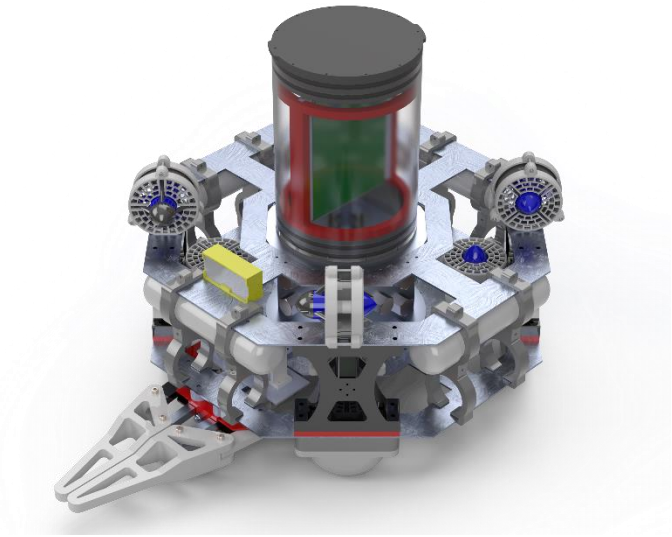


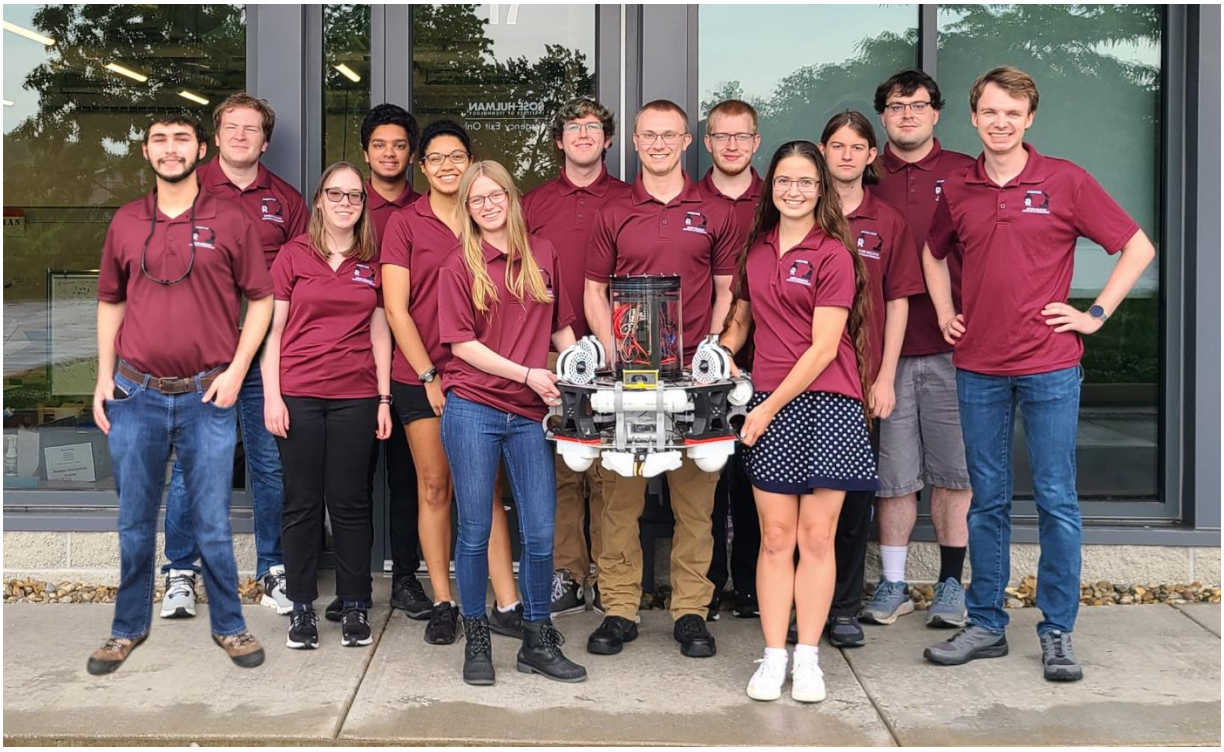
Duck

Rose Robotics

Rose-Hulman Institute of Technology
Terre Haute, IN

Team Members

'22 Timothy Beuchel, CEO, CFO, Firmware Engineer
'22 Danielle Villa, CTO Software and Firmware Systems
'22 Charlie Hahm, CTO Mechanical Systems, Safety Officer
'24 Nathan Quets, CTO Electrical Systems
'24 Tori Robinson, Firmware lead
'24 Amara Green, Mechanical Engineer, Pilot
'24 Tristan Scheiner, Software Engineer
'24 Rishi Singh, Mechanical Engineer
'24 Courtney Pepper, Mechanical Engineer
'25 Taytum Newell, Mechanical Engineer, 3d Printer Officer
'24 Gus Larson, Physics Subject Matter Expert
'24 Ross Hewitt, Electrical Engineer
'22 Robert Hairston, Firmware Engineer
'24 Jacob Branniger, Mechanical Engineer
'25 Laura Smith, Mechanical Engineer
'25 Kiana Fan, Software Engineer
'24 Quinlan White, Software Engineer
Advisors: Dr. Miles Canino and Dr. Eric Constans
No current members have attended an Explorer class MATE competition, however a team from Rose-Hulman attended in 2018.



