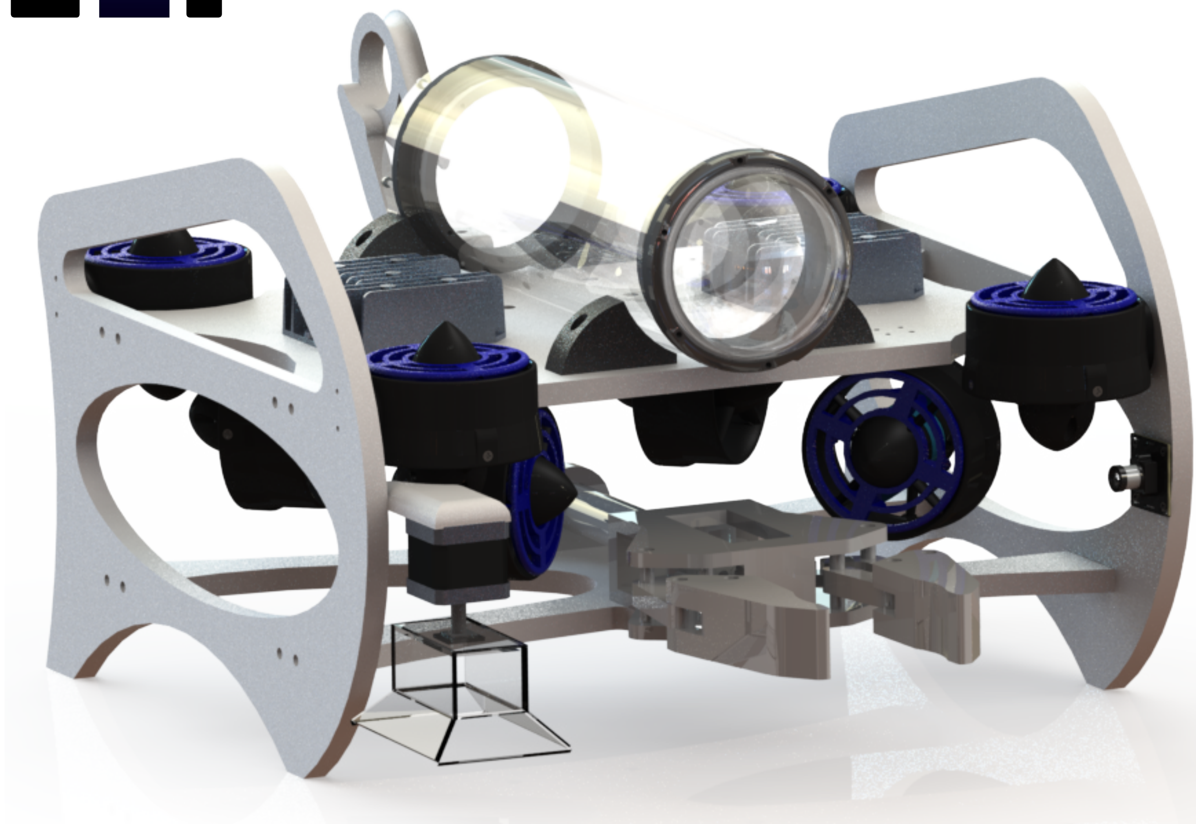
KEA GOES DEEP

KEA
COPENHAGEN SCHOOL OF DESIGN AND TECHNOLOGY

MATE



Christopher Nielsen
mentor / CFO
teacher
36 years old, from Denmark

Stefan Oehme
CEO / project manger
Production Technology - 4th semester
36 years old, from USA

Matej Majtan
software team
IT Technology - 4thsemester
20 years old, from Slovakia

Jakub Petrík
software team
Computer Science - 4th semester
21 years old, from Slovakia

René Michal Cehlár
software team
Computer Science - 4th semester
21 years old, from Slovakia

Martynas Bliuvas (new)
hardware team
IT Technology - 4th semester
26 years old, from Lithuania

Miroslav Lakota
hardware team
IT Technology - 4th semester
22 years old, from Slovakia

Jakub Machacek (new)
software team
Computer Science- 4th semester
21 years old, from Solvakia

Cristi Betivu (new)
software team
Computer Science- 4th semester
20 years old, from Moldova

Domenico Villani (new)
software team
IT Technology - Electronics 4th semester
30 years old, from Italy

Bogdan Ionescu
mechanics team
Production Technology - 4th semester
21 years old, from Romania

Jacob Sommerseth
mechanics team
Production Technology - 4th semester
25 years old, from Norway

Paweł Waliszewski
mechanics team
Production Technology - 4thsemester
26 years old, from Poland

Abstract

KEA Goes Deep is a student run company from the Copenhagen School of Design and Technology that was founded with the intent to develop and produce an ROV with the capability to fulfill the various requirements of the RFP posted by the Applied Physics Laboratory at the University of Washington.

Our team consists of twelve students from nine different countries, attending three different school programmes. We pride ourselves on our international outlook and ability to utilize our cultural diversity to come up with creative solutions to challenging problems.

2018 will be the second year KGD attends the MATE competition, and the first year participating in the explorer class. To compete in this years explorer class, it was necessary to widen the functionality, rework the design and build upon our experiences from our last ROV design, Robbie.
Using the foundation and experiences gained from competing in the 2017 MATE competition and many late nights of work, we have developed our current ROV, “Bubbles”, that we proudly look forward to demonstrating at the competition in Seattle.

Through the course of this project, we have also helped to develop smaller ROV prototypes for the “ROV Maker Competition” - an organisation that offers an intensive, creative and exciting learning program aimed at students from technical high schools in Denmark. By doing this, we hope to have promoted the interest in ROVs and underwater technology, as well as shared some of the experiences and knowledge we have gained.

Table of contents

KEA Goes Deep

The Company

As mentioned in the abstract, KGD consists of 12 members and our mentor, Christopher Nielsen.

The company is organized into three main teams, plus a CEO/project manager, responsible for being the communicative link between the team and our mentor/university.

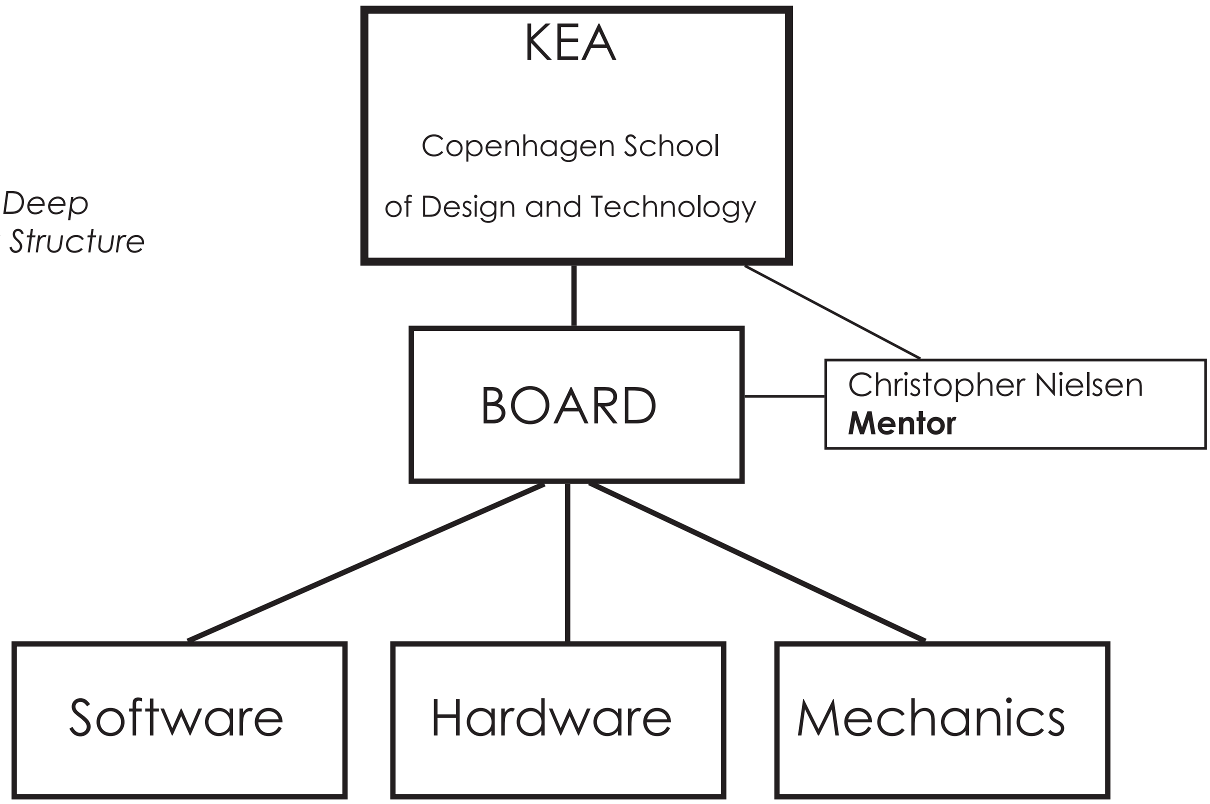
Hardware team: Responsible for development of electronics hardware
Software team: Responsible for developing software for controlling the ROV and any supplementary tools developed.
Mechanical team: Responsible for design of physical parts of the ROV, including tools necessary to complete the missions as described by MATE

Each of these teams has a designated team leader who is responsible for communication between teams, and represents the teams in board meetings. Internally, each team has a flat structure, meaning that each team member has an equal vote. Any decisions that has a cross-team impact were made during team meetings with all members having an equal vote.

