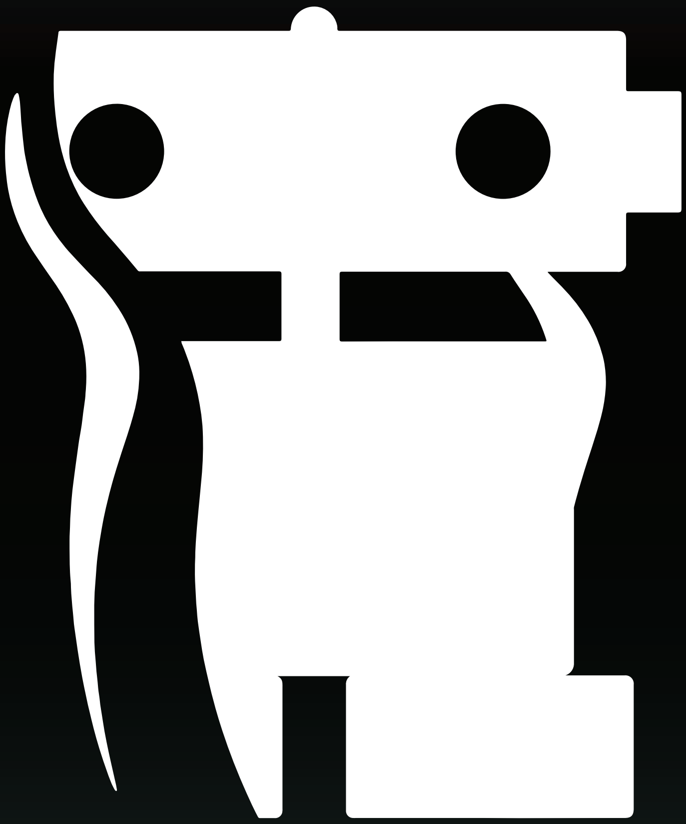
kelpie robotics



```
> echo $SCHOOL
UNIVERSITY_OF_OTTAWA

> echo $LOCATION
\Canada\Ontario\Ottawa

> echo $CATEGORY
EXPLORER

> teamfetch

TEAM_NAME:
        KELPIE ROBOTICS

OFFICERS:
        JUAN HIEDRA PRIMERA: CEO
        STEFAN TODOROVIC: CFO
        SOPHIE TOMLIN: COO
        RIKKI ROMANA: SAFETY_OFFICER
        JASON GONZALEZ: CTO_MECHANICAL
        SEBASTIAN LARRIVEE: CTO_ELECTRICAL
        ETHAN BOWERING: CTO_OS

MECHANICAL:
        FREDERIC ELOY
        AMANDA DOKU

ELECTRICAL:
        MIHIR JAKHI
        MOHAMAD ALI JARKAS
        ETHAN GRAVES
        TAHMEED KHAN

SOFTWARE:
        KAITLYN DOMPAUL
        ALAE BOUFARRACHENE
        ERIC EATON
        MING YE CHEN
        ADAM DIA
        UMER QAMAR

MENTORS:
        JASON DEMERS
        HASSAAN ZAKI
```

INTRODUCTION

ABSTRACT

Kelpie Robotics presents KELP-II, a remotely operated vehicle (ROV) that can perform a wide variety of sustainability-oriented tasks. Kelpie Robotics, with its 19 members, is a group of dedicated individuals with a wealth of knowledge and technical experience capable of producing an adaptable, and quality product. With four sub-teams and Project Managers for complex tasks, Kelpie Robotics is able to take a multidisciplinary approach to design, while encouraging members to use their knowledge and expertise in all aspects of design, manufacturing, and management.

We share a deep passion for innovation at Kelpie Robotics, and a desire to see change for better. One of the most pressing issues we face today is concern for our environment, and within each of our team members is a drive to find solutions to problems faced. Supporting our team in our endeavour to create an ROV that is expertly suited to sustainability-oriented tasks is IEEE Canada Foundation, and one of their main goals is sustainability.

After months of research, design, development, and strict testing, KELP-II proudly features a design with increased maneuverability, modularity, and efficiency. Its improved systems include a modular ecosystem for onboard electronics, a custom PMBus-adherent power monitoring system, and a virtual reality (VR) compatible pilot interface and copilot graphical user interface pair. This document outlines the development process, technical specifications, and design rational that led to the development of KELP-II, in order to meet the United Nations' Requests for Proposals (RFPs).

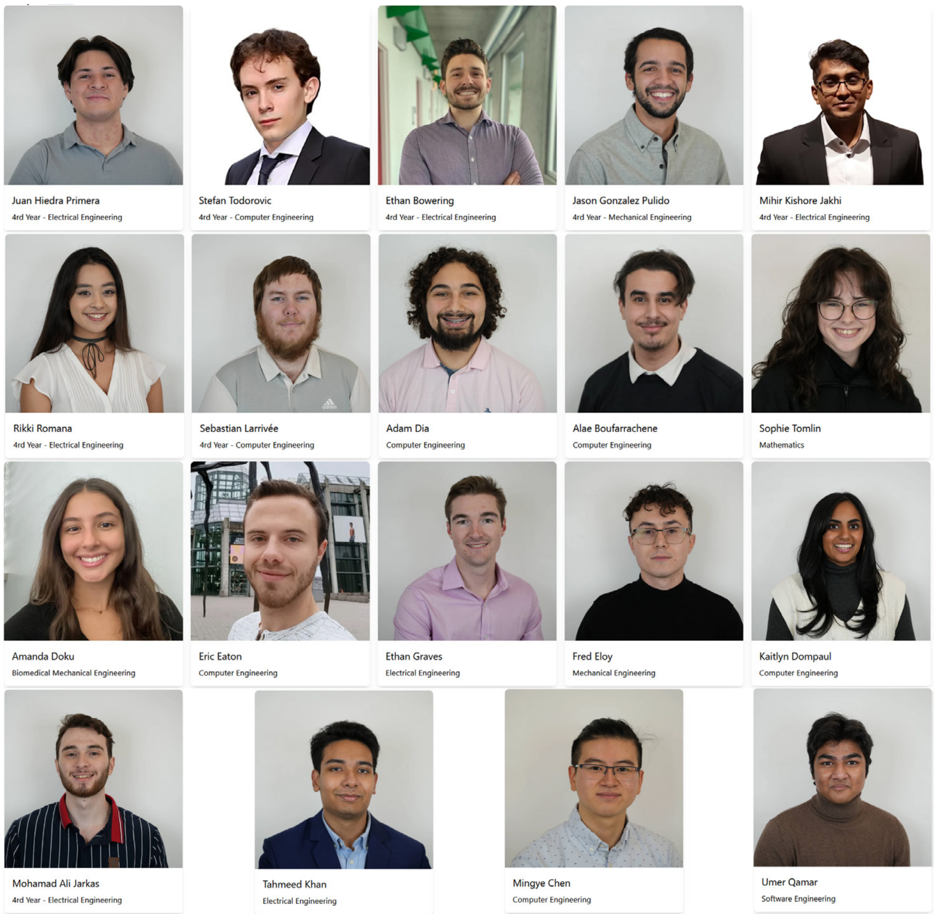


Figure 1 - Kelpie Robotics Team Members

Table of Contents

Lists and Figures

DESIGN RATIONALE

MECHANICAL COMPONENTS, MANUFACTURING

Frame

Design

The team began by testing out several designs, building off knowledge and lessons learned from last year, and running simulations in Fusion. Some potential designs included a design similar to our TIE-fighter design as well as a metal frame design (like was chosen this year).

Manufacturing

The frame is constructed from aluminium extrusion and taps, and then those pieces were connected using corner connectors. This design allows for a lightweight, durable frame, and for ease of movement through the water.

Electronics Enclosure

Design

We first began with a design for a Polycase box with several flanges. However, unfortunately, the team ran into some issues with this design. As this is where electronics would be housed, it is imperative that the enclosure be water-tight, but the sides were not sealing, so a cylindrical enclosure was used instead. Similar to the previous iteration, but wider to make more room for electrical components and to allow for ease of access. Detailed CAD drawings and specifications for each penetrator were analyzed to create an acceptable drilling template on the endcap model.
The template was drawn based on the dimensions and requirements of the penetrators, ensuring accurate placement and resolving any size and clearance constraints.

Manufacturing

A model of the polycarbonate outdoor box was utilized to determine the appropriate flange sizes and ensure sufficient clearance from the edges. Flange designs were machined out of aluminum and securely attached to the polycarbonate box using epoxy, allowing it to cure before conducting testing and evaluation of the enclosure concept.

Enclosure Stands

Once a metal frame design was scratched off the board, we pivoted to using a Blue Robotics 8” Enclosure. At this point, it was necessary to design mounts allow the cylindrical enclosure to be securely fastened to the frame. The stand was designed using Fusion360, and was 3D printed, however no printers available were large enough to print the stand in a single piece. It was then divided into 4 pieces, which were then joined once again with epoxy.

Thrusters and Shrouds

KELP-II features the same thrusters used in its previous iteration -- Blue Robotics T200 thrusters -- and the shrouds the same. However, they now needed to be fastened to a different frame, and as such, new mounts were designed. Using previous CAD models, new mounts were designed using Fusion360. The fact that the attachment method for extrusion the mounting points are a few centimetres away from the thruster to allow for tool access was necessary to consider while designing the mounts.

ELECTRICAL SYSTEMS

Topside Control Unit

The topside control unit was designed to be a central hub for communication and control of the ROV, including all software, networking, and hardware required for full functionality. The control station is effectively a computer built inside of a Pelican case.

Onboard Electronics

To begin the breakdown of the power boards, some background information on the circuits is needed. Our boards consist of 3 different circuits for power and power monitoring while including a circuit for the embedded micro-controller. First, we will focus on the power monitoring circuitry and will exclude the DC-DC converter design and embedded micro-controller design as these change may from project to project.