Rose Robotics team members

Abstract

Duck is Rose Robotics' latest Remotely Operated Vehicle and is designed to operate at a depth of up to 5 meters. *Duck* has the ability to remove debris, plant and prune seagrass, and perform a variety of other tasks.

Duck was developed with about 2500 hours of virtual and in-person engineering, including design, fabrication, testing, and redesign. We are particularly proud of our product because we have had a variety of disruptions to our operations, including unexpected loss of experienced members, poor leadership, and of course the pandemic. This is the first time in four years we have had a competitive ROV to present at the competition.

Table of Contents

Safety

Company Safety Philosophy

Safety is very important to us at Rose Robotics. As such, we have a number of safety protocols that our employees must follow. There are also safety protocols which our school has put in place that the company must follow, in addition to those published by MATE. Due to the COVID-19 risk, we wore masks and social distanced until it was deemed safe to return to normal by our local government and school administration.

Lab Protocols & Training

The company had stringent safety measures when working with the ROV, especially during the manufacturing processes. While in the main work bay, all members were required to wear proper PPE to prevent any accidental injuries and a medical kit was available in the event that someone became injured while working with the robot. Any employee with long hair also had to have their hair tied up when working with machinery or with the ROV, both when dry and while in the water. When working with heavy machinery, a buddy system was implemented so that no one person was working alone with high power equipment.

Peer-to-peer training was used to train new employees on the operation and development of the ROV. New employees were also required to attend additional safety training through our local innovation center. Any employee wishing to use heavy machinery also had to attend specialized training before being allowed to operate the machines.

Vehicle Safety Features

Duck has a number of features designed to keep itself, its operators, and its environment safe. Every motor is fitted with thrust covers to prevent debris from entering them, and the ROV has rounded grippers and feet to prevent damage to its surroundings. Additionally, all screws were covered in heat-shrink and all water-jet cut edges were sanded down to prevent injuries when handling the ROV. Strain relief secures the connections in both the control station and the ROV itself, and a clear canister

allows for easy inspection of all electrical components.

Additionally, O-rings, welding, and epoxy were waterproofing techniques used to ensure that electronics remained dry. A safety fuse is used to disconnect power from the ROV in the event of overcurrent to prevent electrical and fire hazards. If the ROV begins operating outside of safe operating ranges, a safety switch is located by the control station that can immediately cut all power to the ROV. If power is cut from the ROV or the motors stop working for other reasons, *Duck* will rise to the top of the water for easy retrieval.

Operational and Safety Checklists

Safety protocols are documented in the Operational and Safety Checklists in the appendix. Employees also adhere to these protocols for launch, recovery, and waterside safety. Employees consist of the pilot, one or more deck managers, and the remaining employees are deck crew. The deck manager(s) may give instructions and ensure that steps are completed correctly.

Design Rationale

Design Evolution

Duck is Rose Robotics second generation of our most recent ROV system, based on our 2018-2021 ROV, *Kevin*. Most of the mechanical design was reused while the electrical design was completely redone, and the software was heavily refactored. This allowed us to focus our efforts more heavily on the electrical and software design.

The previous ROV design made use of a smaller frame and cannister to save weight and material, but *Duck* switched to a larger frame and cannister for ease of assembly and access during manufacturing and operation. This allowed the company easy access to all the electronics to aid in the maintenance of strong electrical connections.

Mechanical Design and Manufacturing

Duck was modeled using SOLIDWORKS as the primary modeling software and GrabCAD as the file sharing interface for the mechanical design team. The company decided that to maximize strength and minimize weight that any large metal component would be made from aluminum. The added bonus was that aluminum is non-corroding and is very easy to machine, making the manufacturing process quicker. Any metal parts such as hardware were stainless steel or aluminum whenever possible to prevent rusting of critical components.

A big push for the company this year was to transition to 3D-printed parts for larger volume parts to save materiel costs as well as production time. This led to the supports, weight clamps, feet, any spacers, and

most components of the gripper being printed from either PLA or PETG. An added benefit to this change was an increase in buoyancy since the volume displace by the parts was the same, but the weight was less than if they had been machined from solid material.

Mechanical Components

Frame

Duck was designed using the same frame concept as the previous ROV, *Kevin*, with the only difference that the frame of *Duck* was scaled up from *Kevin* to increase stability and ease of manufacturing. Figure 1 shows the SOLIDWORKS model generated during the design process.

