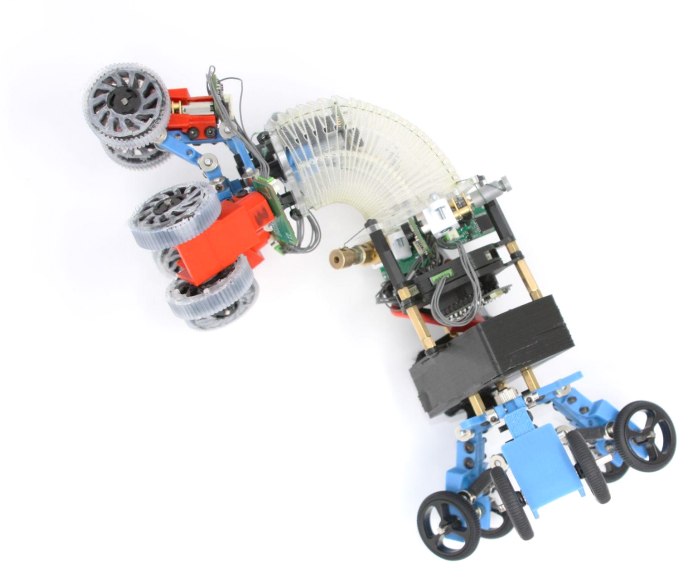
Continuum Locomotive Alternative for Robotic Adaptive-exploration (CLARA)



A Major Qualifying Project Report
submitted to the Faculty of the WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science

BY:
Brian Katz (RBE & ME)
Kate Wheeler (RBE & ME)

ADVISOR:
Professor Cagdas Onal

DATE:

This report represents the work of two WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its website without editorial or peer review. For more information about the projects program at WPI, see http://www.wpi.edu/Academics/Projects

Abstract

Current forms of remote pipe inspection, such as borescopes, are limited in their maneuverability, reach, and adaptability. We propose a salamander inspired soft robot as a novel battery powered pipe inspection tool that can maneuver in many/varying pipe sizes, vertical pipes, tees, and bends using a variable diameter suspension mechanism and without the need for a tether. The salamander robot has an origami-inspired continuum body structure with driven wheels on a suspension mechanism, running in closed-loop control to modify the body shape and change length to negotiate pipe networks and adjust force on the tube. The origami body utilizes a Yoshimura crease pattern to achieve continuously deformable cable driven bellows that enable steering by body deformation without traditional rigid mechanisms. To enable closed-loop force, velocity, and position control on each degree of freedom, we introduce a smart motor driver control board that can be mounted on the back of small electric motors and used in a broad range of applications beyond the scope of this project. The motor drivers are commanded via I2C by a custom mainboard that receives remote commands via WiFi and UART from a gamepad or similar user input. The system achieves 4.5” of linear compression, 1.6” of diameter range and 85.8° maximum bending angle. This project provides a foundation for future research into multi-robot inclusion and sensor integration.

Acknowledgements

The CLARA MQP Team would like to thank the following people and groups for their support throughout this project:

- WPI Soft Robotics Lab for lending their support, mentorship, materials, and lab space for use by the team.
- Gabby Conard for her mechanical design knowledge, project management advice, guidance, and support throughout the project.
- Yinan Sun for his mentorship throughout the duration of the project and help with understanding the principles and design behind the Salamanderbot.
- Dan Moyer for his expertise and advice in PCB design, component selection, and project guidance.
- Yoni Weiner for providing us with a custom Serial communication protocol with a USB gamepad and his overall support for our project.
- The members of the OREO'd MQP team (Tyler Looney, Archie Milligan, Daniel Perno, Nathan Savard, Michael Scalise, Augustus Teran, Ryley Wheelock) for their continued support throughout our project.
- Professor Greg Lewin and Professor Markus Nemitz for their design advice.

Lastly, the CLARA MQP team wanted to thank our advisor, Professor Onal, for his guidance throughout our project.

In Memoriam

We dedicate this Major Qualifying Project to Jiyang “Jeffrey” Wu, our third partner, who passed away in the summer of 2021.

Jeffrey was a Robotics Engineering and Mechanical Engineering double major at WPI who was actively involved in many areas on campus and his local community, such as founding the LEGO Club, being a member of WPI’s Christian Bible Fellowship and a member of the Bethlehem Bible Church.

Jeffrey was an incredibly generous person, always checking in on others and sharing his passions with everyone. He had many passions such as music, cooking, hiking, building with LEGOs, photography, and spending his time with friends and family. He always wanted to include his friends in his passions and was always eager to try new things with his friends.

Jeffrey cared for his dog, Clara, who was named after Clara Schumann. We have decided to name this project Continuous Locomotion Alternative for Robotic Adaptive-exploration (CLARA) in honor of her.

In early discussions with Jeffrey regarding this project, we had not yet decided what the project direction would be, however, we all agreed that the goal was to produce something new and fascinating, as well as a robot we could all be proud of. It was this goal that has driven us throughout the last year and kept the team going through many long days and nights of work. While Jeffrey may not be here to see the finished product of CLARA, we hope that he would also be as excited and proud of what we have created as we are.

Thank you for everything Jeffrey. Though you are not with us right now, we hold your thoughts, memories, and smile very close in our hearts. Rest in peace.

Table of Contents

List of Figures & Equations

List of Tables

1. Introduction

Complex pipe systems are frequently found in fields ranging from HVAC to aerospace. It is important to perform routine inspection and maintenance of these pipes to prevent and diagnose problems. Problems requiring inspection can include fouling, cracks, leaks, debris, and more, which are prone to occur at any point within a pipe network. However, these vast systems are challenging to navigate for humans as they can span entire buildings or countries, pose temperature risks, pose chemical risks, and are often too compact for inspectors to enter.

Modern technology, such as cameras and borescopes, allow for inspectors to visually examine the inside of pipes of various diameters and lengths without exposing themselves to the adverse conditions around the pipe surfaces [1]. These solutions are often waterproof and built to withstand temperature fluctuations to allow for inspection of underwater and high temperature environments. To maneuver around pipe bends, curves, and intersections, borescopes are flexible, allowing users to navigate through complex passageways and view potential cracks and leaks with LED lighting at the probe tip. This technology allows for a non-invasive and non-destructive method for inspectors to diagnose a location of interest, and follow up with proper repair methods.

Figure 1: Example of Borescope [2]

However, these systems feature limited length and maneuverability. Borescopes have a defined length and cannot continuously explore throughout a pipe system, making inspection more tedious. While it is possible to articulate the tips of borescopes to change the field of view, doing so proves difficult for those who are not experienced, as the only means of control is typically a joystick or 4-directional D-Pads. While industry considers borescopes to be a standard for pipe inspection, they are prone to breakage and require costly repairs. Additionally, borescopes only provide the potential for inspection and afford no possibility for resolution of issues found during inspection.

In contrast to traditional cameras and borescopes, robots can travel throughout a system indefinitely with their only limits being tether length or on-board battery capacity. Several types of robots have been developed in recent years to address this problem, including both passive

and active designs. In passive designs, tetherless soft robots such as Lighthouse from WatchTower Robotics utilize flow pressure to continue their motion throughout pipes and detect potential leaks with skirt sensors [3]. Active robots are more rigid, such as the TriTrax from Eddyfi Technologies, which utilizes tracks spaced 120 degrees apart to travel through pipes and a camera to find discrepancies [4]. Additionally, these robots advertise a myriad of possible attachments suitable for custom solutions, such as additional types of cameras and sensors, attachments to drill through fouling, debris, and more.

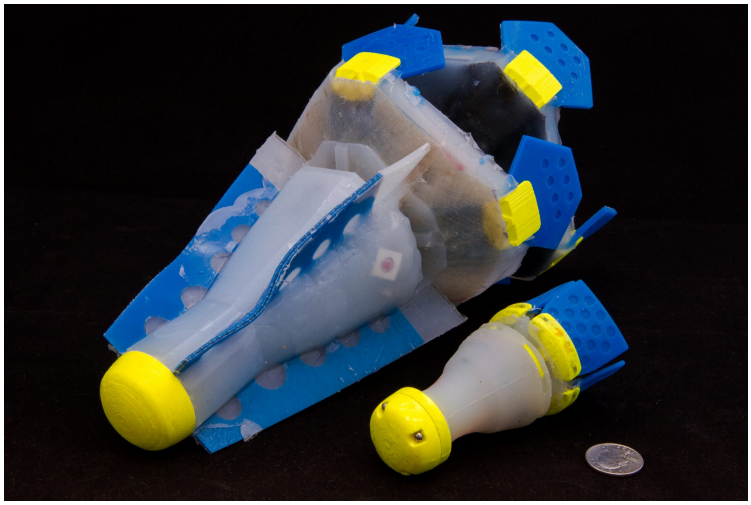


Figure 2: Lighthouse Robot [3]

Continuum Locomotion Alternative for Robotic Adaptive-exploration (CLARA) is a Major Qualifying Project (MQP) to incorporate semi-rigid continuum actuators in the design of a pipe inspection robot to further the research in WPI Soft Robotics Lab on deformable robotics. This project adapts the Yoshimura cable-driven origami module from previous WPI Soft Robotics Lab research for use in pipes of varying diameters as well as decreasing the size of the module overall for use in a wider variety of pipe networks and applications. This necessitated the creation and overhaul of preexisting custom printed circuit boards (PCBs) to accomplish actuation, power distribution, and control tasks with a much smaller body size and modular electrical control architecture. The robot is adaptable, being able to conform to various diameter pipes, bend radii, vertical pipe sections, and pipe reductions and expansions. This project performed a design analysis between two diameter expansion mechanisms, a lead screw drive and a cam-slot design, to determine which design is more effective for maintaining and controlling diameter in variable size pipes. This project serves to expand the applications of origami robots and display their functionality and benefits in challenging environments.

2. Background

2.1 Origami in Engineering

Origami folding takes its origins from nature and proves to be an important tool to solve complex engineering problems when traditional rigid manipulators are not sufficient. Single sheets and multiple sheets of material can form complex geometries to create various desirable mechanical behaviors. Origami, in theory, offers universality, or the ability to create any geometric shape from a combination of folded patterns. However, in practice, the number of folds that can be performed before a structure becomes too difficult to fold limits origami as a design tool. Additionally, typical shapes, such as smooth surfaces, may be more challenging to create using origami patterns and require many folds, thus using more material. Due to the nature of the folds, origami robots have built-in compliance and exhibit the properties of both soft and more traditional rigid robots, making them semi-rigid [5].

The mechanical properties of origami modules are based on several factors, such as properties of the chosen folding material and crease pattern design. Computational origami, a software tool that converts input 3D structures into crease patterns, is becoming a more prominent tool towards the design of complex origami structures. To create the sheet used to fold these complex shapes, subtractive manufacturing, often with laser cutters or engravers, is used to generate perforations and creases. The benefit of adding creases and perforations, as opposed to using a plain sheet of the folding material, is that it allows the folds to hold even when using materials with greater stiffness as well as provide ease of folding for greater assembly efficiency [6]. However, researchers are advancing additive manufacturing technology to enable the creation of integrated systems within robotic materials. In the future, it is likely this technology could be applied towards the creation of origami folds, buckling structures, and semi-rigid robots [5].

2.2 Existing Origami Patterns for Robotic Applications

A variety of origami folding patterns have been adapted for use in robotic applications. Desirable attributes of a folding pattern that can lend itself to these sorts of applications can include but are not limited to, compressibility, semi-rigidity, and patterns that are able to combine form with function.

Researchers have developed a myriad of patterns to perform controllable mechanical motion and functionality. The achievable functions are broad and can include but are not limited to contractive motion, flat panels for self-assembly, compressible and inflatable modules [7], and more generally, modules that allow for multiple degrees of freedom articulation without multiple rigid rotational or prismatic joints.

For example, WPI Soft Robotics Lab identified three types of segments to perform contractile motion: the diagonal pattern, the Yoshimura pattern, the Miura Ori (herringbone) pattern, and the waterbomb base pattern (Figures 3-6). Each pattern features a combination of perforated creases and hard folds to maintain high stiffness with compliancy, as well as easy

folding. The diagonal pattern (Figure 3) offers desirable traits as it couples rotational with translational motion in its folded form.

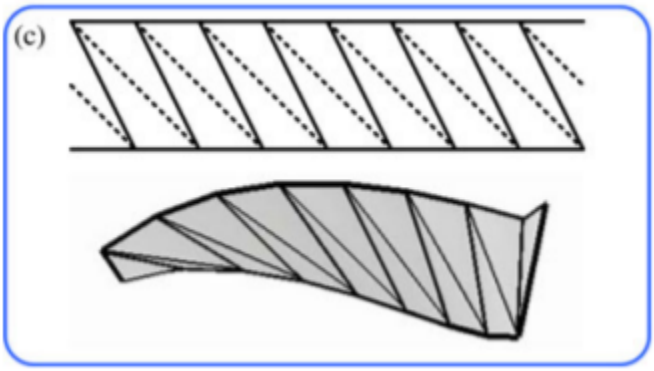


Figure 3: The Diagonal Pattern [6]

In contrast, the waterbomb base, shown in Figure 4**,** is widely used as a result of its ability to let a flat sheet collapse on itself, making it ideal for situations where axial contraction is desired. An example of a robot that takes advantage of both the waterbomb base pattern and the diagonal pattern is a peristaltic worm robot that has a soft body capable of movement via NiTi (Nitinol) coil spring actuators [6].

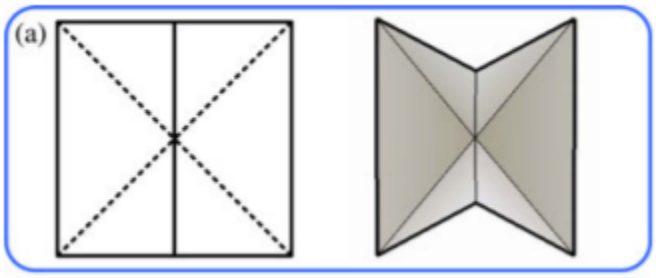


Figure 4: The Waterbomb Base Pattern [6]

The Miura Ori (Figure 5), frequently called the herringbone pattern, was originally invented for use in space solar panels, although is frequently also used for foldable maps and other situations where flat, compact, and rigid foldability with a single degree of freedom actuation is desired [8].

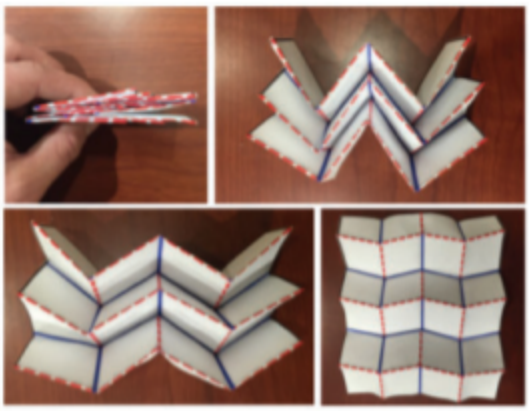


Figure 5: The Miura Ori Pattern [8]

2.3 The Yoshimura Module

The Yoshimura pattern features a purely axial compression, which is much simpler to fold and simple for attaching actuators. The pattern was named after a scientist who observed this pattern in thin-walled cylinders buckling under axial compression [9].

The Yoshimura pattern is particularly useful as a continuum actuator due to its torsional strength and ability to bend in two degrees of freedom. The design features a combination of horizontal valley and mountain folds, with diagonal mountain folds to create the collapsible bellows-like structure. To create a Yoshimura module in our project, we connect three folded patterns axially to maintain stiffness through a more uniform module. When connected in this module, the Yoshimura module features three bending degrees of freedom and one axial degree of freedom which provide a high degree of maneuverability and control for origami robot motion [10].

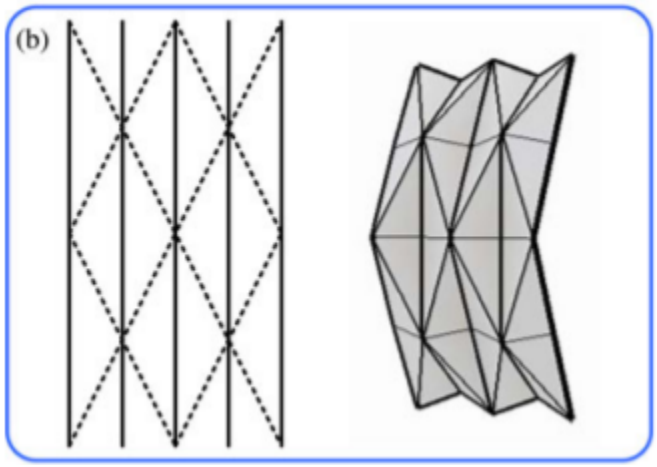


Figure 6: The Yoshimura Pattern [10]

2.4 Applications of the Yoshimura Module

WPI Soft Robotics Lab is a major proponent of the Yoshimura module in origami robots. One of their most notable achievements was the creation of a robotic snake, the OriSnake, which features four individually-controlled continuum modules mounted in series actuated via three N20 gear DC motors and nylon fishing line to compress the modules. As the motors rotate, the cables are retracted, bending the module in the direction of the cable and maintaining position through the axial stiffness of the pattern. For this design, the modules were laser cut from 7-mil thick polyethylene terephthalate (PET) sheets, which provide ample stiffness and compliance. The design is completely hollow, allowing for transfer of power and data connections within the body of the robot. Through precise control and feedback of the lengths of the cables, the researchers created a robotic snake that could perform lateral undulation and side-winding locomotion gaits to propel itself forward on a planar surface [11].

They then took the design one step further in the creation of the Salamanderbot, a continuum robot that can effectively traverse three dimensional environments, climb sharp inclines, and navigate through tight turns. It features a very similar design, applying the Yoshimura module for continuum motion but differs in its transmission. The robot features a gear transmission design for actuated wheels to generate high speeds and torque to overcome inclines. The robot also features radial symmetry, allowing the robot to turn over and continue its motion. This radial symmetry opens the question of traveling inside cylindrical surfaces, such as pipes and tubes [12].

2.5 Existing Pipe Inspection Robots

Pipe inspection robots are becoming incredibly popular due to their high potential in exploring pipe networks safely and efficiently. The most popular is the borescope, which is an optical tool inserted into pipe networks by operators as a non-invasive form of inspection [13]. Borescopes are categorized into two main categories: rigid scopes and fiberscopes [14]. Rigid scopes are non-flexible but feature increased visual clarity using mirrors and lenses to deliver an image to an eyepiece. Fiberscopes, on the other hand, feature high flexibility and manual operation.

Each type consists of an appendage with an illuminated focal point at the tip, allowing for remote inspection. To be suitable for typical HVAC scenarios, borescopes are waterproof and can withstand high temperature environments. Operators respond to video feedback at the point of entry into the pipe network via a display screen, so they can manually operate the borescope and direct it towards different areas in the pipe network.

Many pipe inspection robots take the form of crawlers, using tracks and wheels to navigate through pipe solutions. Deep Trekker specializes in these robots, with two models called the DT340 and DT320 Mini which are portable and battery-operated. The DT340 is the larger of the two, being able to inspect pipes as small as 8”. It is incredibly modular, allowing for variations in wheel kits, external manipulators, cameras, and more to expand its functionality. The robot features a forward facing camera to capture pipe internals and display

them on an external handheld controller. The DT340 is also submersible, being able to handle underwater pipes with regular maintenance [15]. However, the system requires a tether to display information on the handheld controller despite the onboard battery solution. The DT320 Mini, however, can explore pipes from 6” to 12” in diameter. Unlike the DT340, this robot features a manipulator to move a front-mounted camera to the center of a pipe, regardless of diameter [16]. The robot features a pan-tilt camera which can rotate 360 degrees to perform a full inspection.

Figure 7: Deep Trekker Pipe Inspection Robots (left: DT340, right: DT320 Mini) [15], [16]

However, other configurations of pipe inspection robots use three contact points such as the TriTrax robot from Eddyfi Technologies. This is more advantageous, as increased contact points allow for more traction and can allow the robot to travel through inclines and vertical pipes. The robot features three tracks that can be independently controlled to propel the robot through a pipe. Each track is mounted on a slider-crank system that operates within the diameter of the robot to allow the operational diameter of the robot to alter its size. This makes the robot adaptable to various sizes of pipes, as well as changing diameters which are common in applications where operators require controlling flow rate. The tracks do not feature magnets, as the expansion force generated by the slider crank system provides enough traction for the robot to propel itself forward. The robot operates between pipe sizes of 8” and 16” and like the robots from Deep Trekker, is tethered to provide camera footage to an external controller [4].

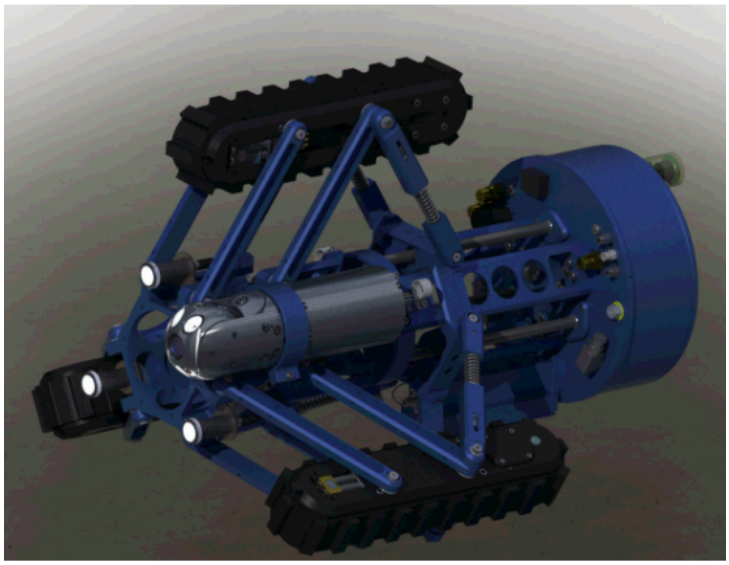


Figure 8: TriTrax Robot by Eddyfi Technologies [4]

Researchers at the Pukyong National University in South Korea developed an alternative design, which features two modules, an active module and a passive module. The active module features a lead screw design with a coupler that rides along the lead screw. The coupler is connected to three wheel modules via a slider-crank linkage mechanism. When the screw turns via an internal expansion motor, the coupler moves along the screw, changing the operational diameter of the robot. On the other end, the passive module operates in a similar manner. Rather than a lead screw, it utilizes a compression spring that passively maintains the diameter of the robot to be that of the pipe. The two modules are connected via a universal joint, which allows the robot to bend within 90 degree pipe elbows. It is important that the active mechanism is considered the front of the robot, as the active side can change its diameter to accommodate pipe size changes, while the passive mechanism follows through and maintains contact. The robot can operate between 12” and 20” pipes, making it a larger robot. This robot can operate wirelessly, utilizing a wireless camera and custom GUI to control the robot [17].

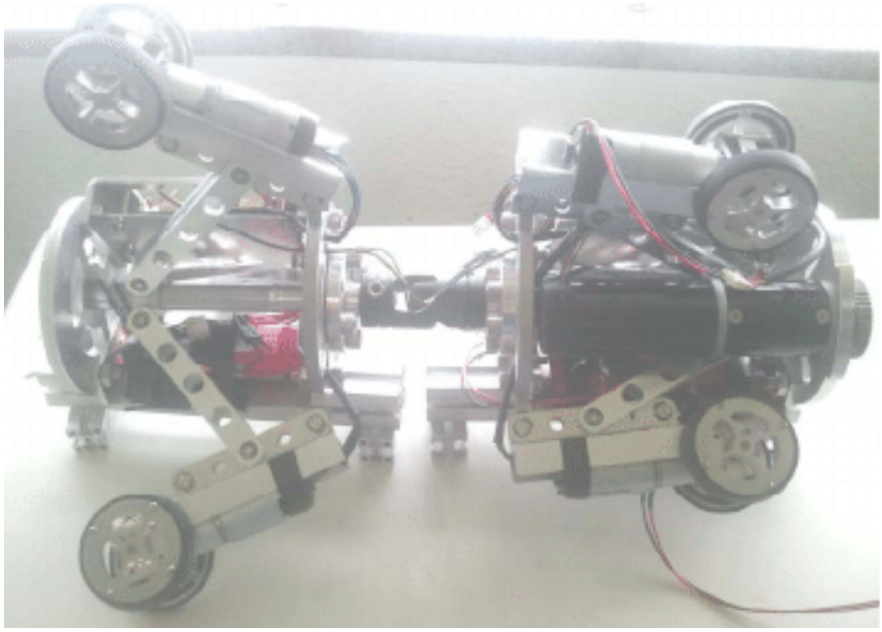


Figure 9: Pukyong National University Wheeled Pipe Inspection Robot [17]

Worm robots are becoming increasingly popular, such as GE's Pipe-worm (Programmable Worm for Irregular Pipeline Exploration). This robot is entirely soft using fluid-powered soft actuator muscles which inflate and bend, to grip onto pipe surfaces and propel itself forward like an earthworm. This robot features cockroach-like whiskers and built-on artificial intelligence that allow it to sense new environments such as junctions, pipe diameter, and its own orientation [18].

Figure 10: GE Pipe-worm (Programmable Worm for Irregular Pipeline Exploration) Robot [18]

However, not all pipe inspection robots feature active components. Lighthouse, from WatchTower Robotics, is a soft, tether-less robot that utilizes fluid flow within pipe networks to

explore and map potential leaks. The robot, shown in Figure 11, does not feature any form of locomotion but rather propels itself forward using fins to take advantage of passive pipe flow within a pipe. It features soft skirt sensors, which can sense potential leaks as the robot passes by using the suction force created by the fluid. Lighthouse then records the location of the leak and creates a virtual map with accuracy of one foot in all environments to help pipe operators perform repair in the right locations. Operators can simply drop Lighthouse into a pipe network at a T-junction and captured using a net in the same scenario [3].

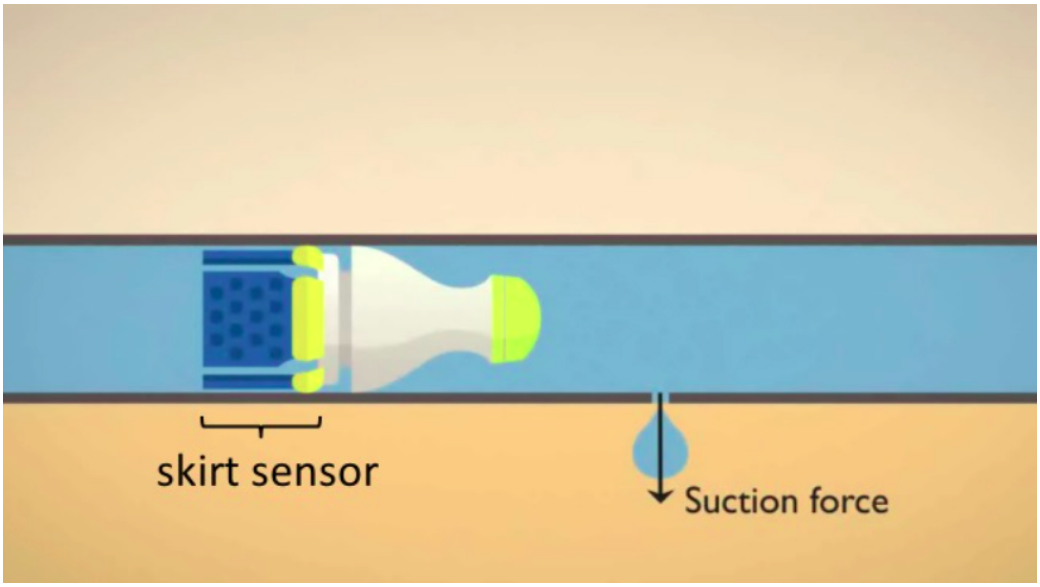


Figure 11: Lighthouse Operational Diagram [3]

2.6 Shortfalls of Existing Solutions

2.6.1 Shortfalls of Pipe Inspection Robots

While the various types of pipe inspection robots are suitable for their individual applications, they could be improved tremendously to overcome more challenging scenarios.

Borescopes, while one of the industry's most popular pipe inspection tools, are limited by their range. They feature a set length, therefore not permitting full inspection of a pipe network from start to finish. This requires inspectors to probe various sections of the network, making it more challenging to diagnose the location of a leak. For rigid scopes, operators can only inspect straight pipes, as the appendages cannot deform to pipe bends while for fiberscopes, the camera quality can make it difficult to pinpoint potential problems on pipe walls [14].

Crawler robots are one of the most effective solutions towards the inspection of entire pipe networks but are limited largely by their size and cost. For example, the robots offered by Deep Trekker, the DT340L and DT320, are only capable of inspecting pipes between 8-12" and 6-18", respectively. While the latter is more adaptable, neither robot can inspect smaller pipe sizes, commonly used in appliances. Additionally, both robots are very expensive, with the

cheapest costing $18,310 USD. These robots are also only limited to two contact points within a pipe, and do not feature diameter expansion mechanisms to overcome vertical pipes [15], [16]. The TriTrax from EddyFi looks to solve this issue but is still limited in the fact that it requires a tether for teleoperation from an external controller [4]. The robot developed by the researchers at Pukyong National University is the closest to a fully-operational pipe inspection robot that can overcome bends, vertical pipes, and diameter changes but is still lacking in its diameter range. The robot can only traverse diameters between 12” and 20” and due to its universal joint, it cannot overcome very small turn radii [17]. Additionally, these crawler robots may exert an excessive friction force against the inner surface of a pipe, causing internal damage to the pipe walls.

Worm robots, such as GE’s Pipe-worm robot, seem incredibly effective due to their soft bodies which can explore through non-uniform and unexpected terrain with their physical intelligence [18]. However, they are propelled using fluidic devices and pressure, therefore requiring a tether. This limits their capability, as they can only explore a network until the tether cable runs out. Additionally, the motion of worm robots is much slower than that of wheeled robots as they require the inflation/deflation of several muscle segments for propulsion.

Finally, passive robots such as Lighthouse are at a disadvantage because operators cannot control their motion [3]. The robots move as fast as the flow within the pipe and cannot remain stationary to analyze a potential leak further. Lastly, there is a flow threshold for these robots of 0.1 m/s, meaning that Lighthouse cannot inspect situations for low flow. Retrieving the robot can also be challenging, as inspectors must plan in advance to catch the robot within motion at the retrieval point.

2.6.2 Shortfalls of Salamanderbot

While the WPI Soft Robotics Lab did not design CLARA’s precursor, Salamanderbot, with tubular systems in mind, it features incredibly powerful technology useful towards pipe inspection. Its Yoshimura continuum module allows it to bend around variable turns, which can be useful in unconventional pipe networks with custom joints [12]. However, the robot does not feature dynamic diameter expansion and retraction, meaning it is only suitable for pipe diameters larger than the robot itself. While the robot features robust radial symmetry through its triaxial wheel symmetry, the toothed wheels would have trouble gripping onto pipe surfaces.

3. Project Strategy

3.1 Project Goal

The overarching goal of this project is to develop a robotic module capable of achieving tight bending radii for use in multiple pipe sizes for tetherless in-pipe inspection.

3.2 Objectives

The scope of this project is to develop a flexible robotic module that utilizes origami designs to maintain a flexible body capable of achieving tight bending radii, along with an expandable suspension mechanism which can be used in multiple pipe sizes. The CLARA MQP aims to study the capabilities of a soft robotic origami module in the inspection and exploration tasks in pipe networks too small, expansive, or dangerous for humans or existing standard technology to accomplish. Our work adapts and improves the existing Salamanderbot design, both mechanically and electrically, to make it more versatile for various types of terrain, with a focus on pipe systems. The goal for improving the robot includes making the robot smaller overall, including the capability to adapt to varying pipe diameters, providing a modular electrical control structure, and improving drive mechanisms with the addition of active and passive compliant mechanisms. The following project objectives were set, with the intention of completing all objectives by the conclusion of the project:

1. Create and test a fully functional module with the aforementioned capabilities.
2. Design and test at least three different types of active actuators for use in the variable diameter suspension mechanism.
3. Design and test at least one passive suspension mechanism.
4. Create a more inclusive electrical schema for projects outside of CLARA through the creation of smart motor drivers capable of controlling individual N20 motors.
5. Develop a control architecture that allows for wireless communication of autonomous as well as teleoperated control of the module.

