Blue Nemo: PWr Diving Crew's AUV Technical Design Report (June 2022)

Emilia Szymańska (PM), Marcel Skrok, Damian Kociołek, Aleksandra Komorowska, Jakub Papież,
Igor Zieliński, Krystian Matusiak, Maja Bielawska, Radosław Tomzik, Michał Purzycki, Artur Stopa, Piotr Szlęg,
Wroclaw University of Science and Technology

***Abstract*—Blue Nemo is the currently developed AUV being a part of PWr Diving Crew project. Improving its features with each vehicle iteration, this year's solution is an implementation of three main aspects of each department work: research, workflow improvement and testing procedure development with a stress on modularity. We believe that with relying on two computer vision approaches fusion and introducing new elements such as a gripper, torpedoes and a marker dropper, we have a high chance to successfully complete all the tasks.**

I. Competition Strategy

We want our robot (Fig. 1) to mostly rely on computer vision – having three cameras (front, bottom and on the gripper) opens doors to various analysis. Our vision is based on two approaches – a fusion of classical image processing methods and neural network output. We believe that such system is capable of a better performance and is more fail-safe, therefore, our strategy is based on this solution.

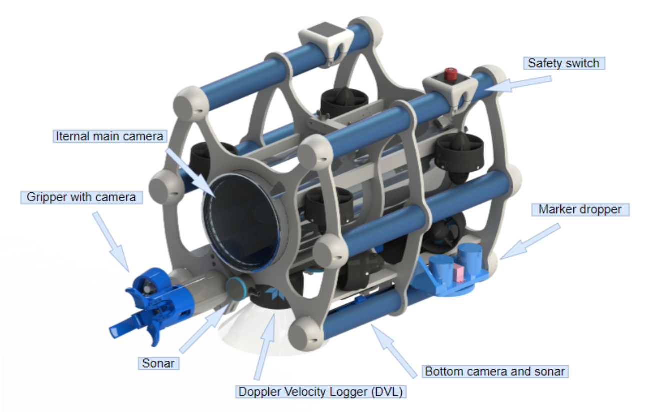

Fig. 1. Blue Nemo design.

Another key to our predicted success is the PID controller. After receiving data from Doppler Velocity Log and Attitude and Heading Reference System [1] sensors, it keeps the set speed and depth of the robot in alternating water environment.

However, we did not want to choose the side at the moment of writing this report – we believe it would limit our options if it turned out that our detection algorithms work better with a certain side in various conditions. Right now, we still have over a month full of testing and possible improvements and we want to take advantage of this fact by staying open to changes.

This year, when preparing for the offline competition, we focused on three main aspects of each team's work:

- **research** – exploring new solutions in both hardware and software and sharing them with professional researchers in a form of conference papers,
- **workflow improvement** – using GitLab, GrabCAD and other forms of work enhancements,
- **testing procedure development** – introducing **modularity** in electronic, mechanical and software aspects, developing solutions for robot status real-time supervision.

A. *With Moxy & Choose Your Side*

These two tasks were very similar in implementation. All gate detections are possible due to convolutional neural network utilization on images received from the front camera. The custom-made training dataset was prepared in such a way that apart from the simple label *gate/no_gate*, the center of gates and bounding boxes were assigned to each image. It is very helpful for further control algorithms [2] – we wanted to make sure that the robot will not collide with the gate, but with some extra distance from the poles it will navigate through the center.

In *With Moxy*, we plan to score additional points for requesting a coin flip to determine the heading of our AUV at the start. Robot's holonomic properties allow for rotation in place without unnecessary translation. Therefore – right after submerging – the AUV will keep turning around until the gate is detected.

As for the images in all competition tasks (e.g. G-Man/Bootlegger figures in *Choose Your Side*), the distinction between sides is made based on the feature matching and color ratios on each quadrilateral. Considering *Choose Your Side*, apart from the similar procedure as in the previous task, we want to get extra multipliers for rotation in the Z-axis while passing through the gate. It is the most effective tactic for our robot as its center of gravity is not at its geometric center, but below it. This significantly increases the stability of our vehicle in water, but makes it difficult to perform a rotation around horizontal axes, as gravity counteracts these rotations.

