs into physical motion resulting in a visible output on the paper. This requires both high positioning velocity as well as accuracy, which we enable via a stable mechanical design that bears the load with bearing in place of the motor shaft and the accuracy and high holding torque of stepper motors.

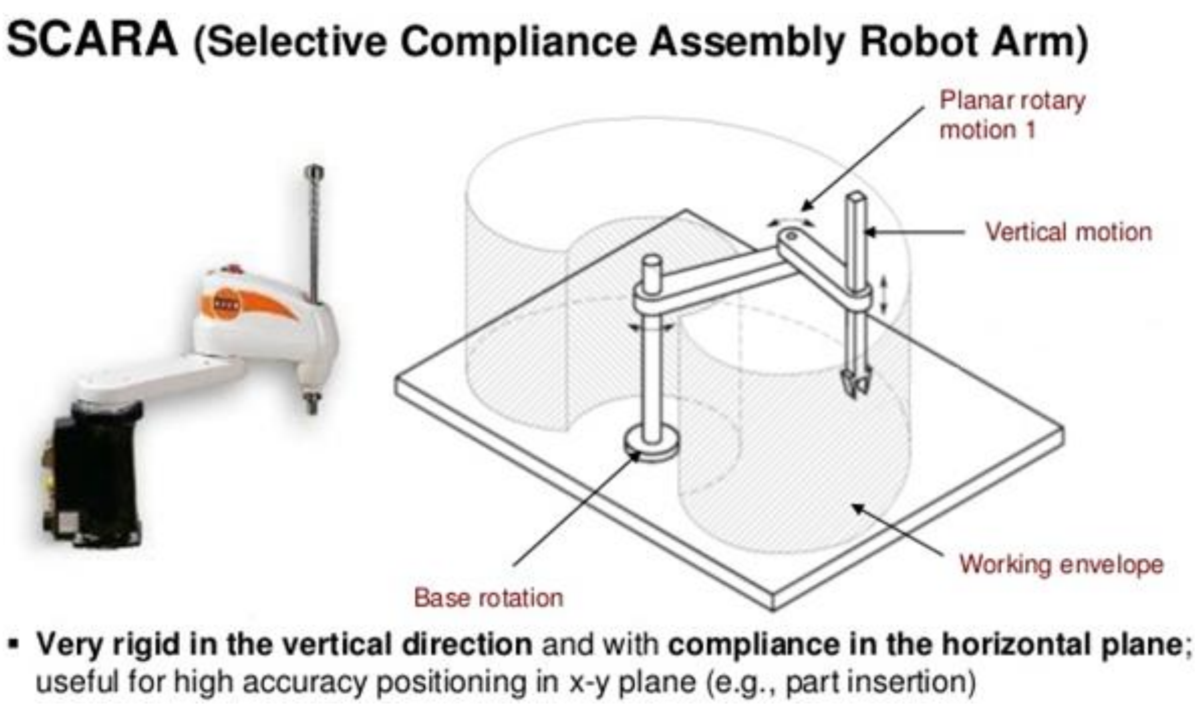


Figure 5: SCARA Robotic Arm Workspace

As the workspace of a SCARA robotic arm is a distorted cardioid-like shape, we plan to have each arm segment be eight inches long, such that a piece of Letter-sized paper fits within the workspace of the robotic arm.

Please see Appendix A for details about our physical design, including workspace reachability calculations, derivation of inverse kinematics equations, modal simulations, motor torque requirement calculations, and tolerance analysis, and more details about the pen actuator mechanism.

2.1.1 Joint Motor 1, Motor 2

These stepper motors move arm segment one and arm segment two respectively. They require high holding torque to ensure no slippage and fast movement. As our mechanical design has the motors mounted to the base with power transmission via timing belts and gears, which means the motors do not themselves need to be moved as part of either arm segment, neither mass nor heat dissipation i.e. current consumption, is a significant concern.

2.1.2 Pen Actuator

This actuator rotates the pen up and down onto the paper to enable drawing. While high speed is preferred to allow for fast drawing of certain styles e.g. points, very little travel is required. The actuator needs to prevent compliance in the x-y plane, but should be somewhat compliant in the z-axis to prevent breaking the pen or scratching the paper at high arm travel velocities. The linear actuator is controlled by a sub-micro servo responding to appropriate PWM waveforms generated by the microcontroller in the control board. Unlike the arm joint motors, PID control is integrated within the servo, and hence a control loop running on the microcontroller is not required for this linear actuator. To enable pen changing, we use a claw mechanism driven by a rack and pinion inspired by [3] to allow for an arbitrary number of colors to be drawn. The servo model used in Figure 2 was found on GrabCAD [4].

2.1.3 Arm Segment/Joint

These arm segments form the actual robot arm, and necessitates a stable mechanical design that bears the load with bearing in place of the motor shaft to ensure positioning accuracy and speed. They must have very small deflection under full scale load to prevent vibrations and resonance that impair accuracy, and be of minimal mass to avoid loading down the joint motors. Please note that these components have been grouped together as our fabrication process is expected to arm segments and joints as integrated units, rather than as separate units.

2.1.4 Limit Switches

These limit switches are used to determine if an arm segment has reached the end of its allowable travel range and thus prevent the robotic arm from colliding with the base or itself, and are also used at startup to calibrate the zero position of the robotic arm, as by counting the number of steps sent to the stepper motor controllers allows relative position tracking, but not absolute position tracking. For simplicity, these switches will be connected directly to the microcontroller's GPIO pins and utilize built-in pull-up resistors.

2.2 Software

Users are expected to have an image and a desired style as input. Given an input image and a style, the PC software first performs style transformation on the input image using a deep neural network for the selected style, then converts the result into a series of arm angles by clustering colors and thresholding to get color regions, which are then converted into a series of lines via a vectorization algorithm, with fill patterns optimized for the polar nature of the SCARA robotic arm, before finally transforming the lines into a list of arm angles via inverse kinematics. Subsequently, the list of arm angles is transferred to the microcontroller via the USB-to-Serial converter IC. Meanwhile, the microcontroller software simply waits for the list of arm angles, and upon receiving the list of arm angles, actuates the motors and servos as necessary, halting if any abnormal conditions including motor overtemperature, motor overcurrent, and motor stall are detected.

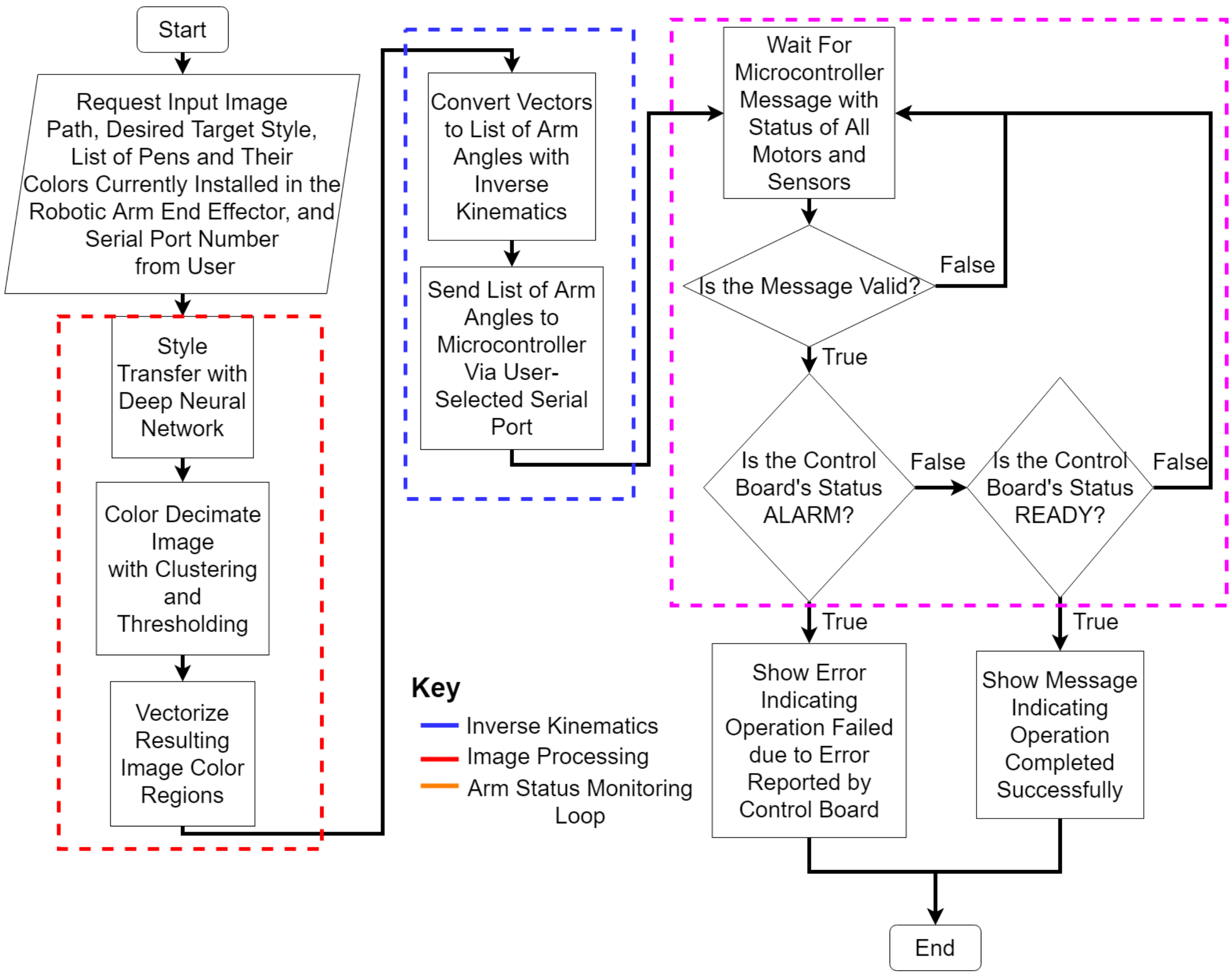


Figure 6: Software Logic Overall Flowchart

Please see Appendix B for more detailed software logic flowcharts, including the microcontroller software logic flowchart.

2.2.1 Neural Network Style Transfer

Given an input image and a style image, the PC software is able to transform the input image using a deep neural network for the selected style. The implementation uses TensorFlow [5] to train a fast style transfer network based on the method reported by Gatys et al. [6], using a content weight of five and a style weight of five hundred.

2.2.2 Image Color Quantization

As our robot uses pens and therefore supports certain discrete colors color, the input image must be converted to our robot's colorspace. Specifically, we choose to support red, orange, yellow, green, blue, purple, aqua, black, and white. We implement this via a palettization algorithm using the ***convert()*** method from the Image module of the Python package Pillow [7].

Table 12: Supported Colors and Their Corresponding RGB Values

Color	RGB Value
Red	255, 0, 0
Orange	255, 127, 0
Yellow	255, 255, 0
Green	0, 255, 0
Blue	0, 0, 255
Purple	127, 0, 127
Black	0, 0, 0
White	255, 255, 255
Aqua	0, 255, 255

Figure 7: Before Quantization

Figure 8: After Quantization

2.2.3 Filling Holes

Quantization may result in many small regions, each of a different color, scattered in inside another large region of color, with the color of the large region differing from the color of any aforementioned small region. As these small regions are due to quantization noise and are not drawable due to their small size relative to the diameter of the pen tip, we used the OpenCV ***binary_fill_holes()*** function [8], which fills such holes by filling the complement of the input image's outer boundary using binary dilations and erosions. Since holes are not connected to the outer boundary, they are thus filled.

2.2.4 Space Filling and Vectorization

To determine a path for the robot arm, we use graph search for speed and provable optimality. Each pixel is assigned a node number using the following equation, where x is the column location of the pixel, y is the row location of the pixel, and width is the width of the image in pixels.

$$Node\ Index = width \times y + x. \tag{1}$$

After appending all the nodes to the graph, we need to add edges between adjacent pixels. We consider eight types of edges: up, down, left, right, up left, up right, down left, and down right, via conditional statements for boundary checking, and iterate through every pixel to build the necessary graph for the search algorithm.

2.4.2 Depth-first Search Algorithm

To determine a path between nodes in a graph, Depth-first search (DFS) is an optimal algorithm. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking. We use the method ***dfw_edges(G, source=none)*** method inside NetworkX [9] to perform a DFS traversal on each connected component of the graph built in Section 2.2.4. Using the ***connected_component_subgraphs(G, copy=True)*** method of NetworkX, we split the resulting traversal into subgraphs, one for each connected component of the original graph.

```
procedure DFS(G,v):
    label v as discovered
    for all edges from v to w in G.adjacentEdges(v) do
        if vertex w is not labeled as discovered then
            recursively call DFS(G,w)
```

Figure 9: Depth-first Search Pseudocode

2.4.3 Optimization

To reduce the amount of data transmitted to the robot arm, the pixel in the middle of any straight line of pixels is omitted, as the microcontroller software linearly interpolates between the endpoints of movement commands and our program is such that the pen is always contact with the paper when drawing, and so any pixels in on such lines, other than the endpoints of such lines, would be drawn without requiring an explicit movement command.

Figure 10: Before Optimization

Figure 11: After Optimization

2.4.4 Vectorize and Inverse Kinematics
After the preceding steps, the edge list is vectorized to a list of nodes, then converted to angular positions of the joint motors/servos using the inverse kinematics algorithm described in Appendix A.

2.3 Electronics

2.3.1 Control Board
The control board acts as the central coordinator of the system, receiving lists of arm angles from the PC via a USB-to-Serial interface IC, and sends appropriate command sequences to the joint motors and linear actuators while maintaining accurate position control throughout via generate appropriate SPI messages and PWM signals respectively. The control board monitors for abnormal conditions, including Motor Board overtemperature, motor overcurrent, motor stalling, and arm segments reaching end of travel, as determined by the limit switches and other sensors built into the stepper motor controller IC, and stops the system as necessary, and reports any such conditions to the PC software and via LEDs such that the user can take corrective action. The control board has been designed with servo-style connectors to ease servo and active sensor connections, as well as protection resistors on the SPI and serial lines to enable programming even while motor boards are attached.

Please see Appendix C for circuit schematics and PCB layout diagrams.

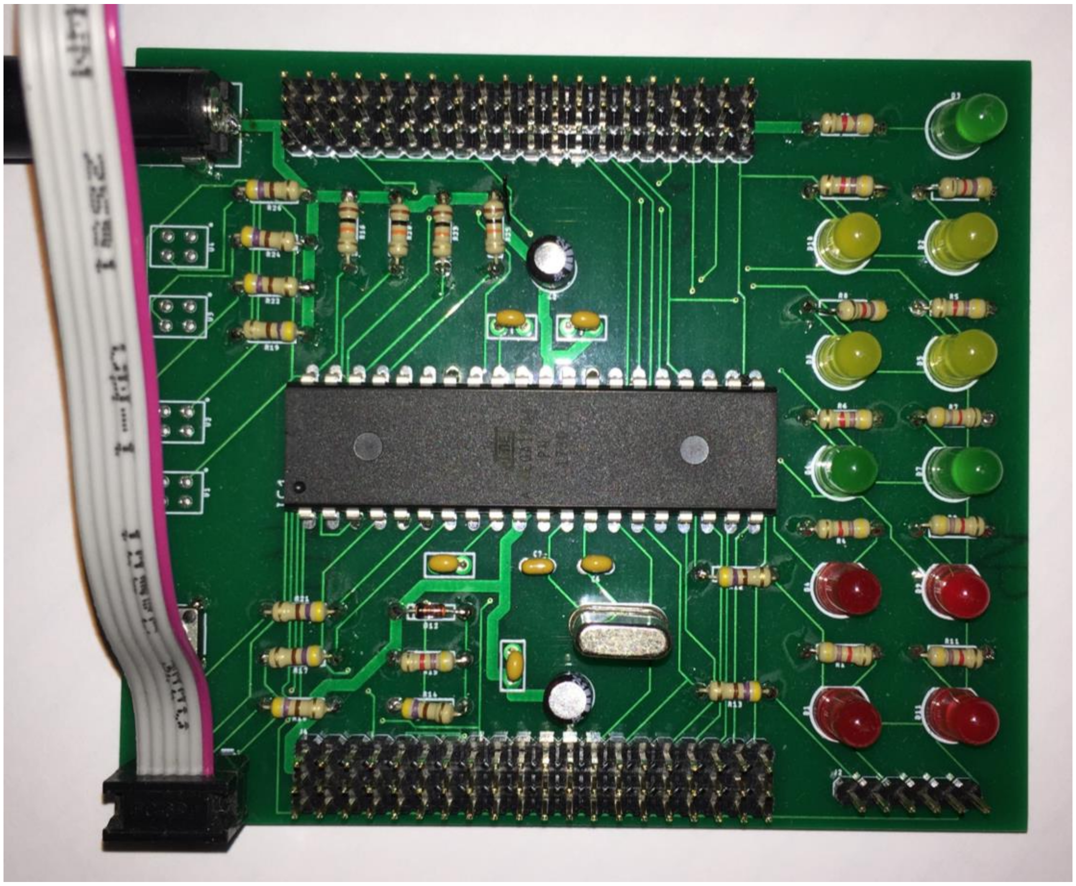


Figure 12: Control Board PCB Assembled

2.3.2 Motor Board

The Motor Board simply houses the stepper motor controller ICs and any needed heatsinks. Due to the higher drive voltages used, to ensure isolation, and to allow for mounting closer to each joint motor, the Motor Board is a separate, self-contained unit. Each stepper motor controller IC monitors for abnormal conditions, including overtemperature, motor overcurrent, and motor stalling, and relays this information back to the microcontroller via the SPI bus, and simultaneously outputs the necessary current-limited drive waveform to position the stepper motors at the correct angles as commanded by messages sent from the microcontroller over the SPI bus. Due to the inherent back-EMF of the stepper design, stepper motors require high voltage for high rotational speeds. We use a constant current PWM waveform to prevent the motor coils from overheating, and also to allow for microstepping, increasing stepper motor positioning accuracy and movement smoothness. The stepper motor controller ICs communicate with the microcontroller over the SPI bus, while power is supplied by a separate +24V bench power supply. To allow for heatsinking via the underside thermal pad, the motor board has been designed with two ground planes (using a four-layer PCB process), exposed copper areas, and thermal vias. Traces with a width of 80 mils are used to handle the high motor currents, while duplicate SPI connectors enable multiple Motor Boards to be chained, simplifying wiring requirements.

Please see Appendix C for circuit schematics and PCB layout diagrams.

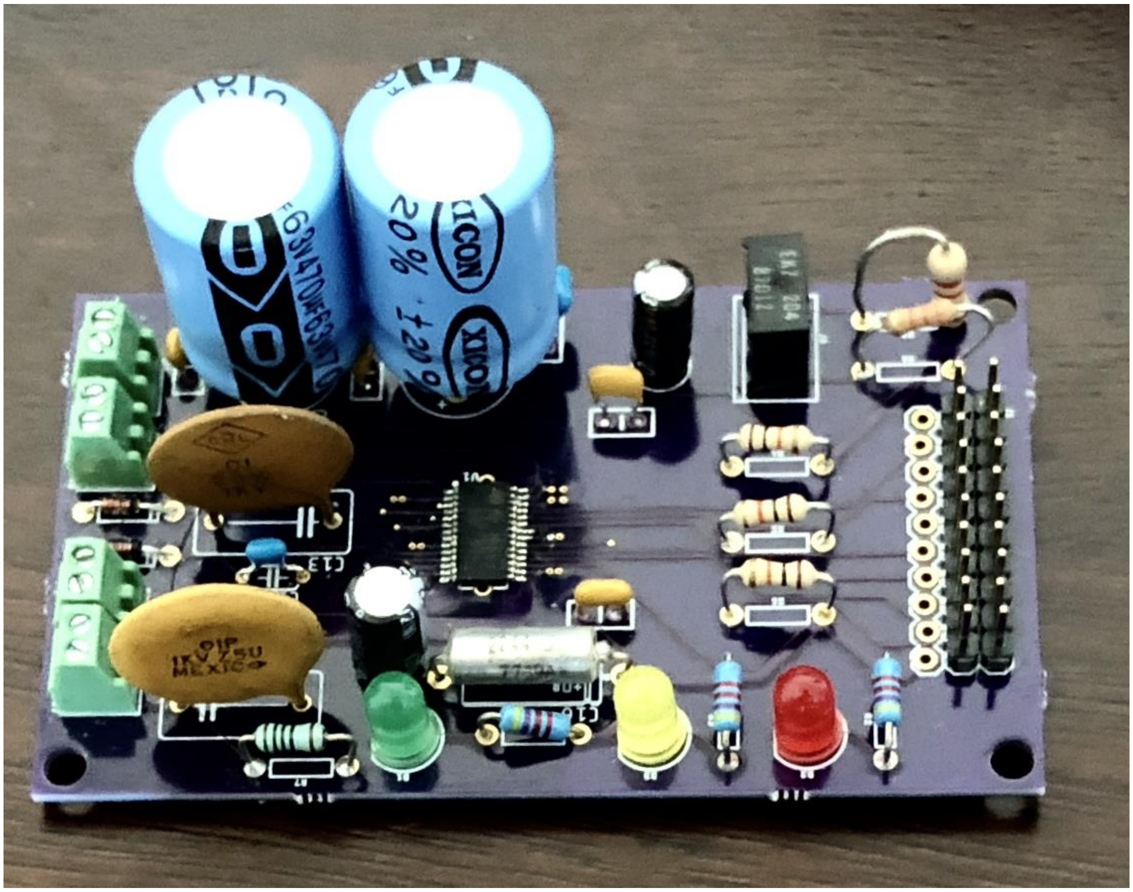


Figure 13: Motor Board PCB Assembled

3. Design Verification

3.1 Software Test

We demonstrate the PC image processing software's functionality below.