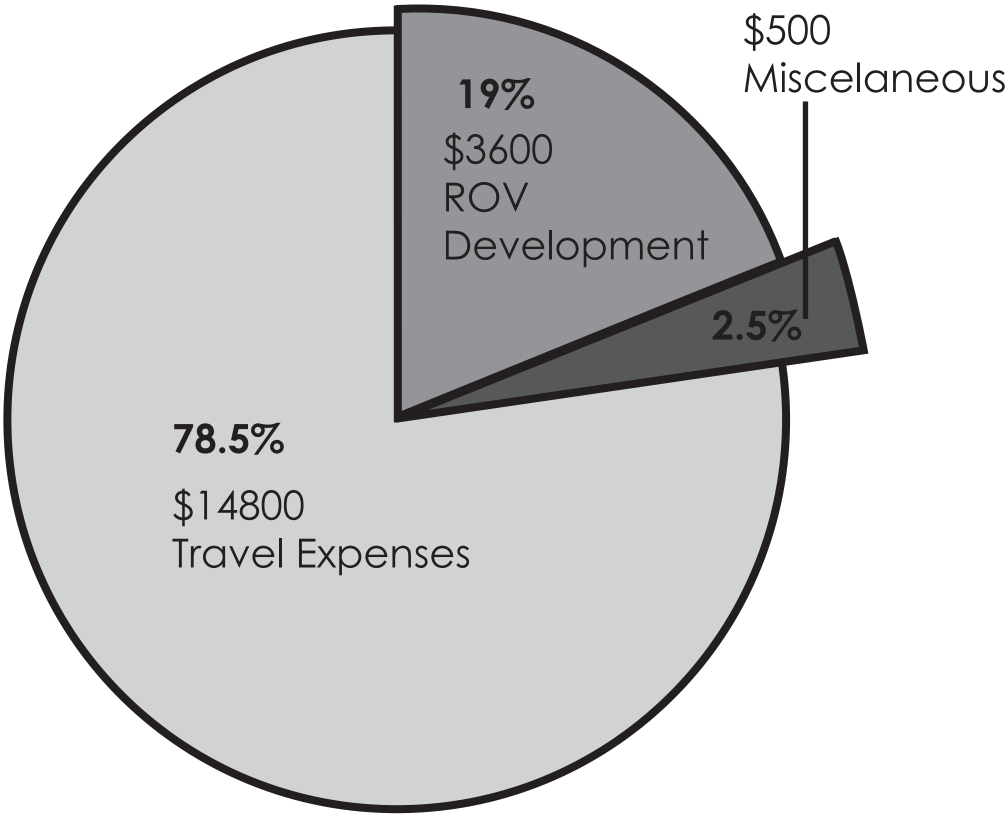
With the increased workload from competing in the Explorer class and some some previous members graduating, our demand for manpower increased. Because of this, in September of 2017 we organized events where we presented our project and recruited new potential members.

All the new members quickly became a part of our team dynamic, and we kept the camaraderie that won us the 2017 Aloha Team Spirit award.

KEA Goes Deep
Company Structure



Budget



Mechanical parts	Type	Cost (USD)	Market value (USD)
PP stock for frame/manipulator	Purchased	293,64	
Fasteners/inserts	Purchased	58,93	
3D-printing filament	Donated	N/A	44,07
Epoxy/acetone/etc.	Purchased	61,7	
Acryllic	Donated	N/A	8,4
Foam/buoyancy materials	Re-Used	N/A	15,7
Plywood for prototyping	Donated	N/A	39,74
Electronics enclosure	Re-Used	N/A	142,9
Parts for manipulator (lead screw, bearings etc.)	Purchased	42,7	
Air supply tube	Purchased	47,21	
		Sum paid	$504.18
		Total value	$754.99

Electronic parts	Type	Cost (USD)	Market value (USD)
Speed Controllers	Purchased	298	N/A
Thrusters	Purchased	330	N/A
Thrusters	Re-Used		660
Stepper motors	Purchased	58,47	N/A
Stepper motor drivers	Purchased	22,05	N/A
Power supply 48V 30A	Purchased	88,5	N/A
DC-DC converters	Purchased	56,43	N/A
Electrical wiring	Re-Used		50
Electrical wiring	Purchased	87	N/A
PCB manufacturing	Purchased	63	N/A
PCB components	Purchased	130	N/A
Raspberry Pi + camera	Re-Used		97,6
ESP32	Purchased	49,7	N/A
Additional small components	Purchased	200	N/A
Cost of shipping	Purchased	240	N/A
		Cost	$1623,15
		Market value	$2430,75

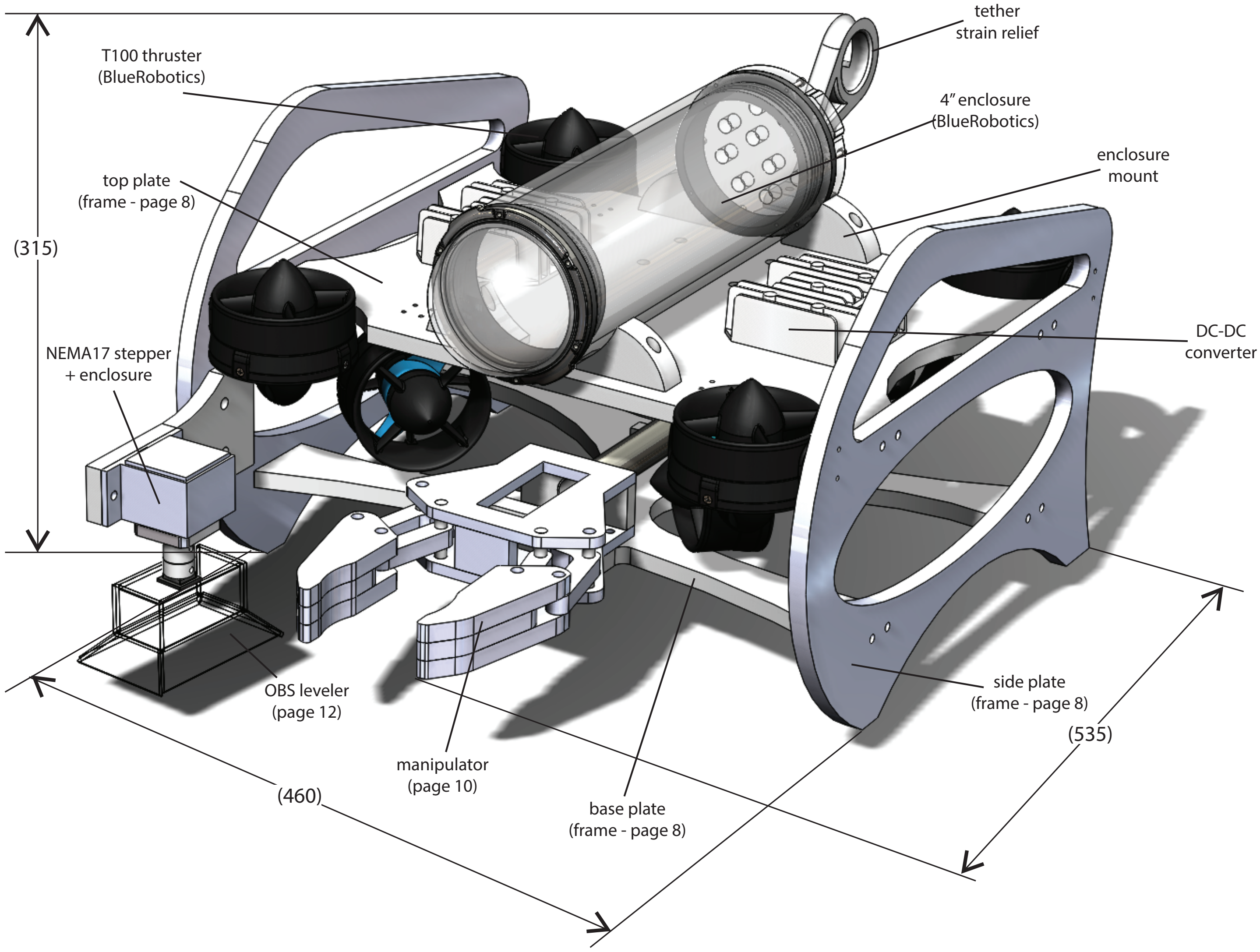
Ground station	Type	Cost (USD)	Market value (USD)
Xbox controller	Re-Used		50
Notebook Lenovo Thinkpad T410	Donated		500
Waterproof case	Re-Used		300
		Cost	$0
		Market value	$850