B. *Path*

Although there are no points for following the path, we want to leverage this opportunity for in-between-tasks navigation [3]. We plan to use our bottom camera's output in combined detections from a hand-written OpenCV algorithm and a pre-trained neural network. This makes the whole decision-making algorithm more fail-safe, especially if one of the methods turns out to be faulty due to on-site conditions.

C. Make The Grade

According to the previous side selection, the robot will approach the appropriate buoy using the front camera. The targets themselves will be found by edge detection due to their rectangular shape.

D. Collecting

The bin itself will be found with the help of color detection – we focus on white and purple color localization in the front camera output. Our standard approach to bin's images identification will be applied. The initial idea was to bump into the bin's cover to drop it, but it could make the entire basket flip with everything inside it. Therefore, we decided to apply a more gentle solution -— utilizing a gripper (Sect. II-A5). Its three-fingered design ensures traction and grip to a variety of items with different shapes.

Marker dropper's design was quite a challenge, with a detailed explanation in Sect. II-A6. We wanted to minimize the number of moving elements underwater, so we applied one servomechanism to drop both markers. Bottom camera is used to ensure that the dropping is performed over the correct bin.

E. Survive The Shootout

The shooting target is found in two steps. First, we detect and approach the boards using neural networks and – when close enough – the vision is switched to color detection algorithm that is both faster and more dependable in this situation. As images have specific sizes, we can calculate the actual distance to them. It allows us to fire a torpedo at a properly close distance so it does not have to swim too long and jeopardize the mission.

There are two torpedoes attached to Blue Nemo. Initially, they were supposed to be launched using an electromagnet and reed switch, but this solution demanded too much space. Therefore, we are going to trigger them with a light signal. This type of launching allows for the simplest and the most resistant design, further described in Sect. II-A4.

F. Cash or Smash

The last task consists of two main aspects: approaching the acoustic pinger and moving the bottles. Analyzing the data received from two hydrophones [4] mounted on Blue Nemo makes it possible to localize and navigate to the sound source. Data analysis is based on feeding neural network with both acoustic signals.

Localizing the bottles is as simple as color detection, with a neural network solution as a backup. To move the bottles from one table to another, we will use the gripper (already mentioned in *Collecting* task). The main camera is going to lead us to an object, then by using another one placed in the gripper and utilizing a reflective sensor the robot will grab and place the object on the other table. Recognizing the proper table is going to be done with the camera placed under our vehicle.

II. Design Creativity

A. Mechanics

1) Modularity in Mechanical Design: This year, we focused on ensuring the greatest possible flexibility, as well as convenience in component assembly and disassembly. Electronic components were mounted on a specially designed frame. Its purpose was to fix the components in such a way that they did not interfere with each other and that there was satisfactory access to them during tests performed at the pool to facilitate possible repairs.

In the currently used version, the parts are attached to flat bases, which are then successively inserted into the previously prepared undercuts and then locked. Special holes were also made to match the shape of the mounting nuts to facilitate the assembly. The geometry of the printed stands is designed to allow the ventilation of individual electronic components. The above-described improvements make it much easier to quickly and conveniently rearrange components and prevent them from moving in all planes. The current solution is presented in Fig. 2.

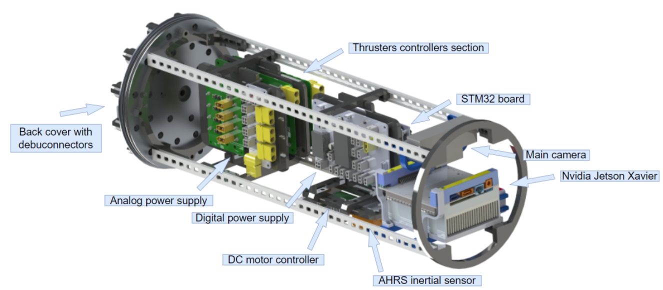

Fig. 2. Internal frame design.

