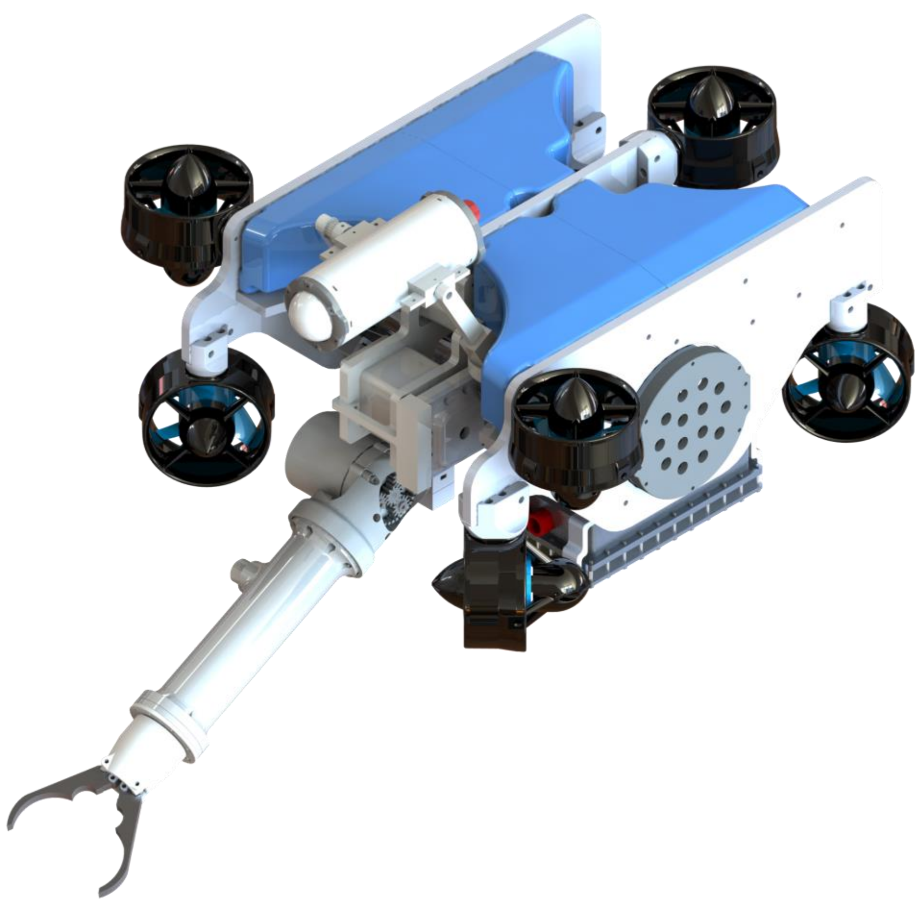
MATE ROV COMPETITION

POLITECNICO DI TORINO

POLITOCEAN

TECHNICAL DOCUMENTATION 2019

AIDA



MENTOR: Professor Claudio Sansoè

MEMBERS	TEAM ROLE	ACADEMIC YEAR
Balvis Massimo	CEO and CFO	4th Academic Year
Calasso Giuseppe	Mechanical Chief Engineer	3th Academic Year
Ruggiu Riccardo	CTO and Electronic Chief Engineer	5th Academic Year
Stagi Roberto	R&D Responsible and Software Chief Engineer	4th Academic Year
Scantamburlo Damiano	Safety Officer and Electronic Engineer	5th Academic Year
Sivieri Flavio	Pilot and Fluid Dynamic Supervisor	5th Academic Year
Martellone Lorenzo	Software Engineer	2th Academic Year
Gallotti Davide	Software Engineer	5th Academic Year
Sabino Nicola	Software Engineer	5th Academic Year
Cota Davide	Software Engineer	5th Academic Year
Pettinelli Angelo	Software Engineer	3th Academic Year
Iurlaro Gabriele	Software Engineer	1th Academic Year
Alasio Matteo	Electronic Engineer	4th Academic Year
Sciortino Dario	Electronic Engineer	5th Academic Year
Bartolotti Giulia	Electronic Engineer	3th Academic Year
Silvestro Simone	Electronic Engineer	5th Academic Year
Pertosa Federico	Electronic Engineer	5th Academic Year
Ronco Andrea	Electronic Engineer	3th Academic Year
Musso Emanuele	Mechanical Engineer	3th Academic Year
Pavone Federico	Mechanical Engineer	2th Academic Year
Greco Tommaso	Mechatronic Engineer	5th Academic Year
Salvatelli Michele	Mechatronic Engineer	5th Academic Year
Rosato Giovanni	Mechatronic Engineer	5th Academic Year
Grilli Gianmarco	Aerospace Engineer	2th Academic Year
Valpreda Adele	Materials Engineer	4th Academic Year
Masnovi Patrick	Graphic Designer	3th Academic Year

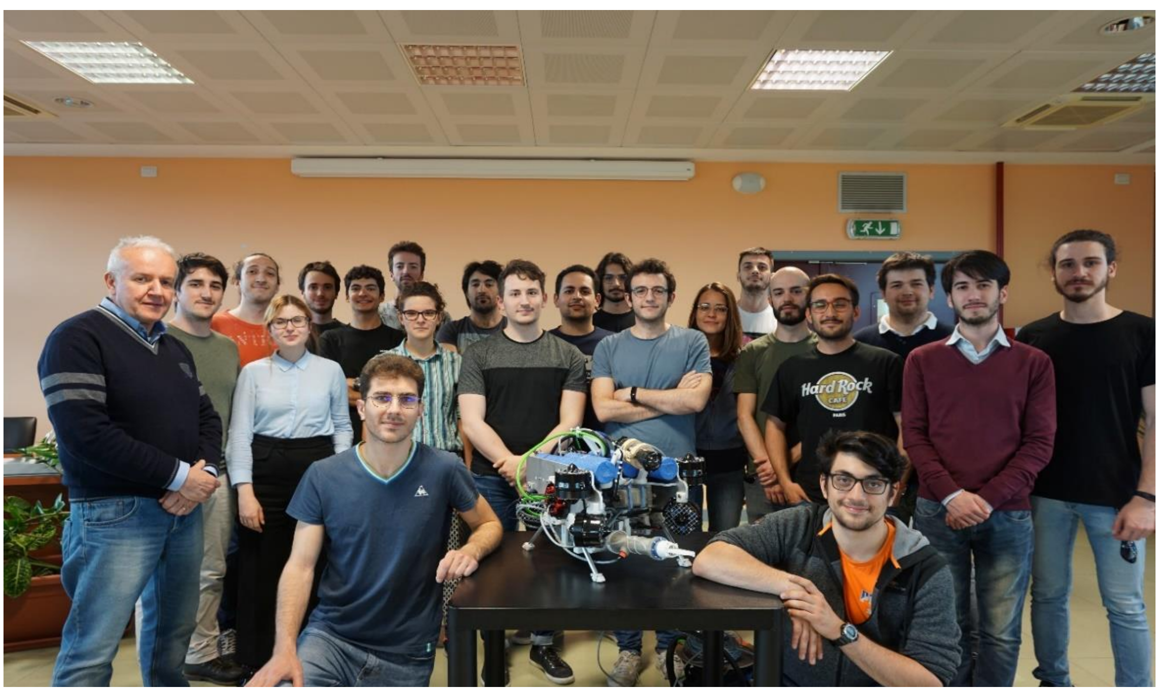


FIGURE 1 – TEAM'S PHOTO

I. Abstract

Team PoliTOcean was founded in May 2017, since that year we have developed three ROVs. At the moment the team is made up of twenty-eight enthusiasts students specialized in different academics' fields that have used their skills and their past experiences to make our best ROV.

Aida is the last ROV made by PoliTOcean, developed with the aim of the participation to the 2019 MATE ROV Competition.

Aida is a small ROV equipped with different tools and features developed to complete the tasks required in the competition. It is a powerful instrument to inspect and repair different parts of a dam, like locating cracks or to repair a trash rack and if you want to easily inspect inside a pipe, you can do it, thanks to the micro-ROV installed on it. It can also be used to maintain the healthy in waterways by analyzing water using its accessories, by determining habitat diversity and restoring fish habitat. Other features concern preserving history, it is the perfect device if you have to recovery cannons from the depth, indeed, it can lift up to 120 N using a lift bag. If you don't know the weight of the cannon, don't worry, Aida can measure the volume and by the density, users are able to calculate the weight of it.