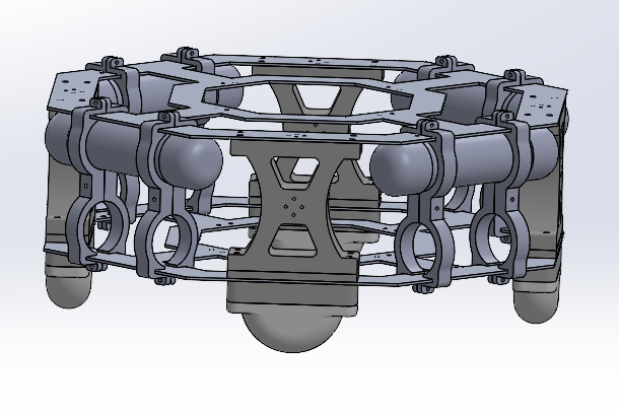


Figure 1: SOLIDWORKS screen capture of the frame of the ROV

The supports are 6.5 inches tall to allow for the cannister (Figure 2**Error! Reference source not found.**) to be mounted securely between the layers of the frame. The tubing along the sides of the ROV are filled with sand and allowed the company to adjust the overall mass of the ROV so that the positive buoyancy and balance could be adjusted easily. Space was allotted to up to eight tubes, but over the course of testing, it was learned that only four of the tubes were needed for the ROV to behave in the way the company wanted. Additionally, the feet were mounted to allow approximately 3.5 inches of clearance so that the gripper could be mounted on the underside of the ROV.

Electronics Housing

Figure 2**Error! Reference source not found.** shows the SOLIDWORKS model of the electronics cannister. Made from eight-inch diameter acrylic tubing attached to a six-inch-tall portion of six-inch square aluminum tubing, the electronics cannister supported a total of twenty bulkhead penetrators and allowed for all heat sensitive components such as the power board and ESCs to be heat-sunk directly to the walls or lid of the cannister.

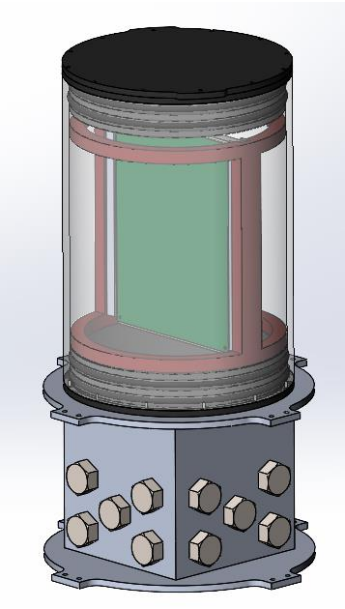


Figure 2: Electronics cannister assembly

The acrylic tubing, double O-ring flanges, and lid were all purchased through BlueRobotics and the full height of the tubing was used for the cylindrical part of the cannister. The aluminum connecting discs were used to attach the cannister to the frame via eight bolt holes on each disc, two on each tab. The upper disc was ¼ inches thick while the lower disc was ½ inches thick. The reason for this difference in thickness was because when the discs were welded to the square tubing, there were issues with getting a waterproof weld on the lower disc. This was because the upper weld has a large cutout to allow wiring to pass up to the upper portion of the cannister (Figure 3), making it much easier to reach the proper welding temperature. The lower disc did not have this cutout, and this resulted in difficulty welding and significant warping of this disc. By doubling the thickness, we were able to reduce the warping on the disc, making it easier to get a watertight weld.

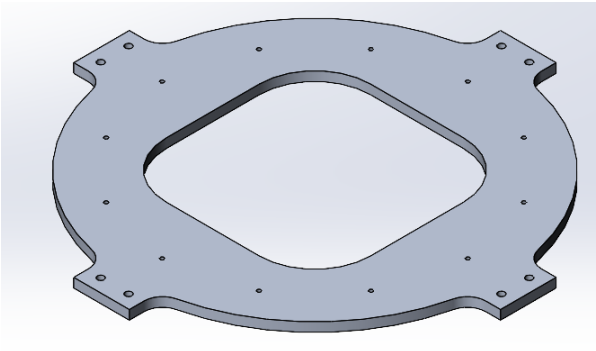


Figure 3: Upper disc SOLIDWORKS model

Thrusters

Duck used a total of eight T200 thrusters, four vertical thrusters and four horizontal thrusters as seen in Figure 4.

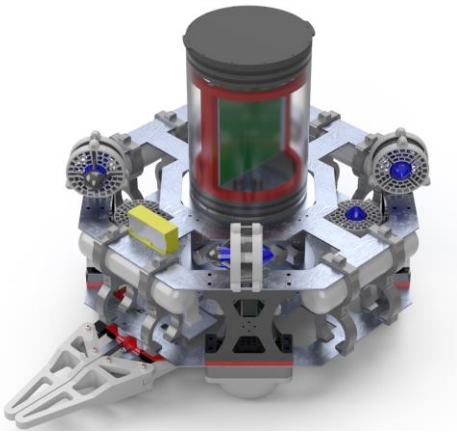


Figure 4: Rendered image of the full robot assembly from SOLIDWORKS

By placing the horizontal motors on the diagonals, it allowed us to use all four motors for forwards, backwards, and strafing movements by combining

the thrusts of the motors to achieve the desired movement.

Gripper

The company opted to use an electrically powered stepper motor to operate the gripper on *Duck*. The gripper is comprised of two four-bar linkages driven by a threaded rod connected to the stepper motor via a shaft collar. Figure 5 shows the two four-bar linkages outlined in purple with the red bar acting as the ground link.