The vehicle's microcontrollers have to be connected to its external parts (e.g. thrusters, sensors, batteries) with wires. So far, we have used ready-bought glands, but we needed a better solution as robot repairs required removing every cable from the back cover, which was time-consuming and caused cable twisting. None of the available products met our expectations. Therefore, we designed *Debu-connectors*. We used Weipu cable connectors, latches instead of threads and a custom housing, which was compatible with the robot's main body. *Debu-connectors* ensure the required tightness as well as a possibility to unplug the cables on demand to perform necessary servicing.

2) Workflow Improvement: We have improved the information flow and archivization of our technical documentation by introducing the GrabCAD platform, which solved problems with file management. Now our work is much more efficient, we are able to stores previous and current models' versions, which allows us to compare solutions. What is more, we can open the whole model and update it online.

We also decided to expand the cooperation with our software SolidWorks provider. DPS Software company organized workshops in the field of SolidWorks modeling and analysis only for our members, which allowed us to unify our modeling standards and improve the quality of the designs.

3) Research: We performed a comparative analysis of the underwater robot design steps throughout the PWr Diving Crew project, where simplified Remotely Operated Vehicles were improved to AUVs with enhanced environment perception and capability of autonomous task performance. We formed a conference paper [5] on robots' characteristics and the altered approach to the problem in four successive vehicle generations. The most important part of this research were the conclusions on the structure optimization and we applied this knowledge to improve the current design.

4) Torpedo: Torpedoes' design (Fig. 3) allowed for high repeatability and ease of implementation. After several test iterations, the size of the torpedo's outer diameter was set to 33 mm. The stabilizers were added to stabilize the trajectory, the propeller shape was redesigned according to the flow analysis "Flow simulation" performed in SolidWorks software. Based on the data obtained in SolidWorks, the optimal shape for neutral or slightly positive buoyancy was calculated. The final buoyancy of the torpedo depends on the load, which can be reconfigured according to technical needs.

The triggering system of a torpedo is equipped with the time mechanism set on the timer 555 system which causes launched torpedoes to operate for a maximum of 10 seconds. The circuit is activated via a light emitter inside the torpedo holder inside the drone.

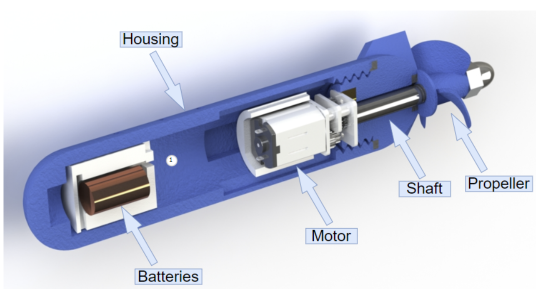

Fig. 3. Torpedo render.

5) Gripper: The main operating aspect of the gripper (Fig. 4) is a screw rotated by an electric motor which moves the nut up and down the screw. The nut is connected with fingers in a way that the movement of the nut rotates the fingers, thus allowing it to grab the object. Due to the fact that the gripping part of the gripper is 3D-printed, we can easily adjust the geometry of the fingers as well as their number to better suit the geometry of the object that is to be moved.

The front camera turned out to be insufficient – the robot kept losing sight of the objects to grab. Therefore, we made some space for an additional camera and we put a laser distance sensor on the axis of the gripper. The combination of the camera and the sensor allows us to achieve desired control to perform the gripping action.

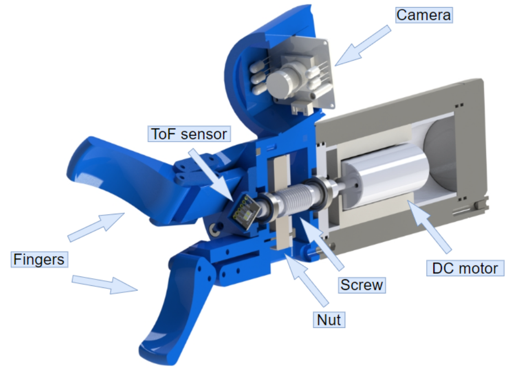

Fig. 4. Gripper design.