This paper describes the process used by mechanical, electronic and IT areas to create the ROV Aida, started with an idea based on the experience gained during the develop of our previous ROVs and ended with its creation.

Table of Contents

II. Core Design

2.1 Mechanical

2.1.1 Manipulator

Aida is equipped with a manipulator, designed and developed by the team, according to the specific request for the different tasks.
The manipulator is an essential thing in the MATE ROV Competition because of its versatility and easiness way to perform many tasks.
Manipulator's design started considering the tasks required in the competition and analyzing the manipulator of the previous year, substantially, it has the same components and the same degrees of freedom.
Therefore, like the previous year, the main components of the manipulator's assembly are four: a shoulder fixed to the chassis, the arm, a wrist and a claw.
The total volume occupied by the manipulator has been substantial reduced, in order to respect the maximum dimensions of the ROV imposed by the competition's manual and to decrease the production's costs.
The shoulder allows pilot to change the angulation of the arm, that ensure an easier and a more accurate way to install and to position tools, accessories and components which is requested the movement during the perform of the different tasks.
It contains one stepper motor waterproofed using a special rotary seal installed on the shaft, customized for our application by a company placed in the Turin's area.
The shoulder is linked to the arm through a gear system, installed to increase the torque of the motor. Iit guarantees an output torque of about 2 Nm.
The arm enclosure contains a stepper motor and a micro actuator, the first one used for the wrist rotation and the second one to open/close the claw. It has been waterproofed in the same way of the shrouded, using the same rotary seal.
The shoulder and the arm, where a watertight enclosure and a high precision is needed, have been manufactured in resin by using a 3d printer. We have evaluated other ways of production, but the used technique has been the better choice because of the lower costs, lower production times and greater design freedom compares to a classical production method. The gear system, the wrist and the claw are manufactured by FDM 3d printer, using a filament suitable for underwater applications.
Claw has been designed to grab things in an appropriate way, indeed, the internal shape of it follows the external shape of tools and accessories used to perform the tasks.

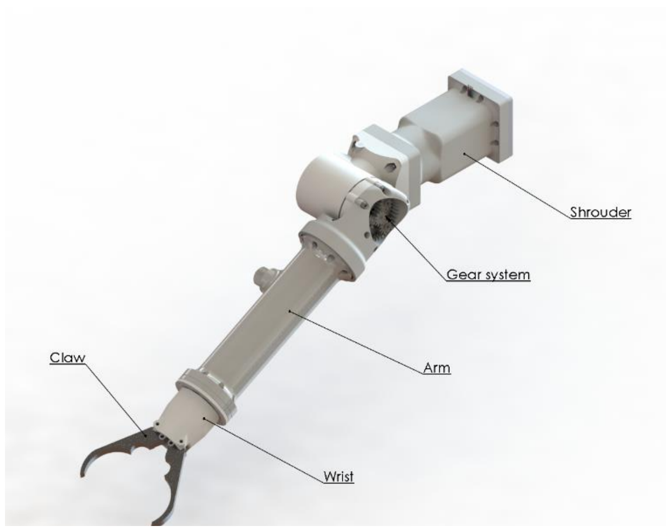


FIGURE 2 - MANIPULATOR RENDER

2.1.2 Chassis

This year in preparation for 2019 Mate Rov competition and after some reunions we have decided to redesign the chassis of our rov.
It is totally different from previous version because we have decided to adopt different materials, new layout for motors and electronics.

The new chassis is built in aluminium instead hdpe because alu makes the stucture rigid and ligth, infact it is 20% ligther than previous version.
There is a new distribution for thrusters indeed we adopt a seven thrusters layout instead the 4 x 4 motors layout, this solution have some advantages for example a less weight and a great maneuverability.
This great maneuverability is the consequence of an optimized distance between each motors that have the task to move the Rov in right-left and forward-rear direction.
During the design process we have considered a 2 x 4 motors layout but after some evaluations we abandoned it because 2 instead of 3 motors could produce instability in the Rov alignment.
We think that three motors compensate in right way random movement introduced for example by robotic arm, harmful for autonomous guide task.
During design process we adopt different instrument to reach an optimum product, for example CAD for 3d model and CAE for stress analysis.

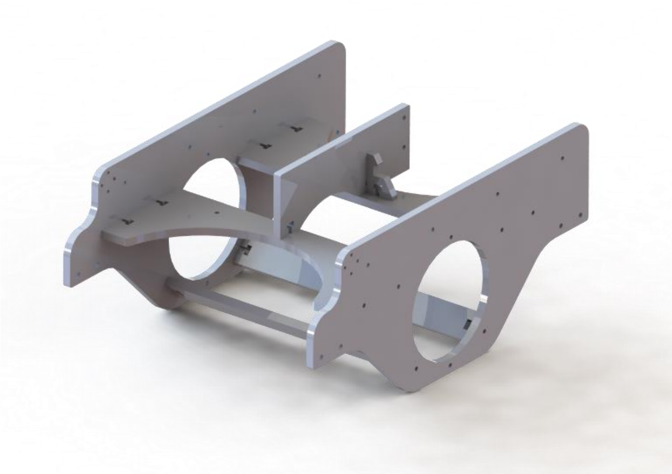


FIGURE 3 – CHASSIS'S RENDER

2.1.3 Buoyancy

The Rov's buoyancy is studied to make a neutral alignment in water. We reach it with an experimantal method: we have built a test bed with 4 load cells connected to each corner of Rov when it is in the pool and in this way we have found the weight distribution of our robot.

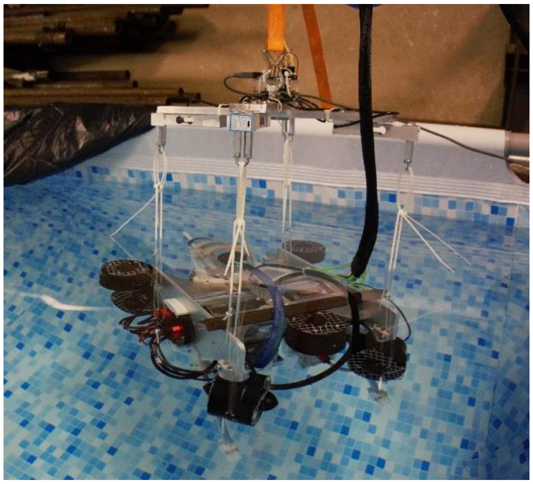


FIGURE 4 - ROV DURING WEIGHT DISTRIBUTION ANALYSIS

2.1.4 Electronic enclosure

As for the electronics container, like the previous year, we decided to install a 4'' acrylic tube purchased from BlueRobotics.
This year the tube has been mounted transversely to the chassis, to leave an empty space in the back and front the ROV, in order to install other accessories like the tiltable camera on the front and the cable reel on the back, used to connect the micro-ROV to the ROV.
Another big upgrade of this year has concerned the installation of a second electronic enclosure, called power PCB box, made in aluminum by CNC milling machine, installed on the bottom of the ROV. The power PCB box works like a big heat sink and avoid overtemperature inside the container, hence, allows power converters to work correctly.

2.1.5 Accessories

A series of accessories has been installed to facilitate the pilot during the ROV's operations.

This year the camera has not been installed in the main electronic tube, but it has been built an independent tiltable system.
The mechanism has been developed to enable the pilot to have a wider view of the ROV's arm and the surrounding environment. The system has only one degree of freedom and it is moved by a pulley and a belt connected to a stepper motor. The enclosure, that host the camera, is a 2" acrylic tube with its relative end caps made by Bluerobotics, while all the other mechanical parts are 3D printed by the team. By using this device, we can do all tasks using only one camera, avoiding the installation of another camera on the arm.