3.3 Design Specifications

Following the objectives laid out in Section 3.2, by the conclusion of the MQP, the team intended to achieve the following features:

1. The total cost of fabricating a single module should not exceed $500.
2. The module will consist of an origami midsection.
3. The module will be able to travel within a minimum pipe size of 4” and a maximum pipe size of 6”.
4. The module will be able to travel at a minimum speed of 4 in/s.
5. The module will be capable of traveling through both concentric and nonstandard expansions and contractions of pipes within its specified size range.

6. The module will be capable of traversing steep inclines, as well as able to travel in completely vertical pipes within its specified size range.
7. The module will be able to maneuver through 90 degree bend radii.
8. The module will not require a tether for operation.
9. The module will use individual PCBs for the control of each motor, communicating with a main board facilitating communication and power distribution.
10. The module will make use of current sensors on all motors to detect real time current usage and avoid stalling of motors.

3.4 Approach and Timeline

3.4.1 Origami Body Design Process

We designed the origami body within the first quarter of the project. The team decided this was the necessary first step in the design of the CLARA robot, by reducing the overall size of the Yoshimura module from the Salamanderbot project. This process included adapting the folding pattern from the Salamanderbot, in terms of size and fold structure, and attempting to manufacture the body at various sizes by scaling the pattern down uniformly. The team placed a design constraint on the overall size of the module at 1.72” in diameter to be more adaptable towards smaller pipes. We found that at smaller sizes, manufacturing time greatly increased as the pattern became too tedious and inefficient to fold, while at larger sizes our design constraint was difficult to fulfill.

3.4.2 Mechanical Design Process

The design of the active and passive mechanisms occurred within the first quarter of the project. Initially, the team performed research to determine which types of expansion mechanisms existed, which were more space efficient, and the effectiveness of each design. The team created two types of mechanical mechanisms as proof of concepts operated by hand, and one mechanism for use with a vacuum pump. The team evaluated each mechanism using the aforementioned criteria, selecting two mechanisms for further testing.

Next, we designed and prototyped mechanical power transmission methods to complete the overall mechanical structure of the robot. The team first explored methods of connecting the active and passive mechanisms to the origami module with off-the-shelf fasteners, and then transitioned into the placement of motors for actuation. We developed an initial mechanical prototype of the robot in the first quarter of the project.

3.4.3 Printed Circuit Board (PCB) Design Process

Most PCB design occurred in the second quarter of the project. First, the team discussed several control architectures for the control of motors and intra-robot communication. The team discussed what communication protocols we wanted to use, how many motors we required to be controlled at once, and how we wanted to tackle PCB design, which neither of us attempted prior to this project. The team initially took inspiration from the Salamanderbot project, which

utilized a singular PCB for the actuation and control of the Yoshimura module motors and an external ESP32 and Featherboard to power the drive motors. The project then focused on the creation of a singular PCB which included onboard motor drivers for all motors on the robot. However, the team quickly realized this was inefficient in space, and we could disperse functions throughout the robot.

The next design focused on the creation of PCBs mounted on individual motors that would communicate to a central mainboard powered by an ESP32. The team quickly realized this was a more efficient design and could be adapted to many projects beyond the scope of the CLARA project. By migrating functionality to individual circuit boards on motors, future projects can skip including functionality on to custom hardware and instead use these PCBs. The team took inspiration from a previous project called SAMI, which created a smart motor driver to control N20 motors [19]. However, the driver went out of production several years ago, so the team attempted to reverse engineer the circuit. Due to the chip shortage, all components on the circuit were also out of stock, prompting an entire redesign of the circuit.

The team took this opportunity to increase the functionality density of the PCB, adding components such as quadrature encoders and current sensors to be more adaptable for future projects. This saw the creation and testing of several iterations of the circuit, considering various communication protocols to communicate to a mainboard ESP32 module.

3.4.4 Control and Communication System Development Process

Towards the end of the second quarter and through the third quarter of the project, the team heavily focused on the programming of communication and control protocols for the CLARA robot. After becoming more confident with circuit design and creating a PCB that looked reliable, the team began testing all communication protocols such as WiFi, I2C, and Serial to ensure full functionality of the circuits.

The team initially validated all functions of the Smart Motor Driver PCBs, testing individual components such as the hall-effect latches (encoders) and current sensor on each board, confirming the expected output first on an oscilloscope and later using Serial output data. Using the output from the hall-effect latches, the team created a control schema using PID control to adjust speed and position of each motor.

We then verified all potential uses of I2C, being able to request data from the Smart Motor Drivers for the mainboard in future scenarios for reading current sensors and encoder values. Next, the team verified being able to send one-way commands through I2C without any intent of getting data back for applications where a motor is simply commanded to move.

We verified functionality with multiple Smart Motor Drivers connected at once (each individually addressed) to verify that we did not overload the I2C bus with multiple motors connected and commanded each motor to move individually.

Next, the team verified WiFi communication using the ESP-NOW protocol, with the onboard TinyPICO as the receiver and an external TinyPICO connected to Serial input as the

sender. We then made the communication bidirectional, to send data from the mainboard TinyPICO to the external board and read Serial output data.

After we validated the communications protocols, the team connected an external USB gamepad to make controlling the robot more intuitive during demonstration. This required a custom protocol to read inputs and translate them to a TinyPICO board using Serial. After deciding on a control schema to read inputs and determine how we wanted to control the robot, the team was able to control the robot precisely using variable input.

As we tested the robot within pipe systems, we addressed several mechanical issues such as overall diameter, compliance, and traction to propel ourselves through the pipe and enhance the performance of the robot.

3.4.5 Testing and Validation Process

In the second quarter of the project, the team created a testing matrix for both short-term PCB design validation and complete robot testing for the third quarter, considering the design specifications and goals we created in the first quarter. Additionally, the team categorized each entry in the matrix as either a test or demonstration, planning for the fourth quarter of the project and the demonstration phase of a functional robot. We designed an obstacle course using various pipe components to create an all-inclusive benchmark for the robot.

Requirement/Spec	Test Case Description	Means of data collection	Testing Faults
Robot must be able to travel in a straight line	Place robot on horizontal, flat surface, home robot, and drive straight	motion capture or slow-motion video	1) homing robot is necessary to produce good results, surface could alter results
Robot maximum speed must be __ m/s	Utilize multiple flat surfaces with varying coefficients of friction (tile, cardboard, acrylic sheet, carpet) and propel the robot forward for a straight-line distance of 2 feet. Record the time it takes for the robot to travel across the surface for several trials, and record the average linear speed	motion capture or slow-motion video	1) homing robot is necessary to produce good results, surface could alter results
Robot must be able to traverse inclines	Use a smooth surface and gradually increase incline until robot is no longer able to climb	measurement tools- protractor	this is surface dependant- so determining the surface(s) to use is a good question
Robot minimum turning radius must be __ mm.	Run motors to bend body as much as possible and process results	Motion capture or imaging we can analyze	Will be dependent on robot length, different lengths will produce different results - can find the absolute minimal radius and record that result

Figure 12: Testing Matrix Snippet

In the third quarter of the project, the team built the obstacle course for testing. Using commonly found PVC pipe from local hardware stores as well as clear PVC pipe, the team built a demonstration course to show CLARA's functionality in various pipe network configurations.

3.4.6 Timeline of Milestones

As this project took place over an entire academic year, the team has condensed the major milestones achieved throughout the year in Table 1.

Notable Milestones:	**Time Accomplished:**
Project Proposal	4/7/2021
Yoshimura Module Redesign + Fabrication	9/7/2021
First Active Suspension Mechanism Design	9/19/2021
Passive Suspension Mechanism Design	9/28/2021
PCB Control Schematic	9/28/2021
Second Active Suspension Mechanism Prototype	10/5/2021
Iteration 1 Smart Motor Driver PCB + Mainboard	10/5/2021
Active + Passive Suspension Mechanism Fabrication and Assembly	10/5/2021
Third Active Suspension Mechanism Design	10/12/2021
Third Active Suspension Mechanism Fabrication and Assembly	11/5/2021
Iteration 1 Smart Motor Driver + Mainboard PCB Fabrication and Assembly	11/5/2021
Iteration 2 Smart Motor Driver	11/12/2021
Iteration 3 Smart Motor Driver	11/19/2021
Iteration 2 Mainboard	11/23/2021
Flashing Smart Motor Drivers using SPI	11/23/2021
Iteration 3 Smart Motor Driver PCB, Iteration 2 Mainboard Fabrication + Assembly	12/3/2021
PCB and Robot Testing Matrix Creation	12/3/2021
Successful communication on Smart Motor	12/10/2021

Drivers using I2C, Serial, and WiFi	
Successful spinning of motor in two directions with variable speed	12/16/2021
Iteration 3 Mainboard	12/20/2021
Iteration 3 Mainboard Fabrication + Assembly	1/8/2022
Validation of sensors (encoder + current sensor) on Smart Motor Drivers	1/12/2022
Successful driving of robot using external gamepad	1/20/2022
First successful test of robot in a horizontal pipe system	2/2/2022
Validation of I2C and encoder Interrupt Service Routines working simultaneously	2/10/2022
First successful test of robot in a vertical pipe system	2/24/2022
PID control of motor speeds	3/1/2022
First successful test of robot in a bend	3/7/2022
PID position control	3/12/2022

Table 1: Major Milestones

4. Design

4.1 Mechanical Design

The mechanical design of CLARA consists of three major components - the Yoshimura origami module, the active front suspension, and the passive back suspension. Each of these individual systems were designed, prototyped, tested, and integrated to yield the final design.

4.1.1 Yoshimura Module Design

The Yoshimura model portion of CLARA is comprised of two major parts: the origami body and the cable actuators that drive the body. The origami body of the module enables the robot to expand and contract its length and bend its body into arcs for navigation around corners, climbing inclines, and bridging gaps.

4.1.1.1 Folding Design

The body of CLARA is an adaptation of the Yoshimura origami folding pattern described in previous sections. The Yoshimura crease pattern is advantageous in this application as it provides controllable linear compression with increased torsional stiffness to reduce twisting of the robot.

In developing the final crease pattern for CLARA, we developed and evaluated several types of patterns. For manufacturing these prototypes, the patterns were laser cut out of thin sheets of clear PET (polyethylene terephthalate) sheets of varying thicknesses. We compared many versions with the pattern used in Salamanderbot, with versions varying the body size, PET thickness, and laser settings such as raster and vector cuts. The purpose of producing these prototypes was to determine which parameter set would result in the smallest possible size of the body with the highest efficiency in folding and cleanest laser cuts.

In addition to varying the factors mentioned, a new pattern was also developed based on the research from the Korea Advanced Institute of Science and Technology, which tested the rigidity, foldability, and benefits of using triangles that are both right and isosceles in particular for the Yoshimura cylinder, rather than just traditional isosceles triangles [20]. This proposed folding methodology was reported to decrease the occurrence of wrinkles in the fully folded configuration, require less force to contract the mechanism in its folded configuration, and increase the efficiency of folding. Researchers made an attempt at producing a pattern following this methodology but it did not increase folding efficiency.

For the final design, we mostly maintained the origami body from the previous design of Salamanderbot. However, we made slight tweaks to lengthen the pattern and decrease the radial size as shown in Figure 13**.** Ultimately, the best combination of the varying factors was found to be 7 mil PET sheets with vector cuts in a 11.36” (288.54 mm) by 2.08” (52.83 mm) flat pattern, resulting in a folded body length of approximately 4” (Figure 14). In comparison, the

Salamanderbot's body is 10.255" (260.48 mm) by 3.47" (88.14 mm) in its flat pattern, resulting in a folded body length of 5.5" (139.7 mm).

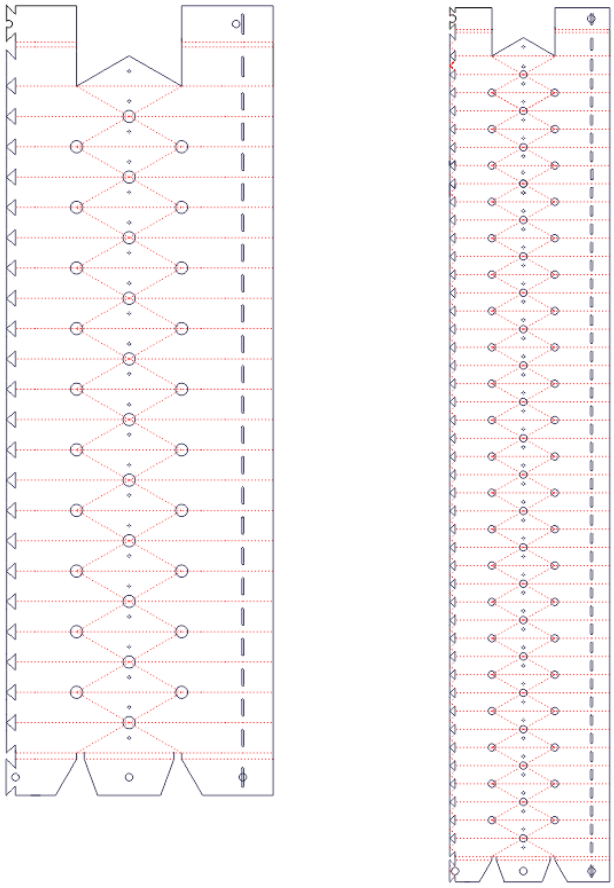


Figure 13: Comparison of Salamanderbot folding pattern (left) and CLARA pattern (right)

When three of the flat patterns are combined in 120 degree increments circumferentially using a set of slot-key attachments, they form a triangular module with the long axis being vertical.

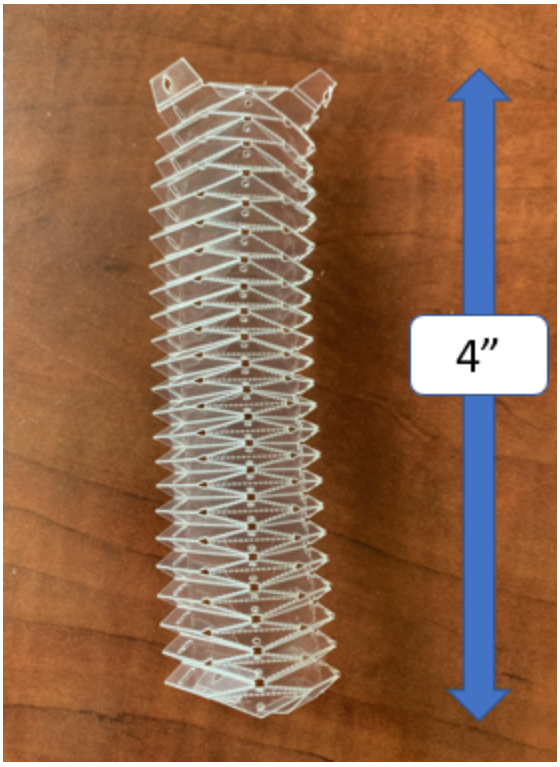


Figure 14: Final Pattern Size in Folded Configuration

A compatible mounting plate, as shown below in Figure 15, was created for simple attachment of diameter expansion mechanism modules, without disassembly of the Yoshimura module. Three slots were created to allow the Yoshimura module flaps to fold through the plate and be secured on the opposite side using an M3 screw.

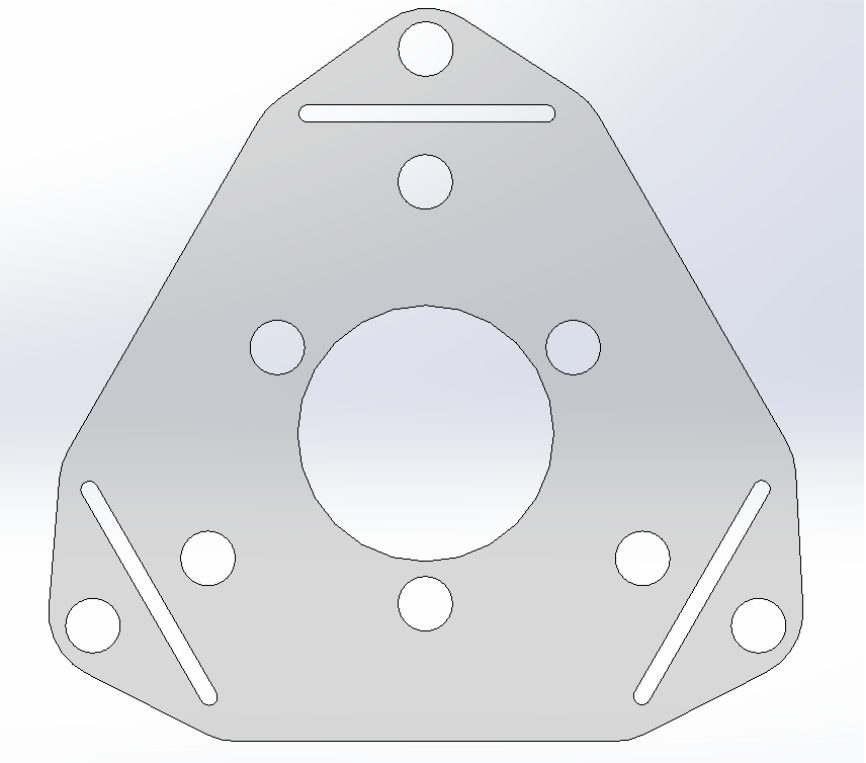


Figure 15: Yoshimura Mounting Plate CAD Rendering

4.1.1.2 Cable Actuator Design

In the design of Salamanderbot, which this design was based upon, actuation of the origami body was accomplished through tightening and loosening cables on each of the three edges of the body via N20 motors with brass winches attached (Figure 16). We determined that this was a very effective method for actuation and control of the body, so the cable-driven bellows concept was carried over to CLARA with some changes to accommodate the new, smaller body. Initially, the robot used Tuf Line, a waterproof braided thread that can withstand upwards of 65 lb of tensile force [21]. However, we switched to PowerPro 50 lb microfilament braided fishing line [22] to make manufacturing simpler, as this cable is thinner and can be threaded with a typical sewing needle.

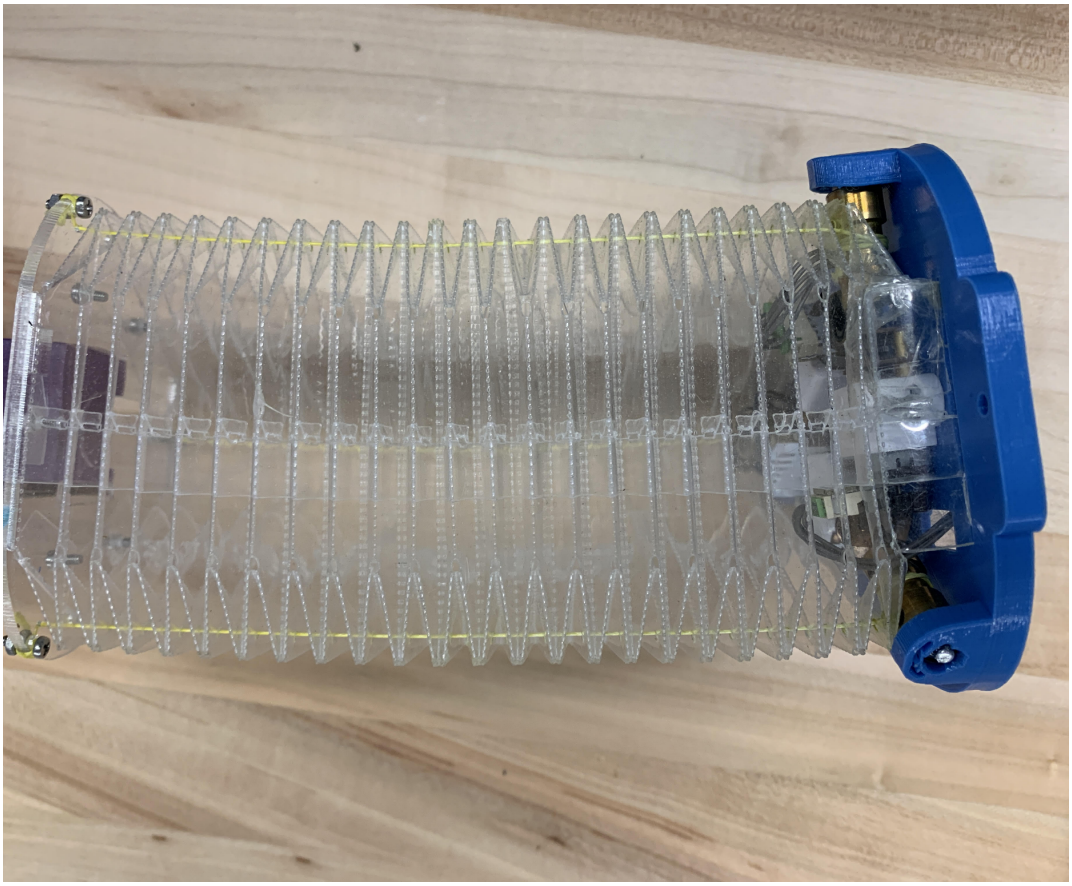


Figure 16: Salamanderbot Cable Drive Design

We used the same N20 motors, mounts, and brass winches (spools) but were resituated in a triangular pattern (Figure 17) to fit them spatially within the cross section of the new body size. A piece of laser cut acrylic was used as the base, and paracord used for the cables themselves.

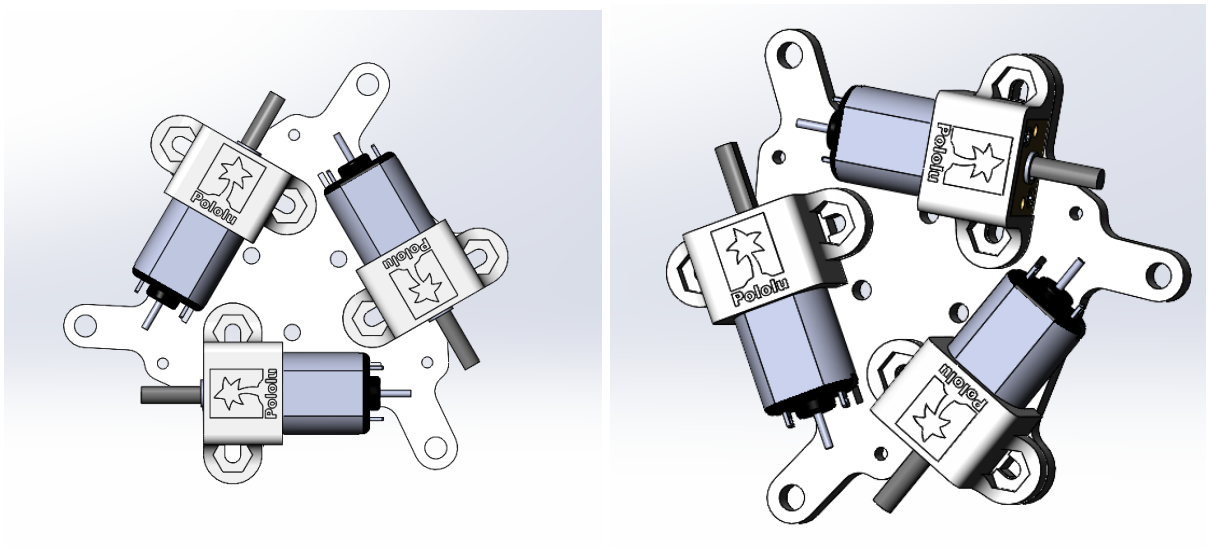


Figure 17: Triangular Layout of Cable Motors

Later, a revised acrylic plate was created to allow electrical wires to pass through the center of the robot, saving space and reducing chances for wires getting pinched between the robot and pipe walls. Additionally, the holes for the cable were extended outward, to limit fatigue and prevent pinching of the cables.

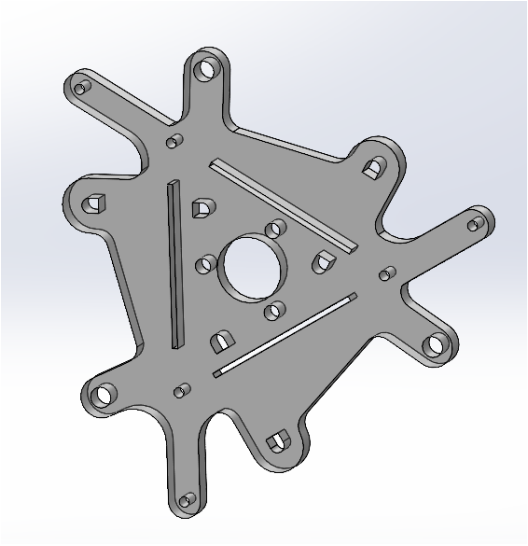


Figure 18: Revised Acrylic Plate CAD Rendering

4.1.2 Diameter Expansion Mechanisms

Objective #2 of the CLARA project was to create multiple diameter expansion mechanisms and evaluate their effectiveness for use in a pipe inspection robot. Below are three of the designs that were designed and prototyped before two were selected for future testing.

4.1.2.1 Slider-Crank

This mechanism was designed to be a precision controlled diameter expansion mechanism using rigid components, similar to the team from Pukyong National University. The mechanism features three identical linkages (Figure 19), joined together using 3mm D shafts and shaft collars. The lower link is a simple straight link that features several 3mm holes for the attachment of driven or non-driven wheel modules using M3 fasteners. The upper link takes the shape of a "Y," and features a large gap to maximize rotation around the lower link, increasing the overall diameter variability of the robot. The upper and lower linkages were initially both 3D printed using PLA but with further investigation, the team found desirable compliant properties when printing the upper link from NinjaFlex TPU, a flexible thermoplastic filament.

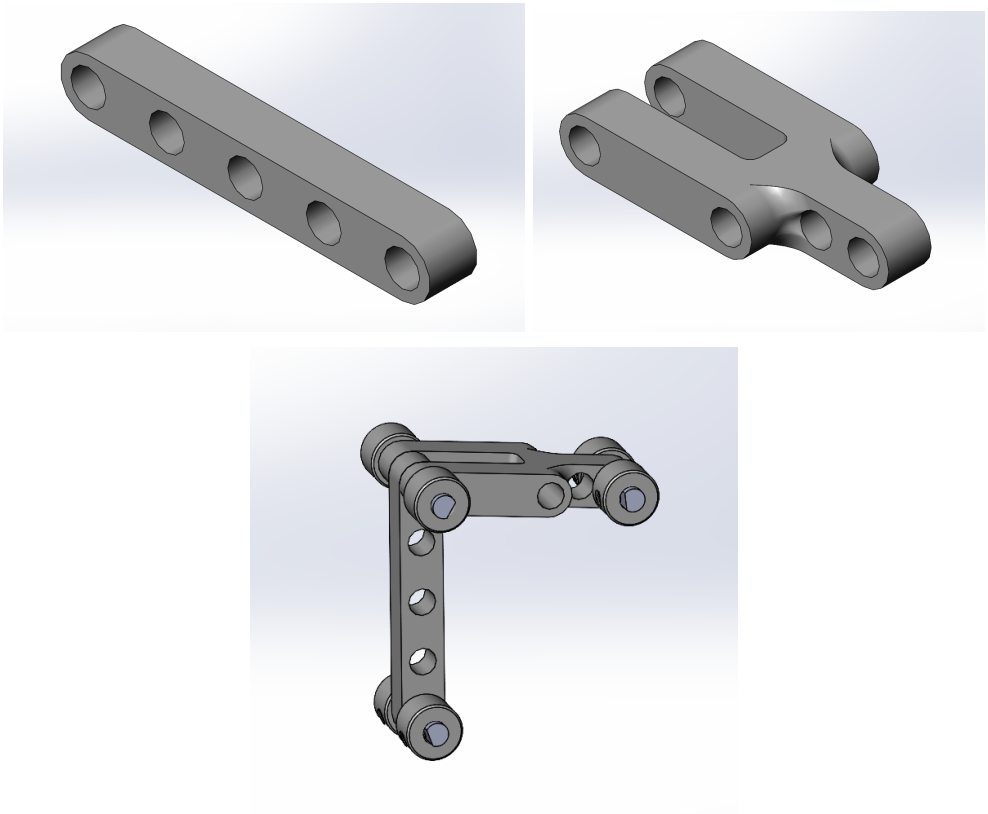


Figure 19: Lower Link, Upper Link, and Link Assembly CAD Renderings

Next, we created two connection points to join the linkages to the robot. A baseplate (Figure 20) takes the shape of the Yoshimura module cross-section and expands outwards triangularly to provide three attachment points for the lower linkages using 3mm D shafts and shaft collars, along with three internal mounting holes for attachment to the laser cut plate attached to the Yoshimura module and a hole for a ¼"-20 bore bearing. A triangular lead-screw rider (Figure 21) was created to connect the three upper links to a ¼"-20 lead screw, thus constraining their motion linearly. We then pressed an appropriate heat-set threaded insert into the center of the lead-screw rider, to allow for tolerance fitting of the rider on a lead screw.

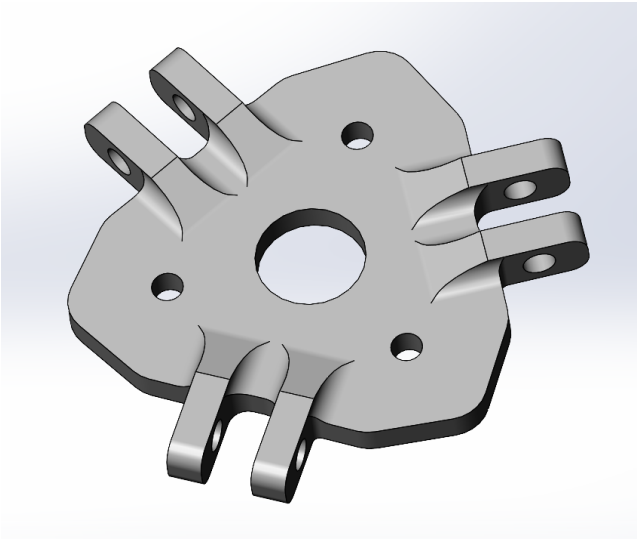


Figure 20: Slider-Crank Baseplate CAD Rendering

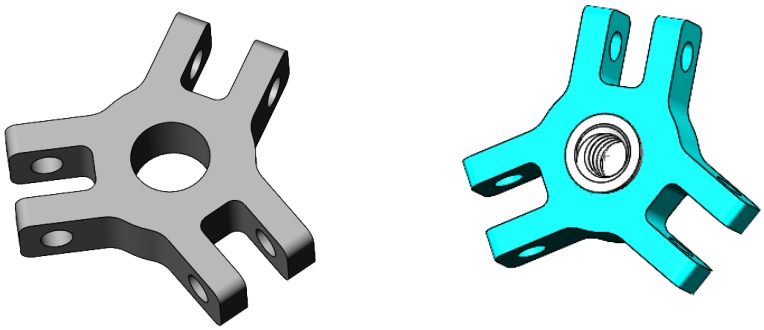


Figure 21: Lead-Screw Rider CAD Rendering with and without Heat-Set Insert

Shown below in Figure 22, the completed mechanism features the aforementioned components connected to a central ¼”-20 lead screw. When the lead screw is turned, the lead-screw rider translates along the lead screw, changing the position of the linkages and effectively changing the outer diameter of the robot. To characterize the system and achieve desired maximum and minimum diameter values, we altered the sizes of the links to achieve a wide range of motion and diameter variability. In this configuration, the robot features a minimum diameter of 3.59” and a maximum diameter of 5.23”. To control the outer diameter, the robot was equipped with an N20 motor mounted directly beneath the baseplate, hidden within the Yoshimura module. The team determined this placement would be advantageous, as we would save space by utilizing the Yoshimura module’s hollow properties. This motor would then rotate the lead screw, actuating the diameter change of the robot. A flange (Figure 23) and press-fitting shaft coupler (Figure 24) were created to clamp the motor to the laser cut plate, constraining its motion and reducing effects of vibration using three M3 screws and nuts.