First, the theory behind the current sense circuit. The basic principle of the current sense circuit involves the use of Ohm's law;

$V = IR$

The idea is that given a resistor of known resistance, and the voltage drop across the resistor, the current through the resistor can be calculated.

Since the value of the resistor is typically very small to reduce the power dissipation and voltage drop, the voltage differential between the input and output of the resistor is typically also quite small. This differential is normally too small to be read properly by a micro-controller over the entire range of current, and thus must be amplified into a voltage that is used by the micro-controller. This means that a maximum voltage differential corresponding with the maximum current going through the circuit would have to be amplified to use the entire voltage range of the micro-controller. This is accomplished through the use of an instrumentation amplifier which takes an input differential and amplifies it to a higher or lower voltage value in respect to ground. This has 2 effects: other than just amplifying the voltage differential, it also puts the voltage differential in terms of ground and VCC which means the load and power sense circuitry can share a common ground. The design of such a circuit can be seen in Figure 2.

The various components of the KelpieSense circuitry can be observed. R1 is the shunt resistor that has a very small known value, r3-6 along with U1 form the instrumentation amplifier, R2 simulates a load close to the maximum theoretical value the system would have, and C1-3 are used for voltage ripple suppression and for decoupling.

Secondly, we have the voltage sense circuit or voltage measuring circuit. The voltage sense circuit also relies heavily on Ohm's law to convert a voltage too high for the micro-controller into a voltage safe for the micro-controller. This is done through the use of a voltage divider in which 2 resistors are carefully selected so that the voltage across the bottom resistor is the output value selected. This is typically used to convert a 5V signal into a 3.3V signal by dropping 2.7V across the first resistor and then dropping 3.3V across the second resistor. By reading the voltage in between the 2 resistors, the value would be 3.3V when the input value is 5V. In the case of the power boards, a voltage higher than 5V would normally be read so the resistors calculated must take that into consideration.

While the basic principle behind the voltage divider is sound, there are a few issues that arise from its operation. If the load at the output of the voltage divider is too big, it will lower the voltage dropper across the bottom resistor, and can also negatively affect the power source by drawing too much current through the circuit. To avoid this issue and to give the micro-controller a clean voltage to read, the output of the voltage divider is then used as an input to a voltage follower, which is a special op-amp circuit in which the input voltage and output voltage are the same.

By placing the output of the voltage divider into a voltage follower, there is a high impedance load on the voltage divider, and the voltage follower creates a low impedance load for which the micro-controller can use in its analog inputs. This is necessary as the data sheet to the given micro-controller typically specifies the maximum input impedance allowed for accurate ADC readings. The final circuit can be seen in Figure 3.

Design

One of the most important steps in any project is to actually design the circuit needed based on specifications given. In the case of the power boards, this specification was that the DC-DC converters used would be off-the-shelf parts, would contain PMBus, and would be rated to the currents needed in the ROV. While 2 of the 3 design specifications were met, an unfortunate parts shortage left the ROV without PMBus on the DC-DC converters. This is what led to the design of the power monitoring circuitry. Without PMBus there was no way of knowing what the output current of the module was, as well as what the input and output voltage was. This was the precursor into the researching and creation of current sense circuits and voltage monitoring circuits.

Figures 4-7 show the schematics of the boards with Figures 4-6 being specific portions of the circuit, and Figure 2.6 the entire circuit.

Figure 4 is the core of the board, and its most important part. It contains the murata power converter that does the 48V to 12V conversion, as well as the plans for an inline fuse, and shunt resistors for the current sensing. Figure 5 shows the circuit for a bare-bones STM32 micro-controller. It has the recommended circuits for reset and boot, has an external crystal oscillator, and contains all of the necessary connections to read the analog values for current and voltage. Figure 6 contains the op-amps used in the instrumentation amplifiers, and voltage followers. This is designed in such a way as to protect the micro-controller by bringing the voltage differential from the instrumentation amplifiers, and the voltage value from the voltage followers into the 0-3.3V range used by the STM32. This has to be designed as precisely as possible to allow for maximum range in the 0-3.3V range. The calculations for the resistors can be found in the appendix. The designs for the circuits were peer-reviewed to catch basic errors, and as a sanity check, to see if the circuit should work.

The overall design was built in such a way as to introduce as many safety features as possible. The boards were designed to be use in conjunction with fuses to prevent complete failure if a device goes short circuit, and the power modules selected contain various safety features like over current protection, short circuit protection, and over temperature protection. the idea to add in the voltage and current reading circuits was to allow for software control of load currents to keep them below maximum levels alongside data logging.

Error Analysis

As with most designs, issues start creeping up as soon as testing occurs. When the testing for the above circuits was underway, an issue was discovered in the way the schematic was laid out. There were 2 issues with the schematic. The first issue was that the op-amps for the instrumentation amplifiers were receiving a voltage much higher than the max tolerated on the input pins which was causing the input protection diodes to work more than they were designed to, and was also causing all sorts of electrical abnormalities on the output pins.

The second issue in the design involved how the high voltage and low voltage lines were connected. From the initial design, the high voltage lines (12-48V DC) were connected to the low voltage side through the ground, and that ground was connected on a separate board. This meant that if the PMBus cable was not connected to the power boards, the digital ground would

be floating, and would allow the 48V signal to make its way into the STM32, the Op-Amps, and the 12V side of the converter if the output of the converter was physically separated. When the PMBus cable was connected, the resistors for the instrumentation amplifier would give a path from 48V to ground, and since the resistors were of relatively low resistance in comparison to the voltage divider, it would dissipate a lot of energy and would get the board really hot.

Both of these issues were temporarily solved by removing the op-amps, and the input resistors effectively disconnecting the high voltage side from the low voltage circuitry. This was not the final solution though as PMbus functionality was very much beneficial to this application.

After some deliberation amongst the team, it was determined that the circuit would need to introduce more voltage followers to lower the input voltages from the shunt resistor to safe voltages for the micro-controller. This gives each instrumentation amplifier their own low impedance signals, and partially decouples the high voltage side from the low voltage side. The boards are shown in Figure 8 and Figure 9.

Going Forward

With the analysis of the issue, and a solution in the works, all that is left is to mod the existing boards with the necessary fixes, and to confirm that the software on the embedded STM32 micro-controllers communicate properly with the rest of the system. With this, the first revision boards should be fully functional. In the future, the boards should be redesigned with the

fixes implemented, and more thorough testing of the design.

Conclusions

Throughout the course of the design and implementation process, a lot of knowledge was gained in terms of op-amp circuits, and Power PCB design. Troubleshooting and debugging skills were also improved upon as needed to ensure that the final board was as error free as possible.

MCUB

The MCUB (Figure 10) is a custom-made PCB that acts as a central hub for all the onboard electronics. The MCUB receives a 5v signal from the power distribution boards and uses this to power the onboard control signals. The STM32 [1] and some other electronics use a 3.3V signal instead of a 5V signal so a 5V to 3.3V DC to DC converter [7] was included integrated as surface mount components. The PCB connects 5 sensors and three attachments to the board through Molex connectors [11] [2]. Molex connectors were chosen because they are widely available in different footprint sizes and provide stability in the electrical connections. The Sensors include temperature, humidity [6], leak [2], and pressure [5]. The attachments include three servos a light and a claw [12]. Some extra 3.3V headers with I2C connections were added to allow for design contingency. Electrical components that did not require specific placement on in our outside the enclosure were integrated into the PCB to save space and remove wiring mess inside the enclosure. Included was a data level shifter [8] to convert the 5V claw signal to 3.3V so the STM32 could read the signal. The STM 32 was also attached with headers to allows for stability and easy removal while debugging [1]. A 10 pin Molex header was also included to output eight PWM signals from the STM32 to electronic speed controllers [9]. Headers were also included to provide and receive data from the pie attached below [14]. These headers were attached along the bottom to allow for easier and less wiring. In terms of PCB specifications, we chose to use a 2-layer board as it was the cheapest and quickest to manufacture that provided enough room for the design. 2 oz 16 mil traces were chosen to minimize the resistance in the long traces and to allow for all current needs. If a much thinner trace was chosen it would cause possible line losses and thermal issues. A 1900 x 2050 mil cut-out [14] was added to the top of the board to allow for wires to pass through the layer stack.

Tether

The tether is an integral part of the ROV, and as such, needed to be constructed with care. Calculations and specifications were used to select the correct wire gauge for the voltage lines in the tether. This ensured minimal power losses and proper delivery of voltage and current to power converters within the ROV. All necessary cables had to be sleeved and potted to make it waterproof and easy to move. Once cables were sleeved, the ends were secured with tape and heat shrink, and the Ethernet and power cables crimped.

Buoyancy

The tether must be neutrally buoyant to allow for ease of movement for the ROV, with less obstruction and less extra work for the propulsion systems. First, the team attempted to determine this theoretically, however there were difficulties with the measurements, and so it was determined experimentally.

SOFTWARE

Embedded

The micro-controller used in KELP-II is an STM32, which allowed for team members to code precisely the functions desired from start to finish. One positive factor when using an STM32 is that it has sufficiently many PWM channels so as to remove any need for an external PWM controller unit, which has the benefit of simpler code, and fewer parts involved overall. Furthermore, the ADC pins from the STM32 allows for the analog signal (which is generated from the water pressure so we know how deep the ROV is) used from the pressure transducer to be converted into a digital signal. We can then apply an equation to the raw data to get usable data so we can read it on a computer and further analyze the data received in real time.