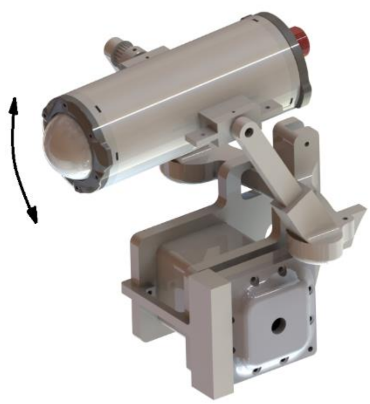


FIGURE 5 - TILTABLE CAMERA SYSTEM

The Rov is also equipped with a lift bag that can be use to lift and transport object to the surface.
To be sure of securely grab objects of various diameters a system of two grappling hook were designed and assembled to the lift bag.
The Rov drags the hooks close to the object, once the hooks are in the correct position pneumatics inflate the bag and the hook starts to close, the Rov releases the the hooks and then the lift bag floats to the surface where can be retrieved.

2.2 Electronic

2019 Aida's electronic design has been strongly focused on solving the main heat dissipation issue that in Nereo lead to high uncontrolled pressure inside the core electronic tube and a fatal opening of it.
The problem has been addressed by separating the power-board in a dedicated external box: this has been shaped on it so that a DC-DC converter built-in metal extension touches the aluminum power box, for an enhanced and uniform dissipation through water contact. An external temperature sensor has been further added on this box.
Regarding the main electronic enclosure improve space management has been our second aim. With the help of Solidworks and Fusion360 tools we considered different PCBs placements alongside understanding the number and the functional division of the boards needed to address the competition tasks. Our final solution provides the ThinPad, a round PCB connected both with the external power-board and the internal ESCs PCB. Through this connection we manage to supply the ROV and the thrusters without further cables. More cables are avoided with a socket connector that allows a neat link between ESCs and a Control

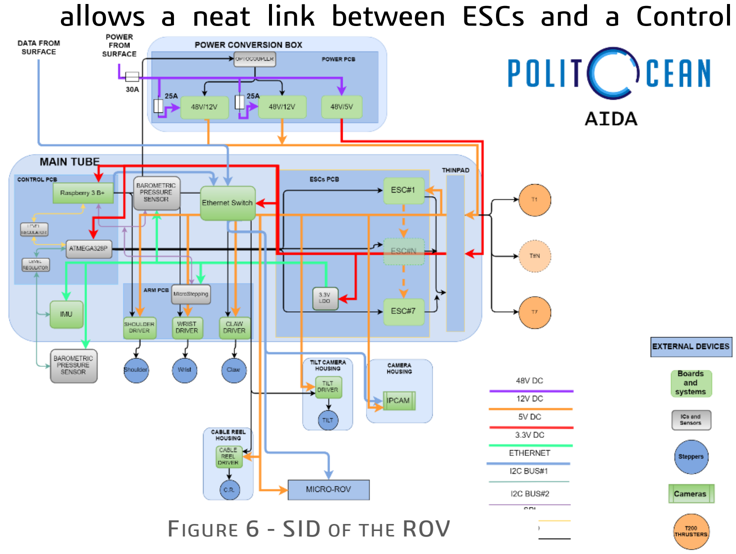


FIGURE 6 - SID OF THE ROV

PCBs, the thinking-core of our ROV. This two boards represents a further modular improvement from the last year only board from a functional and space management point of view. They let us add to the family an Ethernet switch, strongly required for increased data rate needs of this year real-time cameras.
The electronic set is completed by an Arm PCB and two simple Camera and Cable Reel PCB, placed outside the main tube with the functionality of managing the main camera tilt movement and the cable reed, needed this year for the MicroROV.

2.2.1 Power conversion box

This year one of the bigger updates is the developing of a separate Power PCB. In fact, we chose to build a PCB with the specific purpose of manage the power supply of the entire ROV. This choice gives us the possibilities to create a proper dedicated dissipation system. To provide all the power possible with the 48V 30A input we use (like the year before) two 48-12 DC- DC (600W each) converters working in parallel. However, this time we choose specifically DC-DC that comes with dissipation plate build in. Dissipate the power has been an important task for us this year: the amount of power consumption is about 1.2kW. With higher temperature the efficiency decreases and so the power dissipated and the internal temperature increase accordingly. As a consequence, we design an aluminum enclosure which is directly in contact with the metal thermal dissipation panel of the DC-DC. Clearly, we also use thermal paste to improve the thermal exchange and the final result is very good: the aluminum case is directly in contact with the water that functions as an immense bin for the thermal exchange; so in the end we achieve e decrease of the temperature of at least 60% compared to last year, with the consequential increase in efficiency term. In this way we can give to the thrusters all the available power from the surface that means a faster ROV. We provide the power supply conversion box with a digital temperature sensor that communicates the internal temperature, in a first moment it was an analogic sensor but due to the parasite capacitance of the wire in the water we need to switch it with a digital version.
However, we still maintained some of the good design choice of the last year and implement some other. We chose to maintain the optocoupler to provide galvanic insulation for the 12V. This is important also to have the possibility of switch off all part working at 12V. We also chose to maintain the input circulation diode to manage an inversion voltage dropout. This year we provide two different and separate input circuit for each 48/12 DC-DC, this design has one principal advantage: the possibility to insert one fuse on each DC-DC input to provide an even greater insurance for the circuit. We also paid more attention to design the PCB planes, incrementing the clearance and dividing as much as possible the input and output part to respect even more the design rule of a power PCB. Moreover, we designed an in-build trimmer to regulate the output of the 48/5 DC-DC to improve one of the last year troubleshooting. In addition, 5V are always on, the DC-DC of 5V are never switch off, because the logic part needs to be always on.

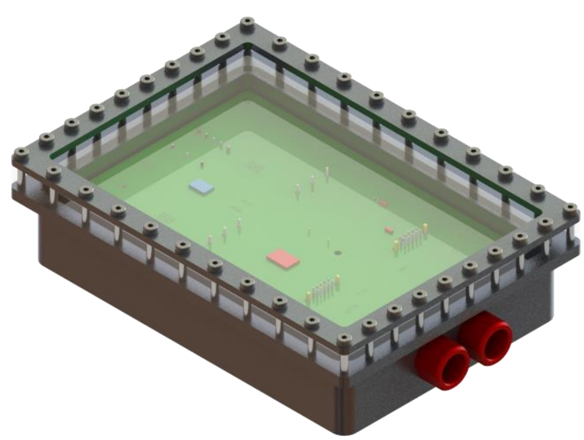


Figure 7 - Power conversion box

2.2.2 Main tube

Another main improvement of this year was the development of a panel to connect the converted power coming from the power box and the phases of the seven T200 thrusters. In this way we removed the 30% of cables (comparing to Nereo's design) inside the main tube, and each pin of the microcontroller is uniquely associated with its correspondent thruster. A very challenging task was to find the optimal connectors in terms of easy of assembly and security, ensuring that they support the very high currents. We came up with two different connectors in one single panel, one for the 12V and another for the lower currents, such motor phases and logic power.

ThinPad is connected from one side with the thruster phases and main power, with their cables directly soldered on it, and, from the connector side, to the main block composed by ESCs PCB and the Control PCB lodged on the top of it. The ESCs PCB embeds seven BlueRobotic's Basic ESCs, a voltage converter and three status led. Its connected directly with both ThinPad and Control PCB. It gives to the control the power and receives from it the signal for the ESCs. It also has seven JST connectors to give the power to the rest of the main tube, such as the arm PCB and the ethernet switch, and to the external devices (cam, microROV, tilt mechanism and cable reel).

Thanks to Thinpad we can easily access inside the tube removing only one flange without manually remove the cables, which means a lot of time saved and a more reliable connection, since last year's PCB screw terminal blocks caused many problems.

The Control PCB is the thinking-core of our ROV, containing all the logics and the computation modules responsible for balancing and motion, as well as the communication and the diagnostics of the machine. We designed this board starting from the last year design, and we made some serious improvements, with the aim to reduce the size at the minimum, enhance performances and remove all the unused hardware. The low-level real time computation is performed by the ATmega328P microcontroller, which is displaced on the bottom side of the PCB through the appropriate socket. This allows flexibility and a quick replacement of the microcontroller in case of failure. The ATmega has its own I2C network, connecting the external barometer and the IMU (Inertial Motion Unit). These are the components responsible for handling the ROV's motion.

The main software runs on a Raspberry Pi 3 model B+, which fits on the top side of the PCB. Among the many tasks, the Raspberry generates the controls signals for all the other PCBs, including the controls signals for the arm and the 12V power-up signal. It also performs the diagnostics, collecting data about the temperature of the housing for power circuitry, as well as the pressure inside the main electronics chamber.