Figure 14: Input Image

Figure 15: Style Image

Figure 16: Input Image After Style Transfer

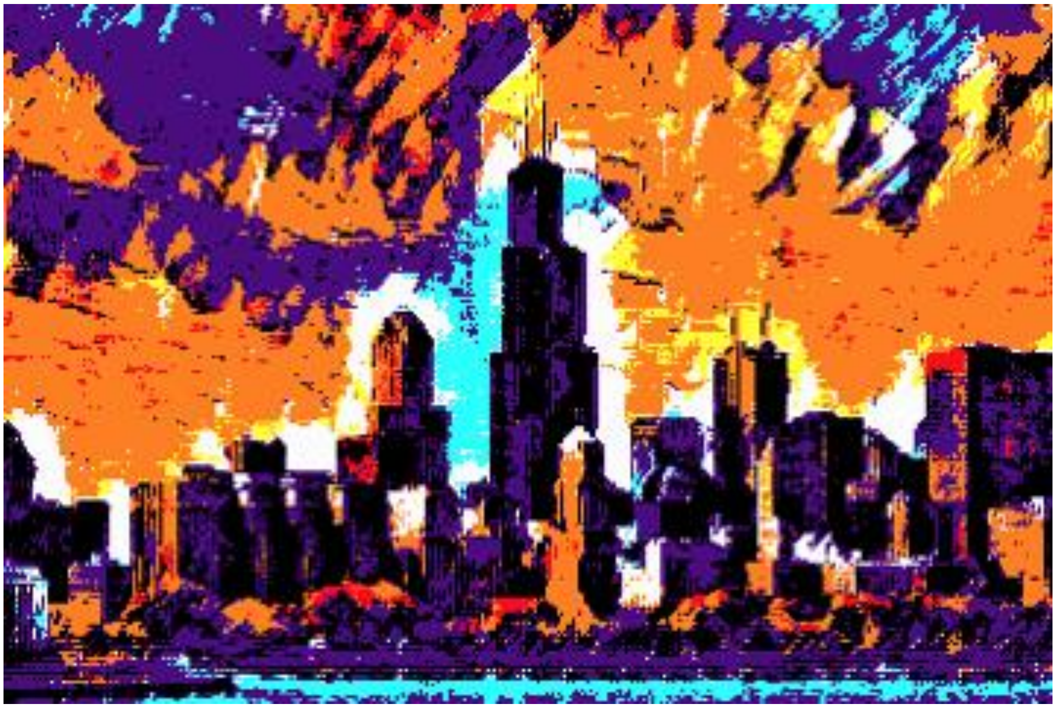


Figure 17: Input Image After Style Transfer and Quantization

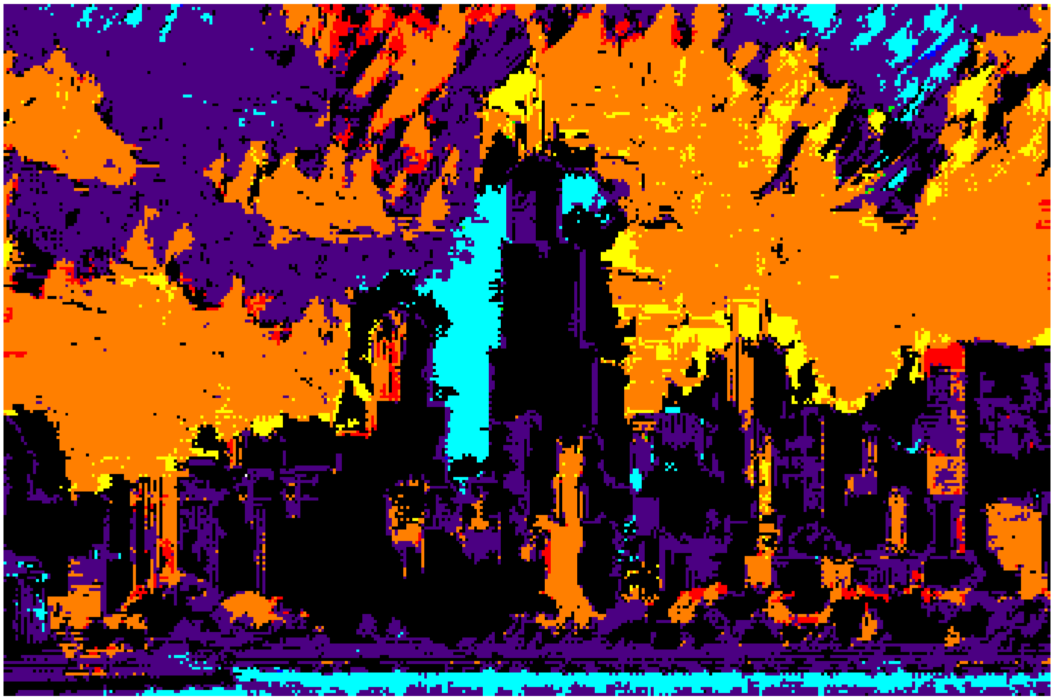


Figure 18: Input Image After Style Transfer, Quantization, and Hole Filling

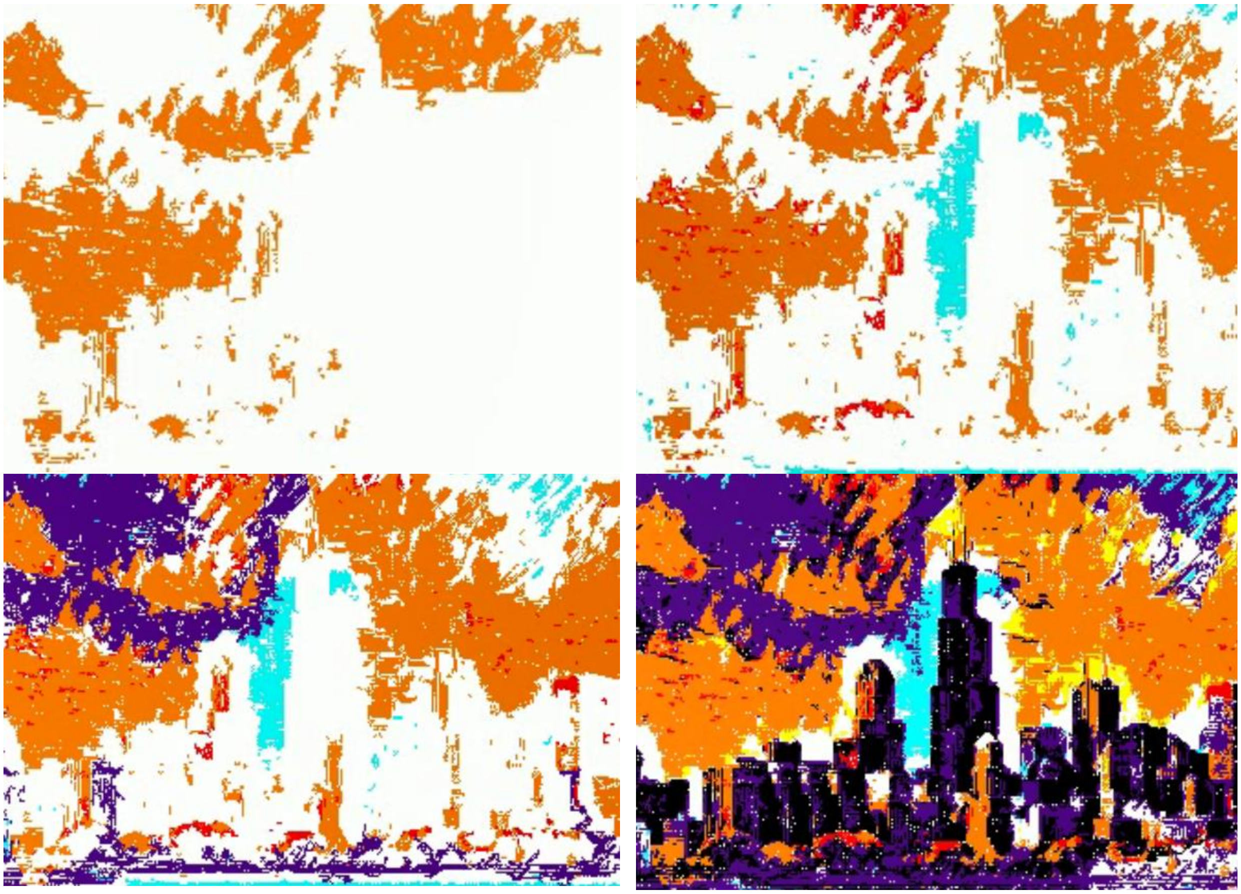


Figure 19: Final Generated Path Animation – Selected Frames

```
up
first: 9147
second: 12197
down
first: 9231
second: 12136
first: 9276
second: 12124
first: 9366
second: 12058
first: 9461
second: 11988
first: 9562
second: 11913
first: 9671
second: 11831
first: 9727
second: 11808
first: 9850
second: 11714
up
first: 9732
second: 11765
down
first: 9617
```

Figure 20: Final Generated Motor Commands

3.2 Integration Tests

In the tests that follow, we integrate all components and demonstrate achieved system functionality. As our tests results are video recordings, we have placed selected frames that demonstrate the necessary functionality below. The results of Integration Tests 1-3, which test the ability of the two joints to move independently, are shown in Appendix F.

3.2.1 Integration Test 4: Zero All Joints and Set Home

In this test, we test the functionality of the home subroutine. As we are using open loop control with stepper motors, we need to set all joints to a zero position upon beginning any drawing commands, as the position of each motor is unknown upon start up.

Figure 21: Test 4 Initial Arm Pose

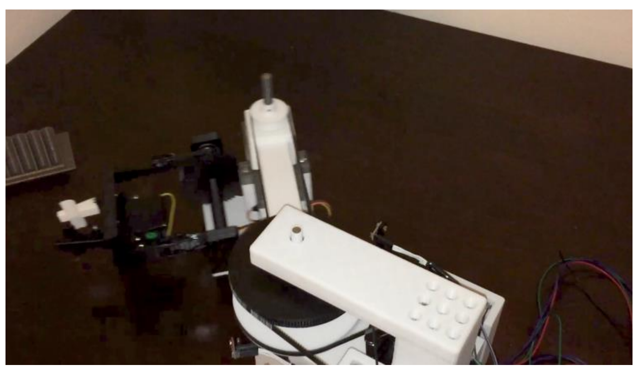


Figure 22: Test 4 Intermediate Arm Pose

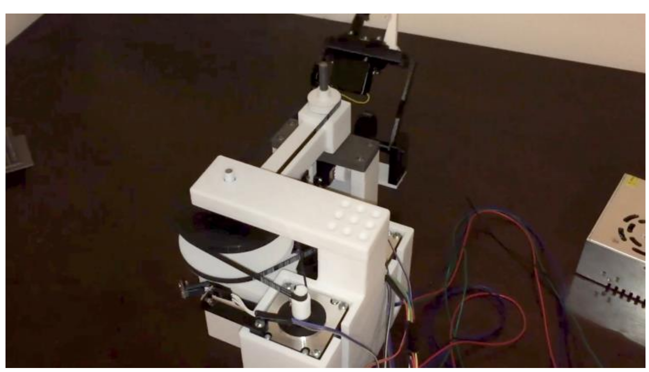


Figure 23: Test 4 Final Arm Pose

3.2.2 Integration Test 5: Pen Actuator Test

In this test, we command the pen actuator to raise and lower, and to open and close.

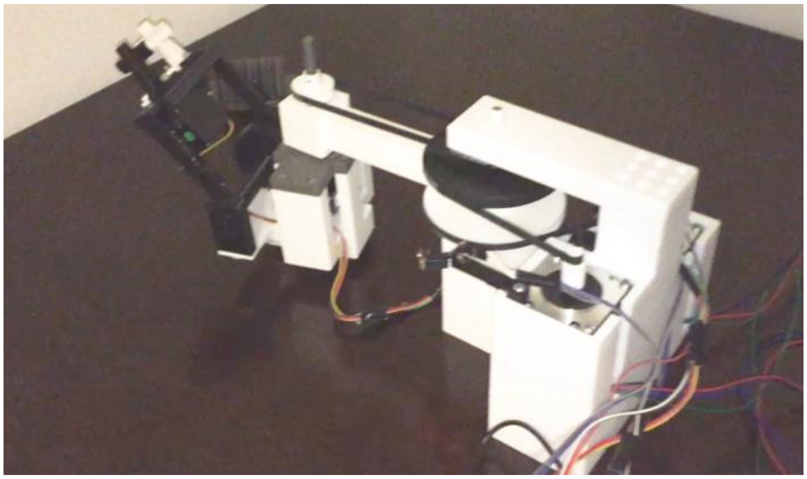


Figure 24: Test 5 Initial Arm Pose

Figure 25: Test 5 Intermediate Arm Pose

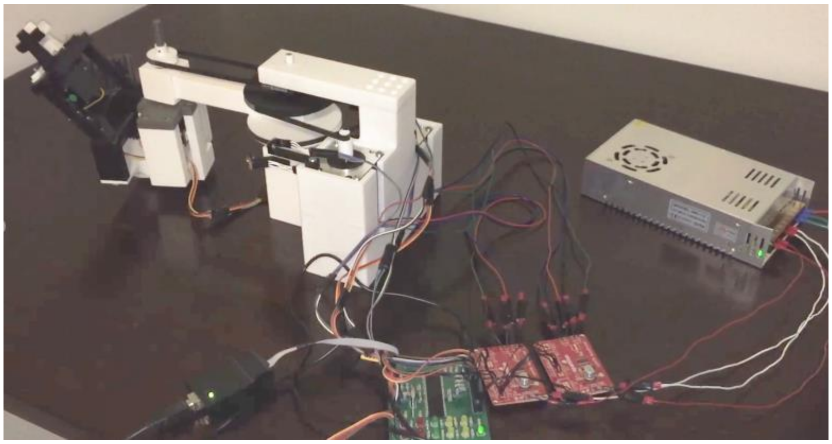


Figure 26: Test 5 Final Arm Pose

3.2.3 Integration Test 6: Drawing Test

In this test, we draw a small circle.

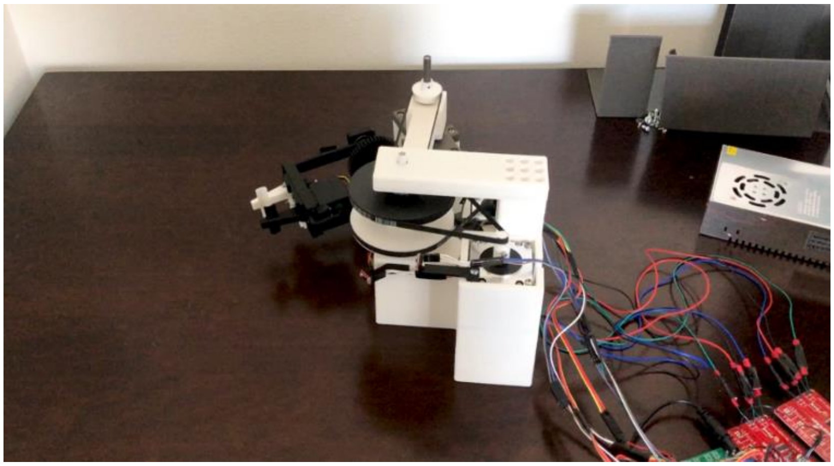


Figure 27: Test 6 Initial Arm Pose

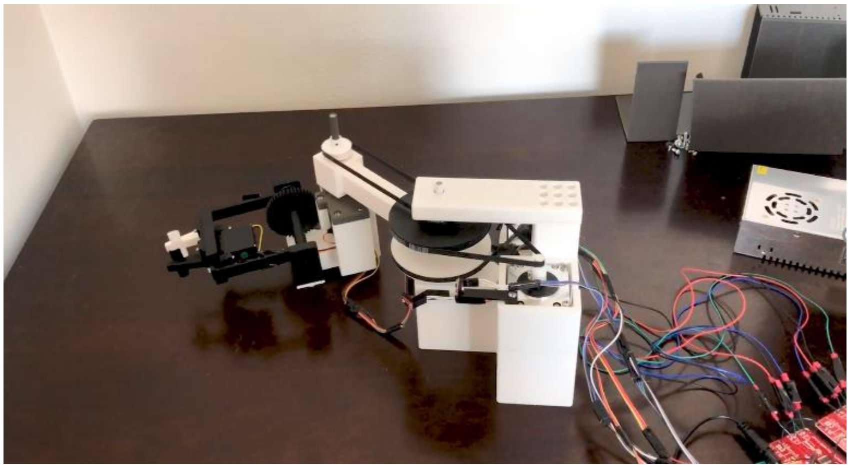


Figure 28: Test 6 Intermediate Arm Pose

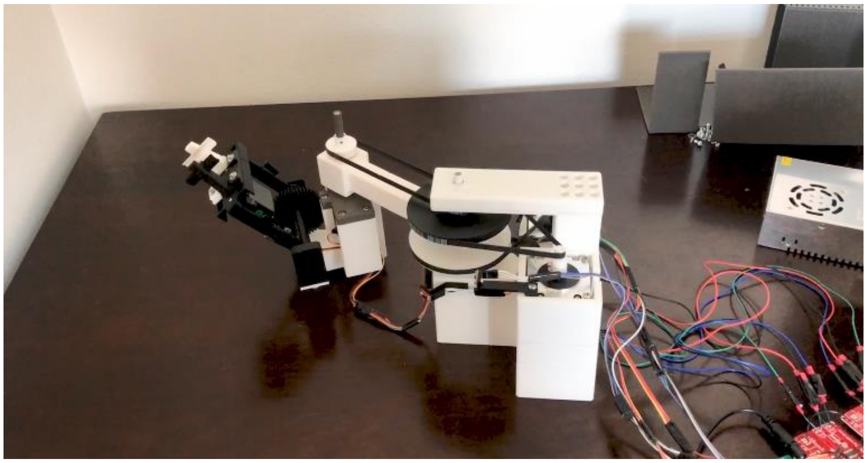


Figure 29: Test 6 Final Arm Pose

4. Costs

4.1 Cost Analysis

4.1.1 Labor

Based on team members' personal experiences and research on the salary of the ECE and CS graduate students in the Urbana-Champaign area and the Bay area [10], we estimate an approximate salary of $50 per hour. Based on the time team members' have spent so far per week on the project, we estimate a workload of twenty hours per week for a total of ten weeks, giving a total labor time spent of two hundred hours per person. We compute the salary per person via the following formula as found on the ECE 445 website [11]:

$$Total\ Salary\ Per\ Person = Salary\ (Hourly) \times Hours \times Multiplier \tag{2}$$

Table 1: Labor Cost Estimate

Member	Salary (Hourly)	Hours	Multiplier	Total Per Person
Bingzhe Wei	$50.00	200	2.5	$25000
Chenghao Duan	$50.00	200	2.5	$25000
Tianhao Chi	$50.00	200	2.5	$25000
			Labor Total	$75000

4.1.2 Parts and Materials

Please see Appendix E.

4.1.3 Grand Total

$$Grand\ Total = Labor\ Cost + Parts\ and\ Materials\ Cost$$

$$= \$75000 + \$435.32 = \$75435.32 \tag{3}$$

4.2 Schedule

Please see Appendix E.

5. Conclusion

5.1 Accomplishments

We are able to successfully convert input images to arbitrary artistic styles, quantize the resulting image with the given palette mentioned in Section 2.2.2, calculate an optimal robot arm path, convert the path into arm angles via inverse kinematics, and send arm positional commands to the control board, which are then translated via the motor boards into joint motion. As shown in our test results above, we can achieve high joint velocity and smooth motion without excessive arm segment deflection. The robot can draw simple shapes, but due to time restrictions, we have not yet finished tuning the robot sufficiently to complete a full painting.

5.2 Uncertainties

Although the robot arm can position accurately at high speed, consecutive small movements with large direction changes as required for drawing induces Arm Segment 2 to vibrate about Joint 2 visibly. This is detrimental to drawing fine details especially with pens with thin, stiff tips that provide less paper contact area as the higher tip stiffness and lower contact pressure results in decreased vibration damping.

5.3 Future Work

There are some improvements that could be made to enhance the robot arm's performance. While the overall movement of the robot arm is smooth, there is still some vibration due to the high accelerations involved. We plan to add joint position sensors and implement close-loop control to dampen these residual actuator vibrations and thus allow higher joint velocities with increased precision. Also, we also plan to improve our motor board by redesigning our motor boards around discrete power MOSFET-based H-bridges, which enable significantly higher power dissipation as the exposed metal tab of TO-220 packages allows external heatsinks to be attached, allowing for higher motor drive currents and increased maximum acceleration, which would aid precision by reducing joint position overshoot.