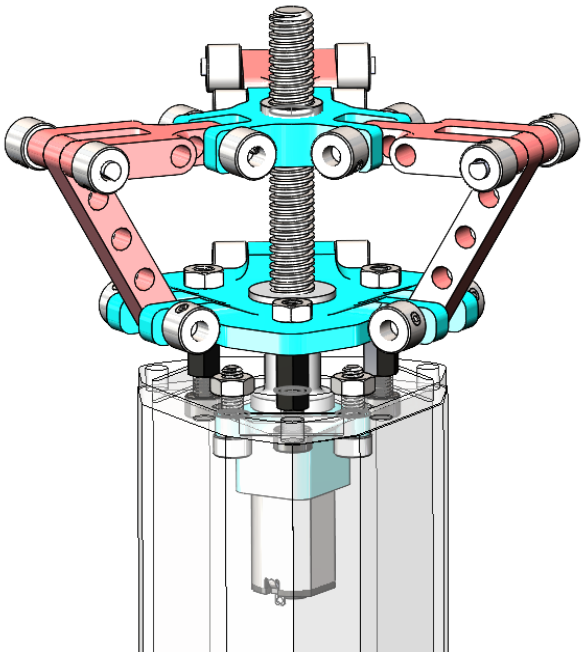


Figure 22: Slider-Crank Mechanism CAD Assembly Rendering

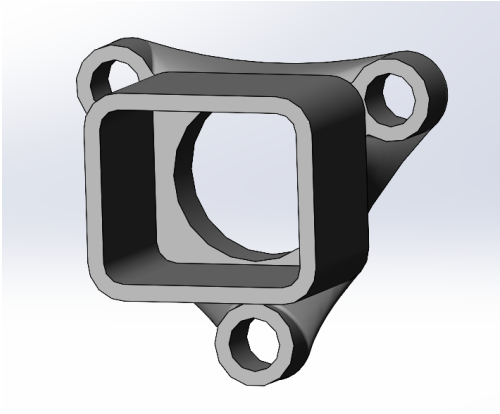


Figure 23: Diameter Expansion Motor Flange CAD Rendering

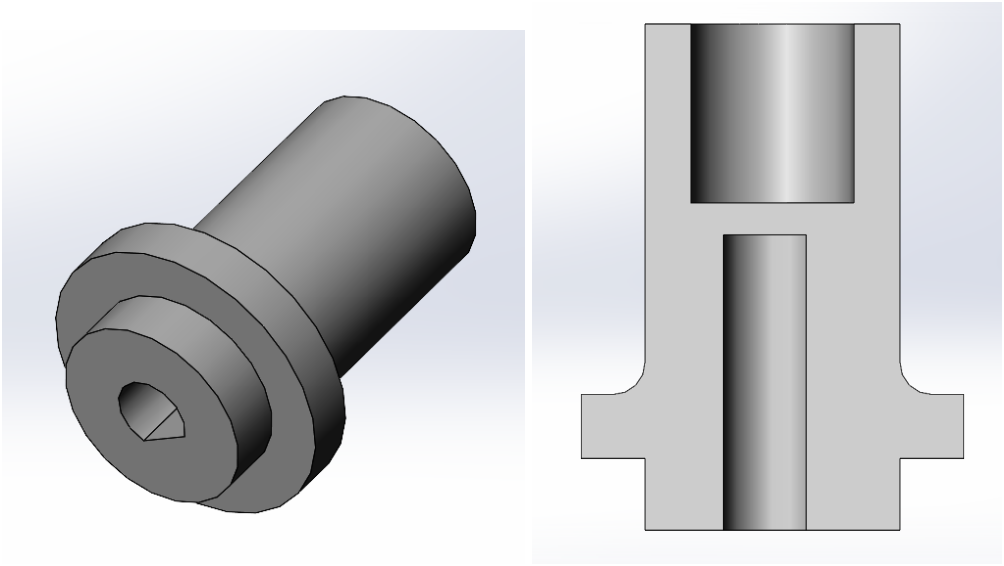


Figure 24: Shaft Coupler CAD Rendering and Section View

4.1.2.2 Cam-Slot

The cam-slot mechanism was designed to be an alternative to the slider-crank, and an attempt at minimizing the number of components and vertical space taken up by the mechanism. The mechanism features three sliding components (Figure 25), which feature mounting holes for the wheel modules and pins to interact with the cam module. The slides were 3D printed from PLA. These slides are free to move within a custom housing (Figure 26), which constrains their motion into linear motion using toleranced channels. The housing is attached to the laser cut plate using three M3 mounting holes and features a ¼”-20 bore bearing to allow for a shaft to rotate.

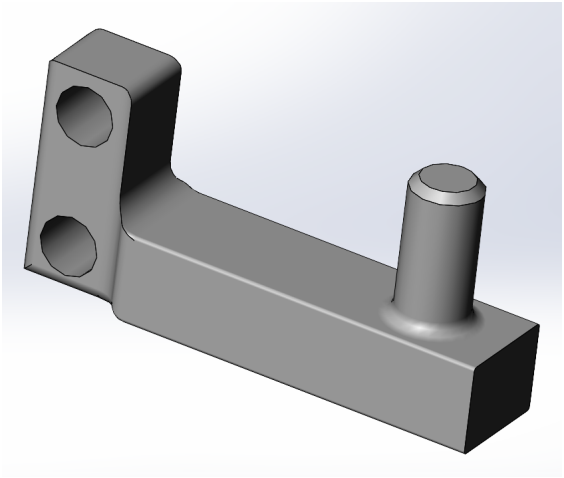


Figure 25: Cam Slide CAD Rendering

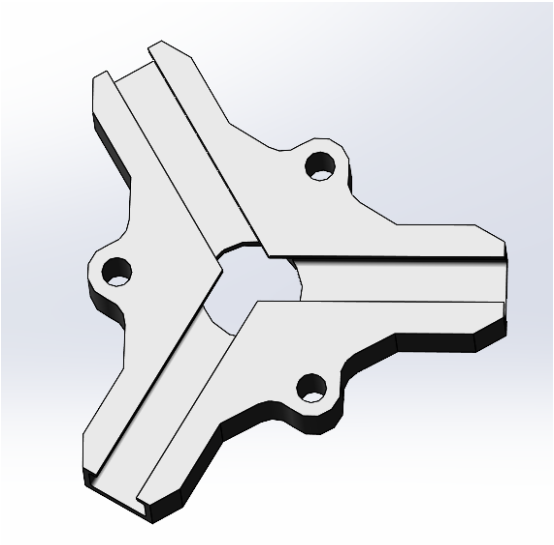


Figure 26: Cam Slide Housing CAD Rendering

To complete this design, the team created a thin disk with three arched cam profiles (Figure 27). This disk was placed directly on top of the housing, with the pins of each slide interacting with the inside of the arched cam profiles (Figure 28). This setup uses the same configuration for the mounting of the N20 motor within the Yoshimura module. When the motor rotates the central shaft, the cam disk rotates and forces each slide to move along the arched profiles, expanding the diameter of the robot. With this configuration, the robot features a minimum diameter of 4.38” and maximum diameter of 5.16”.

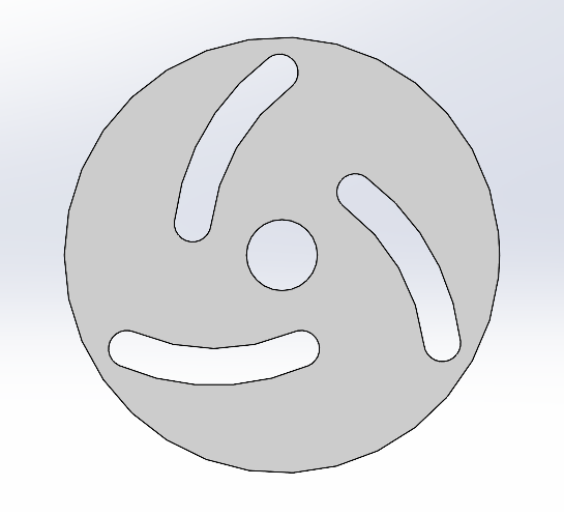


Figure 27: Cam Disk CAD Rendering

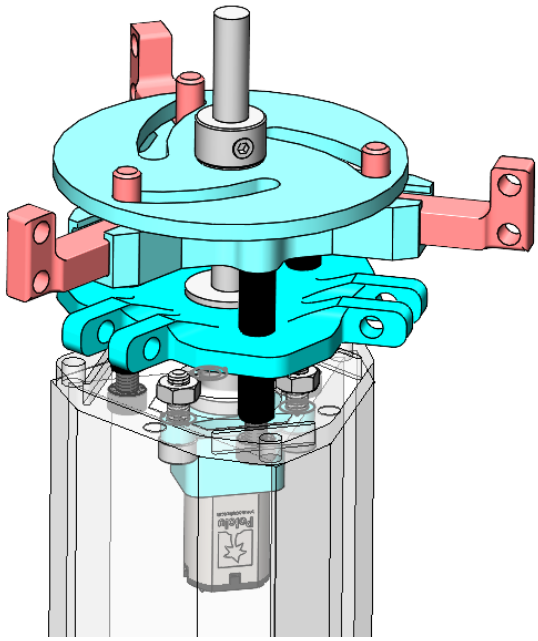


Figure 28: Cam-Slide Mechanism CAD Assembly Rendering

4.1.2.3 Pneumatically Actuated Origami Muscles

In addition to the two active suspension models previously mentioned, the team also felt it would be valuable to pursue at least one other type of non-linkage mechanism for achieving the desired motion. This culminated in the design and testing of a pneumatically actuated origami muscle that could be used for each smaller transmission. The benefit achieved with a pneumatically actuated method is that, if functional, it would eliminate the need for any sort of lead screw or linkage and could allow for the independent actuation of each wheel without the

addition of extra motors to do so. Additionally, it offered the opportunity to reduce the length of rigid section required for the mechanism, as it would eliminate the need for a central motor or lead screw.

to actuate each leg of the suspension individually, the piping and instrumentation diagram shown in Figure 29 was developed. In this configuration, this actuation could be accomplished with two pumps and three valves. For these pumps, small 2.5 liter-per-minute air DC motor pumps were used. These pumps were selected on the basis that the flowrate could be adjusted by varying the PWM signal, and that they exceeded the maximum flow that would likely be needed to inflate three muscles at once. Additionally, these pumps were able to provide both the function of a positive pressure pump and a vacuum pump in a single package- which would be required for actuation of the muscle. For the valves, small 6 volt three way valves were used with functionality similar to a pneumatic relay. To connect these elements, 2.5mm silicone tubing was used, as it was rated for well above the pressure requirements of the system.

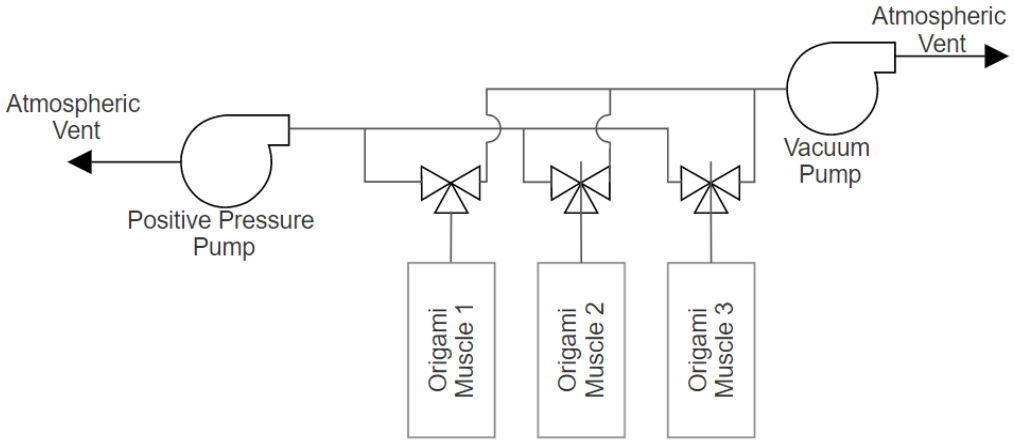


Figure 29: Origami Muscle Piping and Instrumentation Diagram

To prototype and test the concept of the origami muscle, a simple setup with two pumps, one three-way valve, and a single muscle was used.

For the muscle portion of the actuator, we used the same pattern for the main body with slight modifications in size and length. This origami inner portion would allow for rigidity and linear actuation in a single direction, rather than simply having a balloon that would inflate and deflate in all directions. Then, we encased the inner muscle in a thin thermoplastic polyurethane (TPU) plastic jacket that would allow for an airtight seal so the mechanism could be inflated and deflated as desired. This TPU jacket was made by simply taking sheets of TPU, fitting them to size around the origami section, and heat sealing the jacket around the origami section to eliminate leak points. For the prototype, two TPU jackets were added to the muscle to allow for operation in the event of a leak in one of the jackets. to connect tubing to this jacket, a plastic luer was perforated through and screwed into the base of the muscle where tubing could then be connected. The extended and contracted state of this origami muscle can be seen in Figure 30.

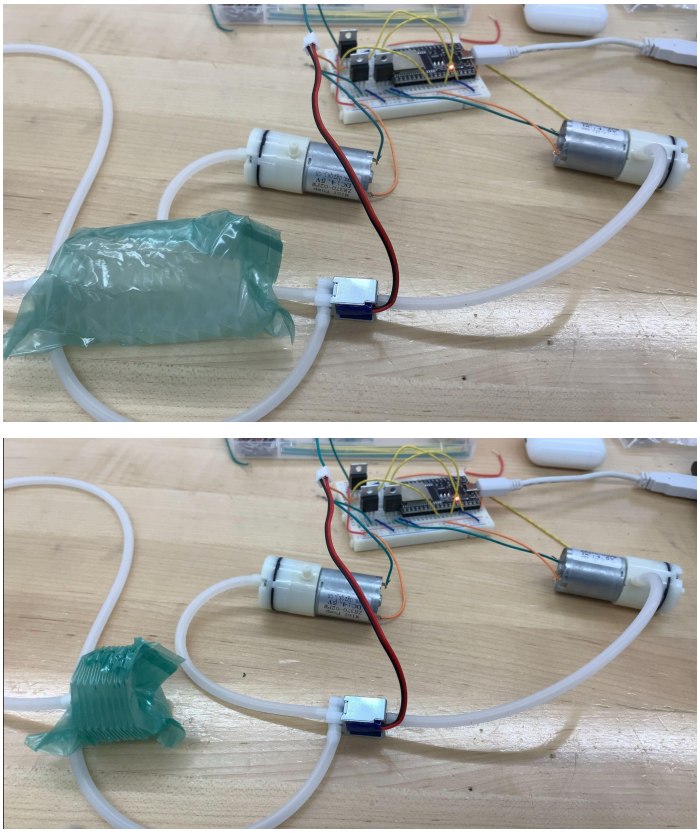


Figure 30: Extended (top) and Contracted (bottom) state of Origami Muscle

While initial testing of this mechanism seemed to indicate that the actuator could provide a promising solution with regards to the active suspension problem, it was ultimately discarded as a viable solution due to a number of concerns. For example, avoiding leaks in the plastic jacket that would inhibit proper actuation of the muscle was a recurring problem. The jacket was flimsy and easy to perforate, which would likely be an issue within the environments in which the robot is intended to operate. We also attempted a version of the muscle that did not require the jacket but instead utilized glue to seal leak points but was deemed unsuccessful early on, as the leaks could not be effectively sealed with the materials at hand. In addition to the aforementioned issues, there were also concerns regarding the rigidity of the mechanism, which appeared to be too compliant in its extended state and would be unlikely to be able to support the weight of a transmission attached to the end. Furthermore, it was still a concern that it would be difficult to fold the links at the sizes required, as the prototype muscle was already much larger than what would be needed.

Lastly, additional concerns were vetted regarding the use of the pumps and valves in this system. While the pumps and valves were effective, they were each larger than any of the single N20 motors used in other designs. Additionally, the pumps generate a large amount of vibration in use, which could cause many issues not only mechanically but could also greatly impact sensor readings by generating excess noise. Ultimately, while a pneumatically actuated

mechanism was valuable and informative to explore, testing was not continued further for the aforementioned reasons.

4.1.2.4 Passive-End Design

On the passive side of the robot, we applied a very similar design to the slider-crank. However, instead of an actuated lead screw, a compression spring and modified lead-screw rider were used. Shown below in Figure 31, the modified lead-screw rider features an increased bore to allow for the compression spring to sit below a ball bearing rather than a heat-set threaded insert. Shown in Figure 32, the completed assembly features a shaft collar above the rider, limiting its motion and preventing the spring from deflecting.

Figure 31: Modified Lead-Screw Rider CAD Rendering

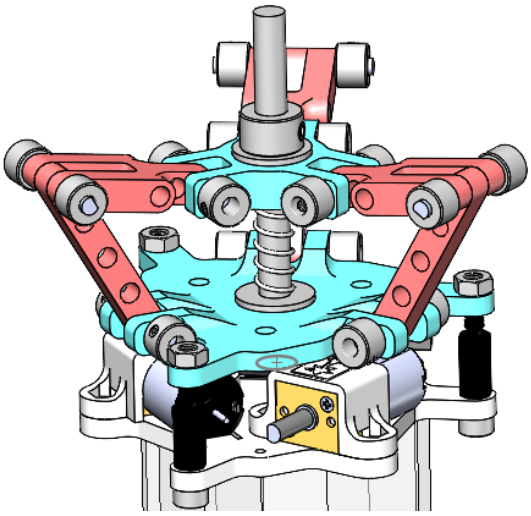


Figure 32: Passive-End Design CAD Assembly Rendering

4.1.3 Compliant Linkage Design

We quickly realized that rigid diameter expansion mechanisms, while reliable, are limited in the range that they can expand. We found that there was an opportunity to add additional compliance in the slider-crank mechanism by substituting the upper links with compression springs. However, we quickly realized that substituting a compression spring for an object that is free to rotate causes problems, such as innate buckling within the spring. To solve this problem, the team took inspiration from the Salamanderbot's flexible tendons, and created compliant upper links. We fabricated these links from a flexible/stretchable material, in our case 3D-printed NinjaFlex TPU (Shore Hardness 85A) to add additional compression into the system.

To maximize compression, the team experimented with several designs: a chevron pattern, a spring pattern, and a filled-in pattern. The team found initial success with the chevron pattern, shown below in Figure 33, in increasing the maximum compression of the system but faced reliability issues in the chevrons being structurally weak. After trying various 3D printer settings, such as changing retraction, printing temperature, and printing speed, the team determined that the chevron pattern and thus the spring pattern would not be reliable for the size of the application. These patterns could be very useful for larger applications where more force can be applied and more material exists between the chevrons and spring patterns.

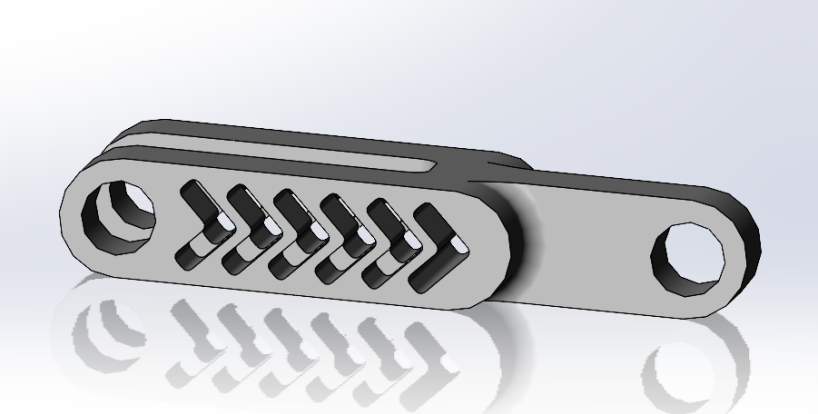


Figure 33: Chevron Pattern Compliant Link CAD Rendering

Following the reliability issues posed by the chevron and spring pattern compliant links, the team attempted printing a complete rigid link shown in Figure 19, using TPU instead of PLA as the material. This worked well, adding upwards of 0.5" of compression into the system to decrease the robot's minimum diameter. Like compression springs, the compliant links buckle upwards, causing additional compression when necessary. However, this buckling is more controlled than compression springs, as the links can still freely rotate around the central lead-screw rider and maintain stability within the system.

We found through extensive testing that when the compliant links were used on the active suspension side, they caused a torsional load that impacted the traction of the wheels on

inner pipe surfaces. Therefore, we replaced the compliant links with rigid links on the active side but kept them on the passive side as they provided the necessary compression to maintain robot position within a pipe.

4.1.4 Transmission Modules

4.1.4.1 First Iteration

To provide the robot with the ability to move freely without pipe systems, transmission modules were created using N20 motors, 3D printed bevel gears, and 32 mm wheels. Shown below in Figure 34, the transmission module assembly features a singular N20 motor surrounded by a clamping mount, mounted to the lower link using two M3 screws. Attached to the motor is a 14 tooth bevel gear which meshes with an identical bevel gear mounted on a horizontal 3mm shaft. This design was used to have wheels on both sides of the transmission module, therefore increasing the number of contact points within a pipe.

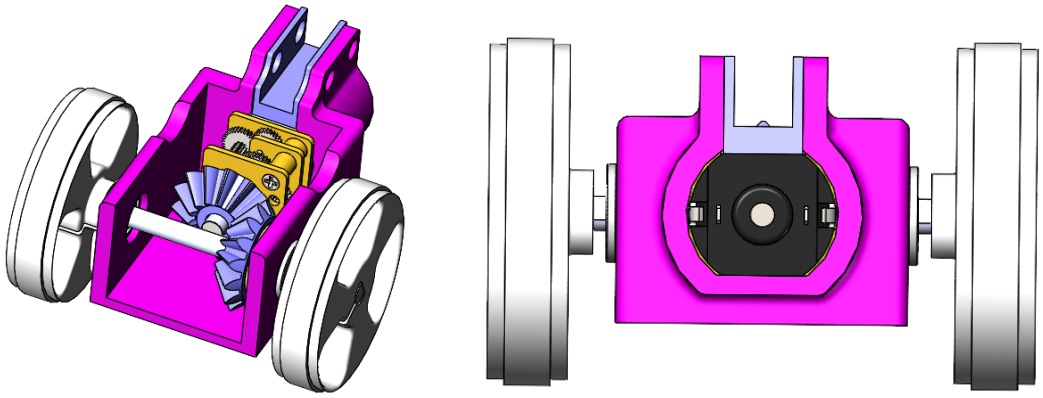


Figure 34: Active Transmission Module CAD Assembly Rendering

On the passive end, a very similar design was used. Since the passive end featured trailing wheels, a motor was not required, and thus only the clamping mount, horizontal shaft, and wheels were retained as shown in Figure 35.

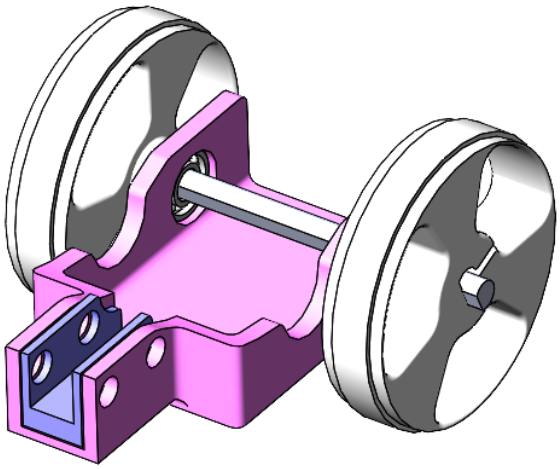


Figure 35: Passive Transmission Module CAD Assembly Rendering

4.1.4.2 Second Iteration

In the second iteration, the team realized that they could drastically improve the amount of space occupied by the active transmission modules. This issue was brought up when the team attempted to flash the Smart Motor Driver PCBs onboard the robot and found that the contact pins were blocked by the transmission module shell.

Therefore, we reduced the amount of space the shell occupied while maintaining functionality, as shown in Figure 36. To ensure the cable connector had enough space to fit, the team cut out a small slot in the side of the motor mount, ensuring that the clamping force required to keep the motor in place would be maintained.

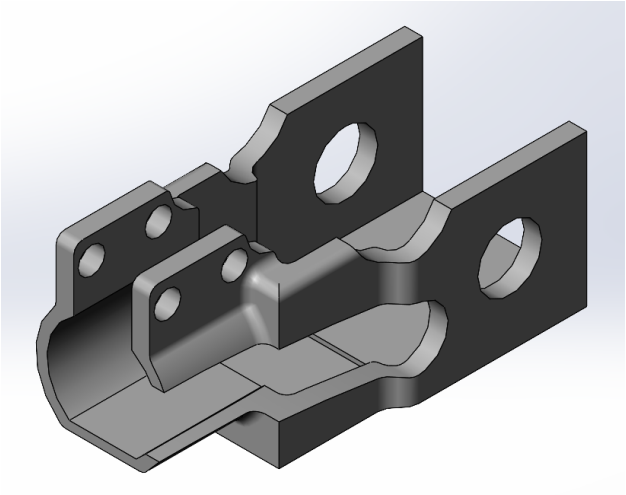


Figure 36: Iteration 2 Active Transmission Motor Mount CAD Rendering

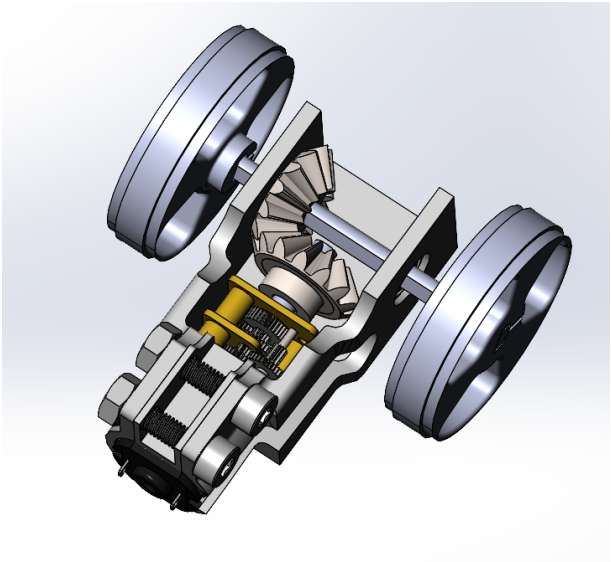


Figure 37: Iteration 2 Active Transmission Module CAD Assembly Rendering

4.1.5 Compliant Wheels

4.1.5.1 First Iteration

After experimenting with printing NinjaFlex TPU to create compliant links, the team became interested in developing wheels with increased traction to reduce slip within pipes and reduce motor current draw through compliant structures. Using a compliant wheel allowed us to move compliance from the links on the active side to the wheels, maintaining diameter variability and reducing torsional load.

In the first iteration of the compliant wheel, the team expanded the size of the 32 mm Pololu wheel to a 44 mm compliant wheel. The tire on the wheel features a honeycomb pattern around the wheel to allow for compression and diameter reduction of the wheel. This increases the amount of normal force on the sides of the pipe walls, therefore increasing the friction of the system. The wheel also has extended treads to increase the grip on pipe surfaces.

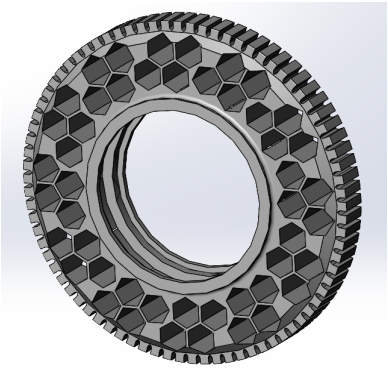


Figure 38: Compliant Wheel v1 CAD Rendering

4.1.5.1 Second Iteration

However, the team found that the increased diameter of the wheels posed several issues. First, the smallest effective diameter of the robot was greatly reduced. Second, the ratio of the width of the wheels to their diameter caused perpendicular forces to have greater effect, pulling the tires off the wheels. Third, the honeycomb pattern was ineffective in producing the amount of compliance necessary for the system.

Therefore, the team reduced the diameter of the wheels to that of the Pololu wheels, 32 mm and greatly reduced the size of the plastic insert going within the tire. The team examined various compliant wheel designs such as those from AndyMark (shown in Figure 39), which are used in high school robotics competitions.

Figure 39: AndyMark Compliant Wheels [23]

The team modified this design by adding an angle to the cutouts as shown below in Figure 40. Adding an angle increases radial compression rather than a pure linear compression, allowing for increased deformation of the wheel and stability around the central wheel axis. The central piece, pictured in blue, was printed from PLA to act as the wheel hub interface to a 3 mm axle. It was made much thicker, at 10 mm, to prevent torque from pulling the wheel from the hub. The surrounding tire was printed from NinjaFlex TPU. To ensure the wheel did not slip from the hub like the previous iteration, we decreased the size of the inner diameter to be smaller than that of the hub. We then wrapped the wheel around the hub, using the compliance of the TPU to create a tight fit.

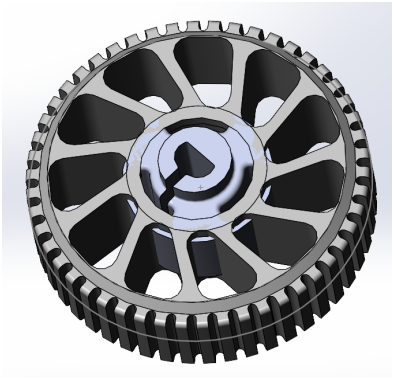


Figure 40: Compliant Wheel v2 CAD Rendering

This drastically improved the traction of the robot within PVC pipes but was still not enough to consistently climb up a pipe vertically. To further increase the traction, the team created 3D printed molds to cast DragonSkin 10 NV [24], a low viscosity silicone material, in a ring around the surface of the wheel, as shown below.

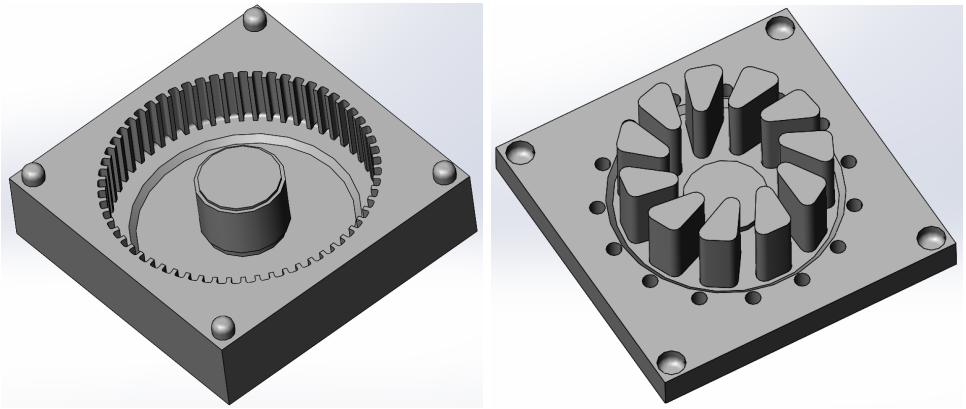


Figure 41: Compliant Wheel v2 Mold (left: bottom cavity, right: top mold)

To mold the silicone ring, the team mixed 10 g of DragonSkin 10 NV Part A and Part B in a disposable container, and stirred until the parts were well mixed. Next, we placed the wheel in a vacuum chamber for 5 minutes, or until all air bubbles were removed from the mixture. Then, we poured the mixture into the bottom cavity until it was filled entirely with DragonSkin. Then, we placed the TPU tire into the mold, ensuring that it was flush with the bottom surface of the PLA mold. Next, the top mold was aligned with the spokes of the TPU tire and then pressed down with the four alignment holes aligned. After 2 hours of curing, the wheel was removed from the mold, yielding the result shown below in Figure 42.