Travel Expenses	14800
Airfare (12 members + mentor)	$8000
Accommodation (7 rooms 5 nights)	$6500
Transportation (rental van, gas)	$300

ROV development	
Mechanical budget	$1100
Electronics budget	$2500

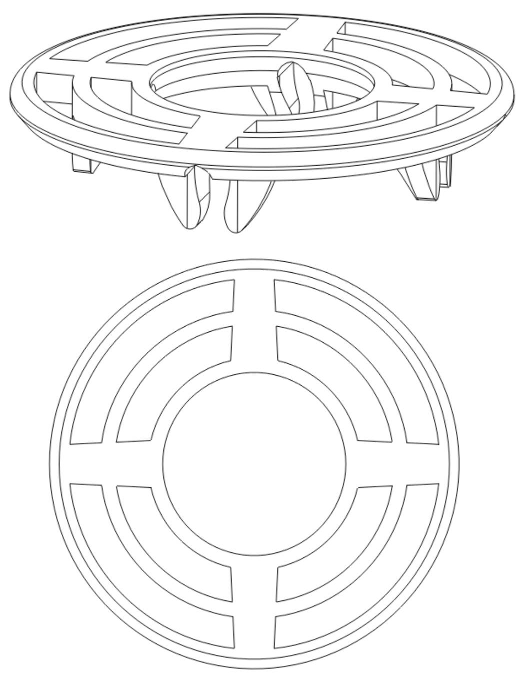
Total budget	$18,900

Bubbles



Thrusters

We have re-used the same thrusters as last year; the BlueRobotics T-100. Our experience is that they provide ample thrust, as well as good reliability. We considered developing our own thrusters, but in the end decided against it on the basis of time constraints and the fact that we already had 6 of 8 thrusters needed.



The thrusters are shrouded with in-house designed thruster guards that were 3D printed using PLA and comply with the IP 21 standard. This was done to reduce the risk of damaging people, surroundings, or getting the tether stuck in the propellers. The thruster guards are easily removable, as they are designed for tool-less mounting. This makes it easy to remove any seaweeds or debris that may find its way into the propellers.

Design Rationale

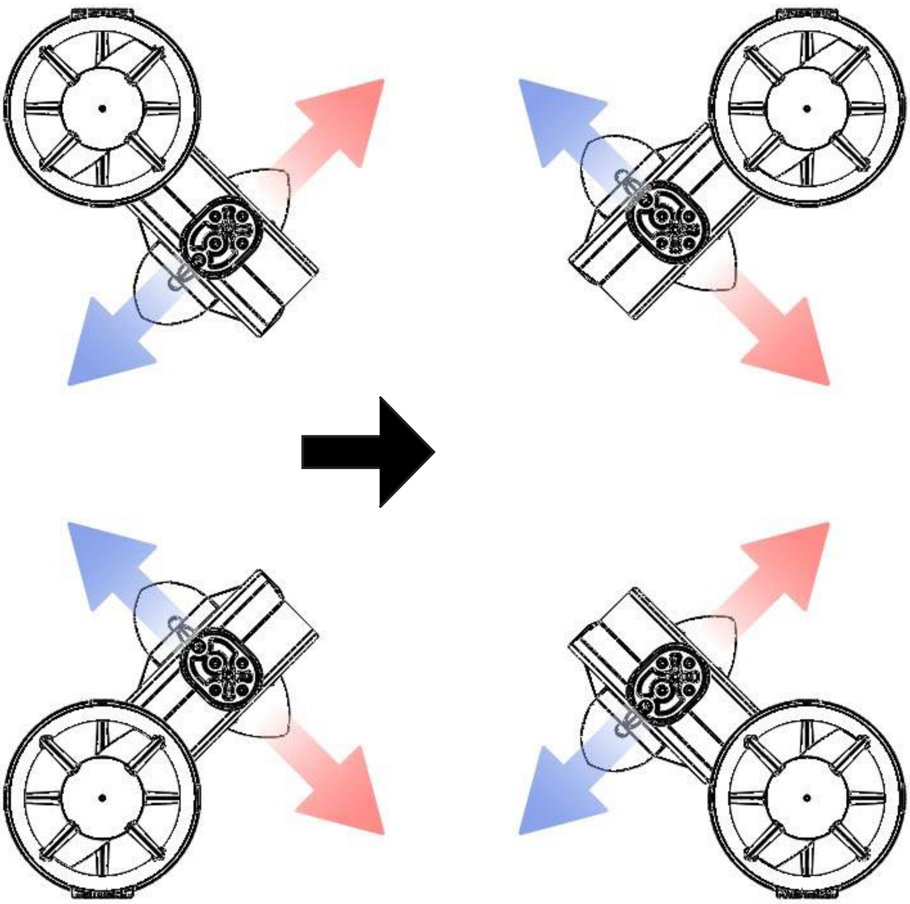
Our overarching goal throughout this project has been to utilize and develop our team members skills to the maximal degree possible. This in turn lead to us focusing on designing, developing, and manufacturing as much of the ROV system in-house as possible.

In the initial planning phase of the project, before the posting of the spec briefing, the whole team sat down to reflect upon our experiences from last year's competition. Since this is the second year KEA Goes Deep participates in the MATE ROV competition we had a lot of new experiences to discuss.

Our mechanical design team has focused on creating designs that we would be able to fabricate in-house utilizing the digital fabrication equipment we have had access to through our schools makerspace. The mentality behind this was to reduce development costs, enable rapid prototyping with short lead-times and last but not least; to increase the learning outcome for the team participants.