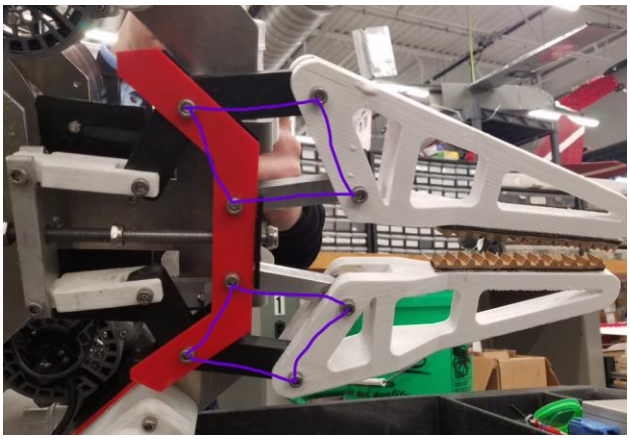


Figure 5: Robot gripper with four-bar linkages outlined

As the aluminum link at the rear of the gripper moves towards the front of the ROV, the claws on the gripper open and vice versa when the link moves backwards. This linkage allows the gripper to fully open and fully close, allowing it to grab a wide variety of objects. The stepper motor powering the assembly is capable of producing up to 294 oz-in of torque, providing enough grip strength to lift any object encountered by the ROV.

To waterproof the stepper motor, the motor was disassembled, and silicone was used to fill all the spaces inside the motor where we did not want epoxy to go such as the place for the magnet to go. Figure 6 shows the silicone filling the motor before the coils were potted.

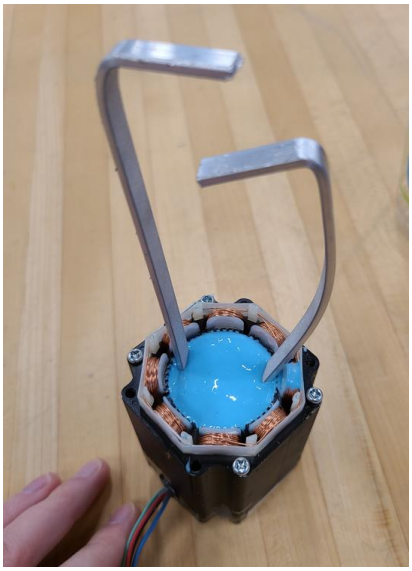


Figure 6: Silicone mold filling the motor voids

After the silicone had set, epoxy was poured over the coils using a syringe until the remaining spaces were filled with epoxy. It was then left to cure for 48 hours. After the epoxy had cured, the silicone was removed, allowing the motor's magnet to be reinserted. The resulting product was a motor whose coils were completely covered in epoxy, preventing water from coming into contact with the coils. Additionally, the company tested the resistance between the leads while the motor was submerged to

verify the coil insulation was intact before powering the motor.

Buoyancy

Duck is a positively buoyant ROV. The company made the decision to err on the side of positive buoyancy because the last generation, Kevin, was negatively buoyant. This was problematic when power was lost during testing since the ROV would sink to the bottom of the pool and was difficult to retrieve. Due to this, it was decided that Duck would be slightly positively buoyant so that the ROV would come to the surface if power is lost.

Table 1 is a buoyancy estimate of the ROV on a component by component basis.

Table 1: Buoyancy Calculations by component

Component	Mass (kg)	Volume Displaced (m^3)	Buoyant Force (N)
Cannister	7.91479496	0.019243688	111.1364428
Supports (x4)	3.208	0.00256	-6.3537408
Thin Spacers (x4)	0.361	0.000288	-0.71479584
Thick Spacers (x4)	0.7372	0.00076	0.223668
Feet (x4)	1.368	0.001092	-2.71026756
Gripper Claws (x2)	0.509	0.000406	-1.00766358
L link (x2)	0.115	0.000092	-0.22833756
Internal Link (x2)	0.013	0.000042	0.282954735
Straight Link (x2)	0.092	0.000034	-0.56898
Assorted 3D-printed Gripper Parts	0.119	0.000095	-0.23578335
Threaded Link	0.074	0.000028	-0.45387927
Threaded Rod	0.069	0.000026	-0.42183
Stepper Motor	1.361	0.000232	-11.07549
Thruster Covers (x16)	0.642	0.000512	-1.27074816
T200 Thrusters (x8)	2.752	0.001504	-12.24288
Upper Frame	1.244	0.000461	-7.68123
Lower Frame	1.244	0.000461	-7.68123
Top Camera	0.454	0.000123	-3.24711
Bottom Camera	0.454	0.00022	-2.29554
PVC Weights (x4)	5.444	0.003132	-22.68072
Weight Clamps (x16)	1.263024	0.001008	-2.50178544
Total Mass (kg):	29.438	**Total Buoyant Force (N)**	28.271054

This table shows both the mass and buoyancy of *Duck* but is an estimate because the tether is not included in the mass and

because the masses of 3D-printed parts were estimated based on volume since they were not weighed after they were printed.

Table 1 shows that the mass of the ROV is 29.438 kg which is far enough below the weight limit that we do not anticipate the tether pushing us beyond the weight limit. It also shows that the ROV is positively buoyant with a buoyant force of 28.27 N or 6.36 lbf. Based on our observation of the ROV this seems reasonable because it would indicate that the ROV should be mostly submerged when resting in the water which is what was observed during testing.