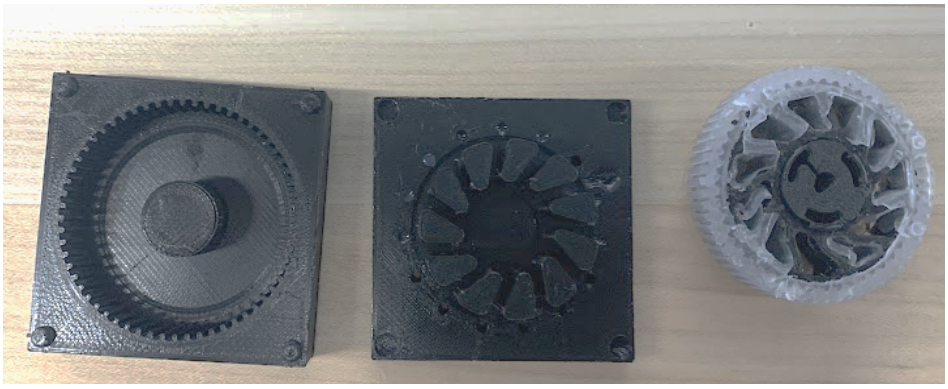


Figure 42: Compliant Wheel v2 Mold Result

4.1.6 Completed Robot Mechanical Design

Shown below are renderings and pictures of the completed robot, featuring simple representations of the Yoshimura continuum module.

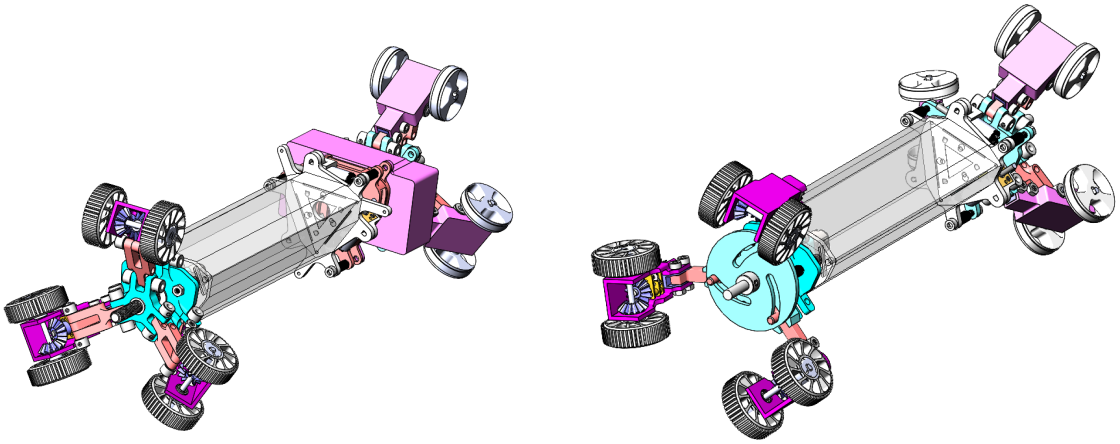


Figure 43: Completed Robot CAD Assembly Rendering (top: slider-crank mechanism, bottom: cam-slot mechanism)

4.2 Circuit Design

The circuit design of CLARA features two sets of PCBs, the smart motor drivers soldered to all seven N20 motors and a central mainboard mounted within the cross-section of the robot which controls communication and power supply for all seven smart motor drivers.

Prior to finalizing this circuit layout, the team discussed several alternatives to distribute power, processing, and control of the motors. Initially, the team thought of using larger chips such as the ATMega32U4 and fitting seven motor drivers, current sensors, and counters onto a single mainboard and reading encoder values from the chips that come pre-soldered to N20 motors, similar to the previous iteration of the Salamanderbot control board. However, we found that PCB layout became an issue, as the size of all components surpassed our PCB size

constraint of the cross-section of the robot. Rather than increasing the size of the robot, the team decided to rethink how they would distribute the components required in the system.

To resolve these issues, the team determined the best course of action would be to continue to have a central mainboard responsible for wireless communication, power regulation and distribution, and control of several satellite smart motor driver boards. These smart motor drivers have a microcontroller on board, as well as encoders, current sensors, and other components. This organization structure allows for boards with smaller profiles capable of fitting within the envelope of the robot body, without sacrificing desired components or ease of communication.

4.2.1 Smart Motor Drivers

4.2.1.1 SAMI Board by 2BRobots

When initially starting development of the Smart Motor Drivers, the team found a project called SAMI by 2BRobots, which is a custom PCB to control an N20 gearmotor using PID control and relaying feedback over I2C. The SAMIs provided a favorable design for the application they were intended to be used for, as they offered every component that was desired, minus current sensors [19]. However, the drivers were out of production for several years, so the team needed to reverse engineer them or produce a similar product.

Using Altium as the program of choice, the team imported the schematics of the SAMI boards from their open source repository by importing the Gerber files yielding the results shown in Figures 44 and 45.

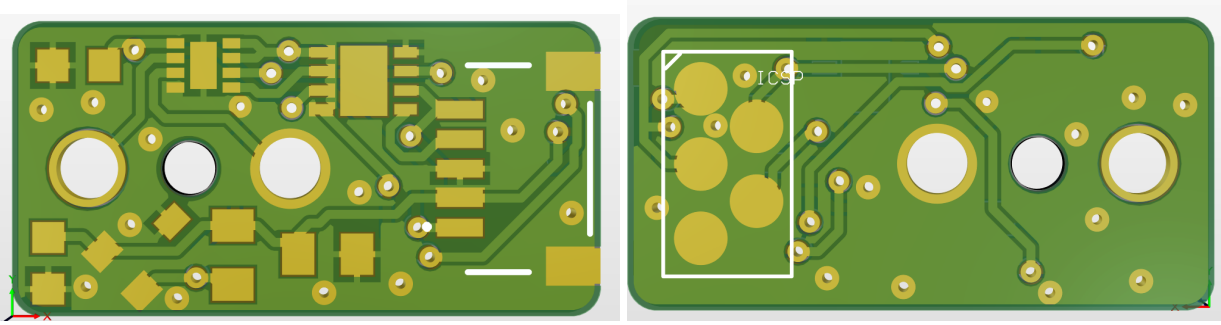


Figure 44: SAMI Board PCB Layout (left: Front of PCB, right: Back of PCB) [19]

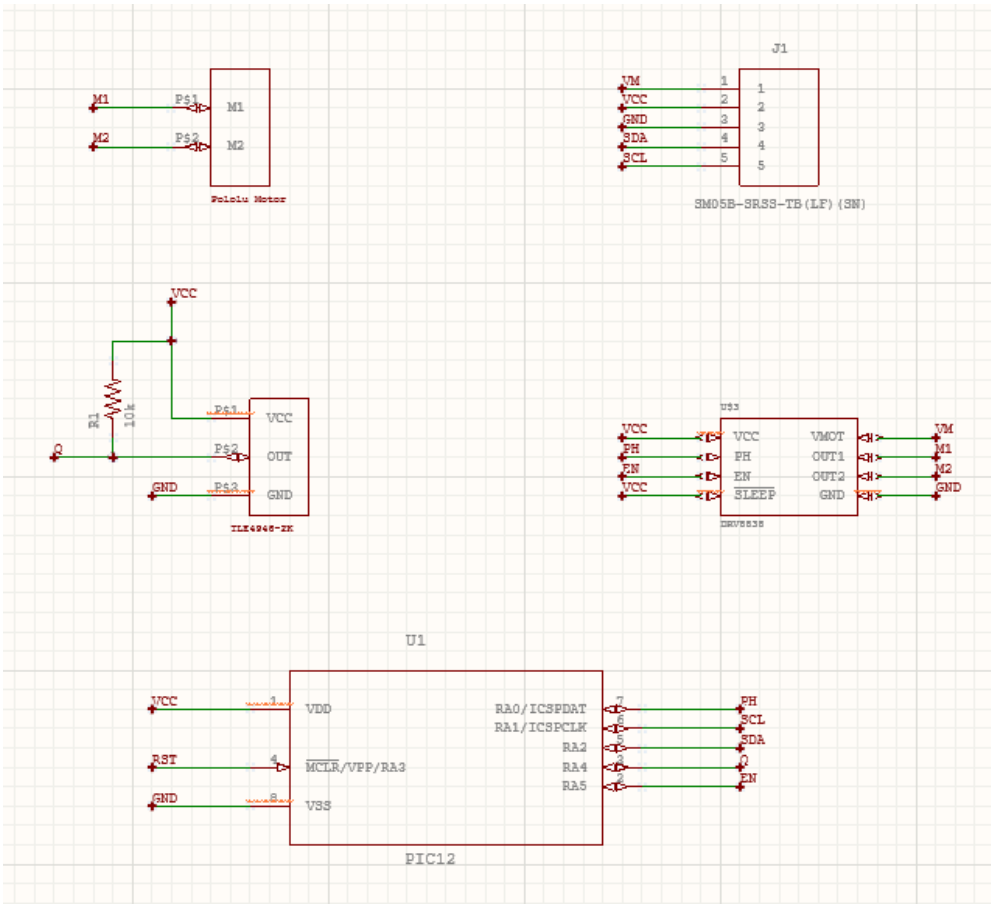


Figure 45: SAMI Board Schematic Snippet [19]

The team thought that this board would be a great start towards development of an all-in-one motor driver. We wanted to add new components such as current sensors, quadrature encoders, Serial capability, and more JST connectors for daisy-chaining capability to further the capabilities of the SAMI board. However, we found that most components from the SAMI design were out of production or unavailable due to ongoing chip shortages. This meant that the team needed to refactor the entire PCB and perform component selection using more recent integrated circuits and passive components, so the design of the board was essentially overhauled, with replacement of every component, addition of components, and decreasing the overall size.

4.2.1.2 Preliminary Design

During the preliminary design phase of the Smart Motor Drivers, the team took inspiration from the schematics of the Salamanderbot and selected similar nMOS motor drivers, as the drivers used previously were unavailable, (DRV8838DSGT) and single output current sensors (LT1999IMS8-20#PBF) while changing the microcontroller to an Atmel instead of a PIC, as Atmel controllers are more easily programmable. We selected a readily available microcontroller in the ATTiny2313V-10MU, which is an 8-bit microcontroller with 18 I/O pins

and I2C capability. The addition of a second hall-effect latch sensor (TLE4946-2K) would drastically increase our encoder resolution. The current sensor would help us maintain stability of the system, preventing large currents should the N20 motor stall, and even provide capabilities such as current-homing of the robot for the cable actuators. Current sensors read a voltage drop across a small resistor, so we implemented a 100 mOhm resistor across the output line of the motor driver. The team also added bypass capacitors as recommended by the datasheet. The team added a 5 pin JST connector (SM05B-GHS-TB(LF)(SN)) to allow for power transmission from an external source and I2C communication. This resulted in a single-sided circuit board as shown in Figure 46.

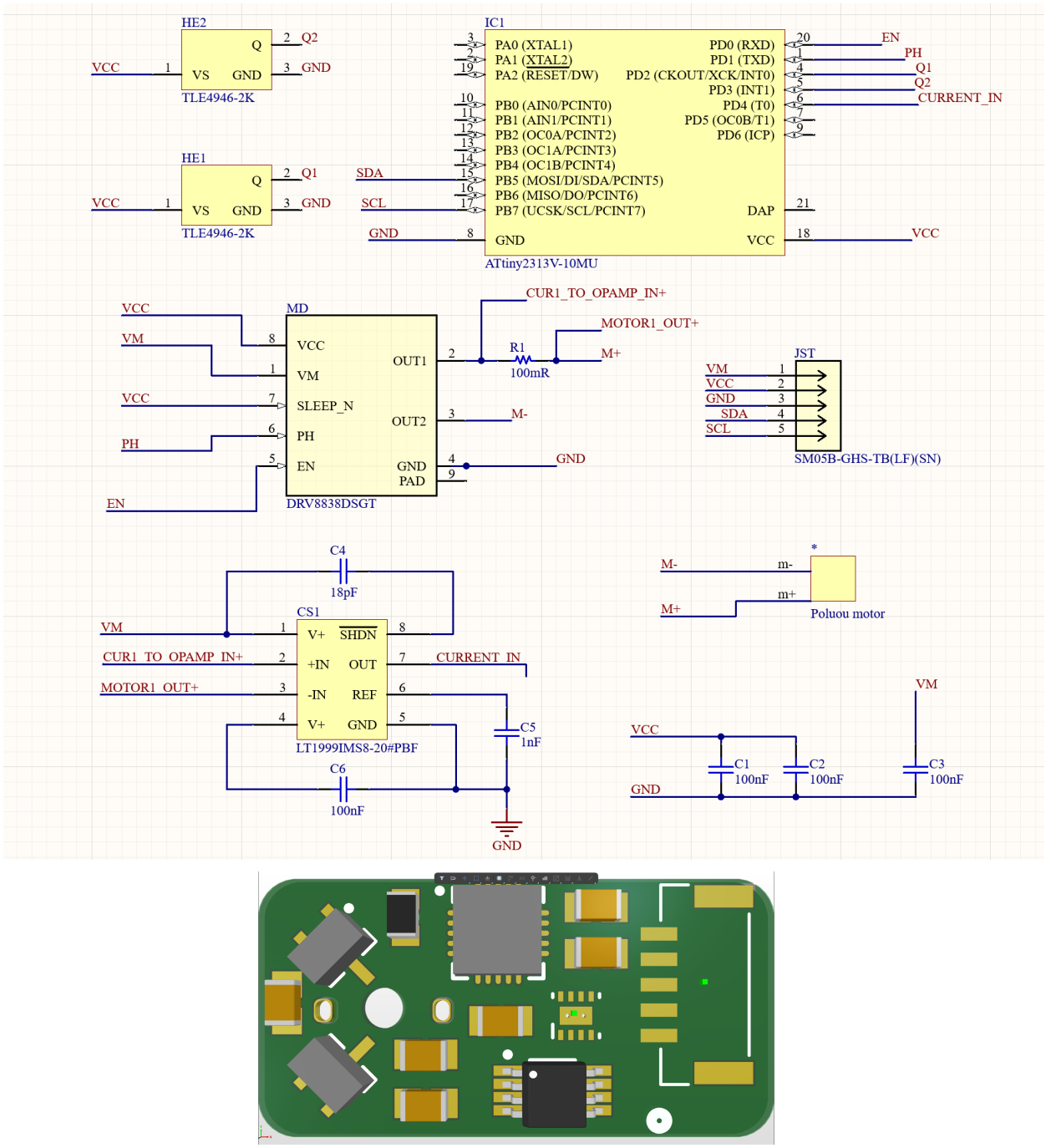


Figure 46: First Iteration of Smart Motor Driver Schematic and Theoretical Layout

4.2.1.3 First Iteration

During the first production iteration of the Smart Motor Drivers, the team finalized component selection and PCB layout. Between the time of the preliminary design process and

the first iteration, the motor drivers, current sensors, hall-effect latches, and microcontrollers went out of stock due to a current chip shortage.

The first major alteration was the motor driver. Rather than using an nMOS chip, the team selected an A3910EEETR-T from Allegro MicroSystems, a dual half-bridge which uses a Power MOSFET technology to drive a motor in two directions. This motor driver was selected due to its ability to output upwards of 500 mA in current and compatibility for voltages between 2.5V and 5.5V, suitable for driving the N20 motor [25]. However, this motor driver required four dedicated inputs excluding motor and logic voltages rather than two from the previous selected component. As recommended by the driver's datasheet, the team added a recommended bypass capacitor to the motor voltage to filter out noise.

Next, the current sensor was replaced with an INA2180A1IDGKR from Texas Instruments as it featured the same voltage compatibility and smaller package size. While our system only required one current reading along the motor terminal, this current sensor features two sets of inputs and outputs for two readings [26]. As recommended by the sensor's datasheet, a bypass capacitor was added to the logic level input. Next, the team performed calculations to determine a suitable resistor for current readings using the equation given in the driver's datasheet (Equation 1). In this equation, our maximum current (I_{MAX}) would be that of a singular 0.6 W N20 motor of 0.55 A [27]. The gain was determined by the component we selected, In this case, our gain was 20 V/V as specified by the datasheet. The team selected the output voltage of the sensor, V_{SP}, to be just below 3 V, as the output featured a small voltage bias of -0.03V. Using the equation and known parameters, the team determined that a sense resistor of 0.27 Ohms was appropriate for this circuit.

$$I_{MAX} * R_{SENSE} * GAIN < V_{SP}$$

Equation 1: Sense Resistor Selection Equation [26]

To determine the power rating of the sense resistor, the team used another equation within the datasheet for the current sensor as shown in Equation 2. Using the acquired sense resistance and known maximum current, the team found that the power rating of the resistor needed to be larger than 0.08 W. Knowing these two parameters, the team selected the RL1220S-R27-F, a 0.27 Ohm resistor with ⅓ W power rating for our current sensor [28].

$$R_{SENSE} < \frac{PD_{MAX}}{I_{MAX}^{2}}$$

Equation 2: Sense Resistor Power Rating Equation [26]

The next component that the team altered was the hall-effect latch sensors. The team found a near-identical sensor, the SS360PT by Honeywell, which featured the same capabilities and an identical PCB footprint [26].

We also added an additional JST connector, to allow for multiple Smart Motor Drivers to be connected to one another along the same I2C bus. The JST connectors we originally selected

went out of stock, so we replaced them with S5B-ZR-SM4A-TF (LF) (SN) connectors which feature five connections and are surface-mount [30].

To flash the Smart Motor Drivers, the team added a six pin contact pad directly into the board, which featured connections for an ISP (In System Programmer) for programming. The design of this component was created using a recommended pad layout from a TagConnect (TC2030-NL) ISP connector, which features six spring-loaded connectors and three feet for stability [31].

Lastly, the team replaced the microcontroller. The microcontroller we selected previously went out of stock, and did not feature an ADC for current sensor reading. Additionally, we wanted to select a microcontroller that would be readily programmable with the Arduino IDE, which features libraries for programming Arduino boards and Atmel microcontrollers, to save time flashing the boards in the future. Therefore, we selected the ATmega328-AU, which features a 10-bit ADC, larger program memory, faster speeds, and more GPIO (General Purpose Input/Output) pins [32]. We connected the microcontroller to a 16 MHz crystal to allow for on-board program flashing, a 10 kOhm pullup resistor to the RESET line as recommended in SPI flashing, and several capacitors to set the reference voltage for the ADC.

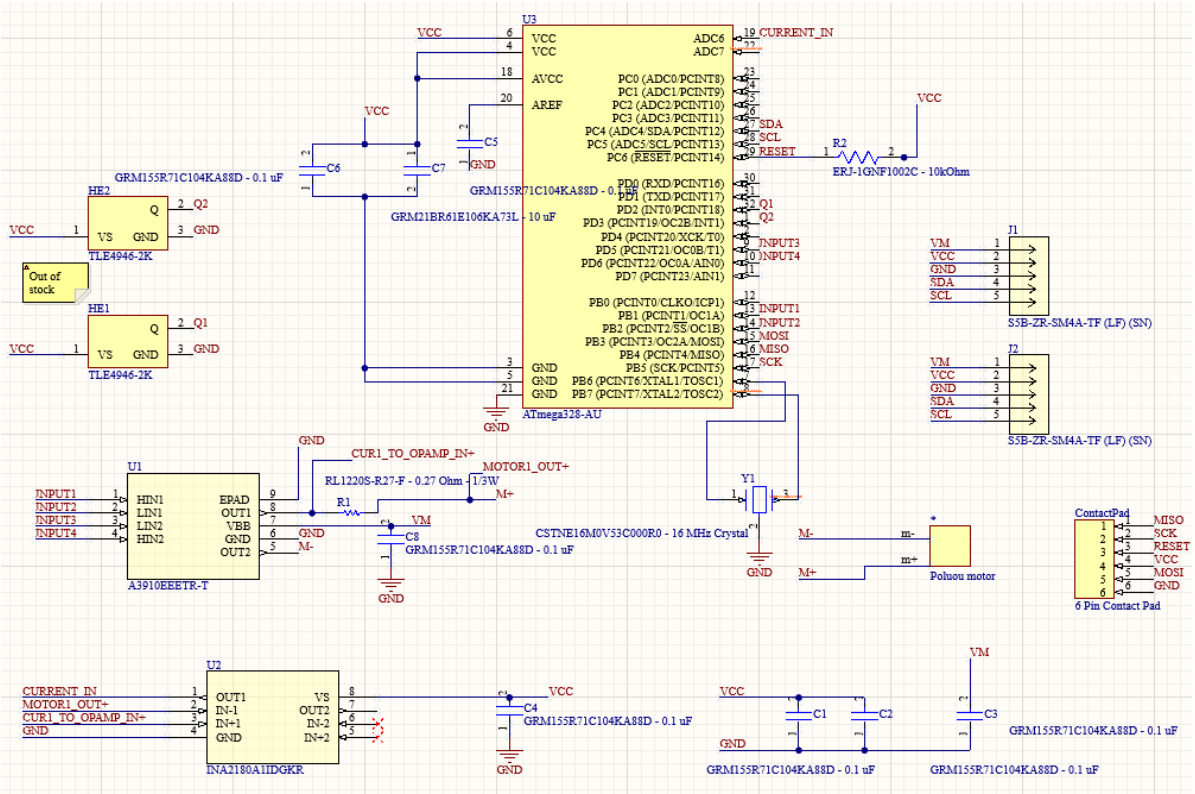


Figure 47: First Iteration Smart Motor Driver Schematic

As shown below in Figure 48, the first iteration PCB was much more compact than the preliminary design due to the increased number of components and large component sizes. The layout features a cutout for the motor leads towards the left side of the board, with hall-effect sensors on both sides and facing 90 degrees apart. The microcontroller lies in the center of the board, with the motor driver being as close to the motor leads as possible. The team used a trace width of 6 mil for the entire board, with a spacing of 6/6 mil, specified in the design rules. The JST connectors take the most physical space, and are isolated to the right side of the board. To save time on routing, the team used Altium's auto-routing feature, ensuring that it completed all connections. This resulted in a two-sided PCB (Figures 49-51), with components such as the current sensor, crystal, and ISP connections moved to the back to save space. The completed board measured 29.46 mm in length and 12.83 mm in width.

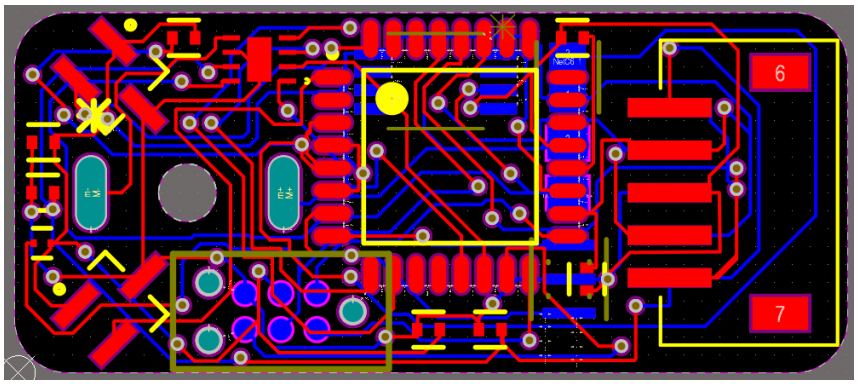


Figure 48: First Iteration Smart Motor Driver PCB Trace Layout

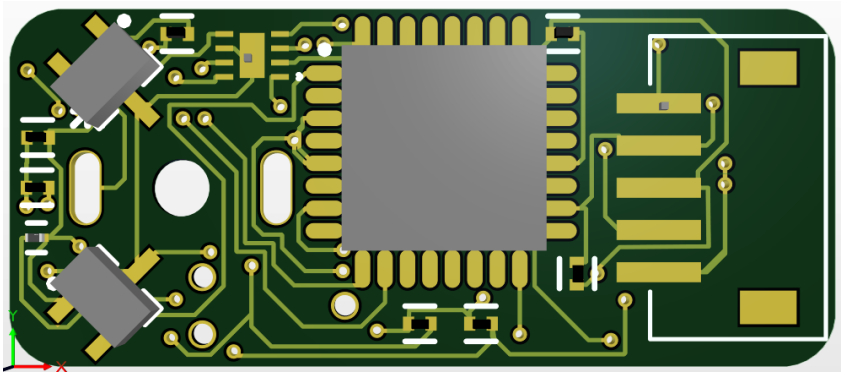


Figure 49: First Iteration Smart Motor Driver PCB Front View

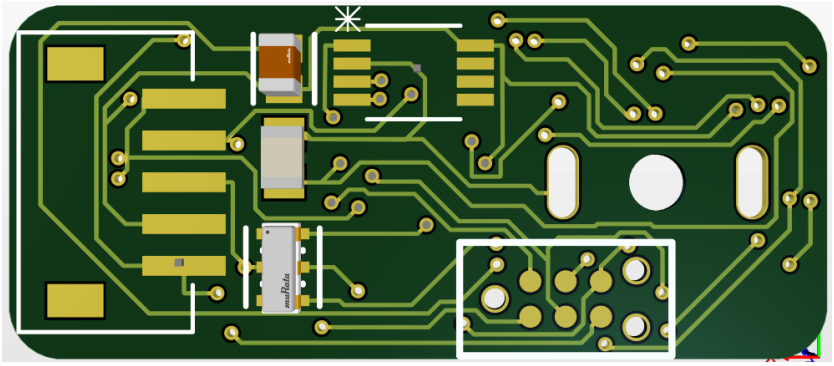


Figure 50: First Iteration Smart Motor Driver PCB Back View

This board was eventually manufactured using a third party PCB manufacturing service, PCBWay. To generate the manufacturing files, we used Altium's fabrication outputs menu and generated Gerber and NC Drill files, ensuring that both files were scaled in mm. In our Gerber files, we selected the most important layers for the manufacturer: the top and bottom layers (traces and pads on the top and bottom of the PCB), the top and bottom overlay layers (screen-printed graphics), the top and bottom solder layers (solder masks), and the keep-out layer (board shape).

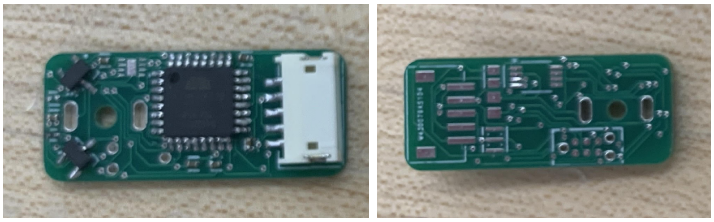


Figure 51: First Iteration Smart Motor Driver PCB

4.2.1.4 Second Iteration

The second iteration of the Smart Motor Drivers consisted of no component and schematic changes but rather changes to improve the performance of the PCB.

The team found that with the first iteration, there were troubles flashing the microcontroller, due to the placement of components and length of traces. Therefore, all components were rearranged to be closer to one another, and on the same side of the board. As a result, the board width increased by 2.7 mm to fit all components and leave enough room for traces. The six pin contact pad was placed as close as possible to the microcontroller pins and the 16 MHz crystal to avoid any noise affecting flashing the system. Additionally, we placed bypass capacitors near their designated pins to reduce the impact of any noise. Lastly, the current sense resistor was placed as close to the motor driver and current sensor as possible, to reduce trace lengths and receive the most accurate current reading.

To improve performance and reduce the possibility of current issues, we altered the widths of traces to high current consumption components, such as the N20 motor. Traces that were connected to the motor leads and operating at the motor power level (3.7 V) were widened to 14 mil, while normal traces remained 6 mil. Additionally, a ground plane was added on both sides of the PCB using Altium's repour feature to reduce noise for high frequency signals when flashing or sending I2C commands.

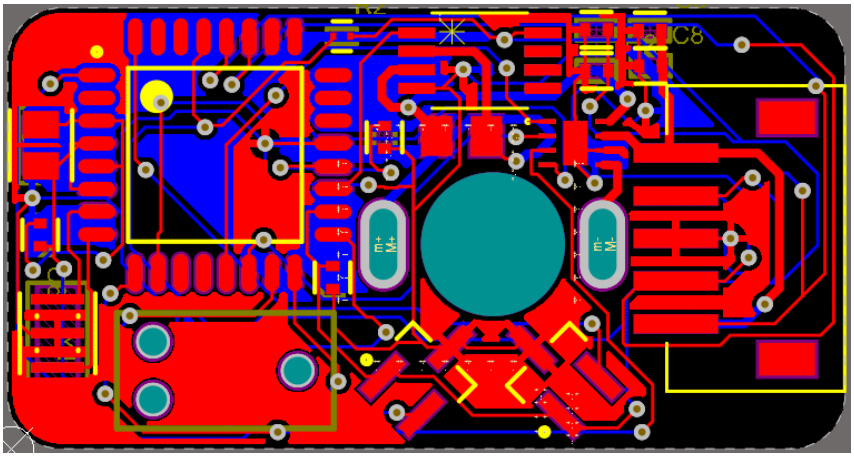


Figure 52: Second Iteration Smart Motor Driver PCB Trace Layout

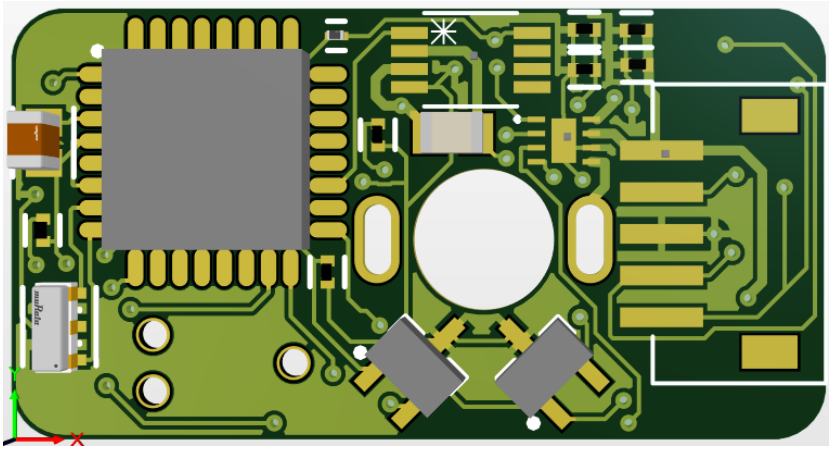


Figure 53: Second Iteration Smart Motor Driver PCB Front View

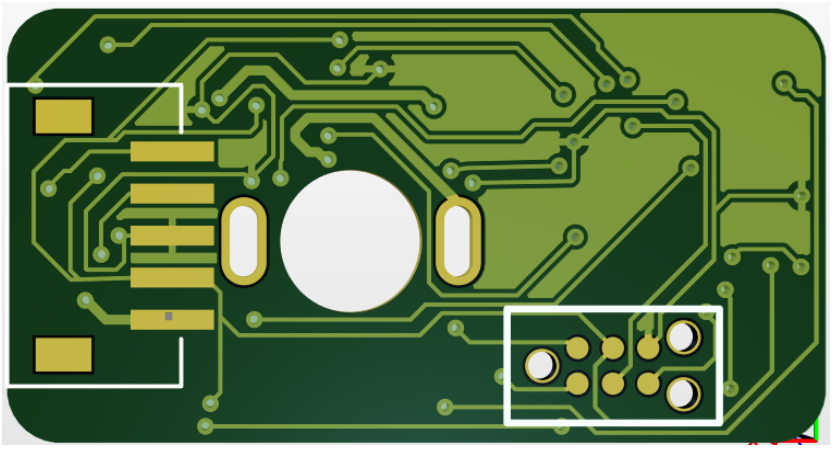


Figure 54: Second Iteration Smart Motor Driver PCB Back View

4.2.1.5 Third Iteration

During the third iteration, the team encountered a large issue. We found that operating the motors at 3.3 V was possible but only for short periods of time. Therefore, we decided to upgrade our circuit to handle a 2S LiPo battery input at 7.4 V. This required new components to handle the increased voltage. We replaced the motor driver with an MD9927 from Shanghai Mingda Microelectronics, which can handle upwards of 12 V on the motor voltage and 7 V for the logic voltage. This motor driver is a h-bridge motor driver, and features motor direction reversing with two inputs [33]. We also replaced the current sensor with a ZXCT1010E5TA from Diodes Incorporated, which can only read one current rather than the previous sensor which read two. Additionally, the sensor can handle inputs upwards of 20 V [34].