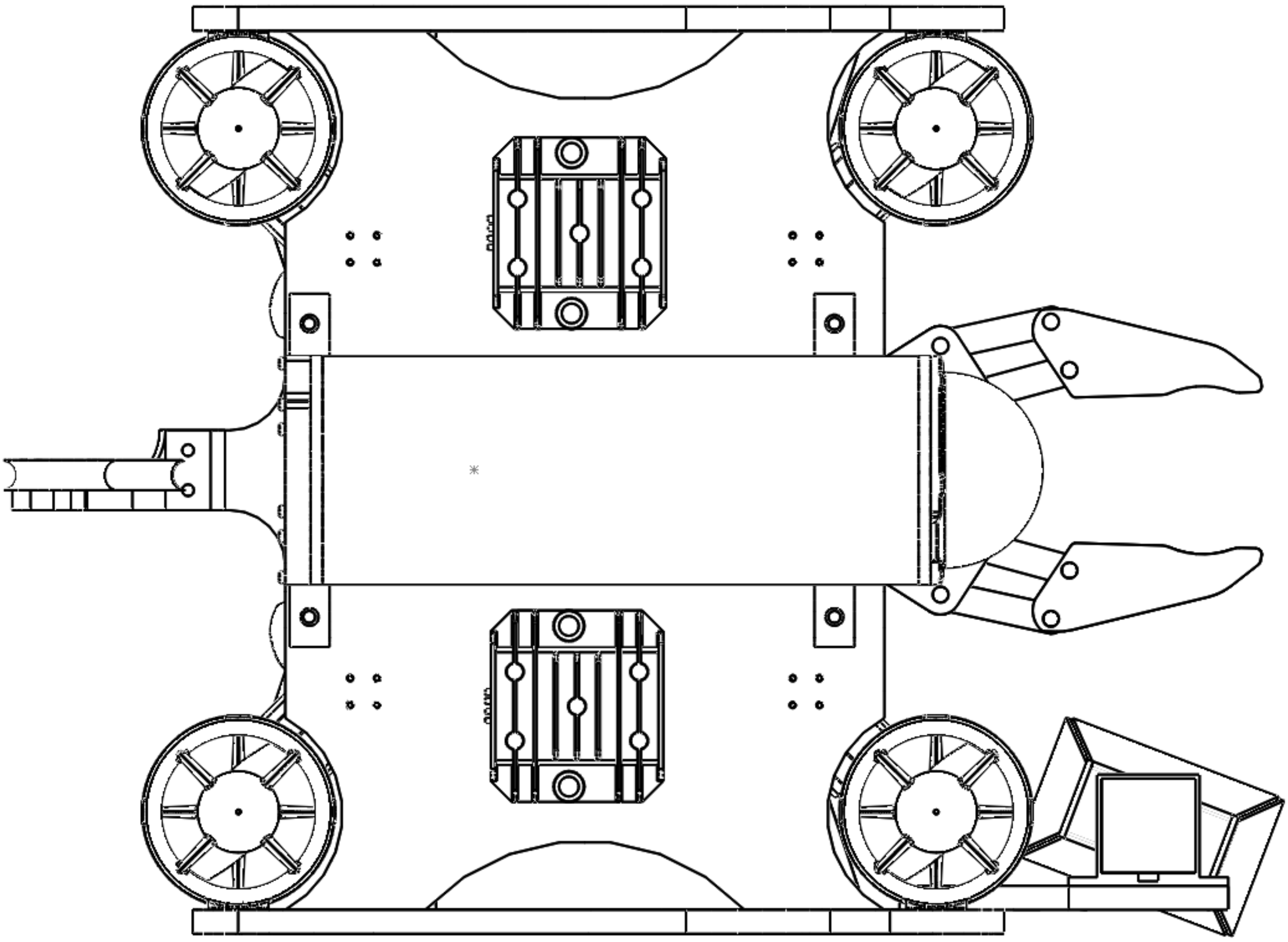
The electronics team picked tasks based not only on experience, but the personal interest of individual members as well. The tools used during the prototyping and development also varied based on the aforementioned. Consistency and proper functioning of developed modules was one of our main priorities. Based on experience from last year, the team focused on developing scalable and flexible modules which can be easily adjusted to lower the time needed to change the code as much as possible, should some complications or unforeseen issues arise.

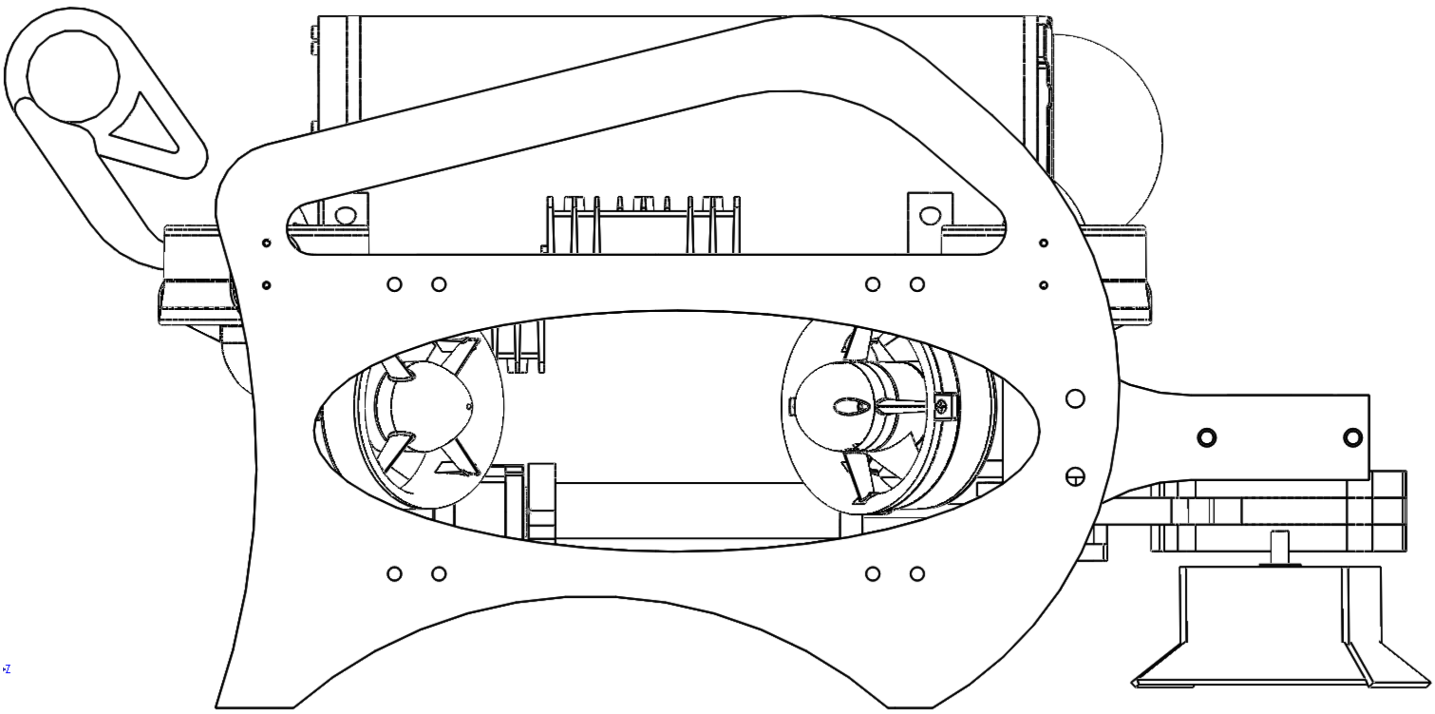
Thruster Layout



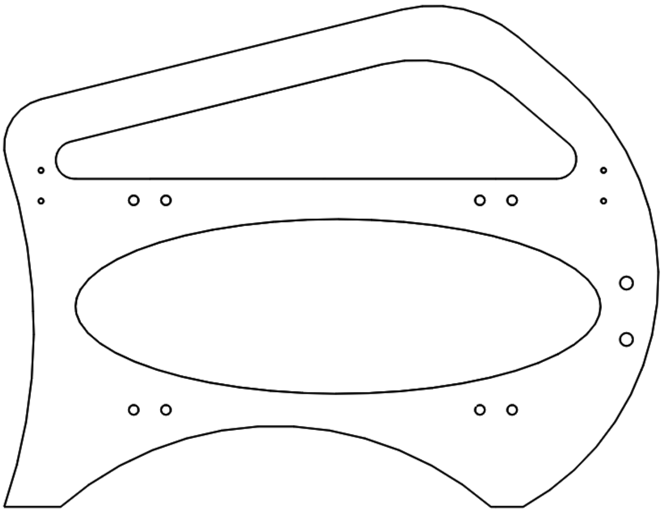
In last year's report, we mentioned several potential improvements we would implement if given enough time. Foremost among these was 3D stabilization of the ROV. To achieve full 3D stabilization we would have to move away from the 6-thruster configuration that was utilized last year in favour of an 8-thruster configuration.
Horizontal thrusters are equally spaced and vectored at 45 degrees.