Software and Firmware Systems

System Overview

The control system of *Duck* is comprised of two major subsystems, the above water system and the on-board system on the ROV. The above water system consists of the control station and the joystick used to control the ROV. The on-board system consists of a Raspberry Pi, a microcontroller, the motor controllers, the motors, and the cameras. See Figure 7 for a more detailed context diagram for *Duck's* control system. Words on arrows specify the communication method; arrows without words do not require any defined communication protocol.

In order to have as much control over the system as possible, we avoided using external software systems unless the company lacked the expertise to write it from scratch. Specific cases of such are documented below. Performance and responsiveness were of particular importance to the company when designing the software and firmware for *Duck*. Writing almost everything in-house allowed us to prioritize these and design alternatives that we could switch to if necessary.

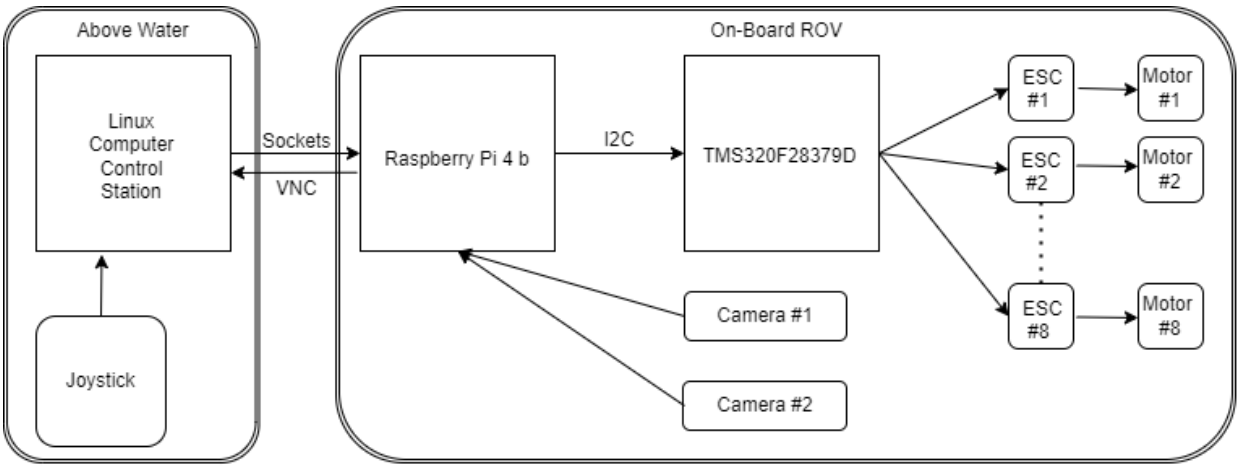


Figure 7: System Context Diagram

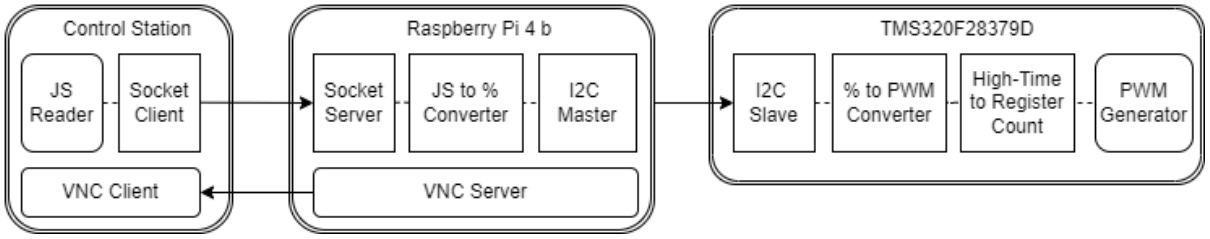


Figure 8: System Architecture Diagram

Due to time and labor supply limitations, we were unable to add additional sensors to *Duck*. We did design a service-oriented architecture that was more modular and expandable and would have been used if additional sensors could have been added, but since they could not all communication was able to be broken down into two pipelines flowing in opposite directions. A VNC network was setup to handle viewing through the cameras, and as such flows from the ROV to the control station. Controlling the ROV flows from the control station to the motors, and as such our system for that has a pipeline architecture. This is shown in Figure 8. Additional details on individual modules are described below.

Control Station

The control station runs on a computer running Ubuntu 20.04. We use on open source joystick reader, jstest-gtk by Ingo Ruhnke, because the company lacked the time and expertise to write and design it from scratch. The driving commands from the joystick is read by that module, which then sends it to the Raspberry Pi on *Duck* through a communication module. That module was written to use sockets for the inter-process communication. As mentioned above, the control station runs a VNC client to view the camera feeds.

Our piloting interface is very simple. It consists of video feeds from *Duck* and command line interfaces that are used to start up various modules and debug motor thrusts when necessary. All pilot inputs are done from the joystick, except switching between cameras which is done through the computer directly.