The pressure is measured through a BMP280 directly integrated on the board. We are particularly proud of this enhancement, since last year the uncontrolled chamber pressure caused a fatal failure, a problem we should tackle with this monitoring. Furthermore, integrating this particular chip has been quite a challenge, because of its small size and embrittlement.

The two computational nodes are connected by a point to point SPI network, which ensures a high speed and robust communication between these fundamental nodes and allows us to update the microcontroller firmware directly from the Raspberry Pi.

All the routing on the PCB where designed carefully, trying to avoid tight corners which could have increased the reflection coefficient of the lines. Particular attention has been paid to the I2C lines, since their high frequency protocol and their importance make them the most critical part. This year more than ever the room at our disposal was low, hence the displacement of the connectors was a critical matter. We managed to reduce their

number to the minimum, displacing them strategically as close as possible to the appropriate device.
The design of the PCB in charge to control the mechanical arm was changed only in terms of shape and dimensions, keeping the same functionality of the one inside our previous ROV Nereo. The first idea was to separate this board from the main tube in order to increase the ease of work on the entire arm, but doing a separate waterproof chamber was too costly, so we came up with a compromise placing this board inside the main tube in such a way that we can easily work on the drivers mounted on it, allowing the replacement of them in a very short time. To optimize the space inside the main tube we changed the shape of the board, creating an L-shaped PCB. On this board there are three drivers, one for each motor of the arm and an IC connected on the I2C bus that allows to set the resolution of the steppers.

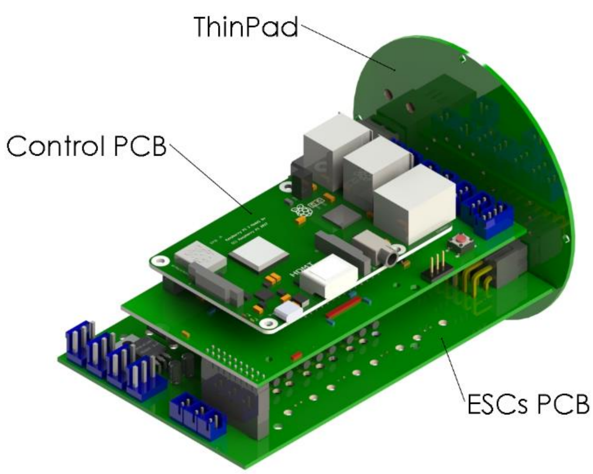


FIGURE 8 - RENDER OF THE MAIN ELECTRONICS BOARD

2.2.3 Tether

The ethernet switch is needed to connect multiple devices with a single cable to the surface. To this switch there are connected: Raspberry, the micro-ROV and the IP camera, there is also a slot for another camera, if needed. We bought a very small switch that ensures the data stream up to 1Gb/s.

The main tether is composed by two cables united wrapped together with a cable sock, one that gives the power from the surface and one for the data communication. The first one was chosen looking for the copper size dimensions, the second one looking for the data transmission rate, without ignore the flexibility of both. The power cable has a copper size of 2.5mm2 that guarantees us a proper power transmission and the data cable chosen is an Ethernet cable Cat6E that allows data transmission up to 1Gb/s.

2.2.4 Accessories

One of the Aida's tools we developed is the one to measure the pH and temperature of a solution. We opted out the idea of an on-board device because of the very specific nature of the task, choosing instead to produce an external tool to be handled by the ROV itself. Hence, we designed a small custom PCB which integrates both the functionalities.
Measuring the PH precisely is a very hard task because of the nature of the measurand, hence we decided to buy a PHmeter, instead of making one out of scratch. This sensor came with a board with a calibration trimmer and an analogic output. The temperature sensor is based on the TSY01from Measurement Specialties, customized and waterproofed by BlueRobotics.
The architecture is based on an Arduino Nano, which runs a modified version of the software provided by the pHmeter vendor. The pH value of the solution is read the data are collected by the microcontroller, adjusted with the calibration parameters and then sent to surface through a serial connection.

The board is supplied with 12V coming from surface, which are then reduced to 5V using a LM7905 to supply both the Arduino and the temperature sensor. A 2 A fuse provides safe shutdown in case of failure.

Another Aida's tool is the metal detector. The development of it began with analysing the physical problem. The main issues we have to deal with have been those related to the dimension of the device. We have noticed that using a very large diameter inductor, the detection of the metal is much more sensitive than a smaller diameter, and therefore a compromise for the dimensions has been made, reaching an acceptable size, where the coils dimension are still enough to have the sensibility to detect metals.

A first prototype was developed on a breadboard just to demonstrate the correct behaviour of the physical principle and an optimization process was done on the dimension of the coils, giving us a second PCB prototype.

The Arduino placed on that PCB converts this signal in a digital one, noticing us the presence or not of the metal, sending this information to the control station, in order to display it on the screen. This second prototype was tested for a long time, without positive results. This caused a delay on the mechanical enclosure design and production. We found an alternative solution, buying a kit which consists in a very small board that detects the presence of metal, lighting up a led. We adapted this kit to satisfy our needs, connecting this led output on a waterproof external led provided by BlueRobotics.

2.3 Software

2.3.1 Concept

Aida's communication system is based on an IP computer network, which means that the different components have their own IP address. For this reason, a Raspberry Pi, a system which is able to interconnect over an IP network, has been chosen to be the main component on both the micro ROV and the main ROV itself. To complete the picture, a NVIDIA Jetson TX2 board, thanks to its computational power, especially in image processing, has been chosen to be the computing system of the control station, while an ATMega328 is the controller in charge of reading the sensors and controlling the thrusters.

The communication at a high-level relies on the MQTT protocol. The first solution, inherited from last year software system, provided ROS (Robotics Operating System) as the base of the whole IT infrastructure. The reason we got over it, even considered how powerful it could be, is basically because of some lack in performance for some over-network communications we noticed last year. Indeed, migrating to MQTT, the system hasn't lost its modularity, but could visibly increase in performance. For the same reasons, all the programs have been written in C++, instead of Python, in order to make much more efficient programs, thanks to the fact that it is a compiled programming language and not an interpreted one like Python.

2.3.2 Communication with MQTT

MQTT is a low-latency, low-bandwidth, publish/subscribe based messaging protocol, the idea behind its use, is quite similar to the one behind ROS. That parallelism let us re-use old code, by just re-platforming the communication interface. The basic logic is that all the messages are exchanged over 'topics', and all the components act as publishers or subscribers to those topics. Each component participating in the connection can receive data subscribing in a topic, moreover it can send data publishing on it. The protocol ensure that all the listener subscribed to a specific topic receive the messages published on it. All the components communicate using dedicated topics for each main function (e.g. shoulder, wrist, hand, ATmega, micro ROV, errors, states, info messages, etc.). Like all publish/subscribe protocols, MQTT needs a broker. Mosquitto has been chosen as MQTT broker for Aida's communication system.

Thanks to MQTT's modular nature, it has been possible to write different modules for similar tasks. For example, as it will be explained later, even if it could be possible to integrate the joystick reading and the commands forwarding operations in a unique program, it has been decided to split them in order to gain flexibility in using them over the same device. It has been very useful for testing purposes, because the only constraint was to have a connection to the system network and to the right MQTT broker.

Furthermore, the broker location, in terms of device where it runs on, has been object of study. The final solution provides three different MQTT brokers: two for commands, where one resides on the main ROV and the other one on the micro ROV, and one for messages of all kind (joystick, info messages, sensors and components states), which resides on the control station. The two for commands have been chosen to be on the respective devices for performance and safety reasons. Indeed, in this way all the modules which receive commands have to subscribe to the broker which resides on their own system, avoiding disconnection issues which can affect the correct behaviour of the programs (even if a reconnection procedure has been implemented, it's safer to don't make them to disconnect). Moreover, MQTT clients periodically "ping" the broker they are connected to, which is obviously faster if it's on the same device.

To implement the communication over MQTT, a custom library, which relies on the library libmosquittopp, has been developed.