5.4 Ethical Considerations

It is our responsibility to make sure our project is consistent with the IEEE Code of Ethics [12]. As is consistent with Items 1 and 6, we have implemented mitigation procedures for pinch and temperature hazards in our system via stall detection, overtemperature and overcurrent limits, as well as a case with appropriate isolation for all electronic components, placed appropriate warnings labels/markings, and provided training for users of our system. Since our project mainly just serves an artistic purpose, and is unlikely to be a part of critical infrastructure, our project is very unlikely to harm humanity or society in any form, and thus consistent with Item 9. As a non-sponsored project and with group members not having any particular relevant conflicts of interests, our team fulfills Items 2 and 4. As stated earlier in the introduction, we seek explore and to advance the state of the art of artificial intelligence and its interactions with society, as is consistent with Item 5. Finally, as engineers, all the members of our team have developed this project under a code of utmost integrity, mutual support and nondiscrimination, fulfilling Items 3, 7, 8, and 10.

Appendix A Physical Design Calculations

A.1 Robot Arm Workspace Calculation

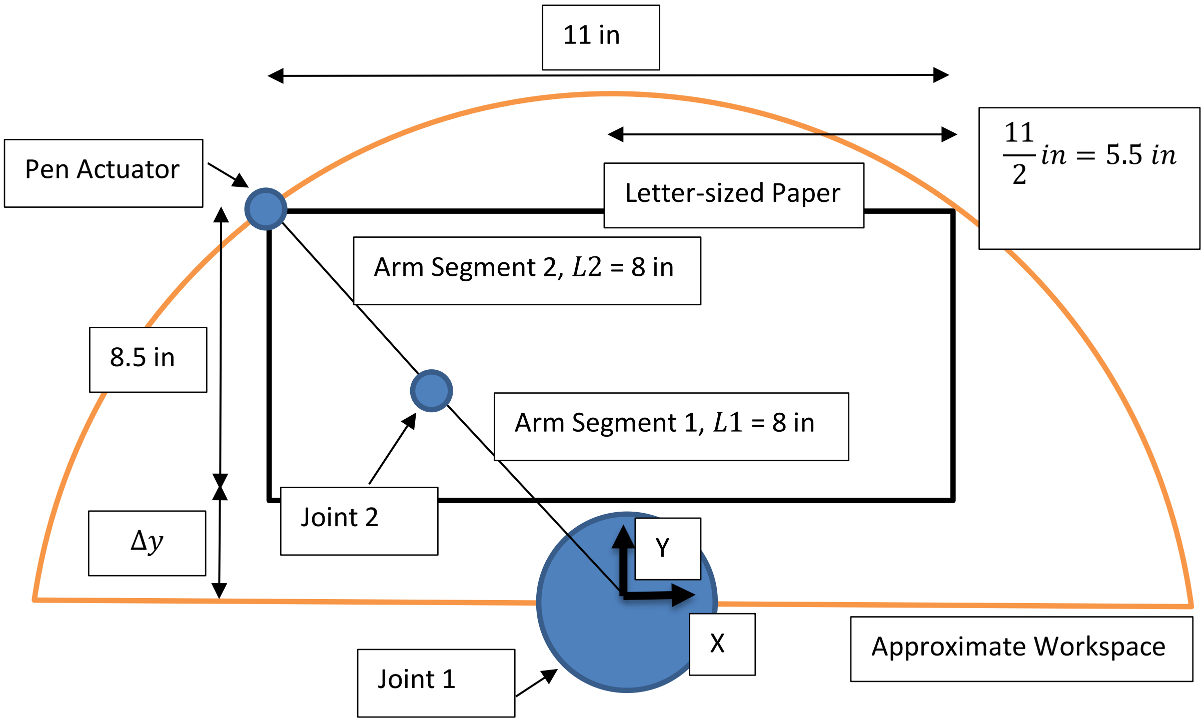


Figure 30: SCARA Robotic Arm Workspace Calculations

The robot's inverse kinematics are defined using the following as shown in Figure 7 below, where E is the elbow angle, the angle of Joint 2, S is the shoulder angle, the angle of Joint 1, $L1$ is the length of arm segment 1, and $L2$ is the length of arm segment 2. Assuming the base of the robot is located at the origin, given the desired (X, Y) coordinates, we can compute the needed Joint 1 angle and Joint 2 angle such the pen can be moved to the desired (X, Y) coordinates.

$$E = \cos^{-1}\left(\frac{X^2+Y^2-(L1)^2-(L2)^2}{2(L1)(L2)}\right) \tag{4}$$

$$S = atan2(Y, X) - \cos^{-1}\left(\frac{X^2+Y^2+(L1)^2-(L2)^2}{2(L1)\sqrt{X^2+Y^2}}\right) \tag{5}$$

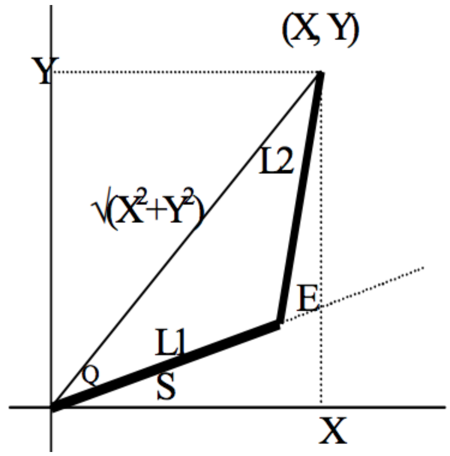


Figure 31: Inverse Kinematics for the SCARA Robotic Arm

We define $atan2(y, x)$ as

$$atan2(y,x) = \begin{cases} \tan^{-1}\left(\frac{y}{x}\right) & x > 0 \\ \tan^{-1}\left(\frac{y}{x}\right) + \pi & x < 0, y \geq 0 \\ \tan^{-1}\left(\frac{y}{x}\right) - \pi & x < 0, y < 0 \\ \frac{\pi}{2} & x = 0, y > 0 \\ -\frac{\pi}{2} & x = 0, y < 0 \\ undefined & x = 0, y = 0 \end{cases} \tag{6}$$

Suppose we place our piece of letter shaped paper as in Figure 6, where the base of robot, and by extension, Joint 1, is located at the origin of the coordinate system. From the above diagram, we have

$$L1 = L2 = 8\ in \tag{7}$$

By the Pythagorean theorem,

$$\Delta y = \sqrt{(8\ in + 8\ in)^2 - (5.5\ in)^2} - 8.5\ in = 6.5\ in \tag{8}$$

Hence, we can bound X and Y, since the paper defines a rectangular domain on the $X - Y$ plane. We have

$$-5.5\ in \leq X \leq 5.5\ in \tag{9}$$

$$6.5\ in = \Delta y \leq Y \leq 8.5\ in + \Delta y = 8.5\ in + 6.5\ \text{in} = 15\ \text{in} \tag{10}$$

Now, this implies that we have

$$0\ in^2 \leq X^2 \leq 30.25\ in^2 \tag{11}$$

$$42.25\ in^2 \leq Y^2 \leq 225\ \text{in}^2 \tag{12}$$

and from the subexpression to calculate Joint 2 angle

$$\frac{X^2+Y^2-(L1)^2-(L2)^2}{2(L1)(L2)} \tag{13}$$

we have, for possible boundary value of the subexpression

$$\frac{0\ in^2+42.25\ in^2-(8\ in)^2-(8\ in)^2}{2(8\ in)(8\ in)} = -0.670 \tag{14}$$

$$\frac{30.25\ in^2+42.25\ in^2-(8\ in)^2-(8\ in)^2}{2(8\ in)(8\ in)} = -0.434 \tag{15}$$

$$\frac{0\ in^2+225\ in^2-(8\ in)^2-(8\ in)^2}{2(8\ in)(8\ in)} = 0.758 \tag{16}$$

$$\frac{30.25\ in^2+225\ in^2-(8\ in)^2-(8\ in)^2}{2(8\ in)(8\ in)} = 0.994 \tag{17}$$

Hence the above subexpression, Equation 9, is bounded as follows, as $X^2 + Y^2$ gives a function that does not change in concavity and has no saddle points

$$-0.670 \leq \frac{X^2+Y^2-(L1)^2-(L2)^2}{2(L1)(L2)} \leq 0.994 \tag{18}$$

Since the domain of $\cos^{-1}(x)$ is $x \in [-1, 1]$, we see that for all points on the paper, there is a solution to the Joint 2 angle. Similarly, for the Joint 1 angle, we have the subexpression

$$\frac{X^2+Y^2+(L1)^2-(L2)^2}{2(L1)\sqrt{X^2+Y^2}} = \frac{X^2+Y^2+(8\ in)^2-(8\ in)^2}{2(8\ in)\sqrt{X^2+Y^2}} = \frac{1}{16\ in}\sqrt{X^2+Y^2} \tag{19}$$

we have, for possible boundary value of the subexpression

$$\frac{0\ in^2+42.25\ in^2+(8\ in)^2-(8\ in)^2}{2(8\ in)\sqrt{0\ in^2+42.25\ in^2}} = 0.406 \tag{20}$$

$$\frac{30.25\ in^2+42.25\ in^2+(8\ in)^2-(8\ in)^2}{2(8\ in)\sqrt{30.25\ in^2+42.25\ in^2}} = 0.532 \tag{21}$$

$$\frac{0\ in^2+225\ in^2+(8\ in)^2-(8\ in)^2}{2(8\ in)\sqrt{0\ in^2+225\ in^2}} = 0.938 \tag{22}$$

$$\frac{30.25\ in^2+225\ in^2+(8\ in)^2-(8\ in)^2}{2(8\ in)\sqrt{30.25\ in^2+225\ in^2}} = 0.999 \tag{23}$$

Hence the above subexpression, Equation 9, is bounded as follows, as $\sqrt{X^2 + Y^2}$ gives a function that does not change in concavity and has no saddle points

$$0.406 \leq \frac{X^2+Y^2-(L1)^2-(L2)^2}{2(L1)(L2)} \leq 0.999 \tag{24}$$

Since the domain of $\cos^{-1}(x)$ is $x \in [-1, 1]$ and the domain of $atan2(y, x)$ is all values of x and y except $x = 0, y = 0$, we see that for all points on the paper, there is a solution to the Joint 1 angle.

Hence, we see that with each arm segment being eight inches long, there is sufficient space to place centrally in a relatively easy to reach portion of the workspace a piece of Letter-sized paper, provided the piece of Letter-sized paper is placed as discussed above. Meanwhile, as the pen actuator only needs a very small amount of z-travel on the order of millimeters, we can approximate our pen actuator as a linear actuator via the small angle approximation for sine, and as we are using a Cartesian coordinate system, this implies that our robotic should be able to reach any desired value of z as long as the value of z is such that the small angle approximation for sine is valid, since both the axis and the joint are of the same nature, that is, linearly varying.

A.2 Motor Torque Calculations

To determine the stepper motor torque needed, we researched some industrial robotic arms and decided to target a joint velocity of at least $100\ °/s$, reached in $2s$ or less from zero angular velocity. Assuming that we start from zero angular velocity and do not attempt to brake after reaching the target velocity, we have, where ω is the angular velocity about the rotational axis and α is the angular acceleration about the rotational axis

$$\omega = \alpha t \tag{25}$$

Since torque is

$$\tau = I\alpha \tag{26}$$

where I is the moment of inertia about the desired rotational axis, we have

$$\tau = \frac{I\omega}{t} \tag{27}$$

From Autodesk Inventor, the rotational inertia of the arm segment about the rotational axis is $I = 9.845\ lb \cdot in$. Thus, the required torque is

$$\tau = \frac{9.845\ lb{\cdot}in{\cdot}100\frac{deg}{s}{\cdot}\frac{2\pi}{360\ deg}}{2\ s} \cdot \frac{1\ N}{0.22481\ lb} \cdot \frac{0.0254\ m}{1\ in} = 0.971\ N \cdot m \tag{28}$$

which we round to approximately $1\ N \cdot m$ to allow for some part variation tolerance.

A.3 Modal Analysis Results

From the simulations below, we can see that the structure as designed has sufficient structural integrity for the anticipated loads and forces, evident from the very low displacement and high safety factor of the structure. To analyze vibration stability, we first convert our targeted joint velocity to frequency f via the following formula, where all other symbols are as defined below, and obtain

$$f = \frac{\omega}{2\pi} = \frac{100\frac{deg}{s}{\cdot}\frac{2\pi}{360\ deg}}{2\pi} = 0.278\ Hz \tag{29}$$

Hence, modal analysis shows that all resonance frequencies are at least three times higher compared to the joint angular velocity we are designing for and thus vibrations will be minimal.