Frame

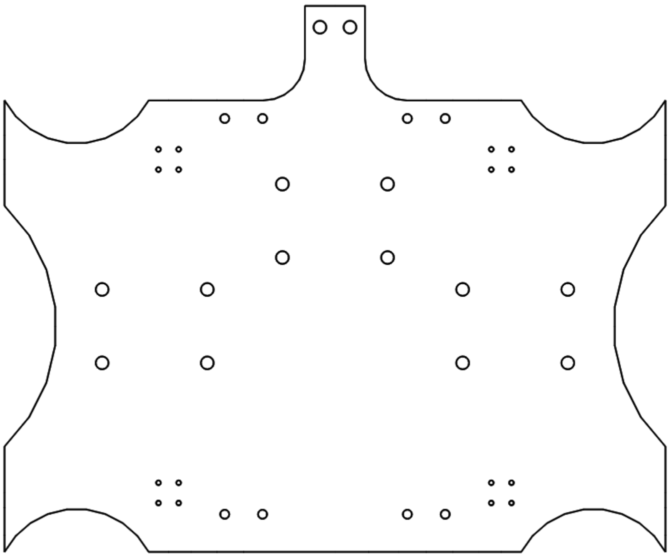




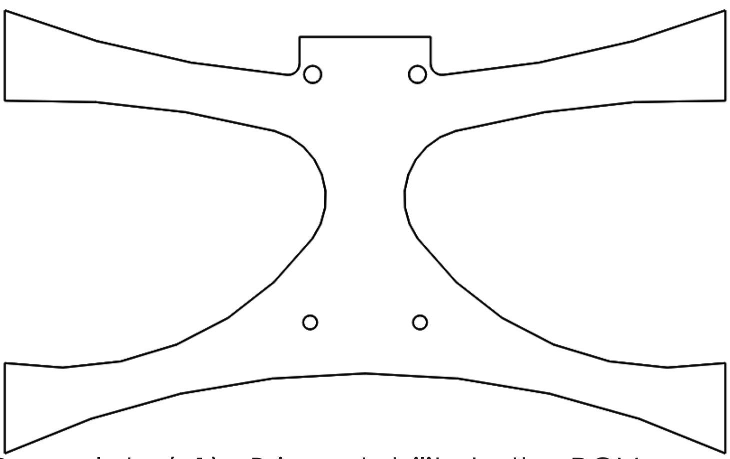
Frame



Side plate (x2) - mounting points for the top and base plates, the vertical thrusters, and the OBS leveler extension.



Top plate (x1) - mounting points for the tether strain relief, the 3 DC-DC converters, the horizontal thrusters, and the enclosure supports.



Base plate (x1) - Brings stability to the ROV assembly, mounting points for the manipulator assembly

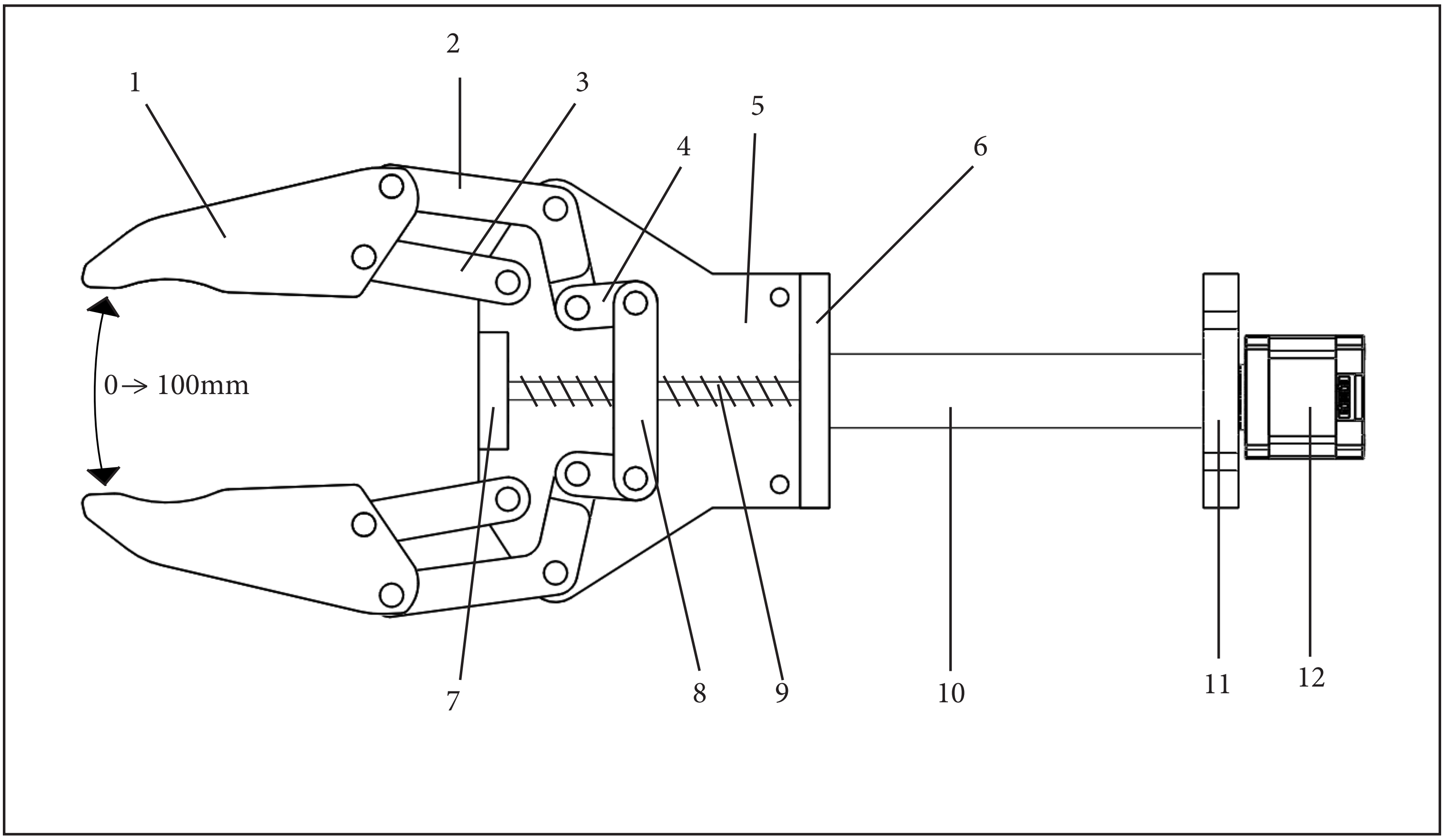
After electing to go for an 8-thruster ROV-design based on last years' experiences and after receiving the physical measurement goals from the spec-briefing, we created a parametric design for our thruster layout in solidworks and designed our frame around these mounting points.
The reasoning behind this was that we did not want to limit our options for tooling. This is also the reason why we opted for a tiered design where we had ample space for manipulators, cameras, and mission specific tools, at the bottom tier of our ROV. The upper tier was reserved for functionally critical components such as our waterproof electronics compartment. We have also strived to keep our weight distribution as central as possible, in order to prevent having to make up for poor component placing with ballast and buoyancy foam.

We chose to construct our frame in polypropylene plastic (PP), as it is excellent for machining, has a low water absorption rate, and has a density close to that of water, which reduces the impact it has on the balance of the ROV, while having a high hensile and impact strength.

As the ROV has been designed to be brought with us on the 8000km trip from Copenhagen to Seattle, being able to disassemble and pack the ROV for shipping has been an important consideration for us. Because of this, we opted to use threaded inserts for our mounting points to decrease the chance of stripping threads in our plastic.