For this current sensor, we selected a new sense resistor using suggested equations from the ZXCT1010E5TA datasheet as shown below in Equation 3 and using the typical application circuit to select an output resistor. To use the equations, we first defined the voltage drop across the sense resistor, R_{SENSE}, to be between 50 and 500 mV at full load as recommended by the datasheet. In our case, we chose 250 mV at our maximum current load of 0.55 A for a singular

N20 motor. Using Ohm's law as shown in Equation 4, we found the size of the sense resistor to be approximately 0.5 Ohms. To find the appropriate power rating for the sense resistor, we used the same power relationship in Equation 2, using our calculated resistance and maximum N20 stall current resulting in a minimum power rating of 0.151 W.

$$V_{SENSE} = V_{IN} - V_{LOAD}$$
$$V_{OUT} = 0.01 * V_{SENSE} * R_{OUT}$$

Equation 3: ZXCT1010E5TA Current Sense Resistor Equation [34]

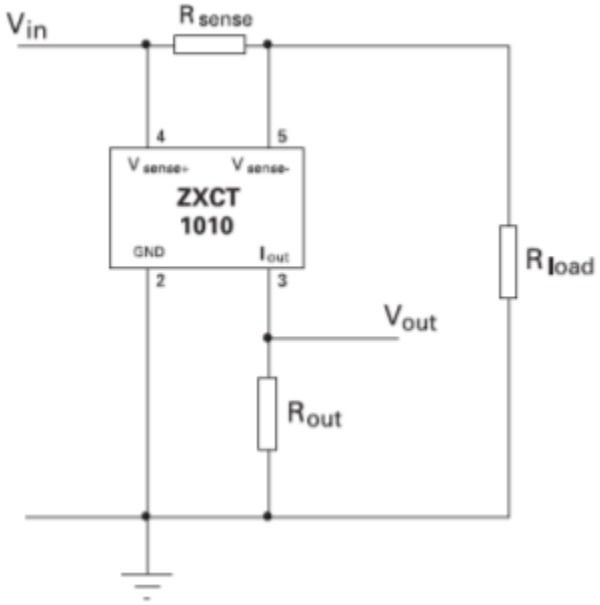


Figure 55: ZXCT1010E5TA Typical Application Circuit [34]

$$R_{SENSE} = \frac{V_{SENSE}}{I_{SENSE}} = \frac{0.25\,V}{0.55\,A} \approx 0.5\,\Omega$$

Equation 4: R_{SENSE} Ohm's Law Calculation

This resulted in the selection of a 0.5 Ohm, ⅛ W resistor [35]. Next, the team determined the value of the output resistor. Using Equation 1 once more, we chose our output voltage to be 3 V as specified in the calculations of our previous current sense resistor and found an output resistance of 1200 Ohms. Using Ohm's law once more, we determined the power rating of the system given our known voltage drop and output resistor as shown in Equation 5. This resulted in the selection of the ERA-3AEB122V from Panasonic, which is a 1.2 kOhm resistor with a 1/10 W power rating [33].

$$P = \frac{V^2}{R} = \frac{(3\,V)^2}{1200\,\Omega} = 0.0075\,W$$

Equation 5: R_{OUT} Power Rating Calculation

We also realized that in our previous schematic, current sensor readings would only occur when voltage went through the positive lead of the motor. To allow for current readings when the motor runs in both directions, we moved the sensor to the motor voltage input line on the motor driver, where the current can still be read.

The team also wanted to include Serial capability for debugging purposes. Since the ATMega328-AU had direct pins dedicated for Serial communication (RX/TX), we created two through hole pads for 20 gauge jumper wire.

Lastly, the team discovered that some pins, such as the ADC pins, were not accessible using Arduino software due to internal Arduino Nano pin mappings. Therefore, the team relocated the CURRENT_IN and all three INPUT pins as shown below in Figure 56.

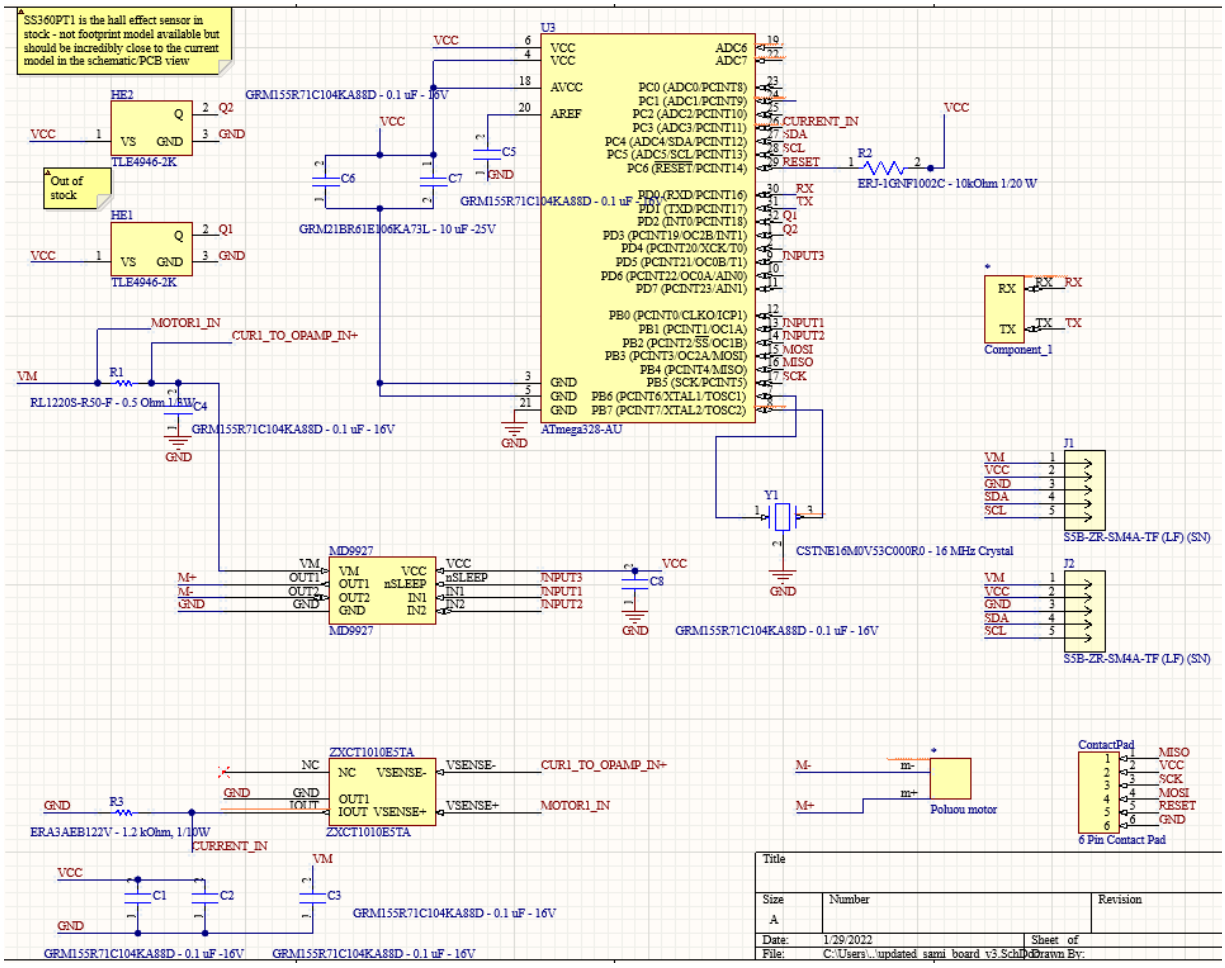


Figure 56: Third Iteration Smart Motor Driver Schematic

The PCB layout was also adjusted so that the motor leads were placed closer to the board's edge. This would allow for the N20 motor to be mounted flush to a flat surface. Lastly, we found issues in soldering the PCBs from iteration one due to vias bridging to nearby solder pads. Therefore, we tented the vias to prevent unintentional short circuits that could hinder the performance of the Smart Motor Drivers.

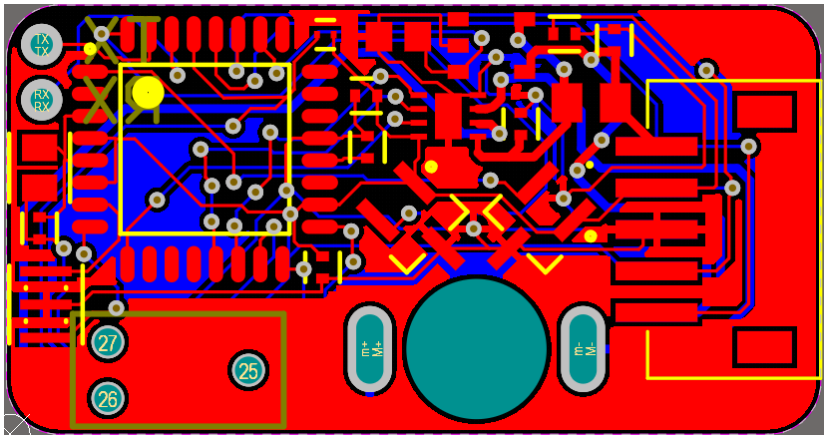


Figure 57: Third Iteration Smart Motor Driver PCB Trace Layout

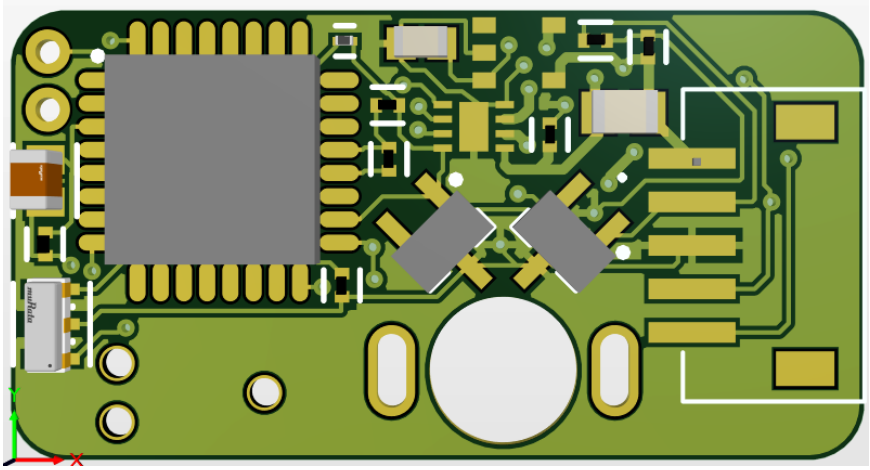


Figure 58: Third Iteration Smart Motor Driver PCB Front View

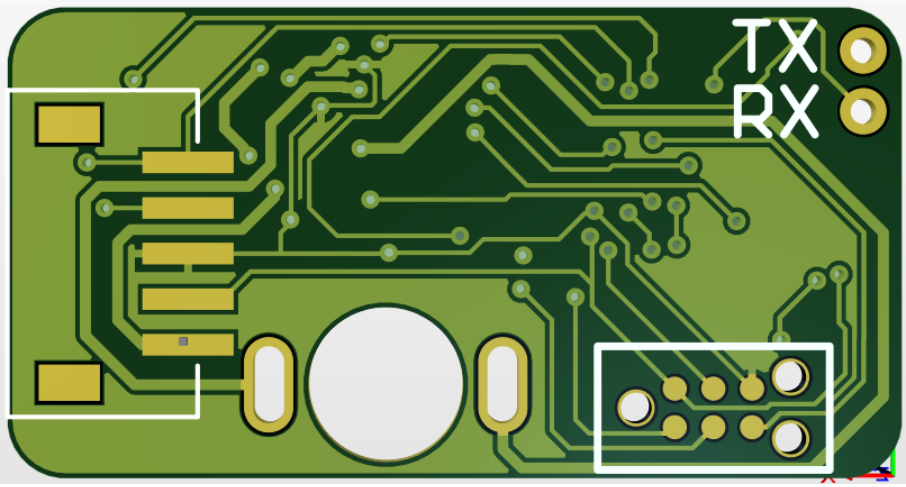


Figure 59: Third Iteration Smart Motor Driver PCB Back View

4.2.2 Mainboard

4.2.2.1 First Iteration

In the first iteration of the mainboard, the team decided upon a microcontroller. Initially, we looked at using the ESP32-WROOM32 microcontroller due to its small size, fast processing speed, and communication protocols such as I2C and WiFi. However, we found that the ESP32-WROOM32 required several other components to resemble an ESP32 Devkit Module, used throughout the undergraduate RBE course sequence.

To avoid reinventing the wheel, the team found an appropriate ESP32 module that fit within the size constraints of the robot. We selected the TinyPICO v2 board, created by Seon Rozenblum, as it uses a smaller ESP32 package, while maintaining the desired communication protocols [37].

To connect the mainboard to all of the Smart Motor Drivers, the team selected a five pin JST connector which would carry the battery voltage, logic level voltage, ground, and two I2C connections (SDA and SCL). While the robot design only requires seven Smart Motor Drivers to operate, the team added an additional two connectors for future modification. The team also added a two pin JST connector to connect a battery directly to the PCB.

The team needed to regulate input power to avoid overvolting components on the smart motor drivers. For this iteration, the team decided that we could use a 1S LiPO battery at 3.7 V for our motor voltages and use a 3.3 V linear regulator from onsemi (NCP1117DT33T5G) [38]. Due to the low voltage regulation, the team decided to use a linear regulator rather than a buck regulator as losses would be small for a 0.4 V voltage drop. The team added 10 uF bypass capacitors on the input and output of the regulator as recommended by the datasheet. Due to the low power consumption of the TinyPICO and microcontroller units on the Smart Motor Drivers, the current output of the regulator was selected at 1 A.

Lastly, the team added two 4.7 kOhm pullup resistors to the I2C lines, since I2C will pull lines low during communication.

The board takes the shape of the Yoshimura module cross section. The TinyPICO lies at the center of the board on female header pins, while the JST connectors are dispersed on both sides of the board. For this board, the team used 10 mil wide traces.

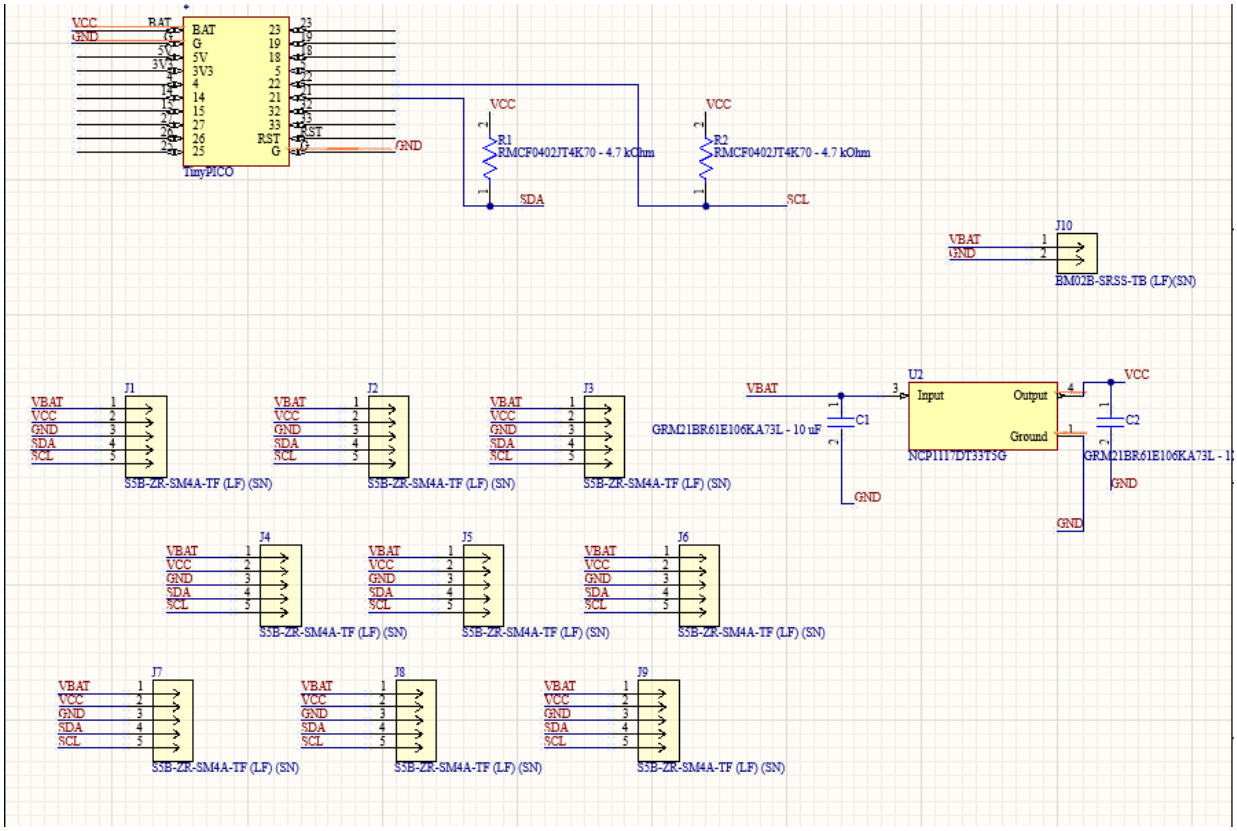


Figure 60: First Iteration Mainboard Schematic

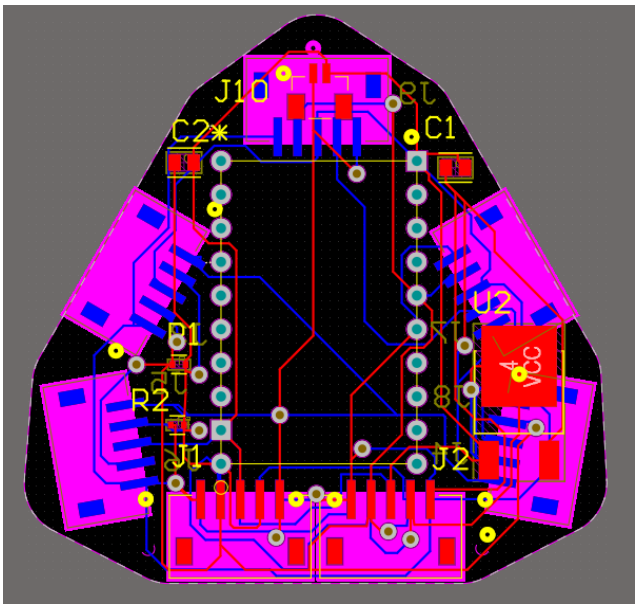


Figure 61: First Iteration Mainboard PCB Trace Layout

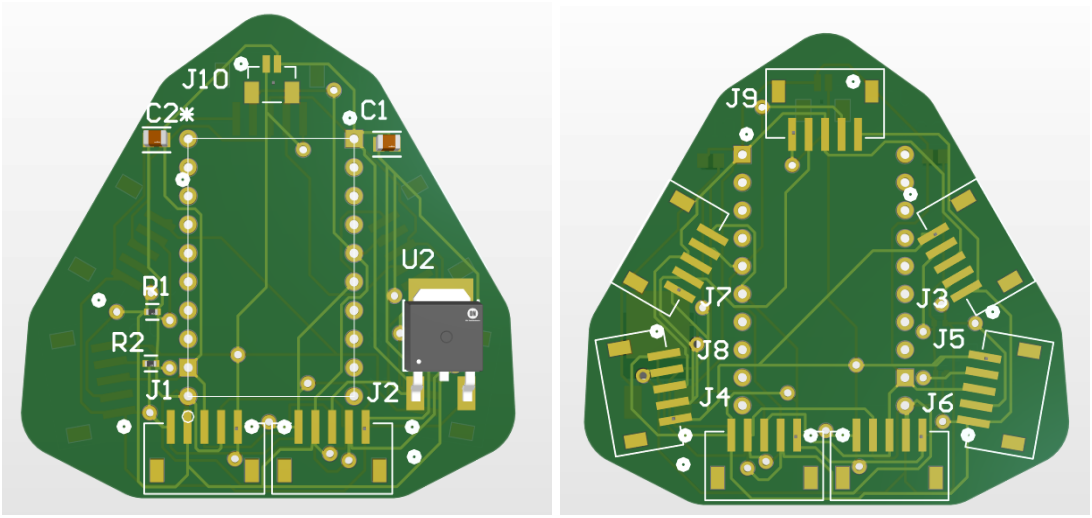


Figure 62: First Iteration Mainboard PCB 3D View (left: front, right: back)

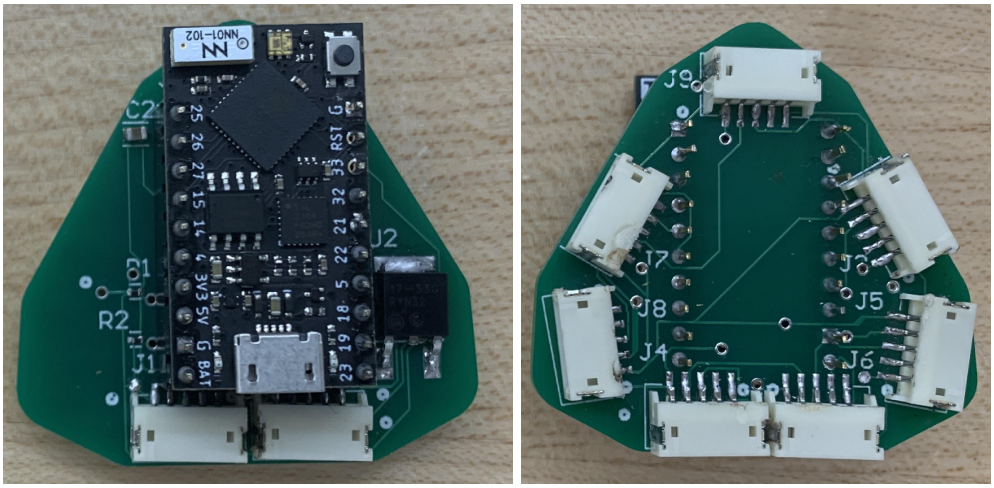


Figure 63: First Iteration Mainboard PCB

4.2.2.2 Second Iteration

In the second iteration of the mainboard, the team decided to upgrade the motor voltage of the system. After testing an N20 motor at 3.7 V using an external DC power supply, the team found that the motors could not withstand small amounts of torque making them unusable for our application as they are typically rated for 6 V. Therefore, we decided to use a 2S LiPO battery rated at 7.4 V as our power voltage.

Unlike the previous iteration, we selected a buck regulator to step our power voltage down from 7.4 V to our logic level voltage to 3.3 V. For larger voltage drops, linear regulators have lower efficiency, and will produce a lot of heat. However, buck regulators consist of several passive components such as inductors, capacitors, and resistors. Rather than going through the component selection process for a buck regulator, the team found a pre-built solution offered by Pololu called the D24V10F3. This buck regulator features all components on a small breakout board that can be plugged into the mainboard using header pins without the trouble of component selection. Since it is a buck regulator, it features very high efficiencies (upwards of 80 and 90 percent) and regulates down to 3.3 V with voltage inputs as high as 36 V. The regulator is rated for 1 A, like the linear regulator we used in the previous iteration [39].

The regulator features five pins, PG (“power good”), SHDN (shutdown), VIN (input voltage), VOUT (output voltage) and GND. For our purposes, we only require the use of the VIN, VOUT, and GND pins, so the other two pins were left unconnected.

Like our process for the Smart Motor Drivers, we added ground planes to this iteration of the mainboard PCB and tented the vias to avoid problems when soldering. We also added three mounting holes for fastening to the robot body.

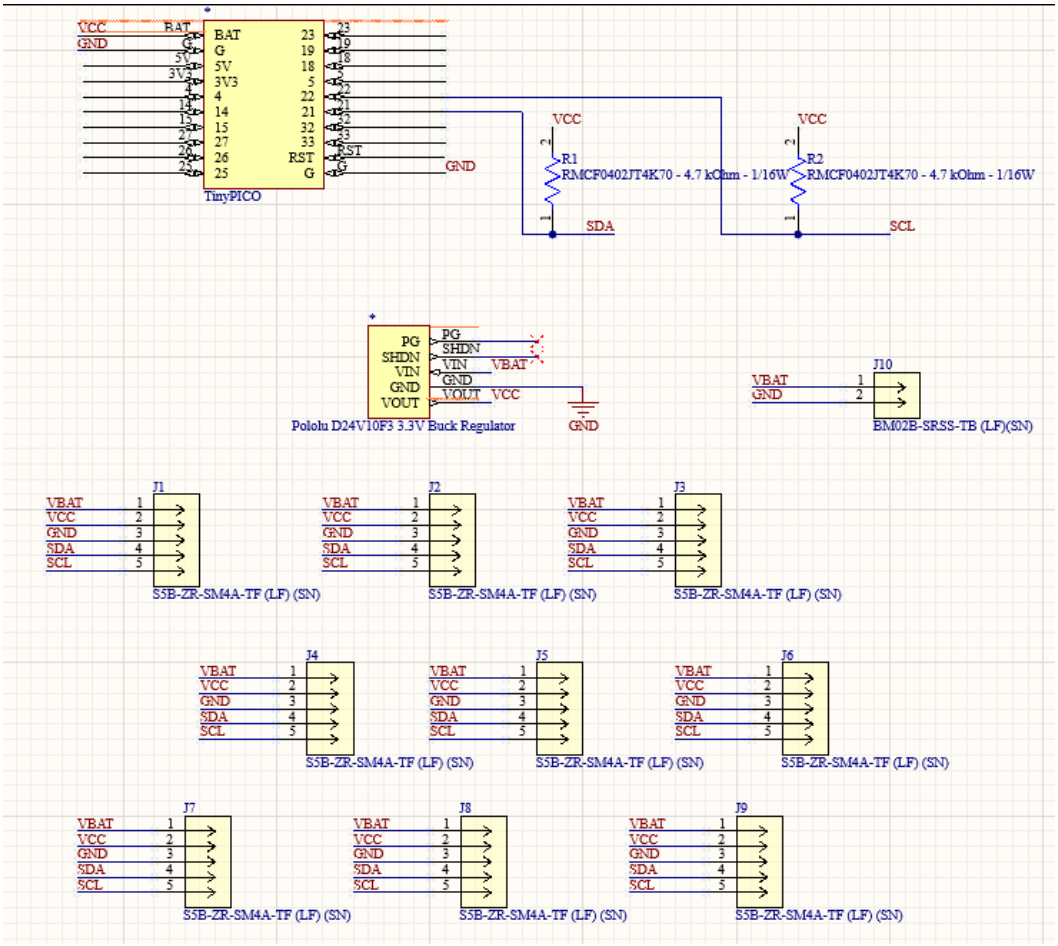


Figure 64: Second Iteration Mainboard Schematic

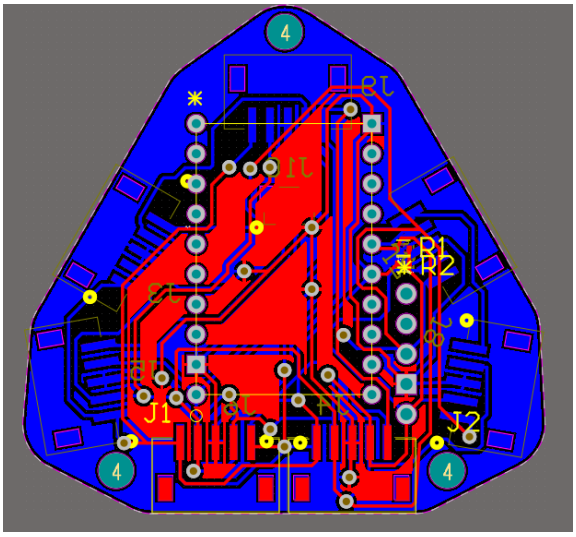


Figure 65: Second Iteration Mainboard PCB Trace Layout

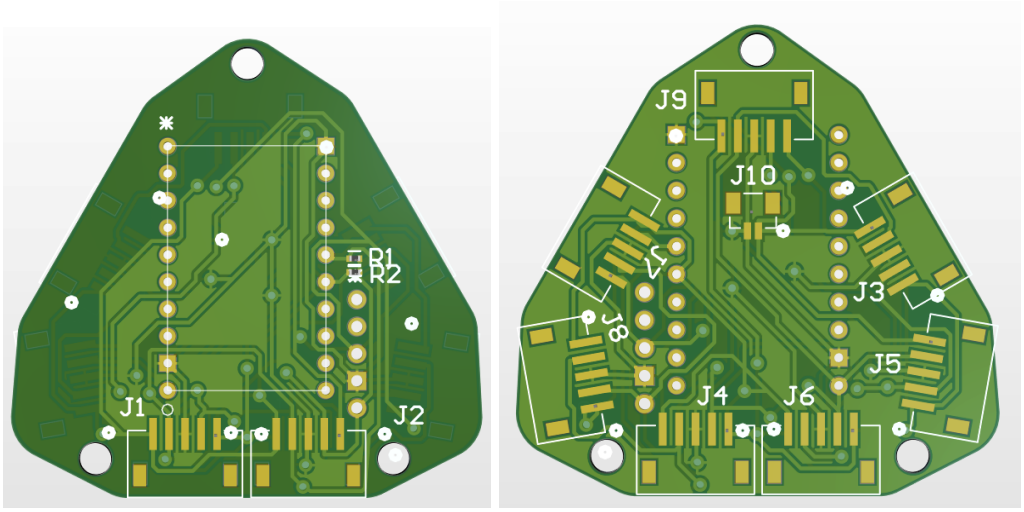


Figure 66: Second Iteration Mainboard PCB 3D View (left: front, right: back)

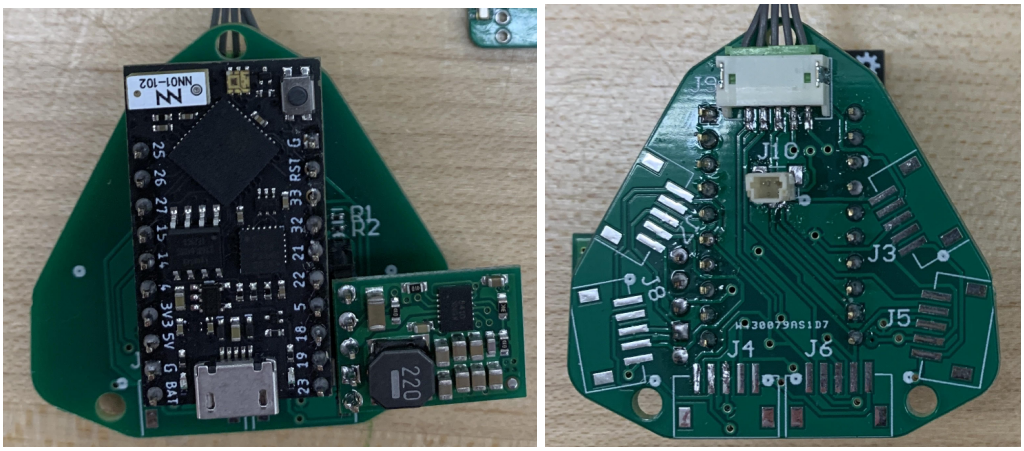


Figure 67: Second Iteration Mainboard PCB (left: front, right: back)

4.2.2.3 Third Iteration

In the next iteration of the mainboard, the team made small refinements concerning the power distribution. Rather than a 2-pin JST connector with 30 gauge wire, the team created two through hole pads for 14 gauge wire to prevent overheating of wire under large loads. Additionally, all traces routing back to the battery voltage were made wider to 30 mm to support larger current and decrease the temperature change of traces.