ROV Software

The software that runs on-board *Duck* runs on a Raspberry Pi 4 b running the Raspberry OS. This was chosen for its versatility and the strong debugging support of its community. A socket server receives the driving commands, which are then given to a separate module. This module converts these commands to the thrust percentage for each motor. Since the Raspberry Pi does not have a direct connection to the motor controllers, it sends the thrust percent for each motor to the microcontroller running the

ROV's firmware. This is done over an I2C protocol written in-house. The cameras are directly connected to the Raspberry Pi, which uses a VNC server to allow the control station to view their feeds.

ROV Firmware

The firmware runs on a TMS320F28379D microcontroller. We chose this microcontroller because of its speed and abundance of hardware PWM generators. The microcontroller receives the calculated thrust percentages through the I2C communication line and then converts them to PWM high-times through a lookup table. The next module converts these to a register count that can be used by the hardware PWM generators. These are then sent directly to the electronic speed controllers (ESCs), which are connected to the motors.

Electrical System

Printed Circuit Boards

Though our final electrical system primarily uses off-the-shelf modules and breakout boards, we spent most of the year developing fully hand-soldered PCBs to house all essential components. These boards proved too difficult to successfully design, build, and test within the year, but they did serve to inform the ultimate design of the robot for use and served as a worthwhile endeavor throughout the year.

The main control board was a 4-layer PCB designed in-house and manufactured by JLCPCB and housed the bulk of the electronic systems used by the ROV. It contained headers for programming the microcontroller, 3.3V and 1.8V linear regulators for powering the microcontroller and other digital systems, 5V and 24V regulators for powering the gripper motor and Raspberry Pi, temperature and humidity sensors, a gyroscope and accelerometer, connectors and headers for external sensors and motor signals, and a header for connecting to the Raspberry Pi. Aside from the Pi, all systems were implemented using discrete components and ICs, using a combination of soldering iron and hot air techniques that unfortunately proved unreliable for fine-pitch surface mount ICs. The inherent dangers of hand soldering SMT parts were a large contributor towards the ultimate abandonment of the control board, as many systems could not be quickly and effectively fixed or analyzed due to the wide variety of factors that could lead to their failure; in particular, the microcontroller and regulators suffered from many electrical faults due to the difficulty of soldering their ICs by hand. Ultimately, we stepped away from the custom PCBs to implement the same system using pre-built modules, sacrificing system features and form factor for reliability. The design lessons

learned from this board, both in design and manufacturing, will be brought forward in future years to improve the function of our custom PCBs.

The main power regulation board was a 4-layer PCB that served as a platform for Analog Devices' LTC7871 6-phase switching regulator controller, containing the controller IC, filtering capacitors, switching MOSFETs and inductors, a header for the digital inputs and outputs of the controller, and all passive hardware required to support the ICs and switching channels. In addition to basic hardware, special attention was required for the layout of the board, since the power being regulated was of large potential magnitude and the controller required sensitive control and feedback routing to function appropriately. Combined with the heat and power dissipation requirements, this resulted in a substantial degree of challenge in laying out the PCB. Ultimately, though circuit simulations of the schematic revealed proper function of the board, there were numerous issues in layout and parts selection that resulted in a need for a more reliable and tested backup. Therefore, the power board we ultimately used had most of the complex heat and magnetics management done through modules soldered to a 2-layer PCB with some components for programming, filtering, and protection. The prior design will be revisited for future robots, however, so the efforts made over the course of the year were worth it for future designs.

Lastly, in order to send power out from the regulator to the motors (with protection for each), we created a small 2-layer PCB that contained a power input from the regulator, power outputs for each motor, and fuses for each motor in case of a fault. This board is currently in use, as it is very simple and basically serves as a bus.

System Modules

We retained the Raspberry Pi as our main interface from the surface, connecting through ethernet to the computer so that we could control the internals through the Pi rather than having to send down more complex communications to the robot from the surface. In lieu of our discretely implemented microcontroller, we used a TI Launchpad breakout board as our microcontroller platform, with jumper wires connecting it to the Raspberry Pi, ESCs, and stepper driver module. The ESCs are controlled using PWM signals from the Launchpad, and the stepper module is controlled using a STEP/DIR interface with an enable input, allowing us to turn the motor off when not in use and easily use basic GPIO to drive the stepper motor.

Tether

Our robot's tether consists of an ethernet cable and power cable (with 48V and return from the power supply). The power cable is in-line fused for 25A, within 30cm of the power supply, and both ends of the tether are relieved of strain using ropes and carabiners. The ethernet cable connects the surface laptop to the Raspberry Pi, which enables the driver to interface directly with the robot from the surface using ssh and VNC Viewer to remotely access the Pi. The tether is wrapped in braided cable tubing for wire management and connects to the robot using 8-pin and 5-pin Seacon bulkhead connectors.

Build vs. Buy

For the final design, most of the critical electronic systems were off-the-shelf modules, a decision motivated by a lack of time to acceptably test and repair our custom PCBs for the main control systems. Due to the difficulty of hand soldering SMT parts, troubleshooting and redesign efforts were consuming too much of our time, and we eventually decided that the only way to get satisfactory function in a reasonable time frame was to replace our main control board with modules that served the same function. Thus, we implemented the core parts of the system using a 5V regulator module, a TI Launchpad microcontroller breakout board, and a Pololu stepper motor driver connected by jumper wires, powering the microcontroller off one of the Pi's USB ports. Though we would have preferred to implement these systems on our custom PCBs, time constraints and unclear failure causes led us to use more off-the-shelf modules than previously envisioned in our final product.