Tools

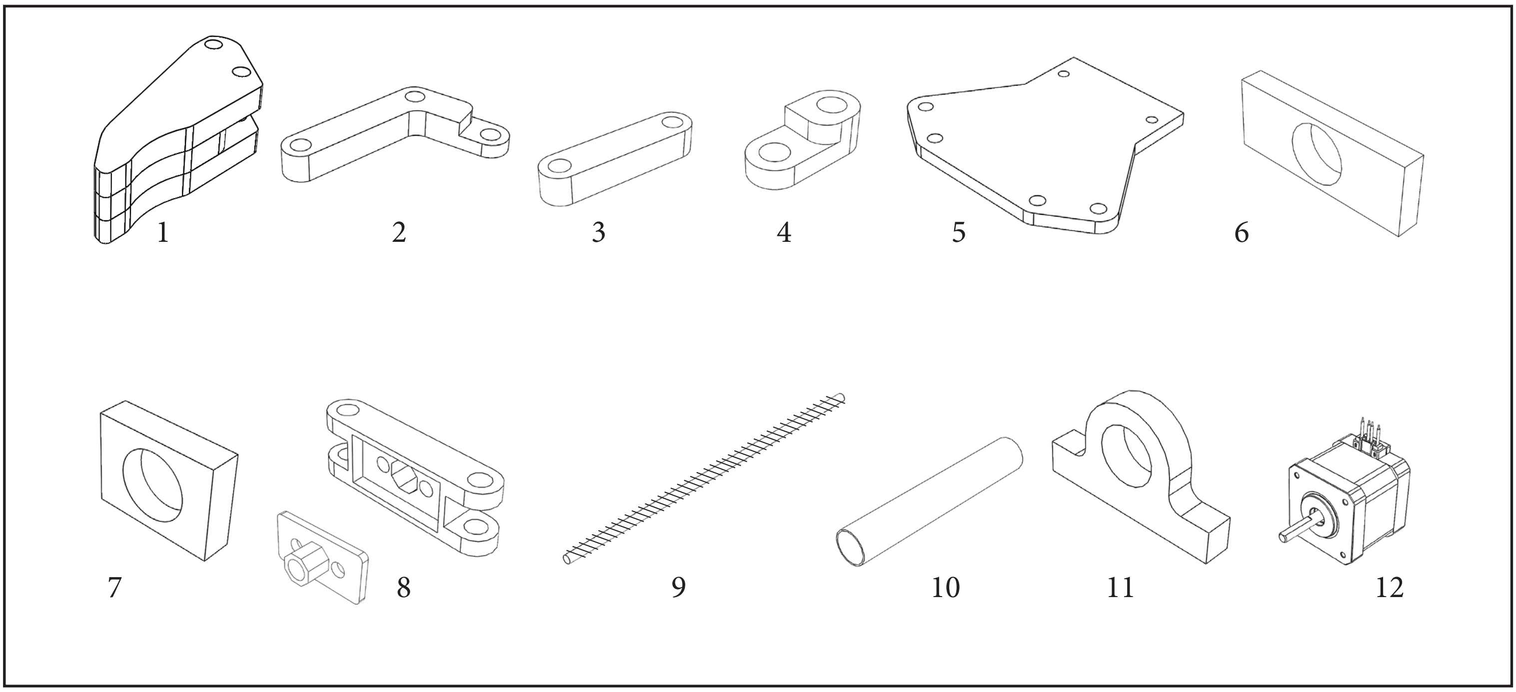
Manipulator



The manupulator is built out of Polypropylene plate (12mm). This material is used for the manipulator and the frame, due to its high ratio of tensile and impact strength to low density, coupled with its 0.03% water absorption coefficient. All the parts made with this material were CNC-machined.

The remaining components are as follows: 20mm ACME rod (9), connected directly to the drive shaft of the NEMA stepper motor(12). The threaded wing-nut is bolted to a 3D-printed part (8), that drives the arms (4,2). The stepper motor is enclosed in a 3D-printed 2-part case, which is waterproofed with epoxy. The 20mm PVC pipe (10), is waterproofed by press-fitting two 20mm ball bearings inside, through which the ACME rod runs, the area between the bearing is filled with marine grease.

Manipulator



This system of converting the rotational motion of the stepper motor's driveshaft, to the linear motion of the arms, using the threaded wingnut, was chosen for its reliability, and the ratio, offering a large amount of torque.

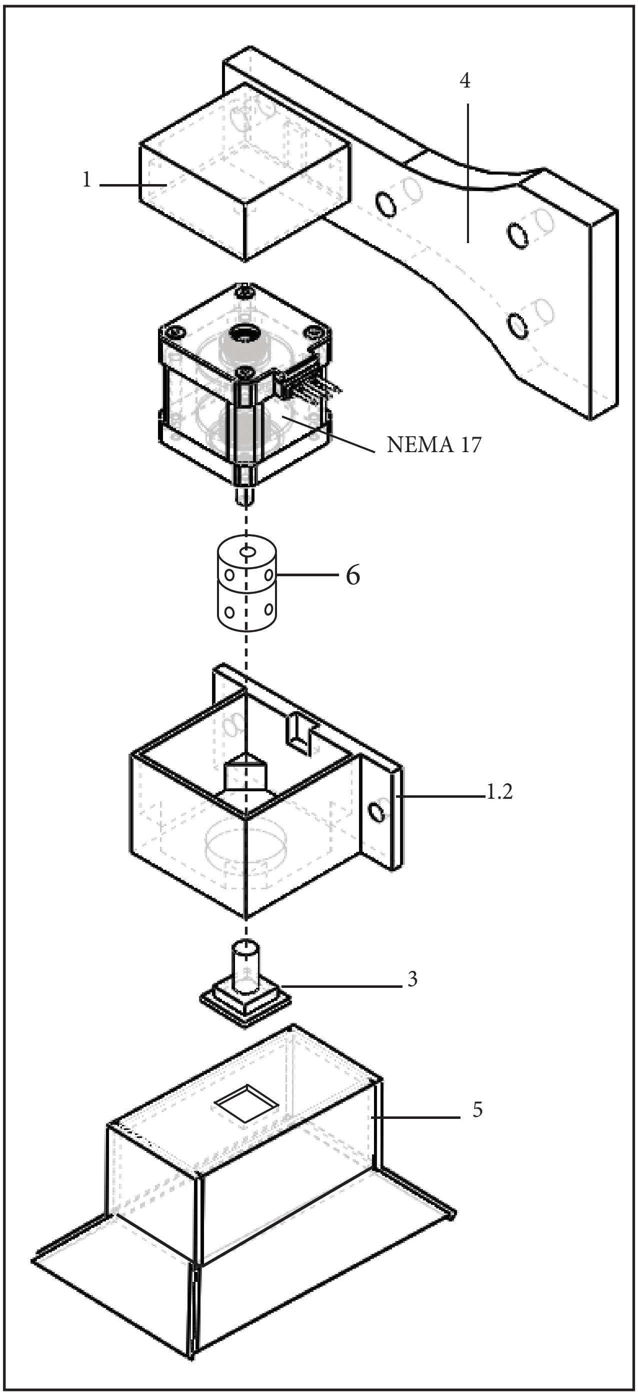
An ACME rod is used for its steep thread pitch, resulting in only 3.5 rotations of the stepper motor needed to move the gripper claws from fully open to fully closed, and vice versa. The rod runs through three 22mm ball bearings - one in the front of the manipulator (7), and two in front of the PVC pipe (10), to keep it centered and to reduce friction.

The axes around which the arms rotate are 8mm 3D-printed rods (ABS). This was a simple way to achieve quick and custom sizes, which smooth surfaces to reduce friction, solid ABS offering high tensile strength.

The manipulator

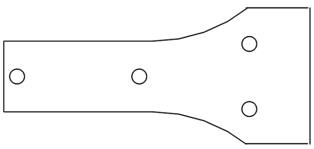
The manipulator assembly is mounted to the bottom plate of the ROV with two bolts in the front (5), and two bolts in the stepper motor enclosure.

OBS Leveler

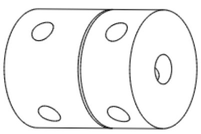


Shown here is the tool developed for leveling the OBS. It consists of an Acrylic interface (5), designed to fit around the T-joints used in the corners of the OBS. The flanged ends mean less precision is required when maneuvering into position above the T-joint, as the ROV will self-center around it when downward thrust is applied.
The interface (5), constructed out of 9 pieces of 3mm acrylic, lasercut, is joined to a connector (3) (3D printed, PLA). The stepper motor collar (6) is fastened to the driveshaft of the NEMA 17 stepper motor, and the shaft of the connector (3), using set screws.
The motor is enclosed in two 3D printed (ABS) enclosure parts (1, 1.2), which is screwed onto the mounting plate (4) (PP, CNC machined). This mounting plate is then screwed onto the frame of the ROV, using quick-release screws for quick mounting and dismounting.

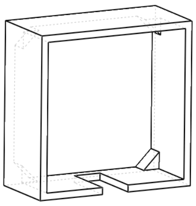
This tool was designed in the idea of having a specialised tool for the task, due to it having elements not found in any other task, namely the rotation around a vertical axis.
As well as being an important task, it is also time consuming, due to the requirement of rotating all 4 corners. Introducing this axis of rotation to the manipulator would have required a number of additional design changes, as well as making the gripper over-engineered for the rest of the tasks.
For these reasons, it was more efficient, as well as more reliable, to create a seperate tool, specifically for this task, which can be removed from the ROV with quick-release screws when needed.