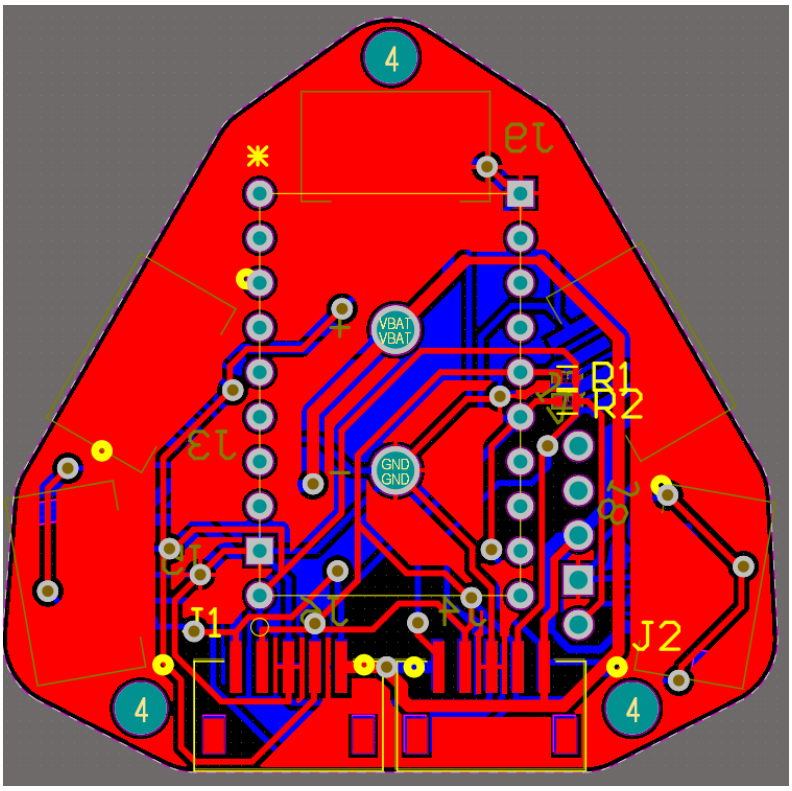


Figure 68: Second Iteration Mainboard PCB 3D View (left: front, right: back)

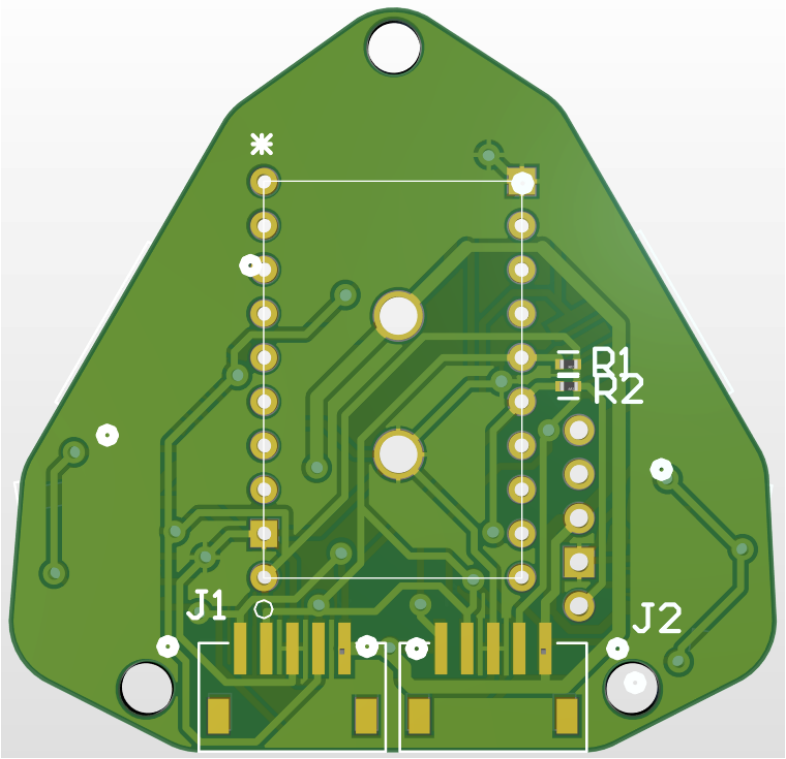


Figure 69: Third Iteration Mainboard PCB 3D View (left: front, right: back)

4.3 Control Architecture

The control structure for operation of CLARA can be broken into two main components, with the first being WiFi communication for control of the robot and the second being I2C communication between the main onboard ESP and the smart motor drivers. An overview of the communication structure can be seen in Figure 70.

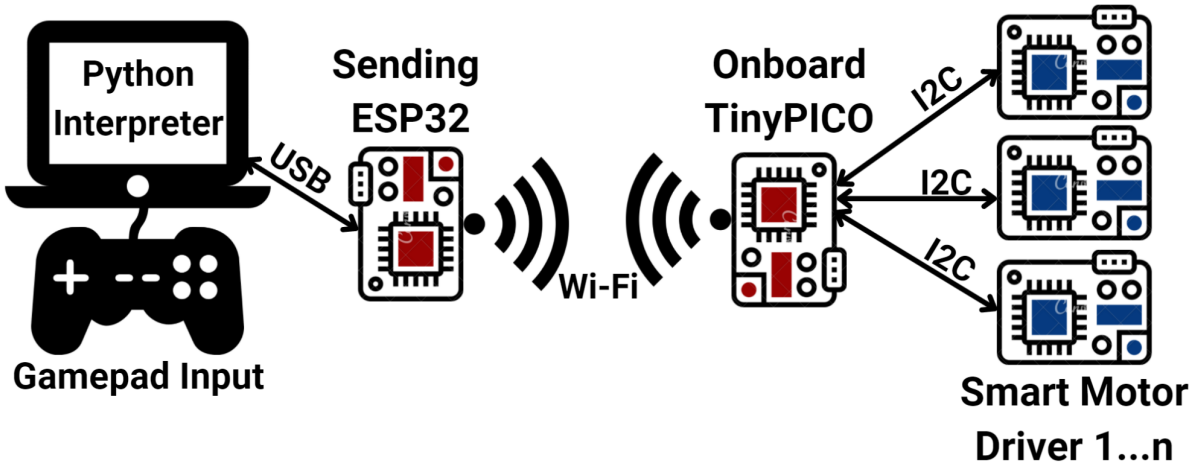


Figure 70: CLARA Communication Structure

4.3.1 Communication Architecture

4.3.1.1 Wi-Fi Communication

Operating the robot required a number of communication protocols to meet the design constraints of the project, such as having untethered control of the robot and having a central communication board with satellite boards for control of each individual N20 motor. to send remote signals to the robot, two TinyPico development boards were used- one onboard CLARA, and one connected to the controlling computer [37]. The TinyPico boards have an ESP32 microcontroller, which enables the use of ESP-NOW, a type of connectionless Wi-Fi communication protocol developed by Espressif for use with ESP-32 microcontrollers [40]. ESP-NOW provided a lightweight framework for the desired tetherless communication which was favorable in this application. The TinyPico boards were chosen over more typical ESP32 development boards simply for the reason that they provide nearly all of the same features of the typical boards but in a much smaller package, with a size comparison shown in Figure 71.

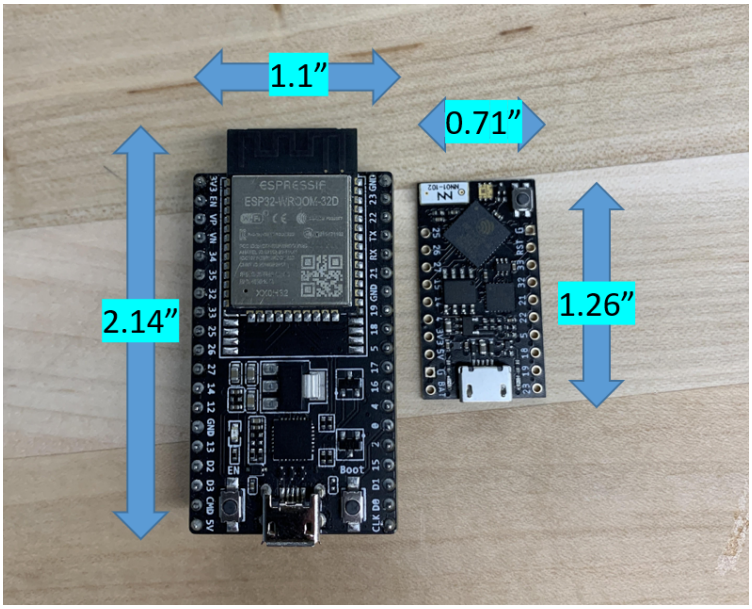


Figure 71: ESP32 Development Board and TinyPico Sizes

4.3.1.2 I2C Communication

As the electrical design of this robot revolved around having one central board for communication and multiple satellite boards on each motor, the I2C communication protocol was favored for this application as it would minimize the number of wired connections onboard the robot. In this configuration, only two wires were required to go to each of the smart drivers from the main board, with the possibility for either a direct connection from the board to each satellite board, or a single connection from the mainboard to a satellite board with the rest of the satellite boards daisy-chained together, or even a combination of both if that was favored for wire management purposes.

For developing code with the I2C protocol, the built-in Arduino Wire libraries were used [41]. Then, each smart motor driver could be flashed with its own unique device address and two way communication could be achieved with the mainboard acting as the leader device and the motor drivers being the follower devices.

4.3.1.2 Serial Communication with Gamepad

To make controlling the robot more intuitive for the team and for others, we incorporated a Logitech F310 Gamepad to take inputs and translate them into commands for the robot. To read and interpret the inputs on the controller, Yoni Weiner helped us by developing a Python script that registers the device, opens a serial port, and translates the controller input data into a set of 42 bits that can be interpreted from the external ESP as shown below [42]. The script allocates certain bits for various inputs to tell if a button is being pressed or if a stick is being moved, meaning that both analog and digital inputs are read.

Bits	0-3	4	5	6	7	8	9	10-17	18-25	26-33	34-42
Ref	D-PAD Dir*	Y	X	B	A	RB	LB	LJoy 1	LJoy2	RJoy 1	RJoy2

Figure 72: CLARA-Gamepad Communication Protocol [39]

From this, we checked the bits individually on the external ESP, and passed along Strings to the mainboard corresponding to each controller action.

4.4.1 Smart Motor Driver Flashing Procedure

As the smart motor drivers do not have any ports for uploading code to the microcontroller via USB, the microcontroller was flashed using in-system programming, or ISP. ISP flashing can be accomplished by having six contact pads on the board that connect to power, ground, MISO, MOSI, SCLK, and RESET, lines on the ATMEGA328 microcontroller (Figure 73), and using a TC2030-NL six pin pogo connector from TagConnect with an ISP programmer [31].

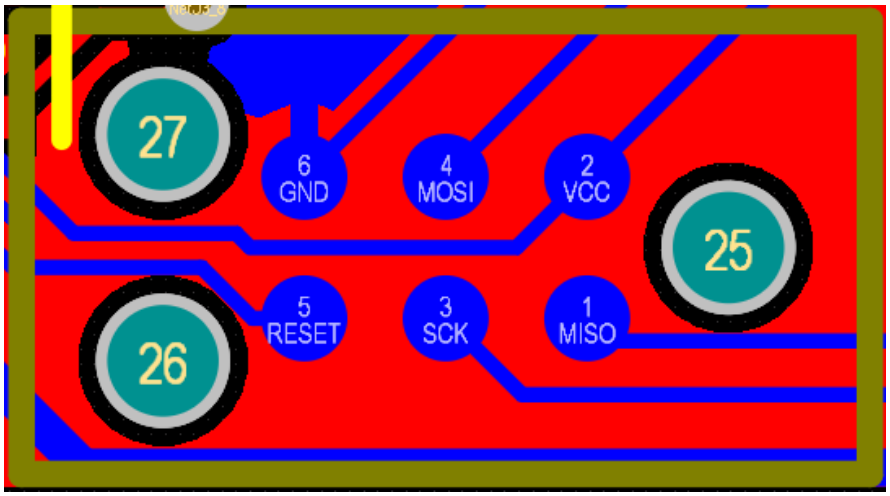


Figure 73: Six Pin ISP Contact Pads

For an ISP programmer, the team chose to utilize an Arduino Uno board with an external breadboard circuit (Figure 74). The external circuit makes use of a crystal oscillator and pull down resistors so that the Arduino Uno can be used as an ISP.

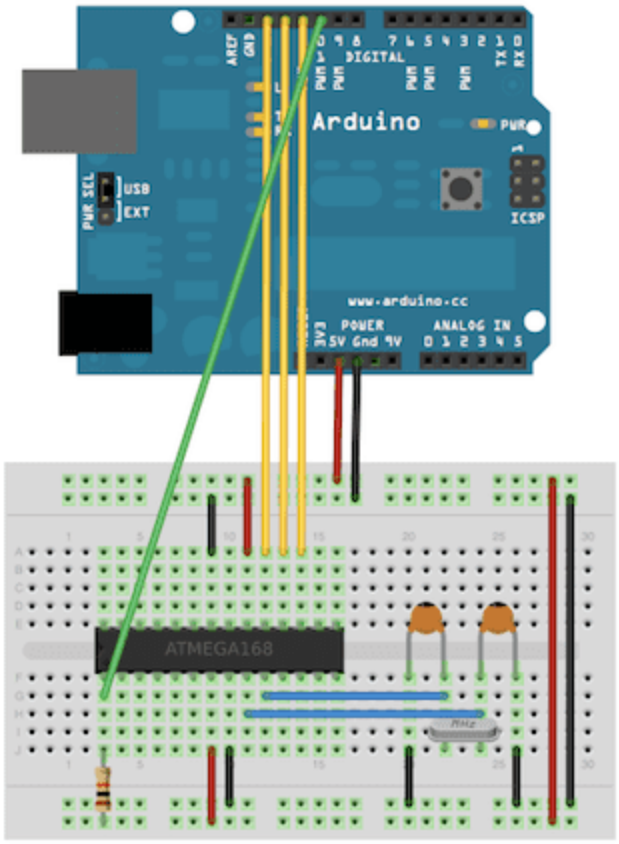


Figure 74: Arduino Uno with Flashing Circuit [43]

To upload code to the microcontroller onboard, the Arduino IDE was used as it already provides a built-in interface for using an Arduino board as an ISP programmer. Using this interface only requires uploading the Arduino as ISP example sketch provided with the IDE to the Arduino Uno, and then setting up the circuit shown above [44]. Then, code can be uploaded by selecting the onboard microcontroller as the target board and then using the option to "Upload Using Programmer" to flash the microcontroller.

In the particular case of the ATMEGAs we used, some additional steps were taken to enable the use of existing Arduino libraries for the Arduino Nano board so firmware writing was not required. The microcontroller on board of the Nanos is an ATMEGA328-P, however due to ongoing microchip shortages, the team was only able to acquire ATMEGA328's, which carry a different chip signature than the 328-P's. Since using an ISP programmer is dependent on having the proper chip signature for the target, we applied external libraries to make the Arduino Nano libraries compatible with the smart driver microcontrollers. The Minicore Arduino Library provided a workaround for the problem described above, as it enabled support of the Arduino Nano libraries for other variants of the ATMEGA328 chips [45].

When uploading code to the smart motor drivers, there were several modifications that could be made to the above flashing circuit to simplify usage. One method used by the team involved using the flashing circuit to initially burn a bootloader onto the ATMEGA, and then

rather than using ISP, code could be uploaded using UART by simply hooking up the TX and RX lines to the Arduino Uno and removing the chip on the Uno. This method takes advantage of the serial programmer onboard the Uno and was a simpler method for uploading code once TX and RX pins were broken out in later iterations of the board. However, ISP was still the favored method for uploading code to the smart motor drivers, as the TX and RX lines were typically occupied as they were used for serial communication for debugging during usage.

When assembling the robot, the team attempted to use UART to flash the lead screw motor. However, the team experienced issues as the RESET connection was not synced from the smart motor driver to the mainboard and to the external Arduino UNO. Therefore, the team soldered another six pin pogo connector directly onto the contact pads of the smart motor driver, flashing the board over SPI through a cable threaded through the robot.

4.4 Programming

4.4.1 External ESP32

The external ESP acts as the interpreter between the Logitech gamepad and the mainboard. In its program, called pico_sender, it reads the 43 bit data via Serial from the gamepad, and stores the data into a character array. Through a series of if statements in the function interpretData, each bit is checked to determine which buttons are being pressed, and a corresponding command is returned based on a generated protocol designed to map button inputs to smart motor driver controls as shown below in Table 2. The bit value is then translated into a character command that is sent to the mainboard to interpret and process.

To achieve greater functionality of the robot, we used a combination of singular and multi-inputs from the gamepad. Using the Serial interface from the gamepad, we can read in multiple inputs at a time, allowing for the creation of “profiles” in which we can program custom or special actions. For example, we programmed the Y button to increase the robot’s diameter using the lead screw and the right bumper to drive the wheel motors forward at a preset fast speed. When pressed together at the same time, however, the robot’s diameter increases until the current sensors detect a high current threshold. This allows for more intuitive control using singular inputs for simple tests and new users, while allowing more complex functionality for combination inputs.

Input	WiFi Sender Command	I2C Status Command	Smart Motor Driver Action
Default	“0”	0	Brake all motors
Y Button	“1”	1	Rotate lead screw (0x07) - increase diameter
X Button	“2”	N/A	Request encoder, current sensor, motor rpm data from all motors
A Button	“4”	2	Rotate lead screw (0x07) - decrease diameter

Right Bumper	“8”	8	Drive wheel motors forward - fast speed
Left Bumper	“9”	9	Drive wheel motors reverse - fast speed
DPad 0 (North)	“10”	0	Brake all motors - homing
Dpad 1 (Northeast)	“11”	N/A	Decrease all cable lengths
Dpad 2 (East)	“12”	Cable 1: 12, Cable 2: 0, Cable 3: 0	Decrease cable 1 (0x04) length
Dpad 3 (Southeast)	“13”	Cable 1: 12, Cable 2: 12, Cable 3: 0	Decrease cable 1 (0x04) and cable 2 (0x05) length
Dpad 4 (South)	“14”	Cable 1: 0, Cable 2: 12, Cable 3: 0	Decrease cable 2 (0x05) length
Dpad 5 (Southwest)	“15”	Cable 1: 0, Cable 2: 12, Cable 3: 12	Decrease cable 2 (0x05) and cable 3 (0x06) length
Dpad 6 (West)	“16”	Cable 1: 0, Cable 2: 0, Cable 3: 12	Decrease cable 3 (0x06) length
Dpad 7 (Northwest)	“17”	N/A	Increase all cable lengths
B Button + D Pad 2	“18”	Cable 1: 13, Cable 2: 0, Cable 3: 0	Increase cable 1 (0x04) length
B Button + D Pad 4	“19”	Cable 1: 0, Cable 2: 13, Cable 3: 0	Increase cable 2 (0x05) length
B Button + D Pad 6	“20”	Cable 1: 0, Cable 2: 0, Cable 3: 13	Increase cable 3 (0x06) length
B Button + Right Bumper	“21”	10	Drive wheel motors forward - PID slow speed
B Button + Left Bumper	“22”	11	Drive wheel motors reverse - PID slow speed
A Button + Right Bumper	“23”	3	Rotate lead screw (0x07) - decrease diameter until current limit
Y Button + Right Bumper	“24”	4	Rotate lead screw (0x07) - increase diameter until current limit

Table 2: Button to Control Mapping Protocol

The program features two callback functions with custom structs for data storage as shown in Figure 75. These callback functions are defined as part of the ESP-NOW framework

and specified in the setup of the program. ESP-NOW requires these structs to match across WiFi devices so the data sent from one device is expected by the other. Once the structs were defined, we created an instance of each struct that we populated with data when sending information to a WiFi device.

```
typedef struct data_struct_rec { //data struct to receive wifi commands from the exter
  String wifiData;
} data_struct_rec;

typedef struct data_struct { //data struct to receive from the mainboard containing ad
  // data types must match data struct that mainboard is expecting
  int smdAddress;
  int currentData;
  int encoderData;
} data_struct;

data_struct_rec test; //create instance of sending struct to be populated with data
data_struct receiveData; //create instance of receiving struct
```

Figure 75: pico_sender WiFi Data Struct Components

The first callback function, OnDataRecv, is called automatically when the mainboard sends data via WiFi to the external ESP. The mainboard only sends data via WiFi when the external ESP requests encoder, current sensor, and motor rpm data via the X button on the gamepad. The function prints the address of the motor, the current sensor data, and the encoder data for debugging purposes.
The second callback function, OnDataSent, is called automatically when the external ESP sends data to the mainboard via WiFi. This function prints out a status message on Serial to ensure the message successfully sends.

4.4.2 Mainboard

The mainboard is the communicator between the external ESP and the Smart Motor Driver boards. Pico_master, the program uploaded to the TinyPICO, initially registers ESP-NOW, establishes a connection over I2C with all connected Smart Motor Drivers, printing their addresses in the Serial Monitor to confirm their connections, and ends the I2C transmission.

Upon receiving a command from the external ESP, the OnDataRecv function is called automatically through the ESP-NOW protocol. The function converts the input character data from the pico_sender program to an integer, and then checks the data amongst several if statements for all expected integer inputs. Each if statement corresponds to the mainboard sending an I2C command to a Smart Motor Driver address via a function called sendMsg and the command protocol shown in Table 2. To command the motors to move, we created separate functions to control the drive motors (0x01, 0x02, 0x03), the cable motors (0x04, 0x05, 0x06), and the lead screw motor (0x07) separately.

RequestData, on the other hand, is called when we want to receive data back from the Smart Motor Drivers over I2C. We call this function sequentially, allowing us to receive data from all connected Smart Motor Drivers at once to debug any potential sensor failures. While the boards feature Serial capability, using I2C eliminates the addition of two more wires per motor driver, and is generally quicker than Serial communication. This function individually reads the data sent along I2C. First, it reads the address of the motor that it is receiving data from. Next, it performs two reads for the two encoder data bytes sent from the Smart Motor Drivers. To piece them back together, the first byte is stored in an integer variable count, while the second byte is appended to the end of the count variable by shifting the first value by eight bits. Lastly, the current sensor reading and motor rpm are read in and stored. To prevent overflow errors on the encoder count for Serial debugging and prolonged encoder usage, the count is inverted and subtracted from the maximum 16 bit value, 65535. The current sensor value, which is sent by the Smart Motor Drivers as an ADC value, is divided by the resolution of the ADC and the sense resistor value of 0.5 Ohms, as shown below in Equation 6.

$$I = \frac{V}{R} = \frac{(ADC\ Reading * \frac{1}{ADC\ Resolution})}{R_{SENSE}}$$

Equation 6: Current Sensor Analog to Digital Conversion Equation

4.4.3 Smart Motor Driver

Using the above structure to move communication functionality away from the Smart Motor Drivers, we were able to program each driver with the same code, where the only difference was the I2C address of each microcontroller. Therefore, the communication functionality is largely controlled by the pico_sender and pico_master programs, while the control processing is controlled by the SAMI_nonreversed code.

In the setup function of the program, we first open the I2C bus for the specified address, and set up the sensor pins for the current sensor and the two hall-effect encoder sensors. We then attach software interrupts on each encoder pin, with two separate interrupt service routine functions that trigger on a value change.

Initially, we used a singular ISR to keep track of the quadrature encoder count. In this ISR, we called digitalRead twice, once for each encoder. We then appended the values together using a bitshift and an or operator. To determine the direction of that the motor was spinning, we created a two dimensional lookup table as shown below in Figure 76, which mapped the current encoder reading to the previous to determine the change in encoder ticks.

New value

Old value	00	01	10	11
00	0	+1	-1	X
01	-1	0	X	+1
10	+1	X	0	-1
11	X	-1	+1	0

0	Unchanged
+1	increment
-1	decrement
X	Invalid

Figure 76: Quadrature Encoder Decoding Lookup Table [46]

However, we noticed that I2C commands would not send properly when the encoder ISRs fired, due to the low interrupt priority of I2C and the frequency of encoder interrupts. To solve this issue, we attempted to optimize the ISRs for encoders. First, we removed the digitalRead calls and replaced them with direct register inquiries, which are much faster computationally. Next, we split up the singular ISR into two identical ISRs that can be activated by either hall effect sensor. This lets us read the registers and update the count variable without using a 2D lookup table array. We found that this strategy was less computationally expensive and allowed for I2C commands to be sent without interruption.

When the Smart Motor Driver receives a command via I2C, we store the integer into a variable called I2Cstatus. This variable is then used in a switch statement, where each I2C command is translated into actuation using one of three functions: forward, reverse, or brake, at varying speeds, depending on the given command. These functions change the output voltage from the ATMega328 on pins 9 and 10 to power the motor driver according to the H-bridge logic table from the MD9927 motor driver datasheet, as shown below in Figure 77.

Table 1.MD9927 Device Logic

nSLEEP	IN1	IN2	OUT1	OUT2	FUNCTION (DC MOTOR)
0	X	X	Z	Z	Coast
1	0	0	Z	Z	Coast
1	0	1	L	H	Reverse
1	1	0	H	L	Forward
1	1	1	L	L	Brake

Figure 77: MD9927 H-Bridge Logic Lookup Table [30]

To incorporate control in the system, we created PI controllers for both position and velocity. The calcPID function, used for controlling motor speed, calculates the effort by multiplying the input error by a proportional constant, which we determined to be 4 through experimentation. A result variable is updated depending on the sign of the error, and then scaled

between a range of preset speed values. Since we use analogWrite to set a PWM duty cycle, our maximum value for speed is 255. However, we encountered issues with I2C interrupts activating due to the ISRs firing too quickly, and reduced the maximum speed of our motors to 200, as shown below in Figure 78. Only proportional control was used here as the system did not require integral or derivative terms to consistently reach its setpoint within a reasonable range, likely because the system is already relatively damped due to its construction.

```
/**
   This function calculates PID for the motor speed
   @param error - the given error
   @return result - an integer clamped value for the speed of the motor
*/
int calcPID(int error) {
  result = abs(error * kp);
  if (error < 0) {
    result -= result;
  }
  else {
    result += result;
  }
  result = max(min((result), 200), 0);
  return result;
}
```

Figure 78: calcPID Function Code

In the loop function, we calculate our error using the difference in encoder counts from the current measurement to the previous measurement over a constant timestep of 20 milliseconds as shown in Figure 79. This difference is then converted into RPM as shown in Equation 7, and then subtracted from the target motor speed to become the motor speed error passed in as a parameter in the calcPID function.

```
void loop() {
  if ((millis() - lastTime) >= 20) { //calculate PID every 20ms

    //calculate current RPM and compute PID with it
    countdiff = count - lastcount; //difference in encoder count
    lastcount = count;

    //calculate motor speed in RPM
    motSpeed = abs(countdiff * 0.84); //((1000*60)/(12*20*Ngear)))
    currenterror = targetSpeed - motSpeed; //calculate current error from our target speed
    calcSpeedPID = calcPID(currenterror); //calculate PID
    //cablelength = calcCablelen(count)*10;

    lastTime = millis(); //update timer
  }
```

Figure 79: PID Velocity Error Code

$$countdiff * \frac{(1000 \frac{ms}{s} * 60 \frac{s}{min})}{12 \frac{ticks}{rev} * 20\, ms * N_{GEAR}} : N_{GEAR} = gear\ ratio,\ 298\ for\ drive\ motors$$

Equation 7: Encoder Difference to RPM conversion

PID position, on the other hand, takes in a target position and calculates an effort that is directly used to spin the motors forward or reverse, causing a variable speed. To avoid overshoot, we clamped the maximum absolute value of the speed to 100, and allowed the motor to spin both forward and reverse to settle at the position setpoint as shown below in Figure 80. The function returns a boolean indicating if the motor arrived at the setpoint, and continuously runs in the loop function until the desired setpoint is reached.

```
bool pidposition(long pidtarget) {
  float ppos = 0.1;
  float ipos = 0.0;
  float dpos = 0.0;

  long error = pidtarget - count;
  uint16_t adjeffort = ppos * error;
  adjeffort = abs(max(min(100, adjeffort), 0));
  if (error >= 0) {
    reverse(adjeffort);
  }
  else {
    forward(adjeffort);
  }
  if (abs(error) < 100) {
    return true;
  } else {
    return false;
  }

}
```

Figure 80: PID Position Function Code

4.4.4 Motor Stall Protection

Within the code onboard the smart motor driver, the current draw of the motor is calculated on the same interval as the timer for PID velocity control. By experimentally determining the thresholds that correspond to motor stall, we added functionality that protects the motor against stall when it recognizes the current being above the stall threshold. This code shuts off the motor if it stalls by checking the current sensor reading for more than 2 timer intervals (40 milliseconds). The primary factors for implementing these protections were maintaining the motor functionality and limiting battery draw to maximize battery life.

Additionally, this functionality was repurposed on the lead screw motor to allow the suspension to not only retract itself all the way down to achieve minimum diameter but also pretension the suspension within pipes by allowing the leadscrew to proceed until slightly before stall, which was experimentally determined to be a satisfactory amount of wheel compression. This was implemented to protect the motor and power consumption but also improve ease of use for the user driving the robot.

5. Results

In this section, we discuss the results from rigorous testing of the C.L.A.R.A. robot and the final design, implementation, and status of the project.

5.1 Task Completion

Sections 3.2 and 3.3 detail the objectives and design specifications we laid out during the initial planning phase of our Major Qualifying Project. We have summarized the final outcomes of our project below in Table 3.

Task #	**Specification**	**Status**
1	The total cost of fabricating a single module should not exceed $500	Complete
2	The module will consist of an origami midsection.	Complete
3	The module will be able to be able to travel within a minimum pipe size of 4“ and a maximum pipe size of 6”	Complete
4	The module will be able to travel at a minimum speed of 4 in/s.	Complete
5	The module will be capable of traveling through both concentric and nonstandard expansions and contractions of pipes within its specified size range.	Complete
6	The module will be capable of traversing steep inclines, as well as able to travel in completely vertical pipes within its specified size range.	Complete
7	The module will be able to maneuver through 90 degree bend radii.	Complete
8	The module will not require a tether for operation.	Complete
9	The module will use individual PCBs for the control of each motor, communicating with a main board facilitating communication and power distribution.	Complete
10	The module will make use of current sensors on all motors to detect real time current usage and avoid stalling of motors.	Complete

Table 3: Task Completion Table

5.2 Final Mechanical Design and Fabrication

Shown below in Figure 81 is the completed design and fabrication of the pipe inspection robot. The robot features a continuum origami midsection, with an active suspension in the front and a passive following suspension in the back. The robot weighs approximately 500 grams.