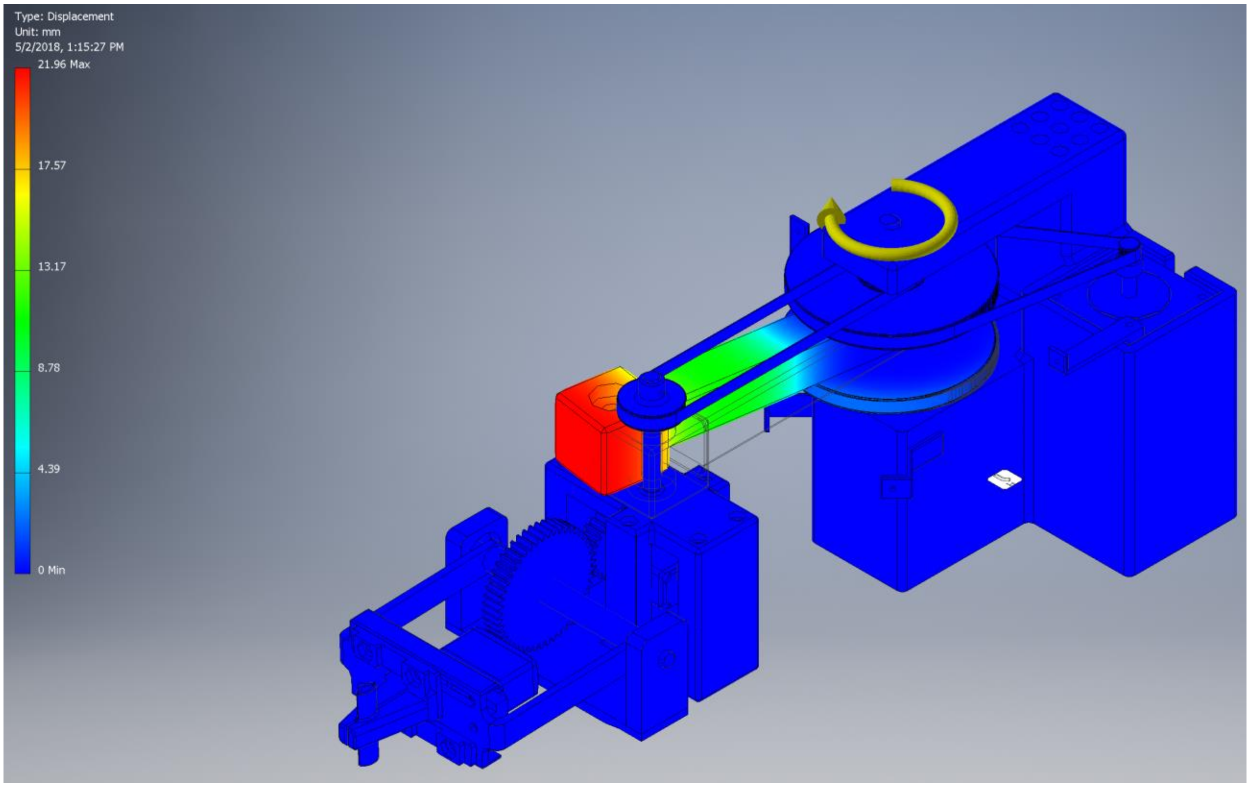


Figure 32: Robot Arm Static Analysis - Displacement

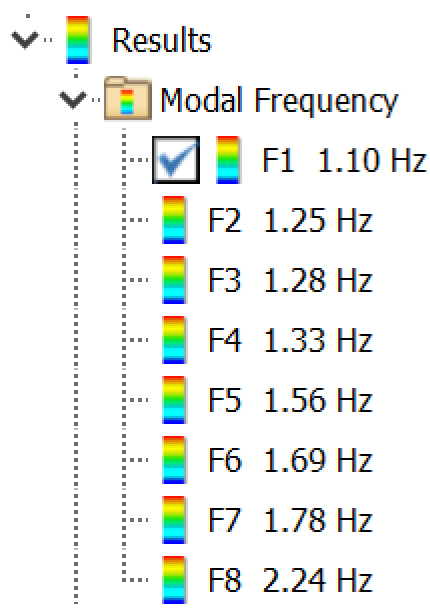


Figure 33: Robot Arm Modal Analysis – List of Modes

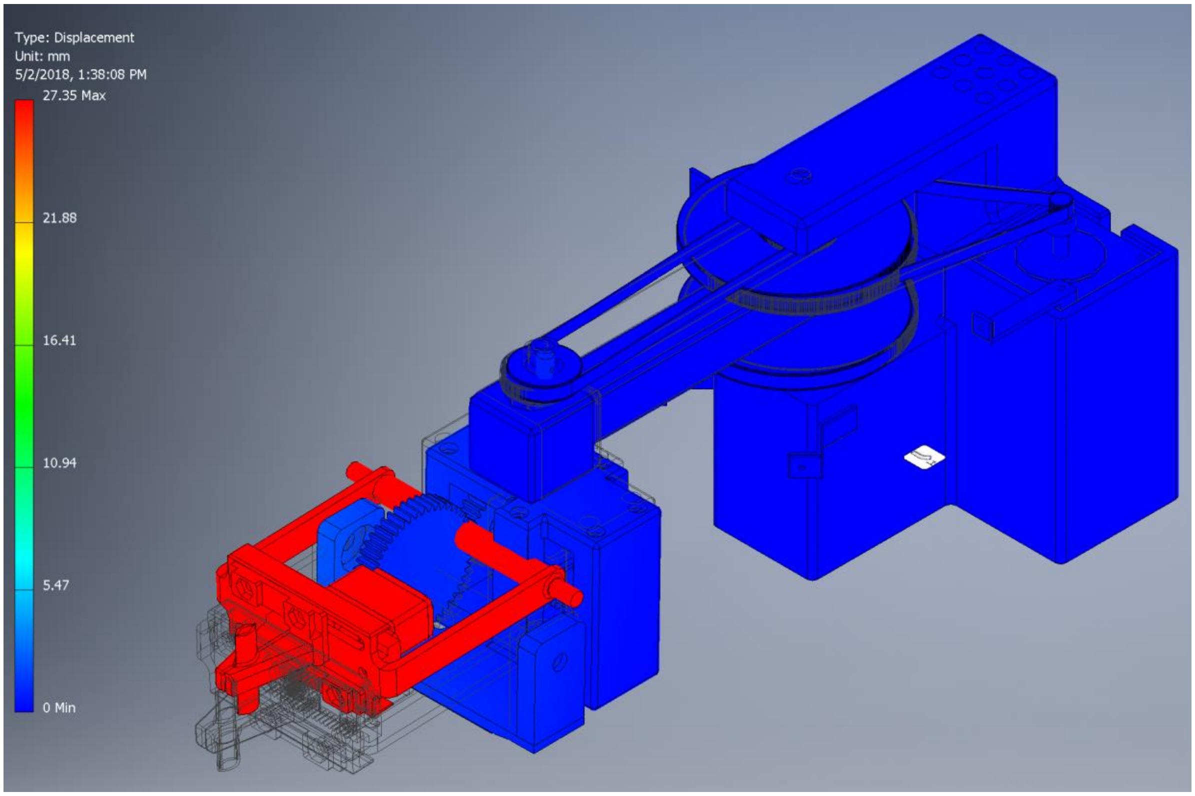


Figure 34: Robot Arm Modal Analysis – Mode 1

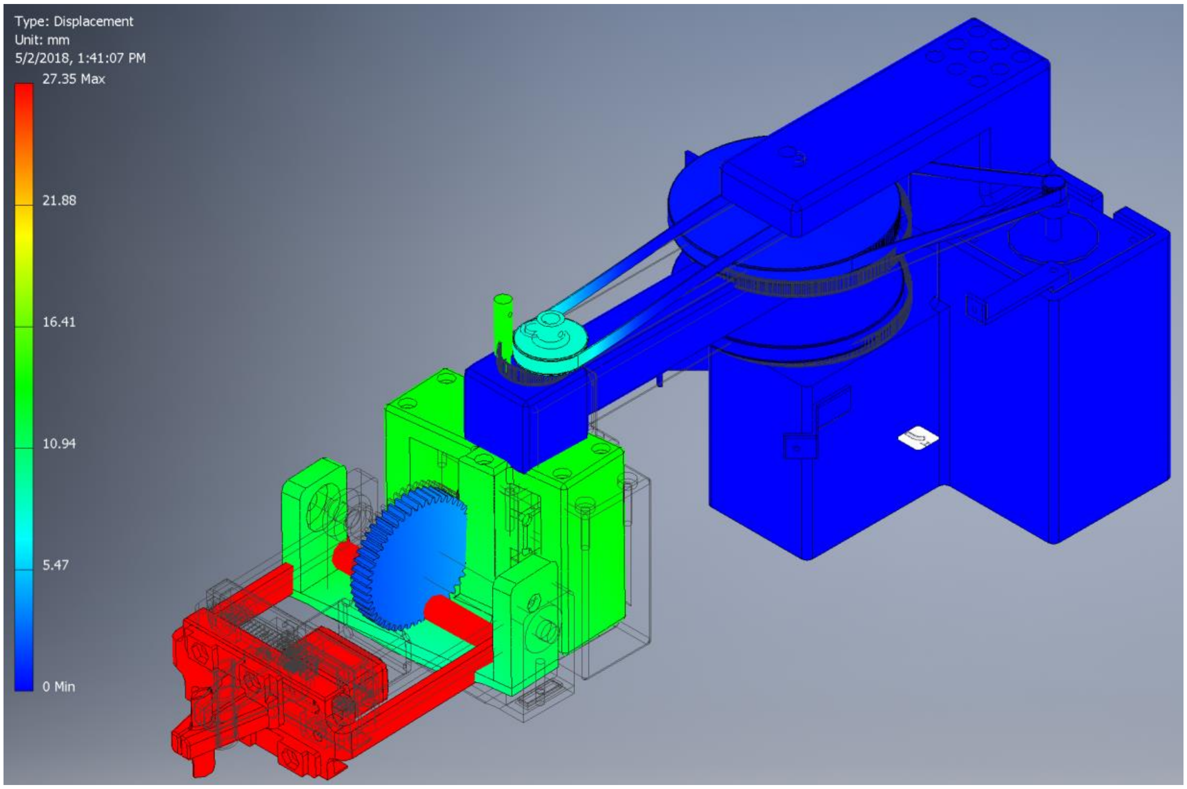


Figure 35: Robot Arm Modal Analysis – Mode 2

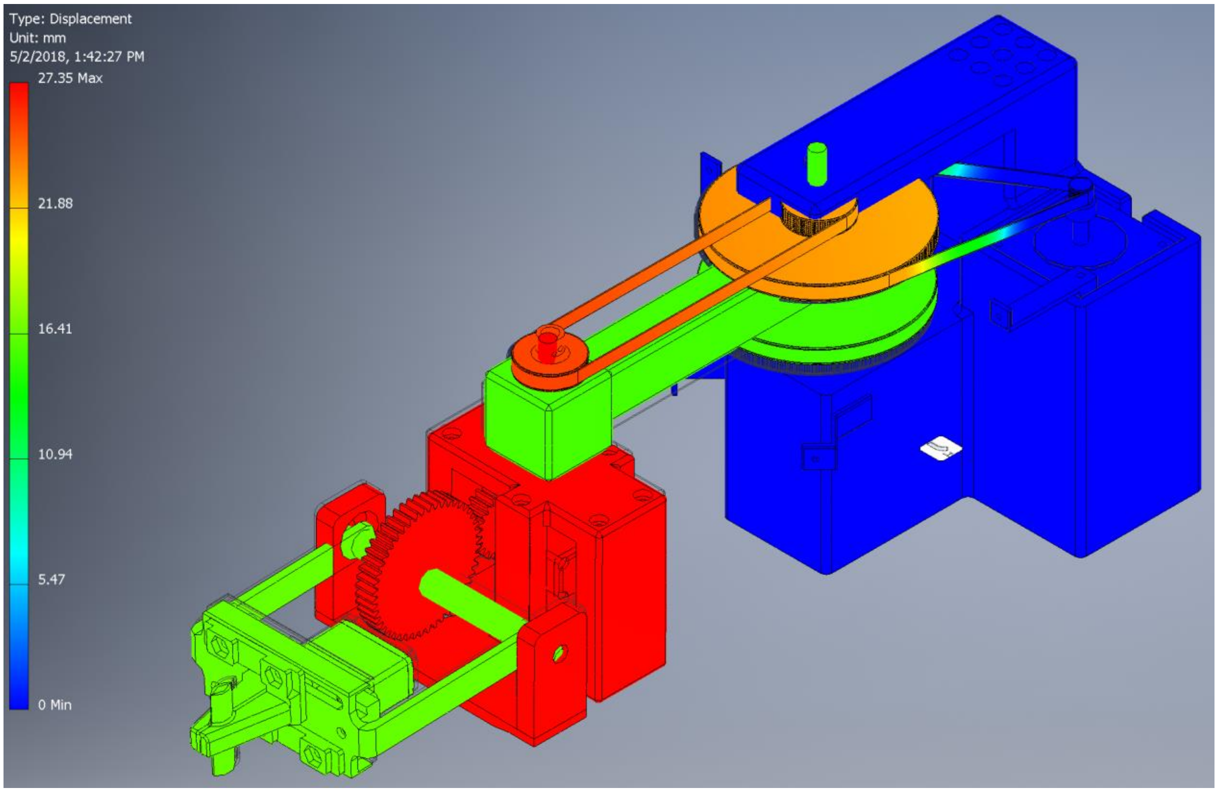


Figure 36: Robot Arm Modal Analysis – Mode 3

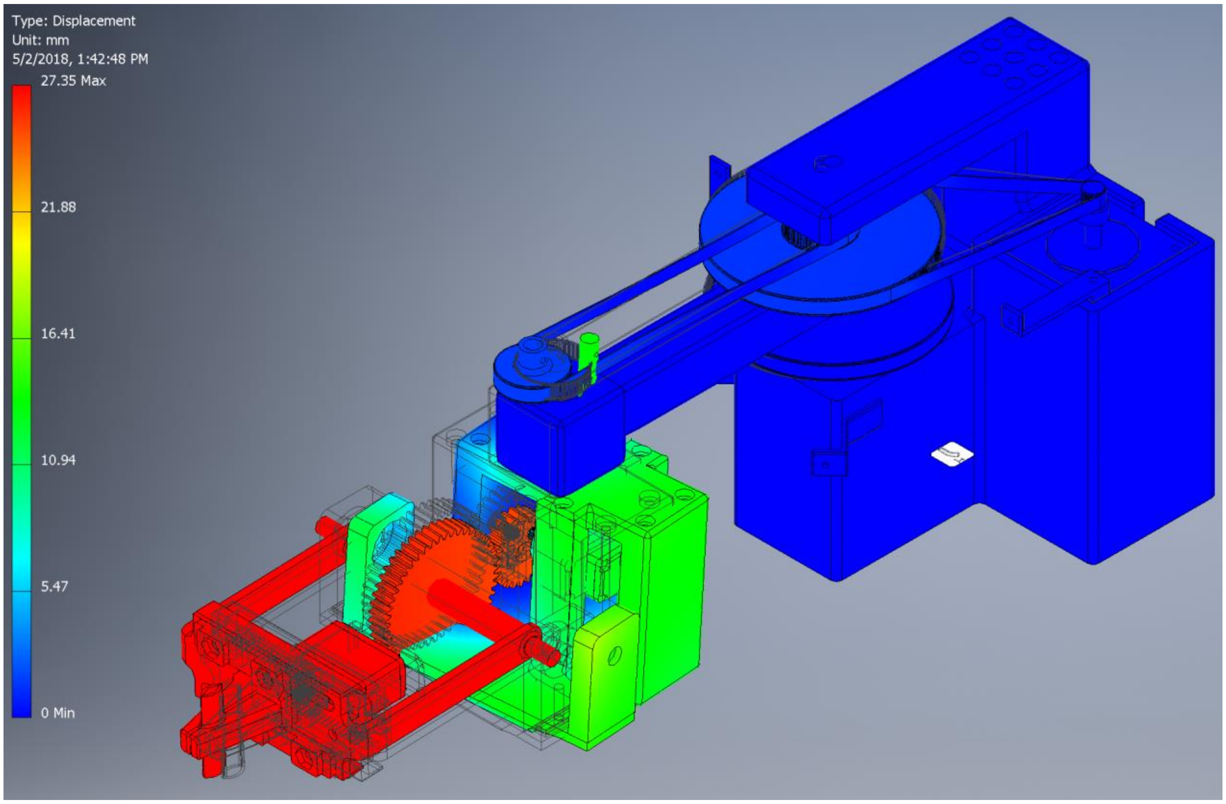


Figure 37: Robot Arm Modal Analysis – Mode 4

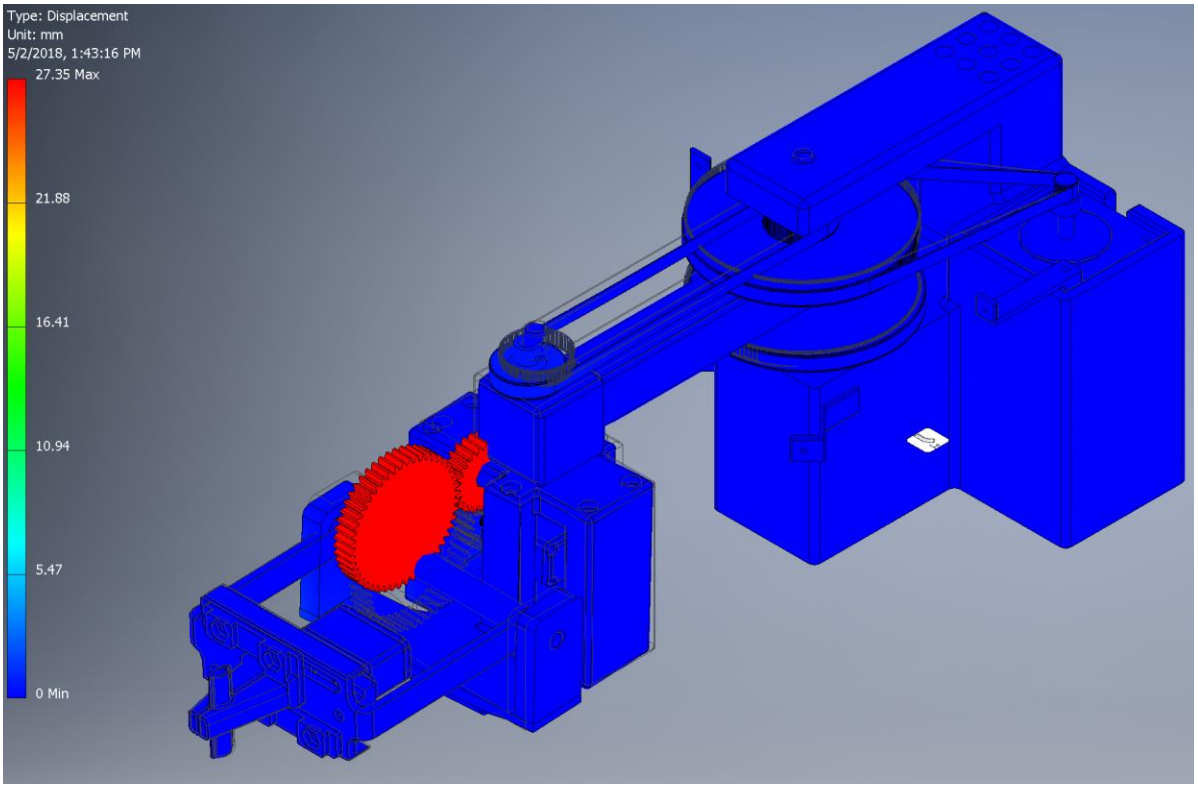


Figure 38: Robot Arm Modal Analysis – Mode 5

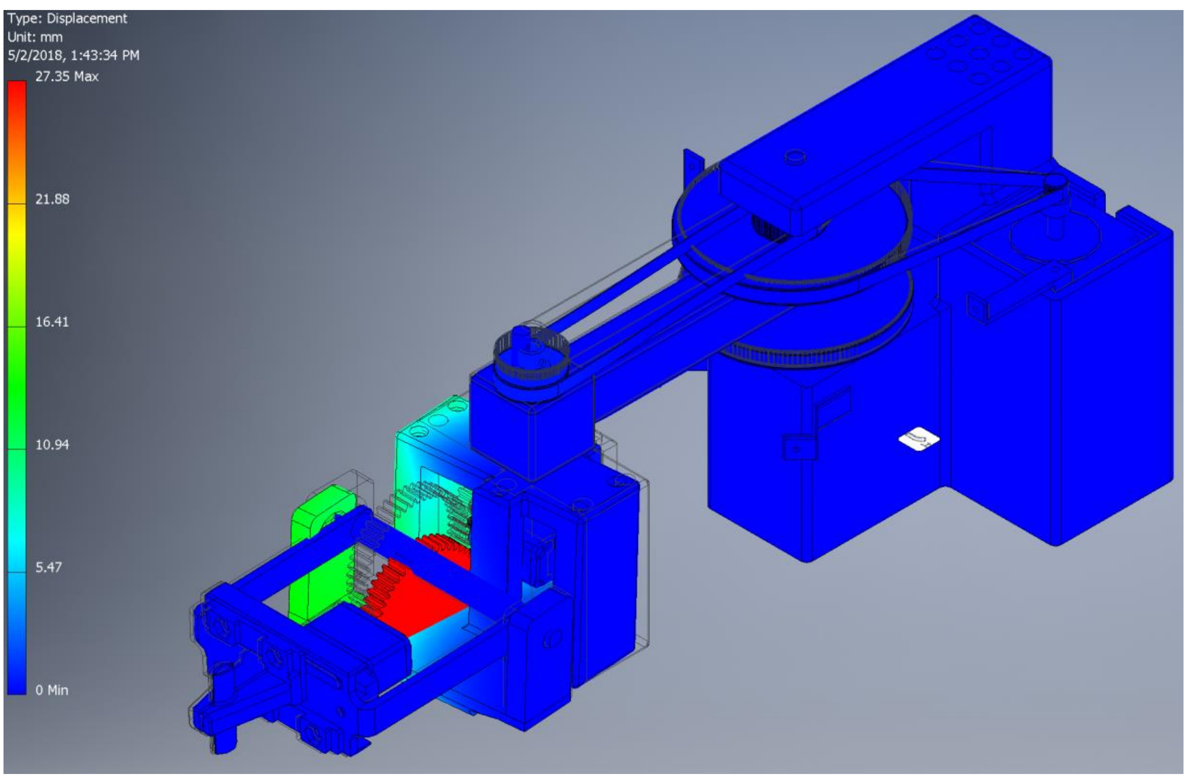


Figure 39: Robot Arm Modal Analysis – Mode 6

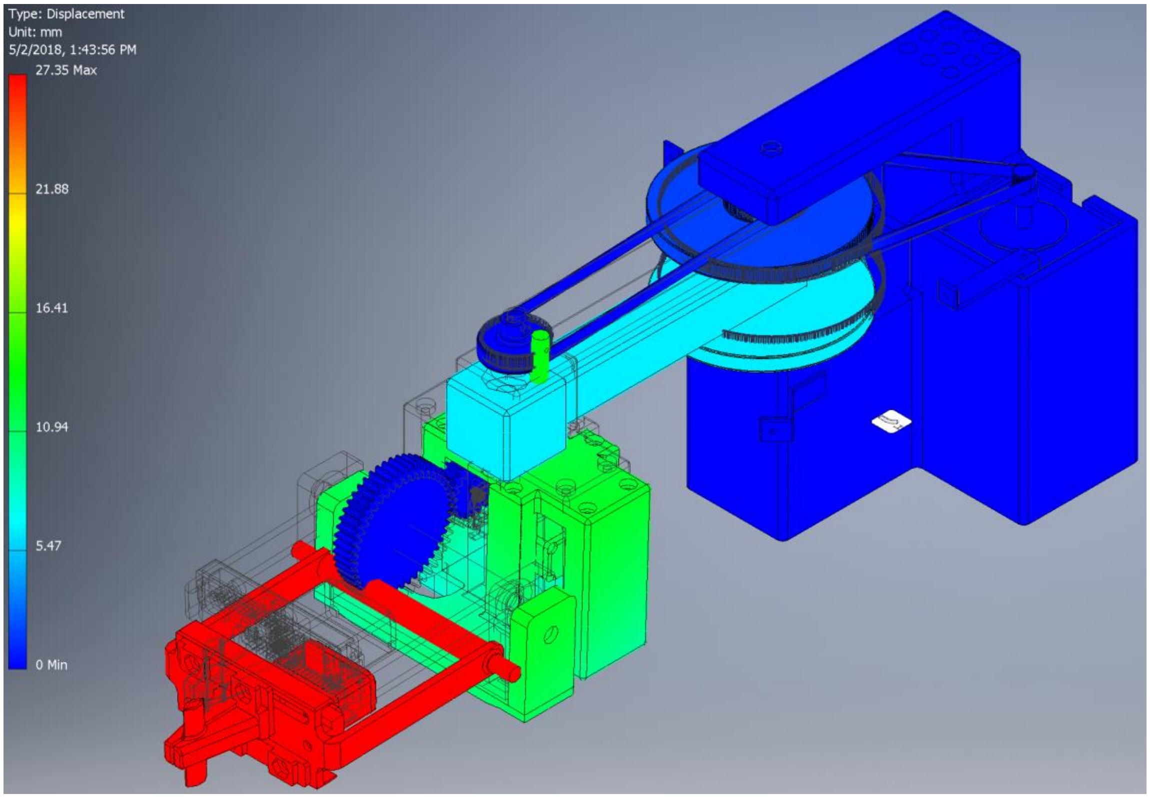


Figure 40: Robot Arm Modal Analysis – Mode 7

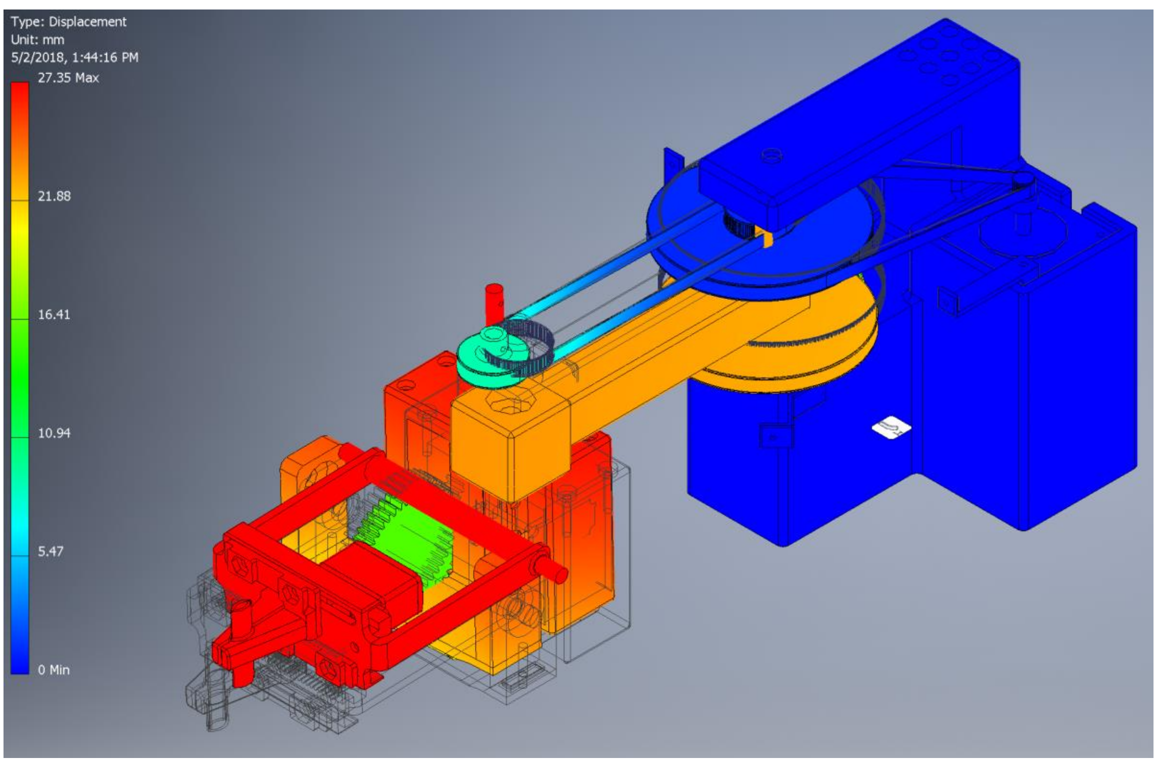


Figure 41: Robot Arm Modal Analysis – Mode 8

A.4 Tolerance Analysis

The most significant risk to the successful completion of this project is the mechanical design of the SCARA robotic arm, in particular, we require accurate positioning to produce a high-quality output with true edges instead of wavy lines. Using the same setup as in Section 2.3.1, the inverse kinematic equations for the SCARA robotic arm are as follows:

$$E = \cos^{-1}\left(\frac{X^2+Y^2-(L1)^2-(L2)^2}{2(L1)(L2)}\right) \tag{30}$$

$$S = atan2(Y,X) - \cos^{-1}\left(\frac{X^2+Y^2+(L1)^2-(L2)^2}{2(L1)\sqrt{X^2+Y^2}}\right) \tag{31}$$

To attain a resolution of 200 px x 200 px, each pixel is spaced $\frac{8.5}{200}\ in = 0.0425\ in$ apart in the y-axis and $\frac{11}{200}\ in = 0.055\ in$ apart in the x-axis, hence, the maximum permissible error in X, ΔX, is $\Delta X = \frac{\frac{11}{200}in}{2} = 0.0275\ in$ and the maximum permissible error in Y, ΔY, is $\Delta Y = \frac{\frac{8.5}{200}in}{2} = 0.0213\ in$. From [13], using differentials, we have

$$\Delta E = \frac{dE}{dX}\Delta X \tag{32}$$

where ΔE is the permissible error in Joint 2 angle for the given permissible error in X, ΔX.