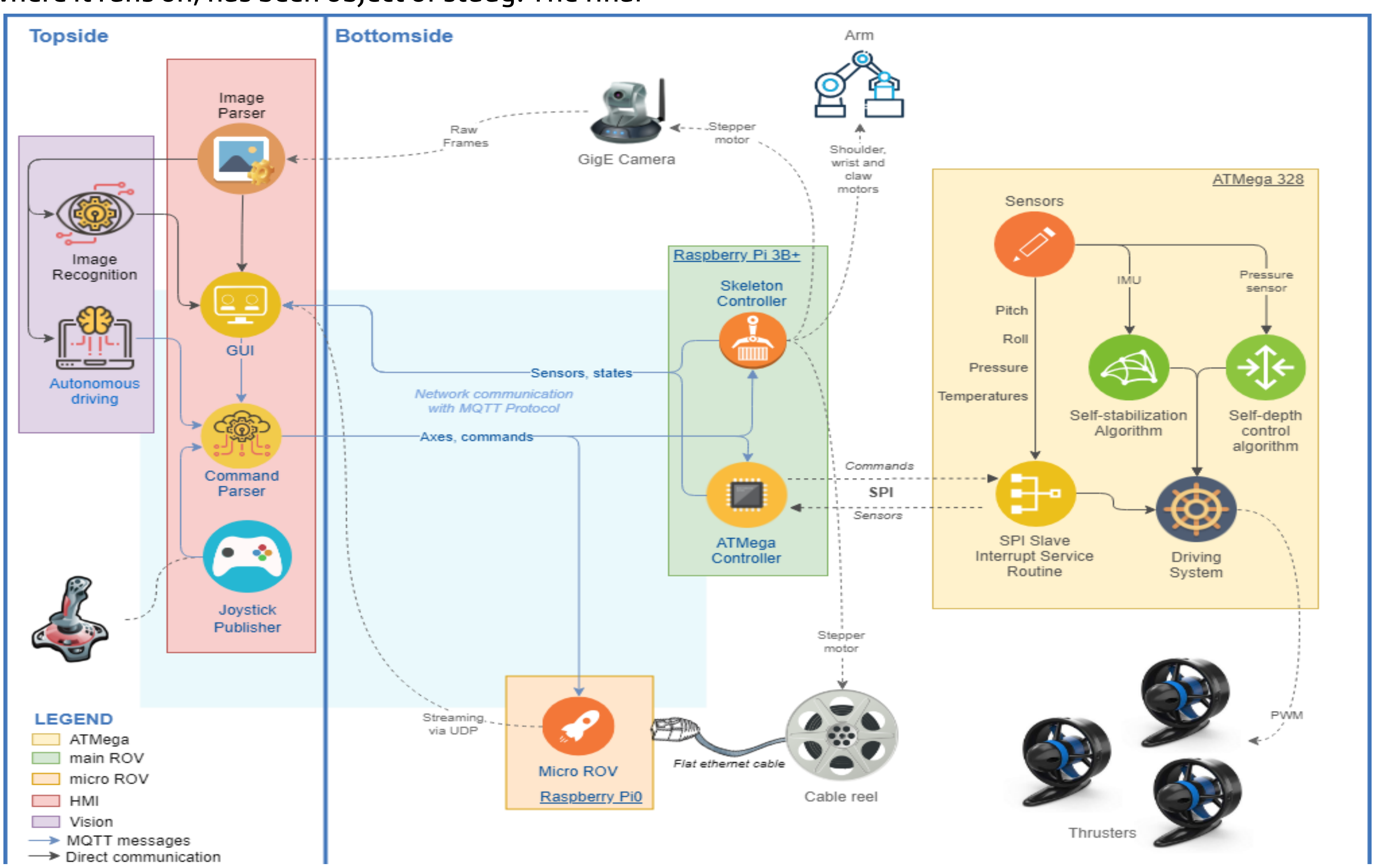


Figure 9 - IT architecture diagram

2.3.3 GigE Camera and Image parser

The reason behind the choice of using a GigE Camera is to overcome the important delays in last year's solution (~ 700 ms). It allows us to retrieve the frames uncompressed, in order to make all the conversion and compression operations on the control station, which let us achieve the superb result of ~ 40 ms of delay.
To accomplish image parsing operations, the libraries FlyCapture and OpenCV have been used. The former has been used for the direct acquisition of the images from the memory buffer of the camera.

2.3.4 Joystick Publisher

The Joystick Publisher is the module responsible for reading the data from the Joystick and sending them over an MQTT topic. It's connected to the Control Station's broker, in order to avoid flooding over the network for not needed messages. To read data from the joystick, the linux/joystick library has been used.

Since MQTT allows only the exchange of messages using strings, to send joystick's complex messages, we rely on the JSON standard. It is a standard format to encode more complex objects, consisting of attribute-key pairs, into a readable string.

2.3.5 Command parser

The Command Parser is the program in charge where all the data meant as "commands" are manipulated, translated and forwarded to the ROV. Those include all the data coming from Joystick Publisher, Autonomous Driving and GUI, all of them received using their own appropriate MQTT topics.
Several topics are used to forward messages too. Those are divided according to the data types (e.g. Commands or Axes) and to the components they belong to (e.g. Shoulder or Wrist).
Both for axes and for commands, in order to prevent network overloading, the publishing is done only when there is a change of their last input state.

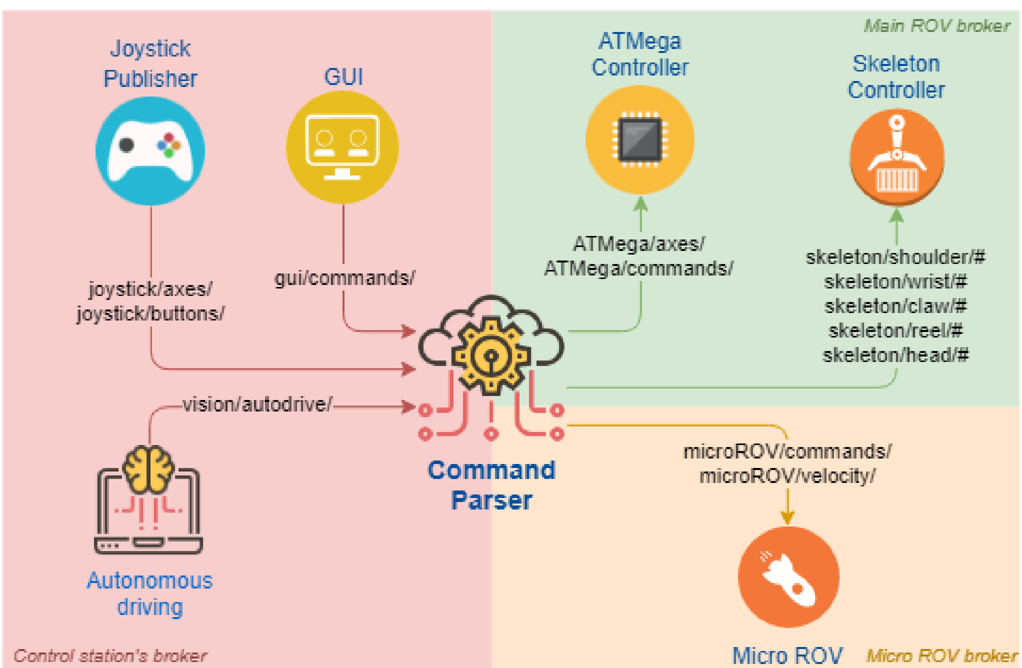


Figure 10 Command Parser conections summary. The arrows represent communications over MQTT topics

2.3.6 Skeleton Controller

The Skeleton Controller module is the one in charge of controlling arm, camera and cable reel motors. All of them are stepper motors, except from the one for the claw which is a DC motor. The module receives from the Command Parser all the commands for enabling, disabling and moving the components, as well as the direction and the speed of movement.
Since all the mentioned motors are attached to the pins of the Raspberry Pi board, their control is done using the C++ pigpio library, which has been used to implement some other custom classes.