Drivers

First, we provide an overview of the development and integration of the thrusters module, task, and driver, as well as the updates made to the Data Aggregator module. These enhancements were implemented to enable the operation and control of the remotely operated vehicle (ROV) by facilitating its movement through the use of thrusters. The thrusters module is responsible for converting an 8-bit integer input into pulse-width modulation (PWM) values that control the movement of the ROV's thrusters. The PWM values represent the desired speed and direction of each thruster. To convert the 8-bit integer input to PWM, the thrusters module utilizes an algorithm that maps the input range to a corresponding PWM range. The specifics of this mapping algorithm can be found in the module's implementation. Once the conversion is performed, the resulting PWM values are passed to the Data Aggregator module for storage.

Next, the claw module was developed to handle the control of the ROV's claw mechanism. It receives a mode parameter, which can be set to open, closed, or idle. Based on the selected mode, the module converts it into corresponding values for two GPIO (General Purpose Input/Output) pins. These GPIO pins are responsible for controlling the movement of the claw.The Claw task plays a crucial role in the actuation of the claw mechanism. It initializes the GPIO pins associated with the claw to an idle state, ensuring the initial readiness of the claw. Additionally, the Claw task regularly updates the state of the GPIO pins based on the values stored in the Data Aggregator module.

Leak Sensor

The development and integration of the Leak Sensor driver and module provide the ROV with the ability to detect and provide feedback if there is a leak in its system. The Leak Sensor module initializes the driver, which incorporates an interrupt mechanism triggered by changes in the associated sensor pin. Upon detection of a change, the driver's method is called, currently printing debug information as a temporary feedback mechanism. This functionality enhances the ROV's safety by alerting users to potential leaks and enabling timely actions to address them.

PiComms

The development and integration of the PiComms driver and Internal Comms module enable the ROV to receive instructions via UART character messages. The STM32 micro-controller is configured with specific pins for UART communication. The initializer function sets up the UART registers and establishes a callback function triggered by received characters. Upon completion of a message, the ID is used to locate the appropriate callback function, which is then passed the received data. The length component ensures the validity of the data. This implementation allows external devices, such as a Raspberry Pi, to send command signals to the STM32 micro-controller, facilitating control and coordination of the ROV's core functionality.

MS5837 Sensor

In order to accurately determine the depth of our system, it was necessary to obtain pressure readings from the MS5837 pressure and temperature sensor. To achieve this, the team developed a driver in the C programming language specifically for this sensor. By adapting the existing Arduino library, originally written in C++, into C, we ensured that the pressure sensor would function correctly and provide accurate pressure data. This driver enables our system to effectively measure pressure, which is crucial for calculating depth accurately.

COBS

We implemented Consistent Overhead Bit Stuffing in both Unity VR and the STM32 micro-controller. This was done to address the complexity introduced by requiring messages to be terminated with two characters. By using the cobs algorithm, we were able to simplify the micro-controller callbacks responsible for receiving new data from the Unity VR scene, as it only required a single termination character for messages.

To achieve this, we found existing online libraries in both C and C# that implemented the cobs algorithm. We incorporated these libraries into our micro-controller code and Unity C\# code, respectively. Afterward, we conducted testing on the Unity scene to ensure the successful integration and functionality of the cobs bit stuffing implementation.

TCPClient

The TCPClient class creates a client socket to connect to a server using TCP/IP protocol. It uses Python's socket and threading modules for socket communication and multi-threading respectively.

TCP SERVER

The purpose of the TCP server is to enable communication between a topside computer and a Raspberry Pi on the remotely operated vehicle (ROV). Both the Pi and the computer are connected to the same network through a switch, allowing them to communicate using the TCP protocol. This is facilitated by a C\# class that simplifies the setup of TCP communication. The advantage of using a C\# class is that it can be easily integrated into Unity, a popular game development platform.

RPI UART Client

The development and integration of the TCP-UART bridge enable bi-directional communication between the STM and the Topside computer. The Raspberry Pi acts as a middleman, relaying information between the systems. The Python program running on the Raspberry Pi establishes a TCP connection with the Topside computer while communicating with the STM via UART. Debug information and mission-critical data are transmitted through separate UART channels. By implementing this bridge, reliable communication is established, allowing for effective control and coordination between the STM and the Topside computer.

Control and Comms within Unity: KelpUI

The development and integration of controls and communications within Unity enabled the ROV to receive control commands and display received data in a VR scene. The setup involved three Unity scenes, each serving a specific purpose. The implementation included creating input action assets, initializing callbacks, and managing commands and communications through dedicated managers. The UI elements within the VR control room facilitated the visualization of ROV data. This development enhanced the control and monitoring capabilities of the ROV system, providing a more immersive and interactive experience for users.

Streaming Cameras

The setup of webcam streaming from a Raspberry Pi to Unity VR allowed for the visualization of the Pi camera feed within the VR environment. By implementing the exploreHD camera streaming setup and utilizing OBS with virtual cameras, multiple camera feeds could be received and processed in Unity. This setup enhanced the immersive experience of the VR environment by providing real-time video input from the Pi cameras.

Protobuf

The update to data transfer protocols using Google ProtoBuf addressed issues encountered during thruster testing. By implementing data compression, compact messages, and controlled transmission rates, the efficiency and reliability of data transfer between Unity and STM were significantly improved. The use of ProtoBuf reduced the number of bytes required to store messages, ensuring more streamlined communication. The optimized data transfer protocols contributed to the successful operation of the ROV system and enhanced its overall performance.

piCore on linux

The team set up piCore, a lightweight version of Tinycore Linux for the Raspberry Pi, with camera streaming and communication utilities. This was done to create a fast-booting and lightweight system that runs in RAM, making it resistant to file corruption during shutdowns. The file system is recreated from a backup at each boot, ensuring a fresh and reliable system.

To enable camera streaming, we utilized the GStreamer framework. The process involved writing the piCore image to an SD card, configuring the partitioning, and installing necessary packages such as v4l2-utils, GStreamer, and Python libraries. We also copied the Python client and launch script from a specified GitHub repository into the home directory.

Additional steps included adding the launch script to the boot process, setting a static IP address for the control network, and modifying the /config.txt file to enable UART communication. Changes were saved using the filetool.sh utility.

UART ON PICORE

The task of adding UART support to the PiCore installation on the SD card was accomplished by modifying the config.txt file. This addition was essential for facilitating communication between the Network Client and the PiCommsModule. By enabling UART functionality, the Pi and the STM were able to exchange data effectively. The modifications to the config.txt file showcased an understanding of system configurations and allowed for the successful implementation of UART support on the PiCore installation.

ATTACHMENTS AND MISSION-RELATED TOOLS

Camera Mounts

This year, KELP-II uses three DeepWater Exploration exploreHD 3.0 cameras, which have their own mounts, but these needed attachments in order to be able to be secured to the frame. These mounts were designed to allow for a servo to move the camera so different angles could be used to get a wide field of vision.

Float

Kelpie Robotics has designed a vertical profiling float capable of completing two vertical profiles in an underwater environment.

It utilizes a buoyancy engine made a 300mL syringe to displace seawater and control the density of the float, and thus control the vertical height of the float.

Using an Inertial Measurement Unit, a Real Time Clock module, a motor controller, and a Wi-Fi-enabled microcontroller unit, it is able to perform vertical profiles and send messages back to the ground station. For more information, see the submitted Non-ROV Device Specification document.

SAFETY

Safety is of the utmost importance for Kelpie Robotics, and as such, we are thoroughly dedicated to safe lab practices as well as ensuring the safety of all team members when interacting with the ROV.

LAB PROTOCOLS

All members of the company must have their Dry Lab Risk Management Training before being allowed to work in the Team Space, which is classfied as a Dry Lab. Furthermore, Job Site Safety Analyses (JSA) are used to ensure a safe work environment for all team members. Material Data Sheets (MDS) are also available for team members' use.

TRAINING

In addition to the Dry Lab Risk Management training, team members must also complete training through the University of Ottawa before use of various types of machinery so as to ensure the safety of the members using it.

VEHICLE SAFETY FEATURES

KELP-II is in compliance with several specifications to allow for the ROV to be as safe as possible. Firstly, Anderson Powerpole connectors will be the main point of connection to the MATE power supply. It also has a suitable fuse, a 30A fuse, within 30cm of the main point of connection. The control box does not have any exposed wiring and is neatly and carefully laid out with wiring clearly labelled and separated. The topside control unit is also carefully designed to eliminate any exposed wiring, with components arranged in an efficient, easy-access manner. The KELP-II tether also has appropriate, supported strain relief and waterproofed cables. The enclosure for our electronics is specifically designed for underwater environments, and is rated for 200ft. All propellers are shrouded and guarded.

LOGISTICS

Project Management

Kelpie Robotics made use of one system mainly - Trello - to keep members up-to-date with all aspects of the project, manage deadlines, and assign tasks. The Trello was split into two sections: stages of tasks and phases of design.

Design Phases

To be able to clearly see, at a glance, which tasks had had what amount of progress, we had 7 main stages, as seen in Figure 11. First was Backlog, i.e. what had not yet been started in any capacity. Next, we had Design: which tasks were currently in the process of being designed. After that, the third stage was split into three categories of tasks that were not actively being worked on: Parked, Blocked, and To-do. Parked tasks were tasks that were currently on-hold, and would need to be picked up at a later date. Blocked tasks were tasks that were being held up by outside factors, such as another task, unavailable parts, etc. To-do was simply which tasks had finished designs but needed to then be started. The fourth stage was Doing, which referred to tasks that were in-progress. Next stage was Peer-Review, where whoever was working on a task would have another team member review their work to ensure quality. The sixth stage was Testing, for tasks that had been completed, and peer-reviewed and now were ready to undergo testing to ensure safety, functionality, and efficiency. Last, we had “Done!” which was where tasks that were completely finished the whole process would be left.