The gripper with its own PCB has an integrated DC motor H-bridge driver which, combined with the use of magnetic encoders, allows for precise control of motor speed corresponding to closing/opening speed. The board provides motor current measurement used as additional information for proper actuation as well as a method for detecting if the object was properly gripped and as a safety measure for detecting motor malfunction.

6) Marker dropper: We decided to replace previously used coil with a servo motor. The marker's drop design consists of two tubes (closed on the side, on which the markers are placed) and a cam mounted on the servo shaft, which blocks the markers. As the cam rotates, one of the markers is released from the tube.

B. Electronics

1) Modularity in Terms of Processing Units: We created three additional PCBs, each with a dedicated microcontroller: Lights Controller, Gripper Controller and Hydrophones Data Processor. Their main task is to relieve the computational burden of NVIDIA Jetson AGTX Xavier and Main Controller board. Excluding the functionalities to separate boards, forming a distributed system, allows for parallel execution of tasks. What is more, software development, debugging and testing become easier as a single module is responsible for a smaller set of functionalities.

2) Workflow Improvement: Embedded software and PCB projects are now stored in GitLab repositories. Thanks to the version control system, it is possible to facilitate cooperation and synchronization between team members. Moreover, each project has its own README.md file, which accurately describes requirements and necessary information. To further facilitate the workflow, we created our own library for KiCAD components.

Our self-configurated environment allows for an easier and wider selection of used toolchains and helps to keep the configuration of the development environment as well as toolchain versioning compatible between different machines. Visual Studio Code, being newer and based on Electron, is also significantly faster, more stable, easily configurable and gives huge possibilities for modification in comparison to typically used CubeIDE and can also be used to work with languages other than C/C++.

3) Research: One of the problems that can be encountered in an underwater environment is limited visibility. Average cameras, even when equipped with additional lighting systems, do not allow for data collection from a greater distance. However, a different kind of imaging can be used. Based on the laser triangulation approach, a three-dimensional model of the surroundings can be created and subsequently analyzed. Therefore, to look for future camera replacements, electronic team members did a research in the design of a laser triangulator and successfully presented their solution in a conference paper [6] at MIPRO 2022 convention.

4) CAN: To standardize communications and avoid various sources of interference, CAN (Controller Area Network) has been applied. This reliable interface connects the main onboard device, namely NVIDIA Jetson AGX Xavier and several modules based on STM32 responsible for tasks on data acquisition and system control. The whole system works according to Master-Slave model, where Xavier as a major device communicates with other devices through polling. It sends information about the device ID and the task that is to be executed. Examples of such functions are thrusters control, gripper manipulation, lights control, sensor data gathering and torpedoes triggering.

C. Software

1) Modularity in Software Architecture: We aimed for the controller code, namely PySA (Python Submarine Autonomy), to be maximally modular and immune to potential programming errors or unexpected events (e.g. crashes). The key concept is to follow Domain-Specific Languages. Inside Python syntax, we define a smaller declarative language with its own special semantics and operators which is then evaluated by a virtual machine – its state can be displayed on the screen, modified during runtime, and saved or preloaded from a file which allows for simpler diagnostics and error handling, completely separated from the task code itself. The language defines a lot of popular tasks such as repeating tasks, fitting in a given time span or trying different solutions. A key element of our software development is also the simplification and unification of the communication between Xavier and STM units as well as making it resistant to information loss by using the circuit redundancy check algorithm.