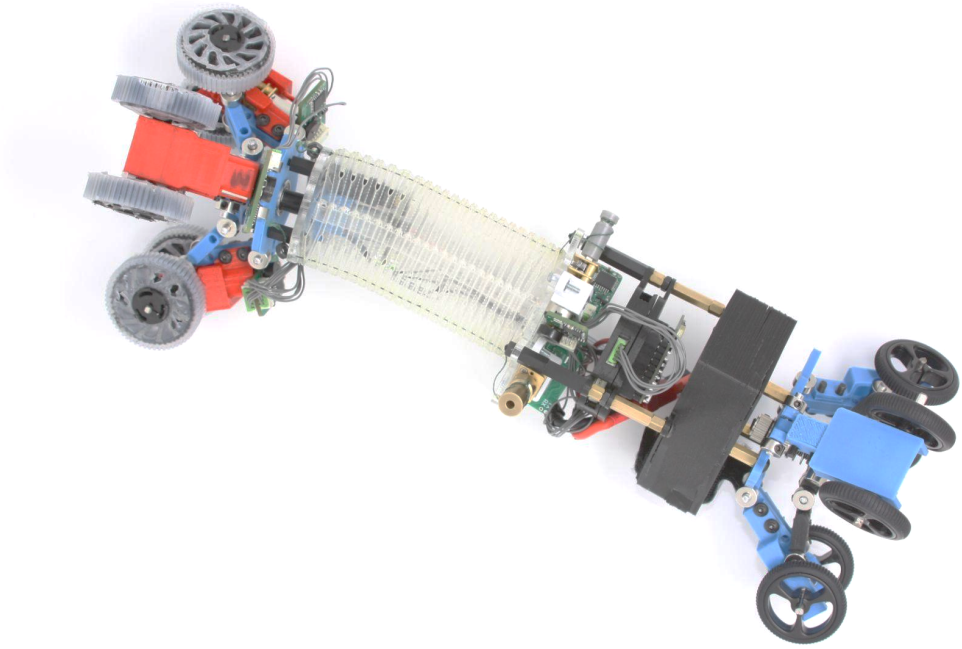


Figure 81: Final C.L.A.R.A. Pipe Inspection Robot

5.2.1 Yoshimura Origami Module Midsection

The final design for the origami midsection is shown below in Figure 82. The module's maximum length is approximately 6" when fully uncompressed, while its minimum length is approximately 1.5" when fully compressed. The module is driven by three microfilament braided fishing line segments rated for 50 lb [22]. Each segment is attached to 1:298 N20 motors rated at 6V and 3.5 in-lb of stall torque from Pololu [47]. When certain cables are contracted while others are left extended, the module is able to create a bending radius which is incredibly important for navigating through pipe bends. The module's minimum turning radius was measured to be 85.8 degrees.

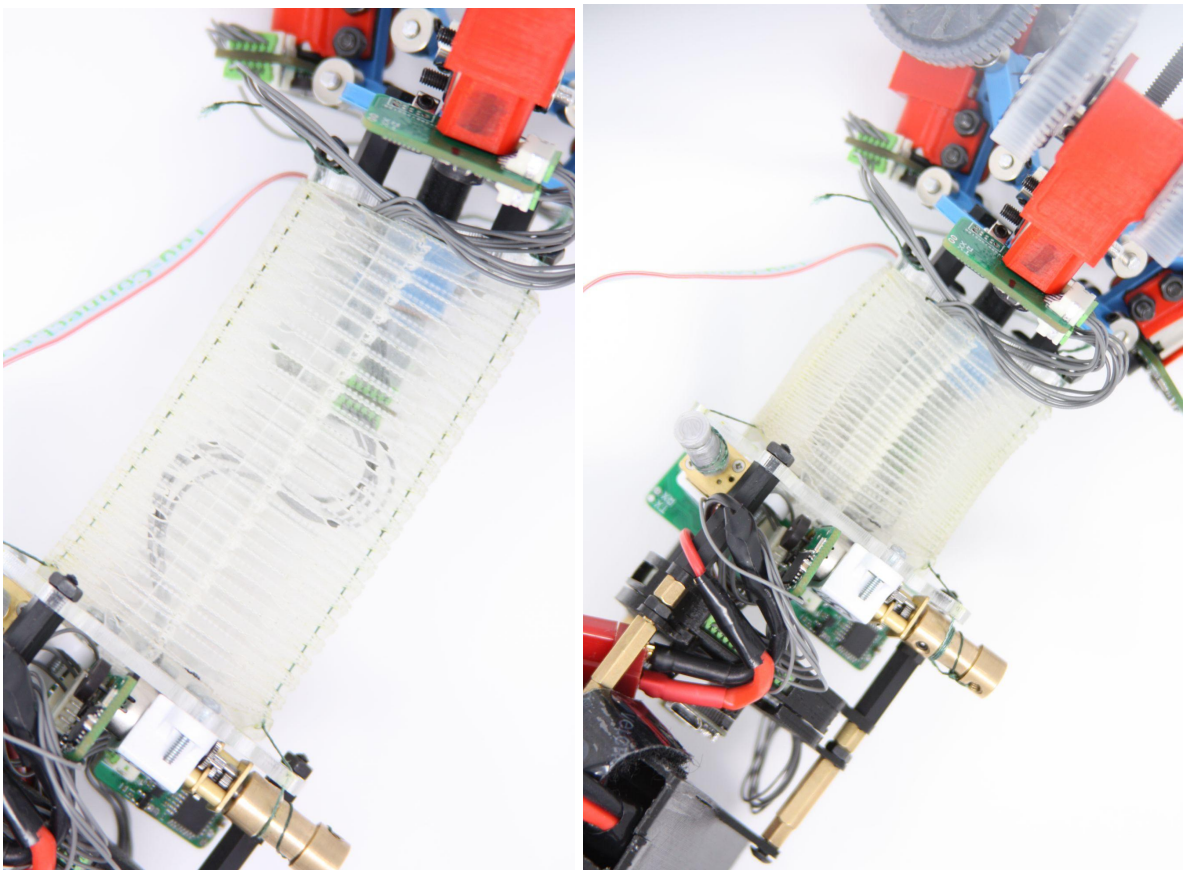


Figure 82: Final Yoshimura Origami Midsection (left: uncontracted, right: contracted)

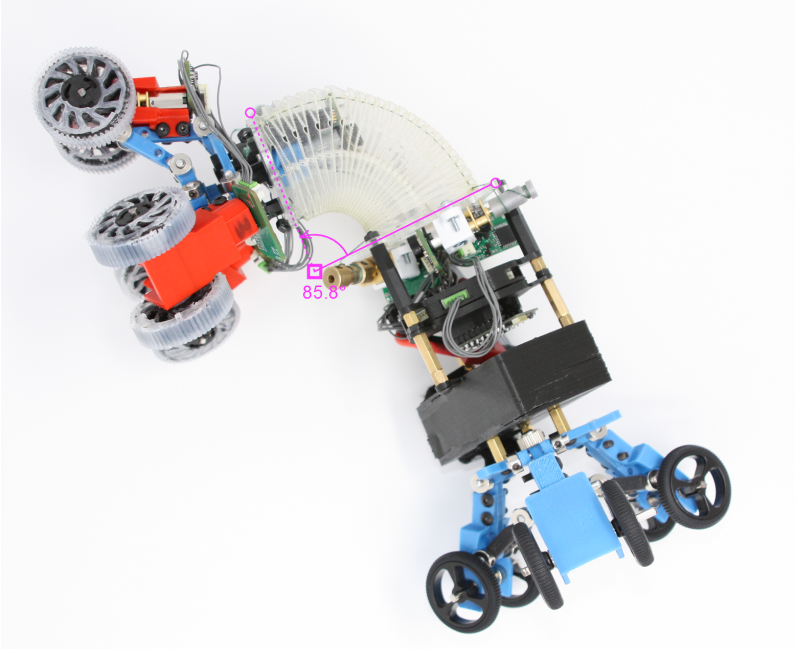


Figure 83: Yoshimura Module Minimum Radius

5.2.2 Active Suspension Mechanism

The completed active suspension mechanism is shown below in Figure 84. The mechanism features three, rigid slider-crank four bar linkage mechanisms attached to a central slider that translates along an M5-1.25 lead screw driven by a 1:150 N20 motor. On each linkage is a motor transmission module, described in Section 4.1.4, which powers the drive wheels of the robot. On each axle is a set of compliant TPU-Dragonskin wheels which can compress upwards of 0.3” to provide additional traction to the surfaces of the pipe. The diameter expansion mechanism’s minimum effective diameter is 3.6” and its maximum effective diameter is 5.23”, allowing us to drive through pipes between 3.5” and 5.5” using the compressive wheels to decrease the diameter even further.

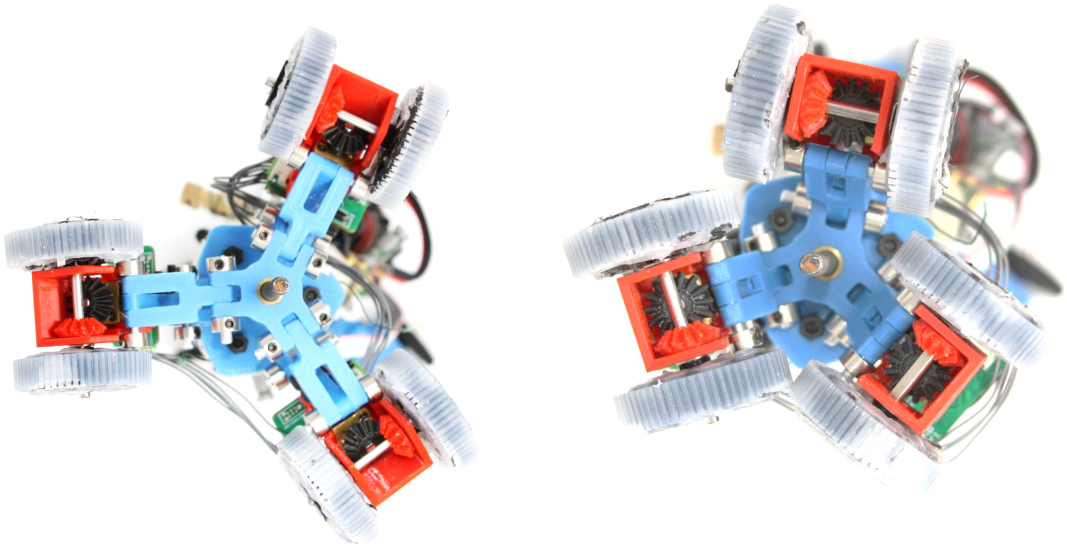


Figure 84: Final Active Suspension Mechanism (left: maximum diameter, right: minimum diameter)

5.2.2 Passive Suspension Mechanism

The completed passive suspension mechanism is shown below in Figure 85. The passive suspension mechanism acts as a follower, and passively conforms to the diameter of the environment using a combination of TPU NinjaFlex compliant links and a central compression spring.

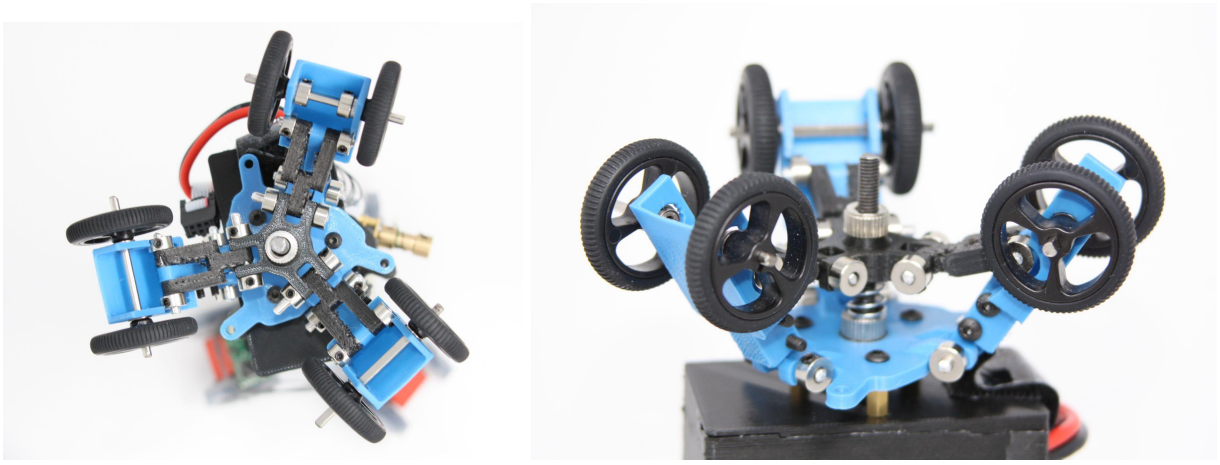


Figure 85: Final Passive Suspension Mechanism

5.3 Final Printed Circuit Boards

5.3.1 Smart Motor Driver PCB

The final version of the Smart Motor Driver PCB is shown below in Figure 86 (refer to Section 4.2.1.5 for schematic and layout diagrams). The boards feature an ATMega328 as the microcontroller, and include current sensing, quadrature encoder, and motor driving capabilities with onboard components. The boards feature two 5-pin JST connectors, to allow for easy daisy-chaining to other Smart Motor Driver boards.

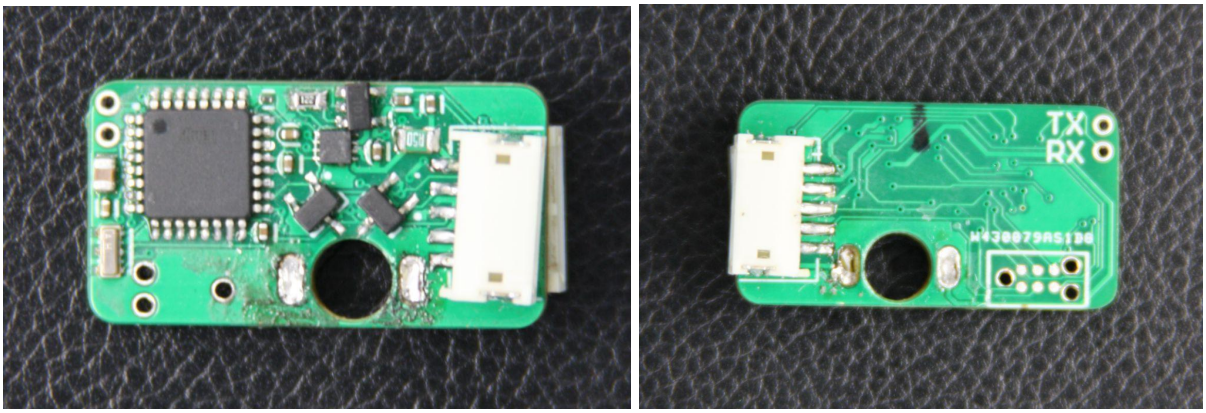


Figure 86: Smart Motor Driver PCB

The tests performed to validate each component as well as the completed Smart Motor Driver PCB are listed below.

- Send an output to the pins connected to the MD9927 driver using the H-bridge lookup table to ensure a motor can spin in both directions
- Send PWM signals at varying duty signals to change motor speed

- Send data back to the mainboard over I2C to verify component functionality and allow for debugging
- Read the current sensor output pin through the ADC at no load and stall conditions to validate current sensor functionality
- Create a stall prevention function using an experimental current sensor threshold and stop the motor once the threshold is reached to reduce power consumption and protect the motor
- Spin a magnetic encoder wheel over a hall-effect sensor and verify that a square wave is produced
- Create an interrupt on the hall-effect sensor pin and verify that exactly 12 encoder ticks are seen in one rotation while running the motor
- Create a second interrupt to enable both hall-effect sensors and verify that 24 encoder counts are printed to verify quadrature encoder functionality
- Create a PID controller to control the motor's speed for more controlled speed applications
- Create a PID controller to control motor position for precise movement of drive, lead screw, and cable motors

5.3.2 Mainboard

The final version of the mainboard PCB is shown below in Figure 87 (refer to Section 4.2.2.3 for schematic and layout diagrams). These boards include female header pins for a TinyPICO, used to control all connected Smart Motor Drivers via I2C and take in commands over WiFi from an external ESP32 device. While the board features nine 5-pin JST connectors for multiple connections, only one is required for the C.L.A.R.A. robot as each Smart Motor Driver can be individually addressed and daisy-chained.

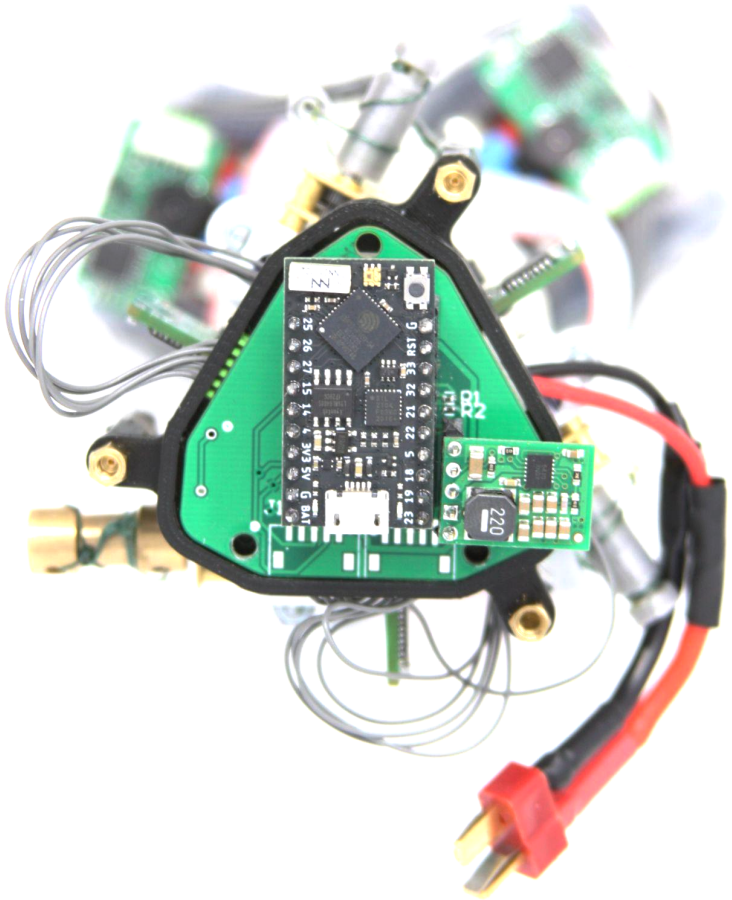


Figure 87: Mainboard PCB

The tests performed to validate the functionality of the mainboard PCB are listed below.

- Create an I2C connection between the mainboard and one Smart Motor Driver and verify that its address is found upon connection
- Using Serial Monitor input, command the mainboard to send an I2C signal to a connected Smart Motor Driver board to rotate a motor to verify I2C commands
- Connect multiple Smart Motor Drivers to the mainboard and command each one to spin at a different speed to verify individual addressing of Smart Motor Drivers
- Send a command from an external device using ESP-NOW and WiFi to the mainboard to enable WiFi communication
- Relay data from the Smart Motor Driver through the mainboard and to the external ESP to verify bi-directional WiFi communication and enable debugging through a remote device

5.4 Programming and Control

5.4.1 Control System and Power Distribution

The control system for CLARA consists of the electronics, the gamepad controller, and the 3 control C++ programs for standard operation, and 3 more programs for the inverse kinematics demonstration.

Throughout the testing process, our control system involving Serial communication through the gamepad and WiFi communication among the ESP32 and TinyPICO was very reliable and responsive to input. Using I2C to control each of the Smart Motor Drivers created a robust system, where each motor was individually addressed and controlled via a series of preprogrammed I2C status commands.

The power distribution held up well during our testing, allowing us to test upwards of 15 minutes on a fully charged battery in conditions where motors could stall and require high current load.

5.4.2 Inverse Kinematics

To quantify the motion of our robot, we implemented an inverse kinematics control model to characterize the Yoshimura module as shown below, where the configuration parameters are κ, the curvature of the robot, φ, the bending direction around the z-axis, and s, the arc length of the module [10].

$$l_1 = 2n \sin\left(\frac{\kappa s}{2n}\right)\left(\frac{1}{\kappa} - d\sin(\phi)\right) \quad (1)$$

$$l_2 = 2n \sin\left(\frac{\kappa s}{2n}\right)\left(\frac{1}{\kappa} + d\sin\left(\frac{\pi}{3} + \phi\right)\right) \quad (2)$$

$$l_3 = 2n \sin\left(\frac{\kappa s}{2n}\right)\left(\frac{1}{\kappa} - d\cos\left(\frac{\pi}{6} + \phi\right)\right) \quad (3)$$

Figure 88: Inverse Kinematics Equations for the Yoshimura Module [10]

These equations work for any Yoshimura module where three cables are used. The only parameters that change between modules are *d,* or the distance from the center of the Yoshimura module to the cable attachment point and *n*, the number of connected sections.

We created another program that allowed us to demonstrate the effectiveness of these equations by inputting the configuration parameters s, Θ (the bending angle which is equal to s times κ), and φ into the Serial monitor connected to our external ESP, calculate the cable lengths and send them to our mainboard. On the mainboard side, we convert the cable lengths to absolute encoder counts and send the corresponding integer counts to cable motors.

Using a PI controller for encoder position, we drive the motors until they achieve their setpoint. As we regularly send encoder data from the Smart Motor Drivers to the mainboard, we

utilized the data to increase the performance of our PI controller. We created a sine function and sent each point in the function as a setpoint for the Smart Motor Drivers. By taking the difference between the input setpoint and the received encoder data, we tuned the system until the error was minimal for both the step response and changing position trajectory as shown below in Figure 89.

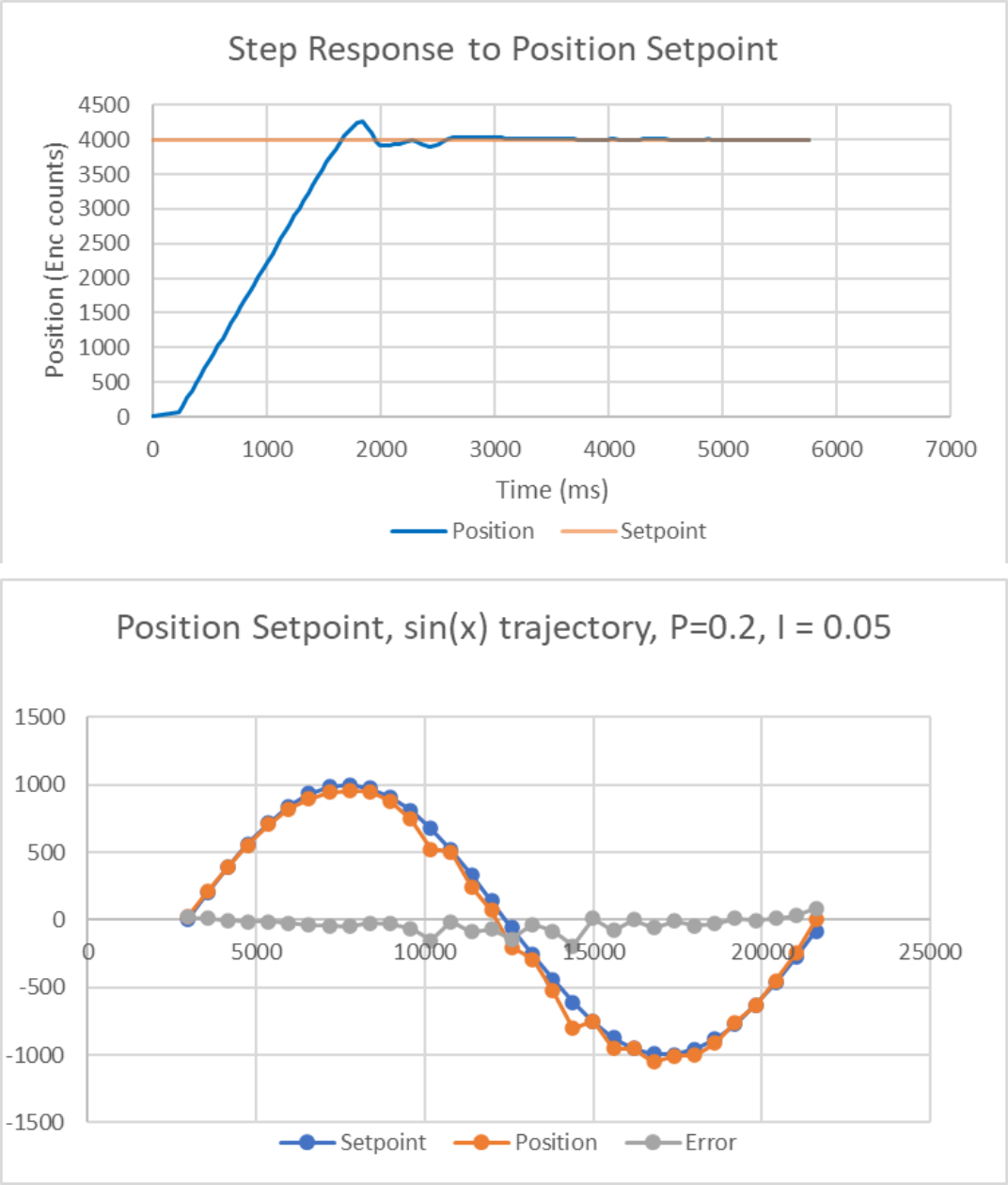


Figure 89: PI Controller Tuning Results

We found that the equations mathematically fail when we attempt to go back to an upright position ($\Theta = 0$), so we bypassed the equations when sending the values 0,0,0 for s, Θ, and φ to reset the encoder counts to zero.

We noticed throughout our testing that the slack in the fishing line drastically impacted the end result of our inverse kinematics. While the encoders may have rotated to the correct position, the slack in the cables prevented the robot from achieving the desired position. Therefore, we properly tensioned the system at its most extended point to ensure the system is tensioned in all configurations.

5.4.3 Further Information

For more information on the operation of the programs, refer to Appendix B for the README of the project.

5.5 Testing

Throughout the course of the project, we subjected the C.L.A.R.A. robot to challenging environments within pipe systems and evaluated its performance.

5.5.1 Horizontal and Inclined Flat Plane Test

The robot was tested on a horizontal surface made of wood to test the basic driving capabilities. When driving on a flat surface, there were two configurations: one in which 2 sets of wheels are making ground contact at a given time, and one in which only a single set makes contact at a given time. For purposes of speed testing, the single set configuration was tested. Using a video filmed from above of the robot driving in which the single set point of contact configuration was used, the maximum travel speed of the robot was determined analytically using the Tracker [48] software. By graphing the y-position of a point of the robot over time, as the robot traveled vertically within the video frame, the graph shown in Figure 90 was produced. From a linear regression on these data points, the travel speed in this configuration was determined to be 15.96 cm/s.

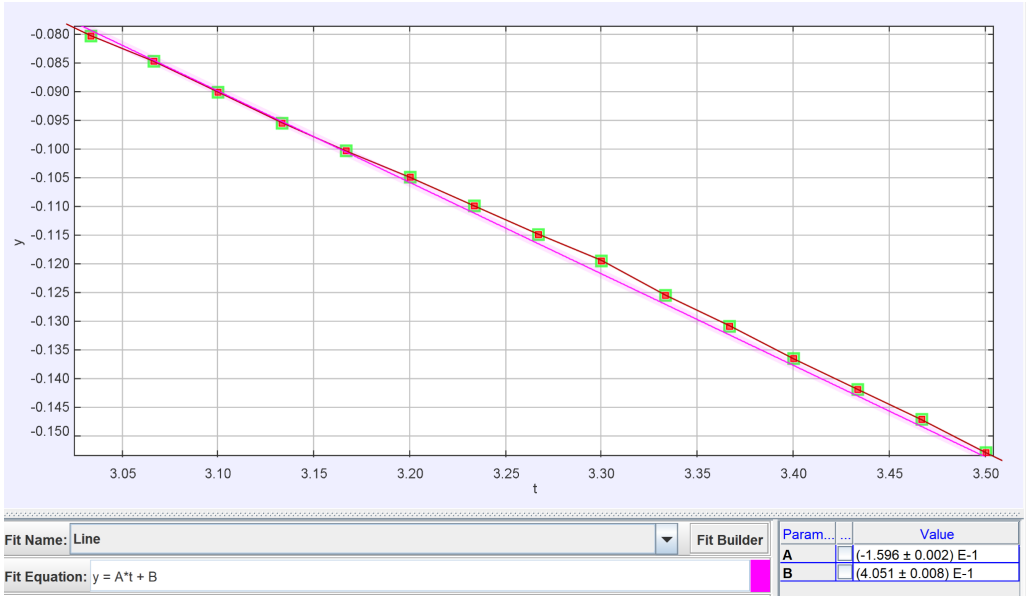


Figure 90: Robot Driving Speed Graph

The robot was also tested on a flat wooden surface with a variable incline to determine the maximum angle that could be traversed. When the driven wheels are in front with two sets of wheels contacting the surface, the robot achieves a maximum incline of approximately 30 degrees. When two sets of driven wheels are in the back, the robot achieves a maximum incline of approximately 45 degrees. When only one set of driven wheels was used, wheel slippage occurred at approximately a 60 degree incline.

5.52 Horizontal and Vertical 4” Pipe Test

PVC piping with a 4” OD and schedule 40 thickness were purchased for basic pipe traversal testing. The robot was first tested within this pipe, only needing to drive horizontally. This test was successful upon the first try.

The robot was then tested on its ability to traverse this pipe with it in a fully vertical configuration (Figure 91). This test was originally performed with uncoated TPU wheels, which experienced too much slippage to climb vertically. Once these wheels were replaced with silicone-coated TPU wheels, this test was successful. We noticed that the robot performed better when the driven wheels followed the passive wheels in a vertical configuration.

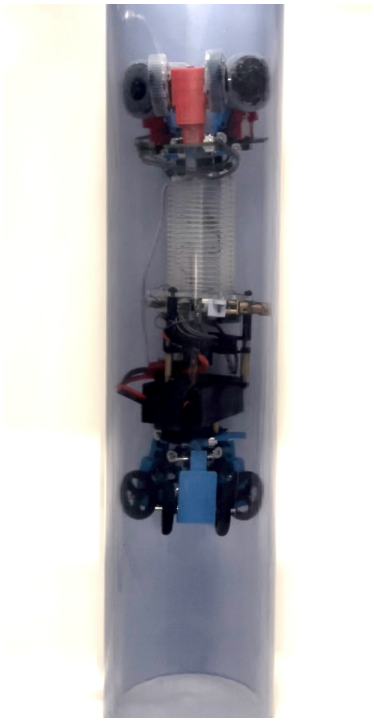


Figure 91: Vertical Pipe Traversal

5.5.3 Bend Test: 45 Degree

To test the bending capabilities of the robot within a pipe, a 45 degree short-radius bend was attached to the 4” straight pipe. The robot was first tested with the active-suspension side leading through the bend, which was successful but required many adjustments of the lead screw mechanism to accommodate the changes in diameter at the connection site of the two pipe sections. The robot was then tested with the passive-suspension side leading through the bed, and was found to succeed at the task and was performable with more ease than the latter test.

5.5.4 Tee Test- 45 Degree

To test the abilities of the robot within a Tee intersection, a 4” 45 degree tee was added to the straight pipe section. The robot was sent through this tee with the active-suspension side leading, and was able to successfully pull itself through the branched pipe section of the tee.

6. Discussion

6.1 Test Performance

6.1.1 Mechanical System Robustness During Testing

While the system mostly held up well mechanically during testing, there were a few instances in which testing had to be halted due to failure of a single component. For example, the bevel gears within the transmission were frequently known to wear out where the d-shaft is inserted, which halted testing until new sets could be printed. Additionally, the transmission housings were prone to cracking, along with the acrylic mounting plate that holds the cable motors. The team believes these failure points to be due simply to unforeseen stresses that occur during testing that could not have been predicted prior. While some changes were made to mitigate these issues, such as switching from prints made out of PLA to PETG, further work could certainly be done to improve the robustness of the robot.