At Kelpie Robotics, we go through five stages of design: Preliminary Design Review, Critical Design Review, Core Systems Testing, Comprehensive Systems Testing, and Pre-Competition. These phases are shown in Figure 12. Preliminary Design Review included rough ideas for designs, things that would then need refinement during Critical Design Review. Next, Core Systems Testing involves dry-tests and testing of integral systems. After, Comprehensive Systems Testing refers to tests that involve all aspects of the ROV. Finally, Pre-Competition is the phase in which the team will practice the completion of tasks, and make sure all submissions are finished. These are all tracked on our Trello for team member tracking.

Company Organization

Kelpie Robotics is structured into four main sub-teams: Mechanical, Electrical/Embedded, Software, and Business/Logistics. Each team has at least one officer; for technical sub-teams, each has a CTO and Business/Logistics is headed by the COO and CFO. All members report to the CEO.

Specific projects were led by Project Managers (PMs), allowing significant tasks, such as the float, PCB design, and the tether, to have more specific leadership.

Team members were not necessarily assigned to a single sub-team in order to encourage a multi-disciplinary approach to development.

Code Management

Kelpie Robotics uses GitHub to facilitate code development across multiple team members, and allows for all code to be efficiently managed across multiple versions and projects. Github is a Version Control System, or VCS, that is very widely used.

To ensure efficient parallel code development across multiple software employees, Rovotics utilizes GitHub, a Version Control System (VCS). By using a VCS, Rovotics kept track of overall changes to the software and managed multiple versions. GitHub was selected because it is a well-supported distributed VCS that provides each programmer with a remote and local copy of the code repository. GitHub also enabled software branching and merging which is important when multiple people are working on interdependent files. Should problems arise, GitHub allows restoration of previous versions.

Project Cost and Budgeting

As we near the end of our competition season, many of the items on our budget have been purchased. These costs were first estimated by considering last year's costs and our hopes for new designs, as well as getting quotes and researching the best quality, balancing this with cost-efficacy. For travel and lodging, as the location of the competition is sharing partway through the season, we must consider all locations that are possible when making these predictions. We were also able to be more cost-effective by using parts from our previous iteration, KELP-I.

See Appendix B for a detailed budget breakdown.

CONCLUSIONS

Challenges, Skills, Lessons

Throughout the process of research, design, manufacturing and testing, our company learned many valuable lessons and faced many challenges. In particular, our team members faced some difficulties with trying new technologies, but in the process, learned a lot that they will be able to take with them in their future endeavours. One of our main challenges is finding pools to test in while we are still at the end of the winter, however this gave our company the opportunity to engage more with our community. Our members also had the opportunity to hone many skills, including manufacturing, coding with new languages, and working together in trying conditions. We hope to take this newfound knowledge, and newly developed skills into our future designs and future opportunities.

We would like to extend a huge thank you to our mentors, Jason Demers and Hassaan Zaki. We could not have accomplished what we have without the contributions of our sponsors Ciena, Larus Technologies, and RBR. We also are very grateful for the support of the IEEE Canada Foundation

Acknowledgements

Kelpie Robotics would like to thank:

- MATE Center and Marine Technology Society
- National Science Foundation - For support of the MATE Competition
- University of Ottawa - John McIntyre Team Space - For their continued support
- University of Ottawa - CEED - For their continued support
- University of Ottawa - Engineering Endowment Fund - For their financial support
- IEEE Canada Foundation - For their support in our endeavours
- Ciena - For their support in our endeavours
- Jason Demers - For his support, insight, and guidance
- Our Community - For their support, encouragement, and patience with pool testing
- JMTS Teams - For their encouragement, advice, and laughter during the late nights

APPENDICES

Appendix A - Figures

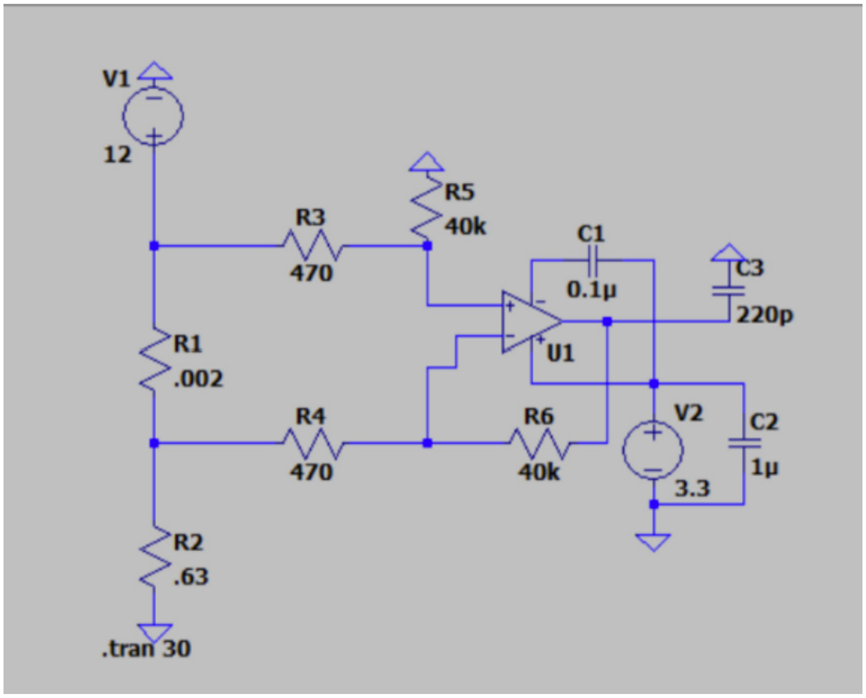


Figure 2 - Current Sense Circuit using a Shunt Resistor and Instrumentation Amplifier

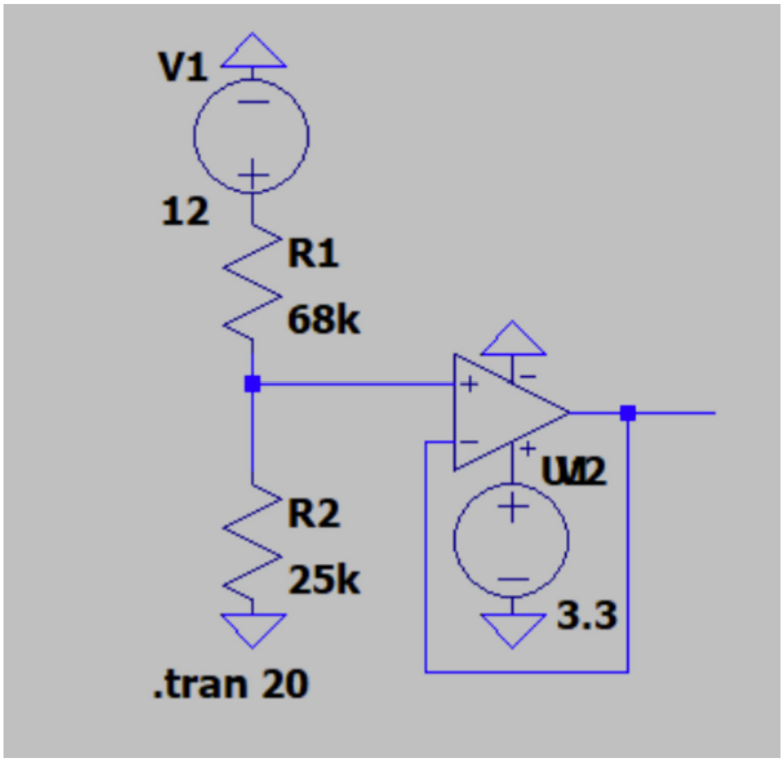


Figure 3 - Voltage Measuring Circuit Consisting of a Voltage Divider and Voltage Follower