2) Workflow Improvement: Our improvements reduced the debugging time, assure code quality and speed up testing. Every module, for both physical devices (e.g. AHRS, DVL) and abstract ones (e.g. ColorDetector or ML solution) is optional. If any dependency is not installed or a physical device is not connected, PySA still works and just "ignores" the module. This is not only a very flexible solution, as we do not need to connect all modules every time or change our code to run the autonomy – we can test our code on a PC (e.g. CV code using only an USB camera) without any problems. Compared to the previous versions of the systems, the automatic tests count increased exponentially. There are end-to-end tests and unit tests for utility and base classes as well as different devices and algorithms. Vision algorithms are tested using labeled sample images. GitLab CI/CD pipelines system is used to test new code with pylint and black formatter in the cloud using Docker. This is possible due to a very modular design of the system and the ability to mock almost all of the external devices. Git tags are used to mark well-tested code, allowing for stable system installation in matter of minutes. Every merge request is reviewed by two peers, one of them being a "senior" who has rights to accept the new code into the repository.

3) Research: The benefits of using a simulated environment for running software tests, training neural networks, and testing the AUV's performance in arbitrary scenarios were explored by our programmers. By utilizing Unity3D, a customized framework was developed, allowing for setting up environments for simulating various aspects of underwater operation. This framework supports collecting observations through simulated sensors and controlling simulated actuators, gathering datasets for offline training, and randomizing parts of the environment to test the system's robustness. All technical data and conclusions on the simulation were formed into a paper [7], presented at IAS-17 conference.

What is more, we are also preparing a paper on reinforcement learning approach for AUV training. We have created our own reward system and used the simulator as environment. We failed to finish our research in time for the competition but the experimentation with reinforcement learning contributed to the development of the simulator in the field of collecting data and error handling.

4) PID controller: We implemented three PID controllers responsible for height, speed and rotation in three axes, running on a separate thread using our own multithreading object sharing system. Their parameters were chosen experimentally after various test iterations. Introducing controllers allows for maintaining the desired movement properties despite the environment's interference.

5) Computer Vision: We combine two approaches to Computer Vision: classical image processing and neural network analysis. They complement each other in different tasks. It is also possible to merge two detections that together are a new, higher-order object, e.g. two poles make a gate. Color detection based on mask application, edges detection with Hough Line Transform, feature matching or color ratios are just some of the algorithms from OpenCV library that are helpful in robot tasks execution. As for machine learning [8], we use supervised learning methods. Most of the utilized neural networks have convolution layers and one of them is based on a pre-trained YOLOv5 model.

To improve gate detection efficiency we decided to build our own NN model. The model consists of two separate convolution models – one that checks if the gate is visible and one that will locate the gate. By combining the results of both models we will be able to know where the gate is located if it is visible. The model that we implemented is replaceable with YOLOv5 [9], because both models produce gate location as an output.

Image analysis is performed by NVIDIA Jetson AGX Xavier so its processing power limitations had to be taken into account. YOLOv5 project offers a few different sizes of the prediction model, namely: small, medium, large and extra large. Each one slightly differs in architecture and has different performance. In the end though, a small model has been chosen as it needs to be evaluated in real time and it was the only reliable option able to meet performance expectations with data from a 30 FPS camera producing image analyzed in 480x480 resolution.

The dataset for artificial intelligence algorithms is obtained by recording submerged physical elements (e.g. gates) created accordingly to the handbook instructions. After that, the recordings are uploaded to a private labeling server with a Computer Vision Annotation Tool on it. The data is labeled manually by every member of the team. To aid learning and improve the training efficiency, we also decided to implement scripts for data augmentation and train test splits.

If the number of gathered images is too small, we complement them with an automatically-generated dataset from our AUV simulation environment. Gates, boards with images, and other elements were added to the simulator, and then – as Unity3D is omniscient and knows the location and type of the props – the dataset can be easily created.

Additionally, we are in the middle of implementing an auto-labeler for real-life images to eliminate the need for manual labeling by team members.

III. Experimental Results

Testing our solutions was possible in three different ways. The performance of control algorithms and computer vision was checked in our custom simulator (already mentioned in Sect. II-C3), single devices' operation was examined in university's rowing pool and the whole robot was tested on a local swimming pool. Swimming pool tests were performed every weekend, with elements such as gates, bins, boards with images etc. submerged in the water.