As mentioned in the ROV Software section of this document, while we built the majority of the software and firmware used, we did use some externally-developed software. JSTest-GTK is an open-source, GPL 3.0, joystick reader developed by Ingo Ruhnke that we used because we were unable to develop a joystick reader in-house. We also used VNC Connect in order to stream our camera feeds because we were unable to find an alternative. Using these products allowed us to focus on writing our communication protocols; we wanted to minimize communication lag and focus on skill development, which writing our own software and firmware allowed us to do.

Testing and Troubleshooting

Water Testing

Initial thrust vectors were written based on ideal motor behavior and response. However, they had to be tuned manually to account for individual motor performance, hydrodynamic effects, and discrepancies in

the ROV's center-of-mass. The ROV was taken to the pool, driven, and its response to directional inputs and general behavior were noted. Upon noticing an undesirable response or persistent error, the vectors were tweaked poolside by the pilot. We paid particular attention to levelness, straightness of path, and any persistent yaw. Initial transient response was prioritized over steady-state response to make fine control more consistent. Swimmers in the pool helped the testing process by monitoring motor performance and provided a second angle to observe the ROV from.

The vertical rise or sink was tuned first to ensure a level ROV, followed by the horizontal directions and yaw. Tuning continued until the ROV was deemed to be consistently and easily controllable by the pilot. Tuning was performed again in the event of a change to the thrusters or ROV weight distribution. Once the vectors were tuned to satisfaction, water testing continued in order to practice tasks and gain general pilot experience.

Mechanical Testing

Mechanical testing was conducted primarily on two systems, the electronics cannister and the gripper. To test the electronics cannister, it was submerged in 2 feet of water in a tub for 24 hours to verify that there were no leaks at low pressures. Once it successfully completed that testing, it was then attached to the ROV and taken to the pool where it was then weighed down to the bottom of the deep end of the pool and left for fifteen minutes. When no leaks were observed, the cannister was considered waterproof.

To test the gripper, after the motor was waterproofed, it was assembled on the ROV along with the gripper and it was verified that the gripper moved on dry land. After that was confirmed, the ROV was taken to the pool and the gripper was used to manipulate props.

Electrical Testing

Electrical testing was done in several stages. First, a basic continuity check was applied to verify the circuit was free of shorts. Then the circuit was powered with a variable power supply with a low current limit, and the functions of the circuit were tested. Circuits were also built in stages to avoid malfunctions in one stage damaging another and to make it easier to debug.

Software & Firmware Testing

The software was tested through a combination of manual exploratory testing and automated testing. The mathematical modules, the driving control to thrust percentage and percentage to PWM modules mentioned in the ROV Software section, were tested with a suite of automated tests

following boundary value testing principles. Other modules were thoroughly tested manually, both by inspection of console output and measurement of motor thrust when connected to the ROV.

The firmware was also tested with manual exploratory tests because we were able to fit the microcontroller we used with a test harness. Tests were evaluated by measuring motor thrust and gripper movement when connected to the ROV, after observing the signals with an oscilloscope.

Logistics

Company Organization

Rose Robotics is organized into three departments, mechanical, electrical, and software. Firmware was folded into the software department because there were not enough firmware engineers to justify the overhead of a separate department. Each department had a department leader (CTO) that reported to the CEO, and the members of the department reported to their department head.

Project Management

The company applied a hierarchal project management strategy. Each CTO was responsible for assigning and tracking individual tasks, while the CEO was responsible for managing larger projects and the overall design and fabrication project. Before the school year started, the CEO met with all the CTOs and developed a rough schedule for larger tasks. We attempted to follow this schedule, but quite badly misestimated the human resources we would have available and the time it would take to complete tasks, so we fell behind schedule very quickly. Fortunately, we had built in a few months of time as pilot practice and were able to dip into that time to make up for the delays.

We used GitHub to manage our code and electrical schematic files and GrabCad to manage our mechanical CAD files.

Budget

Engineering Components					
Category	Budgeted	Expended		Income	
Cannister	500	430.36		BIC	5600
Electronics	3000	2268.08		SGA	12485.72
Thrusters	2000	1648		Total	18085.72
Mechanical	2000	2137.98			
Cart	350	325.68			

Registration fee	400	400			
Total	**8250**	**7210.1**			
Travel					
Category	Budgeted	Expended			
Gas	1009.5	Unknown			
Flights	4377.1	4522			
Parking	216	216			
Hotels	3726.1	Unknown			
Rental car	700	900			
Total	**10028.7**				

Conclusion

Challenges

We were able to recruit a large amount of people this year, almost doubling the company in size. Due to this large increase in numbers, we created a very tight schedule where lots of things would be done in parallel. However, we failed to consider the amount of time spent training new employees. This limited the amount of time that experienced employees could spend designing and building the ROV, and new employees did not have all of the skills to do all of the tasks. This led to us having to adjust to a more conservative schedule and not being able to accomplish everything we had planned.