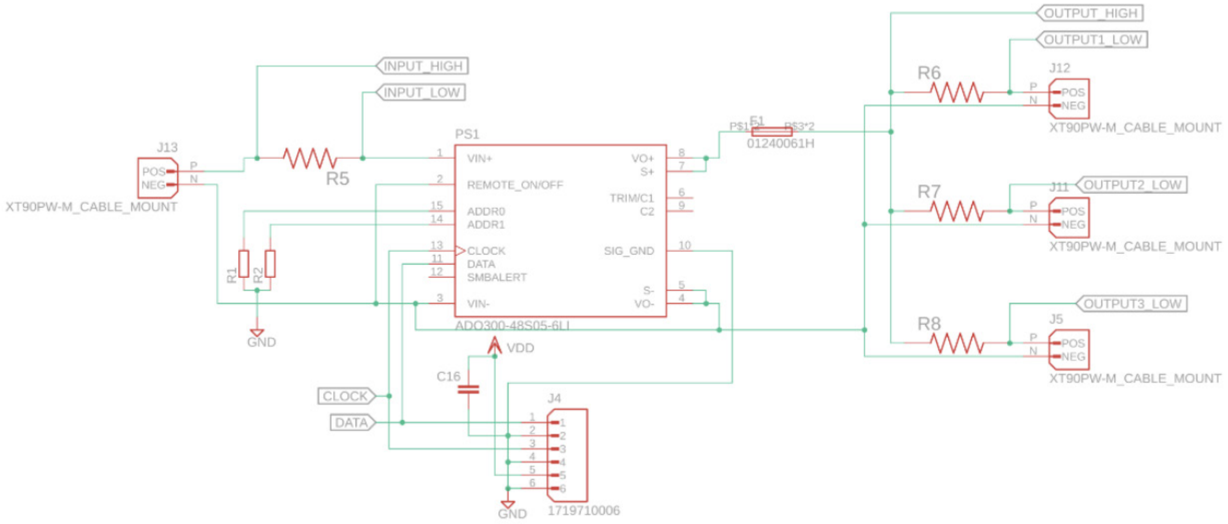


Figure 4 - Power Conversion Circuit

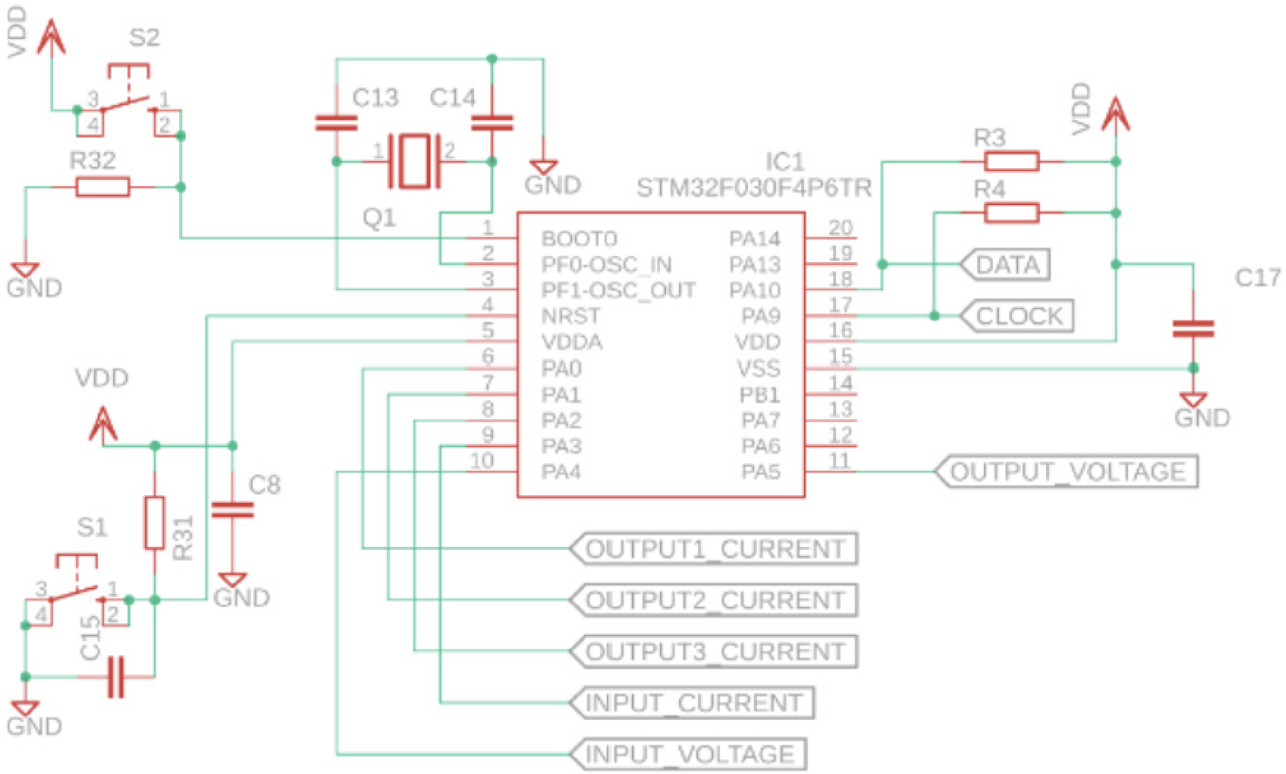


Figure 5 - Micro-controller Unit (Power Boards)