Different aspects were investigated each week. We managed to create a robot, whose design offers good streamline properties, slightly positive buoyancy and proper tightness. The new gripper, torpedo launcher and marker dropper work well underwater and allow for task executions.

The electronic team successfully finished the implementation of CAN interface. Additionally, new PCBs were created for external components such as hydrophones, lights and gripper.

Our software developers created a new system architecture for well-handled task management, sensoric data acquisition and external devices control. Most importantly, various neural networks and classical image processing algorithms were implemented to execute tasks with computer vision.

A. Testing Improvements

1) Testing procedure: There were several situations, in which some team members worked on Blue Nemo parts right before the tests and on the swimming it turned out that the robot is not working. To prevent such situations, each time a group of three representants (one of each technical team) is formed and two hours before leaving for tests they perform inspections of both software and hardware in accordance to our checklist. Only then, when we were sure the robot is operating, the vehicle is taken to the swimming pool.

2) Status indicator: In order to improve the debugging process during pool tests, we added an RGB LED, connected directly to the Xavier, with the purpose of AUV status indication. It lights with different colors corresponding to various robot's software states. The main functionalities include informing about the correct work of the main autonomy program loop, and listing detected objects – specific object types are paired with predefined colors. This form of feedback is especially helpful when wireless operation is tested.

3) Graphical User Interface: GUI (Fig. 5) allows us for following robots' sensoric data and controlling the robot in remote operation mode. Currently implemented functions are: observation of the camera output, checking the sensors values, enabling or disabling the autonomous mode and operating the vehicle with a gamepad.

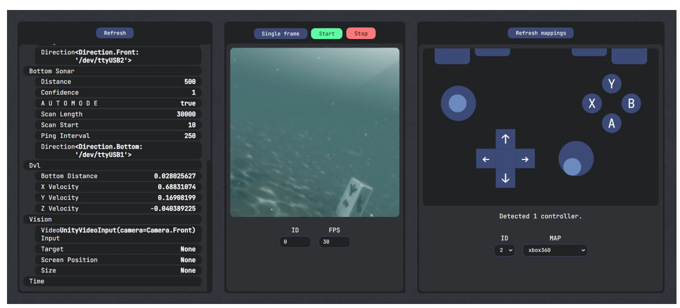

Fig. 5. Custom Graphical User Interface.

4) Modularity: Each of the team performed modularity application – mechanical (Sect. II-A1), electronic (Sect. II-B1) and software (Sect. II-C1) aspects contributed to the easiness of servicing and debugging robot's behavior on swimming pool tests.

Acknowledgments

PWr Diving Crew would like to thank all those who were involved in the creation of this project, most importantly the members of the entire team and our parent unit, Wroclaw University of Science and Technology. It would also be impossible to complete the project without the help of our partners, that is Bosard, Mitutoyo Polska, 3DConnexion, Alox, DPS Software, TME, Bossard, Manus, Botland, Blue Robotics, Wólczanka, Skład Techniczny, AWF Wrocław, and Manus Foundation. We would also like to thank our team's mentors: PhD Jacek Ziemba, PhD Łukasz Dworzak, PhD Marek Kuran for their commitment and shared experience.

References

[1] C. H.-s. . Y. J.-S. Kim, Ho-Sung, “Study on ahrs sensor for unmanned underwater vehicle,” International Journal of Ocean System Engineering, 1(3), 165–170, 2011.

[2] R. Tedrake, “Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832).” Downloaded on 25.03.2022 from http://underactuated.mit.edu/.

[3] Y. Z. P. A. Miller, J. A. Farrell and V. Djapic, “Autonomous underwater vehicle navigation,” IEEE Journal of Oceanic Engineering, vol. 35, no. 3, pp. 663-678, 2010.

[4] M. R. H. Colin A Simpfendorfer and R. E. Hueter, “Estimation of short-term centers of activity from an array of omnidirectional hydrophones and its use in studying animal movements,” Canadian Journal of Fisheries and Aquatic Sciences, 59(1): 23-32.