$$\Delta E = \frac{dE}{dY}\Delta Y \tag{33}$$

where ΔE is the permissible error in Joint 2 angle for the given permissible error in Y, ΔY.

$$\Delta S = \frac{dS}{dX}\Delta X \tag{34}$$

where ΔS is the permissible error in Joint 1 angle for the given permissible error in X, ΔX.

$$\Delta S = \frac{dE}{dY}\Delta Y \tag{35}$$

where ΔS is the permissible error in Joint 1 angle for the given permissible error in Y, ΔY. We have

$$\frac{dE}{dX} = -\frac{\mathrm{X}}{(L1)(L2)\sqrt{1-\frac{\left(X^2+Y^2-(L1)^2-(L2)^2\right)^2}{4(L1)^2(L2)^2}}} \tag{36}$$

$$\frac{dE}{dY} = -\frac{\mathrm{Y}}{(L1)(L2)\sqrt{1-\frac{\left(X^2+Y^2-(L1)^2-(L2)^2\right)^2}{4(L1)^2(L2)^2}}} \tag{37}$$

$$\frac{dS}{dX} = -\frac{Y}{X^2+Y^2} + \frac{\frac{X}{L1\sqrt{X^2+Y^2}} - \frac{X(X^2+Y^2+(L1)^2-(L2)^2)}{2(L1)(X^2+Y^2)^{\frac{3}{2}}}}{\sqrt{1-\frac{\left(X^2+Y^2+(L1)^2-(L2)^2\right)^2}{4(L1)^2(X^2+Y^2)}}} \tag{38}$$

$$\frac{dS}{dX} = -\frac{X}{X^2+Y^2} + \frac{\frac{Y}{L1\sqrt{X^2+Y^2}} - \frac{Y(X^2+Y^2+(L1)^2-(L2)^2)}{2(L1)(X^2+Y^2)^{\frac{3}{2}}}}{\sqrt{1-\frac{\left(X^2+Y^2+(L1)^2-(L2)^2\right)^2}{4(L1)^2(X^2+Y^2)}}} \tag{39}$$

therefore

$$\Delta E = -\frac{\mathrm{X}}{(L1)(L2)\sqrt{1-\frac{\left(X^2+Y^2-(L1)^2-(L2)^2\right)^2}{4(L1)^2(L2)^2}}}\Delta X \tag{40}$$

$$\Delta E = -\frac{\mathrm{Y}}{(L1)(L2)\sqrt{1-\frac{\left(X^2+Y^2-(L1)^2-(L2)^2\right)^2}{4(L1)^2(L2)^2}}}\Delta Y \tag{41}$$

$$\Delta S = \left(-\frac{Y}{X^2+Y^2} + \frac{\frac{X}{L1\sqrt{X^2+Y^2}} - \frac{X(X^2+Y^2+(L1)^2-(L2)^2)}{2(L1)(X^2+Y^2)^{\frac{3}{2}}}}{\sqrt{1-\frac{\left(X^2+Y^2+(L1)^2-(L2)^2\right)^2}{4(L1)^2(X^2+Y^2)}}} \right)\Delta X \tag{42}$$

$$\Delta S = \left(-\frac{X}{X^2+Y^2} + \frac{\frac{Y}{L1\sqrt{X^2+Y^2}} - \frac{Y(X^2+Y^2+(L1)^2-(L2)^2)}{2(L1)(X^2+Y^2)^{\frac{3}{2}}}}{\sqrt{1-\frac{\left(X^2+Y^2+(L1)^2-(L2)^2\right)^2}{4(L1)^2(X^2+Y^2)}}} \right)\Delta Y \tag{43}$$

Using the bounds on X and Y, i.e. Equations 6-9, we get

Boundary Condition	Maximum Allowable ΔS (Joint 1 Error) for Given ΔX	Maximum Allowable ΔS (Joint 1 Error) for Given ΔY	Maximum Allowable ΔE (Joint 2 Error) for Given ΔX	Maximum Allowable ΔE (Joint 2 Error) for Given ΔY
$X = -5.5\ in, Y = 15\ in$	-0.0125 rad	0.0236 rad	0.0219 rad	-0.0462 rad
$X = 5.5\ in, Y = 6.5\ in$	-0.00117 rad	-0.000408 rad	-0.00262 rad	-0.00240 rad
$X = 5.5\ in, Y = 15\ in$	0.00931 rad	0.0226 rad	-0.0219 rad	-0.0462 rad
$X = -5.5\ in, Y = 6.5\ in$	$-0.0376\ rad$	$0.00282\ rad$	$0.00262\ rad$	$-0.00240\ rad$

From the above, the tolerance required is bounded by the smallest $|\Delta S|$ or $|\Delta E|$ found previously, i.e. $maximum\ angular\ error \leq 0.00424\ \mathrm{rad}$, as E and S do not change in concavity and has no saddle points, and hence neither do the derivatives of E and S change in concavity and have saddle points. The 23HS22-2804S stepper motors used have 200 steps per revolution and the L6470 is capable for 1/128 microstepping, i.e. 128 microsteps per motor full step. Additionally, as can be seen from the Figure 2, Arm Segment 1 is driven by a 1:3 ratio synchronous belt drive, while Arm Segment 2 is driven by two 1:3 ratio synchronous belt drive in series. Hence, the angular accuracy of Joint 1 is $\frac{2\pi\ rad}{200} \cdot \frac{1}{128} \cdot \frac{1}{3} = 8.18 \cdot 10^{-5}\ rad$ and the angular accuracy of Joint 2 $\frac{2\pi\ rad}{200} \cdot \frac{1}{128} \cdot \frac{1}{3} \cdot \frac{1}{3} = 2.73 \cdot 10^{-5}\ rad$, both of which fulfill the required angular accuracy. Although microstepping will result in a loss of torque, based on our design criterion, there is sufficient margin to ensure that any torque loss will not result in loss of functionality.

A.5 Claw Mechanism Details

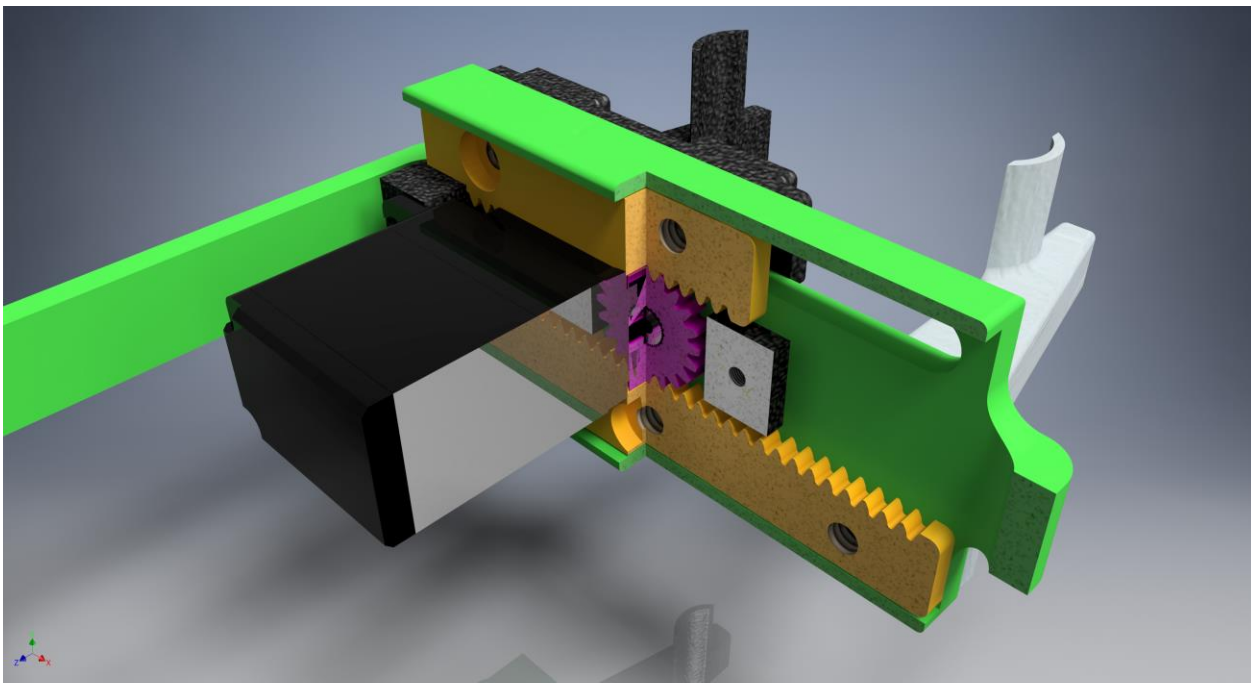


Figure 42: Claw Open

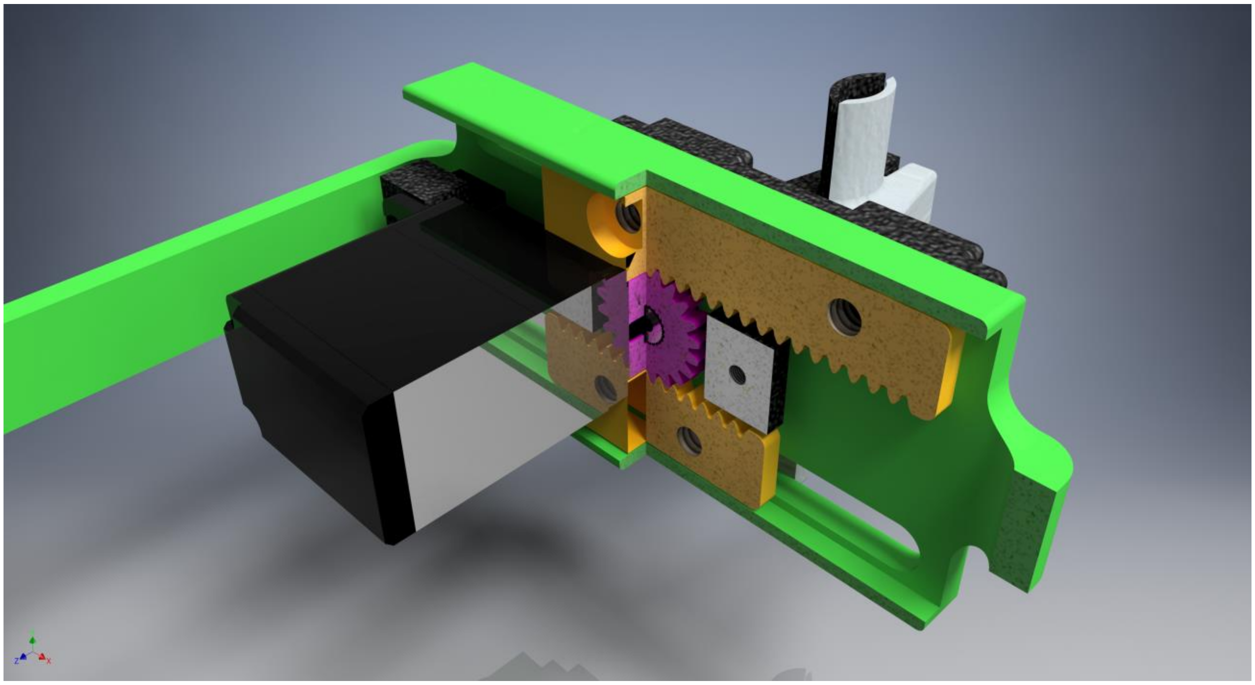


Figure 43: Claw Closed

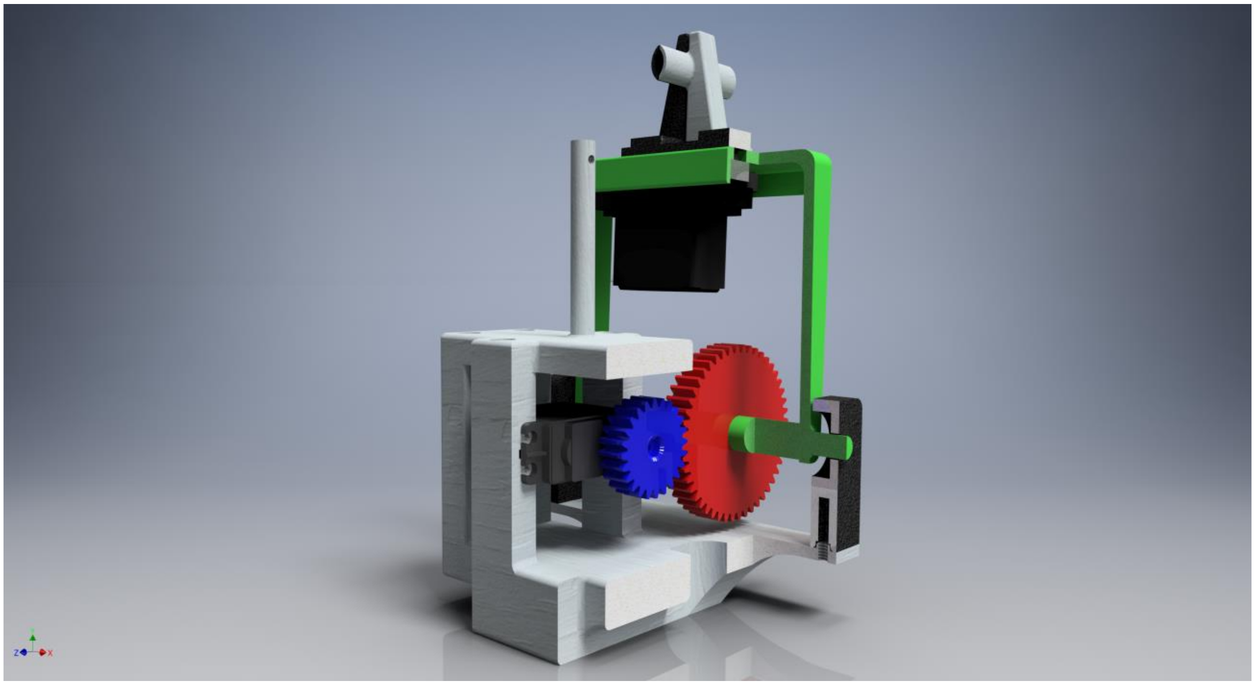


Figure 44: Claw Raised

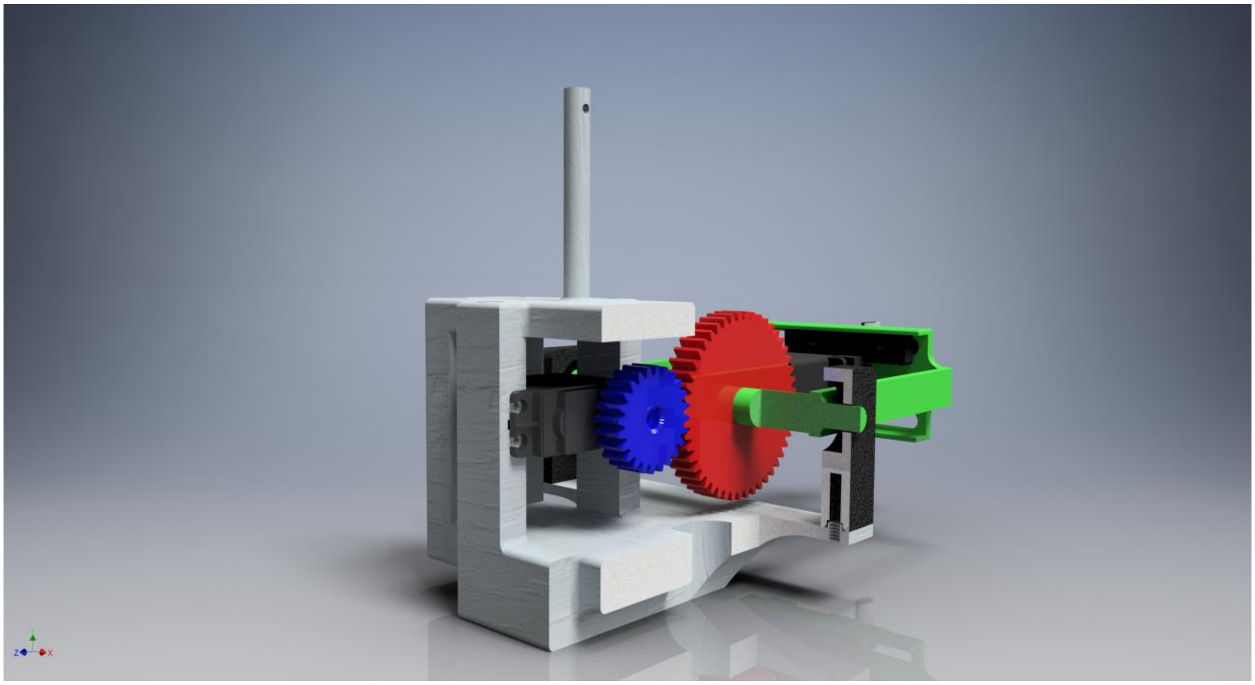
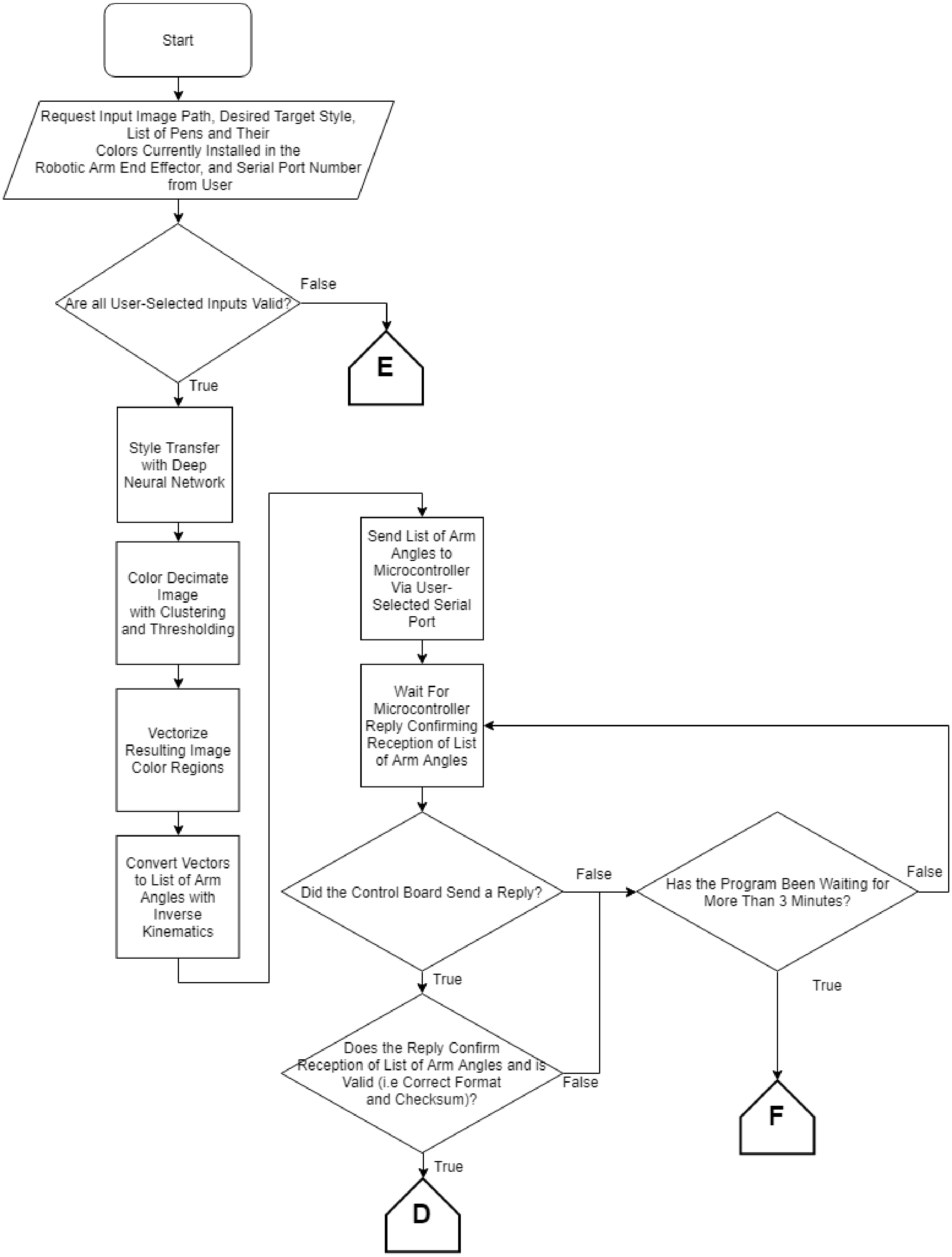