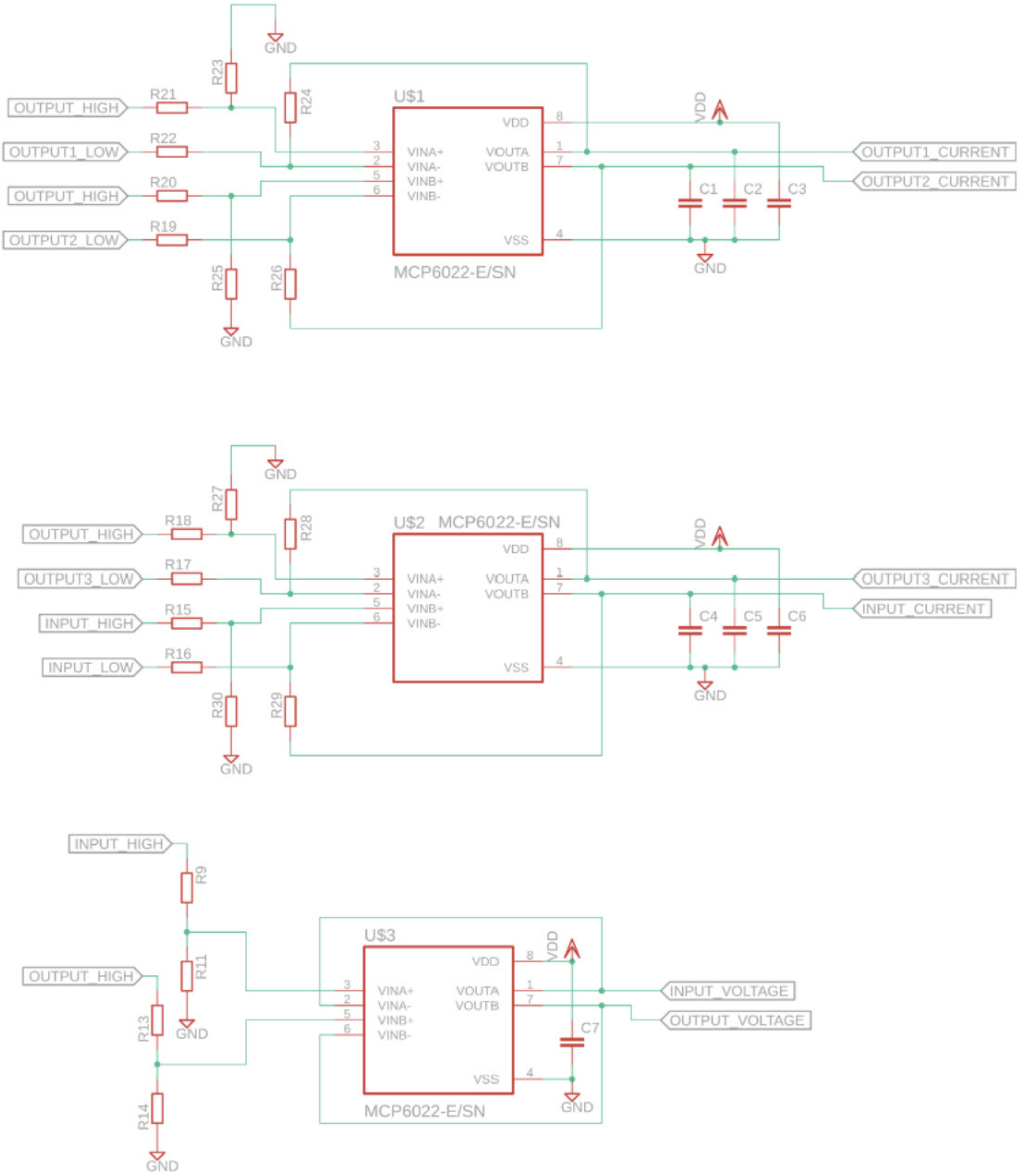


Figure 6 - OP-AMP Circuits

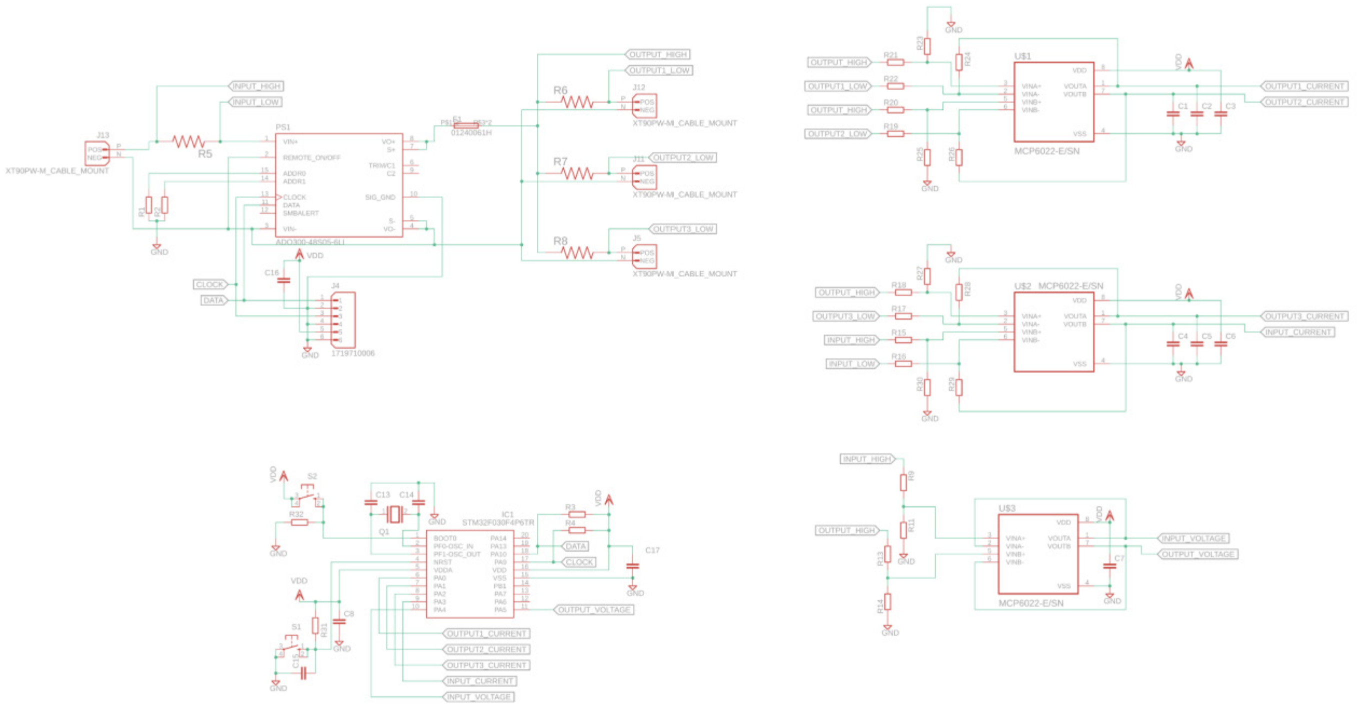


Figure 7 - Completed Circuit for Power Distribution Boards

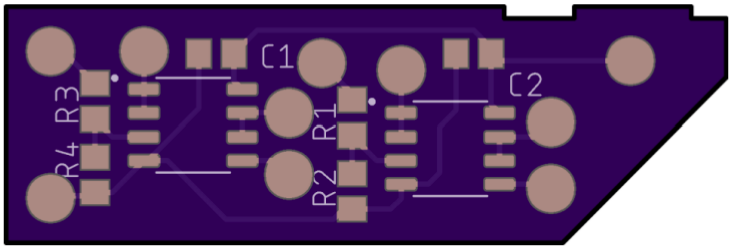


Figure 8 - Mod Board for 48-24V and 48-5V Boards

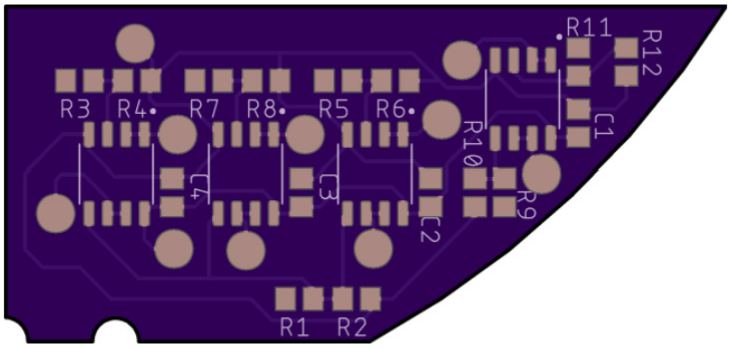


Figure 9 - Mod Board for 48-12V Boards

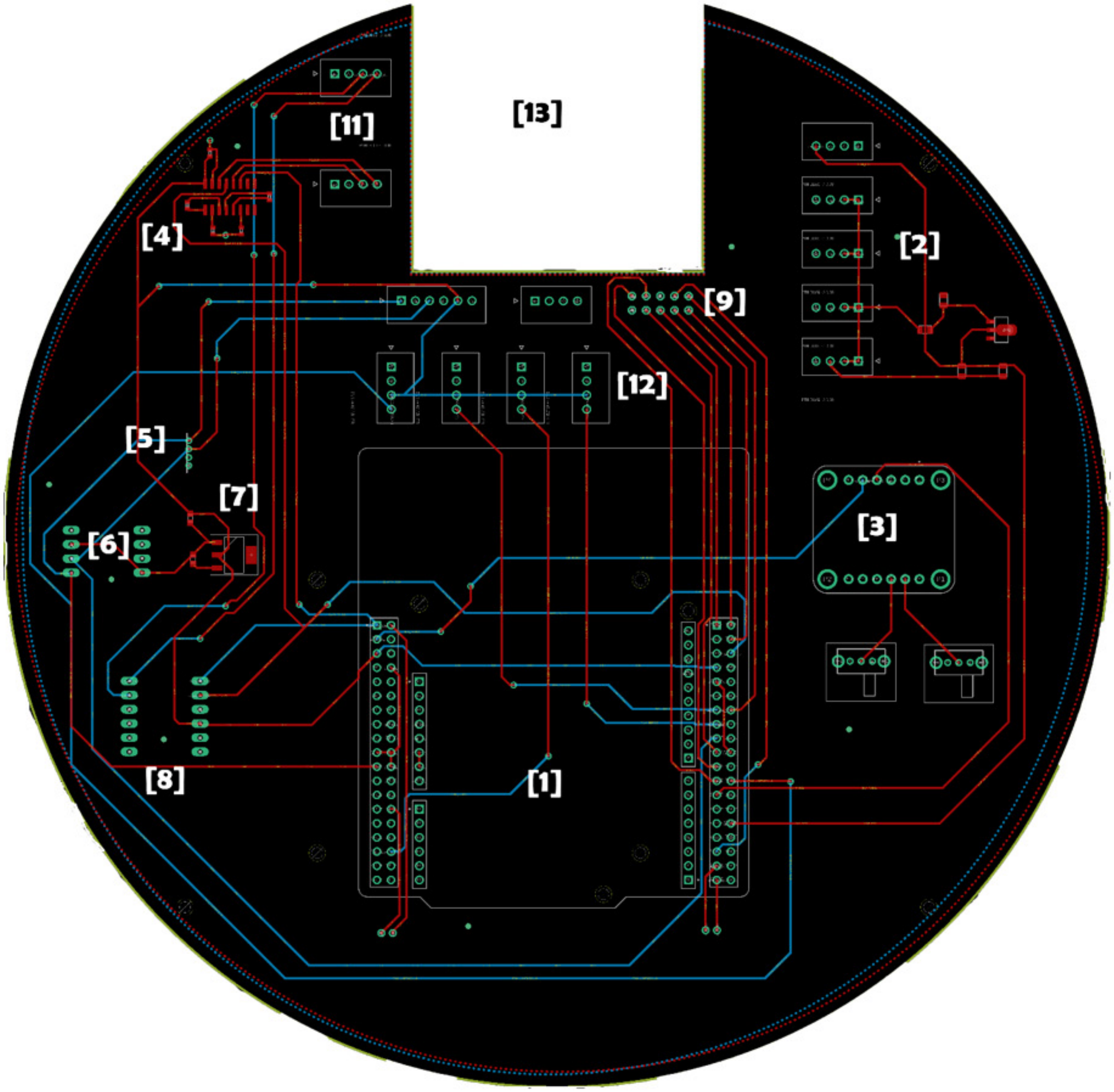


Figure 10 - MCU Board [MCUB]

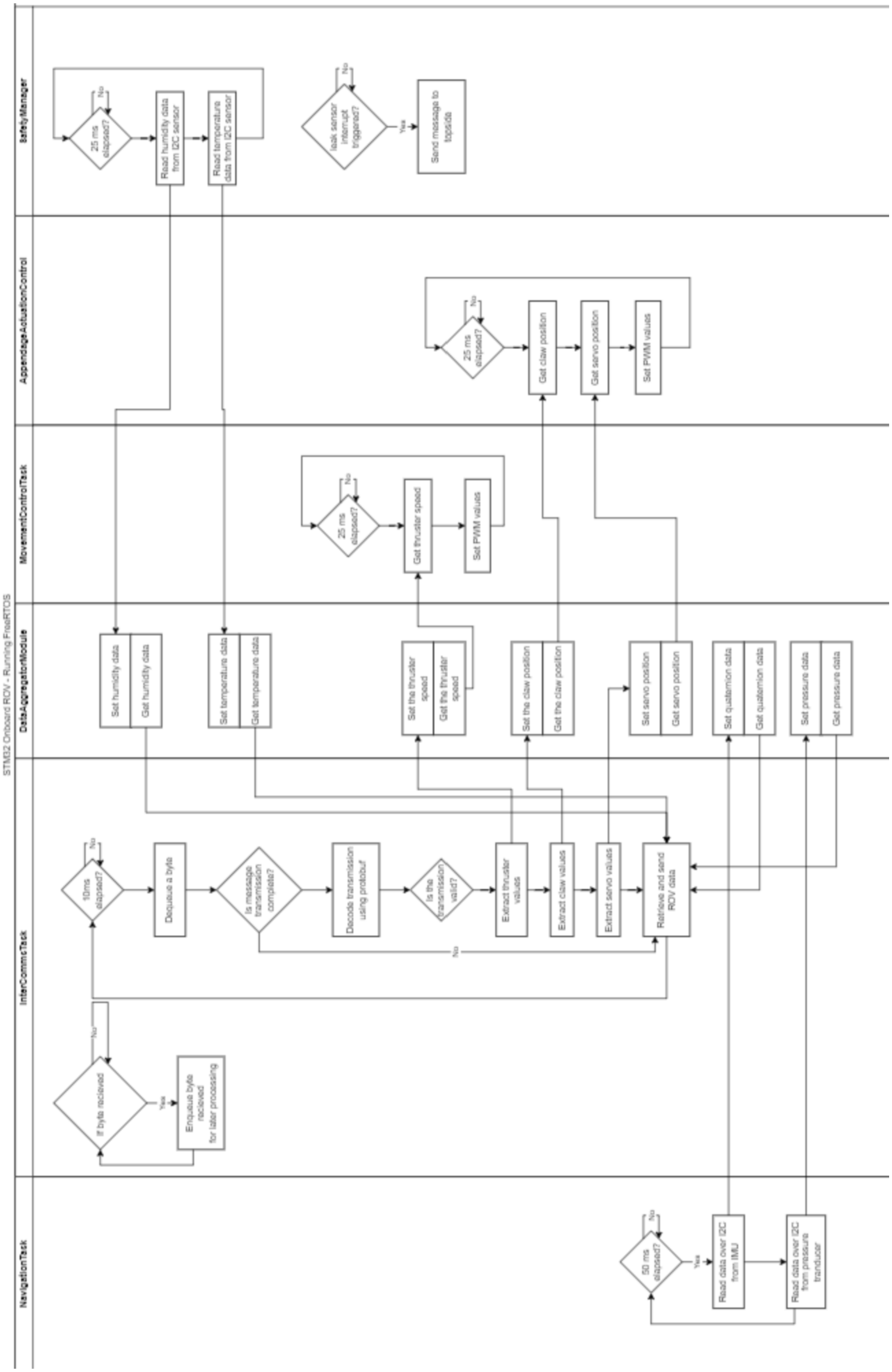
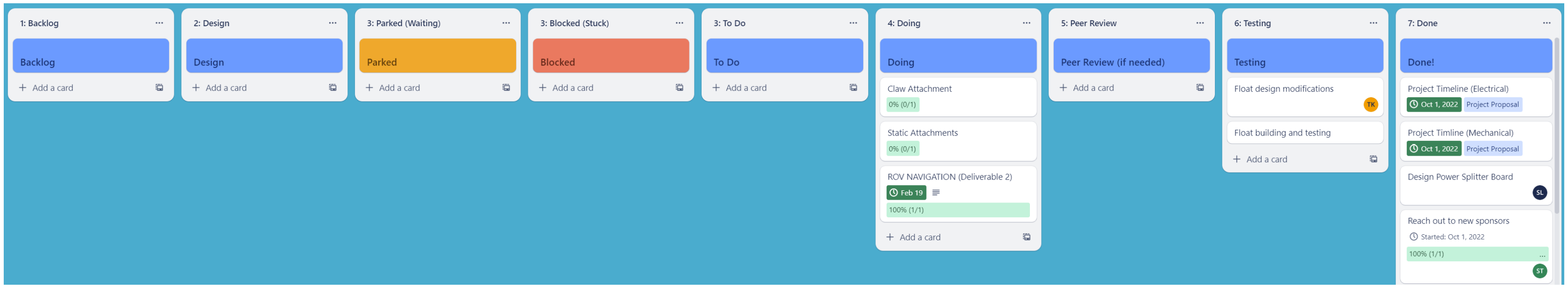