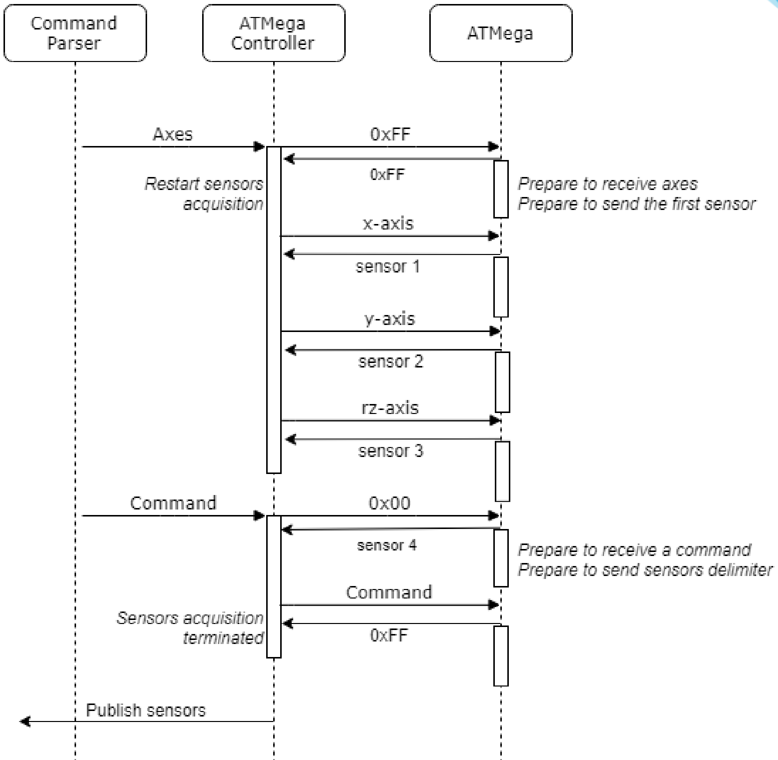


Figure 11 An example of the communication using the described SPI protocol with 4 sensors

2.3.7 ATMega Controller

The communication between the Raspberry and the ATmega controller has been implemented via SPI (Serial Peripheral Interface), which is a serial master-slave protocol. The Raspberry Pi has been chosen to be the master-side. The module in charge of the SPI communication is the ATMegaController. ATMegaController receives the axes and the commands from the Command Parser, manipulate them and send the correct command in a byte form to the ATMega using the SPI. From the master, the communication happens in two cases: axes update or commands forwarding. In both cases, a delimiter signal (0x00 for the commands and 0xFF for the axes) is sent before the main messages.
On the slave-side, instead, the ATMega has to respond to the SPI communication with the sensors' values. It sends cyclically all the sensors by means of an array of byte values. In order to make a robust communication, it uses a delimiter (0xFF) every time it's starting the communication again.

2.3.8 Self-stabilization, PID and sensors

One of the main Aida's features is the self-stabilization controller. This is carried out using a 6-axes IMU, which provides values for the accelerations and the angular velocities over or around the 3 axes. By means of trigonometric transformations and integral computation (for the angular velocities), we have been able to retrieve the values of roll and pitch. The integral computation needs a time constant over which compute the integral. This has been found experimentally, and it has been chosen to be 12ms. A timer is used to implement it in software, and all the computations follow that time. This way of retrieving roll and pitch has been inherited from last year's implementation of the self-stabilization controller.

To handle in an efficient way the problems of the self-stabilization and self-depth control correction, a PID controller has been implemented. In the first case, it takes from the IMU the values of roll and pitch, which are considered as the input-error from the desired orientation: the zero roll and pitch. In the second one, instead, the error that needs to be corrected is the difference in pressure between the desired one (the one in which the ROV has stopped) and the current one, continuously read by the pressure sensor.
All the sensors, which provide us with also the temperatures inside and outside the ROV, are read using I2C.

2.3.9 Driving system and current damping

The driving system is a simple linear combination of three axes coming from the main Joystick: x, y and z. The first two control front and lateral movements, while the third one is used for the rotations around the z-axis.
In handling the thrusters' power, considering some problem with the current intensity we had last year, we've decided to use an approach able to avoid peaks of current consumption whenever some strong variations in the axes occur. Without that expedient, the thruster could be asked to overcome an important torque all at once, which is the main reason for current peaks.
To accomplish this task, the requested power is not written immediately, but it gradually approaches to the requested value by means of a given step. That update is done automatically using one of the ATMega timers.
For reactivity reasons, self-stabilization and self-depth control correction powers don't follow the same "damping" process.

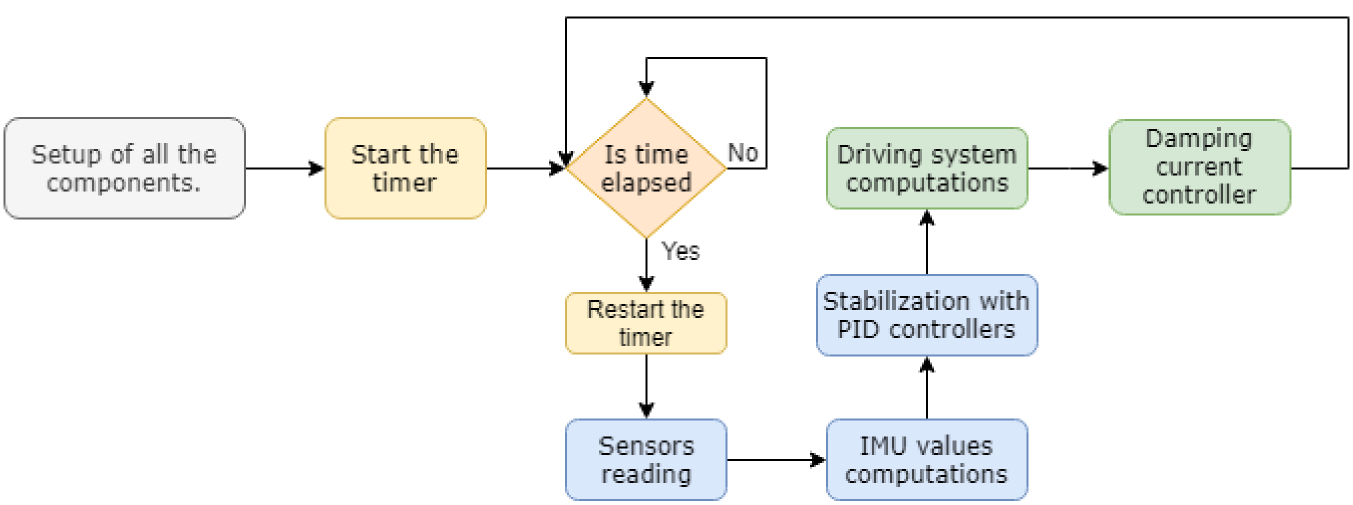


Figure 12 ATMega flow chart. SPI communication is asynchronous, so it is not mentioned here

III. Control Box

This year control station was improved in terms of dimensions and computing power. We decided to replace the Raspberry Pi with a more powerful device, a Nvidia Jetson TX2. This extra computational power is required to process vision's task such as the autopilot part, since this year we are getting the images directly from the camera. We increased the dimensions of the case, allowing us to incorporate a 15.6 inches LCD monitor and an ethernet switch that enables to connect to the ROV up to four technicians at the same time. This feature was very useful in the software development and debug phase to allow working on different devices simultaneously. On the control station there is also a plug to connect the external accessories of Aida such as the metal detector and the pH-meter. All components of the control box have been mounted to the external case by using 3d printed frames.

3.1 GUI

The GUI (Graphic User Interface) has been developed using the QT libraries. There are 3 main buttons, one for each competition task. Each of them leads to a specific layout, which provides the driver some task-related operations, in order to help him to accomplish the tasks more easily. Furthermore, there is a console connected with all the components, which keeps the driver updated with messages, errors and changes of the components states. The last ones are shown also using some images on another area of the GUI.