Figure 45: Claw Lowered

Appendix B Software Logic Flowcharts



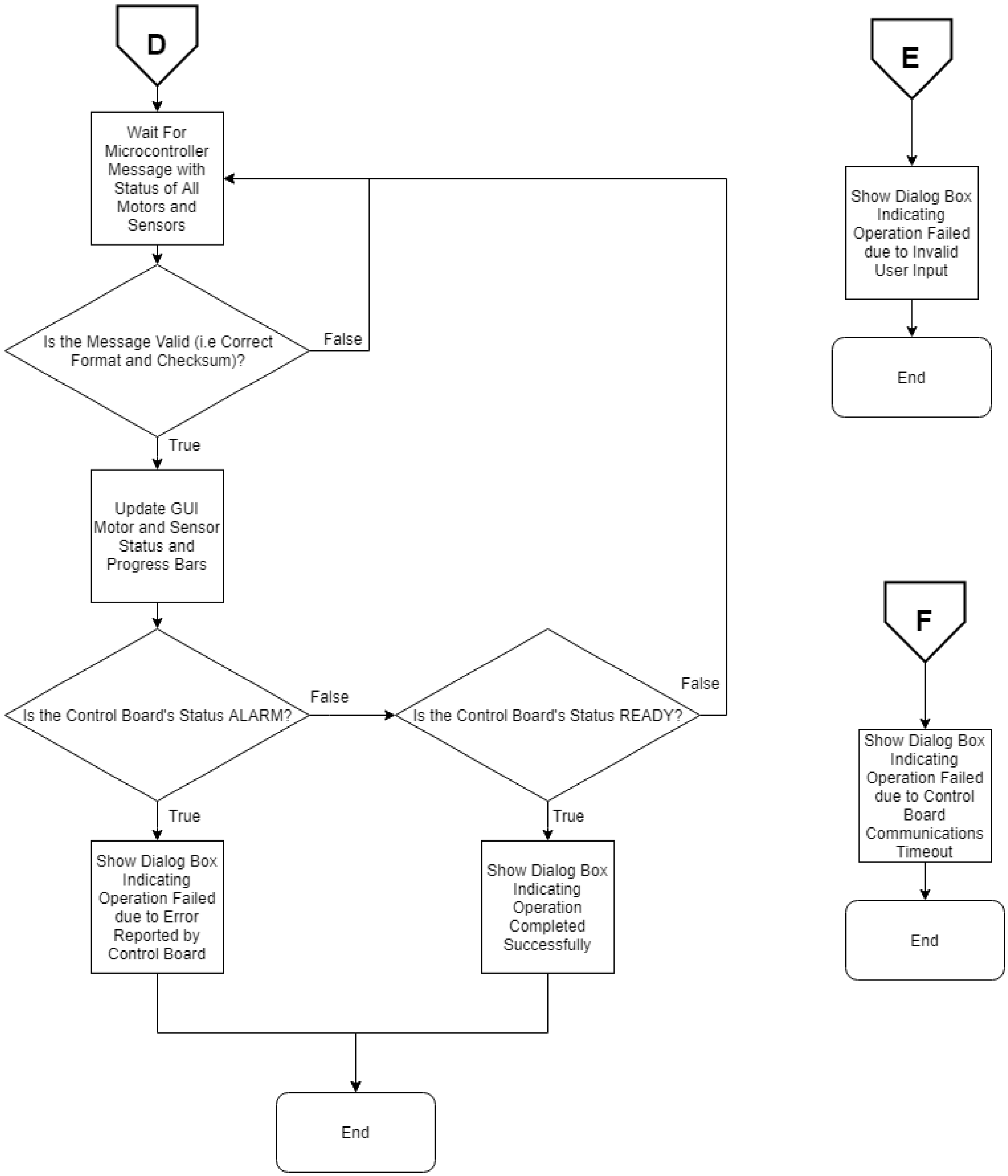


Figure 46: PC Software Logic Block Diagram

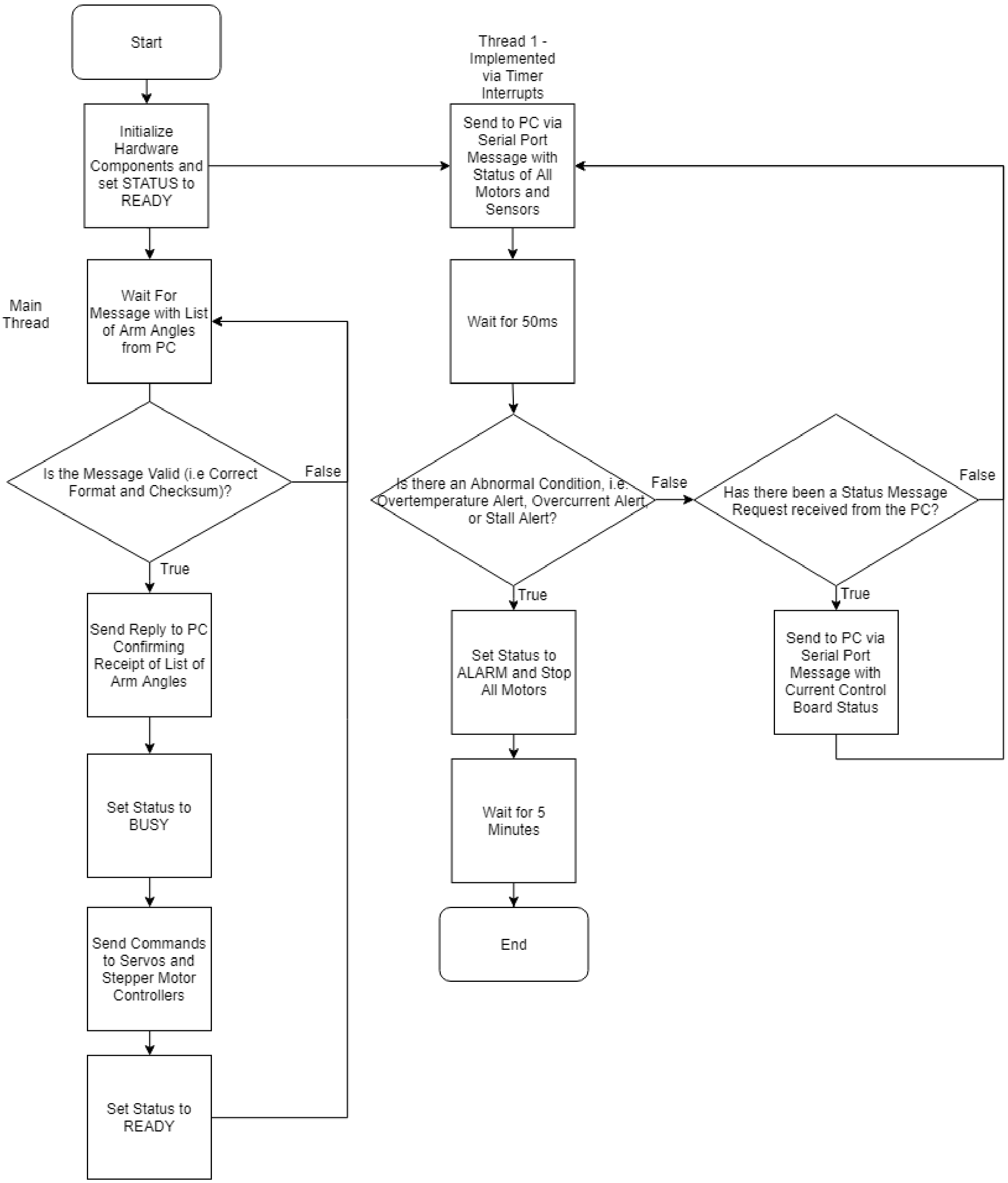


Figure 47: Microcontroller Software Logic Block Diagram

Appendix C Control Board and Motor Board Schematics and PCB Designs

C.1 Control Board Schematic and PCB Design

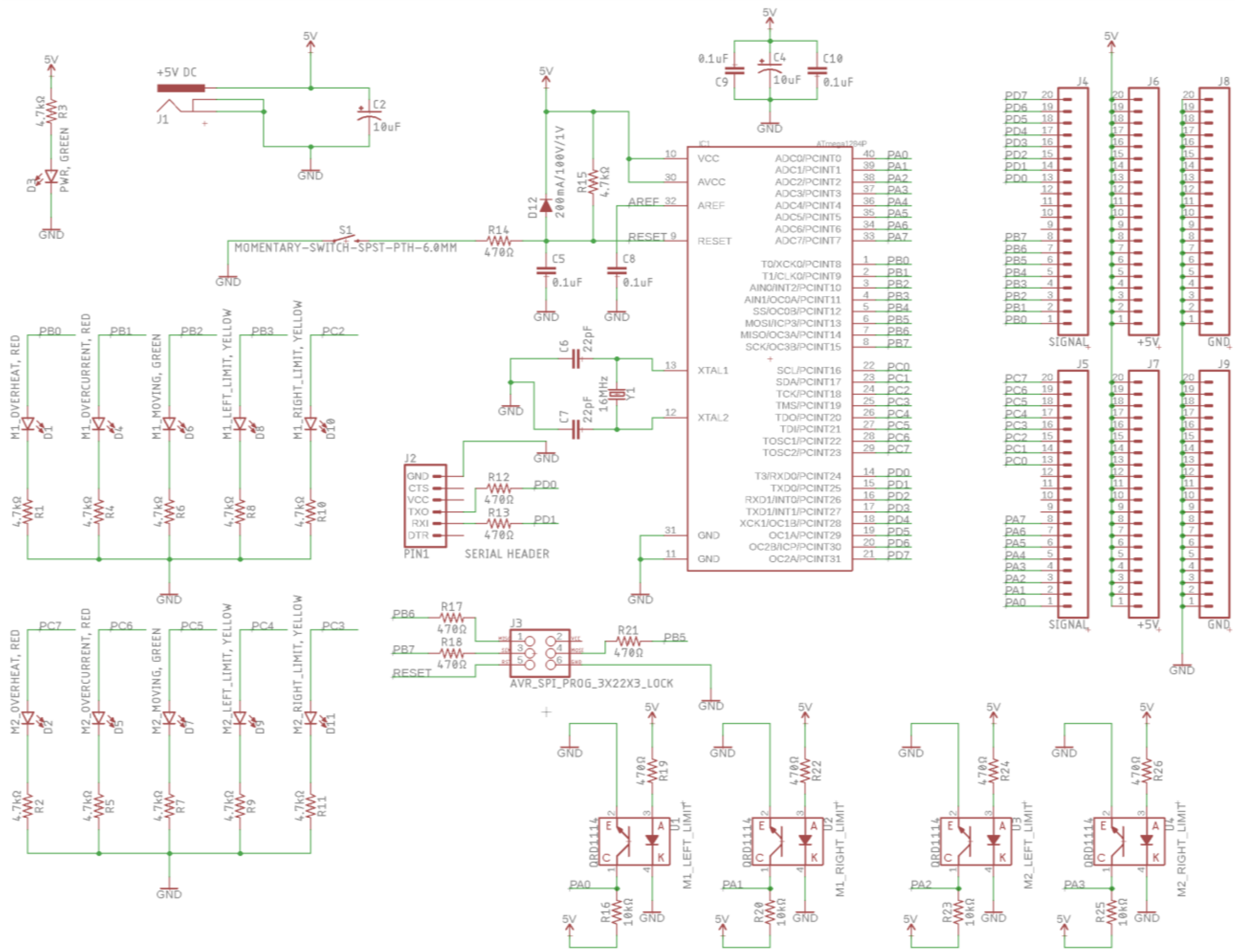


Figure 48: Control Board Circuit Schematic

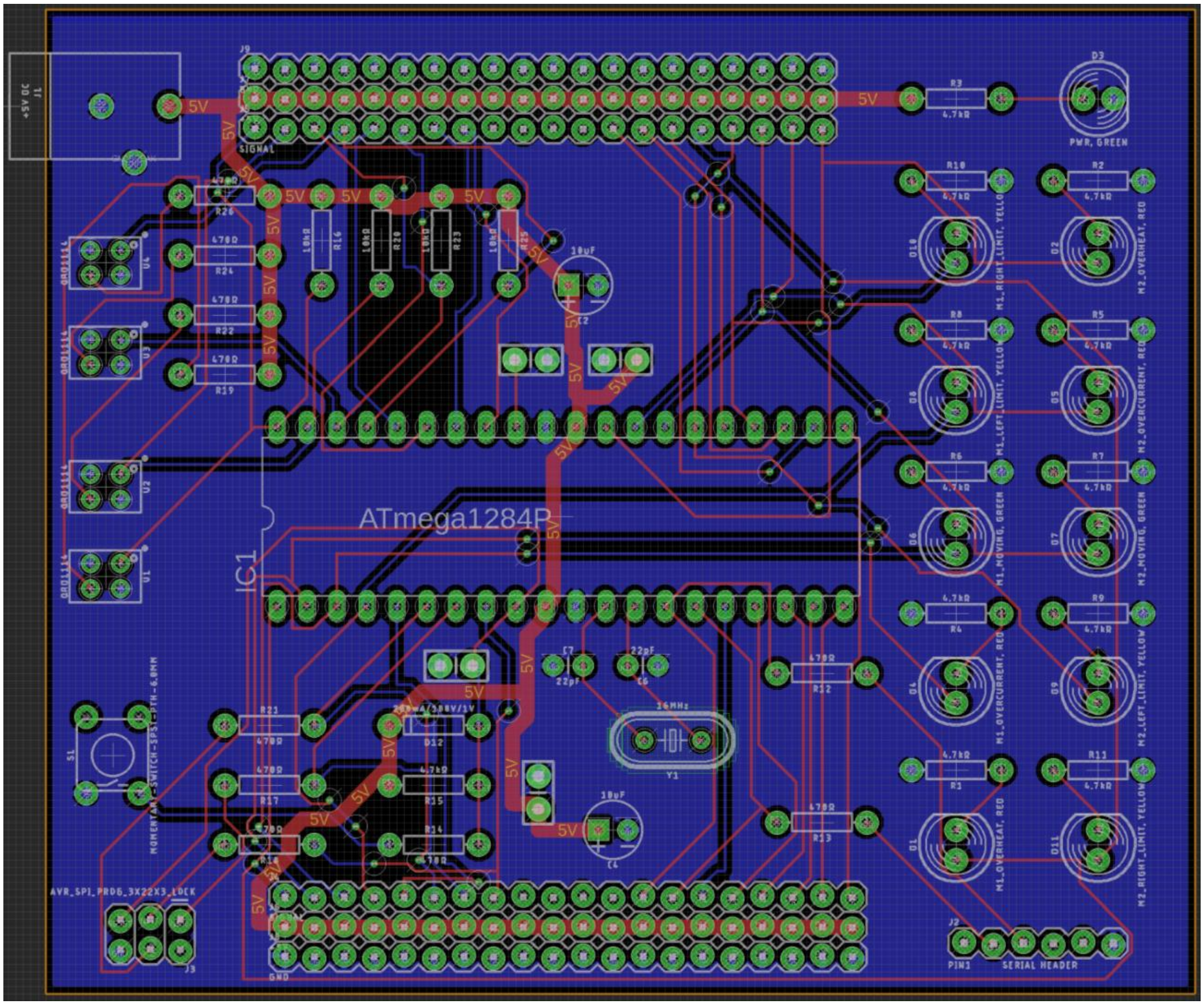


Figure 49: Control Board PCB Design

C.2 Motor Board Schematic and PCB Design

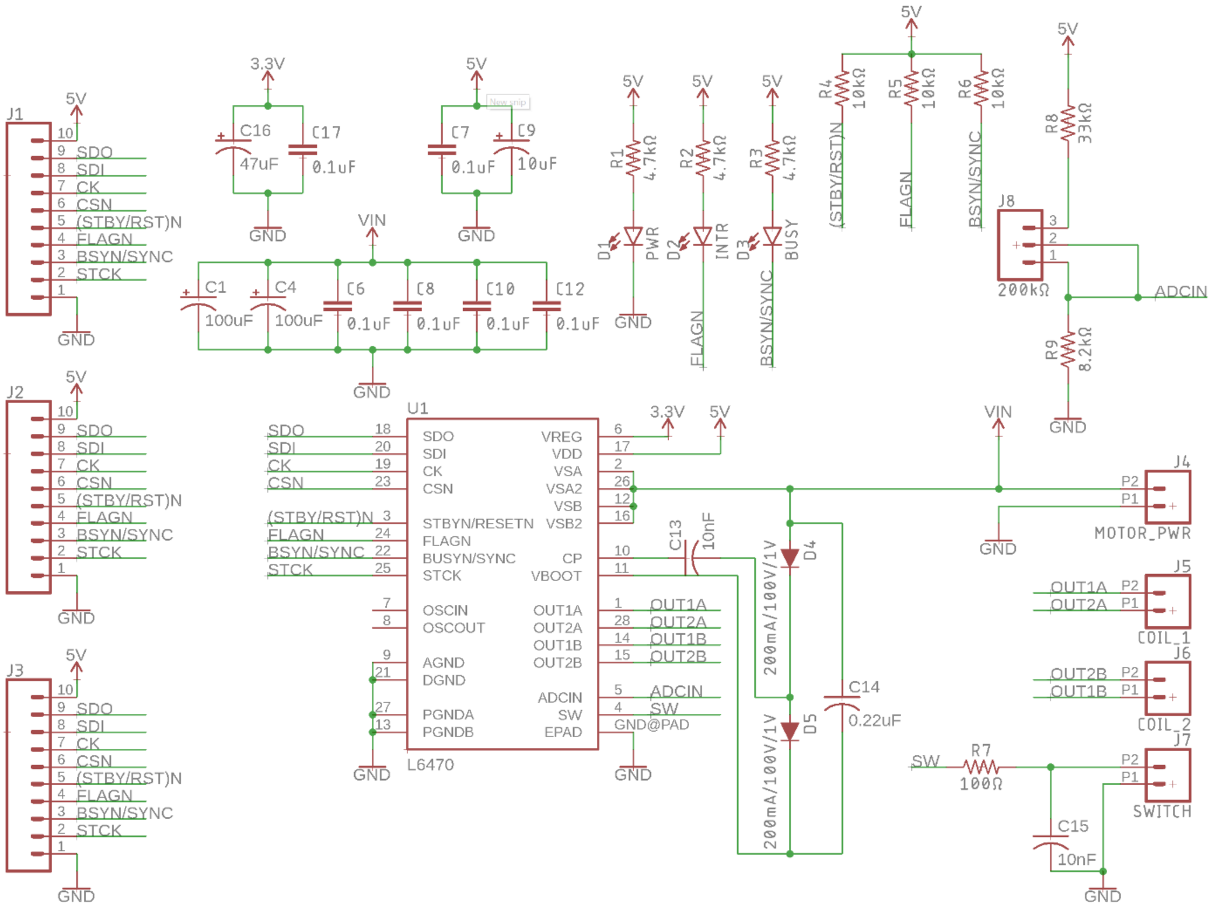


Figure 50: Motor Board Circuit Schematic

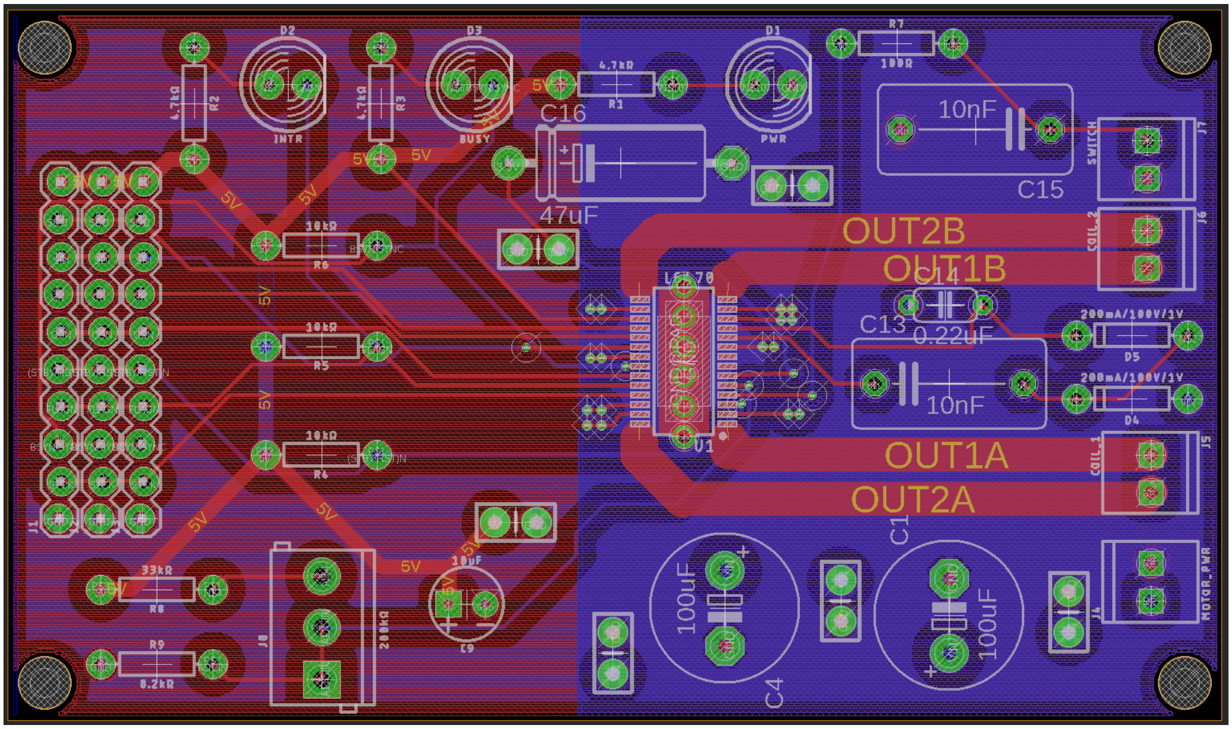


Figure 51: Motor Board PCB Design

Appendix D Requirement and Verification Tables

D.1 Joint Motor 1, Motor 2

Table 2: Joint Motor Requirements and Verification Procedures

Requirement	Verification Procedure
1. Torque of at least 1 N-m.	1. Assemble and connect all electronic components and boards together as specified in the block diagram. 2. Mount a stepper motor using the base with additional holes drilled and bolts to a sturdy table. 3. One end of solid aluminum 6063 bar of dimensions 1 m x 0.1 m x 0.1m will be attached to the motor shaft such that the length axis of the metal bar is perpendicular to the motor shaft axis. 4. A mass of 1 N will be attached to the other end of the metal bar. 5. Using the PC and Control Board, command the Motor Board rotate the motor such that the mass moves upwards. 6. Gradually increase commanded current, starting from 1A, until the mass moves. 7. If the commanded current exceeds 3A, the test has failed.
2. Inductance of less than 10mH.	1. Connect the RED and BLU wire (See Figure 22 above), each to one terminal of an LCR meter. 2. Connect the BLK and GRN wire (See Figure 22 above), each to one terminal of an LCR meter. 3. Measure inductance with the LCR meter and verify.
3. Have standard NEMA-type mounting holes, as shown in Figure 23 below.	1. The relevant measurements' correctness will be confirmed by caliper measurement of the relevant distance between the indicated points, using a with a Husky 6 in Digital Fractional Caliper with a measurement accuracy of +/- 0.001 in.