[5] M. Purzycki, A. Komorowska, A. Ilnicka, J. Papież, E. Szymańska, "From ROVs to AUVs – Optimization and Analysis of Underwater Vehicles Design," proceedings of MIPRO 2022 convention, Opatija, Croatia, May 2022.

[6] K. Kacperowski, E. Szymańska, M. Purzycki, M. Skrok, M. Pietrzak, "Optimized Laser Triangulator for Underwater Robot Vision," proceedings of MIPRO 2022 convention, Opatija, Croatia, May 2022.

[7] P. Szlęg, P. Barczyk, B. Maruszczak, Sz. Zieliński, and E. Szymańska, "Simulation Environment for Underwater Vehicles Testing and Training in Unity3D," proceedings of Conference on Intelligent Autonomous Systems 17, Zagreb, Croatia, June 2022.

[8] D. Brzoza, “The use of machine learning in the process of programming an autonomous underwater vehicle.” Wrocław 2019.

[9] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525, 2017.

Appendix A
Component Specifications

Component	Vendor	Model/Type	Specs	Custom/Purchased	Cost	Year of Purchase
Buoyancy Control	Designed In-House	Developed by team	Steel alloy	Custom	Reused	2020
Frame	Wimarol - machining	Developed by team	POM	Custom	Reused	2020
Waterproof Housing	CNC Kramet - machining	Developed by team	AW-6060	Custom	Reused	2019
Waterproof Connectors	TopService - machining	Developed by team	AW-6060	Custom	170 $ (650 PLN)	2022
Thrusters	Blue Robotics	T200	T200 spec.	Purchased	Reused	2019
Motor Control	Blue Robotics	Basic ESC + custom board	ESC spec.		Reused	2019
High Level Control	-	Custom	PID		-	-
Actuators	Pololu	172:1 Metal Gearmotor 25Dx71L 4828	Pololu 4828 spec.	Purchased	35 $	2021
Propellers	Blue Robotics	Included with T200 thruster	T200 spec.	Purchased	Reused	2019
Battery	GRALMarine	Custom	Battery spec.	Purchased	Reused	2019
Converter	-	-	-	-	-	-
Regulator	Murata; Recom	UWE-12/10-Q12; RPA60-2405SFW;	UWE-Q12 spec.; RPA60-2405SFW	Purchased	Reused	2020
CPU	ST; NVIDIA	Nucleo-F767ZI MCU; Jetson AGX Xavier 16GB SoC	STM32 Nucelo F767ZI spec.;; NVIDIA AGX Xavier spec.	Purchased	Reused / 35 $;	2019
Internal Comm Network	Texas Instruments	CAN bus transceivers + custom modules	SN65HVD230D spec.	Custom	20 $	2019
External Comm Interface	TP-Link	TL-WR802N v4	TL-WR802N v4 spec.	Purchased	Reused	2019
Compass	Xsens	MTI-30	MTI series X-sens spec.	Purchased	Reused	2019
Inertial Measurement Unit (IMU)	Xsens	MTI-30	MTI series X-sens spec.	Purchased	Reused	2019
Doppler Velocity Log (DVL)	Teledyne Marine	Wayfinder DVL	Wayfinder spec	Purchased	11029 $ (42127 PLN)	2020
Manipulator	-	Developed by team	three-finger or two-finger configuration	Custom	150$ (570 PLN)	2022
Algorithms: vision	-	-	Convolutional neural nets, edge and colour detection, Hough transform	-	-	-
Algorithms: acoustics	-	-	Phase difference, neural networks, onset detection	-	-	-
Vision	Logitech: Arducam;	C922 Pro Stream Full HD; OV2710 2Mpx 1/2,7";	Logitech C922 Pro spec. Arducam OV2710 spec.;	Purchased	Reused	2019
Acoustics	Teledyne RESON	TC4013	TC4013 spec.	Purchased	Reused	2019
Localization and Mapping	-	-		-	-	
Autonomy	-	-	YOLOv5, PyTorch	-	-	-
Open source software	-	-	YOLOv5, PyTorch Unity ML Agents Toolkit, Pytest framework	-	-	-