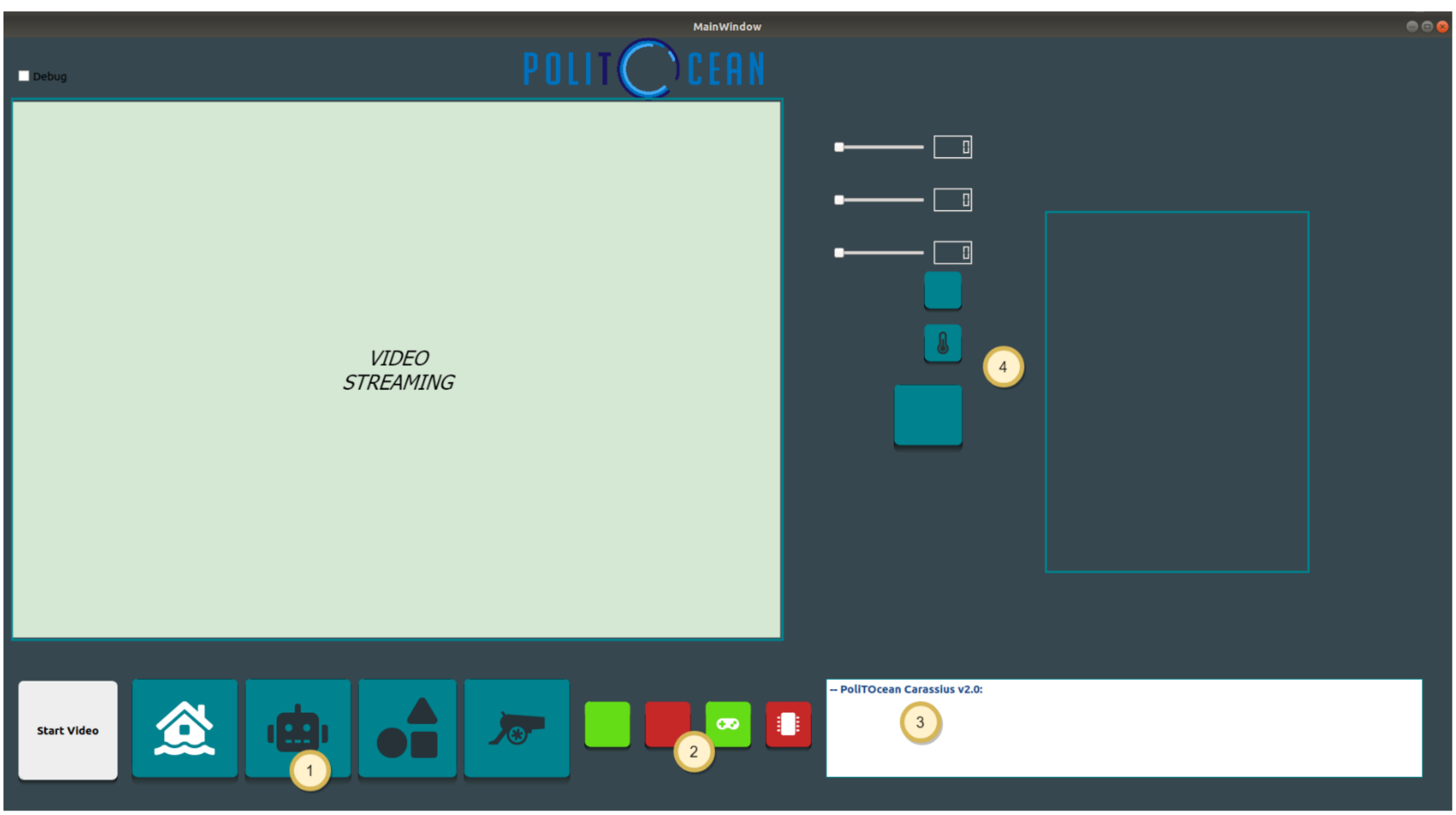


Figure 13 Graphic User Interface. (1) Buttons to switch among the main modes; (2) Components status; (3) System console; (4) Sensors and other info display

IV. Troubleshooting and testing

4.1 Video streaming

The new video-stream technology, has been very tough to handle with. It is a completely different way from last year, and we had to learn how to interface with a GigE camera and its proprietary, not so well documented framework.

4.2 Micro ROV computational power

The micro ROV's Raspberry Pi 0 has been a very difficult device to deal with. Its lack of power and interfaces (i.e.: the Ethernet port) forced us to use some tricks to make it work in a proper way. Indeed, we needed to use a Raspberry Pi 3 to install all the software in a faster way and we had to use an expansion device to let it connect with an ethernet cable. That expansion is very slow, so we had to deal with low bandwidth and high latency connection, over which we had to send a video streaming. We finally found a way of boosting the communication speed between the Pi 0 and the Ethernet expansion, so we could achieve a better result.

4.3 Software Testing

The importance of software testing has not been underestimated in this project. We used the library catch2 to implement some unit and integration tests for the main functions of the system.

We have been able to test also communication over network on different devices, which let us identify more easily where were the problems whenever they occurred.

4.4 Electronic tests

The methodology used for the testing phase of the electronics was quite similar to the one adopted last year. We tested each board separately in order to detect design or assembly issues, before mounting all together. In this way we found a problem with the mounting of the pressure sensor mounted on the Control board: we solved it desoldering the sensor and placing it better on the pins.
A very important test was made for the power conversion box: once mounted we connected it to a resistive load of 350W to test the dissipation in air.

With a thermal camera we monitored the temperature for about fifteen minutes, noticing a constant temperature inside the working range of the power converters. As mentioned before the analogic temperature sensor placed inside this box caused problems. We noticed that only after the first tests in the pool. The value red by the microcontroller oscillated a lot only when the ROV was inside the water, so we replaced it with a digital one, losing a little of test time because we had to open the power conversion box, anchored on the chassis, so it's not very easy to open.

V. Personal Challenges

This year one of the biggest challenges has been to find an appropriate place to test the ROV. After different requests we finally obtained the authorization to buy a pool and to install it inside the university.

5.1 Electrical

A very useful tool we learned to use this year was the Autodesk suite, focusing on Eagle and Fusion. The first one was very powerful for the design of the boards, allowing us to import directly from the web the footprints of the components. This was a time saving solution with respect last year. Fusion instead was a game changing tool that permitted us to create the 3D version of the boards. This was very useful to improve the space handling inside the main tube and in the power conversion box. Was also used to create the ThinPad and its joint. Another useful hardware tool used also this year was the Pick&Place machine which is the best tool to solder small surface mount components.

5.2 Mechanical

One of the biggest challenges of this year has been the studying of the micro-ROV cable reel. The study has taken us a lot of hour and has required a lot of tests to find the final solution. We have found the final solution after a lot of brain storming sessions

5.3 IT

The most difficult challenge of this project, that the team had to face, has been for sure finding a clever way of developing a modular, self-consistent system with its own libraries (some of them shared among the different modules) and easy to compile, test and run over different devices. The main achievement for everyone has been being part of such a pretentious project, using real engineering tools and methodologies, while handling with releases and deadlines. We learned how to properly use tools like git and CMake, which have been crucial for the successful outcome of the project.

VI. Safety

Safety is a key value at PoliTOcean. Each member is conscious about the hazards inside the workplace and knows how to deal with it following several safety's procedures. In order to guarantee the access to the workplace members must follow a safety course and have to pass a test.

All the machinery (such as milling machine, grinder, saw...) were only use by competent staff, some team mates who followed a brief training course. Fumes of soldering are carried up to a roof-mounted vent via ducting. Even when we use the 3D printer we pay attention to safety: members use gloves to pick the pieces from the plate, in order to avoid burns.

One of our main concerns was the possible uncontrolled motion of motors such as thrusters and arm parts during code debug phases. The simple solution of the optocoupler described above allowed us to safely modify the code, enabling the 12V output only after ensuring that no one was handling the ROV in any way.

Another safety feature is the monitoring of temperatures inside the main housings and internal pressure inside the main tube: when the pressure or the temperature reaches critical value we turn-off the ROV to prevent enclosure fails and possible water leakages.

The status leds inside the main tube and power box are useful not only for debug phase but also to check if different voltages are enabled. In this way we can have a check if the thruster are enabled or not.

Before every ROV launch a list of operations is followed (described in our JSA) in order to prevent shortages or other damages.

Figure 14 - Team member working in the laboratory

VII. Budget and Timeline

7.1 Budget

PoliTOcean estimated every year, in October, a required budget for the project. The Politecnico di Torino is our main investor, for this year has funded a total of 31.000,00 €.
Purchases have been very challenging because before buy items we have to issue a public call specifying the characteristics of the requested components, this process many times is very long. As shown in the tables the cost of the ROV and its Control Box is about 8.756,75 €.