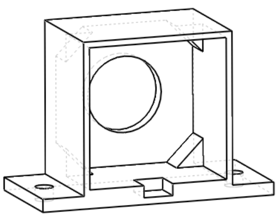
4



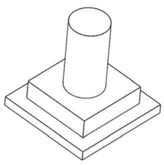
6



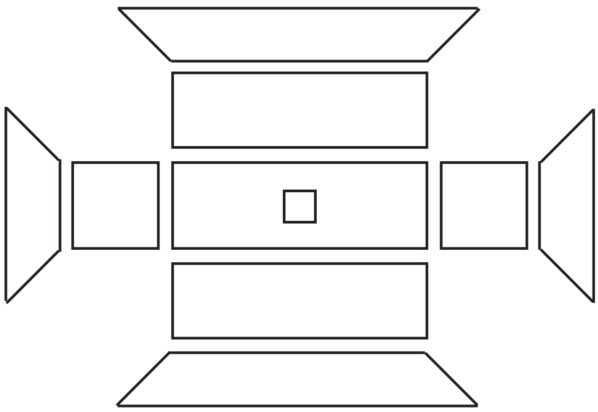
1



1.2



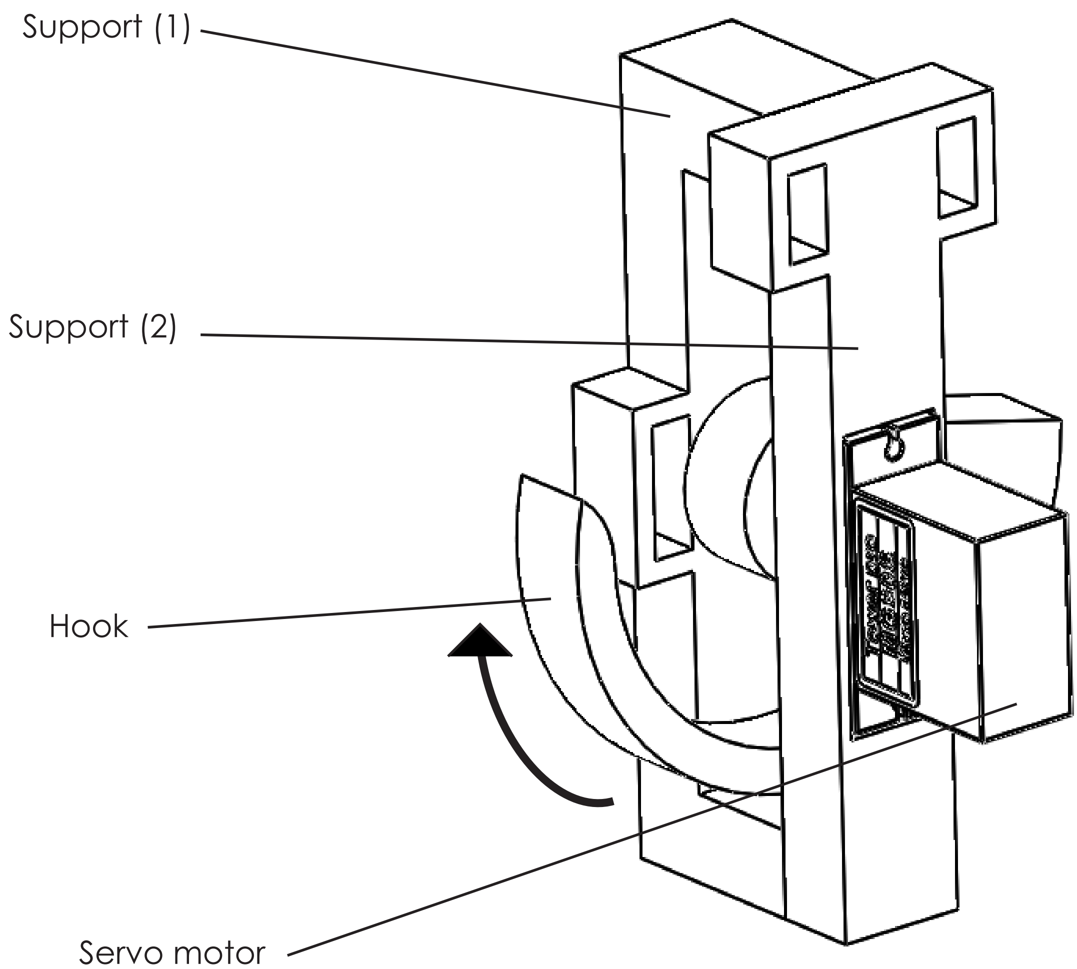
3



5

Liftbag Release System

ESP32 in the ROV sends HTTP request over WiFi to the Liftbag server which is controlling a servo. The only control options are close and open. By default, servo is closed. Both positions are set for hold (servo is powered and holding) and the servo rotates approximately 170 degrees. ESP32 is located inside of a PVC enclosure, servo is attached to the enclosure and has a 3D printed hook-type release mechanism. While holding the enclosure in the gripper, the ROV is manuevered into position. The hook is placed in front of the object to be gripped, and the signal is sent to rotate the servo. The enclosure has a liftbag attached to it which is inflated by an air nozzle at the front of ROV's gripper and the air is being pumped manually from the ground. After lifting the debris and performing the competition task, we release the debris by sending the open request to the server. Then we drag the whole mechanism with the liftbag to the "shore".



Electronics

Ground station

Our ground station has been made in a simplistic way. We are using 3 main components - a laptop, controller and a waterproof case. The laptop model is Lenovo ThinkPad T410 running Ubuntu 16.04. We have decided to use standard Xbox 360 wired controller since we can utilize a number of analog inputs for higher precision.

The software for the ground station is written in C++ using the Qt framework. The choice was based on our needs of having software components that can communicate with each other quickly without slowing down the main thread too much. The Qt framework excels at this, due to its central feature, "signals and slots". This allows us to connect two objects and when a specific event happens in the first object, another method triggers in the second object. These "listeners" run on a single separate thread. Our main goal was to achieve a program with low coupling, high cohesion and high readability.

Electronics Enclosure

For our electronics enclosure we have re-used the same enclosure we had last year. This was done to reduce costs as well as to reduce development time for our limited 3-man mechanical team. We also considered sealing our electronics in epoxy, but decided against this, as having the ability to replace defective components/troubleshoot was crucial to us.

Tether

The tether provides a communication and a power transmission channel between the ROV and the ground station. The tether is 16,5m long and consists of 2 power cables (ground and +48V) with a cross section area of 2mm square , an ethernet cable CAT 5e, an air supply tube and a protective shield. The power cable's cross section area has been calculated based on the requirement of having a voltage drop lower than 10% of the input value. According to our calculations, the voltage drop across the tether is lower than 3.5V, comfortably fulfilling the requirement.

Power Management

One of the Explorer class technical challenges is dealing with the high input voltage, which is 48V nominal and 56V maximum voltage. We chose to use three 30A 12V fully waterproof DC-DC converters connected in parallel, which provide us with a total of 90A and 1080W of power. Since the thrusters don't draw more than 64A together at any point, these converters are more than sufficient for our application.

Inside of the ROV we are using a custom designed power distribution board. This board is the main power point in the ROV. There is a 20mF bulk capacitance, an overvoltage protection, a 5V step down converter and connectors for all devices located on the board. The overvoltage protection protects DC-DC converters and ESCs from voltage spikes and recuperation currents generated by speed controllers.

Control Electronics

Raspberry Pi 3 is the brain of our ROV. It is connected through the ethernet cable to the ground station controls all the devices in the ROV:

- ESP32
- 2 stepper motor drivers
- 8 electronic speed controllers (ESCs)
- USB and CSI cameras
- servo for moving the front camera

Our ROV is using 2 cameras. The 1st is in the front is a CSI Raspberry Pi camera. The tilt of the camera can be changed using a servo. The second camera is USB and it's outside of the ROV.
ESP32 is taking care of communication with wireless sensors - lift bags and OBS.