Figure 11 - Embedded System Flow



1: Backlog
- Backlog
- Add a card

2: Design
- Design
- Add a card

3: Parked (Waiting)
- Parked
- Add a card

3: Blocked (Stuck)
- Blocked
- Add a card

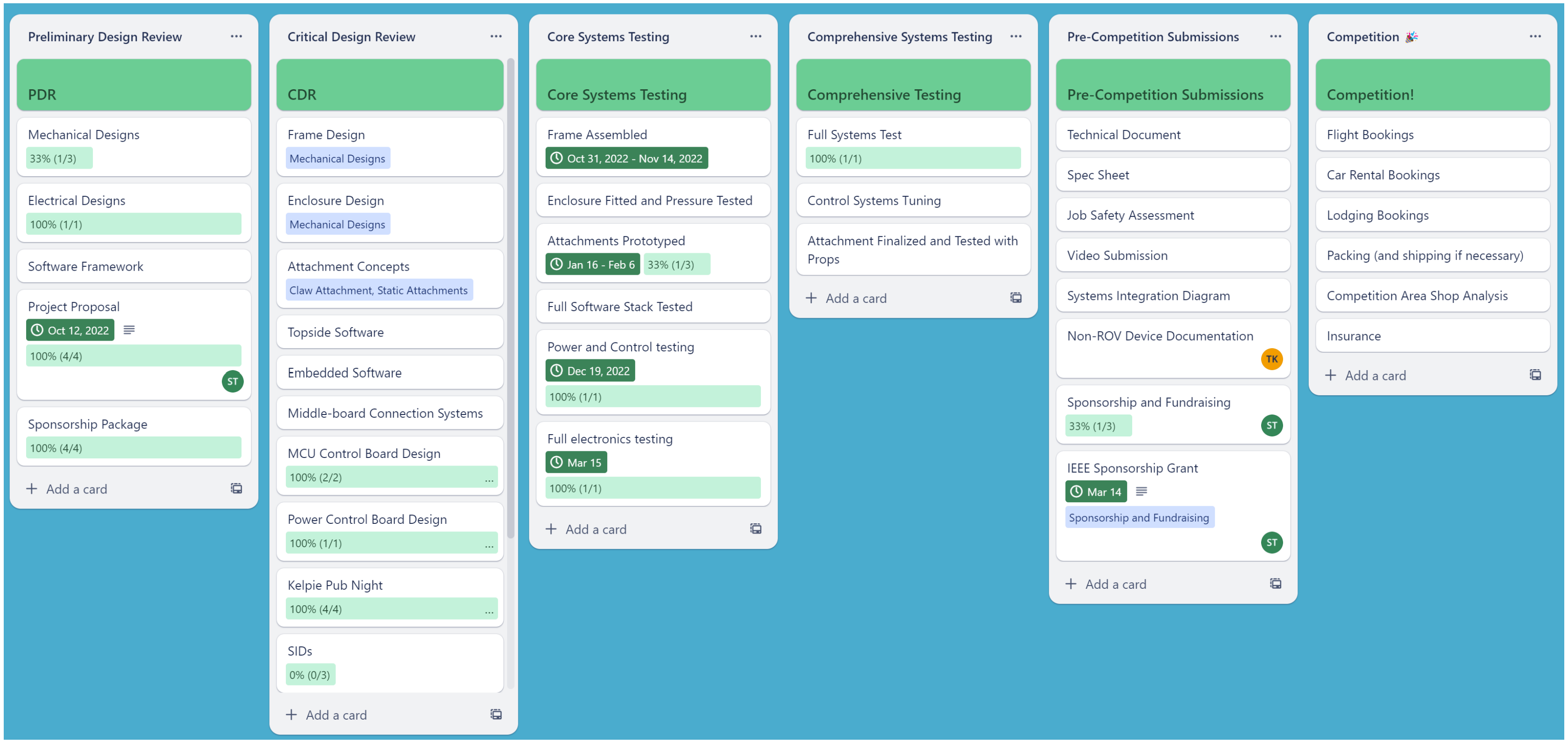
3: To Do
- To Do
- Add a card

4: Doing
- Doing
- Claw Attachment 0% (0/1)
- Static Attachments 0% (0/1)
- ROV NAVIGATION (Deliverable 2) Feb 19 100% (1/1)
- Add a card

5: Peer Review
- Peer Review (if needed)
- Add a card

6: Testing
- Testing
- Float design modifications TK
- Float building and testing
- Add a card

7: Done
- Done!
- Project Timeline (Electrical) Oct 1, 2022 Project Proposal
- Project Timline (Mechanical) Oct 1, 2022 Project Proposal
- Design Power Splitter Board SL
- Reach out to new sponsors Started: Oct 1, 2022 100% (1/1) ST



Preliminary Design Review
- PDR
- Mechanical Designs 33% (1/3)
- Electrical Designs 100% (1/1)
- Software Framework
- Project Proposal Oct 12, 2022 100% (4/4) ST
- Sponsorship Package 100% (4/4)
- Add a card

Critical Design Review
- CDR
- Frame Design Mechanical Designs
- Enclosure Design Mechanical Designs
- Attachment Concepts Claw Attachment, Static Attachments
- Topside Software
- Embedded Software
- Middle-board Connection Systems
- MCU Control Board Design 100% (2/2)
- Power Control Board Design 100% (1/1)
- Kelpie Pub Night 100% (4/4)
- SIDs 0% (0/3)
- Add a card

Core Systems Testing
- Core Systems Testing
- Frame Assembled Oct 31, 2022 - Nov 14, 2022
- Enclosure Fitted and Pressure Tested
- Attachments Prototyped Jan 16 - Feb 6 33% (1/3)
- Full Software Stack Tested
- Power and Control testing Dec 19, 2022 100% (1/1)
- Full electronics testing Mar 15 100% (1/1)
- Add a card

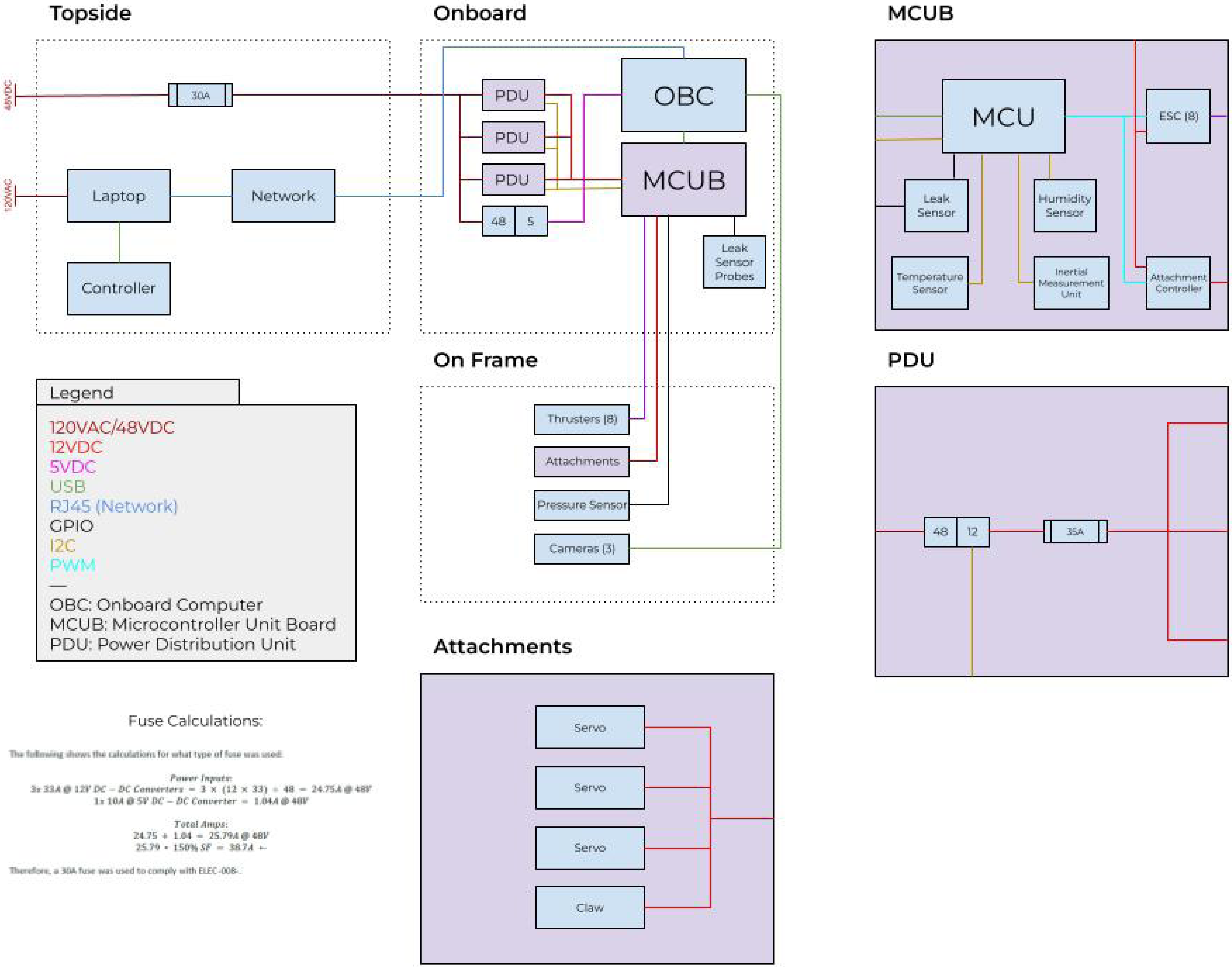
Comprehensive Systems Testing
- Comprehensive Testing
- Full Systems Test 100% (1/1)
- Control Systems Tuning
- Attachment Finalized and Tested with Props
- Add a card