Electronics	Costs
Passive components	250,00 €
Ics and sensors	150,00 €
PCB printing services	1.096,02 €
Power converters	350,00 €
Cables	150,00 €
Cameras	414,80 €
Total	2.410,82 €

Thrusters and motors	Costs
T200	1.443,00 €
Steppers and motors	423,65 €
Total	1.866,65 €

3D printing	
MJP printing service	700,00 €
Total	700,00 €

Mechanical	Costs
Electronic housing set	563,28 €
Chassis	247,00 €
Screws and nuts	500,00 €
Bearings and seals	250,00 €
Raw materials	159,00 €
Total	1.719,28 €

Control station	Costs
Nvidia Jetson	330,00 €
Plastic box	80,00 €
Total	410,00 €

Tools	Costs
Safety equipment	50,00 €
Tools	1.500,00 €
Glues	100,00 €
Total	1.650,00 €
	8.756,75 €

The following table contains the sum of money used for the participation to the competition.

Tools	Costs
Plane tickets	9.450,00 €
Registration Fees	300,00 €
Car rent	1.500,00 €
B&B booking	3.147,00 €
Total	14.397,00 €

As we can see from the previous tables, the amount of the money used by the team is about 23.153,75 €.
The remaining, not mentioned, amount of money has been used for purchases and investments that are not strictly related to the competition, such as: testing equipment, t-shirts, advertising materials and refunds of travel expenses for local events and companies visits.

7.2 Timeline

It has been very challenging to meet deadlines while developing the systems and training the new members of the team.

We started the current year's activity in September 2018. In that time, we planned the different activities and deadlines of the year. The first table shown the supposed timeline, the second table shown the effective timeline. As we can see it has been a delay in Chassis design, that also produced a delay in mechanical production and test. This year, to manage activities we have started to use productivity applications like Trello to limit and sometimes avoid delays.

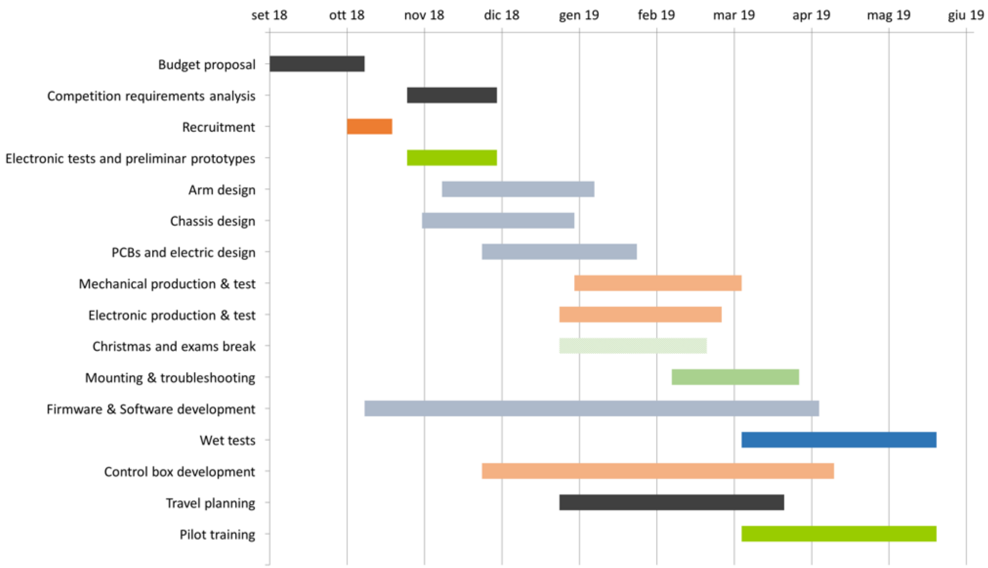


FIGURE 15 - SUPPOSED TIMELINE

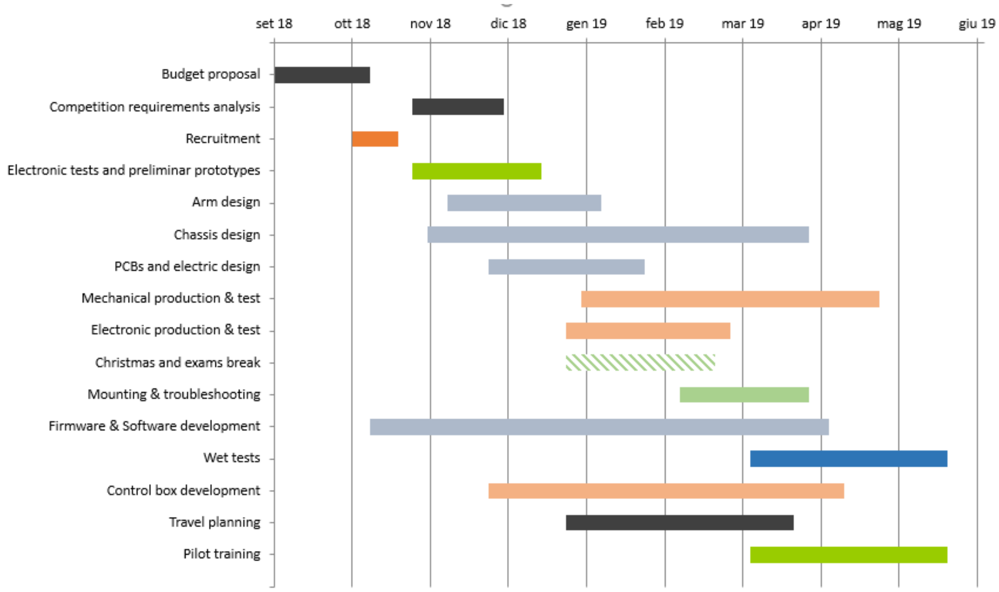


FIGURE 16 - REAL TIMELINE

VIII. Future Improvements

8.1 Electrical

An important improvement for the next ROV could be to dissipate better the heat generated from ESCs.

Another very important improvement that will help on the maintenance of the electronics is a better cabling system, reducing at the lowest the number of cables inside the main tube and using proper copper sizes for each type of connection. We are looking from next year to use a dedicated software to optimize the wiring. From the computational side, we have yet discussed to move from the AtMega328 to a more powerful microcontroller such as the ST32, and we will take a decision soon, after we have discussed all pros and cons.

8.2 Mechanical

This year we have redesigned the chassis from zero and we have mechanically studied it in order to obtain a lighter one. We have abandoned the optimization idea because of tight deadlines. A lighter chassis would be a great improvement.

8.3 IT

This year the project has just put its foundations. We want to start from this basis to improve all the functionalities, making them safer and more efficient. For example, we want to improve the communication robustness over MQTT and to implement some more complex and complete logging and testing techniques.

For what concerns the ATMega, we want to make the reading and related computations of the IMU axes separated by the management of the thrusters and reading of the other sensors. That can let us make a way more precise controller for the self-stabilization, as well as the chance of using a 9-axes IMU (with the magnetometer) in order to get the orientation referring to the North. In this way, we can also make the ATMega's firmware more crash-safe. Furthermore, we would like to find a better way to calibrate the IMU. This year has been made by recursive measurements of known angles, which is a fast and inaccurate method of doing it.

Moreover, this year the Jetson board hasn't been used at its full power. It provides some interesting GPU optimizations and computational power, thanks to the CUDA hardware architecture. One of the future improvements will be using this computational power for the image processing tasks. Some neural networks can also be implemented, if needed.

Finally, we are considering of adopting a TDD (test-driven development) methodology, since this year we have written the test cases after all the system was almost complete.

IX. Acknowledgements

- Politecnico di Torino for giving us the means and the financial resources needed.
- Mentor Prof. C. Sansoè, providing suggestions in technical fields and, most important one, helping us to deal with the bureaucratic affairs inside the University from two years.
- MATE Center for giving us a goal to aim.
- FiloAlpha an italian 3D printer filament producer that provided us a massive amount of filament to create our prototypes and not only

X. References

MATE. "MATE ROV Competition 2019 Explorer Manual"

BlueRobotics. "T200 Thruster Documentation"

BlueRobotics. "Basic ESC"

GE General Electric. "Barracuda III Series DC-DC Converter Power Modules"

SEACON "U-MATE Underwater Electrical Wet-mate Connectors"

BOSCH. "BMP280 Digital Pressure Sensor"

FLIR. "Blackfly GE"

PiGPIO Library

FlyCapture Framework

QT Libraries

OpenCV

Mosquitto and libmosquittopp