Like many teams, we also suffered from supply chain issues. This was not something we could control, but we did have to do more planning since we could not get supplies or parts manufactured on short notice. While we hope these issues get resolved, the additional planning it forced on us is something we are hoping to continue into the future.

Lessons Learned & Skills Gained

In previous years, Rose Robotics has been a very small company. Our growth this year allowed us to do more and eventually compete, but it added many challenges. Delegation became extremely important; one person cannot track everything and as such a more structured leadership approach was adopted. Because of the success of this model, we will be continuing to use it next year.

We also tried a new method of building our canister; we welded our canister together. This led to our waterproofing and water testing phase taking significantly longer than anticipated. We learned that welding was probably not the best technique for us to have used, but we gained many skills in other waterproofing methods in order to compensate for the welding's drawbacks.

Future Improvements

Duck's dual camera system improves on the old single camera system by allowing us a more detailed gripper view and the epoxy potted cameras reduce the bulk required on the ROV. We plan on continuing to improve on this system by adding servo motors to each camera to allow 4 degrees of freedom to one or both cameras. This would give us better range of vision and would help with tasks that would benefit from unique camera angles. We will also be investigating alternatives to decrease the lag on the cameras for next year.

Due to time and labor constraints, the ROV this year did not have any sensors on it, so the ROV is completely controlled by the pilot. We would like to add pressure sensors to be able to automatically control for depth and leak sensors for automatic leak detection.

We would also like to redesign all of the electrical boards to be smaller and more modular. Our design this year worked well but further iterating on it to prioritize flexibility and organization will help improve the next generations of ROVs.

Acknowledgements

MATE Center and Marine Technology Society – Sponsoring this year's competition

National Science Foundation – Funding of the MATE competition

Long Beach City College – Hosting the 2022 MATE competition

Lee Dagle – Her time, guidance, and yearlong support of the company

Larry Waters – His support preparing for & during travel

Branam & Kremer Innovation Center staff – Use of their tools & machines, and their support & advice

Rose-Hulman Student Government Association – Funding and administrative support

Rose-Hulman Sports & Recreation Center – Generous allotment of pool time

Rose-Hulman EIT – Providing software such as CAD software

Our families – Their continued support and encouragement

Our advisors Dr. Miles Canino and Dr. Eric Constans – Their time

References

Blue Robotics. "T200 Thruster: Technical Documentation." Blue Robotics. Blue Robotics, 2022. Web. 10 November 2021.

MATE. "MATE ROV Competition Manual Explorer." MATE. MATE, 22 January 2022. Web. 24 May 2022.

Raspberry Pi. "Raspberry Pi OS." Raspberry Pi. Raspberry Pi, 2020. Web. 4 September 2022.

RealVNC. "VNC Connect." RealVNC. RealVNC, 2002. Web. 15 March 2021.

Ruhnke, Ingo. jstest-gtk (v. 0.1.1). GitHub, 2 June 2018. Web. 13 September 2020

Appendix

Safety and Operational Procedures

Pre-Launch Procedure:

1. Connect the tether to the ROV. Add tether-ROV strain relief.
2. Connect the tether to the surface power and to the control station. Add tether-control station strain relief. Connect the joystick to the control station.
3. Verify that the electronics canister is fully sealed.
4. Turn on surface power.
5. Wait until ESCs have initialized, signaled by loud beeping. Connect to the internal ROV controls and start the camera server and the control server.
6. Ensure that the joystick is properly connected to the control station. Turn on the camera client and the control client in the control station.
7. Pilot will do a brief motor test to ensure communications are working.
8. Pilot will inform the deck crew that communication is established, and they are ready to launch.
9. Begin *Duck* launch procedure.

Duck Launch Procedure:

1. Ensure that *Duck* is as close to the water as possible while remaining stable on dry ground. All but 2 deck crew members clear the area.
2. Check that there are no obstructions in the water.

3. 2 deck crew members simultaneous lift *Duck* and place it in the water. They then clear the area and confirm the launch with the pilot.
4. The pilot begins the mission.

Duck Retrieval Procedure:

1. Ensure that either power is turned off from the control station or all motors are in their idle position.
2. Wait for *Duck* to float to the surface.
3. If *Duck* is not at the edge of the water, then bring it to the edge. If it is safe to do so, a diver enters the water and swims the ROV to shore. If it not safe to do so, the deck crew gently pulls the ROV via its tether.
4. Two deck crew members grab the ROV, ensure that no motors are turned on, and that the surrounding area is clear.
5. Simultaneously, the two lifters lift *Duck* out of the water and set it on the ground.
6. Remaining deck crew removes the remaining tether from the water.
7. If power is still on, turn power off.

Loss of Communication Protocol:

1. Restart the socket connections on the control station. If communication is restored, continue the mission.
2. Cycle the power from the control station and then restart the communications from the control station. If communication is restored, continue the mission.
3. Turn off power from the control station.
4. Begin *Duck* retrieval procedure.

Loss of Power Protocol:

1. Cycle the power from the control station and then restart the communications from the control station. If communication is restored, continue the mission.
2. Turn off power from the control station.
3. Begin *Duck* retrieval procedure.

Electric SID