6.1.2 Control System Reliability and Ease of Use

Generally, the control system for CLARA was found to hold up well during testing, be relatively easy to use, and account for most scenarios faced during testing. However, there are a few features that could be added in the future to improve this. For example, code was added to tension the active suspension within a pipe based on feedback from the current sensor. However, being that a number of factors affected this measurement, such as wheel compression, battery level, and how leveled the robot was within the tube, this was not always the most reliable feature. The team believes more improvement could be made in the future to continue to better this feature and improve robot maneuverability. Additionally, it was often hard to tell what the current battery level was when the robot was in testing, which led to a few scenarios in which the battery died when the robot was in a position that would be complex to remove by hand. In the future, changes to the electrical system to read battery levels could be helpful to provide some warning to the driver.

Generally, while data regarding the state of the robot's encoders, velocities, and more were able to be requested on a button-press, it was often hard to tell more data regarding the state of the robot, such as battery level, current draws of motors, and more. In the future, there could also be potential for development of some dashboard user interface that can relay this data in a more efficient and usable way to the driver.

6.2 Mechanical Assessment

On the basis of the testing matrix and objectives, the robot was mechanically able to accomplish all of the tasks. However, there are still some changes the team feels would benefit and improve future functionality of the robot.

6.2.1 Body Design

After evaluating various designs for linear origami actuators and deciding to pursue the Yoshimura folding pattern, we are very happy with our decision. The torsional strength of the design ensures that the passive suspension mechanism does not twist and make locomotion more difficult within the pipe. The linear compression and precise control using three cables allows us to attain all desired positions, overcome challenging turns, and navigate through a non-uniform maze. However, fabricating a Yoshimura module at such a small scale presents many manufacturing challenges and is incredibly tedious to fold and assemble. We believe that the module material can be changed to be less stiff, possibly with a thinner PET sheet, which will improve the folding of small components such as the folded key-holes.

Additionally, we believe that the attachment of several Yoshimura patterns to form a module can be improved. Rather than using a key-slot system with very small cutouts, we believe that a more efficient method can be used. For example, the PET sheets can be glued together at the attachment seam, which acts as a more permanent connection. The angle at which the keys fold inward can also be changed to make it easier to fit within the slot and unfold, forming a more stable connection.

6.2.2 Suspension Design

6.2.2.1 Active Suspension

While we evaluated the performance of multiple diameter expansion mechanisms, we believe that the slider-crank mechanism we chose for our final design can be improved. The current robot works for a limited range of pipe diameters which can be more challenging to acquire, as larger pipe diameters are more expensive. Additionally, we experimented with the use of compliant linkages in the system but found that the torsion created when the robot was placed in a pipe system hindered performance drastically. There is room for improvement in the design of the compliance of the system such that torsional effects can be minimized, similar to the design of the Yoshimura module.

We also encountered challenges in alignment of the lead screw in the shaft coupler, which caused the system to “wobble” when expanding or retracting the drive wheels. The current design requires a press fit of the lead screw, which was aligned and pressed by eye. For future applications, we recommend using an off-the-shelf shaft coupler with set screws or a proper alignment method to ensure the lead screw is flush with the coupler.

6.2.2.2 Passive Suspension

We were impressed by the performance of the passive suspension, as it was able to support the robot and maintain its position in horizontal, incline, and vertical pipe environments even when the active drive wheels were not contacting the pipe surface.

However, there are several improvements that we envision for future research. We found that the addition of compliant, NinjaFlex TPU links greatly improved the robot’s traction within vertical pipe scenarios. However, we believe that there is an opportunity to improve in

experimenting with cavities in the links themselves to reduce the force required to reduce the effective diameter of the robot.

Additionally, the material and spring constant of the central compression spring can be varied to make testing and operation simpler. We found that it took significant effort to insert the robot into the start of the pipe (which we performed by hand) due to the high spring constant of the compression spring. Including a weaker spring can potentially allow for remote-controlled entry but may harm the robot's traction on smooth pipe surfaces, such as PVC.

6.2.3 Compliant Wheel Design

We went through many iterations of the compliant wheels, and believe that they performed satisfactorily for the application. However, there is room for improvement in the material selection and cavity design to make wheels more suited for pipe environments. After experimenting with TPU, we found that the traction was minimal and the wheels began to slip unless they were significantly compressed. The addition of Dragonskin 10-NV as a coating to the tires increased the traction but was a short-term solution as the silicone coating began to tear after minimal testing. We found that adhering silicone to TPU was challenging and was only temporary. In the future, we recommend experimenting with pure Dragonskin wheels with minimal cavities to maximize wheel rigidity while maintaining the compliance necessary for increased traction benefits.

6.2.4 Other Mechanical Considerations

One of the challenges that we encountered throughout the course of the testing process is that additive manufacturing materials like PLA and PETG wear down over time and can drastically hinder the performance of the robot. Parts such as the bevel gears in the motor transmission had fatigue after hundreds of cycles in testing, and required replacements as their inner bore became too large for engagement with the motor shaft. When creating the transmission modules, we created a cutout on one side to allow for easier access to the 6-pin contact pad which was previously inaccessible due to the size of the connector. However, this made the module very flimsy and prone to fracture during high load applications such as traveling through a bend. We would improve the material selection of the components on our robot to be printed from stronger material to prevent wear and failure during testing and demonstration applications.

6.3 Control System Assessment

6.3.1 Sensor Performance

The smart motor driver boards have sensors in the form of encoders as well as current sensors. The encoders were found to perform satisfactorily with relative accuracy, so long as the magnetic encoder disk was placed such that it was close enough to engage the hall effect sensors but not so close that it caught on the board. This positioning is somewhat particular, and

on a number of occasions produced problems when the disks moved out of place due to jostling. However, the team found this issue to be slightly improved when switching from 9.7mm magnetic encoder disks to the smaller 7.65mm encoder disks. The smaller disks were thinner and appeared to fit more snugly on the motor shaft, and were less likely to move out of place.

Additionally, the aforementioned encoder interrupt service routines were prone to causing bugs with I2C communication when the motor was run at its highest speed. As a result, the motor speed was capped at 80% of its possible max speed. Ideally, the motor should be able to run at top speed, so to mitigate this issue, an encoder disk could be used that has less counts per rotation, as the level of precision that the current 12 CPR disks provide is much greater than required.

The current sensors were also found to perform satisfactorily. They allowed for identifying when the motor was stalled, which was the primary purpose in mind when they were added to the driver boards. However, while they could reliably identify stall, they were not precise enough to be used in other capacities, such as trying to identify when the motor was running at free speed or slipping. Additionally, the current sensor was found to be prone to a large amount of noise, which could also be improved upon in the future by filtering the data.

6.3.2 Control Algorithms

As mentioned previously, there are several control loops operating on the robot, primarily PID for velocity and PID for position. While both of these function relatively well and attain their respective setpoints satisfactorily, more work could certainly be done to improve this control. For example, the P term is the only one currently used as it produced satisfactory performance. However, utilizing the other terms could produce more precise and accurate performance.

Additionally, it was observed in testing that when the robot tried to traverse around bends and curves that it had some difficulty due to the fact that the wheels on the outside of the curve needed to travel at a faster speed than the wheels on the inside of the curve. This could be accomplished in the future with a more complex control loop that examines the current position of cables and could change speeds of wheels accordingly, however the team did not have the time to implement this type of control before the conclusion of the project.

6.4 Cost Assessment

The team performed a cost assessment of the materials used to construct the CLARA robot. As shown below in Tables 4, 5, and 6, we evaluated the cost of the mechanical components, the mainboard, and the smart motor drivers to a total of $439.65.

Part Name	Qty ordered	total cost	cost per component	components per board	cost per board
PCB Board	10	$5.00	$0.50	1	$0.50
ATMega328-AU	25	$79.10	$3.16	1	$3.16
SS360PT Hall Effect Sensor	20	$22.20	$1.11	2	$2.22
MD9927 Motor Driver	30	$5.19	$0.17	1	$0.17
ZXCT1010E5TA Current Sensor	30	$22.51	$0.75	1	$0.75
GRM155R71C104KA88D 0.1 uF capacitor	50	$1.10	$0.02	6	$0.13
CSTNE16M0V53C000R0 16 MHz oscillator	20	$5.46	$0.27	1	$0.27
GRM21BR61E106KA73L 10 uF capacitor	25	$7.30	$0.29	1	$0.29
ERJ-1GNF1002C - 10 kOhm Resistor	25	$1.33	$0.05	1	$0.05
RL1220S-R50-F - 0.5 Ohm Resistor	25	$6.08	$0.24	1	$0.24
ERA-3AEB122V - 1.2 kOhm Resistor	25	$7.48	$0.30	1	$0.30
S5B-ZR-SM4A-TF(LF)(SN) 5 pin JST connector	50	$30.46	$0.61	2	$1.22
				Total	**$9.32**

Table 4: Smart Motor Driver PCB Cost Assessment

Part Name	Qty ordered	total cost	cost per component	components per board	cost per board
PCB Board	5	$5.00	$1.00	1	$1.00
S5B-ZR-SM4A-TF(LF)(SN) 5 pin JST connector	50	$30.46	$0.61	9	$5.48
TinyPICO ESP32 Microcontroller	3	$60.00	$20.00	1	$20.00
D24V10F3 - 3.3 Buck Regulator	2	$14.98	$7.49	1	$7.49
RMCF0402JT4K70 - 10kOhm Resistor	25	$1.33	$0.05	2	$0.11
Dean's Connector	5	$9.49	$1.90	2	$3.80
				Total	**$34.08**

Table 5: Mainboard PCB Cost Assessment

Category	Part Name	Quantity	Unit Cost	Extended Cost
Body				
	PET Sheet	36"x10" sheet	$0.00	$0.00
	PowerPro 50lb fishing line	18"	$20.00	$0.03
	Cable Winches	3	$0.00	$0.00
	Acrylic Mounting Plates	2	$0.00	$0.00
	Motor Brackets	3	$3.95	$11.85
Suspension				
	3D Printed Parts	-	$0.00	$0.00
	5mm Ball Bearing	3	$1.00	$3.00
	5mm Shaft Collar	3	$0.78	$2.35
	M5 Lead Screw (4")	1	$0.00	$0.00
	M3 Shaft Collar	30	$0.27	$7.99
	M3 Assorted Screw Kit	1	$10.98	$10.98
	3mm Ball Bearing	12	$1.70	$20.38
	3mm D Shaft	20"	$0.00	$0.00
	Standoffs	12	$0.00	$0.00
	Dragonskin	0.25 bottle	$0.00	$0.00
	Pololu Wheels	12	$1.48	$17.70
Electronics				
	1:150 N20 Gearmotor	1	$3.00	$3.00
	1:298 N20 Gearmotor	6	$23.45	$140.70
	Smart Motor Drivers (components incl)	7	$9.32	$65.23
	Mainboard (components incl)	1	$34.08	$34.08
	Tiny Pico	1	$20.00	$20.00
	Buck Regulator	1	$6.95	$6.95
	ESP 32	1	$9.00	$9.00
	12 CPR Encoder Disks	7	$2.00	$14.00
	LiPo Battery	1	$10.95	$10.95
	6 pin pogo connector	1	$33.95	$33.95
	Logitech Gamepad	1	$16.39	$16.39
	6 cable JST connectors	7	$1.59	$11.13
Total				**$439.65**

Table 6: CLARA Cost Assessment

7. Conclusions and Recommendations

The CLARA MQP was designed on the premise that origami soft robots could prove to be advantageous in the particular application of navigating pipe systems and other complex environments, due to their unique ability to maneuver their bodies as well as being lightweight and lower cost than existing solutions. By utilizing the Yoshimura module with cable-driven bellows for a soft midsection, combined with a variable-diameter suspension system, the resulting robot was able to traverse not only flat surfaces and steep inclines but also through pipes of all orientations, maneuver through bends and other complex shapes, and travel through pipes of varying diameters.

The work that the team accomplished throughout the duration of this project leads the team to believe there is still room for exploration and improvement of the resulting system, and how soft robots can be used in this application and other relevant applications as well. At the conception of this project, the team set out to adapt and improve an existing soft salamander design to fit a new application. Throughout our project, the team designed, built, tested, and iterated upon not only several mechanical systems but control and electrical systems as well. Custom smart motor driver boards were developed and manufactured for the direct control of N20 gearmotors via I2C, which can be utilized in many other robots beyond the scope of this project.

Based on the results of our work, the team identified several recommendations that could improve this project in the future. First and foremost, shortening the length and reducing fold density of the Yoshimura module that comprises the body would likely benefit the functionality of the robot by requiring less torque and battery power to compress the robot length. It would also greatly improve manufacturing time, as folding the Yoshimura module by hand was tedious at the proposed scale. Researching and designing several diameter expansion mechanisms may yield immense benefits, as we found that our current implementation is limited in its diameter range and can be improved to achieve diameters as small as the robot's cross section. Improving the mechanical transmission design and construction would likely make the robot more robust and promote longevity, as the team experienced difficulties with the current transmission gears wearing out, as well as transmission housings cracking and breaking over time. The team spent time researching and experimenting various flexible materials, such as NinjaFlex TPU and Dragonskin to improve the robot's performance in both uniform and non-uniform environments. While this greatly improved the robot's performance in uniform pipe settings, we recommend that more time is spent researching and selecting materials to achieve desired properties such as compliance, traction, and manufacturability.

There is also potential to further experiment with different control and feedback loops to improve the maneuverability of the robot, as well as make it more intuitive to drive. Using a handheld controller was a great step in this direction, as the controls of the robot could be determined intuitively by a new operator. We recommend including augmented autonomy features into the system, such that the robot is able to automatically control its actuators by sensing the environment. For example, the robot can contract cables automatically if it senses

that it is encountering a bend, while a human simply commands it to go forward. This will make control more intuitive and introduce more complex features into the system that can increase robot driving performance. Furthermore, work could be done to examine how the robot could be operated autonomously, or perhaps work together with multiple CLARA modules at a single time to explore a vast pipe network more efficiently.

Additionally, we recommend an exploration into how various sensors could be added to this robot to turn it into a robot truly capable of inspecting and assessing its environment, such as cameras, temperature and pressure sensors, or potentially examining the possibility of the robot being able to carry tools or payloads to break away debris. Throughout implementation and further examination of these recommendations, the team believes the Continuum Locomotive Alternative for Robotic Adaptive-Exploration (CLARA) can be developed into a much more robust system with inspection and motion abilities unlike other robots currently used in this application.

References

[1] "Pipe Inspections," *Pipe Inspections - USA Borescopes*. https://usaborescopes.com/product-category/applications/pipe-inspections/

[2] "Video Borescope, 5.2mm Camera Head," *Grainger*. https://www.grainger.com/product/FLIR-Video-Borescope-60PT84

[3] "Watch Tower Robotics," *Watch Tower Robotics - Find Leaks - Save Water - Protect Infrastructure*. https://watchtowerrobotics.com/

[4] "TriTrax," *EddyFi Technologies*. https://www.eddyfi.com/en/product/tritrax-vertical-pipe-inspection-crawler

[5] Daniela Rus and Michael T. Tolley, "Design, fabrication and control of origami robots," *Nat. Rev. Mater.*, vol. 3, no. 6, pp. 101–112, May 2018, doi: https://doi.org/10.1038/s41578-018-0009-8.

[6] Cagdas D. Onal, Robert J. Wood, and Daniela Rus, "An Origami-Inspired Approach to Worm Robots," *IEEEASME Trans. Mechatron.*, vol. 18, no. 2, pp. 430–438, Aug. 2012, doi: 10.1109/TMECH.2012.2210239.

[7] Anna Mederer, Maria Medina Martinez, Selina Spry, Cagdas D. Onal, Berk Calli, and Jie Fu, "Flexible Robotic Origami Gripper (FROG)," *WPI Major Qualif. Proj.*, May 2021, [Online]. Available: https://digital.wpi.edu/pdfviewer/8s45qc56q

[8] N. Turner, B. Goodwine, and M. Sen, "A review of origami applications in mechanical engineering," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 230, no. 14, pp. 2345–2362, Aug. 2016, doi: 10.1177/0954406215597713.

[9] G. Hunt and I. Ario, "Twist buckling and the foldable cylinder: An exercise in origami," *Int. J. Non-Linear Mech.*, vol. 40, pp. 833–843, Jul. 2005, doi: 10.1016/j.ijnonlinmec.2004.08.011.

[10] J. Santoso, E. H. Skorina, M. Luo, R. Yan, and C. D. Onal, "Design and analysis of an origami continuum manipulation module with torsional strength," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 2098–2104. doi: 10.1109/IROS.2017.8206027.

[11] Ming Luo *et al.*, "OriSnake: Design, Fabrication, and Experimental Analysis of a 3-D Origami Snake Robot," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1993–1999, Jul. 2018, doi: 10.1109/LRA.2018.2800112.

[12] Y. Sun *et al.*, "Salamanderbot: A soft-rigid composite continuum mobile robot to traverse complex environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 2953–2959. doi: 10.1109/ICRA40945.2020.9196790.

[13] Paul Fitzgerald, "What is a Borescope," *AIT - Advanced Inspection Technologies*, Apr. 07, 2017. https://aitproducts.com/wiki/borescope.html

[14] Joe Escobar, "Borescopes: Some tips on effective inspection techniques," *Aviation Pros*, Nov. 01, 2001. https://www.aviationpros.com/home/article/10387908/borescopes-some-tips-on-effective-inspection-techniques

[15] "DT340L Pipe Crawler Package," *Deep Trekker*. https://www.deeptrekker.com/shop/products/dt340l-pipe-crawler-package

[16] "DT320 Mini Crawler Base," *Deep Trekker*.

https://www.deeptrekker.com/shop/products/dt320-mini-crawler-lite
[17] J. Min, Y. Liauw, P. Pratama, S. Kim, and H.-K. Kim, "Development and Controller Design of Wheeled- Type Pipe Inspection Robot," presented at the Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014, Sep. 2014. doi: 10.1109/ICACCI.2014.6968543.
[18] Brian Heater, "GE's worm robot sports roach-style whiskers to remove fat deposits from sewage pipes," *TechCrunch*, Mar. 08, 2022. https://techcrunch.com/2022/03/08/ges-worm-robot-sports-roach-style-whiskers-to-remove-fat-deposits-from-sewage-pipes/
[19] Danny FR, "Smart Motor Driver for Robotics | Hackaday.io," *Hackaday.io*, 2018. https://hackaday.io/project/158429-smart-motor-driver-for-robotics
[20] J.-E. Suh, T.-H. Kim, and J.-H. Han, "New Approach to Folding a Thin-Walled Yoshimura Patterned Cylinder," *J. Spacecr. Rockets*, vol. 58, no. 2, pp. 516–530, Mar. 2021, doi: 10.2514/1.A34784.
[21] "Western Filament Tuf Line XP - 150 Yards," *Cabela's*. https://www.cabelas.com/shop/en/western-filament-tuf-line-xp-150-yards?ds_e=GOOGLE&ds_c=Shop%7CCAB%7CTopPerformers%7CFishing&gclsrc=ds
[22] "PowerPro Product Specifications," *PowerPro*. https://www.powerpro.com/content/powerpro/northamerica/us/en/homepage/PDP.P-POWERPRO.html
[23] "4 in. Compliant Wheel 8 mm Bore 35A Durometer," *AndyMark*. https://www.andymark.com/products/4-in-compliant-wheel-8mm-bore-35a-durometer
[24] "Dragon Skin™ 10 NV Product Information," *Smooth-On, Inc.* https://www.smooth-on.com/products/dragon-skin-10-nv/
[25] "A3910EEETR-T Allegro MicroSystems | Integrated Circuits (ICs)," *DigiKey*. https://www.digikey.com/en/products/detail/allegro-microsystems/A3910EEETR-T/3973532
[26] "INA2180A1IDGKR Texas Instruments | Integrated Circuits (ICs)," *DigiKey*. https://www.digikey.com/en/products/detail/texas-instruments/INA2180A1IDGKR/8038162?utm_adgroup=Texas%20Instruments&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_Focus%20Suppliers&utm_term=&utm_content=Texas%20Instruments
[27] "Chihai Motor Chf-gm12-n20va Dc 6.0v Gear Motor N20 Mini Dc Gear Motor With Gearwheel 3mm Shaft Diameter," *Alibaba.com*. https://www.alibaba.com/product-detail/ChiHai-Motor-CHF-GM12-N20VA-DC_62238679682.html
[28] "RL1220S-R27-F Susumu | Resistors," *DigiKey*. https://www.digikey.com/en/products/detail/susumu/RL1220S-R27-F/714262
[29] "SS360PT Honeywell," *Mouser*. https://www.mouser.com/ProductDetail/Honeywell/SS360PT?qs=I6adikXrXuA6SFyXzjBzBQ%3D%3D
[30] "S5B-ZR-SM4A-TF(LF)(SN) JST Sales America Inc. | Connectors, Interconnects," *DigiKey*. https://www.digikey.com/en/products/detail/jst-sales-america-inc/S5B-ZR-SM4A-TF-LF-SN/926603
[31] "TC2030-NL no-legs to 6 pin IDC MCU debug cable," *Tag-Connect*.

https://www.tag-connect.com/product/tc2030-idc-nl
[32] “ATMEGA328-AU Microchip Technology | Integrated Circuits (ICs).” https://www.digikey.com/en/products/detail/ATMEGA328-AU/ATMEGA328-AU-ND/2271029?utm_campaign=buynow&utm_medium=aggregator&curr=usd&utm_source=octopart
[33] “Shanghai Mingda Microelectronics | Shanghai Mingda Microelectronics MD9927 | Motor Driver ICs,” *LCSC Electronics*. https://www.lcsc.com/product-detail/Motor-Driver-ICs_Shanghai-Mingda-Microelectronics-MD9927_C2684662.html
[34] “Diodes Incorporated | Diodes Incorporated ZXCT1010E5TA | Current-Sensing Amplifiers,” *LCSC Electronics*. https://www.lcsc.com/product-detail/span-style-background-color-ff0-Current-span-Sensing-Amplifiers_Diodes-Incorporated-ZXCT1010E5TA_C14151.html
[35] “RL1220S-R50-F Susumu | Resistors,” *DigiKey*. https://www.digikey.com/en/products/detail/susumu/RL1220S-R50-F/567251
[36] “ERA-3AEB122V Panasonic Electronic Components | Resistors,” *DigiKey*. https://www.digikey.com/en/products/detail/panasonic-electronic-components/ERA-3AEB122V/1465856
[37] Seon Rozenblum, “TinyPICO | A Mighty, Tiny ESP32 Development Board,” *TinyPICO*. https://www.tinypico.com
[38] “NCP1117DT33T5G onsemi | Integrated Circuits (ICs),” *DigiKey*. https://www.digikey.com/en/products/detail/onsemi/NCP1117DT33T5G/921285
[39] “Pololu 3.3V, 1A Step-Down Voltage Regulator D24V10F3,” *Pololu Robotics and Electronics*. https://www.pololu.com/product/2830
[40] Espressif Systems, “ESP-NOW - ESP32 - — ESP-IDF Programming Guide latest documentation,” *Espressif Systems*. https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html
[41] “Wire Library,” *Arduino References*. https://www.arduino.cc/en/reference/wire
[42] Y. Weiner, *CLARA-Gamepad*. 2022. Accessed: Mar. 16, 2022. [Online]. Available: https://github.com/phonymacoroni/CLARA-Gamepad
[43] “From Arduino to a Microcontroller on a Breadboard,” *Arduino Docs*, Feb. 04, 2022. https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoToBreadboard
[44] “Arduino as ISP and Arduino Bootloaders | Arduino Documentation,” *Arduino Docs*. https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoISP
[45] Hans, *MiniCore*. 2022. Accessed: Mar. 16, 2022. [Arduino]. Available: https://github.com/MCUdude/MiniCore
[46] Brad Miller, “Getting predictable motor performance | Using the Blue Motor,” Worcester Polytechnic Institute.
[47] “Pololu - 298:1 Micro Metal Gearmotor HP 6V with Extended Motor Shaft,” *Polo*. https://www.pololu.com/product/2218
[48] Douglas Brown, Wolfgang Christian, and Robert M. Hanson, “Tracker Video Analysis and Modeling Tool for Physics Education,” *Tracker | Video Analysis and Model Tool*. https://physlets.org/tracker/

Appendices

Appendix A: Final PCB Schematics

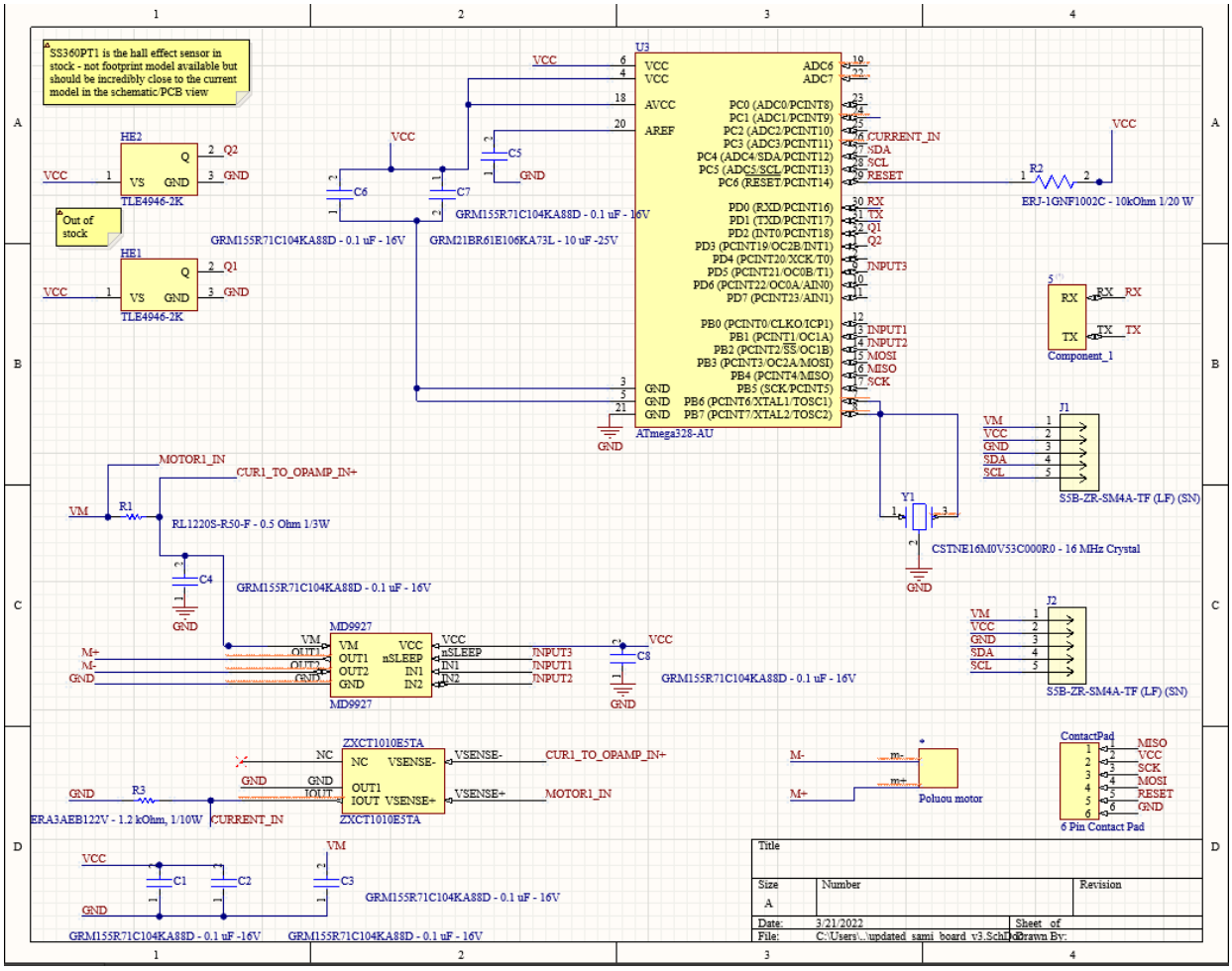


Figure A.1: Smart Motor Driver PCB Schematic

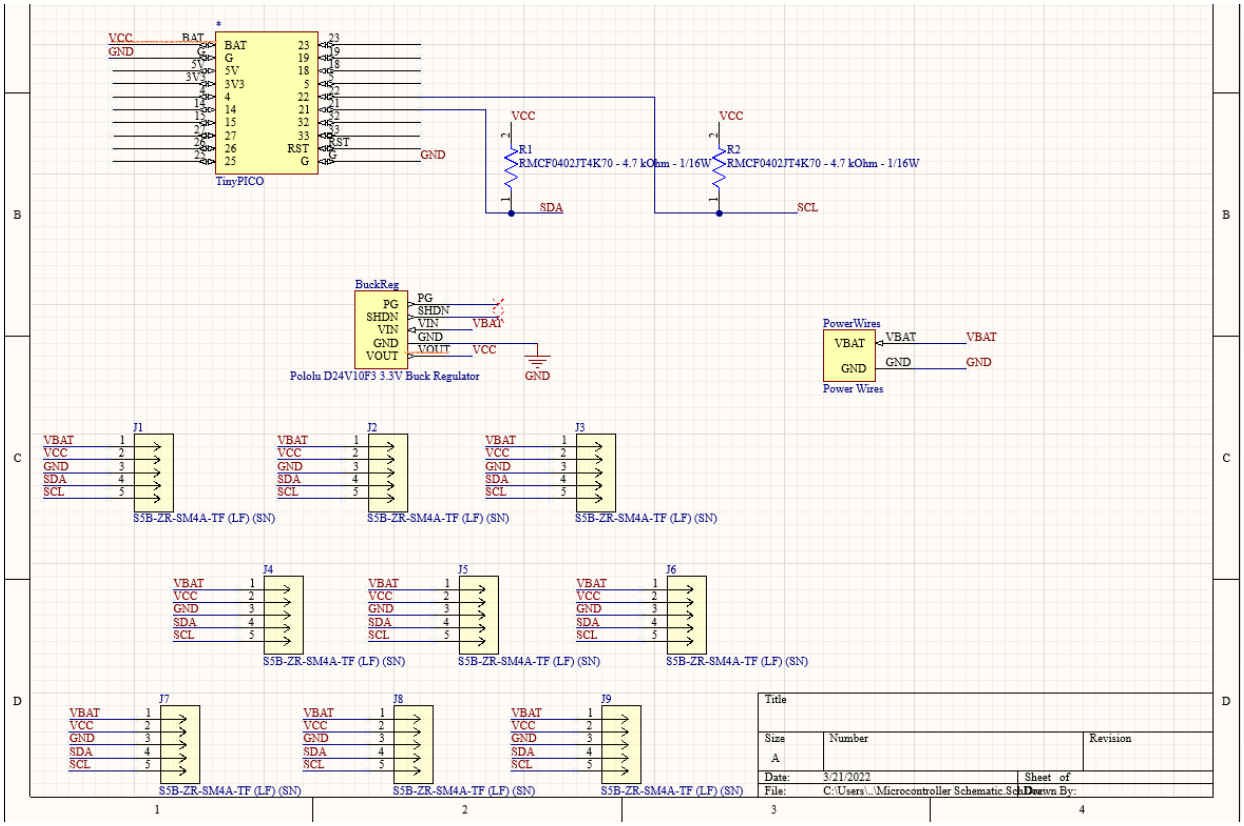


Figure A.2: Mainboard Schematic

Appendix B: Program Operation

The open-source GitHub repository including a README for program operation can be found here: https://github.com/BrianKatz925/C.L.A.R.A.-MQP/.