D.2 Pen Actuator

Table 3: Pen Actuator Requirements and Verification Procedures

Requirement	Verification Procedure
1. Actuation travel of 45 degrees +/- 5%.	1. Before manufacturing, the relevant measurement's correctness will be confirmed via applying the Measure tool to the relevant 3D model using Autodesk Inventor. 2. Assemble actuator. 3. After manufacturing, the relevant measurement's correctness will be confirmed by manually moving the relevant parts and caliper measurement of the relevant distance between the indicated points, using a with a Husky 6 in Digital Fractional Caliper with a measurement accuracy of +/- 0.001 in.
2. Actuation speed of 1s +/- 10%, as measured starting from the paper contacting position to the fully-lifted position, and vice versa.	1. Assemble all physical components and connect all electronic components and boards together as specified in the block diagram. 2. Install a camera such that the actuator is clearly visible. 3. Using the PC, command the actuator to move as required. 4. Use the camera recording to determine the delay.
3. Maximum load of 0.5 lb. +/- 5%.	1. Before manufacturing, the relevant measurement's correctness will be confirmed via applying static simulations to the relevant 3D model using Autodesk Inventor. 2. Assemble arm and add load as needed. 3. Confirm arm does not break or show signs of irreversible damage, such as visible stress patterns.
4. Suspension system enabling a z-axis compliance of 0.25 in +/- 5%.	1. Before manufacturing, the relevant measurement's correctness will be confirmed via applying the Measure tool to the relevant 3D model using Autodesk Inventor. 2. Assemble actuator. 3. After manufacturing, the relevant measurement's correctness will be confirmed by manually moving the relevant parts and caliper measurement of the relevant distance between the

	indicated points, using a with a Husky 6 in Digital Fractional Caliper with a measurement accuracy of +/- 0.001 in.

D.3 Arm Segment/Joint

Table 4: Arm Segment/Joint Requirements and Verification Procedures

Requirement	Verification Procedure
1. Each arm segment should have a length of 5 in +/- 10%, as measured between the rotational axis of Joint 1 and Joint 2 for Arm Segment 1, and as measured between the rotational axis of Joint 2 and the movement axis of the Pen Actuator for Arm Segment 2.	1. Before manufacturing, the relevant measurement's correctness will be confirmed via applying the Measure tool to the relevant 3D model using Autodesk Inventor. 2. After manufacturing, the relevant measurement's correctness will be confirmed by caliper measurement of the relevant distance between the indicated points, using a Husky 6 in Digital Fractional Caliper with a measurement accuracy of +/- 0.001 in.
2. Each arm segment and associated joint (including shafts, bearings, sensors, screws, etc.), excluding the motor/end effector associated with each arm segment should weight at most 3 pounds.	1. Relevant parts will be assembled and placed on a scale to determine if the arm segment assembly is overweight.
3. Sufficient structural integrity to ensure deflection with maximum load and under maximum motor acceleration is under 0.1 inches in in any direction.	1. Assemble all physical components and connect all electronic components and boards together as specified in the block diagram. 2. Install cameras such that every axis of the arm segment is visible, together with appropriate checkboard backgrounds to enable measurement. 3. Use a short PC program to send appropriate commands to the Control Board to actuate joint motors at maximum acceleration and measure flex sensor displacement. 4. Reconcile results and account for actual camera placement with model simulation.
4. Able to bear load of at least 2 pounds at end of Arm Segment 2.	1. Assemble all physical components and connect all electronic components and boards together as specified in the block diagram.

	2. Confirm arm does not break or show signs of irreversible damage, such as visible stress patterns.
5. Each revolute joint should enable 270 degrees +/- 20% of freedom about the joint axis.	1. Assemble all physical components and connect all electronic components and boards together as specified in the block diagram. 2. Use a short PC program to send appropriate commands to the Control Board to actuate joint motor counterclockwise until limit switches are triggered. 3. Measure maximum angular displacement via stepper motor IC controller motor angle feedback. 4. Use a short PC program to send appropriate commands to the Control Board to actuate joint motor clockwise until limit switches are triggered. 5. Measure maximum angular displacement via stepper motor IC controller motor angle feedback.

D.4 Limit Switches

Table 5: Limit Switch Requirements and Verification Procedures

Requirement	Verification Procedure
1. Latency of at most 10 ms +/- 1%.	1. Assemble all physical components and connect all electronic components and boards together as specified in the block diagram. 2. Upload a modified version of the microcontroller software that echoes back any limit switch triggering. 3. Install a camera such that the limit switch and arm's contact area is clearly visible. 4. Using the PC, command the arm such that the limit switch will be triggered. 5. Note the time recorded on the PC, and compare with the camera recording to determine the delay.
2. Able to withstand 10 mA or more of current at 5V +/- 10% for at least 100 hours.	1. Assemble all physical components and connect all electronic components and boards together as specified in the block diagram. 2. Leave system on for 100 hours.

	3. Redo the verification procedures for Requirement 1 to confirm no degradation in functionality.

D.5 Microcontroller

Table 6: Microcontroller Requirements and Verification Procedures

Requirement	Verification Procedure
1. Built-in hardware timer for servo PWM with resolution of 0.1 ms or better.	1. Check microcontroller datasheet and confirm requirements. 2. Assemble and connect all electronic components and boards together as specified in the block diagram. 3. Upload a modified version of the microcontroller software that outputs a 50% duty-cycle, 1 ms period square wave using Timer1 on pin PD4 of the microcontroller. 4. Verify output is as expected, i.e. error in square wave period is less than 0.1 ms, with an oscilloscope measuring the aforementioned pin.
2. 5V TTL logic voltage level.	1. Check microcontroller datasheet and confirm requirements. 2. Assemble and connect all electronic components and boards together as specified in the block diagram. 3. Upload a modified version of the microcontroller software that outputs a 50% duty-cycle, 1 ms period square wave using Timer1 on pin PD4 of the microcontroller. 4. Verify output on PD4 have low levels between 0V to 0.8V and high levels between 2V and 5V with an oscilloscope.
3. Greater than or equal 30 GPIO pins for reading sensors and controlling status LEDs.	1. Check microcontroller datasheet and confirm requirements. 2. Assemble and connect all electronic components and boards together as specified in the block diagram. 3. Upload a modified version of the microcontroller software that allows us to read and write to/from I/O ports from a PC serial connection. 4. Using a short PC program compatible with the aforementioned program, write

	to I/O ports containing LEDs all ones to turn on all LEDs. 5. Verify LEDs light as is as expected. 6. Using a short PC program compatible with the aforementioned program, write to I/O ports containing LEDs all zeros to turn off all LEDs. 7. Verify LEDs turn off as is as expected. 8. Set all switches to OFF position using tape. 9. Using a short PC program compatible with the aforementioned program, read from I/O ports containing switches. 10. Verify switches read as is as expected. 11. Set all switches to ON position using tape. 12. Using a short PC program compatible with the aforementioned program, read from I/O ports containing switches. 13. Verify switches read as is as expected.
4. At least one SPI port for communications with stepper motor drivers.	1. Check microcontroller datasheet and confirm requirements. 2. Assemble and connect all electronic components and boards together as specified in the block diagram. 3. Upload a modified version of the microcontroller software that echoes back all characters sent to it via SPI and Serial. 4. Using a short PC program, send random data to the SPI port. 5. Verify output is as expected.
5. At least one serial port for communication with PC.	1. Check microcontroller datasheet and confirm requirements. 2. Assemble and connect all electronic components and boards together as specified in the block diagram. 3. Upload a modified version of the microcontroller software that echoes back all characters sent to it via SPI and Serial. 4. Using a short PC program, send random data to the Serial port. 5. Verify output is as expected.
6. At least 32K program memory.	1. Check microcontroller datasheet and confirm requirements.

	2. Assemble and connect all electronic components and boards together as specified in the block diagram. 3. Upload a modified version of the microcontroller software that allows us to read and write to/from arbitrary Flash locations from a PC serial connection. 4. Using a short PC program compatible with the aforementioned program, write random data to the entire Flash address space, i.e. every Flash memory bit, and read back the written data from Flash to the PC 5. Verify output is as expected.
7. At least 16K RAM for buffering PC arm angle lists.	1. Check microcontroller datasheet and confirm requirements. 2. Assemble and connect all electronic components and boards together as specified in the block diagram. 3. Upload a modified version of the microcontroller software that allows us to read and write to/from arbitrary RAM locations from a PC serial connection. 4. Using a short PC program compatible with the aforementioned program, write random data to the entire RAM address space, i.e. every RAM bit, and read back the written data from RAM to the PC. 5. Verify output is as expected.

D.6 Status LEDs

Table 7: Status LEDs Requirements and Verification Procedures

Requirement	Verification Procedure
1. 100 mcd brightness under a current draw of 0.5 mA +/- 10%.	1. Make sure the bench power supply mentioned below is plugged in correctly and grounded, and is powered off. 2. Connect one end of a 4.7kΩ resistor to the positive terminal of the bench power supply, and connect the other end of the resistor to the anode of the LED, using a breadboard and breadboard wires as necessary. 3. Connect the cathode of the LED to the bench power supply. 4. Set the power supply to +5V with a current limit of 0.5 mA, and turn on the output.

	5. Verify brightness is as expected using a Digital Light Meter LT300 from Extech Instruments.

D.7 USB-to-Serial Converter

Table 8: USB-to-Serial Converter Requirements and Verification Procedures

Requirement	Verification Procedure
1. Enables PC to communicate with serial port on microcontroller with speed of 115.2 Kbps or greater.	1. Check USB-to-Serial Converter IC datasheet. 2. Assemble and connect all electronic components and boards together as specified in the block diagram. 3. Upload a modified version of the microcontroller software that echoes back all characters sent to it via the serial port. 4. Using a short PC program, send random data to the Serial port at 115.2 Kbps. 5. Verify received output on computer is as expected.
2. Serial port output is 5V TTL compliant.	1. Check USB-to-Serial Converter IC datasheet. 2. Assemble and connect all electronic components and boards together as specified in the block diagram. 3. Using a short PC program, send random data to the Serial port at 115.2 Kbps. 4. Verify received output on serial port pins have low levels between 0V to 0.8V and high levels between 2V and 5V with an oscilloscope.

D.8 Bench Power Supply (+5V)

This power supply provides power for all logic ICs, in particular, the microcontroller and the USB-to-Serial converter IC, and also provides power for the status LEDs and the servo used to drive the linear actuator. As each servo consumes 250 mA and 500 mA peak when stalled, we estimate a 3A total board consumption. As the system is immobile, a static wall-based power supply is acceptable, and necessary due to the high system power required.

Table 9: Bench Power Supply (+5V) Converter Requirements and Verification Procedures

Requirement	Verification Procedure
1. 5V +/- 10%, 3A +/- 5% output.	1. Make sure the bench power supply, multimeter, and electronic load mentioned below are plugged in correctly and grounded, and are powered off.

	2. Turn on and set a Keysight 34461A multimeter into DCV mode, then using banana cables, attach the multimeter such that the positive terminals of the multimeter and bench power supply are connected, and similarly for the negative terminals. 3. Using banana cables, attach the Agilent 6060B electronic load such that the positive terminals of the electronic load and bench power supply are connected, and similarly for the negative terminals. 4. Turn on and set the bench power supply to output +5V with a current limit of 3A. 5. Turn on and set the Agilent 6060B electronic load in the correct mode, to emulate a +5V 3A DC load. 6. Check that the measured voltage and current meets requirements, using the multimeter and an ammeter integrated in the electronic load respectively.
2. Ripple of 100 mV or lower under steady state, no motor movement conditions.	1. Make sure the bench power supply, oscilloscope, and robotic arm electronics mentioned below are plugged in correctly and grounded, and are powered off. 2. Turn on the DSO7104B oscilloscope, then using a probe, attach the oscilloscope such that the signal terminal of the oscilloscope and the positive terminal of the +24V bench power supply are connected, and similarly for the ground terminals of both devices. 3. Using banana cables, attach the robotic arm's Control Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +5V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +5V bench power supply. 4. Using banana cables, attach the robotic arm's Motor Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +24V bench power supply are connected, and similarly for the negative terminals of

	the Motor Board's motor power supply terminal and the +24V bench power supply. 5. Turn on both bench power supplies and command the Control Board via the PC to not attempt to move any motors. 6. Check that the measured ripple meets specifications, using measurement capabilities built into the oscilloscope.
3. Input of 120V AC.	1. Check the specifications of the bench power supply for proper certifications.

D.9 Stepper Motor Driver IC

The motor driver ICs receive commands from the microcontroller and monitors for abnormal conditions, including overtemperature, motor overcurrent, and motor stalling, and relays this information back to the microcontroller, and generate a constant current PWM waveform to drive the stepper motors. Due to the inherent back-EMF of the stepper design, stepper motors require high voltage for high rotational speeds. We use a constant current PWM waveform to prevent the motor coils from overheating, and also to allow for microstepping, increasing stepper motor positioning accuracy and movement smoothness. The stepper motor controller ICs communicate with the microcontroller over the SPI bus, while power is supplied by a separate +24V bench power supply.

Table 10: Joint Motor Requirements and Verification Procedures

Requirement	Verification Procedure
1. 3A R.M.S motor drive current or greater at 24 V operating voltage or greater.	1. Assemble and connect all electronic components and boards together as specified in the block diagram, except for the stepper motors. 2. Make sure the bench power supply, electronic load, and robotic arm electronics mentioned below are plugged in correctly and grounded, and are powered off. 3. Using banana cables, attach the Agilent 6060B electronic load such that the positive terminals of the electronic load and bench power supply are connected, and similarly for the negative terminals. 4. Using banana cables, attach the robotic arm's Control Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +5V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +5V bench power supply.

	5. Using banana cables, attach the robotic arm's Motor Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +24V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +24V bench power supply. 6. Turn on both bench power supplies. 7. Turn on and set the Agilent 6060B electronic load in the correct mode, to emulate a +24V 3A RMS, 5A peak load. 8. Command the Control Board via the PC to move the stepper motors with a 3A R.M.S. drive current. 9. Maintain this load for 1 hour. 10. Check that the measured current meets specifications, using measurement capabilities built into the electronic load and bench power supplies, and check the Motor Board and stepper motor controller for any sign of thermal damage via visual inspection and a Etekcity 630 Lasergrip non-contact thermometer with measurement accuracy of +- 0.1 deg. C. All board and component temperatures should remain below 100 deg. C.
2. 5A peak motor drive current or greater at 24 V operating voltage or greater.	1. Assemble and connect all electronic components and boards together as specified in the block diagram, except for the stepper motors. 2. Make sure the bench power supply, electronic load, and robotic arm electronics mentioned below are plugged in correctly and grounded, and are powered off. 3. Using banana cables, attach the Agilent 6060B electronic load such that the positive terminals of the electronic load and bench power supply are connected, and similarly for the negative terminals. 4. Using banana cables, attach the robotic arm's Control Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +5V bench power supply are connected, and similarly for the negative terminals of

	the Motor Board's motor power supply terminal and the +5V bench power supply. 5. Using banana cables, attach the robotic arm's Motor Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +24V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +24V bench power supply. 6. Turn on both bench power supplies. 7. Turn on and set the Agilent 6060B electronic load in the correct mode, to emulate a +24V 3A RMS, 5A peak load. 8. Command the Control Board via the PC to move the stepper motors with a 3A R.M.S. drive current. 9. Maintain this load for 1 hour. 10. Check that the measured current meets specifications, using measurement capabilities built into the electronic load and bench power supplies, and check the Motor Board and stepper motor controller for any sign of thermal damage via visual inspection and a Etekcity 630 Lasergrip non-contact thermometer with measurement accuracy of +- 0.1 deg. C. All board and component temperatures should remain below 100 deg. C.
3. Overcurrent protection.	1. Assemble and connect all electronic components and boards together as specified in the block diagram, except for the stepper motors. 2. Make sure the bench power supply, electronic load, and robotic arm electronics mentioned below are plugged in correctly and grounded, and are powered off. 3. Using banana cables, attach the Agilent 6060B electronic load such that the positive terminals of the electronic load and bench power supply are connected, and similarly for the negative terminals. 4. Using banana cables, attach the robotic arm's Control Boards such that the positive terminals of the Motor Board's

	motor power supply terminal and the +5V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +5V bench power supply. 5. Using banana cables, attach the robotic arm's Motor Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +24V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +24V bench power supply. 6. Turn on both bench power supplies. 7. Turn on and set the Agilent 6060B electronic load in the correct mode, to emulate a +24V 7A RMS, 7A peak load. 8. Command the Control Board via the PC to move the stepper motors with a 7A R.M.S. drive current. 9. Wait until overcurrent shutdown occurs. 10. During wait, check that the measured current meets specifications, using measurement capabilities built into the electronic load and bench power supplies, and check the Motor Board and stepper motor controller for any sign of thermal damage via visual inspection and a Etekcity 630 Lasergrip non-contact thermometer with measurement accuracy of +- 0.1 deg. C. All board and component temperatures should remain below 100 deg. C.
4. Overtemperature protection.	1. Assemble and connect all electronic components and boards together as specified in the block diagram, except for the stepper motors. 2. Make sure the bench power supply, electronic load, and robotic arm electronics mentioned below are plugged in correctly and grounded, and are powered off. 3. Using banana cables, attach the Agilent 6060B electronic load such that the positive terminals of the electronic load

	and bench power supply are connected, and similarly for the negative terminals. 4. Using banana cables, attach the robotic arm's Control Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +5V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +5V bench power supply. 5. Using banana cables, attach the robotic arm's Motor Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +24V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +24V bench power supply. 6. Turn on both bench power supplies. 7. Turn on and set the Agilent 6060B electronic load in the correct mode, to emulate a +24V 5A RMS, 7A peak load. 8. Command the Control Board via the PC to move the stepper motors with a 5A R.M.S. drive current. 9. Wait until overheat shutdown occurs. 10. During wait, check that the measured current meets specifications, using measurement capabilities built into the electronic load and bench power supplies, and check the Motor Board and stepper motor controller for any sign of thermal damage via visual inspection and a Etekcity 630 Lasergrip non-contact thermometer with measurement accuracy of +- 0.1 deg. C. All board and component temperatures should remain below 100 deg. C.
5. Motor stall detection.	1. Assemble all physical components and connect all electronic components and boards together as specified in the block diagram. 2. Make sure the bench power supply and robotic arm electronics mentioned below are plugged in correctly and grounded, and are powered off.

	3. Using banana cables, attach the robotic arm's Control Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +5V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +5V bench power supply. 4. Using banana cables, attach the robotic arm's Motor Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +24V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +24V bench power supply. 5. Turn on both bench power supplies. 6. Command the Control Board via the PC to move the stepper motors with a 3A R.M.S. drive current. 7. Block the movement and wait for shutdown with a piece of metal placed in the path of the arm. 8. Wait until stall shutdown occurs. 9. Check that the measured current meets specifications, using measurement capabilities built into the electronic load and bench power supplies, and check the Motor Board and stepper motor controller for any sign of thermal damage via visual inspection and a Etekcity 630 Lasergrip non-contact thermometer with measurement accuracy of +- 0.1 deg. C. All board and component temperatures should remain below 100 deg. C.
6. Constant-current motor drive with capability for 1/32 microstep divisions or better.	1. Check the specifications of the stepper motor controller IC for relevant functionality. 2. Assemble all physical components and connect all electronic components and boards together as specified in the block diagram. 3. Make sure the bench power supply and robotic arm electronics mentioned below are plugged in correctly and grounded, and are powered off.