The ROV's Drive

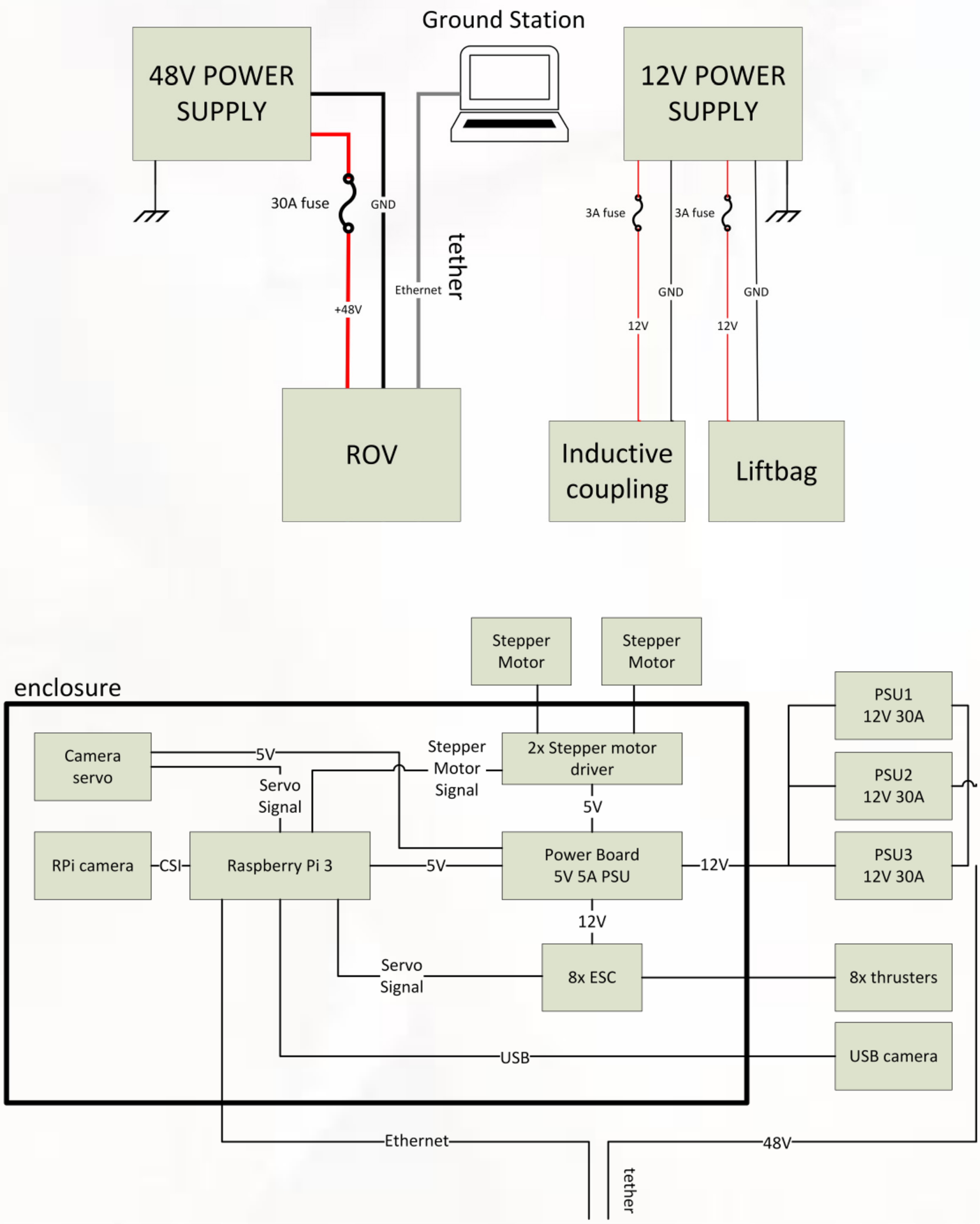
The thrusters of the ROV are controlled by Raspberry Pi, through 8 ESCs Wraith 32 Metal V2 with 60A current rating. We chose these controllers because of their state of the art firmware BLHeli 32 and a high current rating and user definable current limit. Thanks to this feature we are able to limit the current drawn by each thruster to max 8A, 64A in total for all thrusters.

One of the disadvantages of the ESC is that they can create high overvoltage while changing direction or slowing down the thruster. Therefore we implemented an overvoltage protection on our power board, which transforms unwanted voltage over 13,5V into heat.

Gripper and OBS leveler control

The gripper and OBS leveler both use NEMA17 stepper motors, which are controlled by Raspberry Pi using stepper motor drivers A3967SLB. These drivers have adjustable current limit, which is set to 500mA. We chose them because of their simplicity and low cost. Their biggest disadvantage is a substantial power loss and outdated design, but in this case, the positives outweighed the negatives.

System Interconnect Diagram



ROVs theoretical max current draw is 20,375A. Fuse is 20,375A * 150% = 30,563A, therefore 30A fuse is chosen on the input of the ROV. For detailed calculations please refer to Company Safety Review.

Software

The ESP32 chip inside of the ROV is responsible for WiFi communication with Ocean Bottom Seismograph and the liftbag release system.
We are using FreeRTOS which allows us to focus on application development rather than resource management. FreeRTOS creates an illusion of multiple tasks running at once and we can achieve better flow and understanding of the entire application.
The application is developed in Arduino IDE 1.8.5. After handling and receiving messages from Raspberry Pi with use of checksum, the application sends HTTP request to either Liftbag release server or OBS server. Liftbag release is specified further in a separate file. The application uses the same client module to communicate with both liftbag and seismograph. The data from OBS is sent to Raspberry Pi over a serial port.

For communication between the ROV and the Ground Station we chose TCP as it is the most reliable way of transmitting data between a Server and Client. The ROV is connected to a server (Ground Station) and is awaiting messages. After receiving a 16 bytes message, this message will be further reduced to a Key and Value. We first intended to store these values in a dictionary to always have access to default values, but to reduce the amount of memory used we gave up on this idea and switched to using PyDispatcher to dispatch signals with the corresponding Signal(Key) and attribute(Value) to the desired methods for further usage. The basic protocol for the received message in design in a simple pattern that's easily understood and used, the pattern consists of:

NameOfTheRequiredAttribute	+ = +	TheValueItself	+	Underscores_untill_16_ bytes_reached

The transmission is mostly used for PC to ROV data-flow and ROV to PC video-stream, but we also send OBS leveler info back from the ROV. For receiving OBS data we opened a serial port on designated Pi GPIO and simply resend the data to a TCP port.

To make the code cleaner and reusable we created libraries to support each feature. Communication and transport between libraries was done by using PyDispatcher.

```
dispatcher.send(signal=tList[0], sender="TCP", value=tList[1])
```

Safety

The safety of our members is our foremost consideration. Since the beginning of this project we highlighted the necessity of a clear protocol to respect during different steps. In particular during the design process we focused our attention on limiting overheating problems that can cause damage to the ROV and hurt the people close to it. Every connection between the ROV and the groundstation has been designed and realized putting much attention on the reliability of the materials we used.
One of the important step into the production process was when we selected the chemicals used to make the ROV waterproof such as the acrylic enclosure of the electronic hardware components.

During the production phase we ensured that each component of the team was trained properly respect to the task he had. Miroslav Lakota was responsible for checking periodically the safety of the tools we used and that all the necessary safety tools (gloves, glasses etc.) where available into the lab.

In order to be safe, we established a set of rules for being into the MakerLab. Members of the team were required to wear long pants, closed shoes and pull up their hair when they were working into the lab.

Future Improvements

There is always room for improvements in the design of our ROV, code, mechanical parts and the development process. In the future, the team would like to tweak WiFi communication to achieve higher connection stability, lower delays when fulfilling certain tasks and fix minor bugs that are destabilizing the application mostly on ROV ESP32. This would be done by using a different method of connecting.
Communication protocols used, while working well and sufficient for the use case, need to be more robust, scalable and error-proof. This will allow us to structure our messages in new ways.
Side camera position probably needs to be changed/better thought through. We can improve on field of vision, color correction, image quality and frame-rate of received video.
Mechanical team brought up certain ideas to enhance the gripper, especially gripping strength. One of the solutions would be to gear the stepper motor used to open and close the gripper in more optimal way, possibly sacrificing some of the speed for more torque.
While we had a really well-planned and sensible schedule, the whole team felt like we were falling behind, which caused unnecessary stress and even some compromises. To avoid this issue during possible future development, we would like to specify tasks in even more detail and to dynamically redistribute workforce for the tasks that take longer to finish than initially anticipated.
Another improvement we have considered would be to change our vectored thruster layout to maximize for speed in the forward and reverse direction. At present, our horizontal thrusters give equal thrust laterally, which might in some circumstances be beneficial, but through testing we have discovered that strayfing is used significantly less than movement forward and reverse.

Acknowledgements

Our success would not be possible without he help of others. We would like to express our gratitude to all those organisations and persons who contributed in any way - small or large.

Thank you

-KEA Goes Deep