Pre-Competition Submissions
- Pre-Competition Submissions
- Technical Document
- Spec Sheet
- Job Safety Assessment
- Video Submission
- Systems Integration Diagram
- Non-ROV Device Documentation TK
- Sponsorship and Fundraising 33% (1/3) ST
- IEEE Sponsorship Grant Mar 14 Sponsorship and Fundraising ST
- Add a card

Competition 🎉
- Competition!
- Flight Bookings
- Car Rental Bookings
- Lodging Bookings
- Packing (and shipping if necessary)
- Competition Area Shop Analysis
- Insurance
- Add a card

Figure 12 - Stages of tasks shown in Trello (Left)

Figure 13 - Design phases shown in Trello (Right)



Fuse Calculations:

The following shows the calculations for what type of fuse was used:

Power Inputs:

$$3x\ 33A\ @\ 12V\ DC-DC\ Converters = 3 \times (12 \times 33) \div 48 = 24.75A\ @\ 48V$$

$$1x\ 10A\ @\ 5V\ DC-DC\ Converter = 1.04A\ @\ 48V$$

Total Amps:

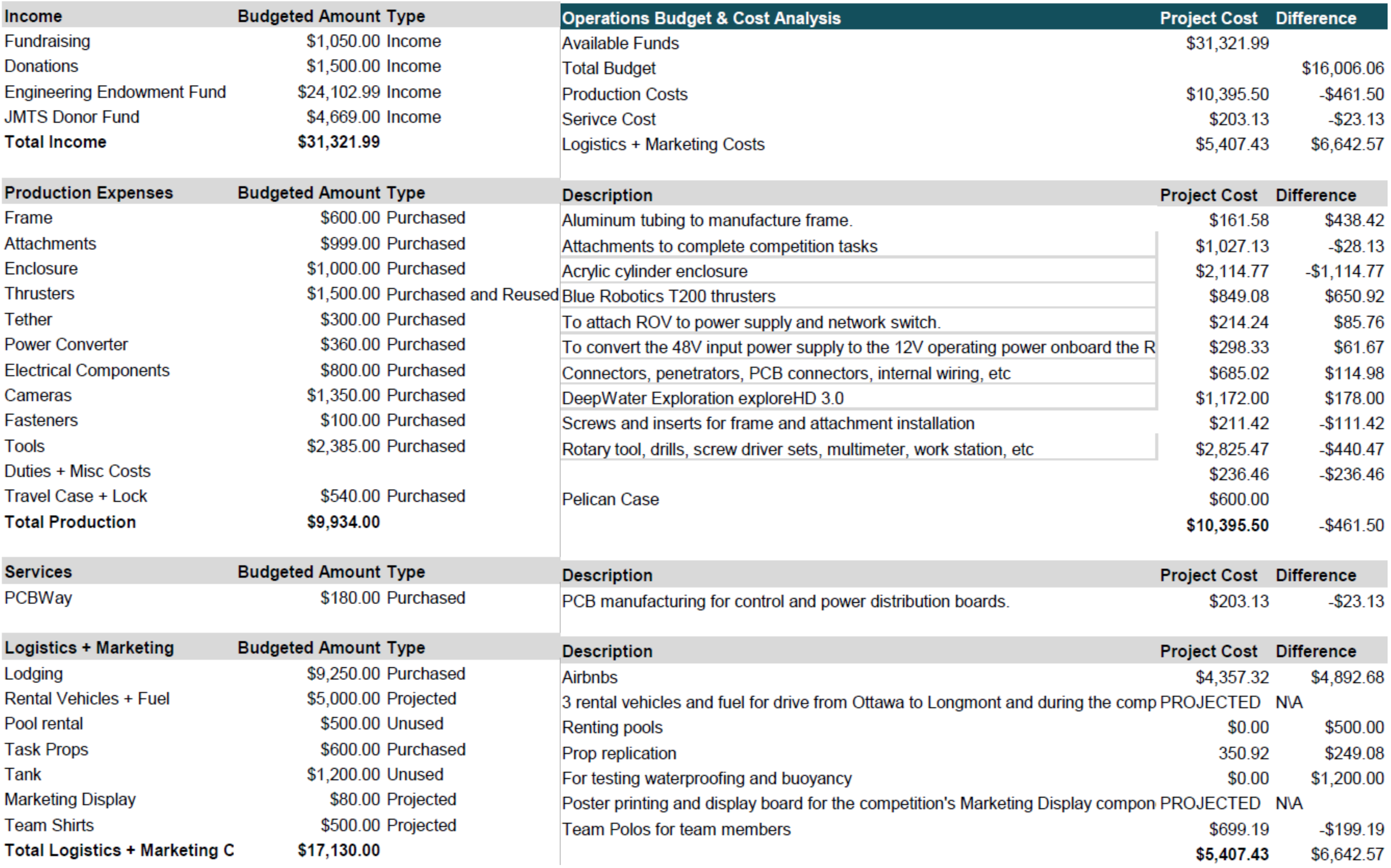
$$24.75 + 1.04 = 25.79A\ @\ 48V$$

$$25.79 \times 150\%\ SF = 38.7A \leftarrow$$

Therefore, a 30A fuse was used to comply with ELEC-008.

Figure 14 - System Integration Diagram (SID)

APPENDIX B - BUDGET BREAKDOWN



Income	Budgeted Amount	Type	Operations Budget & Cost Analysis	Project Cost	Difference
Fundraising	$1,050.00	Income	Available Funds	$31,321.99	
Donations	$1,500.00	Income	Total Budget		$16,006.06
Engineering Endowment Fund	$24,102.99	Income	Production Costs	$10,395.50	-$461.50
JMTS Donor Fund	$4,669.00	Income	Serivce Cost	$203.13	-$23.13
Total Income	**$31,321.99**		Logistics + Marketing Costs	$5,407.43	$6,642.57

Production Expenses	Budgeted Amount	Type	Description	Project Cost	Difference
Frame	$600.00	Purchased	Aluminum tubing to manufacture frame.	$161.58	$438.42
Attachments	$999.00	Purchased	Attachments to complete competition tasks	$1,027.13	-$28.13
Enclosure	$1,000.00	Purchased	Acrylic cylinder enclosure	$2,114.77	-$1,114.77
Thrusters	$1,500.00	Purchased and Reused	Blue Robotics T200 thrusters	$849.08	$650.92
Tether	$300.00	Purchased	To attach ROV to power supply and network switch.	$214.24	$85.76
Power Converter	$360.00	Purchased	To convert the 48V input power supply to the 12V operating power onboard the R	$298.33	$61.67
Electrical Components	$800.00	Purchased	Connectors, penetrators, PCB connectors, internal wiring, etc	$685.02	$114.98
Cameras	$1,350.00	Purchased	DeepWater Exploration exploreHD 3.0	$1,172.00	$178.00
Fasteners	$100.00	Purchased	Screws and inserts for frame and attachment installation	$211.42	-$111.42
Tools	$2,385.00	Purchased	Rotary tool, drills, screw driver sets, multimeter, work station, etc	$2,825.47	-$440.47
Duties + Misc Costs				$236.46	-$236.46
Travel Case + Lock	$540.00	Purchased	Pelican Case	$600.00	
Total Production	**$9,934.00**			**$10,395.50**	-$461.50

Services	Budgeted Amount	Type	Description	Project Cost	Difference
PCBWay	$180.00	Purchased	PCB manufacturing for control and power distribution boards.	$203.13	-$23.13

Logistics + Marketing	Budgeted Amount	Type	Description	Project Cost	Difference
Lodging	$9,250.00	Purchased	Airbnbs	$4,357.32	$4,892.68
Rental Vehicles + Fuel	$5,000.00	Projected	3 rental vehicles and fuel for drive from Ottawa to Longmont and during the comp	PROJECTED	N\A
Pool rental	$500.00	Unused	Renting pools	$0.00	$500.00
Task Props	$600.00	Purchased	Prop replication	350.92	$249.08
Tank	$1,200.00	Unused	For testing waterproofing and buoyancy	$0.00	$1,200.00
Marketing Display	$80.00	Projected	Poster printing and display board for the competition's Marketing Display compon	PROJECTED	N\A
Team Shirts	$500.00	Projected	Team Polos for team members	$699.19	-$199.19
Total Logistics + Marketing C	**$17,130.00**			**$5,407.43**	$6,642.57

Figure B1 - Budget Breakdown