	4. Using banana cables, attach the robotic arm's Control Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +5V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +5V bench power supply. 5. Using banana cables, attach the robotic arm's Motor Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +24V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +24V bench power supply. 6. Turn on both bench power supplies. 7. Command the Control Board via the PC to move the stepper motors with a 3A R.M.S. drive current using the correct microstepping mode register settings. 8. Check that the measured current meets specifications for a microstepping waveform, i.e. is sinusoidal with a Total Harmonic Distortion of less than 10% with the correct R.M.S. current level, using an oscilloscope, and check the Motor Board and stepper motor controller for any sign of thermal damage via visual inspection and a Etekcity 630 Lasergrip non-contact thermometer with measurement accuracy of +- 0.1 deg. C. All board and component temperatures should remain below 100 deg. C.

D.10 Bench Power Supply (+24V)

Table 11: Bench Power Supply (+24V) Requirements and Verification Procedures

Requirement	Verification Procedure
1. 24V +/- 10%, 10A +/- 5% output.	1. Make sure the bench power supply, multimeter, and electronic load mentioned below are plugged in correctly and grounded, and are powered off. 2. Turn on and set a Keysight 34461A multimeter into DCV mode, then using banana cables, attach the multimeter

	such that the positive terminals of the multimeter and bench power supply are connected, and similarly for the negative terminals. 3. Using banana cables, attach the Agilent 6060B electronic load such that the positive terminals of the electronic load and bench power supply are connected, and similarly for the negative terminals. 4. Turn on and set the bench power supply to output +24V with a current limit of 10A. 5. Turn on and set the Agilent 6060B electronic load in the correct mode, to emulate a +24V 10A DC load. 6. Check that the measured voltage and current meets requirements, using the multimeter and an ammeter integrated in the electronic load respectively.
2. Ripple of 100 mV or lower under steady state, no motor movement conditions.	1. Make sure the bench power supply, oscilloscope, and robotic arm electronics mentioned below are plugged in correctly and grounded, and are powered off. 2. Turn on the DSO7104B oscilloscope, then using a probe, attach the oscilloscope such that the signal terminal of the oscilloscope and the positive terminal of the +24V bench power supply are connected, and similarly for the ground terminals of both devices. 3. Using banana cables, attach the robotic arm's Control Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +5V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply terminal and the +5V bench power supply. 4. Using banana cables, attach the robotic arm's Motor Boards such that the positive terminals of the Motor Board's motor power supply terminal and the +24V bench power supply are connected, and similarly for the negative terminals of the Motor Board's motor power supply

	terminal and the +24V bench power supply. 5. Turn on both bench power supplies and command the Control Board via the PC to not attempt to move any motors. 6. Check that the measured ripple meets specifications, using measurement capabilities built into the oscilloscope.
3. Input of 120V AC.	1. Check the specifications of the bench power supply for proper certifications.

Appendix E Parts and Materials and Schedule

E.1 Parts and Materials

Table 12: Parts/Materials Cost

Part ID	Part Description	Assembly	Quantity	Cost Per Unit	Cost Total Per Part
ATmega1284P-PU	Atmel ATmega1284P IC MCU 8BIT 128KB FLASH 40DIP	Control Board	1	$5.68	$5.68
COM-00097	Mini Pushbutton Switch	Control Board	1	$0.35	$0.35
COM-08571	Capacitor Ceramic 22pF 200V	Control Board	2	$0.25	$0.50
N/A	5x PCB Control Board	Control Board	1	$25.84	$25.84
PRT-10811	DC Barrel Jack Adapter - Breadboard Compatible	Control Board	1	$0.95	$0.95
SEN-00246	Optical Detector / Phototransistor - QRD1114	Control Board	4	$0.95	$3.80
3296W-1-204LF	200 kOhms 0.5W, 1/2W PC Pins Through Hole Trimmer Potentiometer Cermet 25 Turn Top Adjustment	Motor Board	2	$2.41	$4.82
COM-00096	Electrolytic Decoupling Capacitors - 100µF/25V	Motor Board	4	$0.35	$1.40
K103K15X7RF5TL2	10000pF ±10% 50V Ceramic Capacitor X7R Radial	Motor Board	4	$0.21	$0.84
K180J15C0GF5TL2	18pF ±5% 50V Ceramic Capacitor C0G, NP0 Radial	Motor Board	4	$0.21	$0.84
L6470H	L6470 Stepper Motor Controller IC HTSSOP28	Motor Board	2	$7.72	$15.44
N/A	3x PCB Motor Board	Motor Board	1	$61.70	$61.70
PRT-10571	Screw Terminals 2.54mm Pitch (2-Pin)	Motor Board	8	$0.75	$6.00
UVP1H470MPD	47µF 50V Aluminum Electrolytic Capacitors Radial, Can 2000 Hrs @ 85°C	Motor Board	2	$0.64	$1.28
UVR2AR22MDD	0.22µF 100V Aluminum Electrolytic Capacitors Radial, Can 2000 Hrs @ 85°C	Motor Board	2	$0.24	$0.48
COM-00523	Electrolytic Decoupling Capacitors - 10µF/25V	Motor Board/Control Board	4	$0.45	$1.80
COM-00536	Crystal 16MHz HC49/US	Motor Board/Control Board	3	$0.95	$2.85

COM-08375	Capacitor Ceramic 0.1µF 50V	Motor Board/Control Board	16	$0.25	$4.00
COM-08588	Diode Small Signal - 1N4148	Motor Board/Control Board	5	$0.15	$0.75
COM-09590	LED (5mm) LED - Basic Red 5mm	Motor Board/Control Board	6	$0.35	$2.10
COM-09592	LED (5mm) LED - Basic Green 5mm	Motor Board/Control Board	5	$0.35	$1.75
COM-09594	LED (5mm) LED - Basic Yellow 5mm	Motor Board/Control Board	6	$0.35	$2.10
COM-10969	Resistor Kit - 1/4W, Includes 0Ω, 1.5Ω, 4.7Ω, 10Ω, 47Ω, 100Ω, 220Ω, 330Ω, 470Ω, 680Ω, 1kΩ, 2.2kΩ, 3.3kΩ, 4.7kΩ, 10kΩ, 22kΩ, 47kΩ, 100kΩ, 330kΩ, 1MΩ	Motor Board/Control Board	1	$7.95	$7.95
PRT-00115	Female Headers 0.1" 40 Position	Motor Board/Control Board	10	$1.50	$15.00
PRT-00116	Break Away Headers – Straight 0.1" 40 Position	Motor Board/Control Board	10	$1.50	$15.00
23HS30-2804S	Nema 23 Stepper Motor 2.8A 1.9Nm (269oz.in) 76mm Length	Robotic Arm	2	$24.50	$49.00
60355K704	Ball Bearing, Sealed, Trade Number R8- 2RS, for 1/2" Shaft Diameter	Robotic Arm	6	$8.74	$52.44
6484K228	XL Series Timing Belt, Trade No. 180xL037	Robotic Arm	1	$6.41	$6.41
6484K237	XL Series Timing Belt, Trade No. 280xL037	Robotic Arm	1	$8.05	$8.05
94613A550	100x Off-White Nylon 6/6 Hex Head Screws, 1/4"- 20 Thread Size, 2" Long, Full y Threaded	Robotic Arm	1	$8.00	$8.00
94812A700	100x Nylon 6/6 Hex Nut 1/4"-20 Thread Size	Robotic Arm	1	$6.47	$6.47

95606A430	100x Nylon 6/6 Plastic Washer for 1/4" Screw Size, 0.312" ID, 0.5" OD	Robotic Arm	1	$7.25	$7.25
HPM# 07828	4x Compression Spring 5/16" x 1-1/2" x .020"	Robotic Arm	1	$2.68	$2.68
ROB-09065	Servo - Generic (Sub-Micro Size)	Robotic Arm	3	$8.95	$26.85
ROB-11884	Servo - Hitec HS-422 (Standard Size)	Robotic Arm	1	$9.95	$9.95
TOL-14584	PLA Filament 2.85mm - 1kg (Black)	Robotic Arm	3	$25.00	$75.00
				Parts Total	$435.32

E.2 Schedule

Table 13: Schedule

Week	Objective	Bingzhe Wei	Chenghao Duan	Tianhao Chi
Week 1 (Feb 19th)	Complete PC Software, Finalize 3D Models	Finalize 3D models of SCARA robotic arm's base and arm segments	Finalize 3D models of SCARA robotic arm's transmission and end effector	Implement all PC side image processing algorithms
Week 2 (Feb 26th)	Assemble Control Board, Start on Microcontroller Software	Design and purchase Motor Board and Control Board	Solder and verify Control Board functionality	Implement PC-microcontroller communication and limit switch reading functionality
Week 3 (Mar 5th)	Finish Microcontroller Software, Print all Physical Parts for Robot Arm	Integrate and test microcontroller software, Solder and verify one Control Board's functionality	Print all physical parts needed at UIUC MakerLab, and buy needed mechanical components.	Integrate and test microcontroller software
Week 4 (Mar 12th)	Assemble Both Motor Boards, Test Software	Solder and verify one Motor Board's functionality	Solder and verify one Motor Board's functionality	Integrate and test microcontroller software with actual Motor Boards connected
Week 5 (Mar 19th) Spring Break	Integrate all Components	Assemble the base and transmission and integrate all other assemblies assembled by other group members	Assemble the end effector assembly	Assemble the arm segments
Week 6 (Mar 26th)	Testing and Fine-Tuning	Motion test – tune microcontroller software as needed	Motion test - tune mechanical design as needed to allow	Motion test - tune PC image processing algorithms as needed

		to allow accurate robot movement	accurate robot movement	to allow accurate robot movement
Week 7 (April 2nd)	Testing and Fine-Tuning	Full system integration test – test drawing capabilities and debug as needed	Full system integration test – test drawing capabilities and debug as needed	Full system integration test – test drawing capabilities and debug as needed
Week 8 (April 9th)	Testing and Fine-Tuning	Tune PC style transfer network for best artistic effect	Continue tuning algorithms to increase robot arm movement speed as much as possible	Tune clustering algorithms for best artistic effect
Week 9 (April 16th)	Testing and Fine-Tuning	Train additional style transfer networks	Continue tuning algorithms to increase robot arm movement speed as much as possible	Tune vectorization algorithms for best artistic effect
Week 10 (April 23th)	Work on Final Report	Work on Final Report, focusing on electrical components of the project	Work on Final Report, focusing on mechanical components of the project	Work on Final Report, focusing on software components of the project
Week 11 (April 23th)	Rehearsal and Preparation Work on Final Presentation	Prepare Powerpoint slides and materials and procedures for a demonstration of robot's drawing capability to be shown at during the Final Presentation, and rehearse for the Final Presentation	Prepare Powerpoint slides and materials and procedures for a demonstration of robot's drawing capability to be shown at during the Final Presentation, and rehearse for the Final Presentation	Prepare Powerpoint slides and materials and procedures for a demonstration of robot's drawing capability to be shown at during the Final Presentation, and rehearse for the Final Presentation

Appendix F Integration Tests 1-3 Results

F.1 Integration Test 1: Synchronized Counterclockwise Rotation of Both Joint 1 and Joint 2

In this test, we command both Joint 1 and Joint 2 to rotate to the same final angle, starting from the same initial angle, in order to demonstrate that despite that both joints are decoupled from each other and that each joint can move independently, the joints can move in unison as well.

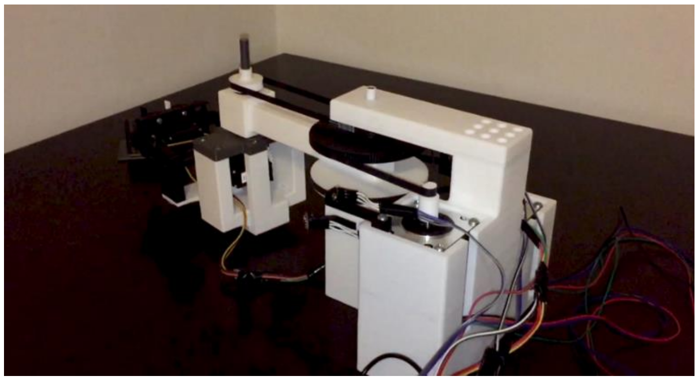


Figure 52: Test 1 Initial Arm Pose

Figure 53: Test 1 Intermediate Arm Pose

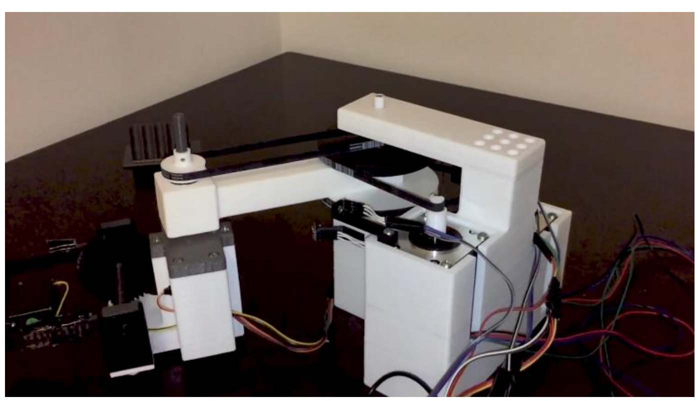


Figure 54: Test 1 Final Arm Pose

F.2 Integration Test 2: Joint 2 Clockwise Rotation, Joint 1 No Rotation

In this test, we command Joint 2 to rotate while Joint 1 remains stationary, in order to demonstrate that both joints are decoupled from each other and that each joint can move independently, and that movement in one joint does not induce movement in another joint.

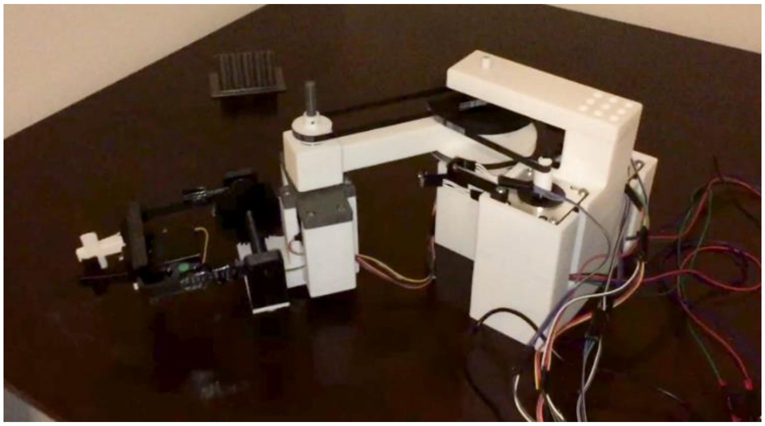


Figure 55: Test 2 Initial Arm Pose

Figure 56: Test 2 Intermediate Arm Pose

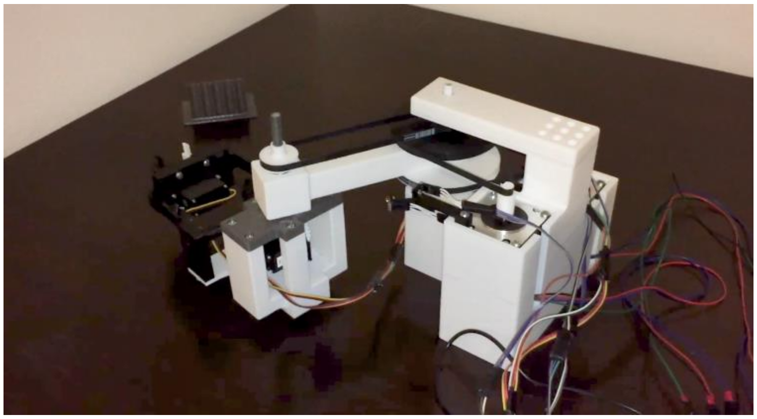


Figure 57: Test 2 Final Arm Pose

F.3 Integration Test 3: Joint 1 Counterclockwise Rotation, Joint 2 Clockwise Rotation

In this test, we command Joint 1 to rotate counterclockwise and Joint 2 to rotate clockwise, in order to demonstrate that both joints are decoupled from each other and that each joint can move independently of the other joint regardless of the angular position and angular velocity of the other joint.

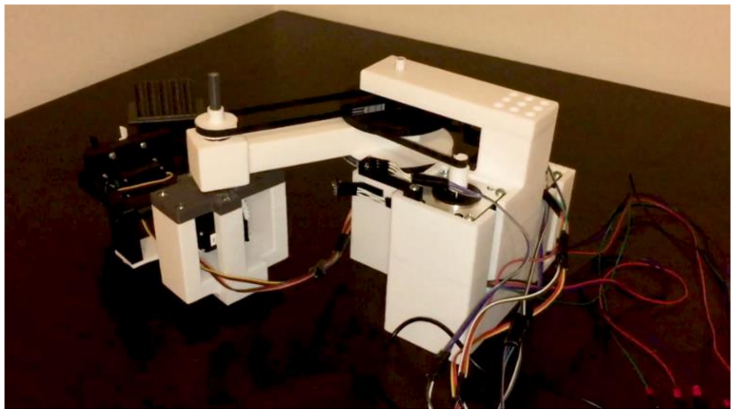


Figure 58: Test 3 Initial Arm Pose

Figure 59: Test 3 Intermediate Arm Pose

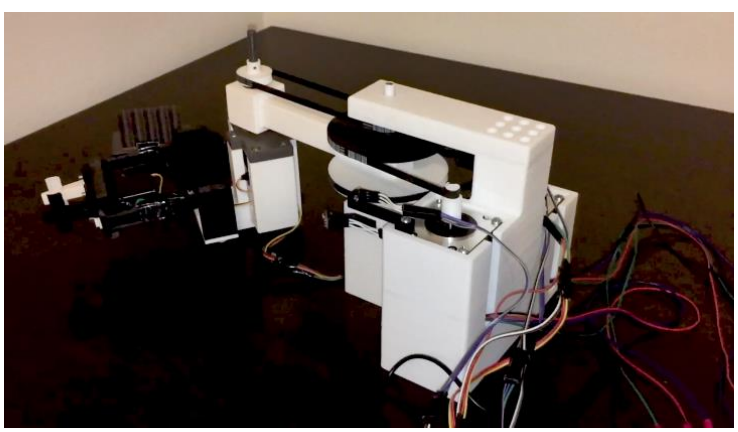


Figure 60: Test 3 Final Arm Pose

References

[1] S. K. Saha, Introduction to Robotics, New Delhi: Tata McGraw-Hill, 2008.

[2] MIT edX, "2.008x: Fundamentals of Manufacturing Processes," [Online]. Available: https://www.edx.org/course/fundamentals-manufacturing-processes-mitx-2-008x-0. [Accessed 8 February 2018].

[3] papabravo, "Rack & Pinion Robotic Gripper Jaw," 2018. [Online]. Available: https://www.thingiverse.com/thing:2661755. [Accessed 13 March 2018].

[4] F. Nicola, "Hitec HS-485HB," 11 January 2014. [Online]. Available: https://grabcad.com/library/hitec-hs-485hb-1. [Accessed 15 March 2018].

[5] Google, "Tensorflow," 2018. [Online]. Available: https://www.tensorflow.org/. [Accessed 1 March 2018].

[6] L. A. Gatys, A. S. Ecker and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in *IEEE Computer Vision and Pattern Recognition*, Las Vegas, 2016.

[7] A. Clark and F. Lundh, "Pillow," 2018. [Online]. Available: https://pillow.readthedocs.io/en/5.1.x/. [Accessed 10 April 2018].

[8] OpenCV Team, "OpenCV," 2018. [Online]. Available: https://opencv.org/. [Accessed 14 March 2018].

[9] NetworkX Developers, "NetworkX: Software for Complex Networks," 2018. [Online]. Available: https://networkx.github.io/. [Accessed 13 March 2018].

[10] College of Engineering, University of Illinois at Urbana-Champaign, "Salary Information 2015-2016," 2016. [Online]. Available: https://engineering.illinois.edu/documents/Salary.Info.Sheet.pdf. [Accessed 26 Feburary 2018].

[11] ECE 445 Course Staff, "Design Document," 2018. [Online]. Available: https://courses.engr.illinois.edu/ece445/guidelines/design-document.asp. [Accessed 26 Feburary 2018].

[12] IEEE, "IEEE Code of Ethics," 2018. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed 8 February 2018].

[13] M.-A. Belabbas, "ECE 486: A note on sensitivity," 2015. [Online]. Available: https://courses.engr.illinois.edu/ece486/fa2017/documents/ECE486note2_edited.pdf. [Accessed 26 Feburary 2018].

[14] S. Monnier, "Algorithmic worlds," 12 September 2014. [Online]. Available: http://www.algorithmic-worlds.net/. [Accessed 2 February 2018].

[15] Center of Excellence Digital Art, "compArt daDA: the database Digital Art," [Online]. Available: http://dada.compart-bremen.de. [Accessed 8 February 2018].

[16] R. Verostko, "The Algorists," 2011. [Online]. Available: http://www.algorists.org/algorist.html. [Accessed 8 February 2018].

[17] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks,* vol. 4, no. 2, pp. 251-257, 1991.

[18] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *25th International Conference on Neural Information Processing Systems*, Lake Tahoe, NV , 2012.

[19] S. Colton, "The Painint Fool," 2018. [Online]. Available: http://www.thepaintingfool.com. [Accessed 8 February 2018].

[20] H. Cohen, "The Further Exploits of AARON, Painter," *Stanford Humanities Review,* vol. 4, no. 2, 1995.

[21] The Franklin Institute, "Maillardet's Automaton," 2018. [Online]. Available: https://www.fi.edu/history-resources/automaton. [Accessed 8 February 2018].